ITERATIVE ALGORITHMS FOR OPTIMAL SIGNAL RECONSTRUCTION AND PARAMETER IDENTIFICATION GIVEN NOISY AND INCOMPLETE DATA

(Vol. I)

by

BRUCE R. MUSICUS

S.B., Harvard College, Cambridge (1975)

S.M., E.E. Massachusetts Institute of Technology (1979)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 1982

Signature of Author ______________________________
Department of Electrical Engineering
August 31, 1982

Certified by ______________________________
Jae S. Lim
Thesis Supervisor

Accepted by ______________________________
Chairman, Department Committee

Iterative Algorithms for Optimal Signal Reconstruction and Parameter Identification Given Noisy and Incomplete Data

Bruce R. Musicus

Submitted to the Department of Electrical Engineering on August 31, 1982 in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering

ABSTRACT

In this dissertation we present a new approach to the problem of estimating multiple unknown signals and/or parameters from noisy and incomplete data. We approximate the various unknowns as being stochastically independent, then fit a separable probability density approximation to the given model density by minimizing the cross-entropy. Given the separable density, all the unknowns can then be estimated independently of each other using conventional methods. Surprisingly, all the well known Maximum A Posteriori and Maximum Likelihood methods for this problem can be viewed as degenerate forms of this cross-entropy approach, in which one or more components of the fitted separable density are constrained to be impulse functions. We solve for the Minimum Cross-Entropy and MAP separable density approximations by iteratively minimizing with respect to each unknown component of the density. This iterative approach takes a particularly simple form when the probability densities belong to an exponential class of densities. Each iteration decreases the cross-entropy, and convergence can be proven under mild conditions. Applications discussed in the thesis include:

a) grouped, truncated, quantized data

b) optimal signal reconstruction from time/frequency constraints
bandlimited extrapolation
phase-only reconstruction
magnitude-only reconstruction

c) multidimensional FIR filter design

d) multidimensional Maximum Entropy spectral estimation

e) optimal signal reconstruction from time/Short Time Fourier Transform constraints

f) penalty functions for constrained minimization

Supervisor: Jae S. Lim

Title: Associate Professor of Electrical Engineering

The only good thesis

is a finished thesis

Acknowledgements

Over the five years I have worked with the Digital Signal Processing Group, an enormous number of people have contributed to my personal and professional development. Most of all, I would like to thank Jae Lim and Alan Oppenheim for having the confidence in me to provide financial support and unlimited freedom, a heady combination that I doubt I will ever experience again. They encouraged and stimulated me, pressed me to improve my understanding and to test my ideas against realistic problems, and most important of all, convinced me to finish. My special thanks also go to Pierre Humblet, who served as a reader for the thesis, and whose timely and probing comments helped to substantially improve many of the arguments.

I'd also like to thank Monty Hayes, Victor Tom, Carol Espy and Naveed Malik for numerous thought provoking conversations which directly led to certain sections of this thesis. Steve Lang's comments were always cogent, and it was he who suggested looking at Shore and Johnson's paper on cross-entropy. Tom Bordley, Greg Duckworth, Webster Dove, and Tae Joo were always happy to discuss ideas, and suggested several improvements to the theories. Webster Dove deserves special thanks for his devotion to keeping the computer running. Andy Kurkjian and Dan Griffiths actually applied some of my ideas to new applications, and helped me to further extend the theory. Special thanks also to the entire Digital Signal Processing Group for an enjoyable and rewarding experience.

It all would not have been possible, of course, but for my parents, who have provided a lifetime of encouragement, support and love. Arlene Bernstein also deserves special thanks; her warmth and kindness have brightened my life, and her patience with my thesis-obsession and my peculiar hours is deeply appreciated. Finally, I'd like to thank the National Science Foundation for a three year graduate fellowship, and also thank Lincoln Labs and the various government agencies which have provided the contracts which supported me during the latter part of my graduate career.

Table of Contents

CHAPTER 4: APPLICATIONS IN STATISTICS

CHAPTER 5: APPLICATIONS IN OPTIMAL SIGNAL RECONSTRUCTION
PART I - BAYESIAN THEORY

CHAPTER 6: APPLICATIONS IN OPTIMAL SIGNAL RECONSTRUCTION
PART II - FISHER THEORY

CHAPTER 7: APPLICATIONS OF OPTIMAL SIGNAL RECONSTRUCTION
PART III - TIME AND FREQUENCY CONSTRAINTS

SECTION A - RECONSTRUCTION FROM TIME AND FREQUENCY CONSTRAINTS

SECTION B - LINEAR EQUALITY CONSTRAINTS

SECTION C - CONVEX CONSTRAINT SETS

SECTION D - NON-CONVEX CONSTRAINT SETS

CHAPTER 8: APPLICATIONS OF OPTIMAL SIGNAL RECONSTRUCTION PART IV - SHORT TIME FOURIER TRANSFORMS, MEM AND PENALTY FUNCTIONS

APPENDIX A: PROOFS OF THEOREMS IN CHAPTER 2

APPENDIX B: PROOFS OF CONVERGENCE OF ITERATIVE ALGORITHMS IN CHAPTER 3

APPENDIX C: PROOFS OF THEOREMS IN CHAPTER 4

APPENDIX G: INTRODUCTION TO PROJECTION MATRICES, EIGENSTRUCTURE OF LINEAR SIGNAL RECONSTRUCTION ALGORITHMS

Chapter 1
Introduction

1. The Subject of the Thesis

Using noisy and incomplete data to reconstruct a signal or identify parameters of the signal model are two of the most common problems in stochastic estimation theory. Applications abound throughout all engineering disciplines. In controlling a chemical plant subject to unknown disturbances and sensor errors, it is necessary to estimate temperature and pressure profiles (the parameters) as well as material flows (the signal) in order to maximize the yield of the reaction. Bandwidth compression or enhancement of noisy speech benefits greatly from accurate estimation of the vocal tract and voicing parameters. Optical images blurred by motion or by instrument inaccuracies can often be restored if an accurate estimate of the distortion is available.

In all these cases, we start with a model of the signal and observation processes. The model describes the inputs and outputs of the system, and mathematically characterizes its internal behavior as well as characterizing its overall environment. The model may be incomplete, with only vague information about the values of various internal parameters. Our measurements of the system may also be poor. Noise may be present, the values being measured may be distorted through transmission, samples may be missing, suspect, or coarsely quantized, and sometimes only short segments of data may be available. Given whatever information we have, the partial model and the partial data, our goal in all these applications is to try to estimate the unknown aspects of the model, reconstruct the internal state of the system, and try to fill in any missing observation data.

Unfortunately, deciding what is the "optimal" estimate of these unknowns depends strongly on how we choose to define "optimality". If we have a specific goal in mind, such as maximizing the plant yield or improving the intelligibility of speech, then optimality of the estimation algorithm can be measured in terms of the improvement in our application. Unfortunately, optimality criteria such as these are usually difficult to quantify in a form that is convenient for processing. A common approach, therefore, is to choose a method which is relatively simple yet works well (though perhaps not "optimally") in a wide variety of applications. Numerous techniques, both ad hoc and theoretical, have been discussed in the literature. Throughout this thesis we will assume that a statistical model for the unknowns is available. If a cost function is also given, describing the relative cost of various types of estimation error, then the "optimal" Bayesian estimation approach is to choose the estimate which, given the available data, would on average result in minimum cost [1] . If the cost function is the mean square error between the actual unknown and the estimate, then the resulting Minimum Mean Square Error estimate would calculate the conditional expectation of the unknowns. Unfortunately, while this may be the best one could do, the multidimensional integrals required to estimate several unknowns simultaneously are usually extremely difficult to evaluate.

The most commonly suggested compromises are Maximum A Posteriori (MAP) or Maximum Likelihood (ML) methods [1,2,3,4] . These approaches try to choose the values of the unknowns which are "likeliest" given the available observations. In effect, these methods replace the multidimensional integration of the Minimum Cost Bayesian method with a computationally simpler maximization of a probability density. Perhaps the most important property of MAP and ML is that, although they give higher costs than the "optimal" Bayesian approach, for many stationary and ergodic systems these techniques yield asymptotically consistent, efficient and normal parameter estimates [5,6,7] . In these cases, MAP and ML are "optimal" in the sense that no other asymptotically consistent technique can yield estimates with asymptotically lower variance. The problem is that for short data lengths, these methods may be quite biased. Moreover, when there are multiple unknowns, there are many ways in which we can apply these methods to the problem, some of which are significantly better than others. In fact, this thesis will treat three fundamentally different ways to apply MAP or ML to a stochastic estimation problem with two unknowns.

The main thrust of this dissertation, however, is to develop a new approach to the problem of stochastic estimation with multiple unknowns and noisy or incomplete observation data. What we would really like is a method which works about as well as the Minimum Cost Bayesian approach, but which doesn't require complicated multidimensional integrations. The real problem with the Minimum Cost approach is that the unknowns are usually closely correlated, and it is the interaction between all the unknowns which causes the computational complexity. If we could uncouple all the uncertainties and deal with only one unknown at a time, then the problem would be substantially simpler.

We start with what is admittedly a shaky basis. Let us pretend that all the unknown variables are independent, and approximate the given model probability density $p(\underline{x},\underline{y},\cdots)$ by a probability density $q(\underline{x},\underline{y},\cdots)$ that is separable. Thus $q(\underline{x},\underline{y},\cdots)=q(\underline{x})q(\underline{y})\cdots$ is simply a product of individual densities involving only a single unknown. Once we have computed such a separable approximation, we could then use it to estimate each unknown independently of the others. The question is, how do we best fit a separable density to the given model? The answer lies in information theory. Shore and Johnson [8] have proven that the "only" method of stochastic inference which correctly incorporates knowledge about the form of the probability density is to choose the density which minimizes the "cross-entropy", an information-theoretic measurement of the difference between the original and the new model density. Our Minimum Cross-Entropy Method (MCEM) thus consists of finding the best separable approximation to the given model density by minimizing the cross-entropy over the infinite dimensional space of separable probability densities.

Surprisingly, all the usual MAP and ML methods can be reduced to degenerate forms of this single cross-entropy method, in which we not only fit a separable approximation to the given model, but also insist that one or more components of that separable approximation be impulse functions. Cross-entropy thus serves as a unifying framework for stochastic estimation of multiple unknowns; it provides a concrete measure for comparing the various MAP methods, and also suggests that better, lower cross-entropies could be achieved by removing this impulse function restriction of the MAP methods. Cross-entropy, like Minimum Cost Bayesian estimation, can also deal properly with generalized probability densities containing impulses. MAP and ML cannot. The cross-entropy approach also tends to retain any symmetry in the underlying model, while the MAP methods often do not. In a number of examples we have tried, cross-entropy also seems to yield estimates with less bias than the MAP methods, and although we have not proven this, it appears to be asymptotically consistent whenever an MAP method would be asymptotically consistent. All these features suggest that this Minimum Cross-Entropy Method is the "natural" alternative to Minimum Cost Bayesian Estimation.

Unfortunately, by introducing cross-entropy, we have converted an estimation problem involving a finite number of unknowns into an infinite dimensional functional minimization over the space of [illegible] possible separable probability densities. We choose the simplest possible approach to implementing this minimization, iteratively minimizing the cross-entropy over each component of the separable approximation in turn. This is comparable to using a "coordinate descent" minimization technique, and though gradient directed methods might be faster, none would be as simple. Minimizing the cross-entropy with respect to a component $q(\underline{x})$ of our separable density $q(\underline{x})q(\underline{y}) \cdots$ involves averaging the model log probability density over all unknowns except $\underline{x}$, then using the result as the estimated log probability density $\log q(\underline{x})$ of the unknown $\underline{x}$. In effect, we average the log model probability density over all variables except one, then assume that the remaining variable must account for any remaining variation. We then move to the next variable and do the same. Each iteration strictly reduces the cross-entropy, and thus strictly improves the separable approximation. Furthermore, since the MAP methods can also be stated in terms of fitting a separable model to the given density, exactly the same iterative approach can be used for solving these problems also. The only difference is that in the MAP methods, the components that are restricted to be impulse functions are estimated by maximizing an averaged log density. Each iteration of the MAP algorithms not only strictly decreases the cross-entropy, but also strictly increases the corresponding likelihood function.

We're still left with some multidimensional integrations, the same problem which curses the "optimal" Minimum Cost Bayesian approach. However, if the given model density forms an exponential class of densities, then the infinite dimensional cross-entropy minimization problem reduces to iteratively calculating the expectation of a finite set of functions, each involving only a single unknown. This restriction to the exponential class is actually not that limiting; included in this class are binomial, multinomial, negative binomial, Poisson, Exponential, Gaussian, Gamma, Chi-Square, Beta, and many other densities. In fact, under mild conditions, one can show [9] that every density which can be characterized by a sufficient statistic must be an exponential class. When the model density is transformed to its "natural" exponential form, the algorithms take their simplest form. Cross-entropy simply alternates between calculating the conditional expectation of each unknown in turn given the latest estimate of the other unknowns. The MAP algorithms differ only in that some or all of the expectations are replaced by maximizations of the conditional density. MAP algorithms are usually computationally cheaper than MCEM, but their estimates are usually worse.

Convergence of all the algorithms can be proven under mild conditions. Two basic approaches are used for proving convergence in this thesis. The first relies on the fact that the algorithms strictly decrease a cross-entropy expression on each iteration, and the MAP algorithms also increase a likelihood function on each pass. Analyzing the shape of these functions then leads to an understanding of how these estimates must evolve. The other approach used is that when the cross-entropy or likelihood functions are concave, each iteration often defines a contraction or non-expansion mapping on the space of unknowns. Well known fixed point theorems can then be invoked to prove convergence of the estimates.

The remainder of this thesis is concerned with applications of these ideas to a variety of problems in statistics and signal processing. The first problem we consider is fitting the parameters of a given model density to a set of data when the data has been coarsely quantized, grouped into bins for convenience in collection, or similarly mangled. We propose four different cross-entropy and MAP algorithms for solving this problem. All four algorithms alternate between estimating the exact values of all data measurements, fitting parameters to the model density using these data estimates, then using the improved parameter estimates to further refine the data estimates. We compare the performance of our algorithms with the Minimum Mean Square Error estimates for a couple of examples involving Exponential and Gaussian densities. Cross-entropy appears to give estimates which are virtually identical to those of Minimum Mean Square Error estimation at a fraction of the computational cost, and its estimates are asymptotically consistent. One of the MAP methods is almost as good, but for small amounts of data it gives biased estimates. The two other MAP methods are asymptotically biased (on the other hand, they take very little computation.)

The next class of applications we consider involves optimal reconstruction of Gaussian signals corrupted by additive Gaussian noise, where we are given separate constraints on the signal and output values. Again we apply four different cross-entropy and MAP algorithms to the problem. Each algorithm filters the output estimates, and applies a conditional expectation or projection operator to estimate the signal. The output is then reestimated by applying a conditional expectation or projection operator to the signal. When the constraint sets are convex, each step defines a contraction mapping on the estimates, and geometric convergence to the unique global optimizing solution is guaranteed. If the constraint sets are not convex, convergence is only guaranteed to a critical point of the cross-entropy or likelihood function, provided that the estimates remain bounded.

We also analyze the limiting behavior of our algorithms when our *a priori* signal density becomes asymptotically flat. We show that our algorithms for this case have a similar form, except that the filtering step is omitted, and the resulting iteration is only a non-expansive mapping. Nevertheless, by using a new upper bound on the variance of a log concave probability density, we prove that if the constraint sets are convex, then the "Fisher" algorithms converge to a global optimizing solution if and only if a solution to the problem exists.

The case when the known constraints on the signal and noisy output are defined by linear equalities is particularly interesting and elegant. All of our estimation approaches give identical algorithms in this case. Each iteration uses a linear filtering step, and two linear projection operations onto each of the constraint sets in order to calculate its estimates. An alternative "dual" algorithm is developed which iteratively calculates transformed Lagrange Multipliers, rather than the variables themselves, by using a similar filter/project/project iteration. The dual projection operators, however, are "orthogonal" to those of the original "primal" algorithm, and the dimensions of the problems can be quite different. Numerous closed-form solutions are developed, and we also suggest several different conjugate gradient and PARTAN algorithms to solve the problem in a finite number of steps. Noise sensitivity is analyzed, and shown to be directly related to the convergence rate. Finally, since both the primal and dual problems define a linear mapping on the signal and output spaces, we can analyze the eigenstructure of these mappings.

The simplest application of these reconstruction algorithms is to the problem of reconstructing signals given noisy constraints on its behavior in the time and frequency domains. We first consider the general linear equality time and frequency constraint problem, and two special cases: bandlimited extrapolation, and reconstruction of a finite signal given the phase of its transform modulo π. For all these problems we develop both primal and dual iterative algorithms, conjugate gradient algorithms, closed-form solutions, and analyze their eigenvalues and eigenvectors. Next we consider applications involving more general convex constraint sets. Reconstruction of finite length signals from noisy measurements of the phase modulo 2π is treated in depth, and we compare its performance with the algorithm suggested by Hayes, Lim and Oppenheim [10]. Another application in this category is a new multidimensional Finite Impulse Response filter design algorithm capable of designing FIR filter's meeting arbitrary time and frequency constraints. Finally, we discuss magnitude-only reconstruction, a problem involving non-convex constraints, and present three different algorithms, one of which is identical to that used by Fienup [11] and Hayes [12, 10]. When the constraint sets are non-convex, convergence is only guaranteed to a critical point of the objective function. As a result, our algorithms in this application tend to converge to a local minimum far from the global minimum.

Next we consider more esoteric applications. A new development of Short Time Fourier Transform is presented, in which we generalize the concept of "windows" to arbitrary one-to-one linear operators, prove that the inverse Short Time Fourier Transform is a projection operator, and develop a Parseval-like theorem equating the energy in the time and Short Time Fourier domains. These properties are used to develop general algorithms for reconstruction of signals from constraints on its time and Short Time Fourier domain behavior. In fact, all the time/frequency domain results generalize directly to time/Short Time Fourier domain algorithms. Next we present a possible improvement to Malik and Lim's algorithm (we have not tested this yet, and so there is no guarantee that it works.) We conclude with a new suggestion for penalty functions for constrained minimization problems.

2. Ongoing Research

There are a number of additional applications which were not included in this thesis due to lack of time. These include:

- Iterative Multidimensional Extrapolation/Interpolation/Smoothing of Noisy Finite Segments of Stationary Rational Processes (this iterates between a Weiner-Hopf smoothing filter, optionally linearly predicts the signal tails, then reestimates the unknown output tails from the signal tails.)
- Iterative Pole/Zero Estimation from a Finite Segment of Noisy Observations (these iterate between a finite length smoothing filter, and linear prediction and cross-correlation parameter estimation.)
- Iterative Pole/Zero Estimation and Extrapolation/Interpolation/Smoothing of Noisy Autoregressive Moving Average Models (these combine the above two algorithms in order to implement the filtering in the frequency domain.)
- Recursive Versions of the Noisy Pole/Zero Modeling Algorithms
- A 3 way Separation Theorem for Optimal Control of Linear Quadratic Gaussian Systems in Standard Controllable Form (these replace the expectation of the quadratic cost function by an expectation operator using the separable density approximation. The standard dynamic programming derivation of the separation theorem then gives an algorithm in which we iteratively fit a signal density, fit a parameter density, then refit a control. Recursive/iterative versions of the algorithm are also possible.)

The first algorithm only involves linear equality constraints; it has been programmed and works well. The MAP algorithms for solving the second and third applications have also been programmed; these work best for estimating all-pole models from noisy data, since the zero estimates converge rather slowly. Using iterative extrapolation of the noisy output is also a rather simple but very effective method of eliminating the boundary effects one would normally encounter when using frequency domain filters on finite data segments. The last two applications are areas of ongoing research effort.

3. Historical Background

The idea of developing connections between information theory and probability has been investigated by numerous authors. Kullback's book [13] is perhaps the best example, although it primarily concentrates on applying statistical analysis to information theory rather than vice versa. Many researchers have tried to derive an axiomatic information theoretic basis for statistical inference [14, 15, 16, 17]. The most successful of these, however, was Shore and Johnson [8] who provided a complete axiomatic justification for cross-entropy as the only viable estimation method for incorporating observation data about the form of the model density.

Much has also been written about stochastic estimation involving multiple unknowns, but most analyses have focused on specific applications in which particular features could be exploited to solve the problem. One common suggestion [18, 19] for dealing with pole/zero parameters of a linear state space model, for example, is to add the parameters to the state vector, then iteratively or recursively linearize the equations about the last parameter estimate and use a Kalman Filter to estimate improved parameter and state values. This quasilinearization "extended Kalman Filter" technique, unfortunately, does not necessarily converge. In statistics, extra parameters or signals are often considered "nuisance parameters" to be eliminated if at all possible. No coherent theory seems to have developed for dealing with these extra parameters, though several suggestions recur throughout the literature. We could estimate the nuisance parameters, then set them permanently to their estimated values. We could jointly maximize over the parameters of interest and over the nuisance parameters. Or we could integrate out the nuisance parameters, leaving only a probability density over the desired parameters. It is well known that only the last approach seems to lead to asymptotically consistent parameter estimates. However, I know of no proof of this, or in fact any theoretically solid treatment of the subject.

The work in this thesis was motivated primarily by research on two rather different subjects: pole/zero estimation, and optimal signal reconstruction. Bar-Shalom [20] and Lim [21,22] independently suggested a new approach for solving autoregressive modeling problems with noisy data in which they search for the combination of signal and pole parameters which are jointly most likely. Each iteration simply filters the noisy observations using the latest pole estimates, then fits a new autoregressive model to the clean signal estimate by using linear prediction. This method, which Lim called LMAP, corresponds to our PSMAP approach. Contrary to Bar-Shalom's implication, however, the pole estimates are not asymptotically consistent; in fact, the iteration tends to pull the poles onto the unit circle and drops the model gain to zero, thus producing exceptionally peaky spectra. Using an intuitive argument, Lim suggested a fix for this, called RLMAP, in which he added the signal variance to the correlations of the signal estimate when computing the pole parameters. He noticed that with this correction, the pole spectra appeared to be much closer to the actual signal spectrum. In fact, except for the gain calculations, this idea is exactly what our PARMAP algorithm would calculate, and it exactly solves what we would consider to be the best MAP approach to the problem. Our Master's thesis [23] develops the three MAP algorithms we use in this thesis. By working backwards from Lim's RLMAP algorithm, we discovered a general approach for iteratively computing MAP estimates of pole/zero models from noisy data.

At that time, we did not understand the full applicability of the idea, and did not understand the connection with cross-entropy. As a result, the derivation of the three algorithms in the master's thesis was rather magical; various functions (now recognized as cross-entropies) with exactly the right properties were invented out of thin air, and used to solve the MAP problems. With the cross-entropy development in this dissertation, this former work now takes a more sensible interpretation.

The second source of inspiration for this thesis was the large literature on signal reconstruction from constraints stated in multiple domains. Most of this work, once again, has narrowly focused on specific applications. This has allowed the authors to exploit particular features of the application, but has also tended to obscure the connections between all the problems. The best known signal reconstruction problem given multiple constraints is extrapolating a finite segment of data given that it is part of a bandlimited sequence. Papoulis [24] originally treated this problem for continuous signals, and proposed an algorithm for solving it which iterated between bandlimiting the estimated signal, and then replacing the known segment with its correct value. Convergence was proved by exploiting the properties of Prolate Spheroid Wave Functions. Sabri and Steenaart [25] proposed a single step, closed-form solution to the problem using an "extrapolation matrix". Cadzow [26] reconsidered the problem, and by discretizing the continuous time problem arrived at a much superior closed-form solution. Gerchberg [27] considered same problem with the frequency and time domains reversed; he uses a similar iterative algorithm to estimate the high frequencies of a finite length signal when the low frequencies were given.

A conceptually related problem is that of reconstruction of a signal from samples of the phase or the magnitude of its Fourier Transform, together with some extra information such as finite time domain support, a minimum phase constraint, etc. Fienup [11] considered the problem of reconstructing a finite length signal from the magnitude of its spectrum, a common problem in optics, and proposed two iterative techniques which alternate between clipping the signal to the correct support and forcing it to have the correct spectral magnitude. By varying the algorithm irregularly, he showed that reconstruction was possible in some test cases. Gerchberg and Saxton considered the case when the signal magnitude was known as well is its spectral magnitude This algorithm alternates between forcing the correct magnitude in the time domain, then forcing the correct magnitude in the frequency domain. Hayes, Lim and Oppenheim [10] considered the related problem of reconstruction of a finite length signal from knowledge of its spectral phase modulo 2π, and proposed an iterative algorithm for solving the problem which alternated between forcing the signal to satisfy the known time domain constraints (finite support, known signal point) and forcing it to have the correct spectral phase (but keeping the spectral magnitude constant.) Quatieri and Oppenheim [28] used a similar procedure to iteratively reconstruct minimum phase signals from their phase or magnitude. Finally, Hayes [12, 10] proved a set of simple conditions under which one could uniquely reconstruct a signal with finite support from samples of its spectral phase or magnitude.

The structures of these algorithms are quite similar; we simply alternate between forcing time domain and then frequency domain constraints on the signal. This simple idea of iterating between two domains has encouraged many others to try to apply the same concept to more complicated problems. Malik and Lim [29], for example, solve a multidimensional Maximum Entropy (MEM) spectral estimation problem by iterating between the correlation domain and the convolutional inverse of the correlation domain, forcing constraints on the model power spectrum in both donains in an attempt to find the MEM power spectrum. Finite Impulse Response filter design algorithms, such as Remez exchange and others [30], have been deliberately designed to try to iteratively adjust the filter coefficients in the time domain in order to decrease the worst errors in the frequency domain. Another example is that in considering a statistical probiem involving grouped data, Hartley [31] discovered one of our MAP algorithms for the special case of fitting a discrete multinomial distribution to a given grouped data distribution. This paper, which we only found after finishing chapter 4 on grouped data problems, had the misfortune to be written in 1958 before the advent of modern digital computers. Since the iteration did not converge in 4 to 5 passes, the idea was apparently discarded. Even more extreme examples are the iterative ARMA modeling algorithms suggested in chapter 7 of [18], or the Iterative Inverse Filtering algorithms of Konvalinka and Mataušek [32] which iterate between estimating residuals, poles and zeroes in a manner that appears to solve the corresponding modeling problems.

Recognizing the conceptual similarity of all these algorithms. as well as their resemblance to certain iterative deconvolution algorithms, numerous authors have tried to unify the presentation and convergence proofs of these algorithms. The most successful attempts revolve around the notion of non-expansive and contraction mappings. Tom, Quatieri, Hayes and McClellan [33], for example, showed that when the solution to the reconstruction problem is unique, then convergence of the bandlimited and the phase-only reconstruction algorithms cculd be proved by showing that each iteration of the algorithms defined a strictly non-expansive mapping. Fixed point theorems of Ortega and Rheinboldt [34] were then invoked to prove convergence. Schafer, Mersereau and Richards [35] took an identical approach in proving convergence of deconvolution and bandlimited extrapolation algorithms. Landau [36], Sandberg [37,38], and Zames [39] proved similar results for systems incorporatir,g nonlinearities. Wiley [40,41] used these nonlinear extensions to analyze iterative wideband FM demodulation algorithms.

Youla [42] considered the reconstruction problem from a different perspective, recognizing that the Papoulis bandlimited extrapolation problem was only one example of a class of iterative projection algorithms involving two sets of constraints on projections of the unknown signal. By considering the more general reconstruction problem in an abstract Hilbert space setting, he was able to characterize the properties of the algorithm in terms of the "angle" between the constraint spaces. The approach we use in the special case of linear equality constraints will be somewhat similar to that of Youla, although we will tighten some of his noise bounds, provide a convergence rate analysis, characterize the eigenvalues and eigenvectors of the problem, and show that additional properties can be proven for finite dimensional spaces. We will only treat finite dimensional problems in detail; many of Youla's conclusions for infinite dimensional spaces will follow, however, from limiting arguments. Perhaps the most important difference between our approach and that of Youla, is that we show that many of the properties of the class of iterative projection algorithms remain true even when the constraints are not linear, but only convex, and even if we use expectation operators of truncated Gaussians instead of projection operators. Mosca [43] also treated the same subject in depth, analyzing the various degeneracies possible in solving ill-behaved linear problems in infinite dimensional spaces.

The paper which comes closest to our approach is that of Jain and Ranganath [44], published six months after this PhD proposal was submitted. They interpreted the bandlimited extrapolation problem as solving a least squares problem. They derive

Papoulis' iterative algorithm, they discuss closed-form solutions in terms of Discrete Prolate Spheroid functions, and they show that Cadzow's closed-form solution is the minimum norm solution to the least squares problem. The least squares approach leads to a conjugate gradient iterative algorithm. It also suggests simple techniques for simultaneously filtering out noise or certain types of clutter.' Our basic approach is conceptually similar to theirs in that we both start with (slightly different) optimality criteria for judging the "goodness" of a signal estimate. We both use this criterion to derive estimation algorithms which can be made robust to noise. The major difference is that we show that the properties of the algorithm which they derive are not particular to the bandlimited extrapolation problem, but hold for an extremely wide class of signal reconstruction problems with constraint sets defined by linear equalities. All these problems have eigenvalues and eigenvectors with properties identical to the Discrete Prolate Spheroid functions, all can be made noise insensitive, all have several different types of closed form solutions, each of which can be efficiently solveo by conjugate gradient or PARTAN algorithms in a finite number of steps. All can be solved by either primal or dual algorithms. (Our dual iterative algorithm appears to be completely new.) Finally, when the noise characteristics are known, and when the constraint sets are convex, though no linear varieties, then cross-entropy and certain MAP approaches provide better optimality criteria than simple least squares. In turn, our major debt to Jain and Ranganath is that their paper encouraged us to examine the use of conjugate gradient methods for the general signal reconstruction problem.

Finally, we remark that fixed point theorems are a fundamental tool of analysis, and the advantage of using this approach is that convergence can be proven even if the algorithm involves non-linearities or convex constraints [37,38,33]. On the other hand, the non-expansive mapping approach is only useful for proving convergence of preexisting algorithms and is not that helpful at suggesting algorithms for solving new problems. A much more rewarding approach is to define an objective function measuring the "goodness' of our estimates, and then to optimize this function iteratively. When the objective function is quadratic, or sometimes even when it is only concave, the resulting iterations are often contraction or non-expansion mappings, and we will have thus generated an algorithm whose convergence can be easily verified.

4. Outline of Thesis

The remainder of this chapter contains a brief summary of some concepts of real analysis that will be used in the convergence proofs, and a list of symbols. Section A of chapter 2 discusses the classical Minimum Mean Square Error, MAP and ML methods of stochastic estimation, then introduces Cross-entropy and lists numerous properties of this information measure. Section B considers stochastic estimation problems involving two different unknowns, which we arbitrarily take to be a signal and a parameter. The Minimum Mean Square Error (MMSE) estimate is briefly described, then three different MAP methods are introduced. One (PARMAP) finds the most likely parameter value; the second (SIGMAP) finds the most likely signal value; the third (PSMAP) finds the combination of signal and parameter values which are simultaneously most likely. The Minimum Cross-Entropy Method (MCEM) is introduced, and we show that all three MAP methods can be viewed as degenerate forms of MCEM. Section C discusses existence and uniqueness theorems for optimization of functions over finite or infinite dimensional spaces.

Chapter 3 develops iterative algorithms for solving our estimation algorithm. The simple idea of minimizing with respect to the signal component and then the parameter component of the separable density, is used to solve MCEM and the three MAP methods. For exponential families of densities, all four algorithms are shown to take a particularly elegant form. Section B of chapter 3 painstakingly develops the mild conditions under which convergence of these four algorithms is guaranteed.

Chapter 4 applies the four methods to statistical modeling problems involving grouped or quantized data. These algorithms all iterate between estimating the actual data values, then estimating the model parameters using these data estimates. Chapter 5 considers optimal signal reconstruction for Gaussian signals corrupted by Gaussian noise, when the available observation information defines constraints on the possible signal and output values. In all four algorithms the signal is estimated by applying a projection or conditional expectation operator to the filtered output estimate. The output is then reestimated by applying another projection or conditional expectation operator to the filtered signal estimate. Lavish attention is given to the case when the constraints are defined by linear equalities, and we develop primal and dual iterative algorithms, conjugate gradient algorithms, closed-form solutions, noise sensitivity analysis and analyze the eigenstructure. Chapter 6 continues analyzing the optimal signal reconstruction problem by treating the behavior of the algorithms when the *a priori* signal density becomes asymptotically flat. The limiting form and convergence behavior of all our reconstruction algorithms is then carefully reexamined for the case when the density is exactly flat.

Chapter 7 applies all this reconstruction theory to problems involving time and frequency constraints. Special cases considered include bandlimited extrapolation, phase-only and magnitude-only reconstruction, and multidimensional FIR filter design. Chapter 8 concludes by extending the algorithms to reconstructing signals given time and Short Time Fourier domain constraints. This chapter also suggests a new MEM spectral estimation algorithm, and a new penalty function for constrained optimization problems.

5. Elementary Concepts of Real Analysis

Several ideas in functional analysis will be used quite heavily throughout this thesis. The following is intended as a quick summary of some of the most fundamental of these concepts. Other definitions and theorems will be introduced as needed. There are many good references for this material; see, for example, Luenberger, [45] Goldstein, [46] or Demyanov and Rubinov [47] . (The casual reader should skip this section and continue with chapter 2.)

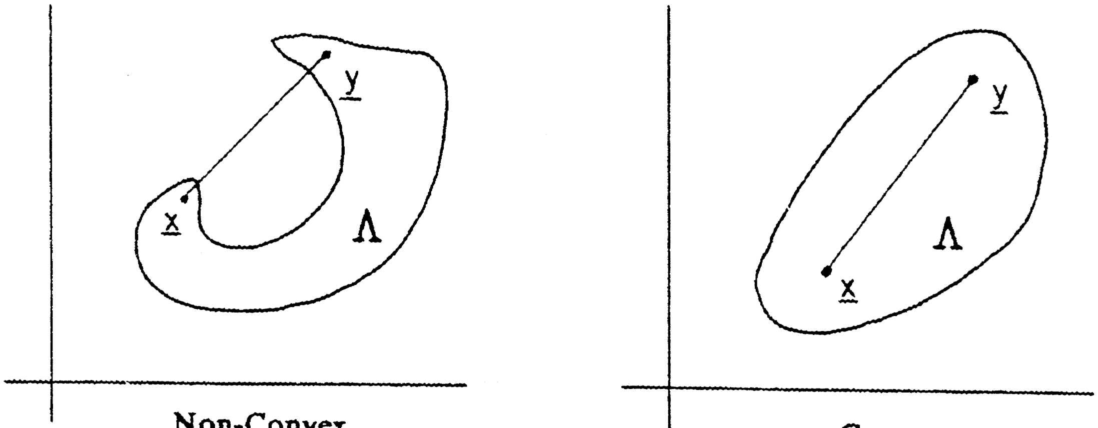
In general, we will restrict our attention to finite dimensional normed linear vector spaces such as the N dimensional real or complex Euclidian spaces $\mathbf{R}^N$ or $\mathbf{C}^N$. Sets Λ will be called bounded if there exists an upper limit M to the norm of every vector in Λ, $||x|| \leq M$ for all $x \in \Lambda$. A sequence of points $\{x_k\}$ is called a Cauchy sequence if for any $\epsilon > 0$, there exists an N such that:

$$||x_n - x_m|| \leq \epsilon \qquad \text{for all } n,m \geq N \tag{1.5.1}$$

The spaces $\mathbf{R}^N$ and $\mathbf{C}^N$ are complete, which means that every Cauchy sequence in the space converges to a point in the space. The set Λ is called "closed" if every Cauchy sequence $\{x_n\}$ in Λ converges to an element of Λ. The complement $\overline{\Lambda}$ of the set Λ, containing all elements not in Λ, is open if Λ is closed. If x_0 is an element of an open set, then there exists a ball of radius $\epsilon > 0$ around x_0 such that every element in the ball also belongs to Λ (thus if $||x - x_0|| < \epsilon$ then $x \in \Lambda$.) Intuitively, closed sets include their boundary, and open sets do not. The closure of a set is the union of the set with all limit points of all infinite converging sequences in the set. A set Λ is called "compact" if every infinite sequence of elements in the set has at least one infinite subsequence which converges to an element of Λ. The Bolzano-Weierstrass theorem guarantees that every closed and bounded set in $\mathbf{R}^N$ or $\mathbf{C}^N$ is compact. The "cluster points" ("limit points") of an infinite sequence $\{x_n\}$ are all points $\underline{x}_*$ such that there is an infinite subsequence $\{x'_n\} \subseteq \{x_n\}$ which converges to $\underline{x}_*$. Equivalently, every neighborhood of a cluster point contains an infinite number of elements of $\{x_n\}$.

A set Λ is called convex if for every two points $\underline{x}, \underline{y} \in \Lambda$, every point on the line connecting $\underline{x}$ and $\underline{y}$ is also in Λ:

$$\lambda \underline{x} + (1-\lambda)\underline{y} \in \Lambda \quad \text{for } 0<\lambda<1 \tag{1.5.2}$$

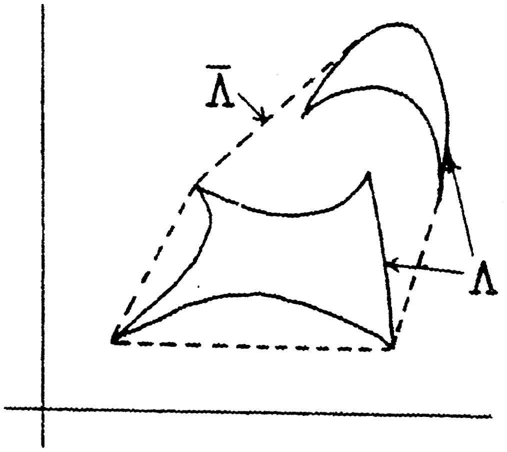


Non-Convex

Convex

Convex and Non-convex Sets

The closed convex hull of a set, sometimes called the "cover", is the smallest closed convex set containing Λ.



Convex Hull of Λ

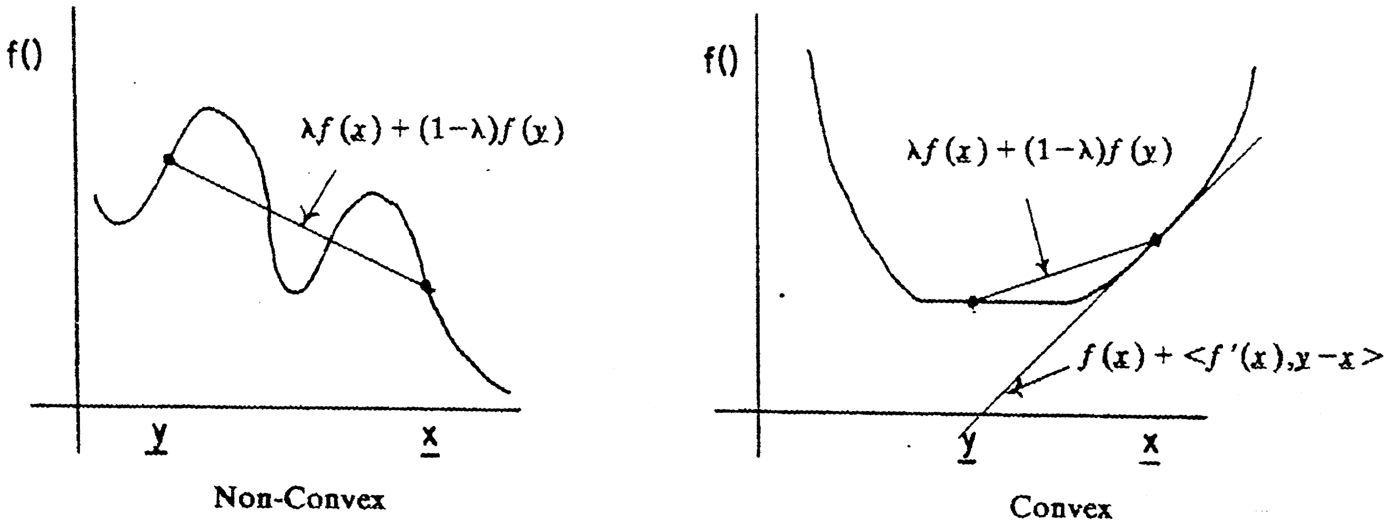
A function $f:\Lambda \rightarrow R$ mapping a convex set Λ into the reals is itself called convex if for all $x,y \in \Lambda$,

$$f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda)f(y) \quad \text{for all } 0<\lambda<1 \tag{1.5.3}$$

In other words, the line connecting $(x,f(x))$ and $(y,f(y))$ always lies above the function f. This function is called strictly convex if equality holds in the definition above if and only if $x \neq y$. "Proper" convex functions also satisfy $f(x)>-\infty$ for all x. (We will assume throughout that all functions are proper.) Convex functions are continuous in the interior of their domain. If a convex function is also differentiable, then the following relationships hold:

$$\begin{aligned} &< f'(x), y-x > \;\le\; f(y) - f(x) \\ &< f'(y)-f'(x), y-x > \;\ge\; 0 \end{aligned} \tag{1.5.4}$$

If $f(x)$ is strictly convex, then strict inequality holds above if $x \neq y$. Intuitively, these relationships imply that the tangent to $f(x)$ lies below the function. The function $f(x)$ is called "concave" if $-f(x)$ is convex.


f()
$\lambda f(\underline{x}) + (1-\lambda) f(\underline{y})$
$\underline{y}$
$\underline{x}$
f()
$\lambda f(\underline{x}) + (1-\lambda) f(\underline{y})$
$f(\underline{x}) + \langle f'(\underline{x}), \underline{y} - \underline{x} \rangle$
$\underline{y}$
$\underline{x}$

Non-Convex

Convex

Convex and Non-convex Functions

We will also need to treat infinite dimensional vector spaces in this thesis. Unfortunately, analyzing convergence in infinite dimensional spaces is considerably more difficult than in finite dimensions. For example, closed and bounded infinite dimensional sets are not compact, and it is easy to find infinite bounded sequences with no limit points whatsoever. This subject is ordinarily treated by generalizing our concepts of convergence to a "weak" topology. We mention this only to stress that extending the results of this thesis to infinite dimensional spaces is generally non-trivial, and we will therefore concentrate primarily on finite dimensional problems.

6. The Cast of Characters

First let us introduce some notation.

A, B - capital Roman letters are matrices

Ψ, Φ - capital Greek letters are sets

$\underline{x}$, $\underline{y}$ - underlined Greek or Roman letters are vectors

α, β - lower case Greek or Roman letters are scalars

$F(\underline{x})$, $f(\alpha)$ - functions

Indexing:

A_{ij} or $[A]_{ij}$ - the $(i,j)^{th}$ element of matrix A (the first row or column may be numbered from 0 or 1 depending on circumstances.)

$\underline{x}_i$ - the i^{th} element of the vector $\underline{x}$

A_k - the k^{th} in a sequence of matrices A_1, A_2, $\cdots$

$\underline{x}_k$ - either the k^{th} vector in a sequence, or a vector of length k, depending on use.

$[A_k]_{ij}$ - the $(i,j)^{th}$ element of the k^{th} matrix A. Analogously for vectors.

Transpose, Inverses, Conjugates

A^T, $\underline{a}^T$ - transpose of A or $\underline{a}$, i.e. $A^T_{ij} = A_{ji}$

A^*, $\underline{a}^*$ - complex conjugate of A or $\underline{a}$

A^H, $\underline{a}^H$ - complex conjugate transpose (Hermitian) of A, i.e. $A^H = A^{T^*}$

A^{-1} - inverse of A

A^{-T}, A^{-H} - inverse of A^T or A^H respectively

Special Functions:

δ_{ij} - Kronecker delta function, $\delta_{ij} = 1$ if $i = j$, and $= 0$ else

$\delta(t - t_0)$ - impulse function, equals zero everywhere except t_0, but integrates to one over all neighborhoods of $\underline{t}_0$.

$|\mathrm{A}|$ - determinant of A

$tr(\mathrm{A}) = \sum \mathrm{A}_{ii}$ - the trace of A

Special Vectors, Matrices

I - identity matrix, $\mathrm{I}_{ij} = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$

I_m - the $m \times m$ identity matrix

$\underline{0}$ - the zero vector, $\underline{0} = (\,0 \;\cdots\; 0\,)^{\mathrm{T}}$

$\underline{0}_m$ - a zero vector of length m

$\mathrm{A} = diag(\underline{a})$ - a diagonal matrix with elements $\mathrm{A}_{ij} = a_i \delta_{ij}$

Special Sets:

$N(\mathrm{A})$ - the null-space of the matrix A

$R(\mathrm{A})$ - the range space of the matrix A

Derivatives of a Scalar Function:

$\dfrac{\partial f(\underline{a})}{\partial \underline{a}}$ - is the column vector, $\left[\dfrac{\partial f}{\partial \underline{a}}\right]_i = \dfrac{\partial f}{\partial a_i}$

$\dfrac{\partial f(\mathrm{A})}{\partial \mathrm{A}}$ - is the matrix, $\left[\dfrac{\partial f}{\partial \mathrm{A}}\right]_{ij} = \dfrac{\partial f}{\partial \mathrm{A}_{ij}}$

$\dfrac{\partial^2 f}{\partial \underline{a}\, \partial \underline{b}}$ - is the matrix, $\left[\dfrac{\partial^2 f}{\partial \underline{a}\, \partial \underline{b}}\right]_{ij} = \dfrac{\partial^2 f}{\partial a_i\, \partial b_j}$

Derivatives of a Vector Function:

$\dfrac{\partial \underline{f}(\alpha)}{\partial \alpha}$ - is the column vector $\left[\dfrac{\partial \underline{f}(\alpha)}{\partial \alpha}\right]_i = \dfrac{\partial f_i(\alpha)}{\partial \alpha}$

$\dfrac{\partial \underline{f}(\underline{a})}{\partial \underline{a}}$ - is the matrix $\left[\dfrac{\partial \underline{f}}{\partial \underline{a}}\right]_{ij} = \dfrac{\partial f_i}{\partial a_j}$

Derivative of a Matrix Function

$\frac{\partial A(\alpha)}{\partial \alpha}$ - is the matrix $\left[\frac{\partial A}{\partial \alpha}\right]_{ij} = \frac{\partial A_{ij}}{\partial \alpha}$

Higher order derivatives will not be needed.

Probability - Let A be an event, and x a random variable

$P(A)$ - the probability of an event A

$p(x)$ - the probability density function of x

$p(x|A)$ - the conditional probability density of x given that A occurred

$E[x] = \int x \; p(x) \; dx$ - the expected value of x

$E[x|A] = \int x \; p(x|A) \; dx$ - the expected value of x given that A occurred

$\mathrm{Cov}[\underline{x}] = E\left[(\underline{x} - E[\underline{x}]) \, (\underline{x} - E[\underline{x}])^H \right] = E[\underline{x}\underline{x}^H] - E[\underline{x}]E[\underline{x}^H]$

- the covariance of $\underline{x}$

$N(\underline{x}, V)$ - the normal distribution with mean $\underline{x}$ and covariance matrix V

Inner Products, Orthogonality:

$<\underline{u}, \underline{v}>_A = \underline{u}^H A^{-1} \underline{v}$ - an inner product, where A is a positive definite Hermitian linear operator, $A^H = A$

$||\underline{v}||_A = \sqrt{<\underline{v}, \underline{v}>_A}$ - the vector norm associated with this inner product

$||B||_A = \max\limits_{\underline{v} \neq 0} \frac{||B\underline{v}||_A}{||\underline{v}||_A}$ - the matrix norm associated with this inner product.

Clearly $||B\underline{v}||_A \leq ||B||_A \, ||\underline{v}||_A$ for all $\underline{v}$.

$||\underline{v}||_2 = (\underline{v}^H \underline{v})^{½}$ - the Euclidian norm

$\underline{x} \perp \underline{y}$ - means $\underline{x}$ is orthogonal to $\underline{y}$, $<\underline{x}, \underline{y}> = 0$

$\underline{x} \perp \Psi$ - means $\underline{x}$ is orthogonal to every element of the set Ψ, $<\underline{x}, \underline{\psi}> = 0$ for all $\underline{\psi} \in \Psi$

$\Phi \perp \Psi$ - means every element of the set Φ is orthogonal to every element of set

Ψ, $<\underline{\phi},\underline{\psi}> = 0$ for all $\underline{\phi}\in\Phi$, $\underline{\psi}\in\Psi$

$\Psi^{\perp}$ - is the orthogonal complement of the set Ψ, i.e. the set of all elements $\underline{x}$ such that $<\underline{x},\underline{\psi}> = 0$ for all $\underline{\psi}\in\Psi$

$N(A)^{\perp}$, $R(A)^{\perp}$ - the orthogonal complements of the null and range spaces of A.

Other Notation:

$\underline{v}\leq\underline{w}$ - every component of the vector $\underline{v}$ is less than or equal to the corresponding component of $\underline{w}$, $v_i\leq w_i$

$A>0$ - the matrix A is positive definite, $\underline{x}^H A\underline{x}>0$ for all $\underline{x}\neq 0$

$A\geq 0$ - the matrix A is semipositive definite, $\underline{x}^H A\underline{x}\geq 0$ for all $\underline{x}$

$A\geq B$ - means that $\underline{x}^H A\underline{x} \geq \underline{x}^H B\underline{x}$ for all $\underline{x}$

(a,b) - the open interval between a and b

$[a,b]$ - the closed interval including a and b

$\Omega\times\Phi = \left\{(\underline{x},\underline{y}) \,\middle|\, \underline{x}\in\Omega, \underline{y}\in\Phi\right\}$ - the Cartesian product of sets Ω and Φ

Other notation will be introduced as needed.

References

1. H. L. Van Trees, *Detection, Estimation and Modulation Theory,* Wiley, New York (1968).
2. C.R. Rao, *Linear Statistical Inference and Its Applications,* John Wiley & Sons, New York (1965).
3. Harald Cramér, *Mathematical Methods of Statistics,* Princeton Univ. Press, Princeton, N.J. (1946).
4. Fred C. Schweppe, *Uncertain Dynamic Systems,* Prentice-Hall Inc., Englewood Cliffs, N.J. (1973).
5. Yaakov Bar-Shalom, "On the Asymptotic Properties of the Maximum Likelihood Estimate Obtained from Dependent Observations," *Jour. Royal Stat. Soc., Ser. B* 33(1), pp.72-77 (1971).
6. Edison Tse and John J. Anton, "On the Identifiability of Parameters," *IEEE Trans. on Auto. Control* AC-17(5), pp.637-646 (Oct. 1972).
7. Martin J. Crowder, "Maximum Likelihood Estimation for Dependent Observations," *Jour. of the Royal Statist. Soc., Ser. B, Vol. 38*(1), pp.45-53 (1976).
8. John E. Shore and Rodney W. Johnson, "Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy," *IEEE Trans. Info. Theory* IT-26(1), pp.26-37 (Jan 1980).
9. Koopman, "On Distributions Admitting a Sufficient Statistic," *Trans. Am. Math. Soc.* 39, pp.399-409 (1936).
10. Monty H. Hayes, Jae S. Lim, and Alan V. Oppenheim, "Signal Reconstruction from Phase or Magnitude," *IEEE Trans. Acoust., Speech, and Signal Processing* ASSP-28(6), pp.672-680 (Dec 1980).
11. J.R. Fienup, "Reconstruction of an Object from the Modulus of its Fourier Transform," *Optics Letters* 3(1), pp.27-29 (July 1978).
12. Monson H. Hayes III., *Signal Reconstruction from Phase or Magnitude,* M.I.T. PhD Thesis (June 1981).
13. Solomon Kullback, *Information Theory and Statistics,* John Wiley & Sons, New York (1959).
14. R.L. Kashyap, "Prior Probability and Uncertainty," *IEEE Trans. Info. Theory* IT-17(6), pp.641-650 (Nov 1971).
15. Pl. Kannappan, "On Shannon's Entropy, Directed Divergence and Inaccuracy," *Z. Wahrsch. verw. Geb.* 22, pp.95-100 (1972).
16. Pl. Kannappan, "On Directed Divergence and Inaccuracy," *Z. Wahrsch. verw. Geb.* 25, pp.49-55 (1972).
17. Arthur Hobson, "A New Theorem of Information Theory," *Jour. Statis. Phys.* 1(3), pp.383-387 (1969).
18. Pieter Eykhoff, *System Identification - Parameter and State Estimation,* John Wiley & Sons, New York (1974).
19. D.G. Lainiotis and S.K Park, "On Joint Detection, Estimation and System Identification: Discrete Data Case," *Int. J. Control* 17(3), pp.609-633 (1973).

20. Yaakov Bar-Shalom, "Optimal Simultaneous State Estimation and Parameter Identification in Linear Discrete-Time Systems," *IEEE Trans. on Auto. Control* AC-17(3), pp.308-319 (June 1972).

21. Jae S. Lim and A. V. Oppenheim, "All-Pole Modeling of Degraded Speech," *IEEE Trans. Acoust. Speech, Signal Proc.* ASSP-26(3), pp.197-210 (June 1978).

22. Jae S. Lim, *Enhancement and Bandwidth Compression of Noisy Speech by Estimation of Speech and its Model Parameters*, Sc.D. Thesis, Dept. of Elec. Eng. and Comp. Sci. M.I.T., Cambridge, Mass. (Aug 1978).

23. Bruce R. Musicus, *An Iterative Technique for Maximum Likelihood Parameter Estimation on Noisy Data*, S.M. Thesis, M.I.T., Cambridge, Mass. (Feb 1979).

24. Athanasios Papoulis, "A New Algorithm in Spectral Analysis and Bandlimited Signal Extrapolation," *IEEE Trans. Circuits Syst.* CAS-22(9), pp.735-742 (Sept 1975).

25. M. Shaker Sabri and Willem Steenaart, "An Approach to Band-Limited Signal Extrapolation: The Extrapolation Matrix," *IEEE Trans. Circ. Sys.* CAS-25(2), pp.74-78 (Feb 1978).

26. J. A. Cadzow, "An Extrapolation Procedure for Band-Limited Signals," *IEEE Trans Acoust., Speech, and Signal Processing* ASSP-27(1), pp.4-12 (Feb 1979).

27. R. W. Gerchberg, "Super-resolution through Error Energy Reduction," *Optica Acta* 21, pp.709-720 (1974).

28. Tom F. Quatieri and Alan V. Oppenheim, "Iterative Techniques for Minimum Phase Signal Reconstruction from Phase or Magnitude," *IEEE Trans. Acoust., Speech, and Sig. Proc.* ASSP-29(6), pp.1187-1193 (Dec 1981).

29. Naveed Malik, *One and Two Dimensional Maximum Entropy Spectral Estimation*, M.I.T. ScD Thesis (Nov 1981).

30. Lawrence R. Rabiner and Bernard Gold, *Theory and Applications of Digital Signal Processing*, Prentice Hall Inc., Englewood Cliffs, N.J. (1975).

31. H.O. Hartley, "Maximum Likelihood Estimation from Incomplete Data," *Biometrics* 14, pp.174-194 (June 1958).

32. Ira S. Konvalinka and Miroslav R. Matausek, "Simultaneous Estimation of Poles and Zeros in Speech Analysis and ITIF-Iterative Inverse Filtering Algorithm," *IEEE Trans. Acoust., Speech, and Signal Processing* ASSP-27(5), pp.485-492 (Oct 1979).

33. Victor T. Tom, Thomas F. Quatieri, Monson H. Hayes, and James H. McClellan, "Convergence of Iterative Nonexpansive Signal Reconstruction Algorithms," *IEEE Trans. Acoust., Speech, Sig. Proc.* ASSP-29(5), pp.1052-1058 (Oct 1981).

34. J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York (1970).

35. Ronald W. Schafer, Russell M. Mersereau, and Mark A. Richards, "Constrained Iterative Restoration Algorithms," *Proc. IEEE* 69(4), pp.432-450 (April 1981).

36. H.J. Landau, "On the Recovery of a Band-Limited Signal, After Instantaneous Companding and Subsequent Band-Limiting," *Bell Syst. Tech. J.* 39, pp.351-364 (March 1960).

37. I.W. Sandberg, "On the Properties of Some Systems That Distort Signals I," *Bell Syst. Tech. J.* 42, pp.2033-2046 (Sept 1963).

38. I.W. Sandberg, "On the Properties of Some Systems That Distort Signals II," *Bell Syst. Tech. J.* 43, pp.91-112 (Jan 1964).

39. G.D. Zames, *Conservation of Bandwidth in Nonlinear Operations*, Quarterly Progress Report, MIT Research Laboratory of Engineering (October 1959).

40. Richard G. Wiley, "On an Iterative Technique for Recovery of Bandlimited Signals," *Proc. IEEE* 66(4), pp.522-523 (April 1978).

41. Richard G. Wiley, "Demodulation Procedure for Very Wide-Band FM," *IEEE Trans. Comm.* COM-25(3), pp.318-327 (March 1977).

42. Dante Youla, "Generalized Image Restoration by the Method of Alternating Orthogonal Projections," *IEEE Trans. Circuits. Syst.* CAS-25(9), pp.694-702 (Sept 1978).

43. Edoardo Mosca, "On a Class of Ill-Posed Estimation Problems and a Related Gradient Iteration," *IEEE Trans. Auto. Control* AC-17(4), pp.459-465 (Aug 1972).

44. Anil K. Jain and Surendra Ranganath, "Extrapolation Algorithms for Discrete Signals with Application in Spectral Estimation," *IEEE Trans. Acoust., Speech, Sig. Proc.* ASSP-29(4), pp.830-845 (Aug 1981).

45. David G. Luenberger, *Optimization By Vector Space Methods*, John Wiley & Sons Inc., New York (1969).

46. A.A. Goldstein, *Constructive Real Analysis*, Harper and Row, New York (1967).

47. Vladimir F. Demyanov and Aleksandr M. Rubinov, *Approximate Methods in Optimization Problems*, Amer. Elsevier Publ. Co., New York (1970).

Chapter 2
Estimation Approaches

SECTION A - ESTIMATION METHODS

1. Introduction

In this chapter we will discuss the problem of parameter and signal estimation given noisy and incomplete data. We consider two fundamentally different types of estimation methods. Point estimation methods based on Minimum Mean Square Error (M'MSE), Maximum Likelihood (ML), and Maximum A Posteriori (MAP) are developed first. These methods use the given data to generate an estimate of the unknowns, possibly together with a confidence interval for their value. We will also consider a quite different approach, based on Minimum Cross Entropy (MCEM), which uses the available observation information to estimate the entire probability density of the unknowns. When only one unknown needs to be estimated (the "classical" estimation problem), all these methods are straightforward. When several signals and/or parameter variables must be estimated from noisy and incomplete observations, however, there are many ways in which each of these criteria could be applied. We therefore propose and compare several different MMSE, ML, and MAP approaches. We also propose a new MCEM method which uses cross-entropy to optimally fit a separable probability density to the given model density. Surprisingly, all of our point estimation MAP methods can be derived as degenerate forms of the cross-entropy method, in which we force one or more components of the separable density approximation to be an impulse function. Cross-entropy thus provides a framework which will unify our treatment of all the MCEM, ML and MAP approaches we present. Iterative algorithms for solving these problems are presented in chapter 3, and the remainder of the thesis is concerned with applying the techniques to various problems.

2. Minimum Mean Square Error Estimation

The problems of parameter identification and signal estimation arise in many system modeling applications. Given a stochastic model relating the output of a system $\underline{y}$ to the unknown signal $\underline{x}$ as a function of the unknown parameters $\underline{\phi}$, our goal will be to optimally estimate $\underline{\phi}$, $\underline{x}$, and $\underline{y}$ from noisy and incomplete observations of the system. Unfortunately, the definition of what is optimal usually depends strongly on the specific requirements of the application. In the fields of speech enhancement or compression, for example, the ultimate criterion is whether the algorithm produces intelligible, natural and pleasant sounding speech, and whether it can be implemented in real-time with inexpensive computer hardware. Unfortunately, criteria such as this may be essential for engineering a "good" system, but they are difficult to quantify or to implement in a general-purpose estimation algorithm.

An alternate approach, the one which we will take in this thesis, is to use parameter and signal estimation techniques which have very well behaved characteristics and are applicable to a wide range of problems. Suppose we are given the conditional probability density $p_{Z|\Lambda}(\underline{z}|\underline{\alpha})$ of the observation information $\underline{z}$ given the unknown to be estimated $\underline{\alpha}$. Also suppose that the unknown can be treated as a Bayesian random variable with a priori probability density $p_{\Lambda}(\underline{\alpha})$. The observations are assumed to be constrained to a subset of their domain $\underline{z} \in Z$ and the unknown is constrained to the set $\underline{\alpha} \in \Lambda$ (we assume that $\underline{z}$ and $\underline{\alpha}$ are finite dimensional vectors). Given all this information, one of the "best" approaches for estimating $\underline{\alpha}$ would be to use an estimator

$\hat{\underline{\alpha}}=\hat{\underline{\alpha}}(\underline{z})$ which on average will have the least possible mean square error in locating the true value of $\underline{\alpha}$:

$$\hat{\underline{\alpha}} \leftarrow \min_{\hat{\underline{\alpha}}\in\Lambda} \mathrm{E}_{\Lambda|Z}\left[\ \|\ \underline{\alpha} - \hat{\underline{\alpha}}(\underline{z})\ \|^2 \ \middle|\ \underline{z}\ \right] \qquad (2.2.1)$$

where the arrow implies that $\hat{\underline{\alpha}}$ is the argument at which the minimum occurs. The notation $\mathrm{E}_{\Lambda|Z}[\cdot|\underline{z}]$ implies that the conditional expectation is calculated over the set of feasible parameter values $\underline{\alpha}\in\Lambda$; for example:

$$\mathrm{E}_{\Lambda|Z}\left[\ f(\underline{\alpha})\ \middle|\ \underline{z}\ \right] = \int_{\Lambda} f(\underline{\alpha})\, \mathrm{p}_{\Lambda|Z}(\underline{\alpha}\,|\,\underline{z})\, d\underline{\alpha} \qquad (2.2.2)$$

Evaluating the expression in (2.2.2) gives:

$$\mathrm{E}_{\Lambda|Z}\left[\ \|\underline{\alpha}-\hat{\underline{\alpha}}(\underline{z})\|^2\ \middle|\ \underline{z}\ \right] = \mathrm{Var}(\ \underline{\alpha}\ |\ \underline{z}\ ) + \|\ \bar{\underline{\alpha}}-\hat{\underline{\alpha}}(\underline{z})\ \|^2 \qquad (2.2.3)$$

$$\text{where:}\ \ \bar{\underline{\alpha}} = \mathrm{E}_{\Lambda|Z}[\ \underline{\alpha}\ |\ \underline{z}\ ]$$

Clearly if the conditional expectation of $\underline{\alpha}$ given $\underline{z}$ is an element of Λ, then minimizing (2.2.3) would give:

$$\boxed{\textbf{MMSE:}\quad \hat{\underline{\alpha}}(\underline{z}) = \mathrm{E}_{\Lambda|Z}[\ \underline{\alpha}\ |\ \underline{z}\ ] \qquad \textbf{(2.2.4)}}$$

Thus the Minimum Mean Square Error (MMSE) estimator of $\underline{\alpha}$ is simply the conditional expectation of $\underline{\alpha}$ over Λ given $\underline{z}$. Assuming that the expected mean square error in (2.2.1) is the best measure of the cost of estimator error, then MMSE must be the best possible Bayesian estimation procedure.

MMSE unfortunately has several drawbacks. If the set Λ is not convex, then there is no guarantee that the expectation of $\underline{\alpha}$ will lie in the set Λ. Appendix A proves that:

Theorem 2.2.1: The element $\bar{\underline{\alpha}}=\mathrm{E}_{\Lambda|Z}[\underline{\alpha}|\underline{z}]$ (if it exists) is an element of the closed convex hull of Λ.

(The convex hull of a set is the smallest convex set which contains the set. See chapter 1, section 5 for more details on convex sets.) Theorem 2.2.1 is illustrated in figure 2.2.1. If the set Λ is closed and convex, then it is its own convex hull, and $\bar{\alpha}\in\Lambda$. If Λ is not convex, however, and $E_{\Lambda|z}[\alpha|z]\notin\Lambda$, then we will either have to accept this non-feasible estimate of α, or else try to find the point in Λ' closest to the expected value.

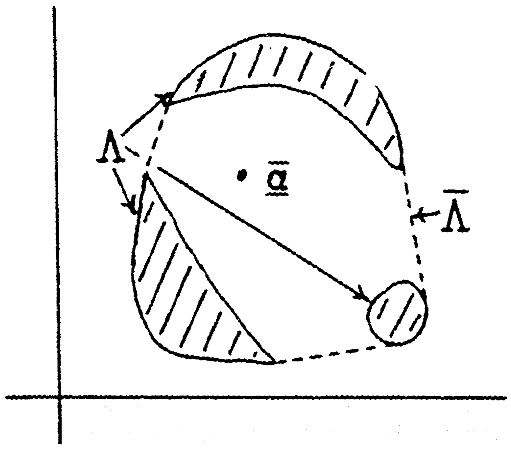


Figure 2.2.1 - MMSE Estimation

Another problem is that MMSE is quite sensitive to the tails of the distribution; in fact, for many legitimate probability densities the expected mean square error in (2.2.1) will be infinite. A related problem is that evaluating the conditional expectation is computationally intensive, as it requires a complicated multidimensional integral over a domain which is often infinite in extent. Finally, MMSE can not be applied to problems in which the unknown to be estimated, α, is most appropriately treated as a Fisher non-random constant.

One possible solution to this last problem is to try to choose an *a priori* density $p(\alpha)$ which contains the least possible information about the unknown, and then apply the Bayesian algorithm using this "non-informative prior". Jeffreys [1] suggested an "invariance" approach for choosing this prior, while Jaynes, [2] Kashyap, [3] and Shore and Johnson [4] have suggested using an information theoretic criterion. These methods are controversial, since the notion of finding an *a priori* density which conveys no *a priori* information is philosophically troublesome. A more conservative approach would be to use a criterion somewhat similar to that in (2.2.1), except that we try to find an estimator $\underline{\hat{\alpha}}=\underline{\hat{\alpha}}(\underline{z})$ whose average value, given many repeated experiments, $\underline{z}$, will be the true value $\underline{\alpha}_*$:

$$E_{Z|\Lambda}\left[\underline{\hat{\alpha}}(\underline{z}) \,\middle|\, \underline{\alpha}_* \right] = \underline{\alpha}_* \tag{2.2.5}$$

(Note that this expectation is over the observation space Z, whereas the MMSE method uses an expectation over the parameter space Λ.) Of all such unbiased estimates, we choose the one with the least variance given $\underline{\alpha}_*$:

$$\underline{\hat{\alpha}}(\underline{z}) - \min_{\underline{\hat{\alpha}}(\cdot)} E_{Z|\Lambda}\left[\,||\underline{\hat{\alpha}}(\underline{z})-\underline{\alpha}_*||^2 \,\middle|\, \underline{\alpha}_* \right] \tag{2.2.6}$$

Unfortunately, this Minimum Variance Unbiased Estimation method (MVUE) is not guaranteed to have a solution, and there does not appear to be any constructive procedure for solving the problem. (See Rao [5] for an excellent discussion of this approach.) We will therefore avoid the MVUE method in what follows.

3. Maximum Likelihood and Maximum A Posteriori Estimation

Because of the difficulties inherent in applying MMSE to many problems of interest, it is worthwhile considering alternative estimation methods which are simpler to apply, but are still well-behaved. Two techniques which have been extensively studied in the literature are Maximum Likelihood (ML) and Maximum A Posteriori (MAP) estimation. These procedures are applicable when we are given the conditional probability density, $p_{Z|\Lambda}(\underline{z}\,|\underline{\alpha})$, of the observation information, $\underline{z}$, as a function of the unknown to be estimated, $\underline{\alpha}$.

Maximum Likelihood (Fisher) estimation is used when the unknown $\underline{\alpha}$ must be considered a fixed but unknown constant (non-random variable). Thus no *a priori* probability density can be assigned to $\underline{\alpha}$. The ML estimate $\hat{\underline{\alpha}}_{ML}$ is then chosen as that value of $\underline{\alpha}$ which is most likely to have resulted in the given observation data:

$$\hat{\underline{\alpha}}_{ML} - \max_{\underline{\alpha}\in\Lambda} p_{Z|\Lambda}(\,\underline{z}\,|\,\underline{\alpha}\,) \tag{2.3.1}$$

The probability density $p_{Z|\Lambda}(\,\underline{z}\,|\,\underline{\alpha}\,)$ is called the "likelihood function" since it measures the likelihood of the value $\underline{\alpha}$ having caused $\underline{z}$.

Maximum A Posteriori (Bayesian) estimation is used when the unknown $\underline{\alpha}$ to be estimated can be considered to be a random variable with known *a priori* probability density $p_{\Lambda}(\underline{\alpha})$. The MAP estimate $\hat{\underline{\alpha}}_{MAP}$ is then chosen as the likeliest value of $\underline{\alpha}$ given the data $\underline{z}$:

$$\hat{\underline{\alpha}}_{MAP} - \max_{\underline{\alpha}\in\Lambda} p_{\Lambda|Z}(\,\underline{\alpha}\,|\,\underline{z}\,) \tag{2.3.2}$$

Bayes' Rule states that:

$$p_{\Lambda|Z}(\,\underline{\alpha}\,|\,\underline{z}\,) = \frac{p_{Z|\Lambda}(\,\underline{z}\,|\,\underline{\alpha}\,)\,p_{\Lambda}(\,\underline{\alpha}\,)}{p_{Z}(\,\underline{z}\,)} \tag{2.3.3}$$

Because $p_{Z}(\underline{z})$ is not a function of $\underline{\alpha}$, (2.3.2) is equivalent to:

$$\hat{\underline{\alpha}}_{MAP} - \max_{\underline{\alpha}\in\Lambda} p_{Z|\Lambda}(\,\underline{z}\,|\,\underline{\alpha}\,)\,p_{\Lambda}(\,\underline{\alpha}\,) \tag{2.3.4}$$

In computing the maximum of (2.3.1) or (2.3.2), it is often more convenient to use the logarithm of the probability density.

$$\boxed{\begin{array}{lll} \text{ML:} & \hat{\alpha}_{ML} - \max\limits_{\alpha\in\Lambda}\left[\log p_{Z|\Lambda}(\,z\,|\,\alpha\,)\right] & (2.3.5) \\ \text{MAP:} & \hat{\alpha}_{MAP} - \max\limits_{\alpha\in\Lambda}\left[\log p_{Z|\Lambda}(\,z\,|\,\alpha\,) + \log p_{\Lambda}(\,\alpha\,)\right] & \end{array}}$$

The only difference between ML and MAP estimation, clearly, is this second term $\log p_{\Lambda}(\alpha)$. As we let the *a priori* probability density of the unknown α approach a flat distribution over the space Λ (i.e. we know as little as possible *a priori* about the parameter value) then $p_{\Lambda}(\alpha)$ approaches a constant, and except in degenerate circumstances, $\hat{\alpha}_{MAP}$ will approach $\hat{\alpha}_{ML}$. Thus in many cases ML estimation can be mathematically considered to be a special case of MAP estimation in which the *a priori* probability density is asymptotically flat.

In general, neither ML nor MAP are unbiased, and the variance of their estimates can sometimes be high. In this respect, MMSE is clearly a superior estimation method. However, if we consider the behavior of ML and MAP as the number of observations grows infinitely large, then the performance of ML or MAP is often asymptotically equivalent to MMSE. Although a great many theorems have been proven about the asymptotic properties of ML and MAP, the most powerful of these theorems are quite complex, and the assumptions they make about the probability densities are difficult to verify in practice. Rather than state these theorems in detail, therefore, we will briefly sketch their assumptions and implications. (See Bar-Shalom, [6] Bhat, [7] Crowder, [8] Cramér, [9] Tse and Anton. [10])

In the following, we will only treat the ML case, but similar remarks apply to the MAP case. Suppose that the system generating the observation data is stationary and ergodic, and that the finite number of unknowns α are "structural" parameters which control the evolution of the observation sequence for all time, rather than "incidental" parameters whose influence lasts for only a finite time. Then provided certain existence and boundedness conditions apply, it can be shown that as the number of observation data points $N \to \infty$, that there exists at least one solution to the ML and MAP problems which is asymptotically consistent, efficient and normal. This is perhaps the most important property of ML and MAP estimation. Asymptotic consistency implies that if the true value of $\underline{\alpha}$ is $\underline{\alpha}_*$, then there exists at least one solution $\hat{\underline{\alpha}}$ to the ML problem (2.3.5) such that $\hat{\underline{\alpha}} \to \underline{\alpha}_*$ as $N \to \infty$. (Of course, under certain conditions, there may be other solutions to (2.3.5) which do not tend to $\underline{\alpha}_*$ as $N \to \infty$.) Note that because $\underline{z}$ is a stochastic variable, the estimate $\hat{\underline{\alpha}} = \hat{\underline{\alpha}}(\underline{z})$ is also a stochastic variable. The Cramér-Rao Lower Bound states that the covariance of *any* unbiased estimator of the Fisher variable $\underline{\alpha}$ is bounded below by:

$$\mathrm{Cov}_{Z|A}\left[\hat{\underline{\alpha}}(\underline{z}) - \underline{\alpha}_* \,\middle|\, \underline{\alpha}_* \right] \geq J_{ML}^{-1}(\underline{\alpha}_*) \tag{2.3.6}$$

$$\text{where: } J_{ML}(\underline{\alpha}_*) = E_{Z|A}\left[\frac{\partial^2 \log p(\underline{z} \mid \underline{\alpha})}{\partial \underline{\alpha}^2} \,\middle|\, \underline{\alpha}_* \right]$$

The fact that ML is asymptotically consistent and efficient implies that the covariance of the ML estimator $\hat{\underline{\alpha}}_{ML}$ asymptotically approaches the Cramér-Rao lower bound. Thus ML is "optimal" in the sense that no other asymptotically consistent estimator, not even MMSE, can have asymptotically lower variance. Finally, asymptotic normality implies that the probability distribution of the estimate $\hat{\underline{\alpha}}$ asymptotically approaches a Gaussian (normal) distribution with mean $\underline{\alpha}_*$ and variance $J_{ML}^{-1}(\underline{\alpha}_*)$.

Because both ML and MAP are asymptotically efficient, the Cramér-Rao Lower Bound J^{-1} gives a rough estimate of the variance of the estimator $\hat{\underline{\alpha}}$. This lower bound is also sometimes useful in devising stopping procedures for deciding when an iterative routine for locating $\hat{\underline{\alpha}}$ is "close enough" to the optimal answer.

Several other features of ML and MAP estimation are quite interesting. It can be shown that if an efficient estimator exists, then ML (and MAP) is efficient. It can also be shown that ML is invariant to one-to-one transformations of the space of unknowns. That is, if $\underline{\phi}=\underline{\phi}(\underline{\alpha})$ is a one-to-one function of $\underline{\alpha}$, and if $\hat{\underline{\alpha}}_{ML}$ is the ML estimate of $\underline{\alpha}$, then $\underline{\phi}(\hat{\underline{\alpha}}_{ML})$ is the ML estimate of $\underline{\phi}$:

$$\underline{\phi}(\hat{\underline{\alpha}}_{ML}) = \max_{\underline{\phi}} p(\, \underline{z} \mid \underline{\phi}\,) \tag{2.3.7}$$

MAP, however, is only invariant to linear transformations of the unknown. Another useful feature of both ML and MAP is that by focusing on the maximum of the probability density, they are insensitive to the shape of the tails of the distribution. Finally, both the ML and MAP estimation procedures take an elegant and computationally convenient form for many common types of probability densities, particularly Gaussians.

ML and MAP unfortunately have certain disadvantages. The optimal asymptotic properties of ML and MAP estimation only apply when the unknowns $\underline{\alpha}$ are a finite set of "structural" parameters, and the number of observation samples N is "nearly infinite". Intuitively, we will need to accumulate an infinite amount of information about each of the unknowns in order to estimate its value with infinite precision. Asymptotic consistency will not occur if, for example, the unknown $\underline{\alpha}$ is a stochastic signal of length N and the observation $\underline{z}$ is a noisy measurement of the signal process. In this case, increasing the observation interval also increases the number of unknown signal points to be estimated. Furthermore, in most stable, stationary and ergodic signal estimation problems, the contribution of each new signal sample to the observations decreases exponentially with time, so that only neighboring observation samples are significantly affected by this value. Accumulating more and more observation samples located farther and farther away from the unknown signal point will not significantly improve our estimate of that signal point.

Finally, even if the problem involves estimating a finite set of structural parameters from noisy observation data, given only a finite (small) set of data there is no guarantee that the ML or MAP estimates are the "best" we could find. This problem is particularly severe for very short data lengths, when the bias in the ML or MAP estimates can become quite noticeable.

4. Minimum Cross-Entropy Method

A completely different approach to the estimation problem is given by the Minimum Cross-Entropy Method (MCEM). [4] The methods we have discussed so far all start with observation data $\underline{z}$, whose stochastic behavior depends on the value of the unknown $\underline{\alpha}$. Using the conditional probability $p_{Z|A}(\underline{z}|\underline{\alpha})$ of the data given the unknown, these methods then construct a point estimator for the unknown. Confidence intervals for this estimate can then be derived via the Cramér-Rao lower bound, or through direct calculation of the estimator's variance. In some circumstances, however, we may be presented with observation information which is difficult to relate stochastically to the unknown. Suppose, for example, that the unknown $\underline{\alpha}$ is a Bayesian random variable with estimated *a priori* density $p_A(\underline{\alpha})$. Now suppose that information becomes available concerning the *form* of the true probability density $q(\underline{\alpha})$ of $\underline{\alpha}$. This information may specify some moments of the density, or otherwise restrict the functional form of $q(\underline{\alpha})$. Usually there is an infinite set of densities Ω that are not ruled out by the given information. The problem is to pick the "best" estimate of the probability density $q(\underline{\alpha})$ of the unknown which incorporates both the *a priori* information and the observation information, but which makes the fewest additional assumptions about the density. Standard Bayesian estimation is incapable of incorporating information such as this.

Shore and Johnson, [4] however, have proposed a new method of probabilistic inference, called the Minimum Cross-Entropy Principle, which generalizes the Maximum Entropy Principle [11,12]. They propose four "consistency postulates" which any reasonable method of inductive inference ought to satisfy. These postulates guarantee that the method will give consistent results when there are different ways of taking the same information into account (for example in different coordinate systems.) Given the *a priori* density $p_\Lambda(\underline{\alpha})$, and given that the actual density $q(\underline{\alpha})$ is an element of the set Ω, Shore and Johnson proved that the only estimation method which obeys all the postulates is to choose the density $q(\underline{\alpha})$ which minimizes the "cross-entropy" function:

$$\boxed{\text{MCEM:} \qquad \hat{q}(\cdot) - \min_{q \in \Omega} \int_{\Lambda} q(\underline{\alpha}) \log \frac{q(\underline{\alpha})}{p_\Lambda(\underline{\alpha})}\, d\underline{\alpha} \qquad (2.4.1)}$$

This Minimum Cross-Entropy estimate of q() can be viewed as the distribution which satisfies the constraints Ω, but is maximally non-committal with regard to missing information. Other authors have also proposed similar ideas; the name "cross-entropy" is due to Good, [13] though the method was first proposed by Kullback [14] and has been advocated in various forms by others under a variety of names, including "expected weight of evidence" and "directed divergence".

MCEM differs fundamentally from MAP and MMSE in that it uses observation data to estimate an entire probability density for the unknown, rather than simply generating a point estimate $\hat{\underline{\alpha}}$. Should a point or interval estimate of the unknown be required, we could first use the given information to calculate the MCEM density estimate $\hat{q}(\underline{\alpha})$, and then apply more standard point estimation methods to $\hat{q}(\underline{\alpha})$.

The cross-entropy expression (2.4.1) has a variety of elegant properties which we will use extensively throughout this thesis. The theorems below summarize some of the more important properties; proofs are contained in Appendix A, and a complete discussion may be found in Kullback [14] . The proofs of most of these properties rely on the fact that $\log x \le x-1$ with equality if and only if $x=1$.

Let us define $H(\mathrm{q})$ to be the cross-entropy function:

$$H(\mathrm{q}) = \int_{\Lambda} \mathrm{q}(\alpha) \log \frac{\mathrm{q}(\alpha)}{\mathrm{p}(\alpha)}\, d\alpha \tag{2.4.2}$$

In effect, $H(\mathrm{q})$ measures the mean information for discrimination in favor of q() against p() given that $\alpha \in \Lambda$. [14] Define the measure:

$$\mathrm{Q}(\hat{\Lambda}) = \int_{\hat{\Lambda}} \mathrm{q}(\alpha)\, d\alpha \tag{2.4.3}$$

for any measurable set $\hat{\Lambda}$, and define $\mathrm{P}(\hat{\Lambda})$ similarly. Also define a finite partition P of Λ as a finite collection of pairwise disjoint measurable sets $\mathrm{P} = \{\Lambda_i\}_{i=1}^{N}$ with $\Lambda_i \bigcap \Lambda_j = \varnothing$ for $i \ne j$, which together span the entire set $\Lambda = \bigcup_{i=1}^{N} \Lambda_i$.

<u>Theorem 2.4.1</u> $H(\mathrm{q})$ is strictly convex in q; that is, for any two probability densities q_1, q_2:

$$H(\lambda q_1 + (1-\lambda) q_2) \le \lambda H(\mathrm{q}_1) + (1-\lambda) H(\mathrm{q}_2) \qquad \text{for } \lambda \in (0,1) \tag{2.4.4}$$

with equality if and only if $\mathrm{q}_1(\alpha) = \mathrm{q}_2(\alpha)$ almost everywhere in Λ.

<u>Theorem 2.4.2</u> For any measurable set $\hat{\Lambda}$:

$$\int_{\hat{\Lambda}} \mathrm{q}(\alpha) \log \frac{\mathrm{q}(\alpha)}{\mathrm{p}(\alpha)}\, d\alpha \ge \mathrm{Q}(\hat{\Lambda}) \log \frac{\mathrm{Q}(\hat{\Lambda})}{\mathrm{P}(\hat{\Lambda})} \tag{2.4.5}$$

with equality if and only if $\dfrac{\mathrm{q}(\alpha)}{\mathrm{Q}(\hat{\Lambda})} = \dfrac{\mathrm{p}(\alpha)}{\mathrm{P}(\hat{\Lambda})}$ almost everywhere.

Theorem 2.4.3 Let $P = \{\Lambda_i\}$ be any arbitrary partition of the set Λ. Then:

$$\int_{\Lambda} q(\alpha) \log \frac{q(\alpha)}{p(\alpha)} \, d\alpha \geq \sum_i Q(\Lambda_i) \log \frac{Q(\Lambda_i)}{P(\Lambda_i)} \tag{2.4.6}$$

with equality if and only if $\dfrac{q(\alpha)}{Q(\Lambda_i)} = \dfrac{p(\alpha)}{P(\Lambda_i)}$ almost everywhere in $\alpha \in \Lambda_i$ for all i.

Theorem 2.4.4 Let $P' = \{\Lambda_{ij}\}$ be any subpartition of the partition $P = \{\Lambda_i\}$; that is $\bigcup_j \Lambda_{ij} = \Lambda_i$. Then:

$$\sum_{i,j} Q(\Lambda_{ij}) \log \frac{Q(\Lambda_{ij})}{P(\Lambda_{ij})} \geq \sum_i Q(\Lambda_i) \log \frac{Q(\Lambda_i)}{P(\Lambda_i)} \tag{2.4.7}$$

with equality if and only if $\dfrac{Q(\Lambda_{ij})}{Q(\Lambda_i)} = \dfrac{P(\Lambda_{ij})}{P(\Lambda_i)}$ for all i, j.

Theorem 2.4.5

$$\int_{\Lambda} q(\alpha) \log \frac{q(\alpha)}{p(\alpha)} \, d\alpha = \sup_P \sum_i Q(\Lambda_i) \log \frac{Q(\Lambda_i)}{P(\Lambda_i)} \tag{2.4.8}$$

where the supremum is calculated over all possible finite partitions P.

The first theorem is quite important, as the convexity of $H(q)$ allows us to draw powerful conclusions about the convergence behavior of our MCEM algorithms. The second theorem implies that if the original $p_\Lambda(\alpha)$ meets the observed constraints, then the unique solution to the cross-entropy problem is just the *a priori* density $q(\alpha) = p_\Lambda(\alpha)$, and the value of the cross-entropy at this minimum is zero. Taking $\tilde{\Lambda} = \Lambda$, this theorem also suggests that $H(q) \geq -\log p(\Lambda) \geq 0$ and so the cross-entropy is always bounded below. Theorems 2.4.3 and 2.4.4 represent another type of convexity property for $H(q)$, and indicate that grouping values of the unknown always decreases the cross-entropy. Theorem 2.4.5 can actually be used to define the cross-entropy for generalized probability densities (Radon-Nikodyn derivatives) where Lebesgue integration is used to evaluate $H(q)$. (See Pinsker [15] for further discussion of this point.) This feature is interesting because Maximum Likelihood methods can not be generalized to deal with continuous probability densities containing impulses. Finally, note that if our *a priori* estimate $p_A(\alpha)$ were flat, then the MCEM problem (2.4.1) would be equivalent to Maximum Entropy.

SECTION B - ESTIMATION OF MULTIPLE SIGNAL AND PARAMETER UNKNOWNS

5. System Model

The most general system model we will need to consider in this thesis is illustrated in figure 2.5.1:

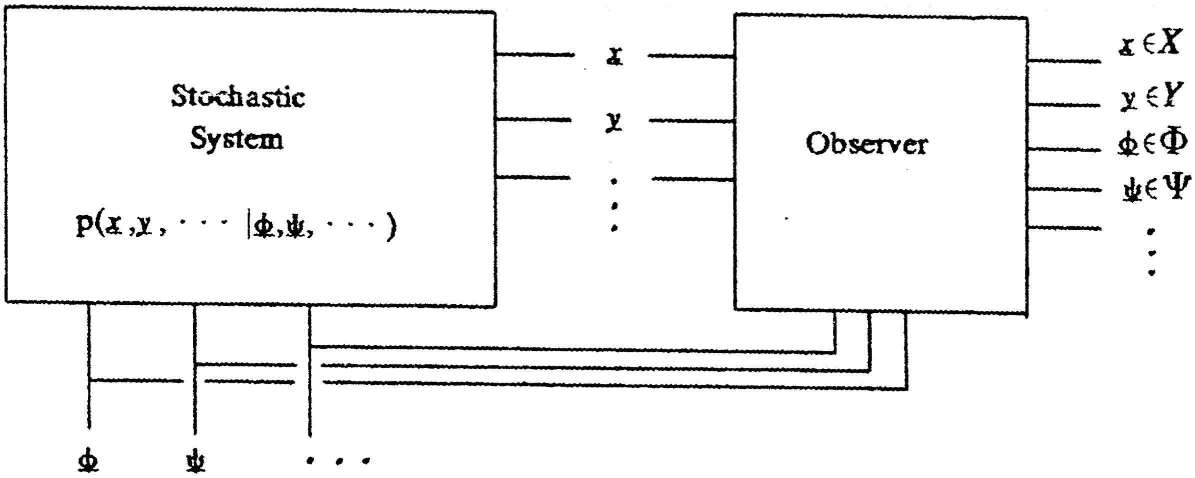


Figure 2.5.1 - General System Model

We are given a stochastic system with signal outputs $\underline{x}$, $\underline{y}$, ... and unknown parameters $\underline{\phi}$, $\underline{\xi}$, $\cdots$ whose behavior, we will assume, can be described by a probability density $p(\underline{x},\underline{y},\cdots|\underline{\phi},\underline{\xi},\cdots)$. Some of these parameters may be considered Fisher non-random but unknown constants, others may be considered Bayesian random variables with given *a priori* density. The observation data we are given is incomplete; rather than specifying the signals exactly, the data simply constrains the signals to lie within certain constraint sets $\underline{x}\in X$, $\underline{y}\in Y$, ... The parameters $\underline{\phi}$, $\underline{\xi}$, ... are also known to be restricted to certain sets $\underline{\phi} \in \Phi$, $\underline{\xi} \in \Psi$, etc. Given this noisy and incomplete data, together with the probability density $p(\underline{x}, \underline{y}, \cdots | \underline{\phi}, \underline{\xi}, \cdots)$ and the *a priori* densities of the Bayesian parameters, our goal will be to estimate all the unknown parameter and signal values.

Unfortunately, the number of different estimation approaches which could be applied to this problem grows geometrically with the number of unknowns that must be estimated. We can develop the essential features of the problem, however, by considering the simpler problem in which the system has only two unknowns, a signal $\underline{x}$ and a single set of Fisher or Bayesian parameters $\underline{\phi}$. The extension of the approaches we will develop to the more general case of multiple signals and/or parameters is straightforward.

We will generally assume that the probability density of the signal $\underline{x}$ given the parameters $\underline{\phi}$ is finite and non-zero over its domain:

$$0 < p(\underline{x} \mid \underline{\phi}) < \infty \qquad \text{for all } \underline{x} \in X, \ \underline{\phi} \in \Phi \tag{2.5.1}$$

If the parameters $\underline{\phi}$ are considered to be Bayesian random variables, then we will also assume that the *a priori* probability density $p(\underline{\phi})$ is given and that it is finite and non-zero over its domain:

$$0 < p(\underline{\phi}) < \infty \qquad \text{for all } \underline{\phi} \in \Phi \tag{2.5.2}$$

In fact, unless otherwise indicated, we will assume that all probability densities we will encounter are finite and non-zero over their entire domain. We will also generally assume that $\underline{x}$ and $\underline{\phi}$ are finite dimensional vectors.

From this point on, we will usually drop the subscripts of the probability densities, since the space over which the probability measure is defined is usually obvious from the context. Thus, for example, if x is an N component vector, $p(\underline{x})$ will refer to the probability density of the unknown x over all possible values of $x \in \mathbf{R}^N$, thus $\mathrm{p}(x)=\mathrm{p}_{\mathbf{R}^N}(x)$. If X is a subset of $\mathbf{R}^N$, then we define the probability of the set X to be:

$$\mathrm{p}(X) = \int_X \mathrm{p}(x)\, dx \qquad (2.5.3)$$

The conditional signal density $\mathrm{p}_X(x)$ given that x is restricted to the set $X \subseteq \mathbf{R}^N$ is proportional to the original density $\mathrm{p}(x)$ but renormalized to integrate to one over X:

$$\mathrm{p}_X(x) = \frac{\mathrm{p}(x)}{\mathrm{p}(X)} \qquad (2.5.4)$$

In a similar manner, we will define the probability $\mathrm{p}(X \mid \phi)$ that the signal belongs to the constraint set X given that the parameter value is ϕ, by:

$$\mathrm{p}(X \mid \phi) = \int_X \mathrm{p}(x \mid \phi)\, dx \qquad (2.5.5)$$

(This is well defined only because of assumption (2.5.1)). Define the probability density $\mathrm{p}(X, \phi)$ by Bayes' Rule:

$$\mathrm{p}(X, \phi) = \mathrm{p}(X \mid \phi)\, \mathrm{p}(\phi) = \int_X \mathrm{p}(x, \phi)\, dx \qquad (2.5.6)$$

In general, probability densities restricted to particular subsets X or Φ of the sample spaces will be defined by renormalizing the unrestricted density so that it integrates to one over the subset. Thus, for example:

$$\mathrm{p}_{X,\Phi}(x, \phi) = \frac{\mathrm{p}(x, \phi)}{\mathrm{p}(X, \Phi)} \qquad (2.5.7)$$

Restricted conditional densities will be defined similarly. Thus, the conditional density $\mathrm{p}_{\Phi \mid X}(\phi \mid X)$ of the parameter value $\phi \in \Phi$ given that the signal is an element of the set X, is given by:

$$p_{\Phi|X}(\phi\,|X\,) = \frac{p(\phi\,|X)}{\int_{\Phi} p(\phi\,|X)\,d\phi} = \frac{\int_X p(\phi,x)\,dx}{\int_X\int_{\Phi} p(\phi,x)\,d\phi\,dx} \tag{2.5.8}$$

6. Classical Filtering and Parameter Estimation Problems

Considerable insight into our estimation problem can be gained by first analyzing the simpler "classical" estimation problems which result when either the signal or the parameters are known exactly. Suppose the parameters ϕ were known exactly, so that the parameter constraint space contains only a single point, $\Phi=\{\phi_*\}$. Then the MMSE signal estimate would be calculated as:

$$\hat{x}_{MMSE} = \min_{\hat{x}\in X} E_{X|\Phi}\left[\ \|x-\hat{x}\|^2 \,\middle|\, \phi_*\ \right] \tag{2.6.1}$$

or if X is a convex set:

$$\hat{x}_{MMSE} - E_{X|\Phi}\left[\ x\ \middle|\ \phi_*\ \right] = \int_X x\ p_{X|\Phi}(x\ |\phi_*)\,dx \tag{2.6.2}$$

If this is too difficult to calculate, the MAP signal estimate $\hat{x}_{MAP}$ could be used instead, where we choose the most likely signal value given ϕ_* and given $x\in X$:

$$\hat{x}_{MAP} - \max_{x\in X} p_{X|\Phi}(\,x\ |\ \phi_*\,) \tag{2.6.3}$$

It is sometimes convenient to express this in a different form. Using Bayes' Rule, and noting that $p(X\,|\phi_*)$ does not depend on x, equation (2.6.3) can be written in the form:

$$\hat{x}_{MAP} - \max_{x\in X} p(x\ |\phi_*) \tag{2.6.4}$$

Finally, if ϕ were Bayesian, then this maximization could be written in yet another form:

$$\hat{x}_{MAP} - \max_{x\in X} p(x\ |\phi_*)\ p(\phi_*) \tag{2.6.5}$$

In the "classical" parameter identification problem, we assume that the signal value is known exactly, so that the signal constraint space contains only a single point, $X = \{x_\bullet\}$. If ϕ is a Bayesian random variable with *a priori* density $p(\phi)$, then we could estimate the parameters by using MMSE:

$$\hat{\phi}_{MMSE} - \min_{\hat{\phi} \in \Phi} E_{\Phi|X}\left[\, \|\phi - \hat{\phi}\|^2 \, \middle| \, x_\bullet \right] \tag{2.6.6}$$

or if Φ is convex:

$$\hat{\phi}_{MMSE} - E_{\Phi|X}\left[\, \phi \, \middle| \, x_\bullet \right] = \int_{\Phi} \phi \, p(\phi \, | \, x_\bullet) \, d\phi \tag{2.6.7}$$

If this is too difficult to calculate, or if the parameters are Fisher unknown constants, then we could use ML or MAP estimation to choose the most likely parameter value given the signal $x_\bullet$ and given that $\phi \in \Phi$. Applying Bayes' Rule, this maximization problem can be put into the form:

$$\begin{aligned} &\text{Fisher:} && \hat{\phi}_{ML} - \max_{\phi \in \Phi} p(\, x_\bullet \mid \phi \,) \\ &\text{Bayesian:} && \hat{\phi}_{MAP} - \max_{\phi \in \Phi} p(\, x_\bullet \mid \phi \,)\, p(\, \phi \,) \end{aligned} \tag{2.6.8}$$

The only difference between the Bayesian and Fisher models is the term $p(\phi)$.

Equations (2.6.7) and (2.6.8) give the "classical" filtering problems for estimating the output of a known system given incomplete observations. Equations (2.6.2) and (2.6.8) are the "classical" system identification problems when the system parameters must be identified from the system output. These "classical" algorithms provide a lower limit to the complexity of any estimation algorithm we develop for noisy or incomplete data, since the additional uncertainty when both the signal and the parameters are imperfectly known can only increase the difficulty of the solution.

Both these "classical" estimation problems are easy to solve for many signal models. For example, if the probability densities characterizing the system are Gaussian, the MMSE and MAP "classical" filtering problems (2.6.2) and (2.6.3) use the same linear smoothing filter to estimate $\underline{x}$. For signals generated by rational signal processes, this filtering operation can be calculated by a smoothing Kalman filter, or by a finite interval smoothing Wiener-Hopf filter. [16]

The "classical" system identification problem can also be easy to solve for certain system models. For example, suppose the signal process generating $\underline{x}$ is a Gaussian autoregressive (all-pole) system, where $\underline{\phi}$ are the unknown autoregressive coefficients. The ML estimate of $\underline{\phi}$ given $\underline{x}_*$ is then calculated by the covariance method of linear prediction, which solves linear equations for the parameters. [17, 18]

7. Estimation of Unknown Signal and Unknown Bayesian Parameters

When neither the signal nor the parameters are known, the estimation problem is considerably more difficult. We will discuss an MMSE estimation approach, as well as three completely different ML and MAP approaches. One gives the "optimal" parameter estimates, one gives "optimal" signal estimates, and the third tries to estimate both simultaneously and so falls somewhere in between. We will also present a separable density Minimum Cross-Entropy Method, which appears to combine the best properties of all the MAP methods. Furthermore, all the MAP methods can be treated as degenerate forms of this single MCEM method. To simplify the presentation, we will only discuss the Bayesian case in detail. The extension to Fisher parameters is straightforward and is given in section 8.

7.1. Minimum Mean Square Error Estimation (MMSE)

In most applications, the "best" estimates of the signal and parameters given $p(\underline{x},\underline{\phi})$ and $\underline{x}\in\mathcal{X}$, $\underline{\phi}\in\Phi$ would be calculated by the MMSE approach:

$$\begin{pmatrix}\hat{\underline{x}}\\ \hat{\underline{\phi}}\end{pmatrix} = E_{\mathcal{X},\Phi}\left[\begin{pmatrix}\underline{x}\\ \underline{\phi}\end{pmatrix}\right] = \int\limits_{\mathcal{X}}\int\limits_{\Phi}\begin{pmatrix}\underline{x}\\ \underline{\phi}\end{pmatrix} p_{\mathcal{X},\Phi}(\underline{x},\underline{\phi})\,d\underline{x}\,d\underline{\phi} \qquad (2.7.1)$$

If $\hat{\underline{x}}\in\mathcal{X}$ and $\hat{\underline{\phi}}\in\Phi$, then this estimator has the least possible mean square error. Unfortunately, even in relatively simple applications this multidimensional integral can be quite difficult to evaluate.

7.2. MAP Optimal Parameter Estimation (PARMAP)

An approach which is often easier is to first use MAP to estimate the parameters from the known information. The signal value can then be estimated by assuming that this parameter estimate is correct.

The "best" MAP estimate of the parameters (PARMAP) is given by that parameter value in Φ which is most likely to have resulted in our observation that $\underline{x}\in\mathcal{X}$. Using Bayes' Rule in an obvious way, we get:

$$\boxed{\textbf{PARMAP:}\quad \hat{\underline{\phi}}_{MAP} = \max_{\underline{\phi}\in\Phi} p(\mathcal{X},\underline{\phi}) = \max_{\underline{\phi}\in\Phi}\int\limits_{\mathcal{X}} p(\underline{x},\underline{\phi})\,d\underline{x} \qquad (2.7.2)}$$

This method makes no assumptions about the exact signal value in choosing the parameter estimates, but instead integrates over all feasible signal values. Unlike the MMSE estimate in (2.7.1), the PARMAP parameter estimate is usually biased for short data lengths. However, for many signal models which are stationary, stable and ergodic and for which the parameters ϕ are "identifiable" and "structural", the theorems discussed in section 3 can be applied to show that PARMAP's parameter estimates are asymptotically consistent, efficient and normal as the number of observations $N \to \infty$. For these models, no other estimation technique yields estimates with asymptotically lower variance.

Once the parameter estimate has been calculated, the signal can be estimated either by MMSE or by MAP techniques:

$$\hat{x} = E_{X|\Phi}\left[x \;\middle|\; \hat{\phi}_{MAP} \right] \qquad (2.7.3)$$

or:

$$\hat{x} - \max_{x \in X} p(\, x \mid \hat{\phi}_{MAP} \,) \qquad (2.7.4)$$

It is interesting to note that if the parameter estimate $\hat{\phi}_{MAP}$ is asymptotically consistent, so that $\hat{\phi}_{MAP} \to \phi_*$ as $N \to \infty$, then the PARMAP signal estimates will asymptotically approach the classical filtering estimator in (2.6.2) and (2.6.3), in which the exact parameter value ϕ_* is used:

$$E_{X|\Phi}[\, x \mid \hat{\phi}_{MAP} \,] - E_{X|\Phi}[\, x \mid \phi_* \,] \quad \text{as } N \to \infty \qquad (2.7.5)$$

7.3. MAP Optimal Signal Estimation (SIGMAP)

Since PARMAP gives the "best" parameter estimates by averaging over all signal values, it is tempting to think that the best MAP estimate of the signal (SIGMAP) would be given by averaging over all parameter values:

$$\boxed{\text{SIGMAP:} \quad \hat{x}_{MAP} - \max_{x \in X} p(\, x, \Phi \,) = \max_{x \in X} \int_{\Phi} p(x, \phi)\, d\phi \qquad (2.7.6)}$$

SIGMAP thus chooses the likeliest signal estimate given that $x \in X$ and $\phi \in \Phi$, and makes no assumptions about the exact parameter values. As a result, this signal estimate is usually quite different from that given by PARMAP in (2.7.3) or (2.7.4).

Unfortunately, the asymptotic consistency theorems of section 3 do not apply to this problem, and so the signal estimate generated by this technique is not necessarily asymptotically consistent or efficient, even when the system model is stationary, stable and ergodic. This can be easily explained by the same argument used in section 3. In the PARMAP parameter estimation case, each new observation adds new information about the parameter values; as $N \to \infty$ an infinite amount of information accumulates, allowing perfect (consistent) estimation. In the SIGMAP signal estimation problem, however, each signal value only affects neighboring observations, and gathering data from the remote past or future will not significantly improve the signal estimate at time n.

Given the signal estimate $\hat{x}_{MAP}$, a convenient and easily calculated parameter estimate would be:

$$\hat{\phi} = \mathrm{E}_{\Phi|X}\left[\phi \,\middle|\, \hat{x}_{MAP} \right] \tag{2.7.7}$$

or

$$\hat{\phi} - \max_{\phi \in \Phi} \mathrm{p}(\phi \mid \hat{x}_{MAP}) \tag{2.7.8}$$

These parameter estimates are quite different from those generated by PARMAP in (2.7.2), and as we will see in a later chapter, for most signal models they are not asymptotically consistent or efficient.

7.4. MAP Simultaneous Parameter and Signal Estimation (PSMAP)

Although the preceding PARMAP and SIGMAP approaches are simpler than the MMSE approach in (2.7.1), the probability densities $p(\underline{\phi} | X)$ and $p(\underline{x} | \Phi)$ can still be quite complicated functions, and the optimization required may be computationally difficult. An alternative, and usually much simpler approach, would be to try to select the combination of signal and parameter values which are simultaneously the most likely given the known information:

$$\boxed{\text{PSMAP:} \quad \hat{\underline{x}}, \hat{\underline{\phi}} = \max_{\underline{x} \in X, \underline{\phi} \in \Phi} p(\underline{x}, \underline{\phi}) \qquad (2.7.9)}$$

The solution to PSMAP must also satisfy:

$$\begin{cases} \hat{\underline{\phi}} = \max\limits_{\underline{\phi} \in \Phi} p(\hat{\underline{x}}, \underline{\phi}) \\ \hat{\underline{x}} = \max\limits_{\underline{x} \in X} p(\underline{x}, \hat{\underline{\phi}}) \end{cases} \tag{2.7.10}$$

although other points besides the global maxmimum might also satisfy (2.7.10). This method is identical to the LMAP procedure of Lim [19,20] and Bar-Shalom. [21] Unlike the previous methods, PSMAP does no averaging in selccting the estimates. Thus, since it is not necessary to integrate $p(\underline{x}, \underline{\phi})$ in solving PSMAP, the computation required is comparatively simple. Contrary to what Bar-Shalom implied, however, the asymptotic consistency theorems of section 3 do not apply to PSMAP, and thus the technique usually yields biased parameter and signal estimates even as the number of observations $N \to \infty$. For example, in an ARMA model estimation problem we will consider later, we will show that this optimality criterion dramatically overestimates the sharpness of peaks in the signal spectrum. The source of this bias can be made much clearer by rewriting the probability density which PSMAP maximizes in the following two ways:

$$p(x, \phi) = \underbrace{p(\phi, X)}_{\text{PARMAP estimator}} \quad p_{X|\Phi}(x \mid \phi) \tag{2.7.11}$$

$$p(x, \phi) = \underbrace{p(x, \Phi)}_{\text{SIGMAP estimator}} \quad p_{\Phi|X}(\phi \mid x). \tag{2.7.12}$$

The function which this procedure maximizes is thus similar to that maximized by the PARMAP or SIGMAP estimation techniques, except that it includes additional terms, $p_{X|\Phi}(x|\phi)$ or $p_{\Phi|X}(\phi|x)$, which bias the estimates. Because the parameter estimate will be asymptotically biased, $\hat{\phi} \neq \phi_*$, the PSMAP signal estimate $\hat{x}$ usually will asymptotically differ from the classical filtering estimate, in which the correct parameter value ϕ_* is used.

7.5. Minimum Cross-Entropy Method (MCEM)

PARMAP estimates the parameters by averaging over the signal constraint space, SIGMAP estimates the signal by averaging over the parameter constraint space, and PSMAP estimates both the signal and parameters simultaneously without any averaging. Considerations of symmetry would suggest that there ought to be a fourth estimation method, with the same structure as the other MAP methods, in which we average over both the signal and the parameters simultaneously in estimating the unknowns. A fourth method which will meet our needs is given by a somewhat unusual Minimum Cross-Entropy Method. Let $p(x, \phi)$ be the known joint probability distribution of x and ϕ, and let the observation information impose constraints $x \in X$ and $\phi \in \Phi$. The problem with using MMSE to estimate x and ϕ is that the probability density $p(x, \phi)$ is usually awkward to integrate. Let us therefore try approximating $p(x, \phi)$ with a separable probability density $q(x, \phi) = q_X(x) q_\Phi(\phi)$, and use Minimum Cross-Entropy to find the best separable approximation to $p(x,\phi)$ over the domain $X \times \Phi$.

$$\boxed{\text{MCEM:} \quad \hat{q}_X(\cdot),\ \hat{q}_\Phi(\cdot) \ - \ \min_{q_X, q_\Phi} H(q_X, q_\Phi) \qquad (2.7.13)}$$

where $H()$ is the cross-entropy function:

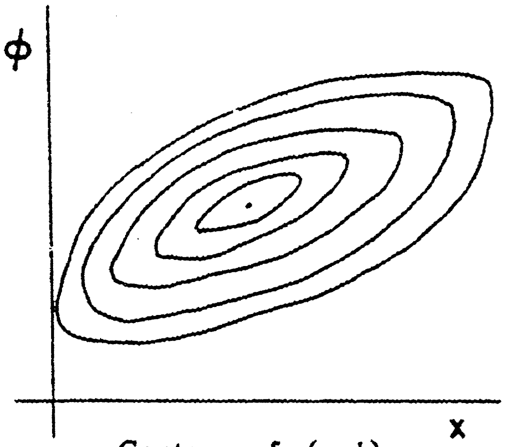
$$H(q_X, q_\Phi) = \int_X \int_\Phi q_X(x) q_\Phi(\phi) \log \frac{q_X(x) q_\Phi(\phi)}{p(x,\phi)} \, d\phi \, dx \qquad (2.7.14)$$

and where q_X and q_Φ are arbitrary probability densities constrained to integrate to one on X and Φ respectively:

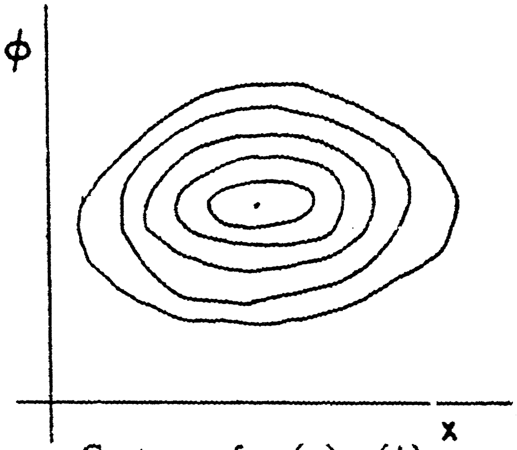
$$\int_X q_X(x)\, dx = 1 \quad \text{and} \quad q_X(x) \geq 0 \quad \text{for all } x \in X \qquad (2.7.15)$$

$$\int_\Phi q_\Phi(\phi)\, d\phi = 1 \quad \text{and} \quad q_\Phi(\phi) \geq 0 \quad \text{for all } \phi \in \Phi$$

This, we must admit, is an unusual approach, since the unknowns are almost always very closely interrelated. Intuitively, this separable density approximation $q_X(x)q_\Phi(\phi)$ will never be able to capture the exact contours of the original density $p(x,\phi)$. Nevertheless, it ought to put peaks with about the right width in about the right locations.



Contours of $p(x,\phi)$



Contours of $q_X(x)q_\Phi(\phi)$

Most importantly, once we have computed $\hat{q}_X$ and $\hat{q}_\Phi$, the signal and parameters can be estimated independently by applying point estimation methods such as MMSE or MAP to this separable density $\hat{q}_X(x)\hat{q}_\Phi(\phi)$. For example:

$$\hat{x} = E_X\left[\, x \,\middle|\, \hat{q}_X(\cdot)\,\right] = \int_X x \; \hat{q}_X(x)\, dx \qquad (2.7.16)$$

$$\hat{\phi} = E_\Phi\left[\, \phi \,\middle|\, \hat{q}_\Phi(\cdot)\,\right] = \int_\Phi \phi \; \hat{q}_\Phi(\phi)\, d\phi$$

We have thus decoupled the signal and parameter estimation problems, thereby (hopefully) reducing the complexity of the computation to the level of the classical estimation problems involving only a single unknown. As we will see, the forms of $q_X(x)$ and $q_\Phi(\phi)$ generated by minimizing (2.7.13) are often quite simple, and the values of $\hat{x}$ and $\hat{\phi}$ found from (2.7.16) are not only easy to compute, but are also often closer to the MMSE estimates than any of our MAP methods. We will also see in later sections that these MCEM estimates are often asymptotically consistent, just like PARMAP and MMSE.

Superficially, MCEM would appear to be unconnected with our first three MAP algorithms; after all, MCEM estimates an entire separable probability density using an information theoretic criterion, while the MAP algorithms only generate point estimates using a purely Bayesian approach. It is possible, however, to treat PARMAP, SIGMAP and PSMAP simply as degenerate forms of MCEM in which we not only restrict the fitted density $q(x,\phi)$ to be separable, but also restrict one or both of the densities q_X or q_Φ to be an impulse function. Thus we can restate these estimation methods in a cross-entropy framework as follows:

MCEM: Minimize $H(q_X, q_\Phi)$

PARMAP: Minimize $H(q_X, q_\Phi)$ but constrain $q_\Phi(\phi) = \delta(\phi - \hat{\phi})$

SIGMAP: Minimize $H(q_X, q_\Phi)$ but constrain $q_X(x) = \delta(x - \hat{x})$

PSMAP: Minimize $H(q_X, q_\Phi)$ but constrain $q_\Phi(\phi) = \delta(\phi - \hat{\phi})$ and $q_X(x) = \delta(x - \hat{x})$

Proof: We first consider PARMAP. The MCEM optimization criterion can be rewritten in the form:

$$H(q_X, q_\Phi) = -\int_\Phi \int_X q_X(x)\, q_\Phi(\phi) \log p(x,\phi)\, dx\, d\phi + \int_X q_X(x) \log q_X(x)\, dx + \int_\Phi q_\Phi(\phi) \log q_\Phi(\phi)\, d\phi \qquad (2.7.17)$$

Let $q_\Phi(\phi)$ approach the form of a delta function, $q_\Phi(\phi) = \delta(\phi - \hat{\phi})$, centered at some value $\hat{\phi} \in \Phi$. Note that although the term $\int_\Phi q_\Phi(\phi) \log q_\Phi(\phi)\, d\phi$ will become infinitely large, its value will be independent of $q_X(x)$ and will also be independent of the location of the delta function $\hat{\phi}$. We can thus ignore this term. Let $\tilde{H}(q_X, \hat{\phi})$ represent the remaining two terms:

$$\tilde{H}(q_X,\hat{\phi}) = \int_X\int_\Phi q_X(x)\,\delta(\phi-\hat{\phi})\log\frac{q_X(x)}{p(x,\phi)}\,dx\,d\phi \qquad (2.7.18)$$

$$= \int_X q_X(x)\log\frac{q_X(x)}{p(x,\hat{\phi})}\,dx$$

By theorem 2.4.2, this expression is bounded below by:

$$\tilde{H}(q_X,\hat{\phi}) \geq q_X(X)\log\frac{q_X(X)}{p(X,\hat{\phi})} = -\log p(X,\hat{\phi}) \qquad (2.7.19)$$

and for fixed $\hat{\phi}$ achieves this minimum cross-entropy at:

$$\hat{q}_X(x) = \frac{p(x,\hat{\phi})}{p(X,\hat{\phi})} = p_{X|\Phi}(x\,|\,\hat{\phi}) \qquad (2.7.20)$$

which is simply the conditional density of the signal given the parameter value $\hat{\phi}$. Substituting this solution into (2.7.18) therefore reduces the optimization over q_X and q_Φ to:

$$\text{PARMAP:}\quad \hat{q}_\Phi(\phi) = \delta(\phi-\hat{\phi})$$

$$\hat{q}_X(x) = p_{X|\Phi}(x\,|\,\hat{\phi}) \qquad (2.7.21)$$

$$\text{where: } \hat{\phi} = \min_{\phi\in\Phi}\left[-\log p(X,\phi)\right]$$

$$= \max_{\phi\in\Phi} p(X,\phi)$$

The estimate of the location of the delta function $q_\Phi(\phi) = \delta(\phi-\hat{\phi})$ is thus found by maximizing $p(X,\phi)$, and then the signal density $q_X(x)$ is simply estimated to be the conditional density of x given the parameter value $\hat{\phi}$. This, of course, is identical to the PARMAP procedure presented earlier.

The proof that SIGMAP results when we restrict q_X to be an impulse is identical, except with the roles of x and ϕ reversed.

To prove the result for PSMAP, let both densities approach delta functions, $q_{\mathcal{X}}(\mathbf{x})=\delta(\mathbf{x}-\hat{\mathbf{x}})$ and $q_{\Phi}(\boldsymbol{\phi})=\delta(\boldsymbol{\phi}-\hat{\boldsymbol{\phi}})$. Although the last two terms of the expression for $H(q_{\mathcal{X}},q_{\Phi})$ in (2.7.17) will become infinitely large, they will be independent of the location of the delta functions, $\hat{\mathbf{x}}$ and $\hat{\boldsymbol{\phi}}$, and can thus be ignored. The remaining first term reduces to:

$$-\int_{\mathcal{X}}\int_{\Phi} \delta(\mathbf{x}-\hat{\mathbf{x}})\delta(\boldsymbol{\phi}-\hat{\boldsymbol{\phi}}) \log p(\mathbf{x},\boldsymbol{\phi})\, d\mathbf{x}\, d\boldsymbol{\phi} = -\log p(\hat{\mathbf{x}},\hat{\boldsymbol{\phi}}) \tag{2.7.22}$$

Thus the cross entropy minimization problem reduces to:

$$\begin{aligned} \text{PSMAP:}\quad \hat{q}_{\mathcal{X}}(\mathbf{x}) &= \delta(\mathbf{x}-\hat{\mathbf{x}}) \\ \hat{q}_{\Phi}(\boldsymbol{\phi}) &= \delta(\boldsymbol{\phi}-\hat{\boldsymbol{\phi}}) \\ \text{where: } \hat{\mathbf{x}},\hat{\boldsymbol{\phi}} &\leftarrow \min_{\mathbf{x}\in\mathcal{X},\boldsymbol{\phi}\in\Phi} \left[-\log p(\hat{\mathbf{x}},\hat{\boldsymbol{\phi}}) \right] \\ &\leftarrow \max_{\mathbf{x}\in\mathcal{X},\boldsymbol{\phi}\in\Phi} p(\mathbf{x},\boldsymbol{\phi}) \end{aligned} \tag{2.7.23}$$

which is precisely the PSMAP problem. □

This interpretation of PARMAP, SIGMAP and PSMAP as degenerate forms of MCEM is extremely important, and will be heavily exploited throughout this thesis in order to achieve a unified treatment of all these estimation methods. For convenience, table 2.1 lists the cross-entropy expressions appropriate for each of our four MCEM and MAP methods.

8. Fisher Model

Variations of the above PARMAP and PSMAP estimation approaches can be easily devised which are suitable for use when the parameters are non-random (Fisher) variables. The chief difference in the Fisher case is that because the parameters are non-random, it will not be possible to calculate their conditional expectation, nor will it

Table 2.1 - Bayesian Estimation Methods

Method	Likelihood Function	$H(q_X, q_\Phi)$	Constraints
MCEM:	*none*	$\int_X \int_\Phi q_X(\mathbf{x}) q_\Phi(\boldsymbol{\phi}) \log \frac{q_X(\mathbf{x}) q_\Phi(\boldsymbol{\phi})}{p(\mathbf{x}, \boldsymbol{\phi})} \, d\mathbf{x} \, d\boldsymbol{\phi}$	*none*
PARMAP:	$\max_{\boldsymbol{\phi} \in \Phi} p(X, \boldsymbol{\phi})$	$\int_X \int_\Phi q_X(\mathbf{x}) q_\Phi(\boldsymbol{\phi}) \log \frac{q_X(\mathbf{x})}{p(\mathbf{x}, \boldsymbol{\phi})} \, d\mathbf{x} \, d\boldsymbol{\phi}$	$q_\Phi(\boldsymbol{\phi}) = \delta(\boldsymbol{\phi} - \hat{\boldsymbol{\phi}})$
SIGMAP:	$\max_{\mathbf{x} \in X} p(\mathbf{x}, \Phi)$	$\int_X \int_\Phi q_X(\mathbf{x}) q_\Phi(\boldsymbol{\phi}) \log \frac{q_\Phi(\boldsymbol{\phi})}{p(\mathbf{x}, \boldsymbol{\phi})} \, d\mathbf{x} \, d\boldsymbol{\phi}$	$q_X(\mathbf{x}) = \delta(\mathbf{x} - \hat{\mathbf{x}})$
PSMAP:	$\max_{\mathbf{x} \in X, \boldsymbol{\phi} \in \Phi} p(\mathbf{x}, \boldsymbol{\phi})$	$\int_X \int_\Phi q_X(\mathbf{x}) q_\Phi(\boldsymbol{\phi}) \log \frac{1}{p(\mathbf{x}, \boldsymbol{\phi})} \, d\mathbf{x} \, d\boldsymbol{\phi}$	$\begin{cases} q_X(\mathbf{x}) = \delta(\mathbf{x} - \hat{\mathbf{x}}) \\ q_\Phi(\boldsymbol{\phi}) = \delta(\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}) \end{cases}$

Note: Fisher Estimation methods replace $p(\mathbf{x}, \boldsymbol{\phi})$ by $p(\mathbf{x} | \boldsymbol{\phi})$.

be possible to estimate the signal without making some specific assumption about the values of the parameters. It will thus not be possible to devise a Maximum Likelihood version of the SIGMAP algorithm, unless we are willing to invent a "non-informative" prior density for the parameters.

An ML version of PARMAP results when we choose the parameter value $\hat{\phi} \in \Phi$ which is most likely to have resulted in a signal value x which belongs to the constraint set X:

$$\boxed{\textbf{PARML:} \quad \hat{\phi}_{ML} - \max_{\phi \in \Phi} p(X \mid \phi) \qquad (2.8.1)}$$

Similarly, an ML version of PSMAP results when we choose the combination of parameter value $\phi \in \Phi$ and signal value $x \in X$ which are most likely:

$$\boxed{\textbf{PSML:} \quad \hat{x}, \hat{\phi} - \max_{x \in X, \phi \in \Phi} p(x \mid \phi) \qquad (2.8.2)}$$

The only difference between the ML and MAP versions of these algorithms is that the ML version effectively assumes that the *a priori* density $p(\phi)$ is flat over the range of interest (compare (2.8.1) with (2.7.2) and (2.8.2) with (2.7.9).)

A rather different approach to the Fisher problem would be to examine the asymptotic behavior of our Bayesian algorithms as the *a priori* density $p(\phi)$ becomes "flat". It is easiest to examine this issue within the cross-entropy framework discussed in the previous section. Substituting $p(x,\phi) = p(x \mid \phi) p(\phi)$, into our cross-entropy expression $H(q_X, q_\Phi)$ gives:

$$H(q_X, q_\Phi) = H_{ML}(q_X, q_\Phi) - \int_\Phi q_\Phi(\phi) \log p(\phi) \, d\phi \qquad (2.8.3)$$

where H_{ML} is a "Fisher" cross-entropy function in which the density $p(x,\phi)$ has been replaced by $p(x|\phi)$:

$$H_{ML}(q_X,q_\Phi) = \int_X\int_\Phi q_X(x)q_\Phi(\phi)\log\frac{q_X(x)q_\Phi(\phi)}{p(x|\phi)}\,d\phi\,dx \tag{2.8.4}$$

As $p(\phi)$ becomes asymptotically flat, the second term in (2.8.3) should become relatively independent of $q_\Phi(\phi)$. We would therefore expect that the separable density $\hat{q}_X$, $\hat{q}_\Phi$ which minimizes the Bayesian cross-entropy $H(q_X,q_\Phi)$ to also asymptotically minimize the "Fisher" cross-entropy $H_{ML}(q_X,q_\Phi)$ in the limit as $p(\phi)$ becomes "flat". We will therefore define the "Fisher Minimum Cross-Entropy Method" as fitting a separable "likelihood" function $q_X(x)q_\Phi(\phi)$ to the given model "likelihood" function $p(x|\phi)$ by minimizing the Fisher cross-entropy:

$$\boxed{\textbf{Fisher MCEM:}\qquad \hat{q}_X(\cdot),\ \hat{q}_\Phi(\cdot) \leftarrow \min_{q_X,q_\Phi} H_{ML}(q_X,q_\Phi) \qquad (2.8.5)}$$

The justification for this algorithm is that it is a limiting form of the Bayesian algorithm as our *a priori* knowledge becomes infinitesimally small. Beware, however, that since $p(x|\phi)$ may not integrate to a finite number over $X\times\Phi$, many of the properties of cross-entropy presented in section 4 may not strictly apply to H_{ML}; in particular, its minimum value may be $-\infty$.

Constraining the parameter density q_Φ to be an impulse function results in an algorithm which is identical to PARML, while constraining both the parameter and signal densities q_X and q_Φ to be impulse functions leads to the PSML algorithm. Constraining only the signal density q_X to be an impulse gives a "Fisher" form of SIGMAP.

9. Extension to More General System Models

When the system model has several signal outputs and several sets of parameters, some Fisher and some Bayesian, the number of possible estimation approaches rises dramatically. The different signals and parameters could be estimated separately or jointly in many different combinations and orderings. For example, suppose we have a system with two signal outputs $\underline{x}$, $\underline{y}$, a Fisher set of parameters $\underline{\phi}$, and a Bayesian set of parameters $\underline{\psi}$. One "obvious" MCEM approach would be to hypothesize a flat *a priori* density $p(\underline{\phi})$ for the Fisher parameters, and then fit a separable probability density:

$$q(\underline{x},\underline{y},\underline{\psi},\underline{\phi}) = q_x(\underline{x})q_y(\underline{y})q_\psi(\underline{\psi})q_\phi(\underline{\phi}) \qquad (2.9.1)$$

to the given density $p(\underline{x},\underline{y},\underline{\psi}|\underline{\phi})$ by minimizing the cross-entropy expression:

$$H(q_x,q_y,q_\psi,q_\phi) = \int d\underline{x} \int d\underline{y} \int d\underline{\psi} \int d\underline{\phi}\; q_x(\underline{x})q_y(\underline{y})q_\psi(\underline{\psi})q_\phi(\underline{\phi}) \log \frac{q_x(\underline{x})q_y(\underline{y})q_\psi(\underline{\psi})q_\phi(\underline{\phi})}{p(\underline{x},\underline{y},\underline{\psi}|\underline{\phi})}$$

Alternative hybrid MAP/MCEM methods could then be devised by constraining one or more of these densities to be impulse functions. Purely MAP procedures result when all but at most one of the densities are constrained to be impulse functions. Still other estimation methods could be devised by jointly estimating two or more variables within the estimation procedures. Thus we could combine $\underline{x}$ and $\underline{y}$ in the problem above by choosing to fit a separable density of the form:

$$q(\underline{x},\underline{y},\underline{\psi},\underline{\phi}) = q_{x,y}(\underline{x},\underline{y})q_\psi(\underline{\psi})q_\phi(\underline{\phi}) \qquad (2.9.2)$$

to the given density. Since densities of the form (2.9.1) are a proper subset of the class of densities (2.9.2), smaller cross entropies can be achieved by partially separated densities (2.9.2) than by fully separated densities like (2.9.1).

$$\min_{q=q(\underline{x},\underline{y})q(\underline{\psi})q(\underline{\phi})} H(q) \leq \min_{q=q(\underline{x})q(\underline{y})q(\underline{\psi})q(\underline{\phi})} H(q) \qquad (2.9.3)$$

In this sense, better estimates are achieved by combining variables as far as possible.

The drawback, of course, is that generating point estimates of x and y from the joint density $\hat{q}(x,y)$ can be more complicated than generating estimates from separate densities $\hat{q}(x)$ and $\hat{q}(y)$.

SECTION C - OPTIMIZATION THEORY

10. Existence and Uniqueness of Global Maxima

All the MCEM, MAP and ML estimation apprpaches we have discussed require maximizing or minimizing a function over a given domain. It is useful, therefore, to consider under what conditions such a maximization has a finite solution and when this solution is unique. The case when the unknowns $\underline{x}$ and $\underline{\phi}$ are finite dimensional is well understood, and a variety of powerful theorems can be applied. Similar theorems also apply if $\underline{x}$ and/or $\underline{\phi}$ are infinitely long, or if we are minimizing a cross-entropy function over an infinite dimensional space of probability densities. However, in this case it is necessary to use weak topologies, and the wording of the theorems is more complicated. In this section, we will present some well known results for finite and infinite dimensional Hilbert spaces. Proofs of the finite dimensional theorems can be found in Luenberger [22] or Ortega and Rheinboldt [23] ; proofs of the generalization to infinite dimensional spaces may be found in Goldstein [24] Vainberg [25] , or Demyanov and Rubinov [26] .

Because maximizing a function $F(\underline{x})$ is equivalent to minimizing $-F(\underline{x})$, to simplify the presentation we will restrict our attention to minimization problems.

10.1. Existence of Global Minima

Let F be a function mapping the domain Λ to the real line, $F:\Lambda\rightarrow\mathbb{R}$. We assume that Λ has norm $\|\cdot\|$. A point $\underline{\alpha}_*\in\Lambda$ is a local minimizer of F if there is an open neighborhood $S_\delta=\left\{\underline{\alpha}\ \middle|\ \|\underline{\alpha}-\underline{\alpha}_*\|<\delta\ ,\ \delta>0\right\}$ of $\underline{\alpha}_*$ with radius δ, such that:

$$F(\alpha) \geq F(\alpha_*) \quad \text{for all } \alpha \in S \cap \Lambda \tag{2.10.1}$$

If strict inequality holds for $\alpha \neq \alpha_*$ in $S \cap \Lambda$, then α_* is a proper local minimizer. If $F(\alpha) \geq F(\alpha_*)$ for all $\alpha \in \Lambda$, then α_* is a global minimizer of F on Λ. It is well known that if $F(\alpha)$ is continuously differentiable in α, and if α_* is a local minimizer of F in the interior of Λ, then $F'(\alpha_*) = \left.\frac{dF(\alpha)}{d\alpha}\right|_{\alpha_*} = 0$. If α_* is a local minimizer of F on the boundary of Λ, then $F'(\alpha_*)$ must be inwardly normal to the boundary at α_*. Technically, this means that:

$$F'(\alpha_*)^T h \geq 0 \tag{2.10.2}$$

for all "sequentially tangent vectors" h at α_*. (A vector h is called a sequential tangent of Λ at α_* if there are a sequence $\{\alpha_k\}$ of points in Λ and a sequence of positive numbers $\{t_k\}$ such that:

$$\lim_{k \to \infty} \frac{\alpha_k - \alpha_*}{t_k} = h \qquad \text{and} \qquad \lim_{k \to \infty} t_k = 0 \tag{2.10.3}$$

Fur further details, see Hestenes, chapter 4 [27].) See figure 2.10.1 for an illustration of interior and boundary local minimizers.

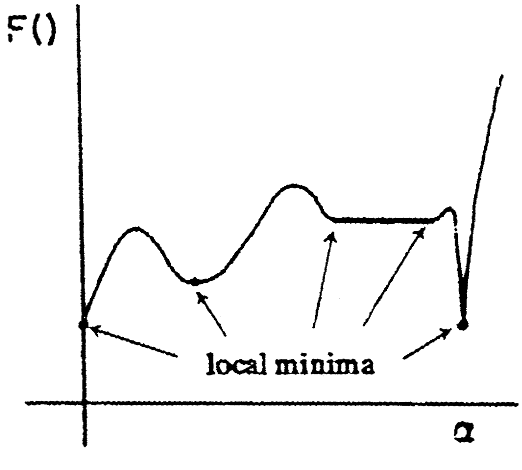


Figure 2.10.1 - Local Minimizers

In general $F(\alpha)$ may not have any minimizing point. In order to guarantee that $F(\alpha)$ will attain a minimum on Λ, we will have to restrict the domain Λ and the function $F(\alpha)$ appropriately:

> Theorem 2.10.1: Suppose $F(\alpha)$ is a continuous function for all $\alpha \in \Lambda$, and suppose that Λ is a compact set (if α is finite dimensional, this is equivalent to requiring that Λ be closed and bounded.) Then $F(\alpha)$ has at least one global minimizer $\alpha_* \in \Lambda$.

Unfortunately, in most of our applications the domain Λ will not be bounded and thus will not be compact. Suppose, however, that we choose an initial estimate $\hat{\alpha}_0$ of α, and form the "level set" $\Lambda_0 = \left\{ \alpha \,\middle|\, F(\alpha) \leq F(\hat{\alpha}_0) \right\}$. Then if Λ_0 is compact, $F(\alpha)$ must have a global minimizer on Λ_0, and this global minimizer must also clearly be a global minimizer on the entire domain Λ.

10.2. Uniqueness of Global Minimizers, Convex Functions

In general, even if the problem $\min_{\alpha \in \Lambda} F(\alpha)$ has a solution, this solution may not be unique. Furthermore, if $F(\alpha)$ has multiple peaks so that it has local as well as global minima, an optimization routine may have difficulty locating the correct global minimum solution. A useful set of conditions for guaranteeing the uniqueness of global minima revolves around the notion of convex sets and functions. (See chapter 1, section 5.) To quickly review, the set Λ is convex if for any two points $\alpha, \beta \in \Lambda$ the line connecting α and β is also contained within Λ:

$$\lambda\alpha + (1-\lambda)\beta \in \Lambda \qquad \text{for all } 0<\lambda<1 \tag{2.10.4}$$

A function $F:\Lambda \rightarrow \mathbf{R}$ is convex on a convex set Λ if, for all $\alpha, \beta \in \Lambda$:

$$F(\lambda\underline{\alpha}+(1-\lambda)\underline{\beta}) \leq \lambda F(\underline{\alpha}) + (1-\lambda)F(\underline{\beta}) \qquad \text{for } 0<\lambda<1 \tag{2.10.5}$$

The function F will be called strictly convex on Λ if strict inequality holds in (2.10.5) when $\underline{\alpha}\neq\underline{\beta}$. F will be called uniformly convex on Λ if there exists a constant $c>0$ such that for all $\underline{\alpha},\underline{\beta}\in\Lambda$ and $0<\lambda<1$:

$$\lambda F(\underline{\alpha}) + (1-\lambda)F(\underline{\beta}) - F(\lambda\underline{\alpha}+(1-\lambda)\underline{\beta}) \geq c\,\lambda(1-\lambda)\,||\underline{\alpha}-\underline{\beta}||^2 \tag{2.10.6}$$

For example, a quadratic function $F(\underline{\alpha})=\underline{\alpha}^T Q\underline{\alpha}$, where Q is positive definite, $Q\geq c\,I>0$, is uniformly convex. Clearly uniform convexity implies strict convexity, which in turn implies convexity.

If F has a second order (Frechet) derivative on a convex set Λ, then F will be convex on Λ if $F''(\underline{\alpha}\cdot) = \left.\dfrac{d^2F(\underline{\alpha})}{d\underline{\alpha}^2}\right|_{\underline{\alpha}\cdot}$ is positive semidefinite on Λ. F will be strictly convex on Λ if $F''(\underline{\alpha}\cdot)$ is positive definite on Λ, and it will be uniformly convex on Λ if and only if $F''(\underline{\alpha}\cdot)$ is uniformly positive definite on Λ, so that there exists a constant $c>0$ such that:

$$\underline{\alpha}^T F''(\underline{\alpha}\cdot)\underline{\alpha} \geq c\,||\underline{\alpha}||^2 \qquad \text{for all } \underline{\alpha}\in\Lambda \tag{2.10.7}$$

The importance of convex sets and convex functions, for our purposes, is that characterizing the global and local minimizers of a convex function is easy.

Theorem 2.10.2: Suppose $F:\Lambda\to R$ is a continuous and proper (i.e. nowhere equal to $-\infty$) convex function over a convex, closed and non-empty (but not necessarily bounded) set Λ. Then the set of all global minimizers of F on Λ is closed and convex (though possibly empty) and any local minimizer will also be a global minimizer. If in addition the set Λ is bounded, then F will have at least one global minimizer on Λ.

Thus, as shown in figure 2.10.2, if F is convex on Λ and $\underline{\alpha}$, $\underline{\beta}$ are both global minimizers on Λ, then every point on the line connecting $\underline{\alpha}$ and $\underline{\beta}$ must also be a global minimizer:

$$F(\,\lambda\underline{\alpha}+(1-\lambda)\underline{\beta}\,) = F(\underline{\alpha}) = F(\underline{\beta}) \tag{2.10.8}$$

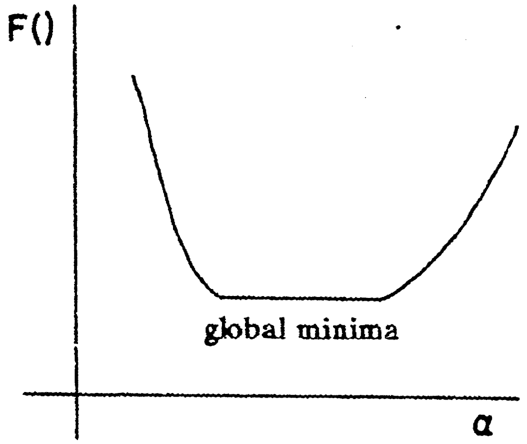


Figure 2.10.2 - Convex Function with Global Minimum

Note that this theorem is true both for finite dimensional and infinite dimensional Hilbert spaces (see Demyanov and Rubinov [26]). If we strengthen the conditions on F to strict convexity, then any global minimum must be unique:

> Theorem 2.10.3: If the conditions of theorem 2.10.4 hold, but F is strictly convex, then F can have at most one global minimizer on Λ. If Λ is also bounded, then F has a unique global minimizer on Λ. Furthermore, if the domain of F is finite dimensional, and F has a global minimizer on Λ, then $F(\underline{\alpha})\to\infty$ as $||\underline{\alpha}||\to\infty$, and all level sets will be bounded and compact.

Again, this theorem applies to both finite and infinite dimensional Hilbert spaces. If the function F is uniformly convex, then it is guaranteed to achieve its minimum even if Λ is not bounded:

Theorem 2.10.4: [Vainberg Thm 9.4 [25]] Let $F:\Lambda\to R$ be a proper uniformly convex functional on a convex, closed and non-empty (but not necessarily bounded) set Λ in a Hilbert space. Assume that F has continuous first and second order (Frechet) derivatives on Λ. Then F is bounded below and has a unique global and local minimizer $\underline{\alpha}_*$ in Λ. In fact, all level sets of F will be bounded, and $F(\underline{\alpha})\to\infty$ as $||\underline{\alpha}||\to\infty$.

Clearly these theorems will apply even if F itself is not convex, but there exists a continuous monotonically increasing function $g:R\to R$ such that $g(F(\underline{\alpha}))$ is convex. We will also sometimes need to maximize a concave function $F(\underline{\alpha})$ over a domain Λ (we define $F(\underline{\alpha})$ to be concave if $-F(\underline{\alpha})$ is convex.) Maximizing $F(\underline{\alpha})$ is equivalent to minimizing $-F(\underline{\alpha})$, however, and thus statements similar to the above theorems can be made about maximizing concave functions, or about finding local or global maxima in general.

A very useful special case for our estimation problem is when the logarithm of the probability density, $\log p(\underline{x},\underline{\phi})$, is concave. More precisely, we will call a function $F(\underline{\alpha})$ "log concave" if

$$F(\lambda\underline{\alpha}+(1-\lambda)\underline{\beta}) \geq F(\underline{\alpha})^{\lambda}F(\underline{\beta})^{1-\lambda} \qquad \text{for all } 0<\lambda<1 \qquad (2.10.9)$$

Strict and uniform log concavity are defined in obvious ways. One reason this case is so interesting is that Prékopa [28,29] has shown that if the constraint sets X and Φ are convex and $p(\underline{x},\underline{\phi})$ is log concave, then the PARMAP and SIGMAP densities $p(X,\underline{\phi})$ and $p(\underline{x},\Phi)$ are also log concave in $\underline{\phi}$ and $\underline{x}$ respectively. When $p(\underline{x},\underline{\phi})$ is log concave, therefore, these convexity theorems can be used to characterize the existence and uniqueness of the solutions to all three MAP algorithms. (See Appendix E.)

11. Conclusion

In this chapter, we have discussed a variety of MMSE, ML, MAP and MCEM procedures for estimating the signal and parameters of an unknown system. In order to apply these techniques, a model of the system must be assumed in which the joint probability density of the signal and parameters is specified. In the classical estimation problem, either the parameters or signal values are known, and the estimation procedure is straightforward. Unfortunately, when several signals and/or parameters are unknown, the estimation problem is much more complex and difficult to solve. The best estimation method in general is Minimum Mean Square Error, in which we use the available information to compute the expected value of the unknowns. This method yields unbiased estimates with the least possible variance; however, evaluating the multi-dimensional integrals can be quite difficult. The method is also not applicable to Fisher non-random parameters, and if the constraint sets are not convex, then the conditional expectation may not satisfy the constraints. Three different MAP and ML approaches were also suggested; one gives the "optimal" parameter estimates (PARMAP, PARML), one gives the "optimal" signal estimates (SIGMAP), and one tries to estimate the parameters and signals together (PSMAP, PSML). A fourth Minimum Cross-Entropy Method was also proposed, which optimally fits a separable probability density to the given density. This MCEM approach serves to unify and generalize our treatment of these estimation problems, since the other MAP methods can be considered degenerate forms of MCEM in which one or more estimated densities are constrained to be impulse functions. All of the MAP optimization approaches are guaranteed to have solutions if the constraint sets are compact (or weakly compact) and the probability densities are continuous (or weakly upper semi-continuous). If in addition, the constraint sets are convex and the objective function can be transformed into a uniformly convex function, then the global and local minima coincide, and can be shown to be unique.

Because of the double uncertainty in the signal and parameter values, all of the estimation techniques are considerably more complicated than the "classical" estimation problem. In the next chapter, however, we will derive iterative algorithms for solving these problems which reduce the computation on each pass to something quite similar to the "classical" estimation problems. These algorithms therefore will provide a fast, simple and elegant procedure for signal reconstruction and model estimation given incomplete observation data.

References

1. Harold Jeffreys, *Theory of Probability,* Oxford at the Clarendon Press (1961).
2. E.T. Jaynes, "Prior Probabilities," *IEEE Trans. Syst. Sci. Cybern.* SSC-4, pp.227-241 (1968).
3. R.L. Kashyap, "Prior Probability and Uncertainty," *IEEE Trans. Info. Theory* IT-17(6), pp.641-650 (Nov 1971).
4. John E. Shore and Rodney W. Johnson, "Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy," *IEEE Trans. Info. Theory* IT-26(1), pp.26-37 (Jan 1980).
5. C.R. Rao, *Linear Statistical Inference and Its Applications,* John Wiley & Sons, New York (1965).
6. Yaakov Bar-Shalom, "On the Asymptotic Properties of the Maximum Likelihood Estimate Obtained from Dependent Observations," *Jour. Royal Stat. Soc., Ser. B* 33(1), pp.72-77 (1971).
7. B. R. Bhat, "On the Method of Maximum Likelihood for Dependent Observations," *J. Royal Stat. Soc., Ser. B* 36(1), pp.48-53 (1974).
8. Martin J. Crowder, "Maximum Likelihood Estimation for Dependent Observations," *Jour. of the Royal Statist. Soc., Ser. B, Vol. 38*(1), pp.45-53 (1976).
9. Harald Cramér, *Mathematical Methods of Statistics,* Princeton Univ. Press, Princeton, N.J. (1946).
10. Edison Tse and John J. Anton, "On the Identifiability of Parameters," *IEEE Trans. on Auto. Control* AC-17(5), pp.637-646 (Oct. 1972).
11. J. Edward and M.M. Fitelson, "Notes on Maximum Entropy Processing," *IEEE Trans. Info. Theory* IT-19, pp.232-234 (March 1973).
12. John P. Burg, *Maximum Entropy Spectral Analysis,* Ph.D. Thesis, Stanford University (May 1975).
13. I.J. Good, "Maximum Entropy for Hypothesis Formulation, Especially for Multidimensional Contingency Tables," *Annals of Math. Stat.* 34, pp.911-934 (1963).
14. Solomon Kullback, *Information Theory and Statistics,* John Wiley & Sons, New York (1959).
15. Pinsker, *Information and Information Stability of Random Variables,* Holden Day, San Francisco (1964). translated by A. Feinstein
16. H. L. Van Trees, *Detection, Estimation and Modulation Theory,* Wiley, New York (1968).
17. J. D. Markel and A. H. Gray, *Linear Prediction of Speech,* Springer-Verlag, New York (1976).
18. J. Makhoul, "Linear Prediction: A Tutorial Review," *Proc. IEEE* 63(4), pp.561-580 (April 1975).
19. Jae S. Lim and A. V. Oppenheim, "All-Pole Modeling of Degraded Speech," *IEEE Trans. Acoust. Speech, Signal Proc.* ASSP-26(3), pp.197-210 (June 1978).

20. Jae S. Lim, *Enhancement and Bandwidth Compression of Noisy Speech by Estimation of Speech and its Model Parameters,* Sc.D. Thesis, Dept. of Elec. Eng. and Comp. Sci. M.I.T., Cambridge, Mass. (Aug 1978).

21. Yaakov Bar-Shalom, "Optimal Simultaneous State Estimation and Parameter Identification in Linear Discrete-Time Systems," *IEEE Trans. on Auto. Control* AC-17(3), pp.308-319 (June 1972).

22. David G. Luenberger, *Optimization By Vector Space Methods,* John Wiley & Sons Inc., New York (1969).

23. J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables,* Academic Press, New York (1970).

24. A.A. Goldstein, *Constructive Real Analysis,* Harper and Row, New York (1967).

25. M.M. Vainberg, *Variational Methods for the Study of Nonlinear Operators*, 1964.

26. Vladimir F. Demyanov and Aleksandr M. Rubinov, *Approximate Methods in Optimization Problems,* Amer. Elsevier Publ. Co., New York (1970).

27. Magnus R. Hestenes, *Optimization Theory,* John Wiley & Sons, New York (1975).

28. András Prékopa, "On Logarithmic Concave Measures and Functions." *(Szeged) Acta Sci. Math.* **34**, pp.335-343 (1973).

29. András Prékopa, "Logarithmic Concave Measures With Application to Stochastic Programming," *(Szeged) Acta Sci. Math.* 32, pp.301-316 (1971).

Chapter 3

Iterative Estimation Methods

1. Introduction

The drawback of all the MCEM, MAP, and ML algorithms presented in the last chapter is that, even for relatively simple system models, they all require solving a complicated nonlinear optimization problem. Of course, brute force can always be used to solve these problems, evaluating the objective function on a coarse grid to roughly locate the global optimum, and then applying a "scoring method" (chapter 4 of [1] or [2,3]) or Newton-Raphson or some other gradient hill-climbing algorithm [4,5,6]. In general, however, such methods are complex and computationally time consuming. In this chapter we will focus on a group of iterative methods for solving these problems which carefully exploit the structure of the stochastic system in order to simplify the calculation. We start with a straightforward iterative procedure for solving MCEM; iterative algorithms for solving PARMAP, SIGMAP and PSMAP are then derived by interpreting them as degenerate forms of MCEM. All these iterative algorithms effectively decouple the uncertainty in the various unknowns, thus reducing the estimation problem to a sequence of nearly "classical" estimation problems involving only a single unknown. Each iteration improves the appropriate objective function, thus improving the estimates, and convergence of the algorithms to a stationary point can be proven under mild conditions.

Once again, rather than treat the most general system model, we will restrict our attention to a model with a single signal $\underline{x}$ and a single (Fisher or Bayesian) set of parameters $\underline{\phi}$. The key idea exploited throughout is that the log likelihood function

$\log p(x,\phi)$ is often a relatively "benevolent" function in x and ϕ, and is much more easily evaluated and manipulated than either $\log p(X,\phi)$ or $\log p(x,\Phi)$, since no multidimensional integration is required. It is this reduction of the MAP and MCEM problems to a new form, which only involves the function $\log p(x,\phi)$, which permits a radical restructuring of the estimation problem.

2. Minimum Cross-Entropy Method (MCEM)

The Minimum Cross-Entropy Method fits a separable probability density $q_X(x)q_\Phi(\phi)$ to the actual probability density $p(x,\phi)$ by minimizing the cross-entropy function:

$$\hat{q}_X(),\, \hat{q}_\Phi() = \min_{q_x, q_\phi} H(q_X, q_\Phi) \tag{3.2.1}$$

$$\text{where: } H(q_X, q_\Phi) = \int\limits_X \int\limits_\Phi q_X(x) q_\Phi(\phi) \log \frac{q_X(x) q_\Phi(\phi)}{p(x,\phi)}\, dx\, d\phi$$

subject to the constraint that $q_X(x)$ and $q_\Phi(\phi)$ are nonnegative and integrate to 1 over X and Φ respectively:

$$\int\limits_X q_X(x)\, dx = 1 \quad \text{and} \quad q_X(x) \geq 0 \quad \text{for all } x \in X \tag{3.2.2}$$

$$\int\limits_\Phi q_\Phi(\phi)\, d\phi = 1 \quad \text{and} \quad q_\Phi(\phi) \geq 0 \quad \text{for all } \phi \in \Phi$$

Solving this problem directly is quite difficult, since the unknowns are functions and the minimization is thus performed in an infinite dimensional space. Fortunately, an iterative algorithm for minimizing this cross-entropy is quite simple to derive. We shall minimize $H(q_X, q_\Phi)$ first with respect to the function $q_X()$, then with respect to $q_\Phi()$, iterating back and forth until the estimates converge:

For $k = 0, 1, \cdots$

$$\hat{q}_{X_{k+1}}() = \min_{q_X} H(q_X, \hat{q}_{\Phi_k}) \tag{3.2.3}$$

$$\hat{q}_{\Phi_{k+1}}() = \min_{q_\Phi} H(\hat{q}_{X_{k+1}}, q_\Phi)$$

This "coordinate descent" method will not be as fast as more sophisticated gradient or Quasi-Newton methods, but it has the advantage of simplicity. To minimize with respect to q_X, first rewrite the cross-entropy (3.2.1) in the following form:

$$\hat{q}_{X_{k+1}}(x) = \min_{q_X} \int_X q_X(x) \log \frac{q_X(x)}{\nu_{k+1}(x)} \, dx \tag{3.2.4}$$

$$\text{where: } \log \nu_{k+1}(x) = \int_\Phi \hat{q}_{\Phi_k}(\phi) \log \frac{p(x,\phi)}{\hat{q}_{\Phi_k}(\phi)} \, d\phi$$

By theorem 2.4.1, this function is strictly convex in q_X and achieves its unique minimum at:

$$\hat{q}_{X_{k+1}}(x) = \frac{1}{c_{X_{k+1}}} \nu_{k+1}(x) \tag{3.2.5}$$

$$\text{where: } c_{X_{k+1}} = \int_X \nu_{k+1}(x) \, dx$$

The value of the cross-entropy at this estimate is:

$$H(\hat{q}_{X_k}, \hat{q}_{\Phi_k}) = -\log c_{X_{k+1}} \tag{3.2.6}$$

Similarly, $H(\hat{q}_{X_{k+1}}, q_\Phi)$ can be shown to be strictly convex in q_Φ, and has the unique minimizer:

$$\hat{q}_{\Phi_{k+1}}(\phi) = \frac{1}{c_{\Phi_{k+1}}} \xi_{k+1}(\phi) \tag{3.2.7}$$

$$\text{where: } \log \xi_{k+1}(\phi) = \int_X \hat{q}_{X_{k+1}}(x) \log \frac{p(x,\phi)}{\hat{q}_{X_{k+1}}(x)} \, dx$$

$$c_{\Phi_{k+1}} = \int_\Phi \xi_{k+1}(\phi) \, d\phi$$

The value of the cross-entropy at this estimate is:

$$H(\hat{q}_{X_{k+1}},\hat{q}_{\Phi_{k+1}}) = -\log c_{\Phi_{k+1}} \tag{3.2.8}$$

If the iteration has not yet converged, each step strictly reduces the value of $H(\hat{q}_{X_k},\hat{q}_{\Phi_k})$ and thus improves the estimates of the separable probability density. Furthermore, theorem 2.4.2 guarantees that the cross-entropy is bounded below:

$$H(q_X,q_\Phi) \geq Q_x(X)Q_\Phi(\Phi)\log\frac{Q_x(X)Q_\Phi(\Phi)}{P(X,\Phi)} = -\log P(X,\Phi) \tag{3.2.9}$$

Thus the cross-entropy of the estimates must converge monotonically downward to a lower limit:

$$H(\hat{q}_{X_k},\hat{q}_{\Phi_k}) \to H_* \quad \text{as } k \to \infty \tag{3.2.10}$$

Equations (3.2.6) and (3.2.8) also imply that the normalization constants c_{X_k} and c_{Φ_k} must converge monotonically upward to limits:

$$c_{X_k} \to e^{-H_*} \qquad c_{\Phi_k} \to e^{-H_*} \qquad \text{as } k \to \infty \tag{3.2.11}$$

Unfortunately, proving convergence of the estimated densities themselves is more difficult, and we will postpone this analysis until section 9.5.

Recognizing the integrals in (3.2.4) and (3.2.7) as conditional expectations of $\log\frac{p(x,\phi)}{\hat{q}_{\Phi_k}(\phi)}$ and $\log\frac{p(x,\phi)}{\hat{q}_{X_k}(x)}$ with respect to the densities $\hat{q}_{\Phi_k}(\phi)$ and $\hat{q}_{X_{k+1}}(x)$ respectively, allows us to write the algorithm in the following simplified form:

MCEM Iterative Algorithm:

For $k=0,1,\ldots$

$$\log \hat{q}_{\mathcal{X}_{k+1}}(x) = E_{\Phi}\left[\log \frac{p(x,\phi)}{\hat{q}_{\Phi_k}(\phi)} \,\middle|\, \hat{q}_{\Phi_k}\right] + \text{constant} \tag{3.2.12}$$

$$\log \hat{q}_{\Phi_{k+1}}(\phi) = E_{\mathcal{X}}\left[\log \frac{p(x,\phi)}{\hat{q}_{\mathcal{X}_{k+1}}(x)} \,\middle|\, \hat{q}_{\mathcal{X}_{k+1}}\right] + \text{constant}$$

where appropriate normalization constants have to be added. Intuitively, the algorithm can be explained as an attempt to compensate for the uncertainty in both the signal and parameters in calculating the estimates. If $\hat{q}_{\mathcal{X}_k}(x)\hat{q}_{\Phi_k}(\phi)$ is a good approximation to $p(x,\phi)$, then $\log \frac{p(x,\phi)}{\hat{q}_{\Phi_k}(\phi)}$ ought to be approximately independent of ϕ. Averaging this function over all parameter values ϕ leaves only the x dependency of $\log p(x,\phi)$, which is used as the new estimate of the log signal density, $\log \hat{q}_{\mathcal{X}_{k+1}}(x)$. The parameter density is then reestimated by averaging the function $\log \frac{p(x,\phi)}{\hat{q}_{\mathcal{X}_{k+1}}(x)}$ over all signal values, thus recovering the ϕ dependency of $\log p(x,\phi)$. Because the "correct" signal and parameter densities are unknown, this averaging process is imperfect. Thus the algorithm iterates, using the improved density estimates to improve the averaging on the next pass, and thus further improve the next density estimates.

Note that the behavior of the expectation of $\log \frac{p(x,\phi)}{q_{\Phi}(\phi)}$ as a function of x, or the behavior of $\log \frac{p(x,\phi)}{q_{\mathcal{X}}(x)}$ as a function of ϕ, is just the same as the behavior of the expectation of $\log p(x,\phi)$. The only reason, therefore, that we divide $p(x,\phi)$ by $q_{\Phi}(\phi)$ or by $q_{\mathcal{X}}(x)$ in (3.2.12) is to ensure that the expectation will be finite.

If point estimates of x and ϕ are needed, they can be calculated by running the MCEM algorithm to convergence, giving estimates $\hat{q}_{\mathcal{X}}(x)$ and $\hat{q}_{\Phi}(\phi)$. and then calculating the conditional expectation or MAP estimates of x and ϕ:

$$\hat{x} = E_{\mathcal{X}}\left[x \,\middle|\, \hat{q}_{\mathcal{X}}\right] \qquad \hat{\phi} = E_{\Phi}\left[\phi \,\middle|\, \hat{q}_{\Phi}\right]$$

or:

$$\hat{x} - \max_{x \in \mathcal{X}} \hat{q}_{\mathcal{X}}(x) \qquad \hat{\phi} - \max_{\phi \in \Phi} \hat{q}_{\Phi}(\phi) \tag{3.2.13}$$

The most important application of this algorithm, discussed in section 7, is when the model density $p(x,\phi)$ forms an exponential family of densities. In this case, we will see that $\hat{q}_{\mathcal{X}_t}(x)$ and $\hat{q}_{\Phi_t}(\phi)$.will also be exponential densities, and the MCEM algorithm reduces to iteratively evaluating the conditional expectation of a few functions, each involving only a single unknown.

3. MAP Optimal Parameter Estimation (PARMAP)

The PARMAP approach estimates the parameters by choosing their most likely value given that $x \in \mathcal{X}$:

$$\phi_{MAP} - \max_{\phi \in \Phi} \log p(\mathcal{X},\phi) = \max_{\phi \in \Phi} \log \int_{\mathcal{X}} p(x,\phi)\, dx \tag{3.3.1}$$

The difficulty with directly maximizing this function is that to compute $p(\mathcal{X},\phi)$ we must evaluate a multidimensional integral, and the resulting nonlinear function can be quite complicated. We therefore propose an indirect iterative method for solving (3.3.1) which reduces the computation to a form similar to that of MCEM. A previous derivation of a similar algorithm was given by Musicus [7] before the connection between PARMAP and MCEM was understood. The derivation we present here exploits the interpretation of PARMAP as a degenerate form of MCEM, as described in chapter 2, section 7.5. We start by minimizing the cross entropy function H with the additional constraint that $q_\Phi(\phi) = \delta(\phi-\hat{\phi})$ for some $\hat{\phi}\in\Phi$. As before, we retain only the first two terms of the cross entropy expression (2.7.17). Abbreviating $\tilde{H}(q_X,\hat{\phi})=\tilde{H}(q_X,q_\Phi)$ where $q_\Phi(\phi)=\delta(\phi-\hat{\phi})$, we find that:

$$\tilde{H}(q_X,\hat{\phi}) = -\int_X\int_\Phi q_X(x)\delta(\phi-\hat{\phi}) \log p(x,\phi)\, dx d\phi + \int_X q_X(x) \log q_X(x)\, dx$$
$$= \int_X q_X(x) \log \frac{q_X(x)}{p(x,\hat{\phi})}\, dx \tag{3.3.2}$$

Let us iteratively minimize this modified cross entropy expression by minimizing first with respect to q_X, then with respect to the location $\hat{\phi}$ of the impulse function q_Φ, iterating back and forth until the estimates converge.

For $k=0,1,\cdots$

$$\begin{aligned} \hat{q}_{X_{k+1}}(x) &- \min_{q_X} \tilde{H}(q_X,\hat{\phi}_k) \\ \hat{\phi}_{k+1} &- \min_{\phi\in\Phi} \tilde{H}(\hat{q}_{X_{k+1}},\phi) \end{aligned} \tag{3.3.3}$$

Minimizing with respect to q_X is easy; by theorem 2.4.1, expression (3.3.2) is strictly convex in q_X and achieves its unique minimum at the estimate:

$$\hat{q}_{X_{k+1}}(x) = \frac{p(x,\hat{\phi}_k)}{\int_X p(x,\hat{\phi}_k)\, dx} = p_{X|\Phi}(x|\hat{\phi}_k) \tag{3.3.4}$$

The signal density estimate is thus simply the conditional probability density of x given the parameter value $\hat{\phi}_k$. Now minimizing $\tilde{H}(\hat{q}_{X_{k+1}},\phi)$ over ϕ yields:

$$\hat{\phi}_{k+1} - \max_{\phi\in\Phi} \int_X p_{X|\Phi}(x|\hat{\phi}_k) \log \frac{p(x,\phi)}{p_{X|\Phi}(x|\hat{\phi}_k)}\, dx \tag{3.3.5}$$

Viewing $\log \frac{p(x,\phi)}{p_{X|\Phi}(x|\hat{\phi}_k)}$ as a function of x, using the relation (3.3.4), and discarding a constant term $\log p(X,\hat{\phi}_k)$ for convenience, the algorithm can written in the simplified form:

```
PARMAP iterative algorithm:
Guess $\hat{\phi}_0$
For $k=0,1,\cdots$
        $\hat{\phi}_{k+1} = \max_{\phi\in\Phi} E_{X|\Phi}\left[ \log \frac{p(x,\phi)}{p(x,\hat{\phi}_k)} \middle| \hat{\phi}_k \right]$        (3.3.6)
```

Clearly each iteration decreases the cross-entropy $\bar{H}(\hat{q}_{X_{k+1}},\hat{\phi}_k)$. Furthermore, by substitution:

$$\bar{H}(\hat{q}_{X_{k+1}},\hat{\phi}_k) = -\log p(X,\hat{\phi}_k) \tag{3.3.7}$$

and thus by construction of the algorithm:

$$\log p(X,\hat{\phi}_{k+1}) = -\bar{H}(\hat{q}_{X_{k+2}},\hat{\phi}_{k+1}) \geq -\bar{H}(\hat{q}_{X_{k+1}},\hat{\phi}_k) = \log p(X,\hat{\phi}_k) \tag{3.3.8}$$

Each iteration therefore also increases the likelihood function $p(X,\hat{\phi}_k)$ and thus yields a better PARMAP parameter estimate.

In the special case of noisy all-pole models, this algorithm is very similar to the RLMAP algorithm of Lim and Oppenheim [8]. For discrete multinomial densities and grouped data, it is also the same algorithm suggested by Hartley [9].

Intuitively, the algorithm can be explained as follows. The quantity inside the expectation, $\log \frac{p(x,\phi)}{p(x,\hat{\phi}_k)}$, represents the log likelihood of the pair of values x,ϕ versus the pair of values $x,\hat{\phi}_k$. If the actual signal value x were known, then the MAP parameter estimate could be calculated by maximizing this log likelihood function with respect to ϕ. However, since the signal value is not known exactly, but can only be inferred via the incomplete observation that $x\in X$, this algorithm instead chooses to maximize the value of the function $\log \frac{p(x,\phi)}{p(x,\hat{\phi}_k)}$ averaged over all possible signal values

$x \in X$. Because the actual parameter values are unknown, this signal averaging is imperfect. Thus the algorithm iterates, using each new parameter estimate to improve the signal averaging calculation on the next pass, and thus improve the next parameter estimates. Note that, as in the MCEM algorithm, the sole purpose for dividing $p(x,\phi)$ by $p(x,\hat{\phi}_k)$ before calculating the expectation is to ensure that this expectation is finite.

A similar algorithm can be developed for the Fisher system model in which the parameters are viewed as non-random variables. We start by trying to solve the PARML parameter estimation problem:

$$\hat{\phi}_{MAP} = \max_{\phi \in \Phi} \log p(X \mid \phi) \qquad (3.3.9)$$

The development of an iterative algorithm for solving this is then identical to the one above, except that probability densities of the form $\log p(x,\phi)$ and $\log p(X,\phi)$ must be replaced by $\log p(x \mid \phi)$ and $\log p(X \mid \phi)$ respectively. The resulting iterative Fisher estimation algorithm will then calculate:

PARML iterative algorithm:
Guess $\hat{\phi}_0$
For $k = 0, 1, \cdots$

$$\hat{\phi}_{k+1} = \max_{\phi \in \Phi} E_{X \mid \Phi}\left[\log \frac{p(x \mid \phi)}{p(x \mid \hat{\phi}_k)} \;\middle|\; \hat{\phi}_k \right] \qquad (3.3.10)$$

If an estimate of the signal x is desired in addition to the parameters, PARMAP can be iterated to convergence, $\hat{\phi}_k \to \hat{\phi}_{MAP}$, and then we could calculate the expected value or MAP estimate of $q_X(x) = p_{X \mid \Phi}(x \mid \hat{\phi}_{MAP})$:

$$\hat{x} = \int_X x \, p_{X \mid \Phi}(x \mid \hat{\phi}_{MAP}) \, dx$$

or $\qquad (3.3.11)$

$$\hat{x} = \max_{x \in X} \log p(x \mid \hat{\phi}_{MAP})$$

The advantage of this algorithm is that it reduces the complicated nonlinear maximization of $p(X,\underline{\phi})$ to an iterative maximization of the expectation of $\log \frac{p(\underline{x},\underline{\phi})}{p(\underline{x},\hat{\underline{\phi}}_k)}$. As discussed in section 7, in the very important special case when $p(\underline{x},\underline{\phi})$ is an exponential density, this expectation operator changes the details but not the overall difficulty of maximizing $\log p(\underline{x},\underline{\phi})$. For these models, therefore, the maximization in (3.3.6) requires about the same amount of computation as the maximization in the classical parameter identification problem (2.6.8). If the latter is "easy" to solve, then the PARMAP algorithm will also be "easy" to solve.

4. MAP Optimal Signal Estimation (SIGMAP)

The SIGMAP estimation algorithm tries to estimate the signal $\underline{x}$ by calculating its most likely value given the known information:

$$\hat{\underline{x}}_{MAP} - \max_{\underline{x} \in X} \log p(\underline{x},\Phi) = \log \int_{\Phi} p(\underline{x},\underline{\phi})\, d\underline{\phi} \tag{3.4.1}$$

The difficulty with this approach, like PARMAP, is that computing $p(\underline{x},\Phi)$ requires evaluating a complicated multidimensional integral, and the resulting nonlinear function can be quite complicated. From a formal standpoint, this estimation problem is identical to the PARMAP problem in (3.3.1), except with the roles of $\underline{x}$ and $\underline{\phi}$ reversed. Exactly the same iterative algorithm used for solving PARMAP can therefore be applied to this problem, provided that we reverse the roles of $\underline{x}$ and $\underline{\phi}$. The SIGMAP algorithm thus generates signal and parameter density estimates:

$$\begin{aligned} \hat{q}_{X_k}(\underline{x}) &= \delta(\underline{x} - \hat{\underline{x}}_k) \\ \hat{q}_{\Phi_k}(\underline{\phi}) &= p_{\Phi|X}(\underline{\phi}\,|\,\hat{\underline{x}}_k) \end{aligned} \tag{3.4.2}$$

where the signal estimates $\hat{\underline{x}}_k$ are iteratively generated by:

```
SIGMAP iterative algorithm:
Guess $\hat{x}_0$
For $k=0,1,\cdots$                                                    (3.4.3)
    $\hat{x}_{k+1} = \max_{x \in X} E_{\Phi|X}\left[\log \frac{p(x,\phi)}{p(\hat{x}_k,\phi)} \middle| \hat{x}_k\right]$
```

Each iteration of this algorithm not only increases the modified cross-entropy $\tilde{H}(\hat{x}_k, \hat{q}_{\Phi_k})$, but also increases the likelihood function $\log p(\hat{x}_k, \Phi)$ and thus "improves" the signal estimates.

This algorithm can be interpreted in exactly the same way we interpreted PARMAP. If the actual parameter value ϕ_* were known, then the MAP signal estimate could be calculated by maximizing the log likelihood function $\log p(x, \phi_*)$ with respect to x. However, since the parameter value is not known exactly, but can only be inferred via the observation that $\phi \in \Phi$, this algorithm instead chooses to maximize the value of the likelihood ratio $\log \frac{p(x,\phi)}{p(\hat{x}_k,\phi)}$ averaged over all possible parameter values $\phi \in \Phi$. Because the actual signal values are unknown, this parameter averaging is imperfect. Thus the algorithm iterates, using each new signal estimate to improve the parameter averaging calculation on the next pass, and thus improve the next signal estimate. Note that the x dependency of the average of $\log \frac{p(x,\phi)}{p(\hat{x}_k,\phi)}$ is the same as the x dependency of $\log p(x,\phi)$; the only reason for dividing by $p(\hat{x}_k, \phi)$ is to ensure that the expectation is finite.

As discussed in chapter 2, the SIGMAP procedure can not be used if the parameters are non-random (Fisher), unless we are willing to create a fictitious *a priori* density for $p(\phi)$.

If an estimate of the parameters is needed, the above algorithm can be iterated to convergence, $\hat{x}_k \to \hat{x}_{MAP}$, and then we can calculate the mean or mode of the parameter density estimate $\hat{q}_{\Phi}(\phi) = p_{\Phi|X}(\phi \mid \hat{x}_{MAP})$:

$$\hat{\phi} = E_{\Phi|X}\left[\phi \,\middle|\, \hat{x}_{MAP}\right]$$

or (3.4.4)

$$\hat{\phi} \leftarrow \max_{\phi \in \Phi} \log p(\hat{x}_{MAP}, \phi)$$

The advantage of this iterative scheme is that we reduce the "difficult" problem in (3.4.1) to one involving the conditional expectation of the "simpler" log likelihood function $\log p(x, \phi)$. As will be seen in section 7, for signal models in which $p(x, \phi)$ is an exponential density this expectation operator does not significantly change the form of the expression being maximized. In this case, if the "classical" filtering problem is "easy" to solve, then each pass of this SIGMAP algorithm will also be "easy" to solve.

5. MAP Simultaneous Parameter and Signal Estimation (PSMAP)

The simplest of the MAP problems to solve is PSMAP, in which we choose the combination of signal and parameter values which are most likely given the known information:

$$\hat{x}, \hat{\phi} \leftarrow \max_{x \in X, \phi \in \Phi} \log p(x, \phi) \qquad (3.5.1)$$

As suggested by the interpretation of PSMAP as a degenerate form of MCEM, let us estimate impulse signal and parameter densities $\hat{q}_X(x) = \delta(x - \hat{x})$ and $\hat{q}_{\Phi}(\phi) = \delta(\phi - \hat{\phi})$ by minimizing the first term of the cross-entropy expression in (2.7.17):

$$\bar{H}(\hat{x}, \hat{\phi}) = -\int_X \int_\Phi \delta(x - \hat{x})\delta(\phi - \hat{\phi}) \log p(x, \phi)\, dx\, d\phi$$

$$= -\log p(\hat{x}, \hat{\phi}) \qquad (3.5.2)$$

Iteratively minimizing this modified cross-entropy expression with respect to q_X and q_Φ is thus equivalent to simply maximizing the log likelihood function $\log p(x,\phi)$ with respect to each argument in turn:

```
PSMAP iterative algorithm:
Guess $\hat{\phi}_0$
For $k=0,1,\cdots$
    $\hat{x}_{k+1} \leftarrow \max_{x \in X} \log p(x, \hat{\phi}_k)$                (3.5.3)
    $\hat{\phi}_{k+1} \leftarrow \max_{\phi \in \Phi} \log p(\hat{x}_{k+1}, \phi)$
```

Each iteration decreases the modified cross entropy $\hat{H}$, and increases the log likelihood $\log p(\hat{x}_k, \hat{\phi}_k)$, thus "improving" the signal and parameter estimates. A Maximum Likelihood version of this algorithm for solving the PSML problem in (2.8.2) looks identical to this, except with the density $p(x,\phi)$ replaced by $p(x \mid \phi)$.

The similarity between this algorithm and the "classical" algorithms in (2.6.3) and (2.6.8) is striking. In the PSMAP algorithm, each unknown is estimated as if the other unknowns were equal to their latest estimated values. The algorithm then iterates to improve the estimates. The PSMAP algorithm therefore requires exactly the same computation on each pass as the "classical" estimation algorithms. If the latter is "easy" to solve, then PSMAP will be just as easy.

6. Comparison of the Algorithms

The chief merit of all four algorithms presented above is that they reduce the computation involved in estimating the unknown signal and parameters to a form similar to the classical estimation case, in which only one of the unknowns must be estimated at a time. Table 3.1 summarizes the Bayesian version of the four iterative algorithms we

Comparison of the Four Bayesian Iterative Algorithms

	Signal Density Estimates	Parameter Density Estimates
CLASSICAL: x_* or ϕ_* known	$\log \hat{q}_X(x) = \log p(x, \phi_*) + c_x$	$\log \hat{q}_\Phi(\phi) = \log p(\phi, x_*) + c_\phi$
MCEM:	$\log \hat{q}_{X_{k+1}}(x) = E_\Phi \left[\log \frac{p(x,\phi)}{\hat{q}_{\Phi_k}(\phi)} \middle\vert \hat{q}_{\Phi_k} \right] + c_x$	$\log \hat{q}_{\Phi_{k+1}}(\phi) = E_X \left[\log \frac{p(x,\phi)}{\hat{q}_{X_{k+1}}(x)} \middle\vert \hat{q}_{X_{k+1}} \right] + c_\phi$
PARMAP: $\hat{q}_\Phi = \delta(\phi - \hat{\phi})$	$\log \hat{q}_{X_{k+1}}(x) = \log p(x, \hat{\phi}_k) + c_x$	$\hat{\phi}_{k+1} \leftarrow \max\limits_{\phi \in \Phi} E_X \left[\log \frac{p(x,\phi)}{\hat{q}_{X_{k+1}}(x)} \middle\vert \hat{q}_{X_{k+1}} \right]$
SIGMAP: $\hat{q}_X = \delta(x - \hat{x})$	$\hat{x}_{k+1} \leftarrow \max\limits_{x \in X} E_\Phi \left[\log \frac{p(x,\phi)}{\hat{q}_{\Phi_k}(\phi)} \middle\vert \hat{q}_{\Phi_k} \right]$	$\log \hat{q}_{\Phi_{k+1}}(\phi) = \log p(\hat{x}_{k+1}, \phi) + c_\phi$
PSMAP: $\hat{q}_\Phi = \delta(\phi - \hat{\phi})$ $\hat{q}_X = \delta(x - \hat{x})$	$\hat{x}_{k+1} \leftarrow \max\limits_{x \in X} \log p(x, \hat{\phi}_k)$	$\hat{\phi}_{k+1} \leftarrow \max\limits_{\phi \in \Phi} \log p(\hat{x}_{k+1}, \phi)$

have discussed, in a convenient form for comparison.

Note the similarity in structure of all four algorithms. The Minimum Cross-Entropy Method treats the signal and parameter unknowns symmetrically, and tries to adjust its density estimates to match the given model density over the entire domain $X \times \Phi$. This symmetry and the attention to the tails of the densities as well as the peaks is similar to the "optimal" MMSE method. PARMAP is asymmetric, integrating over all signals in X, but then maximizing over the parameters. SIGMAP is asymmetric in the opposite way, integrating over the entire parameter space but maximizing over the signal space. PSMAP treats the signal and parameters symmetrically, but it completely ignores the shape of the model density, looking only for the peak. The cross-entropy interpretations of the MAP methods and the resulting iterative algorithms reflect these inherent properties of the estimation methods. MCEM alternates between averaging over the signal and averaging over the parameters, treating both equally and using the tails of the densities to improve its estimates. P RMAP iteratively averages over the signal, but maximizes over the parameter space; this is a direct result of using an impulse function to model the parameter density. SIGMAP does the opposite, iteratively averaging over the parameters and maximizing over the signal. PSMAP treats both unknowns symmetrically, maximizing over the signal and then over the parameters, and completely ignoring any shape information.

Despite these differences, however, the forms of the four algorithms are quite similar, alternating between a signal estimation step and a parameter estimation step. Every iteration decreases the appropriate cross-entropy function and increases the appropriate likelihood function. In the remainder of this thesis we will apply these four algorithms to a variety of signa! and parameter estimation problems, and in the process will discover numerous properties of these estimation methods. To simplify our conclusions somewhat, we will usually find that MCEM is computationally the most difficult, but also generally comes closest to the MMSE estimates. PSMAP is the simplest, since it does no averaging, but it generally gives the worst estimates. For stable and stationary models in which we are estimating a set of structural parameters $\underline{\phi}$ and a signal $\underline{x}$ from noisy observations, we will generally find that MCEM and PARMAP give asymptotically identical results as the observation interval $N \to \infty$. Both will give asymptotically consistent and efficient parameter estimates, and their signal estimates will be asymptotically identical to the classical signal estimates generated using the correct parameter values. SIGMAP and PSMAP, on the other hand, will give asymptotically identical, but heavily biased parameter estimates.

7. Exponential Family of Densities

A very important class of problems for which all our iterative algorithms take a particularly elegant form, is the case when the model density $\mathrm{p}(\underline{x},\underline{\phi})$ forms an exponential family of densities:

$$\mathrm{p}(\underline{x},\underline{\phi}) = h(\underline{\phi})g(\underline{x}) \exp\left[\sum_{i=1}^{r} \pi_i(\underline{\phi})t_i(\underline{x}) \right] \qquad (3.7.1)$$

Examples of densities which can be put into this form include binomial, negative binomial, multinomial, Poisson, Normal, Gamma and Beta densities. Distributions which do not fit this form include Cauchy distributions, or any problem in which the space of feasible signal values depends on the parameters. The exponential class of densities has been very carefully studied because of its close connection to the existence of sufficient statistics (for an extensive discussion of these densities, c.f. Lehman [10], or Ferguson [11].) Suppose we are given a sample of independently generated signal values

$\underline{x}_1, \ldots, \underline{x}_n$, each with probability $p(\underline{x}_i \mid \phi)$. If this density has the form (3.7.1), then:

$$T = \left(T_1, \ldots, T_r\right) = \left(\sum_{j=1}^{n} t_1(\underline{x}_j), \ldots, \sum_{j=1}^{n} t_r(\underline{x}_j)\right) \tag{3.7.2}$$

is a sufficient statistic for estimating the parameters, as may be seen from the following factorization:

$$\begin{aligned} p(\underline{x}_1, \ldots, \underline{x}_n \mid \phi) &= \prod_{j=1}^{n} p(\underline{x}_j \mid \phi) \\ &= \left(\frac{h(\phi)}{p(\phi)}\right)^n \left(\prod_{j=1}^{n} g(\underline{x}_j)\right) \exp\left[\sum_{i=1}^{r} \pi_i(\phi) \left(\sum_{j=1}^{n} t_i(\underline{x}_j)\right)\right] \end{aligned} \tag{3.7.3}$$

Thus the relative likelihood of one parameter value, ϕ_1, versus another, ϕ_2, given this observation sample, is solely a function of T:

$$\log \frac{p(\underline{x}_1, \ldots, \underline{x}_n \mid \phi_1)}{p(\underline{x}_1, \ldots, \underline{x}_n \mid \phi_2)} = \sum_{i=1}^{r} \left(\pi_i(\phi_1) - \pi_i(\phi_2)\right) T_i \tag{3.7.4}$$

In effect, the values $T_1, \ldots, T_r$ summarize all the relevant information in the sample $\underline{x}_1, \ldots, \underline{x}_n$ for estimating the parameters. Most importantly, a converse of this result also holds. If a density $p(\underline{x} \mid \phi)$ has the property that there exists a sufficient statistic $T = (T_1, \ldots, T_r)$ of fixed dimension r, whatever the size of the sample drawn from the distribution, and if the set on which $p(\underline{x} \mid \phi)$ is zero does not depend on ϕ, and if certain mild regularity conditions are satisfied, then the distribution forms an exponential family [12]. A variety of other interesting properties of exponential densities can be derived, but these are all we need for the moment.

7.1. Estimation With an Exponential Class of Densities

Note that even if $p(\underline{x}, \phi)$ forms an exponential family of densities, calculating the PARMAP or SIGMAP marginal densities $p(\underline{X}, \phi)$ or $p(\underline{x}, \Phi)$ can be quite difficult. Surprisingly, however, all four of our iterative algorithms take a remarkably elegant form when $p(\underline{x},\underline{\phi})$ is an exponential density. For example, it is easy to show by substitution that the iterative MCEM algorithm generates density estimates of the form:

$$\hat{q}_{X_{k+1}}(\underline{x}) = \frac{1}{c_{x_{k+1}}} g(\underline{x}) \exp\left[\sum_{i=1}^{r} \overline{\pi_i(\underline{\phi})} t_i(\underline{x}) \right] \tag{3.7.5}$$

$$\text{where: } \overline{\pi_i(\underline{\phi})} = E_{\Phi}\left[\pi_i(\underline{\phi}) \,\middle|\, \hat{q}_{\Phi_k} \right] = \int_{\Phi} \pi_i(\underline{\phi}) \hat{q}_{\Phi_k}(\underline{\phi})\, d\underline{\phi}$$

and:

$$\hat{q}_{\Phi_{k+1}}(\underline{\phi}) = \frac{1}{c_{\phi_{k+1}}} h(\underline{\phi}) \exp\left[\sum_{i=1}^{r} \pi_i(\underline{\phi}) \overline{t_i(\underline{x})} \right] \tag{3.7.6}$$

$$\text{where: } \overline{t_i(\underline{x})} = E_{X}\left[t_i(\underline{x}) \,\middle|\, \hat{q}_{X_{k+1}} \right] = \int_{X} t_i(\underline{x}) \hat{q}_{X_{k+1}}(\underline{x})\, d\underline{x}$$

and where $c_{x_{k+1}}$ and $c_{\phi_{k+1}}$ are normalization constants. The signal and parameter density estimates $\hat{q}_{X_k}$, $\hat{q}_{\Phi_k}$ are thus also exponential densities of a form similar to $p(\underline{x},\underline{\phi})$, but with the functions $\pi_i(\underline{\phi})$ and $t_i(\underline{x})$ respectively replaced by their conditional expectations. The MCEM algorithm thus simply alternates between calculating the r conditional expectations of the $\pi_i(\underline{\phi})$ functions, and calculating the r conditional expectations of the $t_i(\underline{x})$ functions, iterating back and forth until these estimated values and the corresponding densities converge.

PARMAP, SIGMAP, and PSMAP closely resemble MCEM, except that one or both of the estimated densities are constrained to be impulse functions. The derivation of these algorithms is thus quite similar to that of MCEM, and so we simply summarize the results in table 3.2. The chief difference between these algorithms and MCEM is that one or both sets of conditional expectation calculations are replaced by a simpler maximization step. PARMAP, for example, starts by forming the signal density estimate using values of $\pi_i(\hat{\underline{\phi}}_k)$. (MCEM would have used the conditional expectation

Comparison of Iterative Algorithms for Exponential Densities

$$p(x,\Phi) = h(\Phi)g(x)\exp\left[\sum_{l=1}^{r} \pi_l(\Phi)t_l(x)\right]$$

	Signal Density Estimate	Parameter Density Estimate
CLASSICAL:	$\hat{q}_X(x) = \frac{1}{c_x} g(x)\exp\left[\sum_{l=1}^{r} \pi_l(\Phi\bullet)t_l(x)\right]$	$\hat{q}_\Phi(\Phi) = \frac{1}{c_\Phi} h(\Phi)\exp\left[\sum_{l=1}^{r} \pi_l(\Phi)t_l(x\bullet)\right]$
MCEM:	$\hat{q}_{X_{k+1}}(x) = \frac{1}{c_{x_{k+1}}} g(x)\exp\left[\sum_{l=1}^{r} \overline{\pi_l(\Phi)}t_l(x)\right]$	$\hat{q}_{\Phi_{k+1}}(\Phi) = \frac{1}{c_{\Phi_{k+1}}} h(\Phi)\exp\left[\sum_{l=1}^{r} \pi_l(\Phi)\overline{t_l(x)}\right]$
PARMAP:	$\hat{q}_{X_{k+1}}(x) = \frac{1}{c_{x_{k+1}}} g(x)\exp\left[\sum_{l=1}^{r} \pi_l(\hat{\Phi}_k)t_l(x)\right]$	$\hat{\Phi}_{k+1} \leftarrow \max_{\Phi\in\Phi} h(\Phi)\exp\left[\sum_{l=1}^{r} \pi_l(\Phi)\overline{t_l(x)}\right]$
SIGMAP:	$\hat{x}_{k+1} \leftarrow \max_{x\in X} g(x)\exp\left[\sum_{l=1}^{r} \overline{\pi_l(\Phi)}t_l(x)\right]$	$\hat{q}_{\Phi_{k+1}}(\Phi) = \frac{1}{c_{\Phi_{k+1}}} h(\Phi)\exp\left[\sum_{l=1}^{r} \pi_l(\Phi)t_l(\hat{x}_{k+1})\right]$
PSMAP:	$\hat{x}_{k+1} \leftarrow \max_{x\in X} g(x)\exp\left[\sum_{l=1}^{r} \pi_l(\hat{\Phi}_k)t_l(x)\right]$	$\hat{\Phi}_{k+1} \leftarrow \max_{\Phi\in\Phi} h(\Phi)\exp\left[\sum_{l=1}^{r} \pi_l(\Phi)t_l(\hat{x}_{k+1})\right]$
	where: $\overline{\pi_l(\Phi)} = \int_\Phi \pi_l(\Phi)\,\hat{q}_{\Phi_k}(\Phi)\,d\Phi$	where: $\overline{t_l(x)} = \int_X t_l(x)\,\hat{q}_{X_{k+1}}(x)\,dx$

$\overline{\pi_i(\phi)}$.) The next parameter estimate is then calculated by maximizing a function involving the r conditional expectations $\overline{t_i(x)}$. SIGMAP is similar, except that we calculate the r conditional expectations of $\overline{\pi_i(\phi)}$ and then perform a maximization to find $\hat{x}_{k+1}$. PSMAP simply alternates between two maximization steps.

In most of the examples considered in this thesis, the probability densities are not only exponential, but also the terms $\pi_i(\phi)$ and $t_i(x)$ are low order polynomials in the components of ϕ and x. All four algorithms then need only calculate low order moments of ϕ or x, and/or maximize low order polynomials in x or ϕ. This symmetry and computational simplicity is rather remarkable.

7.2. Natural Parameterization of Exponential Densities

Because of the fundamental role of the functions $\pi(\phi)$ and $t(x)$ in the construction of sufficient statistics for exponential families of densities, these functions are considered to be the "natural" parameters for the family. Our algorithms take a particularly elegant form whenever the probability density $p(x,\phi)$ can be transformed into its "natural parameter" form:

$$p(x,\phi) = g(x)h(\phi)\exp\left(\phi^T D x\right) \tag{3.7.7}$$

for some matrix D. Now the density estimates generated by our four algorithms have the simple form:

	$\hat{q}_{X_{k+1}}(x)$	$\hat{q}_{\Phi_{k+1}}(\phi)$
MCEM:	$p_{X\mid\Phi}(x \mid \hat{\phi}_k)$	$p_{\Phi\mid X}(\phi \mid \hat{x}_{k+1})$
PARMAP:	$p_{X\mid\Phi}(x \mid \hat{\phi}_k)$	$\delta(\phi - \hat{\phi}_{k+1})$
SIGMAP:	$\delta(x - \hat{x}_{k+1})$	$p_{\Phi\mid X}(\phi \mid \hat{x}_{k+1})$
PSMAP:	$\delta(x - \hat{x}_{k+1})$	$\delta(\phi - \hat{\phi}_{k+1})$

where these values of $\hat{x}_k$ and $\hat{\phi}_k$ are iteratively calculated by solving:

	Signal Estimate	Output Estimate
MCEM:	$\hat{x}_{k+1} = E_{X\mid\Phi}[x \mid \hat{\phi}_k]$	$\hat{\phi}_{k+1} = E_{\Phi\mid X}[\phi \mid \hat{x}_{k+1}]$
PARMAP:	$\hat{x}_{k+1} = E_{X\mid\Phi}[x \mid \hat{\phi}_k]$	$\hat{\phi}_{k+1} = \max\limits_{\phi\in\Phi} p_{\Phi\mid X}(\phi \mid \hat{x}_{k+1})$
SIGMAP:	$\hat{x}_{k+1} = \max\limits_{x\in X} p_{X\mid\Phi}(x \mid \hat{\phi}_k)$	$\hat{\phi}_{k+1} = E_{\Phi\mid X}[\phi \mid \hat{x}_{k+1}]$
PSMAP:	$\hat{x}_{k+1} = \max\limits_{x\in X} p_{X\mid\Phi}(x \mid \hat{\phi}_k)$	$\hat{\phi}_{k+1} = \max\limits_{\phi\in\Phi} p_{\Phi\mid X}(\phi \mid \hat{x}_{k+1})$

MCEM alternates between calculating the conditional mean of the signal given the parameters, and calculating the conditional mean of the parameters given the signal. PARMAP also uses the mean of the signal, but chooses the mode of the conditional parameter density for its parameter estimate. SIGMAP does the opposite, using the mean of the conditional parameter density and the mode of the conditional signal density. PSMAP uses the modes of both densities. Intuitively, since using the means of densities tends to be a better choice on average than using the peaks, we might expect MCEM to give the best estimates. On the other hand, it is easier to maximize a density than to compute its mean, and thus we would expect the MAP methods to be simpler.

Another feature of this "natural" form is that the MCEM cross-entropy has a simple interpretation:

$$H(\hat{q}_{X_k}, \hat{q}_{\Phi_k}) = \log \frac{p(\hat{x}_k, \hat{\Phi}_k)}{p(X, \hat{\Phi}_k)\, p(\hat{x}_k, \Phi)} \qquad (3.7.8)$$

Thus in this case the cross-entropy takes the form of a combination of the PARMAP, SIGMAP and PSMAP likelihood functions.

8. Extensions to More General Signal Models

As pointed out in chapter 2, when the signal model has several signal outputs and several sets of parameters, then the number of alternative estimation criteria rises dramatically. Consider, for example, the two output, two parameter problem of chapter 2, section 9. We start with the fully separable MCEM approach in which we must solve:

$$\hat{q} - \min H(q_x, q_y, q_\Phi, q_\Psi) \qquad (3.8.1)$$

Starting with any set of initial probability densities, we can iteratively minimize this cross entropy expression with respect to each unknown density in turn, in any order, iterating back and forth until the estimates converge. Minimizing with respect to q_X, for example, gives:

$$\begin{aligned}\log \hat{q}_{X_{k+1}}(\underline{x}) &= \int_Y \int_\Phi \int_\Psi \hat{q}_{y_k}(\underline{y})\hat{q}_{\Phi_k}(\underline{\Phi})\hat{q}_{\Psi_k}(\underline{\Psi}) \log \frac{p(\underline{x}, \underline{y}, \underline{\Phi} \mid \underline{\Psi})}{\hat{q}_{y_k}(\underline{y})\hat{q}_{\Phi_k}(\underline{\Phi})\hat{q}_{\Psi_k}(\underline{\Psi})}\, d\underline{y}\, d\underline{\Phi}\, d\underline{\Psi} + c\delta n \\ &= E_{y\Phi\Psi}\left[\log \frac{p(\underline{x}, \underline{y}, \underline{\Phi} \mid \underline{\Psi})}{\hat{q}_{y_k}(\underline{y})\hat{q}_{\Phi_k}(\underline{\Phi})\hat{q}_{\Psi_k}(\underline{\Psi})} \,\middle|\, \hat{q}_{y_k}, \hat{q}_{\Phi_k}, \hat{q}_{\Psi_k} \right] + \text{constant}\end{aligned}$$

Minimizing with respect to each of the other unknown densities gives similar formulas. Hybrid MAP/MCEM estimation algorithms can be devised by forcing one or more of the unknown densities to be impulse functions. Yet more estimation algorithms result when unknowns are grouped and jointly estimated. All these algorithms will give different estimates, and will have differing properties and computational difficulties. Nevertheless, all can be solved by the same simple trick of minimizing H with respect to each unknown density in turn.

9. Convergence

The four basic MAP and MCEM algorithms we have discussed were all derived by constructing a cross-entropy function $H(q_X, q_\Phi)$ of two density functions q_X and q_Φ, and iteratively minimizing H with respect to each density in turn. Each iteration strictly decreases the cross-entropy H, and if H is bounded below, the cross-entropy of the estimate $H(\hat{q}_{X_k}, \hat{q}_{\Phi_k})$ must converge monotonically downward to some lower limit H_* as $k \to \infty$. All the MAP algorithms also increase the corresponding likelihood function on each iteration, and if the likelihood function is bounded above, then its value must converge monotonically from below as $k \to \infty$. Unfortunately, neither of these statements necessarily imply that the estimated densities themselves converge to any particular values, or that the limiting densities represent global or even local minima of the cross-entropy function. In some applications, the densities may in fact not converge at all, and the point estimates $\hat{x}$ and $\hat{\phi}$ could diverge to $\pm\infty$, or could conceivably "wander around in circles" without converging to any particular value. The fact that the set of densities over which we minimize H is infinite dimensional further complicates the problem.

In this section we will derive sufficient conditions to guarantee that the estimates generated by our iterative algorithms will converge to the set of local minima and critical points of the cross-entropy (and also the local maxima and critical points of the likelihood fur.ction.) We assume throughout that the unknowns x and ϕ are finite dimensional. The MAP problems are all treated by converting the problem into a minimization over a finite dimensional domain. The MCEM convergence proof is quite a bit more complicated, and our present versions contain some technical assumptions which are probably unnecessary. Because the arguments are rather detailed, the casual reader is encouraged to skip this section.

9.1. General Convergence Theorems

We will first consider the problem of minimizing a function $F(\underline{\alpha} ; \underline{\beta})$ over a finite dimensional domain $\underline{\alpha} \in \Lambda$, $\underline{\beta} \in \Phi$. To simplify the presentation, proofs of all the following theorems are contained in Appendix B.

Assume that the function $F(\underline{\alpha};\underline{\beta})$ is continuous for all $\underline{\alpha} \in \Lambda$, $\underline{\beta} \in \Phi$ and that Λ and Φ have norms $||\underline{\alpha}||_{\Lambda}$ and $||\underline{\beta}||_{\Phi}$. We will calculate the minimum of F by starting at some initial estimate $(\hat{\underline{\alpha}}_0,\hat{\underline{\beta}}_0) \in \Lambda \times \Phi$ and then iteratively minimizing F with respect to each variable in turn:

$$\begin{aligned} &\text{For } k=0,1,\cdots \\ &\hat{\underline{\alpha}}_{k+1} - \min_{\underline{\alpha} \in \Lambda} F(\underline{\alpha} ; \hat{\underline{\beta}}_k) \\ &\hat{\underline{\beta}}_{k+1} - \min_{\underline{\beta} \in \Phi} F(\hat{\underline{\alpha}}_{k+1} ; \underline{\beta}) \end{aligned} \tag{3.9.1}$$

We assume that each of these minimization problems has a finite solution, and in case there are several solutions, we use some deterministic rule to choose one. Let us define the set $\Lambda_x \times \Phi_x$ as the set of points satisfying:

$$\Lambda_x \times \Phi_x = \left\{ (\hat{\underline{\alpha}},\hat{\underline{\beta}}) \;\middle|\; (\hat{\underline{\alpha}},\hat{\underline{\beta}}) \in \Lambda \times \Phi \text{ and } F(\hat{\underline{\alpha}} ; \hat{\underline{\beta}}) = \min_{\underline{\alpha} \in \Lambda} F(\underline{\alpha} ; \hat{\underline{\beta}}) = \min_{\underline{\beta} \in \Phi} F(\hat{\underline{\alpha}} ; \underline{\beta}) \right\} \tag{3.9.2}$$

$\Lambda_x \times \Phi_x$ is just the set of stationary points of the iteration; that is, if we started at an initial pair of estimates drawn from $\Lambda_x \times \Phi_x$, our iterative algorithm would not be able to improve on these estimates. If $F(\underline{\alpha}\,;\underline{\beta})$ has a finite global minimizer on $\Lambda\times\Phi$, then this global minimum $(\hat{\underline{\alpha}},\hat{\underline{\beta}})$ must clearly be an element of $\Lambda_x\times\Phi_x$, and thus $\Lambda_x\times\Phi_x$ will be non-empty. Otherwise, it is possible that $\Lambda_x\times\Phi_x$ could be empty.

Let us define the sequence of points $\{(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)\}$ to be compact if it is contained within a compact subset $\tilde{\Lambda}\times\tilde{\Phi}\subseteq\Lambda\times\Phi$ of the domain, $(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)\in\tilde{\Lambda}\times\tilde{\Phi}$ for all k. In particular, if Λ and Φ are finite dimensional spaces and the sequence $(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)$ remains bounded, then the sequence $\{(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)\}$ is compact. Appendix B then proves:

<u>Theorem 3.9.1:</u> Assume that $F(\underline{\alpha};\underline{\beta})$ is continuous for all $(\underline{\alpha},\underline{\beta})\in\Lambda\times\Phi$, and that the domain $\Lambda\times\Phi$ is closed and non-empty. Suppose that the sequence of estimates $(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)$ generated by our iterative algorithm is compact. Then $F(\underline{\alpha};\underline{\beta})$ converges monotonically downward to a lower limit, $F(\hat{\underline{\alpha}}_k\,;\hat{\underline{\beta}}_k)\to F_*$. Also the sequence of estimates $(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)$ must converge to the set $\Lambda_x\times\Phi_x$ in the sense that the distance from $(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)$ to the nearest point in $\Lambda_x\times\Phi_x$ goes to zero as $k\to\infty$:

$$\lim_{k\to\infty}\left\{\min_{\underline{\alpha},\underline{\beta}\in\Lambda_x\times\Phi_x}\left\|\hat{\underline{\alpha}}_k-\underline{\alpha}\right\|_\Lambda^2+\left\|\hat{\underline{\beta}}_k-\underline{\beta}\right\|_\Phi^2\right\}=0 \tag{3.9.3}$$

Furthermore, the value of F at any limit point $(\hat{\underline{\alpha}},\hat{\underline{\beta}})$ of the sequence $(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)$ must be $F(\hat{\underline{\alpha}}\,;\hat{\underline{\beta}})=F_*$. □

A useful corollary is the following. For the given initial estimate $(\hat{\underline{\alpha}}_0,\hat{\underline{\beta}}_0)$, let us define the level set $\Lambda_0\times\Phi_0$ as the set of all values $\underline{\alpha},\underline{\beta}$ for which $F(\underline{\alpha};\underline{\beta})$ is less than $F(\hat{\underline{\alpha}}_0;\hat{\underline{\beta}}_0)$:

$$\Lambda_0\times\Phi_0=\left\{\underline{\alpha},\underline{\beta}\;\middle|\;(\underline{\alpha},\underline{\beta})\in\Lambda\times\Phi\;\text{ and }\;F(\underline{\alpha}\,;\underline{\beta})\le F(\hat{\underline{\alpha}}_0\,;\hat{\underline{\beta}}_0)\right\} \tag{3.9.4}$$

Then:

Corollary 3.9.1: If F is continuous on $\Lambda\times\Phi$ and the level set $\Lambda_0\times\Phi_0$ is compact, then the sequence $(\hat{\alpha}_k,\hat{\beta}_k,$ ll be compact and will converge to the set $\Lambda_x\times\Phi_x$.

Proof: Since $F(\hat{\alpha}_{k+1};\hat{\beta}_{k+1})\leq F(\hat{\alpha}_0;\hat{\beta}_0)$ for all k, the sequence $(\hat{\alpha}_k,\hat{\beta}_k)$ is contained within $\Lambda_0\times\Phi_0$ and is thus compact. Applying Theorem 3.9.1 gives the result. □

The set $\Lambda_x\times\Phi_x$ thus contains all limit points of the sequence of estimates $(\hat{\alpha}_k,\hat{\beta}_k)$. Note that if $\Lambda_x\times\Phi_x$ contains several points, then this theorem does not necessarily imply that the sequence $(\hat{\alpha}_k,\hat{\beta}_k)$ converges to any particular point in $\Lambda_x\times\Phi_x$; it is still conceivable that the iteration could "skip around" the set $\Lambda_x\times\Phi_x$. As shown in figure 3.1, this set $\Lambda_x\times\Phi_x$ contains the global minimum of F (if it is finite). However, it might also contain certain local minima, stationary points, or even certain local maxima and points on sharp ridges.

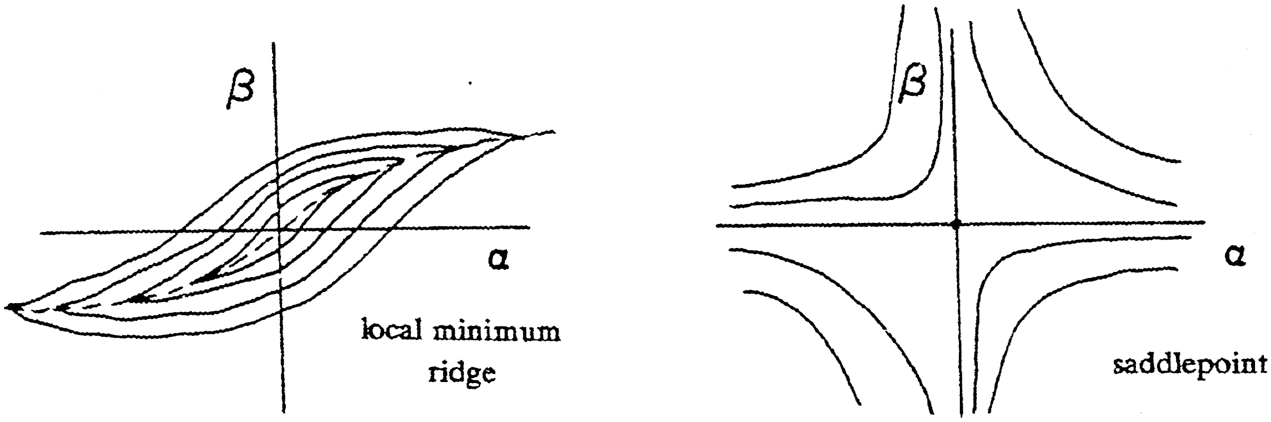


Figure 3.1 - Sets $\Lambda_x\times\Phi_x$

Requiring the function $F(\alpha;\beta)$ to have a continuous first derivative in α and β for all $\alpha\in\Lambda$, $\beta\in\Phi$ eliminates the problem of ridges. With this restriction, Appendix B proves that:

<u>Theorem 3.9.2:</u> In addition to the assumptions of theorem 3.9.1, suppose that $F(\underline{\alpha};\underline{\beta})$ also has a continuous first derivative for all $\underline{\alpha}, \underline{\beta} \in \Lambda \times \Phi$. Then any point $(\hat{\underline{\alpha}},\hat{\underline{\beta}}) \in \Lambda_x \times \Phi_x$ which is in the interior of the domain $\Lambda \times \Phi$ must be a stationary point of $F(\underline{\alpha};\underline{\beta})$. If $(\hat{\underline{\alpha}},\hat{\underline{\beta}}) \in \Lambda_x \times \Phi_x$ is on the boundary of the original domain $\Lambda \times \Phi$, then the derivative of F is inwardly normal to the boundary at $(\hat{\underline{\alpha}},\hat{\underline{\beta}})$; in other words for all sequentially tangent vectors $\underline{h}_\alpha$ and $\underline{h}_\beta$ of Λ and Φ at $\hat{\underline{\alpha}}$ and $\hat{\underline{\beta}}$ respectively:

$$\frac{\partial F(\hat{\underline{\alpha}} \,;\, \hat{\underline{\beta}})^T}{\partial \underline{\alpha}} \, \underline{h}_\alpha \geq 0 \tag{3.9.5}$$

$$\frac{\partial F(\hat{\underline{\alpha}} \,;\, \underline{\beta})^T}{\partial \underline{\beta}} \, \underline{h}_\beta \geq 0$$

(see the discussion in chapter 2, section 10.1) □

Under these continuity and differentiability assumptions, therefore, the iteration is guaranteed to converge to a set of stationary points or local minima of the function F.

This result can be strengthened considerably when the space $\Lambda \times \Phi$ is convex and the function $F(\underline{\alpha};\underline{\beta})$ is convex:

<u>Theorem 3.9.3a:</u> Assume that $F(\underline{\alpha};\underline{\beta})$ is a continuously differentiable and convex function on a convex, closed and non-empty domain $\Lambda \times \Phi$. If the sequence of estimates $(\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k)$ is compact, then $F(\underline{\alpha};\underline{\beta})$ has at least one finite global minimizer, and the sequence of estimates is guaranteed to converge to the closed and convex set of global minimizers. (In fact, the set $\Lambda_x \times \Phi_x$ is just the set of all global minimizers of F.) If the level set $\Lambda_0 \times \Phi_0$ is compact, convergence is guaranteed. □

<u>Theorem 3.9.3b</u> If $\Lambda\times\Phi$ is finite dimensional and F is strictly convex, then if a global minimum exists, it will be unique. In this case, the sequence of estimates $(\hat{\alpha}_k,\hat{\beta}_k)$ will be compact (and thus converge to the unique global minimum) if and only if F has a finite global minimizer. □

<u>Theorem 3.9.3c</u> Finally, if F is also uniformly convex, then it is guaranteed to have a global minimizer on any closed domain $\Lambda\times\Phi$, and convergence to the global minimum is guaranteed. □

Note that all these theorems apply even if $F(\underline{\alpha};\underline{\beta})$ itself is not convex, but there exists a continuous monotonically increasing function $g:\mathbf{R}^1\to\mathbf{R}^1$ such that $g(F(\underline{\alpha};\underline{\beta}))$ is convex and continuously differentiable. Finally, beware that extending these theorems to infinite dimensional domains does not appear to be possible unless many more assumptions are added.

9.2. Convergence of the PSMAP, PSML Algorithms

The application of these convergence theorems to the PSMAP algorithm is direct. Rather than use the MCEM interpretation, we will use the simple interpretation of PSMAP as maximizing the likelihood function $\log p(\underline{x},\underline{\phi})$ with respect to each variable in turn. Then if the sequence $(\hat{\underline{x}}_k,\hat{\underline{\phi}}_k)$ is compact, and $p(\underline{x},\underline{\phi})$ is continuous over $X\times\Phi$, then theorem 3.9.1 guarantees that the parameter and signal estimates converge to the set:

$$X_*\times\Phi_* = \left\{(\hat{\underline{x}},\hat{\underline{\phi}})\;\middle|\;(\hat{\underline{x}},\hat{\underline{\phi}})\in X\times\Phi \text{ and } p(\hat{\underline{x}},\hat{\underline{\phi}}) = \min_{\underline{x}\in X} p(\underline{x},\hat{\underline{\phi}}) = \min_{\underline{\phi}\in\Phi} p(\hat{\underline{x}},\underline{\phi})\right\}$$

In the Bayesian case, it is usually true that $p(\underline{x},\underline{\phi})\to 0$ as $\|\underline{x}\|\to\infty$ or $\|\underline{\phi}\|\to\infty$. Every level set of $p(\underline{x},\underline{\phi})$ will then be bounded, the iterative sequence will be compact, and convergence is assured. If $p(x,\phi)$ is continuously differentiable, then theorem 3.9.2 guarantees that each element $(\hat{x},\hat{\phi})\in X_{\infty}\times\Phi_{\infty}$ must be a critical point of $p(x,\phi)$, or else if it is on the boundary of $X\times\Phi$ then the derivative of $p(x,\phi)$ must be outwardly normal. In some applications we will consider, $X\times\Phi$ will be convex and $p(x,\phi)$ will be log concave on $X\times\Phi$. Theorem 3.9.3 then guarantees that if the sequence $(\hat{x}_k,\hat{\phi}_k)$ is compact, then it will converge to the set of global maxima of $p(x,\phi)$.

The conditions for convergence of PSML are similar to those of PSMAP, except that we replace $p(x,\phi)$ everywhere above by $p(x\,|\phi)$. Note that in this case, however, there is no justification for assuming that $p(x\,|\phi)\to 0$ for all x as $||\phi||\to\infty$, and thus the level sets may not be bounded and PSML may diverge.

9.3. Convergence of the PARMAP, PARML Algorithms

Proving convergence of the PARMAP algorithm is slightly trickier. The key is to use the cross-entropy interpretation of PARMAP, but to rephrase the minimization as a finite dimensional problem. The PARMAP procedure iteratively minimizes the cross-entropy $\bar{H}(q_X,\hat{\phi})$ over all signal probability densities q_X and all impulse functions $q_\Phi=\delta(\phi-\hat{\phi})$. Minimizing with respect to all possible signal densities, however, always yields an estimate of the form $\hat{q}_{X_{k+1}}(x)=p_{X\,|\Phi}(x\,|\hat{\phi}_k)$. Exactly the same answer would be found, therefore, if we restrict the signal density minimization to the finite dimensional class of densities of the form $q_X(x)=p_{X\,|\Phi}(x\,|\psi)$ for $\psi\in\Phi$. Let us define the new cross-entropy expression:

$$\begin{aligned} \bar{H}(\hat{\psi},\hat{\phi}) &\equiv \bar{H}(p_{X\,|\Phi}(x\,|\hat{\psi})\,,\,\hat{\phi}\,) \qquad (3.9.6) \\ &= -\int_X p_{X\,|\Phi}(x\,|\psi)\log\frac{p(x,\phi)}{p_{X\,|\Phi}(x\,|\psi)}\,dx \end{aligned}$$

The PARMAP algorithm can now be written in the following form:

For $k = 0,1,\cdots$

$$\hat{\underline{\phi}}_{k+1} \leftarrow \min_{\underline{\phi}\in\Phi} \tilde{H}(\hat{\underline{\psi}}_k,\underline{\phi}) \qquad (3.9.7)$$

$$\hat{\underline{\psi}}_{k+1} \leftarrow \min_{\underline{\psi}\in\Psi} \tilde{H}(\underline{\psi},\hat{\underline{\phi}}_{k+1})$$

Note that $\hat{\underline{\psi}}_{k+1} = \hat{\underline{\phi}}_{k+1}$. We have thus reduced the problem to an iterative minimization over a finite dimensional domain. Assume that Φ is closed and that the function $\tilde{H}(\underline{\psi},\underline{\phi})$ (and thus also the density $p(X,\underline{\phi})$) is continuous in $\underline{\phi}$ and $\underline{\psi}$. Using the fact that $\hat{\underline{\psi}}_{k+1} = \hat{\underline{\phi}}_{k+1}$, theorem 3.9.1 guarantees that if the sequence $\hat{\underline{\phi}}_k$ is compact, then it converges to the set of limit points Φ_∞ defined by:

$$\Phi_\infty = \left\{ \hat{\underline{\phi}} \;\middle|\; \hat{\underline{\phi}}\in\Phi_0 \text{ and } \tilde{H}(\hat{\underline{\phi}},\hat{\underline{\phi}}) = \min_{\underline{\psi}\in\Phi} \tilde{H}(\underline{\psi},\hat{\underline{\phi}}) = \min_{\underline{\phi}\in\Phi} \tilde{H}(\hat{\underline{\phi}},\underline{\phi}) \right\} \qquad (3.9.8)$$

In the Bayesian case it is usually true that $p(\underline{\phi})\rightarrow 0$ as $||\underline{\phi}||\rightarrow\infty$. Since:

$$p(X,\underline{\phi}) \leq p(X|\underline{\phi})\,p(\underline{\phi}) \leq p(\underline{\phi}) \qquad (3.9.9)$$

then it must also be true that $p(X,\underline{\phi})\rightarrow 0$ as $||\underline{\phi}||\rightarrow\infty$. Every level set of $p(X,\underline{\phi})$ will then be bounded, the iterative sequence will be compact, and convergence is guaranteed. If $p(X,\underline{\phi})$ and $\tilde{H}(\underline{\psi},\underline{\phi})$ are also continuously differentiable in $\underline{\phi}$, $\underline{\psi}$, then Appendix B applies theorem 3.9.2 to show that each point $\hat{\underline{\phi}}\in\Phi$ is either a critical point of $p(X,\underline{\phi})$ or else if $\hat{\underline{\phi}}$ is on the boundary of Φ, then the derivative of $p(X,\underline{\phi})$ at $\hat{\underline{\phi}}$ must be outwardly normal:

$$\frac{\partial \log p(X,\underline{\phi})^{T}}{\partial\underline{\phi}}\, h_{\phi} \geq 0 \qquad (3.9.10)$$

for all sequentially tangent vectors h_{ϕ} of Φ at $\hat{\underline{\phi}}$. Thus $\hat{\underline{\phi}}$ is either a stationary point at which $\frac{\partial \log p(X,\hat{\underline{\phi}})}{\partial\underline{\phi}} = 0$, or else it is a local maximum on the boundary of Φ.

Theorem 3.9.3 can also be applied to the PARMAP algorithm. Suppose that X

and Φ are convex, and that $p(\underline{x},\underline{\phi})$ is log concave and differentiable. Prékopa [13] has proven that the marginal density $p(X,\underline{\phi})$ will then also be log concave (see Appendix E). In this case, the limit set Φ_x can only contain the global maxima of $p(X,\underline{\phi})$, and thus if the estimates remain bounded, they will converge to the set of global maxima of $p(X,\underline{\phi})$.

Finally, Appendix B proves that the difference between successive PARMAP model estimates tends to zero, in the sense that for *any* measurable set $\bar{X} \subseteq X$:

$$\lim_{k \to \infty} \left(p(\bar{X}\,|\hat{\underline{\phi}}_k) - p(\bar{X}\,|\hat{\underline{\phi}}_{k+1}) \right) = 0 \tag{3.9.11}$$

This usually, though not always, implies that the difference between successive parameter estimates $\hat{\underline{\phi}}_{k+1}=\hat{\underline{\phi}}_k$ (the "step size") tends to zero.

9.4. Convergence of the SIGMAP Algorithm

The convergence properties for SIGMAP are identical to those of PARMAP, except that the roles of $\underline{x}$ and $\underline{\phi}$ are reversed.

9.5. Convergence of the MCEM Algorithm

Proving convergence of the iterative MCEM algorithm is greatly complicated by the fact that the domain of the minimization problem is the infinite dimensional space of probability densities. Much of the familiar intuition concerning optimization on finite dimensional domains does not apply to problems such as this. The most general proofs of convergence we have at present unfortunately require some technical assumptions in the middle concerning the rate of oscillation of the estimated densities and the behavior of their tails, which are difficult to verify in practice. We therefore present three separate convergence analyses for MCEM. The first requires minimal assumptions, but proves the existence of a limiting measure with lower cross-entropy than any of the estimates. Unfortunately, this proof does not show that the limiting measure is a stationary point of the algorithm. The second proof considers the case when the constraint sets have a finite number of elements, and proves convergence to the set of stationary points of the algorithm and critical points of the cross-entropy. The third proof analyzes the case of exponential families of densities and proves convergence to a stationary point of the algorithm provided certain boundedness conditions hold. Proofs can be found in Appendix B.

9.5.1. MCEM - General Convergence Proof

Let us first define some notation. Let $Q_X()$, $Q_y()$ and $P()$ be the measures associated with $q_X(\underline{x})$, $q_\Phi(\underline{\phi})$ and $p(\underline{x},\underline{\phi})$:

$$Q_X(\bar{X}) = \int_{\bar{X}} q_X(\underline{x})\, d\underline{x}$$

$$Q_\Phi(\bar{\Phi}) = \int_{\bar{\Phi}} q_\Phi(\underline{\phi})\, d\underline{\phi} \qquad (3.9.12)$$

$$P(\bar{X},\bar{\Phi}) = \int_{\bar{X}}\int_{\bar{\Phi}} p(\underline{x},\underline{\phi})\, d\underline{x}\, d\underline{\phi}$$

For technical reasons, it is easier to analyze the convergence behavior of the sequence of measures $\hat{Q}_{X_k}$, $\hat{Q}_{\Phi_k}$ than to analyze the convergence of the sequence of densities $\hat{q}_{X_k}$, $\hat{q}_{\Phi_k}$. It is therefore convenient to be able to define the cross-entropy of the separable measure $Q_X(X)Q_\Phi(\Phi)$ without using any reference to density functions. We take theorem 2.4.5 as our definition of the cross-entropy of the measures Q_X, Q_Φ, P:

$$H(Q_X,Q_\Phi) \equiv \sup_P \sum_{i,j} Q_X(X_i)Q_\Phi(\Phi_j) \log \frac{Q_X(X_i)Q_\Phi(\Phi_j)}{P(X_i,\Phi_j)} \qquad (3.9.13)$$

where the supremum is taken over all finite partitions P of the space $X \times \Phi$ (for a careful development of cross-entropy from a measure-theoretic point of view, see Pinsker [14].) When the measures Q_X, Q_Φ have Radon-Nikodyn derivatives q_X, q_Φ, then theorem 2.4.5 guarantees that this definition is identical to our previous definition. The advantage of (3.9.13) is that it defines the cross-entropy even for measures which do not have corresponding densities. We can now state our most general MCEM convergence theorem in the following form:

Theorem 3.9.4 Assume that X and Φ are closed and measurable sets, and that $p(x,\phi)$ is a proper, strictly positive and piecewise continuous probability density, so that $p(x,\phi)>0$ for all $x \in X$ and $\phi \in \Phi$ and $0<p(X,\Phi)\leq 1$. Start with strictly positive initial density estimates $\hat{q}_{X_0}(x)>0$ and $\hat{q}_{\Phi_0}(\phi)>0$ for which $H(\hat{q}_{X_0},\hat{q}_{\Phi_0})<\infty$. Then the sequence of density estimates generated by the iterative MCEM algorithm has the following properties:

a) The densities $\hat{q}_{X_k}(x)$, $\hat{q}_{\Phi_k}(\phi)$ are all well defined.

b) $H(\hat{q}_{X_k},\hat{q}_{\Phi_k})$ converges monotonically strictly downward to a finite limit H_*. The normalization constants c_{x_k} and c_{ϕ_k} also converge monotonically strictly upward to the finite limit e^{-H_*}.

c) $\hat{Q}_{X_k}(\tilde{X})>0$ and $\hat{Q}_{\Phi_k}(\tilde{\Phi})>0$ for all measurable subsets $\tilde{X}$, $\tilde{\Phi}$ of X, Φ.

d) The difference between successive measure estimates tends to zero in the sense that:

$$\lim_{k\to\infty} \hat{Q}_{X_k}(\tilde{X}) - \hat{Q}_{X_{k+1}}(\tilde{X}) = 0$$
$$\lim_{k\to\infty} \hat{Q}_{\Phi_k}(\tilde{\Phi}) - \hat{Q}_{\Phi_{k+1}}(\tilde{\Phi}) = 0$$

where $\tilde{X}$ and $\tilde{\Phi}$ are any measurable subsets of X and Φ.

e) Let Ψ be a measurable subset of $X \times \Phi$. Then the measure assigned by the separable densities to Ψ is bounded above by:

$$\int_{\Psi} \hat{q}_{X_k}(x)\hat{q}_{\Phi_k}(\phi)\, dx\, d\phi \le \frac{H(\hat{q}_{X_0},\hat{q}_{\Phi_0}) + \log p(X,Y) + \log 2}{-\log \int_{\Psi} p(x,\phi)\, dx\, d\phi}$$

This implies that the densities $\hat{q}_{X_k}(x)$ and $\hat{q}_{\Phi_k}(\phi)$ are stochastically bounded; that is, for any $0<\delta<1$, there exists a radius T such that:

$$\int_{\substack{X \\ \|x\| \le T}} \hat{q}_{X_k}(x)dx \ge 1-\delta \qquad \text{for all } k$$

$$\int_{\substack{\Phi \\ \|\phi\| \le T}} \hat{q}_{\Phi_k}(\phi)d\phi \ge 1-\delta$$

f) There exists at least one subsequence of measures $\{\hat{Q}_{X_{k_i}},\hat{Q}_{\Phi_{k_i}}\} \subseteq \{\hat{Q}_{X_k},\hat{Q}_{\Phi_k}\}$ which converges to a proper limit measure $\overline{Q}_X$, $\overline{Q}_\Phi$:

$$\lim_{i\to\infty} \hat{Q}_{X_{k_i}}(\tilde{X}) = \overline{Q}_X(\tilde{X}) \tag{3.9.14}$$

$$\lim_{i\to\infty} \hat{Q}_{\Phi_{k_i}}(\tilde{\Phi}) = \overline{Q}_\Phi(\tilde{\Phi})$$

for any measurable subsets $\tilde{X}$, $\tilde{\Phi}$ of X, Φ.

g) The cross-entropy of this limit measure $\overline{Q}_X$, $\overline{Q}_\Phi$ is less than the limit of the cross-entropies:

$$H(\overline{Q}_X,\overline{Q}_\Phi) \le \lim_{k\to\infty} H(\hat{Q}_{X_k},\hat{Q}_{\Phi_k}) = H.$$

$$H(\overline{Q}_X, Q_\Phi) \leq \liminf_{i \to \infty} H(\hat{Q}_{X_{k_i}}, Q_\Phi) \qquad \text{for all } Q_\Phi$$

$$H(Q_X, \overline{Q}_\Phi) \leq \liminf_{i \to \infty} H(Q_X, \hat{Q}_{\Phi_{k_i}}) \qquad \text{for all } Q_X$$

h) The limit measure $\overline{Q}_X$, $\overline{Q}_\Phi$ satisfies the same upper bound as in property (e):

$$\overline{Q}_X(\bar{X})\overline{Q}_\Phi(\bar{\Phi}) \leq \frac{H(\hat{Q}_{X_0}, \hat{Q}_{\Phi_0}) + \log p(X, \Phi) + \log 2}{\log P(\bar{X}, \bar{\Phi})}$$

Statement a) implies that the iteration always produces a valid probability density estimate. Statement b) was proven in section 2 and implies that every density estimate is better than the last in the sense that the cross-entropy decreases with each iteration. Statement c) says that all density estimates are strictly positive. Statement d) says that the difference between successive density estimates (the "step" size) goes to zero as $k \to \infty$. Statement e) says that the density estimates must not put significant probability at values where the original model density $p(x, \phi)$ would not put significant probability. In particular, the estimated measures can not become impulse-like, and they can't put significant probability at infinite values of x or ϕ; this is a boundedness property similar to that we had to assume in our proofs of PSMAP, PARMAP and SIGMAP. Statement f) is a restatement of the Helly Selection Theorem ([15] volume 2, or [16]), which says that every stochastically bounded sequence of measures must have at least one limit, and that limit must be a proper measure. Statement g) is a consequence of the convexity of H, and says that the cross-entropy of any limiting measure $\overline{Q}_X$, $\overline{Q}_\Phi$ must be less than the limit of the cross-entropies of the estimates $\hat{Q}_{X_k}$, $\hat{Q}_{\Phi_k}$. Statement h) follows because this upper bound must hold for any measure Q_X, Q_Φ whose cross-entropy is less than $H(\hat{Q}_{X_0},\hat{Q}_{\Phi_0})$.

9.5.2. MCEM - Convergence for Finite Constraint Sets

The problem with theorem 3.9.4 is that it is not sufficient to show that $\overline{Q}_X$, $\overline{Q}_\Phi$ is actually a stationary point of the algorithm and a critical point of the cross-entropy function. In fact, it doesn't even prove that this limiting measure corresponds to a density at all. This difficulty is caused by working with unbounded constraint spaces and continuous probability densities. Let us therefore consider a much simpler finite dimensional situation in which much stronger convergence results can be stated. Suppose that the constraint space X actually contains only N distinct points $\{x_i\}$ and the parameter space Φ contains onl; M distinct points $\{\phi_i\}$. (N and M may be huge, but they must be finite.) Also suppose that the original model density is atomic, assigning a non-zero probability to each pair (x_i,ϕ_j):

$$0 < \mathrm{p}(x_i,\phi_j) \leq 1 \qquad \text{for all } i,j \qquad (3.9.15)$$
$$\sum_{i=1}^{N}\sum_{j=1}^{M} \mathrm{p}(x_i,\phi_j) = 1$$

The separable densities q_X and q_Φ will also have to be atomic (otherwise the cross-entropy would be infinite).

$$0 \leq \hat{\mathrm{q}}_X(x_i) \leq 1 \qquad 0 \leq \hat{\mathrm{q}}_\Phi(\phi_j) \leq 1$$
$$\sum_{i=1}^{N} \hat{\mathrm{q}}_X(x_i) = 1 \qquad \sum_{j=1}^{M} \hat{\mathrm{q}}_\Phi(\phi_j) = 1 \qquad (3.9.16)$$

We can now view each density $\hat{\mathrm{q}}_X$ or $\hat{\mathrm{q}}_\Phi$ as a finite dimensional vector, each of whose components is the probability $\hat{\mathrm{q}}_X(x_i)$ or $\hat{\mathrm{q}}_\Phi(\phi_j)$ of the corresponding point in the probability spaces. Let us use an l_1 norm on the space of the "probability density vectors":

$$\|\hat{\mathrm{q}}_X\| \equiv \sum_{i=1}^{N} \left|\hat{\mathrm{q}}_X(x_i)\right| \qquad (3.9.17)$$

$$\|\hat{q}_\Phi\| \equiv \sum_{j=1}^{M} \left|\hat{q}_\Phi(\underline{\phi}_j)\right|$$

Because of the constraints (3.9.16), $\|\hat{q}_X\| = \|\hat{q}_\Phi\| = 1$ for all q_X, q_Φ, and the finite dimensional set of all such probability density vectors is closed and bounded.

The cross-entropy function in this case will be a simple summation:

$$H(q_X, q_\Phi) = \sum_{i=1}^{N} \sum_{j=1}^{M} q_X(\underline{x}_i) q_\Phi(\underline{\phi}_j) \log \frac{q_X(\underline{x}_i) q_\Phi(\underline{\phi}_j)}{p(\underline{x}_i, \underline{\phi}_j)} \tag{3.9.18}$$

This cross-entropy function is analytic at all densities which are strictly positive, $\hat{q}_X(\underline{x}_i) > 0$ and $\hat{q}_\Phi(\underline{\phi}_j) > 0$ for all i, j. Our iterative MCEM algorithm minimizes this cross-entropy with respect to each density in turn, giving estimates $\hat{q}_{X_k}(\underline{x}_i)$, $\hat{q}_{\Phi_k}(\underline{\phi}_j)$. Appendix B then proves:

<u>Theorem 3.9.5</u> Let $p(\underline{x}, \underline{\phi})$, $q_X(\underline{x})$, $q_\Phi(\underline{\phi})$ be atomic densities on finite sets X, Φ. Assume that $p(\underline{x}, \underline{\phi})$ is strictly positive, with $p(\underline{x}_i, \underline{\phi}_j) \geq \epsilon > 0$ for all i, j. Then the sequence of density estimates generated by our iterative MCEM algorithm has the following properties:

a) The estimates $\hat{q}_{X_k}$, $\hat{q}_{\Phi_k}$ are all well defined. The cross-entropy $H(\hat{q}_{X_k}, \hat{q}_{\Phi_k})$ converges monotonically strictly downward to a finite limit H_*.

b) The density estimates are all strictly positive:

$$\hat{q}_{X_k}(x_i) \geq \frac{1}{N}\epsilon \qquad \text{for all } i$$
$$\hat{q}_{\Phi_k}(\phi_j) \geq \frac{1}{M}\epsilon \qquad \text{for all } j$$

c) The difference between successive density estimates (the "step size") goes to zero in the sense that:

$$\begin{aligned} \hat{q}_{X_k}(\underline{x}_i) - \hat{q}_{X_{k+1}}(\underline{x}_i) &\to 0 \\ \hat{q}_{\Phi_k}(\underline{\phi}_j) - \hat{q}_{\Phi_{k+1}}(\underline{\phi}_j) &\to 0 \end{aligned} \qquad \text{as } k \to \infty \quad \text{for all } i, j$$

d) There exists at least one convergent subsequence $\hat{q}_{X_{k_i}}$, $\hat{q}_{\Phi_{k_i}}$ with limiting density $\bar{q}_X$, $\bar{q}_\Phi$ such that:

$$\lim_{m \to \infty} \hat{q}_{X_{k_m}}(x_i) = \bar{q}_X(x_i) \qquad \text{for all } i, j$$

$$\lim_{m \to \infty} \hat{q}_{\Phi_{k_m}}(\phi_j) = \bar{q}_\Phi(\phi_j)$$

e) All such limiting densities are strictly positive:

$$\bar{q}_X(x_i) \geq \frac{1}{N}\epsilon \qquad \text{for all } i$$

$$\bar{q}_\Phi(\phi_j) \geq \frac{1}{M}\epsilon \qquad \text{for all } j$$

f) All such limiting densities must be stationary points of the algorithm:

$$H(\bar{q}_X, \bar{q}_\Phi) = \min_{q_X} H(q_X, \bar{q}_\Phi) = \min_{q_\Phi} H(\bar{q}_X, q_\Phi)$$

g) Form the Lagrangian for the problem of minimizing $H(q_X, q_\Phi)$ subject to constraints (3.9.16):

$$L_{x,\phi}(q_X, q_\Phi) = H(q_X, q_\Phi) + \lambda_x \left(\sum_{i=1}^{N} q_X(x_i) - 1 \right) + \lambda_\phi \left(\sum_{j=1}^{M} q_\Phi(\phi_j) - 1 \right)$$

Then for appropriate values of the multipliers λ_x, λ_ϕ, the limit $\bar{q}_X$, $\bar{q}_\Phi$ is a critical point of $L_{x,\phi}$:

$$\frac{\partial L_{x,\phi}(\bar{q}_X, \bar{q}_\Phi)}{\partial q_X} = 0 \qquad \text{and} \qquad \frac{\partial L_{x,\phi}(\bar{q}_X, \bar{q}_\Phi)}{\partial q_\Phi} = 0$$

Thus if the estimation problem involves a finite number of signal and parameter values, then convergence of the MCEM algorithm to the set of critical points of the cross-entropy is guaranteed. We conjecture that this conclusion also holds for the more general case of continuous densities and unbounded constraint sets considered in theorem 3.9.4, but we have not been able to prove this.

Note that under the assumptions of this theorem, our MAP algorithms would converge in a finite number of steps. (This is because a finite number of steps will test every possible parameter or signal value.)

9.5.3. Exponential Densities

If $p(x,\phi)$ forms an exponential class of densities, then yet another type of convergence proof can be given for MCEM. First of all, we assume that $p(x,\phi)$ has the form given in (3.7.1). Substituting into our cross-entropy expression, it is easy to see that all the density estimates $\hat{q}_X$, $\hat{q}_\Phi$ generated by our iterative MCEM algorithm must have the form:

$$\hat{q}_{X,\alpha}(x) = \frac{1}{c_x} g(x) \exp\left[\sum_{i=1}^{r} \alpha_i t_i(x) \right] \tag{3.9.19}$$

$$\hat{q}_{\Phi,\beta}(\phi) = \frac{1}{c_\phi} h(\phi) \exp\left[\sum_{i=1}^{r} \beta_i \pi_i(\phi) \right]$$

where the normalization constants c_x and c_π are given by:

$$c_x = \int_X g(x) \exp\left[\sum_{i=1}^{r} \alpha_i t_i(x) \right] dx \tag{3.9.20}$$

$$c_\phi = \int_\Phi h(\phi) \exp\left[\sum_{i=1}^{r} \beta_i \pi_i(\phi) \right] d\phi$$

Lehmann [10] proves the following:

<u>Lemma 3.9.6.1</u> Let Λ_α and Λ_β be the sets of parameters α and β for which the normalization constraints c_x and c_ϕ are finite. Then Λ_α and Λ_β are convex sets, and c_x and c_ϕ are analytic and convex functions of α and β in the interior of the

"natural" parameter sets Λ_α and Λ_β. Furthermore:

$$\begin{aligned}
\frac{\partial \log c_x}{\partial \underline{\alpha}} &= E_X\left[\underline{t}(\underline{x}) \,\middle|\, \hat{q}_{X,\alpha}\right] \\
\frac{\partial^2 \log c_x}{\partial \underline{\alpha}^2} &= \mathrm{Cov}_X\left[\underline{t}(\underline{x}) \,\middle|\, \hat{q}_{X,\alpha}\right] \;\equiv\; R_t \\
\frac{\partial \log c_\phi}{\partial \underline{\beta}} &= E_\Phi\left[\underline{\pi}(\underline{\phi}) \,\middle|\, \hat{q}_{\Phi,\beta}\right] \qquad (3.9.21) \\
\frac{\partial^2 \log c_\phi}{\partial \underline{\beta}^2} &= \mathrm{Cov}_\Phi\left[\underline{\pi}(\underline{\phi}) \,\middle|\, \hat{q}_{\Phi,\beta}\right] \;\equiv\; R_\pi
\end{aligned}$$

We define R_t and R_π as the covariance matrices above for convenience in later discussions.

Because all the estimates $\hat{q}_{X_k}$, $\hat{q}_{\Phi_k}$ generated by the MCEM algorithm have the form given in (3.9.19), restricting the minimization to this class of densities will not change the estimates generated, nor the solution to which they converge. With this restriction, the densities $\hat{q}_{X,\alpha}$, $\hat{q}_{\Phi,\beta}$ will depend only on the parameters $\underline{\alpha}$, $\underline{\beta}$ and we can view the cross-entropy as solely a function of these finite dimensional vectors $\underline{\alpha}$ and $\underline{\beta}$:

$$H(\underline{\alpha},\underline{\beta}) \;\equiv\; H(\hat{q}_{X,\alpha}, \hat{q}_{\Phi,\beta})$$

Appendix B proves:

<u>Lemma 3.9.6.2</u> The cross-entropy $H(\underline{\alpha},\underline{\beta})$ is an analytic function of $\underline{\alpha}$, $\underline{\beta}$ in the interior of the natural sets Λ_α and Λ_β.

Now we can rephrase our iterative MCEM algorithm as minimizing the cross-entropy $H(\underline{\alpha},\underline{\beta})$ over all $\underline{\alpha}$ and $\underline{\beta}$ in the natural parameter spaces Λ_α, Λ_β:

$$\begin{aligned}
\hat{\underline{\alpha}}_{k+1} &\leftarrow \min_{\underline{\alpha}\in\Lambda_\alpha} H(\underline{\alpha},\hat{\underline{\beta}}_k) \qquad (3.9.22) \\
\hat{\underline{\beta}}_{k+1} &\leftarrow \min_{\underline{\beta}\in\Lambda_\beta} H(\hat{\underline{\alpha}}_{k+1},\underline{\beta})
\end{aligned}$$

or:

$$\begin{aligned}\hat{\underline{\alpha}}_{k+1} &= \mathrm{E}_{\Phi}\left[\,\underline{\pi}(\underline{\Phi})\;\middle|\;\hat{q}_{\Phi,\beta_k}\,\right] \\ \hat{\underline{\beta}}_{k+1} &= \mathrm{E}_{X}\left[\,\underline{t}(\underline{x})\;\middle|\;\hat{q}_{X,\alpha_k}\,\right]\end{aligned} \qquad (3.9.23)$$

Appendix B proves:

Theorem 3.9.6 Let $\hat{\underline{\alpha}}_k$, $\hat{\underline{\beta}}_k$ be the iterative sequence of estimates generated by minimizing the cross-entropy $H(\underline{\alpha},\underline{\beta})$. Suppose the estimates are not only bounded, but are also bounded away from the natural parameter set boundaries, so that there exists a radius $\epsilon>0$ such that all values of $\underline{\alpha}$, $\underline{\beta}$ within distance ϵ of any $\hat{\underline{\alpha}}_k$, $\hat{\underline{\beta}}_k$ are in the interior of Λ_α, Λ_β. Then the sequence $(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)$ converges to the set $\Lambda_{\alpha,\infty}\times\Lambda_{\beta,\infty}$ of stationary points of the algorithm:

$$\Lambda_{\alpha,\infty}\times\Lambda_{\beta,\infty} = \left\{(\hat{\underline{\alpha}},\hat{\underline{\beta}})\;\middle|\;H(\hat{\underline{\alpha}},\hat{\underline{\beta}}) = \min_{\underline{\alpha}\in\Lambda_\alpha} H(\underline{\alpha},\hat{\underline{\beta}}) = \min_{\underline{\beta}\in\Lambda_\beta} H(\hat{\underline{\alpha}},\underline{\beta})\right\}$$

in the sense that the minimum distance from $\hat{\underline{\alpha}}_k$, $\hat{\underline{\beta}}_k$ to this set tends to zero:

$$\lim_{k\to\infty}\left\{\min_{(\hat{\underline{\alpha}},\hat{\underline{\beta}})\in\Lambda_{\alpha,\infty}\times\Lambda_{\beta,\infty}} \|\hat{\underline{\alpha}}_k-\hat{\underline{\alpha}}\|^2+\|\hat{\underline{\beta}}_k-\hat{\underline{\beta}}\|^2\right\}$$

Every limit point $(\hat{\underline{\alpha}},\hat{\underline{\beta}})$ of the iteration must also be a critical point of the cross-entropy:

$$\frac{\partial H(\hat{\underline{\alpha}},\hat{\underline{\beta}})}{\partial\underline{\alpha}} = 0 \qquad \text{and} \qquad \frac{\partial H(\hat{\underline{\alpha}},\hat{\underline{\beta}})}{\partial\underline{\beta}} = 0$$

All limit points must correspond to the same cross-entropy, $H(\hat{\underline{\alpha}},\hat{\underline{\beta}}) = \lim_{k\to\infty} H(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)$. Finally, if we define:

$$\begin{aligned}F(\underline{\alpha}) &\equiv \min_{\underline{\beta}} H(\underline{\alpha},\underline{\beta}) \\ G(\underline{\beta}) &\equiv \min_{\underline{\alpha}} H(\underline{\alpha},\underline{\beta})\end{aligned}$$

then at each limit point $(\hat{\underline{\alpha}},\hat{\underline{\beta}})$:

$$\frac{\partial F(\hat{\underline{\alpha}})}{\partial \underline{\alpha}} = 0 \qquad \text{and} \qquad \frac{\partial G(\hat{\underline{\beta}})}{\partial \underline{\beta}} = 0 \qquad \square$$

As in our previous proofs, convergence is only guaranteed to a set of stationary points, and not to any point in particular or necessarily to a global minimizer. The only unfortunate part of this theorem is that it appears necessary to *assume* that the estimates $\hat{\underline{\alpha}}_k$, $\hat{\underline{\beta}}_k$ not only remain bounded but also remain bounded away from the natural parameter set boundary.

To decide whether or not a limit point $(\hat{\underline{\alpha}},\hat{\underline{\beta}})$ of the iteration is a local minimizer of the cross-entropy, rather than just a saddle point or local maximizer, we can calculate the second derivative of $H(\underline{\alpha},\underline{\beta})$ at $(\hat{\underline{\alpha}},\hat{\underline{\beta}})$ and check whether or not it is positive definite. It is convenient to transform first to variables $\underline{\iota}$, $\underline{\rho}$ defined as the expected values of $\underline{\iota}(x)$ and $\underline{\pi}(\phi)$ given the densities $\hat{q}_{X,\alpha}$ and $\hat{q}_{\Phi,\beta}$:

$$\underline{\iota}(\underline{\alpha}) \equiv \mathrm{E}_X\left[\underline{\iota}(x)\,\middle|\,\hat{q}_{X,\alpha}\right] \tag{3.9.24}$$
$$\underline{\rho}(\underline{\beta}) \equiv \mathrm{E}_\Phi\left[\underline{\pi}(\phi)\,\middle|\,\hat{q}_{\Phi,\beta}\right]$$

The Jacobian of the parameter transformation is:

$$\frac{\partial(\underline{\iota},\underline{\rho})}{\partial(\underline{\alpha},\underline{\beta})} = \begin{pmatrix} \mathrm{R}_\iota & 0 \\ 0 & \mathrm{R}_\pi \end{pmatrix} \tag{3.9.25}$$

where R_ι and R_π are the covariances of $\underline{\iota}(x)$ and $\underline{\pi}(\phi)$ in (3.9.21). Theorem 3.9.4 guarantees that the cross-entropy density estimates remain non-impulse-like. Thus, since the constraint spaces are measurable, the covariance matrices must be strictly positive definite, and the transformation from $(\underline{\alpha},\underline{\beta})$ to $(\underline{\iota},\underline{\rho})$ is invertible. With this transformation, we can view the cross-entropy as a function of $\underline{\rho}$ and $\underline{\iota}$, $H(\underline{\iota},\underline{\rho}) \equiv H(\underline{\alpha},\underline{\beta})$. Minimizing H over $\underline{\iota}$, $\underline{\rho}$ is exactly equivalent to minimizing over $\underline{\alpha}$, $\underline{\beta}$

or over the densities q_X, q_Φ directly. Because of (3.9.23), we will get estimates $\hat{\underline{\rho}}_k = \underline{\alpha}_{k-1}$ and $\hat{\underline{\tau}}_k = \underline{\beta}_k$. This conveniently implies that the limit of $(\hat{\underline{\tau}}_k, \hat{\underline{\rho}}_k)$ will be identical to the limit of $(\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k)$.

The second derivative of $H(\underline{\tau}, \underline{\rho})$ is easily computed:

$$\frac{\partial^2 H(\underline{\tau},\underline{\rho})}{\partial(\underline{\tau},\underline{\rho})^2} = \begin{pmatrix} R_\tau^{-1} & -I \\ -I & R_\pi^{-1} \end{pmatrix} \tag{3.9.26}$$

If this second derivative is positive definite at the limit $(\hat{\underline{\tau}}, \hat{\underline{\rho}}) = (\hat{\underline{\beta}}, \hat{\underline{\alpha}})$, then the limit point must be a local minimum of $H(\underline{\tau}, \underline{\rho})$, and the corresponding densities $\hat{q}_X$, $\hat{q}_\Phi$ must also be a local minimum. Thus a sufficient condition for the limit to be a local minimum is that:

$$R_\pi^{-1} > 0 \qquad \text{and} \qquad R_\tau^{-1} > R_\pi \tag{3.9.27}$$

By using this second derivative, Appendix E proves the following useful result:

Theorem 3.9.7 Suppose the model density $p(\underline{x}, \underline{\phi})$ is in natural exponential form:

$$p(\underline{x}, \underline{\phi}) = g(\underline{x}) h(\underline{\phi}) \exp(\underline{x}^T \underline{\phi}) \tag{3.9.28}$$

and that $p(\underline{x}, \underline{\phi})$ is log concave. Then the cross-entropy $H(\underline{\tau}, \underline{\rho})$ is a convex function of the transformed vectors $\underline{\tau}, \underline{\rho}$ and any limit point of the MCEM iteration which is in the interior of the natural parameter space $\Lambda_\alpha \times \Lambda_\beta$ must be a global optimizing solution.

10. Discussion of Convergence Theorems

These theorems prove that if the probability densities have compact level sets and are continuously differentiable, then all three MAP algorithms converge to a set of estimates where the likelihood function has zero slope, or else where the likelihood function has a local maximum on the boundary of the signal or parameter space. The

MCEM algorithm also converges to a set of limiting measures whose cross-entropy is lower than the cross-entropy of any of the estimates. If we add additional assumptions, such as requiring the density to be an exponential class and require the estimates to remain bounded, then the limiting densities can be shown to be stationary points of the algorithm and critical points of the cross-entropy. All these algorithms thus act something like iterative steepest descent algorithms [4] converging to a set of local extrema or critical points of the objective function. Unless the objective function is convex, there is no guarantee that the convergent estimate is the global extremum. Furthermore, note that convergence is not guaranteed to a particular estimate, but only to a set of estimates, and thus the estimates may actually "wander around" the limit set and not strictly converge.

In the Bayesian case, MCEM always yields stochastically bounded estimates, and thus always has at least one limiting measure. Furthermore, in most Bayesian applications, the probability of extremely large values of $\underline{x}$ or $\underline{\phi}$ will be negligible, all level sets will be bounded, and thus the MAP algorithms will also be guaranteed to converge. The Maximum Likelihood version of the MCEM algorithm, however, is not guaranteed to converge, since the *a priori* density $p(\underline{\phi})$ is implicitly assumed to be flat, and thus $p(\underline{x},\underline{\phi})$ may not be a properly integrable density. Similarly, the level sets of $p(X|\underline{\phi})$ and $p(\underline{x}|\underline{\phi})$ may not be bounded, and it is possible that the PARML and PSML estimates could diverge to infinity. In any case, it should be remembered that regardless of whether or not the convergence theorems apply, each pass of these algorithms improves the value of the log likelihood and cross-entropy functions. Thus, even though the estimates may not converge, in this sense each successive estimate is "better" than the last.

11. Geometric Rate of Convergence - Acceleration Techniques

In the following chapters, we will analyze a variety of applications and prove that, in these examples, our iterative algorithms converge linearly at a rate that can be approximately calculated in terms of the structure of the problem. In fact, it can be shown heuristically that if $F(\alpha;\beta)$ is approximately quadratic near the minimum, then each iteration of our algorithm defines a contraction mapping, and the sequence of estimates $(\hat{\alpha}_k, \hat{\beta}_k)$ will converge at a geometric rate. For simplicity, assume that α and β are scalars. Suppose that $F(\alpha\,;\beta)$ has a local minimum at $(\bar{\alpha},\bar{\beta})$ and that it can be approximated as quadratic near $(\bar{\alpha},\bar{\beta})$:

$$F(\alpha\,;\beta) \approx K + \begin{pmatrix} \alpha-\bar{\alpha} & \beta-\bar{\beta} \end{pmatrix} \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \begin{pmatrix} \alpha-\bar{\alpha} \\ \beta-\bar{\beta} \end{pmatrix} \tag{3.11.1}$$

where: $|\rho| < 1$

Let $K_\alpha()$ and $K_\beta()$ be the mappings from β to α and back again defined by our iteration:

$$\begin{aligned} \hat{\alpha} &= K_\alpha(\hat{\beta}) - \min_\alpha F(\alpha\,;\hat{\beta}) \\ \hat{\beta}' &= K_\beta(\hat{\alpha}') - \min_\beta F(\hat{\alpha}'\,;\beta) \end{aligned} \tag{3.11.2}$$

Then it is easy to show that for any two scalars β' and β'' near $\bar{\beta}$, and any two scalars α' and α'' near $\bar{\alpha}$ that:

$$\begin{aligned} \left[\frac{1}{\sigma_2^2} \left| K_\alpha(\beta') - K_\alpha(\beta'') \right|^2 \right] &\le \rho \left[\frac{1}{\sigma_1^2} \left| \beta' - \beta'' \right|^2 \right] \\ \left[\frac{1}{\sigma_1^2} \left| K_\beta(\alpha') - K_\beta(\alpha'') \right|^2 \right] &\le \rho \left[\frac{1}{\sigma_2^2} \left| \alpha' - \alpha'' \right|^2 \right] \end{aligned} \tag{3.11.3}$$

so that both the K_α and K_β operators are contraction mappings. (See Ortega and Rheinboldt [5] for an extensive discussion of contraction and non-expansion mappings.) This immediately implies that the iteration must converge at a geometric rate; in fact:

$$\hat{\alpha}_k = -\rho^{2k-1}\frac{\sigma_2}{\sigma_1}\hat{\beta}_0 + \bar{\alpha} \qquad (3.11.4)$$

$$\hat{\beta}_k = \rho^{2k}\hat{\beta}_0 + \bar{\beta}$$

Thus $(\hat{\alpha}_k,\hat{\beta}_k)$ approaches $(\bar{\alpha},\bar{\beta})$ geometrically at rate $\rho^2<1$.

This rate of convergence can be quite slow if the variables $\underline{\alpha}$ and $\underline{\beta}$ are highly correlated (in the scalar case, if $\rho\approx1$). Figure 3.2 shows a typical convergence pattern for the scalar case.

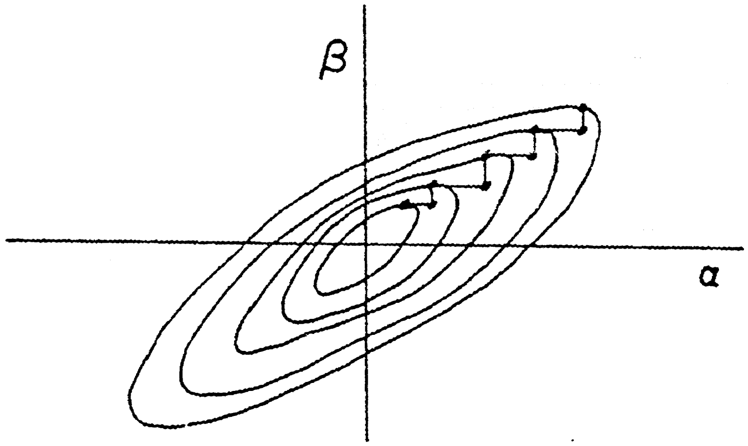


Figure 3.2 - Convergence Pattern of Iterative Algorithm

Note that the estimates zigzag back and forth in an attempt to reach the minimum of $F(\underline{\alpha};\underline{\beta})$. If $\underline{\alpha}$ and $\underline{\beta}$ are highly correlated, then the elliptical contours of $F(\underline{\alpha};\underline{\beta})$ will be very narrow, and each zig and zag can be quite short.

The solution to this problem is to vary the iterative estimation procedure, either to break away from the ridge in the probability density, or else to accelerate the convergence rate along the ridge. This latter approach is quite feasible because, as is obvious from figure 3.2, the estimates $(\hat{\underline{\alpha}}_{k+1},\hat{\underline{\beta}}_{k+1})$ and $(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)$ are generally aligned in the direction of the ridge. If we probe along this line, therefore, we ought to be able to come very close to the global minimum. If the constraint sets are convex, then simple linear extrapolation of the estimates may greatly accelerate the convergence rate:

```
For $k=0,1,\cdots$
    Calculate $\hat{\underline{\alpha}}_{k+1}$
    $\hat{\underline{\alpha}}_{k+1} - \mu\hat{\underline{\alpha}}_{k+1} + (1-\mu)\hat{\underline{\alpha}}_k$        (3.11.5)
    Calculate $\hat{\underline{\beta}}_{k+1}$
    $\hat{\underline{\beta}}_{k+1} - \mu\hat{\underline{\beta}}_{k+1} + (1-\mu)\hat{\underline{\beta}}_k$
```

where μ is a relaxation parameter generally chosen within the range $0<\mu<2$ (see, for example, Dahlquist. [4]) If the constraint sets are convex, values of μ below 1 (under-relaxation) will give new estimates inside the sets Λ and Φ. On the other hand, values of μ greater than 1 (over-relaxation) can extrapolate $\underline{\alpha}$ or $\underline{\beta}$ outside the set Λ or Φ. In this latter case, we will have to project the extrapolated estimates back inside the set Λ or Φ.

A more effective approach, suggested by Hayes and Tom [17,18] is to adaptively modify the relaxation parameter μ in some optimal manner. Their idea can be applied to our model as follows. Rather than use a fixed value of μ to extrapolate along the line connecting $(\hat{\underline{\alpha}}_{k+1},\hat{\underline{\beta}}_{k+1})$ and $(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)$, we will actually search along this line for a minimum of $F(\underline{\alpha};\underline{\beta})$. Experimentation has shown that a particularly effective search procedure is the following, suggested by Carol Espy:

```
Guess $\hat{\underline{\alpha}}_0$, $\hat{\underline{\beta}}_0=\hat{\underline{\beta}}'_0$
For k=0,1,...
    $\hat{\underline{\alpha}}_{k+1} - \min_{\underline{\alpha}\in\Lambda} F(\underline{\alpha}\ ;\hat{\underline{\beta}}'_k)$
    $\hat{\underline{\beta}}_{k+1} - \min_{\underline{\beta}\in\Phi} F(\hat{\underline{\alpha}}_{k+1}\ ;\underline{\beta})$        (3.11.6)
    $(\hat{\underline{\alpha}}'_{k+1},\hat{\underline{\beta}}'_{k+1}) - \min_{\mu} F(\ \mu(\hat{\underline{\alpha}}_{k+1},\hat{\underline{\beta}}_{k+1}) + (1-\mu)(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)\ )$
```

Note that this procedure searches along the line connecting the latest estimate with the last pre-extrapolation estimate $(\hat{\underline{\alpha}}_k,\hat{\underline{\beta}}_k)$. (Searching along the line connecting the latest estimate with the last post-extrapolation estimate $(\hat{\underline{\alpha}}'_k,\hat{\underline{\beta}}'_k)$ does not seem to be as effective.) Of course, it will be necessary to restrict the range of the search values of μ so that the extrapolated values of $\underline{\alpha}$ and $\underline{\beta}$ remain within the constraint sets Λ and Φ.

Often a method such as this can accelerate convergence by a factor of 2 or 3. Whether or not this line search is worth performing, therefore, depends on the relative cost of the line search versus performing another pass or two of the iterative algorithm. "Higher-order" acceleration methods could also be considered in which we search along the q dimensional hyperplane formed from the last $q+1$ estimates of $\underline{\alpha}, \underline{\beta}$:

$$\hat{\underline{\alpha}}_{k+1}, \hat{\underline{\beta}}_{k+1} \leftarrow \max_{\mu_1, \ldots, \mu_q} F(\tilde{\underline{\alpha}} \, ; \tilde{\underline{\beta}})$$

$$\text{where: } (\tilde{\underline{\alpha}} \, ; \tilde{\underline{\beta}}) = \sum_{i=0}^{q} \mu_i (\hat{\underline{\alpha}}_{k-i}, \hat{\underline{\beta}}_{k-i})$$

$$\sum_{i=0}^{q} \mu_i = 1$$

Once again, in considering such methods we must consider the possible improvement in convergence rate versus the cost of this hyperplane search. In some cases, a reasonable compromise is to use a q^{th} order acceleration step after every $q+1$ steps of the usual iteration.

In later chapters we will consider more sophisticated conjugate gradient methods and PARTAN methods for accelerating the convergence of this algorithm. These methods work best when $F(\underline{\alpha};\underline{\beta})$ is quadratic, converging to the global minimum in a finite number of steps. For further details, see chapter 5.

12. Summary

In this chapter, we have developed four iterative algorithms for MCEM, MAP and ML parameter and signal estimation. All reduce the estimation problem to an iterative search for a best separable density approximation to the original density $p(\underline{x}, \underline{\phi})$, possibly constraining one or more fitted densities to be impulse functions. The MCEM algorithm estimates a signal density by averaging over all parameter values, and estimates a parameter density by averaging over all signal values. PARMAP constrains the parameter density to be an impulse function, and thus differs from MCEM in that it only averages over signal values in an attempt to find the best parameter estimate. SIGMAP is the exact opposite, averaging over all parameter values to estimate the signal, while PSMAP performs no averaging at all. Numerous variations of these basic approaches can be devised when the signal model has multiple signals and parameter sets. Convergence of the MAP algorithms to a local minimum or stationary point of the cross-entropy function (and local maximum or stationary point of the likelihood function) can be guaranteed when the probability densities are continuously differentiable and the estimates remain bounded. Existence of limit measures for the MCEM algorithm is guaranteed under more general circumstances, though certain technical assumptions must be added to show that these limits are critical points of the cross-entropy function. If the level sets are convex and the densities can be transformed into convex functions, then convergence of the MAP algorithms can be guaranteed to the global minimum solution. We conjecture that a similar result applies for MCEM. The convergence rate of all these algorithms is approximately linear, and adding line searches or using other related techniques can often accelerate the convergence rate by a factor of 2 or 3.

References

1. C. Radhakrishna Rao, *Advanced Statistical Methods in Biometric Research*, John Wiley & Sons, New York (1952).
2. B.K. Kale, "On the Solution of the Likelihood Equation by Iteration Processes," *Biometrika* 48, pp.452-456 (1961).
3. B.K. Kale, "On the Solution of Likelihood Equations by Iteration Processes. The Multiparametric Case," *Biometrika* 49, pp.479-486 (1962).
4. Germund Dahlquist and Ake Bjorck, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N.J. (1974).
5. J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York (1970).
6. G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden-Day Inc., San Francisco (1976).
7. Bruce R. Musicus, *An Iterative Technique for Maximum Likelihood Parameter Estimation on Noisy Data*, S.M. Thesis, M.I.T., Cambridge, Mass. (Feb 1979).
8. Jae S. Lim and A. V. Oppenheim, "All-Pole Modeling of Degraded Speech," *IEEE Trans. Acoust. Speech, Signal Proc.* ASSP-26(3), pp.197-210 (June 1978).
9. H.O. Hartley, "Maximum Likelihood Estimation from Incomplete Data," *Biometrics* 14, pp.174-194 (June 1958).
10. E.L. Lehmann, *Testing Statistical Hypotheses*, John Wiley & Sons, New York (1959).
11. Thomas S. Ferguson, *Mathematical Statistics: A Decision Theoretic Approach*, Academic Press, New York (1967).
12. Koopman, "On Distributions Admitting a Sufficient Statistic," *Trans. Am. Math. Soc.* 39, pp.399-409 (1936).
13. András Prékopa, "On Logarithmic Concave Measures and Functions," *(Szeged) Acta Sci. Math* 34, pp.335-343 (1973).
14. Pinsker, *Information and Information Stability of Random Variables*, Holden Day, San Francisco (1964). translated by A. Feinstein
15. William Feller, *An Introduction to Probability Theory and Its Applications*, John Wiley & Sons, New York (1966).
16. Patrick Billingsley, *Convergence of Probability Measures*, John Wiley & Sons, New York (1968).
17. Monson H. Hayes and Victor T. Tom, *Adaptive Acceleration of Iterative Signal Recontruction Algorithms*, Technical Note 1980-28, Lincoln Laboratory M.I.T. (to be published).
18. Monson H. Hayes III., *Signal Reconstruction from Phase or Magnitude*, M.I.T. PhD Thesis (June 1981).

Chapter 4
Applications in Statistics

1. Introduction

In chapter 2 we presented four basic MCEM and MAP approaches for estimating multiple signal and parameter unknowns given uncertain observations. In chapter 3 we presented iterative algorithms for solving these approaches, and proved conditions under which they converge. Though not necessarily unbiased like the Minimum Mean Square Error estimates, these MCEM and MAP methods are usually straightforward to compute, particularly for exponential families of distributions, and they often have good asymptotic properties. In the remainder of this thesis, we will consider a variety of applications of these estimation algorithms. In this chapter we study a common problem in statistics in which imperfect observations are used to estimate certain unknown parameters of a probability distribution. We will concentrate particularly on the problem of grouped, truncated, censored and/or quantized observations, though the basic approach can be generalized to an enormous variety of problems.

Suppose we are given the probability density $p(x|\phi)$ of a sample value x given the parameter value ϕ, where all that is known about ϕ is an *a priori* probability density $p(\phi)$ of its possible values. In order to estimate ϕ, we draw N independent samples $x_1, \ldots, x_N$ from the distribution; unfortunately, each measurement is inexact and only indicates that the sample value is within a certain range, $L_i \le x_i \le U_i$. This might occur, for example, if we measured the continuous valued sample with a coarse analog-to-digital converter, or if for convenience in data collection, we simply divided the range of sample values into a few subintervals ("bins") and then counted how many samples fall in each bin.

If we had observed the sample values $x_1, \ldots, x_N$ directly (the classical parameter estimation problem) then we could estimate the parameters by finding the mean or the mode of the density $p(\Phi | x_1, \ldots, x_N)$. If the parameters were known (the classical signal estimation problem) then the actual sample values could have been estimated by finding the mean or mode of the density $p(x_1, \ldots, x_N | \Phi) = \prod_i p(x_i | \Phi)$. Unfortunately, in our problem neither the parameter values nor the sample values are known exactly. Given that the sample x_i is in the range $X_i = [L_i, U_i]$, the most straightforward estimation approach would be to first calculate the marginal density:

$$p(\Phi, X_1, \ldots, X_N) = p(\Phi) \prod_{i=1}^{N} \int_{L_i}^{U_i} p(x_i | \Phi)\, dx_i \qquad (4.1.1)$$

The parameters could then be estimated by calculating the mean (MMSE estimate) or the mode (PARMAP estimate) of the conditional density:

$$p(\Phi | X_1, \ldots, X_N) = \frac{p(\Phi, X_1, \ldots, X_N)}{\int_{\Phi} p(\Phi, X_1, \ldots, X_N)\, d\Phi} \qquad (4.1.2)$$

Needless to say, this may be quite complicated. Maximum Likelihood approaches are also possible; for an exhaustive treatment of these, see Kulldorff. [1]

Fortunately, when $p(x | \Phi)$ forms an exponential family of densities, all four of our iterative algorithms take a particularly simple form. In the next two sections we specifically consider the cases when $p(x | \Phi)$ is Exponential or Gaussian. For these cases, all four algorithms appear quite similar, iterating back and forth between an almost-classical sample estimation step, and an almost-classical parameter estimation step. Strong convergence results can be proven in all cases.

2. Exponential Density

Suppose that p(x |ϕ) and p(ϕ) are both Exponential densities:

$$\begin{array}{lll} p(x \mid \phi) = \phi e^{-x\phi} & & 0 \le x < \infty \\ p(\phi) \;\; = \epsilon_0 e^{-\epsilon_0 \phi} & & 0 \le \phi < \infty \end{array} \tag{4.2.1}$$

(To simulate a Fisher estimation problem, we could choose ϵ_0 very close to zero, thus making the *a priori* density nearly "flat".)

2.1. Classical Estimates

If we were given the exact value of N independent random samples $x_1, \ldots, x_N$ drawn from this distribution, then the classical estimate of the parameter ϕ would be found by forming the *a posteriori* probability density:

$$\begin{aligned} p(\phi \mid x_1, \ldots, x_N) &= \frac{p(x_1, \ldots, x_N \mid \phi)\, p(\phi)}{p(x_1, \ldots, x_N)} \\ &= K(\underline{x}) \left(\phi^N \exp(-\phi \sum_{i=1}^{N} x_i) \right) \left(\epsilon_0 \exp(-\epsilon_0 \phi) \right) \end{aligned} \tag{4.2.2}$$

and then calculating its mean or mode:

<u>Classical Parameter Estimate:</u>

$$E\left[\phi \,\middle|\, x_1, \ldots, x_N \right] = \frac{N+1}{\epsilon_0 + \sum_{i=1}^{N} x_i} \tag{4.2.3a}$$

or:

$$\max_{\phi} p(\phi \mid x_1, \ldots, x_N) = \frac{N}{\epsilon_0 + \sum_{i=1}^{N} x_i} \tag{4.2.3b}$$

Conversely, if the parameter value was known exactly, but the samples were not, then we could estimate the samples by finding the mean or mode of the *a posteriori* density:

$$p_{X_i}(x_i \mid \phi) = \begin{cases} c_\phi e^{-\phi x_i} & \text{for } L_i \le x_i \le U_i \\ 0 & \text{else} \end{cases} \tag{4.2.4}$$

where c_ϕ is a normalization constant. Thus:

Classical Signal Estimate:

$$E_{X_i}\left[x_i \,\middle|\, \phi\right] = L_i + \frac{1}{\phi}\left[1 - \frac{\delta_i e^{-\delta_i}}{(1-e^{-\delta_i})}\right] \qquad (4.2.5a)$$

where: $\delta_i = \phi(U_i - L_i)$

or:

$$\max_{L_i \le x_i \le U_i} p(x_i \,|\, \phi) = L_i \qquad (4.2.5b)$$

In both these classical problems, the mean is usually a better estimate than the mode.

2.2. MMSE

If both the parameters are unknown and the sample data is only known to be within certain ranges, the "best" estimation procedure is to first calculate the marginal densities:

$$p(X\,,\phi) = p(\phi) \prod_{i=1}^{N} \int_{L_i}^{U_i} p(x_i \,|\, \phi)\, dx_i$$

$$= \epsilon_0 \exp(-\epsilon_0 \phi) \prod_{i=1}^{N} \left[\exp(-L_i \phi) - \exp(-U_i \phi) \right] \qquad (4.2.6)$$

and:

$$p(x\,,\Phi) = \frac{N!}{\left(\epsilon_0 + \sum_{i=1}^{N} x_i\right)^{N+1}} \qquad (4.2.7)$$

With a large amount of effort, it is possible to calculate the means of these densities (the MMSE estimates.) Suppose there are M different bins $[L_i, U_i]$ for $i = 1, \ldots, M$

and n_i samples x_i in each bin, $\sum_{i=1}^{M} n_i = N$. Then the MMSE parameter estimate can be written in the form:

$$\frac{\displaystyle\sum_{k_1=1}^{n_1} \cdots \sum_{k_M=1}^{n_M} \left[\prod_{i=1}^{M} (-1)^{k_i} \frac{n_i!}{(n_i-k_i)!k_i!} \right] \frac{1}{\left(\epsilon_0 + \sum_{i=1}^{M} (n_i-k_i)L_i + k_i U_i \right)^2}}{\displaystyle\sum_{k_1=1}^{n_1} \cdots \sum_{k_M=1}^{n_M} \left[\prod_{i=1}^{M} (-1)^{k_i} \frac{n_i!}{(n_i-k_i)!k_i!} \right] \frac{1}{\left(\epsilon_0 + \sum_{i=1}^{M} (n_i-k_i)L_i + k_i U_i \right)}} \qquad (4.2.8)$$

Unfortunately, this formula is not only messy to compute, but is also numerically ill-behaved. A similar formula for the MMSE sample estimates can be given, but it involves logarithms and is even messier and even worse behaved. Furthermore, if any $U_i = \infty$ then the expectation of x_i happens to be infinite.

Fortunately, our four iterative MCEM and MAP algorithms take a much simpler and more robust form in this problem. The reason is that the Exponential density forms a family of exponential densities:

$$\begin{aligned} \mathrm{p}(x_1, \ldots, x_N, \phi) &= \left[\epsilon_0 \phi^N \right] \exp \left[\phi \, (-\epsilon_0 - \sum_{i=1}^{N} x_i) \right] \\ &= h(\phi) \exp \left[\pi_1(\phi) t_1(\underline{x}) \right] \end{aligned} \qquad (4.2.9)$$

where $h(\phi)$, $\pi(\phi)$ and $t(\underline{x})$ are defined in an obvious way.

2.3. MCEM

Substituting formula (4.2.9) into the MCEM algorithm of chapter 3 shows that the estimated signal and parameter densities q_X and q_Φ will be truncated Exponential and Gamma densities respectively:

$$\hat{q}_{X_{k+1}}(x_1, \ldots, x_N) = \prod_{i=1}^{N} \hat{q}_{X_{i,k+1}}(x_i) \tag{4.2.10}$$

$$\text{where: } \hat{q}_{X_{i,k+1}}(x_i) = \begin{cases} c_{x_i} \exp(-\hat{\phi}_k x_i) & \text{for } L_i \le x_i \le U_i \\ 0 & \text{else} \end{cases}$$

and:

$$\hat{q}_{\Phi_{k+1}}(\phi) = c_{\Phi} \phi^N \exp\left[-\phi\, (\epsilon_0 + \sum_{i=1}^{N} \hat{x}_{i,k+1}) \right] \tag{4.2.11}$$

where c_{x_i} and c_{Φ} are normalization constants, and $\hat{\phi}_k$ and $\hat{x}_{i,k-1}$ are conditional expectations:

$$\hat{\phi}_k = \mathrm{E}_{\Phi}\left[\phi \,\middle|\, \hat{q}_{\Phi_k} \right] = \int_0^{\infty} \phi\, \hat{q}_{\Phi_k}(\phi)\, d\phi = \frac{N+1}{\epsilon_0 + \sum_{i=1}^{N} \hat{x}_{i,k}} \tag{4.2.12}$$

and:

$$\hat{x}_{i,k+1} = \mathrm{E}_{X_i}\left[x_i \,\middle|\, \hat{q}_{X_{i,k+1}} \right] = \int_{L_i}^{U_i} x_i\, \hat{q}_{X_{i,k+1}}(x_i)\, dx_i$$

$$= L_i + \frac{1}{\hat{\phi}_k}\left[1 - \frac{\delta_i e^{-\delta_i}}{1 - e^{-\delta_i}} \right] \tag{4.2.13}$$

In fact, from the last formulas, it is clear that explicitly calculating the densities $\hat{q}_{X_k}$ and $\hat{q}_{\Phi_k}$ is unnecessary since we can directly calculate the conditional expectations $\hat{\phi}_k$ and $\hat{x}_k$ as follows:

```
MCEM Iterative Algorithm:
Guess $\hat{x}_{i,0}$
For $k=0,1,\cdots$
    $\hat{\phi}_{k+1} = \frac{N+1}{\epsilon_0 + \sum_{i=1}^{N} \hat{x}_{i,k}}$ .
    $\hat{x}_{i,k+1} = L_i + \frac{1}{\hat{\phi}_{k+1}} \left[ 1 - \frac{\delta_i e^{-\delta_i}}{1-e^{-\delta_i}} \right]$
        where: $\delta_i = \hat{\phi}_{k+1}(U_i - L_i)$
```

We start by guessing the initial sample values $\hat{x}_{i,0}$ to be somewhere in the middle of their known range. The parameter is then estimated exactly as if we were calculating its mean value given the sample values $\hat{x}_{i,k}$. (Compare with the classical estimate (4.2.3a).) The samples are then reestimated exactly as if we were calculating their mean values given the parameter value $\hat{\phi}_k$. (Compare with the classical estimate (4.2.5a).) The algorithm then iterates, using the improved sample estimates to improve the next parameter estimate. Each iteration decreases the cross-entropy, and Appendix C proves that the algorithm converges to the unique solution to the MCEM problem. Furthermore, the convergence rate is geometric in the sense that:

$$\left| \frac{1}{\hat{\phi}_{k+1}} - \frac{1}{\hat{\phi}_k} \right| \le \frac{N}{N+1} \left| \frac{1}{\hat{\phi}_k} - \frac{1}{\hat{\phi}_{k-1}} \right| \qquad \text{for all } k \tag{4.2.14}$$

Appendix C also shows that:

$$H(\hat{q}_{X_{k+1}}, \hat{q}_{\Phi_k}) = K - \log \left[\hat{\phi}_k \, p(X, \hat{\phi}_k) \right] \tag{4.2.15}$$

where K is a constant. MCEM thus increases $\hat{\phi}_k \, p(X, \hat{\phi}_k)$ on each iteration, and Appendix C shows that the limit point ϕ_* will also be the global maximum of $\phi \, p(X, \phi)$. This formula also implies that if PARMAP is asymptotically consistent with

$p(X,\phi)$ approaching an impulse at the true value of ϕ as $N \to \infty$, then MCEM will also be asymptotically consistent.

2.4. PARMAP

The PARMAP algorithm is virtually identical to the MCEM algorithm except that we constrain the parameter density to be an impulse function $\delta(\phi-\hat{\phi})$. The resulting sample density estimate will have the same form as in the MCEM algorithm:

$$\hat{q}_{X_k}(x_1, \ldots, x_N) = \prod_{i=1}^{N} \hat{q}_{X_{i,k}}(x_i) \tag{4.2.16}$$

$$\text{where: } \hat{q}_{X_{i,k}}(x_i) = \begin{cases} c_i \exp(-\hat{\phi}_k x_i) & \text{for } L_i \le x_i \le U_i \\ 0 & \text{else} \end{cases}$$

where the parameter value $\hat{\phi}_k$ is determined by:

$$\hat{\phi}_k - \max_{\phi \ge 0} \phi^N \exp\left[-\phi \left(\epsilon_0 + \sum_{i=1}^{N} \hat{x}_{i,k-1} \right) \right] \tag{4.2.17}$$

$$\text{where: } \hat{x}_{i,k-1} = \mathrm{E}_{X_i}\left[x_i \,\middle|\, \hat{q}_{X_{i,k-1}}(x_i) \right]$$

Solving (4.2.17) gives the algorithm:

```
PARMAP Iterative Algorithm:
Guess $x_i$
For $k = 0, 1, \cdots$
    $\hat{\phi}_{k+1} = \dfrac{N}{\epsilon_0 + \sum_{i=1}^{N} \hat{x}_{i,k}}$
    $\hat{x}_{i,k+1} = L_i + \dfrac{1}{\hat{\phi}_{k+1}} \left[ 1 - \dfrac{\delta_i e^{-\delta_i}}{1 - e^{-\delta_i}} \right]$
        where: $\delta_i = \hat{\phi}_{k+1}(U_i - L_i)$
```

PARMAP thus iterates between calculating the mode of the parameter value as if the sample values were actually $\hat{x}_{i,k}$ (compare with (4.2.3b)) and then calculating the conditional mean of the sample value $\hat{x}_{i,k+1}$ as if the parameter value were actually $\hat{\phi}_k$. Note that, as might be expected from the relationship in (4.2.15), the only difference between MCEM and PARMAP is that the formula for $\hat{\phi}_{k+1}$ in PARMAP has a factor of N, while the MCEM formula has a factor of $N+1$. Each iteration increases the likelihood function $\log p(X_1, \ldots, X_N, \phi)$ and Appendix C proves that as long as at least one U_i is finite, then the iteration converges to the unique solution to the PARMAP problem. Furthermore, the convergence rate is roughly linear, satisfying:

$$\left| \frac{1}{\hat{\phi}_{k+1}} - \frac{1}{\hat{\phi}_k} \right| < \left| \frac{1}{\hat{\phi}_k} - \frac{1}{\hat{\phi}_{k-1}} \right| \tag{4.2.18}$$

If all the U_i were infinite, then the PARMAP solution would be $\hat{\phi}=0$, and the parameter estimates would converge to this estimate at a sublinear rate (see Appendix C).

2.5. SIGMAP

In the SIGMAP algorithm, we constrain the signal density to be an impulse function $\hat{q}_{X_k}(x_1, \ldots, x_N) = \delta(x_1-\hat{x}_1) \cdots \delta(x_N-\hat{x}_N)$. The parameter density has the same form as in the MCEM algorithm:

$$\hat{q}_{\Phi_{k+1}}(\phi) = c_\phi \phi^N \exp\left[-\phi \left(\epsilon_0 + \sum_{i=1}^{N} \hat{x}_{i,k+1} \right) \right] \tag{4.2.19}$$

where the signal estimate $\hat{x}_{i,k}$ is found by solving:

$$\hat{x}_{i,k+1} - \max_{L_i \le x_i \le U_i} c_{x_i} \exp\left(-\hat{\phi}_k x_i \right) \tag{4.2.20}$$

$$\text{where: } \hat{\phi}_k = E_\Phi \left[\phi \,\middle|\, \hat{q}_{\Phi_k} \right]$$

Notice, however, that the solution to (4.2.20) will always be $\hat{x}_{i,k+1}=L_i$, and thus the SIGMAP problem reduces to a single pass:

SIGMAP Algorithm:

$$\hat{x}_i = L_i$$

$$\hat{\phi} = \frac{N+1}{\epsilon_0 + \sum_{i=1}^{N} \hat{x}_i}$$

The mode of the signal is calculated as if the parameter value were known - this always gives the same estimate L_i. Then the mean of the parameters is calculated as if the signal value were known to be $\hat{x}_i$. (Compare with (4.2.5b) and (4.2.3a).) No iteration is necessary. Note, however, that the signal estimate significantly underestimates the correct signal value, and thus we would expect the SIGMAP parameter estimate to be strongly biased toward large values of $\hat{\phi}_k$. Finally, note that this solution could have been derived directly by recognizing that the SIGMAP marginal density:

$$p(x,\Phi) = \frac{N!}{\left(\epsilon_0 + \sum_{i=1}^{N} x_i\right)^{N+1}} \tag{4.2.21}$$

is maximized at $x_i = L_i$.

2.6. PSMAP

The PSMAP algorithm alternates between a maximization over the signal space and a maximization over the parameter space. Substituting (4.2.9) into the PSMAP algorithm gives:

$$\hat{x}_{i,k+1} = \max_{L_i \le x_i \le U_i} c_{x_i} \exp\left(-\hat{\phi}_k x_i\right) \tag{4.2.22a}$$

$$\hat{\phi}_{k+1} = \max_{\phi \ge 0} c_\phi \phi^N \exp\left[-\phi\left(\epsilon_0 + \sum_{i=1}^{N} \hat{x}_{i,k+1}\right)\right] \tag{4.2.22b}$$

The solution for $\hat{x}_{i,k+1}$, however, is always $\hat{x}_{i,k+1}=L_i$. Thus no iteration is necessary, and the solution to PSMAP will be:

PSMAP Algorithm:

$$\hat{x}_i = L_i$$

$$\hat{\phi} = \frac{N}{\epsilon_0 + \sum_{i=1}^{N} \hat{x}_i}$$

The signal estimate is calculated as if we were finding its mode given the parameter value $\hat{\phi}$. The parameter value is then reestimated as if we were finding its mode given the signal value $\hat{x}_i$. Note that the signal estimate $\hat{x}_i$ significantly underestimates the actual value of x_i, and thus the parameter estimate will be strongly biased toward large values of ϕ.

2.7. Maximum Likelihood Applications

In some cases, there is no justification for treating the parameter ϕ as a Bayesian random variable. This difficulty can be handled by treating ϕ as a Fisher non-random constant, as in chapter 2 section 8, and deriving the PARML and PSML algorithms for this example. Fortunately, in this example, an exactly equivalent but simpler approach would be to make the *a priori* density "flat" by choosing $\epsilon_0=0$. No other changes are necessary to the iterative algorithms themselves, although the convergence results must be modified slightly. MCEM still converges at a geometric rate to the unique solution; the problem is that if all $L_i=0$ then $\hat{\phi}_{k+1} \geq \frac{N+1}{N} \hat{\phi}_k$ and $\hat{\phi}_k \to \infty$ as $k \to \infty$ ($\hat{\phi}=\infty$ is the global minimum of the cross-entropy.) PARML is only guaranteed to converge to the unique solution of $\max_{\phi} p(X \mid \phi)$ if at least one value U_i is finite and at least one value L_i is non-zero. Three special cases must be recognized for PARML when $\epsilon_0=0$:

a) All $U_i = \infty$, but at least one L_i is non-zero:

- then $\hat{\phi}_k \to 0$, $\hat{x}_{i,k} \to \infty$ as $k \to \infty$

b) All $L_i = 0$ but at least one U_i is finite:

- then $\hat{\phi}_k \to \infty$, $\hat{x}_{i,k} \to 0$ as $k \to \infty$

c) All $L_i = 0$, all $U_i = \infty$

- then $\hat{\phi}_k = \hat{\phi}_0$ and all values $\hat{\phi} \geq 0$ are global maxima of $p(X \mid \phi)$

This last case is particularly silly, since it implies that nothing whatsoever is known about the value of any of the samples. Finally, SIGMAP and PSMAP will have solutions if $\epsilon_0 = 0$, but if all the $L_i = 0$ then $\hat{\phi} = \infty$.

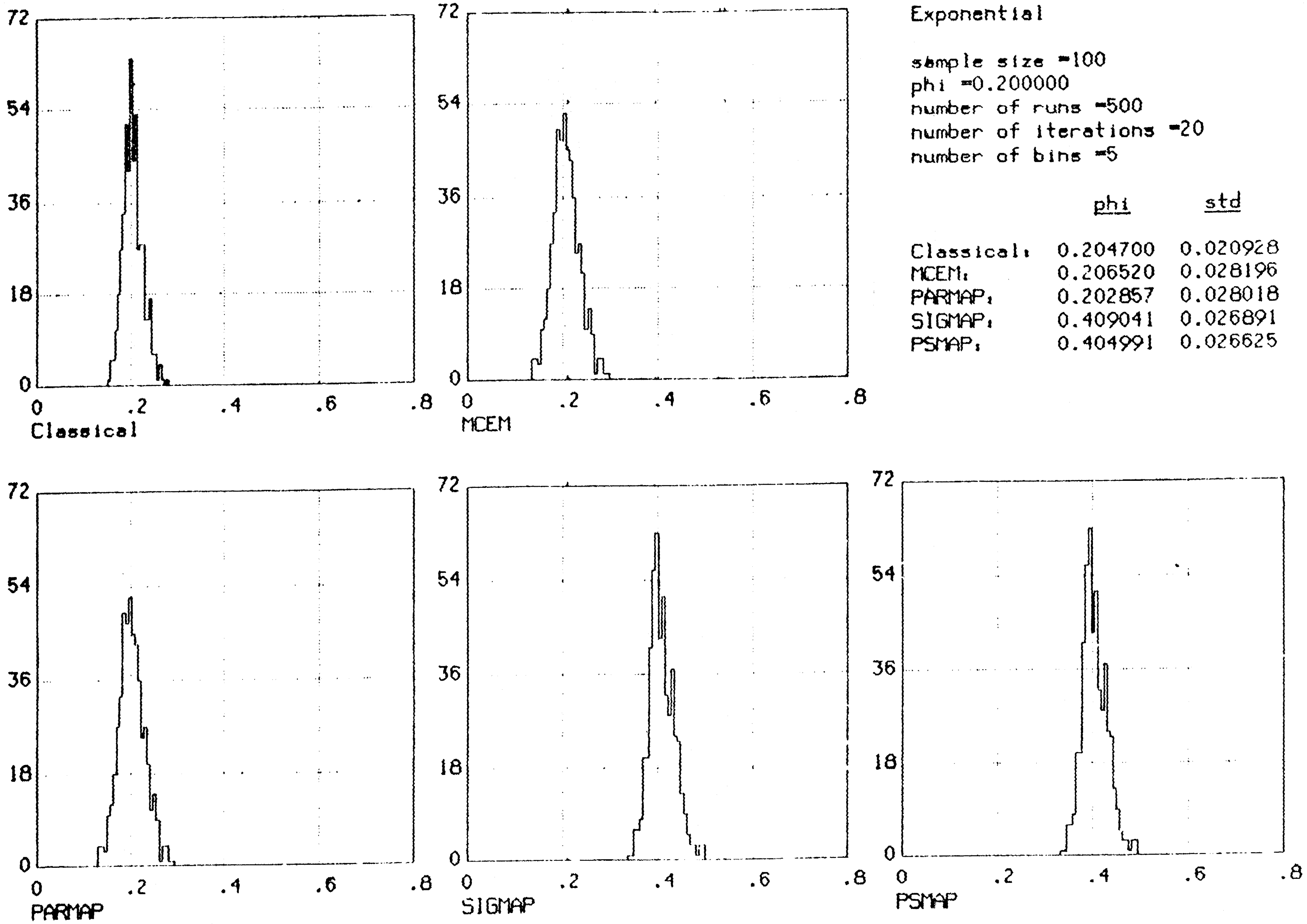
2.8. Comparison of the Algorithms

To compare the algorithms, we consider a specific example. We start with an Exponential density with parameter $\phi = 0.2$, and choose a nearly flat *a priori* density with $\epsilon = 10^{-4}$. We draw N independent random samples x_i, and group them into 5 bins, $0 \leq x_i < 1$, $1 \leq x_i < 2$, $2 \leq x_i < 3$, $3 \leq x_i < 4$ and $4 \leq x_i$. (Note that this is a rather lopsided selection of bins since the average value of x_i will be about 5, and thus over half the samples will fall in the last bin.) Given the count of how many samples are in each bin, we apply each of our algorithms to estimate the parameters of the density. Figures 4.1 and 4.2 show histograms of the parameter estimates generated by our algorithms for 500 sequences of $N = 10$ and $N = 100$ samples each. Table 4.1 below summarizes the average value of $\hat{\phi}$ and its standard deviation for each method:

	$\hat{\phi}, N=10$	$\hat{\phi}, N=100$
Classical:	0.245 ± 0.087	0.205 ± 0.021
MMSE:	0.254 ± 0.104	?????
MCEM:	0.254 ± 0.104	0.207 ± 0.028
PARMAP:	0.216 ± 0.097	0.203 ± 0.028
SIGMAP:	0.467 ± 0.116	0.409 ± 0.027
PSMAP:	0.424 ± 0.105	0:405 ± 0.027

Table 4.1 - Average values of $\hat{\phi}$ for 500 sequences
(true value = 0.2)

The classical parameter estimate, using the actual values of x_i is the best, of course, since grouping the data into bins can only increase our uncertainty about the parameter. MMSE and MCEM using the grouped data comes very close to the classical estimate, giving virtually identical parameter estimates centered around the same value. (For $N=10$, MCEM and MMSE seem to agree to about 3 decimal places.) This is somewhat surprising, since the MMSE estimate (4.2.8) appears quite different from the iteratively calculated MCEM estimate, and is not only much more difficult to compute, but is also numerically ill-behaved. In fact, even with 64 bits of precision, the MMSE estimate (4.2.8) can not be computed reliably for $N>20$. (Note we were unable to compute the MMSE estimate for $N=100$.) MCEM, on the other hand, is simple to compute, numerically robust, and converges at a linear rate, cutting the error by about half on each iteration. PARMAP gives estimates which are somewhat smaller than MCEM, MMSE and the classical estimates, though they are still close. Convergence is at about the same rate as MCEM. (The fact that PARMAP actually comes closest to the true value 0.2 is misleading; the standard for comparison must be the classical estimate, which represents the optimal estimate of $\hat{\phi}$ if the data were uncorrupted by grouping.) SIGMAP and PSMAP are both heavily biased, and in this example are off



```
Exponential

sample size =100
phi =0.200000
number of runs =500
number of iterations =20
number of bins =5

                  phi        std

Classical:     0.204700   0.020928
MCEM:          0.206520   0.028196
PARMAP:        0.202857   0.028018
SIGMAP:        0.409041   0.026891
PSMAP:         0.404991   0.026625
```

Figure 4.2 - Histograms of $\hat{\phi}$ for $N=100$, Exponential Density

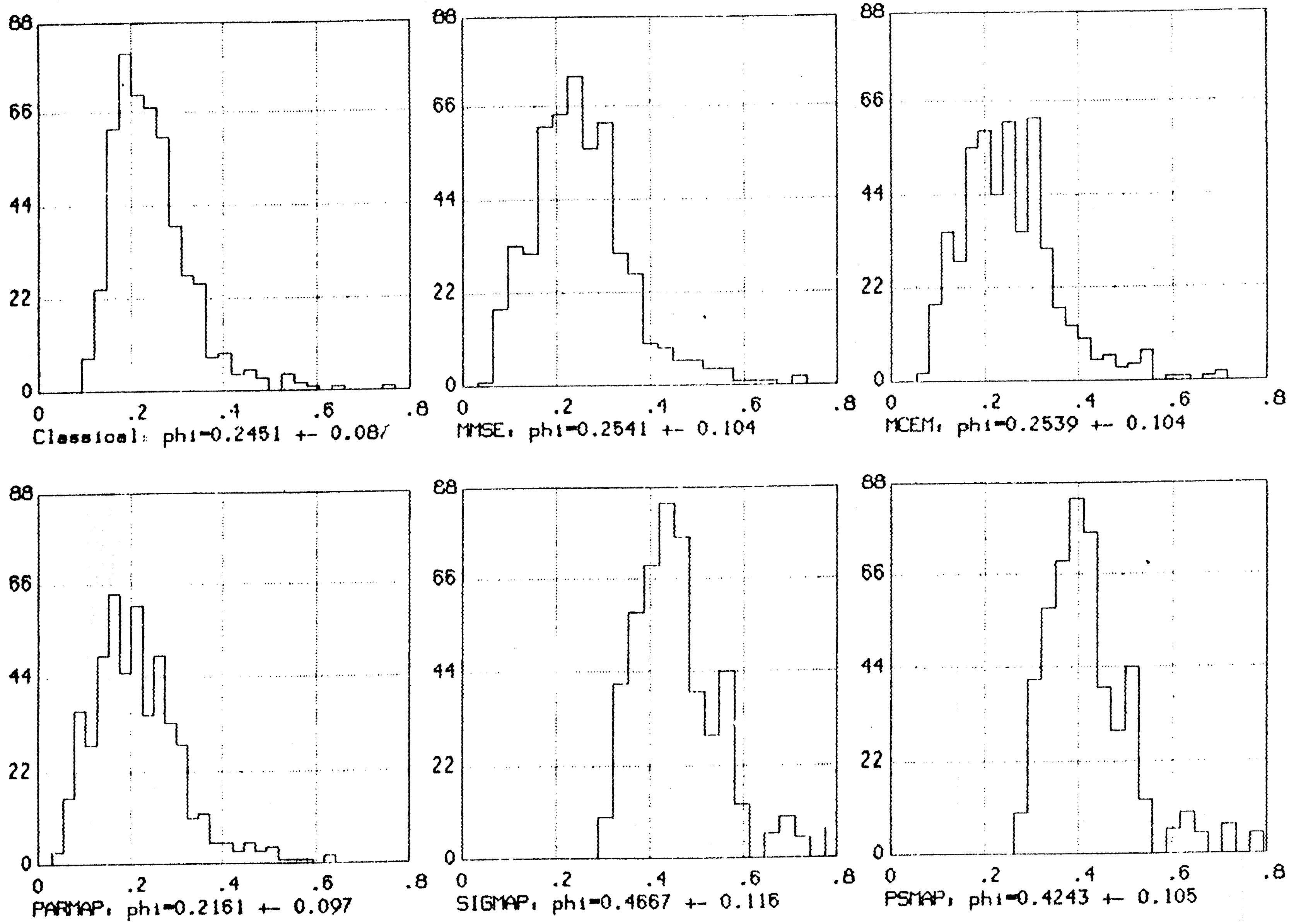


Figure 4.1 - Histograms of $\hat{\phi}$ for $N = 10$, Exponential Density

by nearly a factor of 2 (Choosing a less lopsided selection of bins would decrease the bias of these methods.)

Examining the histograms of the parameter estimates for $N=100$, the classical method, MCEM and PARMAP all give estimates which cluster tightly about the true value $\hat{\phi}=0.2$. This indicates that both MCEM and PARMAP using grouped data are asymptotically consistent. (Appendix C argues that MCEM will be asymptotically consistent whenever PARMAP is consistent. For a detailed discussion of consistency with grouped data, see Kulldorff [1] .) SIGMAP and PSMAP cluster tightly around a heavily biased value of 0.4; these methods are definitely not asymptotically consistent.

To summarize, therefore, given grouped data the MMSE estimate is almost as good as the estimates we could calculate when there are no uncertainties in the measured values. Unfortunately, even though an analytic expression for this estimate exists, it is difficult or impossible to compute reliably. MCEM, PARMAP, SIGMAP, and PSMAP, on the other hand, are numerically robust and quite easy to compute, since they simply alternate between calculating the mean or mode of the parameters $\hat{\phi}_k$, and the mean or mode of the samples $\hat{x}_{ik}$. MCEM appears to perform exactly as well as MMSE, while PARMAP gives estimates which are somewhat low. Both methods appear to be asymptotically consistent, and both converge relatively quickly. Faster convergence could be achieved by using Aitken extrapolation or a related technique. [2] SIGMAP and PSMAP are even easier to compute than MCEM or PARMAP, since no iteration is required; however, they are both strongly biased toward large values of ϕ.

3. Gaussian Density

Very similar conclusions can be drawn about the relative performance of our algorithms when we consider the same problem but with a Gaussian distribution instead of an Exponential density. Suppose that the probability density $p(\mathbf{x} \mid \Phi)$ is Gaussian, $N(\mu,\sigma^2)$, with unknown mean μ and variance σ^2:

$$p(x_i \mid \mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x_i-\mu)^2 \right) \tag{4.3.1}$$

We consider the case when the variance σ^2 is known in section 3.8. When σ^2 is unknown, it is convenient to take the parameters of the sample distribution to be the mean, μ, and the inverse variance $s=\frac{1}{\sigma^2}$. Let us assume an *a priori* density $p(\mu,s)=p(\mu \mid s)p(s)$ in which $p(\mu \mid s)$ is a Gaussian with zero mean and variance $\frac{1}{\epsilon s}=\frac{\sigma^2}{\epsilon}$, and $p(s)$ is Exponential:

$$p(\mu \mid s) = \frac{1}{\sqrt{2\pi/\epsilon s}} \exp\left(-\tfrac{1}{2}\epsilon s \mu^2 \right) \tag{4.3.2}$$

$$p(s) \;= \eta \exp(-\eta s) \qquad \text{for } s \geq 0$$

By letting $\epsilon,\eta \to 0$, we can simulate arbitrarily "flat" *a priori* densities. Given only the information that N independent samples x_i were drawn from this density and that they were in the ranges $L_i \leq x_i \leq U_i$, we wish to estimate $\hat{\mu}$, $\hat{s}$, and the actual values of $\hat{x}_i$.

3.1. Classical Approach

If the samples were known exactly, then the unknown parameters could be estimated by finding the mean of the density $p(\mu,s \mid \mathbf{x})$ over the space $\Phi = \{(\mu,s) \mid s \geq 0\}$:

$$\hat{\mu} = E_{\Phi}[\,\mu \mid \mathbf{x}\,] = \frac{1}{N+\epsilon} \sum_{i=1}^{N} x_i \tag{4.3.3}$$

$$\hat{s} = \frac{1}{\hat{\sigma}^2} = \mathrm{E}_{\Phi}[s \mid \underline{x}\,] = \frac{N+2}{2\eta + \epsilon\hat{\mu}^2 + \sum\limits_{i=1}^{N} (x_i - \hat{\mu})^2}$$

The mean is estimated by averaging the samples, and the variance is estimated by computing the sample variance. Slight corrections for the *a priori* density are included in these formulas.

Conversely, if the mean and variance were known, then we could estimate the samples by calculating their expectation:

$$\hat{x}_i = \mathrm{E}_{X_i}\left[x_i \,\middle|\, \mu, s\right] = \mu - \sigma\left[\frac{\exp\left(-\tfrac{1}{2}\tilde{U}_i^2\right) - \exp\left(-\tfrac{1}{2}\tilde{L}_i^2\right)}{\mathrm{erf}(\tilde{U}_i) - \mathrm{erf}(\tilde{L}_i)}\right] \tag{4.3.4}$$

$$\text{where: } \mathrm{erf}(y) = \frac{1}{\sqrt{2\pi}} \int_0^y \exp(-\tfrac{1}{2}x^2)\, dx$$

$$\tilde{L}_i = \frac{L_i - \mu}{\sigma}$$

$$\tilde{U}_i = \frac{U_i - \mu}{\sigma}$$

A much simpler, though less satisfactory estimate would be the sample mode:

$$\hat{x}_i = \max_{L_i \le x_i \le U_i} \mathrm{p}(x_i \mid \mu, s) = \begin{cases} L_i & \text{if } \mu \le L_i \\ \mu & \text{if } L_i \le \mu \le U_i \\ U_i & \text{if } U_i \le \mu \end{cases} \tag{4.3.5}$$

It is also sometimes convenient to know the variance of the sample:

$$\mathrm{Var}_{X_i}[x_i \mid \mu, s\,] = \sigma^2 \left[1 - \frac{\tilde{U}_i \exp(-\tfrac{1}{2}\tilde{U}_i^2) - \tilde{L}_i \exp(-\tfrac{1}{2}\tilde{L}_i^2)}{\mathrm{erf}(\tilde{U}_i) - \mathrm{erf}(\tilde{L}_i)}\right] - (\hat{x}_i - \mu)^2 \tag{4.3.6}$$

These formulas represent a lower limit on the complexity of any estimation routine which must deal with the far more complicated situation where both the parameters and the samples are uncertain.

3.2. MMSE

When both the parameters μ and s are unknown and the measurements x_i are imprecise, the logical approach would be to compute the marginal densities:

$$p(X,\mu,s) = p(\mu,s) \prod_{i=1}^{N} \int_{L_i}^{U_i} p(x_i \mid \mu,s)\, dx_i$$

$$= p(\mu,s) \prod_{i=1}^{N} \left[\operatorname{erf}\left(\frac{U_i-\mu}{\sigma} \right) - \operatorname{erf}\left(\frac{L_i-\mu}{\sigma} \right) \right] \qquad (4.3.7)$$

and:

$$p(x,\Phi) = \int_0^{\infty} ds \int_{-\infty}^{\infty} d\mu\, p(x,\mu,s)\, d\mu\, ds$$

$$= K \int_0^{\infty} ds\, s^{N/2} \exp\left[-s \left(\eta + \tfrac{1}{2} x^T \Sigma^{-1} x \right) \right]$$

$$= \frac{K'}{\left(\eta + \tfrac{1}{2} x^T \Sigma^{-1} x \right)^{N/2+1}} \qquad (4.3.8)$$

$$\text{where: } \Sigma^{-1} = I - \frac{1}{N+\epsilon} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} (1 \cdots 1)$$

and then we could calculate the mean (MMSE estimate) or mode (PARMAP or SIGMAP) of these densities. Clearly either of these approaches would be rather difficult.

Fortunately, all four of our estimation algorithms take a relatively simple form for this problem. The reason is that $p(x,\mu,s)$ forms an exponential family of densities:

$$p(x,\mu,s) = \left[\sqrt{\epsilon}\eta \left(\frac{s}{2\pi} \right)^{\frac{N+1}{2}} \exp\left[-s \left(\eta + \tfrac{1}{2}(N+\epsilon)\mu^2 \right) \right] \right]$$

$$\cdot \exp\left[-\tfrac{1}{2}s \sum_{i=1}^{N} x_i^2 + s\mu \sum_{i=1}^{N} x_i \right]$$

$$= h(\mu,s) \exp\left(\pi_1(\mu,s) t_1(x) + \pi_2(\mu,s) t_2(x) \right) \tag{4.3.9}$$

where these terms are defined in an obvious way. Note carefully that the "natural" parameters of the density are $\pi_1(\mu,s) = \frac{1}{\sigma^2}$ and $\pi_2(\mu,s) = \frac{\mu}{\sigma^2}$; we will return to this point later.

3.3. MCEM

By substituting the density (4.3.9) into the MCEM algorithm, we can show that MCEM will generate a sequence of truncated Gaussian sample densities of the form:

$$\hat{q}_X(x) = \prod_{i=1}^{N} \hat{q}_{X_i}(x_i) \tag{4.3.10}$$

$$\text{where: } \hat{q}_{X_i}(x_i) = \begin{cases} c_{x_i} \exp\left(-\tfrac{1}{2}\hat{s}\,(x_i - \hat{\mu})^2 \right) & \text{for } L_i \le x_i \le U_i \\ 0 & \text{else} \end{cases}$$

and parameter densities $\hat{q}_\Phi(\mu,s) = \hat{q}_\Phi(\mu|s)\hat{q}_\Phi(s)$ which are the product of a Gaussian density $\hat{q}_\Phi(\mu|s)$ and a Gamma density $\hat{q}_\Phi(s)$:

$$\hat{q}_\Phi(\mu|s) = \left(\frac{s(N+\epsilon)}{2\pi} \right)^{1/2} \exp\left(-\frac{s(N+\epsilon)}{2} \left(\mu - \frac{1}{N+\epsilon} \sum_{i=1}^{N} \hat{x}_i \right)^2 \right) \tag{4.3.11}$$

$$\hat{q}_\Phi(s) = \frac{\hat{V}^{N/2+1}}{(N/2)!} s^{N/2} \exp\left(-\hat{V}s \right)$$

The coefficients $\hat{\mu}$, $\hat{s}$, $\hat{x}_i$, $\hat{V}$ of these densities can be iteratively calculated as follows:

```
MCEM Iterative Algorithm:
Guess $\hat{\mu}_0$, $\hat{s}_0 = \frac{1}{\hat{\sigma}_0^2}$
For $k = 0, 1, \cdots$
    $\hat{x}_{i,k+1} = \mathrm{E}_{X_i}\left[ x_i \mid \hat{\mu}_k, \hat{s}_k \right]$
    $\hat{V}_{i,k+1} = \mathrm{Var}_{X_i}\left[ x_i \mid \hat{\mu}_k, \hat{s}_k \right]$
    $\hat{\mu}_{k+1} = \frac{1}{N+\epsilon} \sum_{i=1}^{N} \hat{x}_{i,k+1}$
    $\hat{\sigma}_{k+1}^2 = \frac{1}{\hat{s}_{k+1}} = \frac{\hat{V}_{k+1}}{\frac{1}{2}N+1} = \frac{2\eta + \epsilon\hat{\mu}_{k+1}^2 + \sum_{i=1}^{N} (\hat{x}_{i,k+1} - \hat{\mu}_{k+1})^2 + \hat{V}_{i,k+1}}{N+2}$
```

We start by guessing estimates of the sample mean and variance. The samples are then estimated by calculating their expected value $\hat{x}_{i,k+1}$ given $\hat{\mu}_k$ and $\hat{\sigma}_k^2$ using formula (4.3.4). The variance of this estimate $\hat{V}_{i,k+1}$ is also calculated using formula (4.3.6). We now reestimate the sample mean $\hat{\mu}_{k+1}$ by averaging the individual sample estimates. The variance $\hat{\sigma}_{k+1}^2$ is reestimated by combining the variance of the sample means about the distribution mean, $\sum(\hat{x}_i - \hat{\mu})^2$, together with the estimated variance $\hat{V}_{i,k+1}$ of each sample x_i about the sample mean $\hat{x}_i$. Additional small correction terms which account for the *a priori* density are also included. We then iterate, using the improved mean and variance estimates to improve our estimates of the samples. Each iteration decreases the cross-entropy, and Appendix C proves that the estimates remain bounded and thus converge to the set of stationary points of the algorithm and critical points of the cross-entropy function. Appendix C also shows that:

$$H(\hat{q}_{X_{k+1}}, \hat{q}_{\Phi_k}) = K - \log\left[\hat{s}_k^{*} \mathrm{p}(X, \hat{\mu}_k, \hat{s}_k) \right] \tag{4.3.12}$$

where K is a constant. As shown in Appendix C, each MCEM iteration thus also increases $\hat{s}_k^{*}\mathrm{p}(X, \hat{\mu}_k, \hat{s}_k)$, and the limit points of the algorithm will also be critical points of s p(X ,μ,s). This implies that MCEM will give parameter estimates which are similar to the PARMAP estimates found by maximizing p(X ,μ,s), except that MCEM will give larger estimates of $\hat{s}$, and thus smaller estimates of the variance $\hat{\sigma}^2=\frac{1}{\hat{s}}$. If PARMAP is asymptotically consistent, with p(X ,μ,s) approaching an impulse at the actual parameter value as $N \to \infty$, then (4.3.12) suggests that MCEM will also be asymptotically consistent as $N \to \infty$. Lastly, we conjecture that both MCEM and PARMAP have a unique global optimum and critical point, and that the algorithm converges to this solution; unfortunately, we have not been able to prove this. (The proof of convergence in Appendix C only guarantees convergence to the *set* of critical points of the cross-entropy.)

3.4. PARMAP

The PARMAP algorithm is derived in a similar manner, except that we constrain the parameter density estimate to be an impulse function.

$$\hat{q}_{\Phi}(\mu,s) = \delta(\mu-\hat{\mu})\,\delta(s-\hat{s}) \tag{4.3.13}$$

The signal density $\hat{q}_X(x)=p_{X|\Phi}(x\,|\hat{\mu},\hat{s})$ will be a truncated Gaussian density, exactly as in the MCEM algorithm (4.3.10). The coefficients of these densities will be calculated by the following:

```
PARMAP Iterative Algorithm:
Guess $\hat{\mu}_0$, $\hat{s}_0 = \frac{1}{\hat{\sigma}_0^2}$
For $k = 0, 1, \cdots$
    $\hat{x}_{i,k+1} = \mathrm{E}_{X_i}\left[ x_i \;\middle|\; \hat{\mu}_k, \hat{s}_k \right]$
    $\hat{V}_{i,k+1} = \mathrm{Var}_{X_i}\left[ x_i \;\middle|\; \hat{\mu}_k, \hat{s}_k \right]$
    $\hat{\mu}_{k+1} = \frac{1}{N+\epsilon} \sum_{i=1}^{N} \hat{x}_{i,k+1}$
    $\hat{\sigma}_{k+1}^2 = \frac{1}{\hat{s}_{k+1}} = \frac{2\eta + \epsilon \hat{\mu}_{k+1}^2 + \sum_{i=1}^{N} (\hat{x}_{i,k+1} - \hat{\mu}_{k+1})^2 + \hat{V}_{i,k+1}}{N+1}$
```

As suggested by the relationship in (4.3.12), the PARMAP algorithm has exactly the same structure as the MCEM algorithm except that the estimate of the variance is slightly larger. We start with an initial estimate of the distribution mean and variance. Each sample is then estimated by calculating its conditional mean $\hat{x}_{i,k+1}$ given $\hat{\mu}_k$ and $\hat{\sigma}_k^2$. The variance $\hat{V}_{i,k+1}$ of this estimate is also calculated. The distribution mean $\hat{\mu}_{k+1}$ is then recalculated by averaging the individual sample estimates. The distribution variance $\hat{\sigma}_{k+1}^2$ is estimated by combining the variance of the sample means about the distribution mean, $\sum(\hat{x}_i - \hat{\mu})^2$, with the variance $\hat{V}_{i,k+1}$ of each individual sample x_i about the sample mean $\hat{x}_i$. Note that in calculating $\hat{\sigma}_{k+1}^2$, PARMAP divides by $N+1$ while MCEM divides by $N+2$; this is the sole difference between the algorithms. Each iteration increases the likelihood function $p(X, \hat{\mu}_k, \hat{s}_k)$, and Appendix C proves that the estimates are bounded and converge to the set of stationary points of the algorithm and critical points of $p(X, \mu, s)$. We conjecture that this density has only a single critical point, though we have not been able to prove this. PARMAP will generally be asymptotically consistent as $N \to \infty$.

3.5. SIGMAP

The SIGMAP algorithm constrains the estimated sample density to be an impulse function:

$$\hat{q}_X(x_i) = \delta(x_i - \hat{x}_i) \tag{4.3.14}$$

while the estimated parameter density $\hat{q}_\Phi(\mu,s) = p_{\Phi|X}(\mu,s\,|\hat{x})$ will be the product of a Gaussian $\hat{q}_\Phi(\mu\,|s)$ with a Gamma density $\hat{q}_\Phi(s)$, as in the MCEM algorithm (4.3.11). Substituting these densities into the SIGMAP algorithm and simplifying yields the following algorithm for calculating the coefficients of these densities:

```
SIGMAP Iterative Algorithm:
Guess $\hat{\mu}_0$, $\hat{s}_0 = \frac{1}{\hat{\sigma}_0^2}$
For $k = 0,1,\cdots$
    $\hat{x}_{i,k-1} = \begin{cases} L_i & \text{if } \hat{\mu}_k \le L_i \\ \hat{\mu}_k & \text{if } L_i \le \hat{\mu}_k \le U_i \\ U_i & \text{if } U_i \le \hat{\mu}_k \end{cases}$
    $\hat{\mu}_{k+1} = \frac{1}{N+\epsilon} \sum_{i=1}^{N} \hat{x}_{i,k+1}$
Iterate until convergence, then:
    $\hat{\sigma}^2 = \frac{1}{\hat{s}} = \frac{2\eta + \epsilon\hat{\mu}^2 + \sum_{i=1}^{N} (\hat{x}_i - \hat{\mu})^2}{N+2}$
```

This algorithm is considerably simpler than the MCEM or PARMAP algorithms. We start with estimates of the distribution mean $\hat{\mu}_k$ and variance $\hat{\sigma}_k^2$. Each sample x_i is estimated by finding the mode of the conditional sample density $p_{X_i}(x_i\,|\hat{\mu}_k,\hat{s}_k)$ (compare with (4.3.5).) This is equivalent to finding the value of x_i in the interval $[L_i,U_i]$ which comes closest to $\hat{\mu}_k$. The distribution mean is then estimated by averaging these sample estimates. Note that no estimate of the variance is required to calculate the next estimate of the sample. Thus we can iterate until the estimates of $\hat{\mu}$ and $\hat{x}_i$ converge, and then at the end estimate $\hat{\sigma}^2$ by calculating the variance of the sample estimates $\hat{x}_i$ about the mean $\hat{\mu}$. Each iteration increases the likelihood $p(\hat{x}_k, \Phi)$, and Appendix C proves that the estimates remain bounded and converge to the unique global maximum of $p(\hat{x}_k, \Phi)$. Furthermore, the convergence rate is geometric, with:

$$\left| \hat{\mu}_{k+1} - \hat{\mu}_k \right| \le \frac{N}{N+\epsilon} \left| \hat{\mu}_k - \hat{\mu}_{k-1} \right| \tag{4.3.15}$$

While SIGMAP is quite simple, its estimates are unfortunately quite poor. The sample estimates are heavily biased in the direction of the distribution mean $\hat{\mu}$; this will usually lead to poor estimates of $\hat{\mu}$. Since the sample estimates $\hat{x}_i$ will be too close to $\hat{\mu}$, and since the formula for $\hat{\sigma}^2$ neglects the variance of the sample within the bin $[L_i, U_i]$, the variance estimate $\hat{\sigma}^2$ will usually be very low. This bias in the SIGMAP estimates usually remains even as $N \to \infty$.

3.6. PSMAP

The PSMAP algorithm iteratively calculates:

$$\begin{aligned} \hat{x}_{k+1} &\leftarrow \max_{L_i \le x_i \le U_i} p(x, \hat{\mu}_k, \hat{s}_k) \\ \hat{\mu}_{k+1}, \hat{s}_{k+1} &\leftarrow \max_{\mu, s} p(\hat{x}_{k+1}, \mu, s) \end{aligned} \tag{4.3.16}$$

Substituting the density (4.3.9) and simplifying yields the algorithm:

```
PSMAP Iterative Algorithm:
Guess $\hat{\mu}_0$, $\hat{s}_0 = \frac{1}{\hat{\sigma}_0^2}$
For $k = 01, \cdots$
    $\hat{x}_{i,k+1} = \begin{cases} L_i & \text{if } \hat{\mu}_k \leq L_i \\ \hat{\mu}_k & \text{if } L_i \leq \hat{\mu}_k \leq U_i \\ U_i & \text{if } U_i \leq \hat{\mu}_k \end{cases}$
    $\hat{\mu}_{k+1} = \frac{1}{N+\epsilon} \sum_{i=1}^{N} \hat{x}_{i,k+1}$
Iterate until convergence, then:
    $\hat{\sigma}^2 = \frac{2\eta + \epsilon\hat{\mu}^2 + \sum_{i=1}^{N} (\hat{x}_i - \hat{\mu})^2}{N+1}$
```

The PSMAP and SIGMAP algorithms are thus identical in this example, except that at the condusion PSMAP estimates $\hat{\sigma}^2$ by dividing by $N+1$ instead of $N+2$. The same interpretation and convergence results for SIGMAP thus apply to this algorithm as well.

3.7. Faster SIGMAP and PSMAP Algorithm

For this problem, it is actually possible to derive an algorithm for solving SIGMAP and PSMAP which converges in a finite number of steps. The key is to note that:

$$p(x, \Phi) = \frac{K}{\left(\eta + \frac{1}{2} x^T \Sigma^{-1} x\right)^{N/2+1}} \tag{4.3.17}$$

and:

$$\max_{\mu, s} p(x, \mu, s) = \frac{K}{\left(\eta + \frac{1}{2} x^T \Sigma^{-1} x\right)^{(N+1)/2}} \tag{4.3.18}$$

$$\text{where: } \Sigma^{-1} = I - \frac{1}{N+\epsilon} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} (1 \cdots 1)$$

Maximizing either of these densities (the SIGMAP and PSMAP problems) is thus equivalent to solving:

$$\hat{x} - \min_{L_i \le x_i \le U_i} x^T \Sigma^{-1} x \tag{4.3.19}$$

Since $\Sigma^{-1} > 0$, this is a quadratic programming problem with positive definite quadratic objective function and convex constraint sets. Numerous algorithms are known for solving this problem in a finite number of steps. [3, 4] A particularly efficient algorithm can be developed for the common statistical problem in which we have divided the sample space into M fixed bins:

$$-\infty = L_0 < U_0 = L_1 < \cdots = L_{M-1} < U_{M-1} = +\infty \tag{4.3.20}$$

and we simply counted the number of samples n_i which fall in each bin, $\sum_{i=0}^{M-1} n_i = N$. Estimating $\hat{\mu}$ involves two conceptually separate steps: finding in which bin $\hat{\mu}$ must lie, and then estimating its exact value within this bin. If we knew that $\hat{\mu}$ were in the ν^{th} bin, $L_\nu \le \hat{\mu} \le U_\nu$, then its exact value could be calculated by:

$$\begin{aligned} \hat{\mu} &= \frac{1}{N+\epsilon}\left[\sum_{i=1}^{\nu-1} n_i U_i + n_\nu \hat{\mu} + \sum_{i=\nu+1}^{M-1} n_i L_i\right] \\ &= \frac{1}{N-n_\nu+\epsilon}\left[\sum_{i=0}^{\nu-1} n_i U_i + \sum_{i=\nu+1}^{M-1} n_i L_i\right] \end{aligned} \tag{4.3.21}$$

If this value falls within the ν^{th} bin's boundaries, then we have indeed located the unique bin containing $\hat{\mu}$, and have calculated its exact value. If, on the other hand, $U_\nu < \hat{\mu}$, then it is easy to show that the true SIGMAP solution $\hat{\mu}$ must lie somewhere in bins $\nu+1$ through $M-1$; if $\hat{\mu} < L_\nu$ then the SIGMAP solution $\hat{\mu}$ must lie somewhere in bins 0 through $\nu-1$. Using a binary search algorithm to efficiently search for the correct bin then gives the following algorithm:

<u>Fast SIGMAP/PSMAP Algorithm:</u>

```
$\nu_0 \leftarrow 0$          ; $\nu_0$ = guess of which bin $\hat{\mu}$ lies in
size $\leftarrow M$       ; size = number of bins remaining where $\hat{\mu}$ could lie
$\hat{\mu} \leftarrow 0$          ; initial guess for $\hat{\mu}$

while (size > 0) {
    $\nu \leftarrow \nu_0 +$ size/2
    Guess $\hat{\mu} \leftarrow \frac{1}{N-n_\nu+\epsilon}\left[\sum_{i=1}^{\nu-1} n_i U_i + \sum_{i=\nu+1}^{M-1} n_i L_i\right]$
    if $(L_\nu \le \hat{\mu} \le U_\nu)$ then:
        $\nu_0 \leftarrow \nu$
        goto DONE
    else if $(\hat{\mu} < L_\nu)$ then
        size $\leftarrow$ size/2
    else if $(\hat{\mu} > U_\nu)$ then
        size $\leftarrow$ size/2 - 1
        $\nu_0 \leftarrow \nu+$size/2+1
}

DONE:    $\nu_0$ is bin in which $\hat{\mu}$ lies
         $\hat{\mu}$ is estimate of mean
         $\hat{x}_i = \begin{cases} L_i & \text{if } \hat{\mu} < L_i \\ \hat{\mu} & \text{if } L_i \le \hat{\mu} \le U_i \\ U_i & \text{if } U_i < \hat{\mu} \end{cases}$
```

3.8. Known Variance

The analysis of the problem changes very little if the variance σ^2 is known and does not have to be estimated. We will assume the same *a priori* Gaussian density p(μ) as before; an *a priori* density for $s=\frac{1}{\sigma^2}$ is no longer needed. The same estimation approach used earlier can be applied, with the sole difference that the estimated parameter density $\hat{q}_\Phi(\mu)$ will only be a function of the unknown mean μ. Happily, the only change in the algorithms will be that we do not need to calculate $\hat{\sigma}_k^2$. In the MCEM and PARMAP algorithms, this also implies that the individual sample variances $\hat{V}_{i,k+1}$

do not have to be calculated. MCEM and PARMAP thus become identical and are considerably simpler than before:

MCEM/PARMAP - Known Variance σ^2
For $k=0,1,\cdots$

$$\hat{x}_{i,k+1} = \mathrm{E}_{X_i}\left[x_i \mid \hat{\mu}_k\right]$$

$$\hat{\mu}_{k+1} = \frac{1}{N+\epsilon}\sum_{i=1}^{N} \hat{x}_{i,k+1}$$

When σ^2 is known, Appendix C proves that this iteration is guaranteed to converge to the unique globally optimum solution to the PARMAP and MCEM problems. Furthermore, the convergence rate is geometric:

$$\left|\hat{\mu}_{k+1} - \hat{\mu}_k\right| \le \frac{N}{N+\epsilon}\left|\hat{\mu}_k - \hat{\mu}_{k-1}\right| \tag{4.3.22}$$

where this guaranteed convergence rate $\frac{N}{N+\epsilon}$ is actually very conservative.

The SIGMAP and PSMAP algorithms will have exactly the same structure as before; the only difference is that it is not necessary to estimate $\hat{\sigma}^2$ at the end.

3.9. Maximum Likelihood Algorithms

If we are not allowed to treat μ and σ^2 as random variables, then it is necessary to use Fisher estimation techniques, rather than the above algorithms. Unfortunately, it is not sufficient simply to set $\epsilon=0$ and $\eta=0$. Forgetting the philosophical implications, let us first derive the Maximum Likelihood version of MCEM. Substituting the model density $p(x \mid \mu,s)$ into the ML version of MCEM and simplifying yields:

MCEM Algorithm - ML Version

For $k=0,1,\cdots$

$$\hat{x}_{k,k+1} = E_{X_i}\left[x_i \mid \hat{\mu}_k, \hat{s}_k \right]$$

$$\hat{V}_{i,k+1} = \mathrm{Var}_{X_i}\left[x_i \mid \hat{\mu}_k, \hat{s}_k \right]$$

$$\hat{\mu}_{k+1} = \frac{1}{N}\sum_{i=1}^{N} \hat{x}_{i,k+1}$$

$$\hat{\sigma}_{k+1}^2 = \frac{1}{\hat{s}_{k+1}} = \frac{\sum_{i=1}^{N} (\hat{x}_{i,k+1}-\hat{\mu}_{k+1})^2 + \hat{V}_{i,k+1}}{N+1}$$

where the expectation and variance of x_i are calculated as before. Note that in addition to the terms ϵ and ν being set to zero, the denominator in the expression for $\hat{\sigma}_{k+1}^2$ has been changed to $N+1$ from $N+2$ in the Bayesian version. This makes the ML version of the MCEM algorithm perform more like our Bayesian PARMAP algorithm in section 3.4. Each iteration still decreases the appropriate cross-entropy expression and thus improves the estimates. Appendix C proves that if any L_i and any U_j are finite, then the parameter estimates will be bounded and thus converge to the set of stationary points of the algorithm and critical points of the cross-entropy.

The PARML algorithm can be derived in the same manner as PARMAP, except that we use the model density $p(x \mid \mu, s)$ instead of $p(x, \mu, s)$. The resulting algorithm is identical to the ML version of MCEM, except that the variance is estimated by:

$$\hat{\sigma}_{k+1}^2 = \frac{1}{\hat{s}_{k+1}} = \frac{\sum_{i=1}^{N} (\hat{x}_{i,k+1}-\hat{\mu}_{k+1})^2 + \hat{V}_{i,k+1}}{N} \qquad (4.3.23)$$

Note that the denominator is N, not $N+1$ as in the Bayesian version of the algorithm. Each iteration increases the likelihood function $p(X \mid \hat{\mu}_k, \hat{s}_k)$, and Appendix C proves that if any interval $[L_i, U_i]$ is finite then the estimates remain bounded and converge to the set of stationary points of the algorithm and critical points of the likelihood function.

A Maximum Likelihood version of SIGMAP is not possible. A Fisher PSML algorithm can be easily derived, however, by iteratively maximizing $p(x \mid \mu, s)$. The calculation of $\hat{\mu}$ and $\hat{x}$ will be unmodified from the Bayesian PSMAP algorithm, except for setting $\epsilon = 0$, but the calculation of $\hat{\sigma}^2$ will be:

$$\hat{\sigma}^2 = \frac{1}{\hat{s}} = \frac{\sum_{i=1}^{N} (\hat{x}_i - \hat{\mu})^2}{N} \qquad (4.3.24)$$

The denominator has changed to N from $N+1$. Each iteration increases the likelihood function, thus "improving" the estimates. Appendix C proves that the iteration will converge to the convex set of global maximum solutions to the PSML problem. (There may be many such solutions.)

3.10. Comparison of the Algorithms

To compare these algorithms, we will apply them to a problem similar to that used in the last section. We start with a Gaussian density $p(x_i \mid \mu, \sigma^2)$ with mean $\mu = 0$ and standard deviation $\sigma = 3$. Nearly flat *a priori* densities for μ and $x = \frac{1}{\sigma^2}$ are assumed with $\epsilon = \eta = 10^{-4}$. To estimate the parameters, we draw N independent samples x_i from the distribution, and count how many fall into each of five "bins": $x_i < 0$, $0 \le x_i < 1$, $1 \le x_i < 2$, $2 \le x_i < 3$, and $3 \le x_i$. (Note that this selection of bins is rather lopsided, since half of the samples fall into the first bin.) Given the count of how many samples fall into each bin, we apply each of our algorithms to estimate the parameters of the density. Tables 4.2 and 4.3 below show the estimates of the "natural" parameters $\pi_1 = \frac{1}{\sigma^2}$ and $\pi_2 = \frac{\mu}{\sigma^2}$ generated by our algorithms (see equation (4.3.9).) The classical parameter estimate, using the actual values of x, must be taken as the best that is possible, since grouping the data into bins can only increase the uncertainty about the parameters. MMSE using the grouped data is orders of magnitude more difficult than the other methods, since calculating each of the numbers in the table required a multidimensional numerical integration involving 120,000 to 250,000 function evaluations, each of which is as difficult as one iteration of MCEM or PARMAP. Surprisingly, MCEM gives estimates of the natural parameters π_1 and π_2 which are very close to those of MMSE. The convergence speed is moderate; starting from deliberately poor initial estimates of $\hat{\mu}_0 = -1$ and $\hat{\sigma}_0^2 = 1$, between 20 to 40 iterations were required to converge to machine precision. (MMSE thus requires at least 6000 times more effort to calculate virtually the same estimates!) As expected, PARMAP gives estimates of $\pi_1 = s$ which are smaller than those of MCEM; as a result, its estimates of $\pi_2 = \mu s$ are also somewhat different. Convergence speed is the same as MCEM. SIGMAP and PSMAP give extremely large estimates of π_1, and their estimates of π_2 are very different from MMSE. The last line in these tables corresponds to a sequence where 8 samples fell into the first bin, and the other 2 were in the second bin; beware that the iterative methods required 5 times more iterations than usual to converge.

The above comparison is actually slightly misleading, since we are usually not interested in the "natural" parameters π_1 and π_2, but would prefer estimates of the mean μ and variance σ^2. In our iterative algorithms, the structure of $\hat{q}_\Phi$ ensures that

$$E[\mu \mid \hat{q}_\Phi] = \frac{E[\pi_2 \mid \hat{q}_\Phi]}{E[\pi_1 \mid \hat{q}_\Phi]} \tag{4.3.25}$$

Similarly, for the classical method:

$$E[\mu \mid x] = \frac{E[\pi_2 \mid x]}{E[\pi_1 \mid x]} \tag{4.3.26}$$

However, in MMSE the expectation of μ does not bear any simple relationship to that of π_1 or π_2. Table 4.4 compares the estimates of $\hat{\mu}$ generated by all our algorithms on the same data as the other tables. Note that the MMSE estimates $E[\mu \mid x \in X]$ do not follow MCEM's estimates very closely, and in fact seem to be closer to PARMAP's estimates.

Figures 4.3-4.6 show histograms of the estimates of $\hat{\mu}$ and $\hat{\sigma}^2$ generated by the classical method and our four iterative algorithms for 500 sequences of $N=10$ and $N=100$ samples each. Note that both MCEM and PARMAP appear to be asymptotically consistent, while SIGMAP and PSMAP appear to be biased. PARMAP's estimates also appear to have more spread than MCEM's - in fact to make the histograms clearer, about 8 outlying estimates of $\hat{\sigma}^2$ were omitted from the PARMAP graphs. These extremely large estimates, like the last lines in the tables, were caused by sequences in which nearly all the samples fell in the first bin. PARMAP's estimates of μ appear slightly closer to the correct value of 0 than MCEM's estimates, although the difference is minor compared to the standard deviation of the estimates. (As pointed out above, MCEM does best at estimating the "natural" parameters.) MCEM's estimates of $\hat{\sigma}^2$ were closer to the classical estimate than PARMAP. (Note that PARMAP is actually closer to the true value of σ^2 than MCEM, but this is misleading since the standard for comparison must be the classical estimate which uses the exact data values that were generated.)

We repeated the same experiment in figures 4.7-4.8 for the case when the variance $\sigma^2=9$ was known. Both the MCEM and PARMAP algorithms are identical in this case, and converge at a fast geometric rate, cutting the error about in half on each iteration. SIGMAP and PSMAP are also identical and converge even faster. When only μ is unknown, it is feasible to evaluate MMSE for all 500 sequences, since only a single integration was needed. (MMSE still required about 20 times more effort than MCEM or PARMAP.) Note that the MCEM/PARMAP estimates are very close to MMSE, and appear to be asymptotically consistent as $N \to \infty$. SIGMAP/PSMAP is asymptotically biased.

To summarize, therefore, given grouped data the MMSE estimates are not computationally practical to calculate. MCEM, PARMAP, SIGMAP and PSMAP, on the other hand, are numerically robust and easy to compute. MCEM's estimates of the "natural" parameters π_1 and π_2 are very close to MMSE's estimates. Both MCEM and PARMAP appear to be asymptotically consistent and both converge relatively quickly. Faster convergence could undoubtedly be achieved by using extrapolation. SIGMAP and PSMAP are the easiest to compute, particularly since an algorithm is available which converges in a finite number of steps, but their estimates are strongly biased.

Seq-	Classical	MMSE	MCEM	PARMAP	SIGMAP	PSMAP
uence	$E[\pi_1 \mid x]$	$E[\pi_1 \mid x \in X]$	$E[\pi_1 \mid \hat{q}_\Phi]$	$E[\pi_1 \mid \hat{q}_\Phi]$	$E[\pi_1 \mid \hat{q}_\Phi]$	$E[\pi_1 \mid \hat{q}_\Phi]$
#1	0.216	0.496	0.480	0.350	3.000	2.750
#2	0.231	0.448	0.437	0.370	1.256	1.151
#3	0.104	0.371	0.362	0.294	1.059	0.971
#4	0.087	0.128	0.119	0.076	0.882	0.809
#5	0.216	0.235	0.228	0.177	1.000	0.917
#6	0.127	0.231	0.218	0.143	1.459	1.338
#7	0.130	0.176	0.166	0.108	1.174	1.076
#8	0.120	0.094	0.088	0.056	0.659	0.604
#9	0.202	0.221	0.211	0.153	1.188	1.089
#10	0.102	0.128	0.119	0.076	0.882	0.809
#11	1.128	5.521	10000.000	4926.108	58823.527	55555.555

Table 4.2 - Estimates of $\pi_1 = \frac{1}{\sigma^2}$, Gaussian Density

(true value = .111)

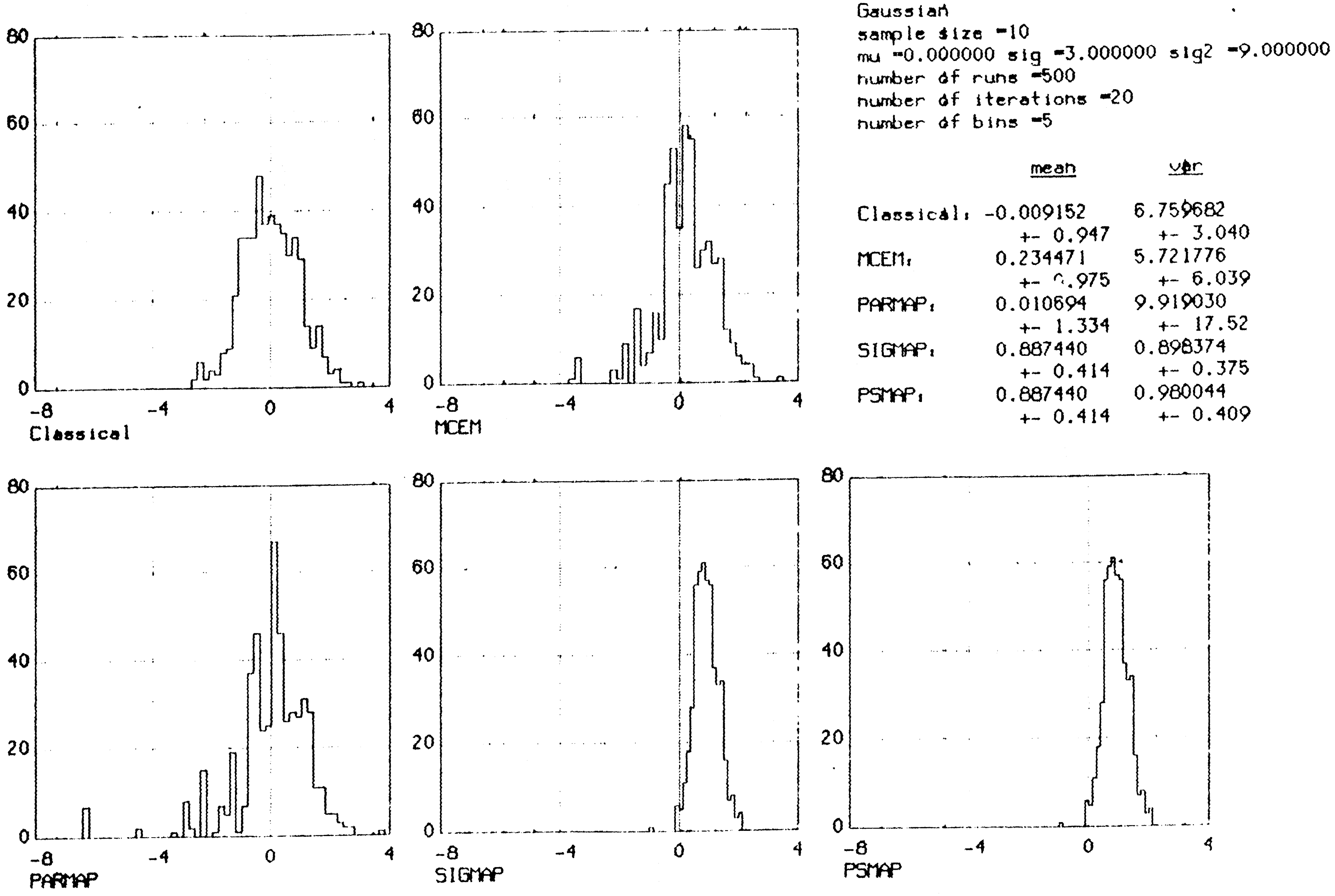
Seq-	Classical	MMSE	MCEM	PARMAP	SIGMAP	PSMAP
uence	$E[\pi_2 \mid x]$	$E[\pi_2 \mid x \in X]$	$E[\pi_2 \mid \hat{q}_\Phi]$	$E[\pi_2 \mid \hat{q}_\Phi]$	$E[\pi_2 \mid \hat{q}_\Phi]$	$E[\pi_2 \mid \hat{q}_\Phi]$
#1	-0.332	-0.246	-0.255	-0.253	1.000	0.917
#2	0.103	0.491	0.478	0.397	1.535	1.407
#3	0.030	0.496	0.483	0.389	1.412	1.294
#4	-0.086	-0.018	-0.031	-0.045	0.706	0.647
#5	0.064	0.137	0.124	0.077	1.000	0.917
#6	-0.222	-0.173	-0.189	-0.176	0.649	0.595
#7	-0.139	-0.142	-0.157	-0.148	0.652	0.598
#8	0.113	0.153	0.143	0.091	1.024	0.939
#9	-0.018	-0.014	-0.028	-0.052	0.832	0.762
#10	-0.082	-0.018	-0.031	-0.045	0.706	0.647
#11	-0.912	-2.022	-84.160	-59.118	-0.118	-0.111

Table 4.3 - Estimates of $\pi_2 = \frac{\mu}{\sigma^2}$, Gaussian Density

(true value = 0)

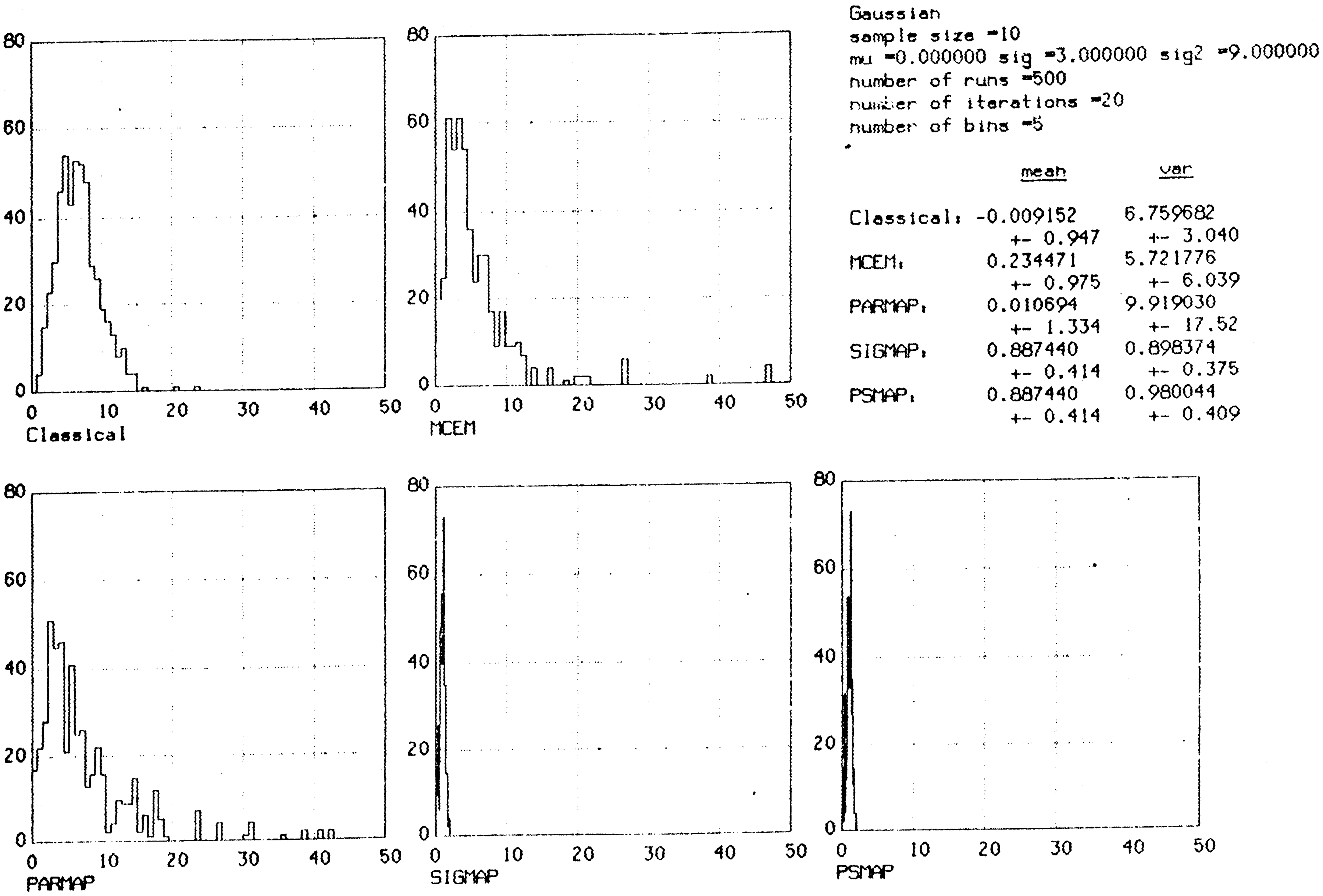
Seq- uence	Classical $E[\mu \mid \underline{x}]$	MMSE $E[\mu \mid \underline{x} \in X]$	MCEM $E[\mu \mid \hat{q}_\Phi]$	PARMAP $E[\mu \mid \hat{q}_\Phi]$	SIGMAP $E[\mu \mid \hat{q}_\Phi]$	PSMAP $E[\mu \mid \hat{q}_\Phi]$
#1	-1.535	-0.837	-0.530	-0.722	0.333	0.333
#2	0.444	1.057	1.094	1.073	1.222	1.222
#3	0.291	1.314	1.337	1.321	1.333	1.333
#4	-0.987	-0.731	-0.258	-0.589	0.800	0.800
#5	0.295	0.368	0.546	0.437	1.000	1.000
#6	-1.749	-1.421	-0.867	-1.231	0.444	0.444
#7	-1.065	-1.571	-0.944	-1.361	0.556	0.556
#8	0.944	1.628	1.628	1.630	1.556	1.556
#9	-0.087	-0.455	-0.134	-0.339	0.700	0.700
#10	-0.807	-0.731	-0.258	-0.589	0.800	0.800
#11	-0.809	-0.472	-0.008	-0.012	-0.000	-0.000

Table 4.4 - Estimates of μ, Gaussian Density
(true value = 0)



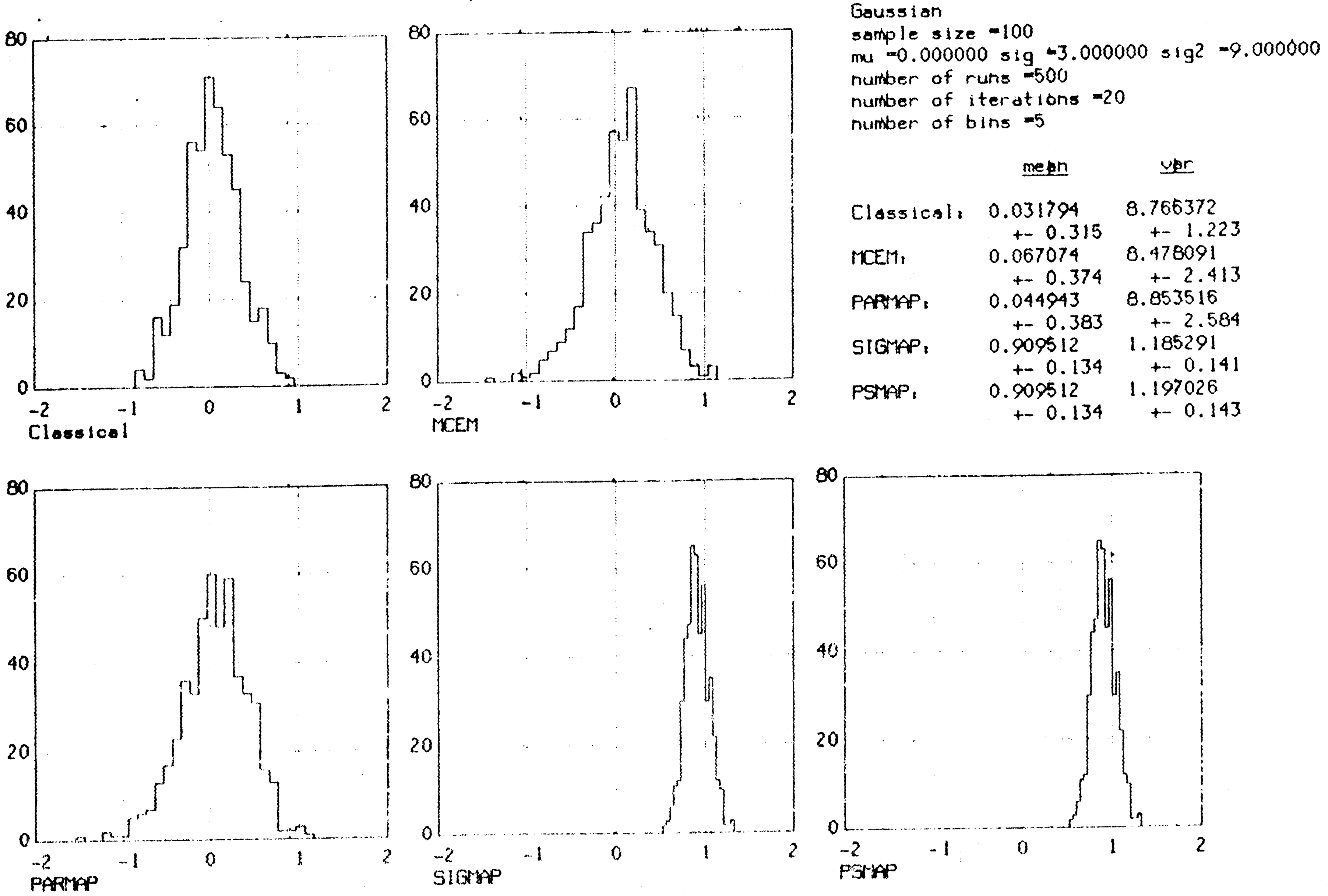
```
Gaussian
sample size =10
mu =0.000000 sig =3.000000 sig2 =9.000000
number of runs =500
number of iterations =20
number of bins =5

                  mean          var

Classical: -0.009152      6.759682
              +- 0.947      +- 3.040
MCEM:       0.234471      5.721776
              +- 0.975      +- 6.039
PARMAP:     0.010694      9.919030
              +- 1.334      +- 17.52
SIGMAP:     0.887440      0.898374
              +- 0.414      +- 0.375
PSMAP:      0.887440      0.980044
              +- 0.414      +- 0.409
```

Figure 4.3 - Histograms of $\hat{\mu}$ for $N=10$, Gaussian Density



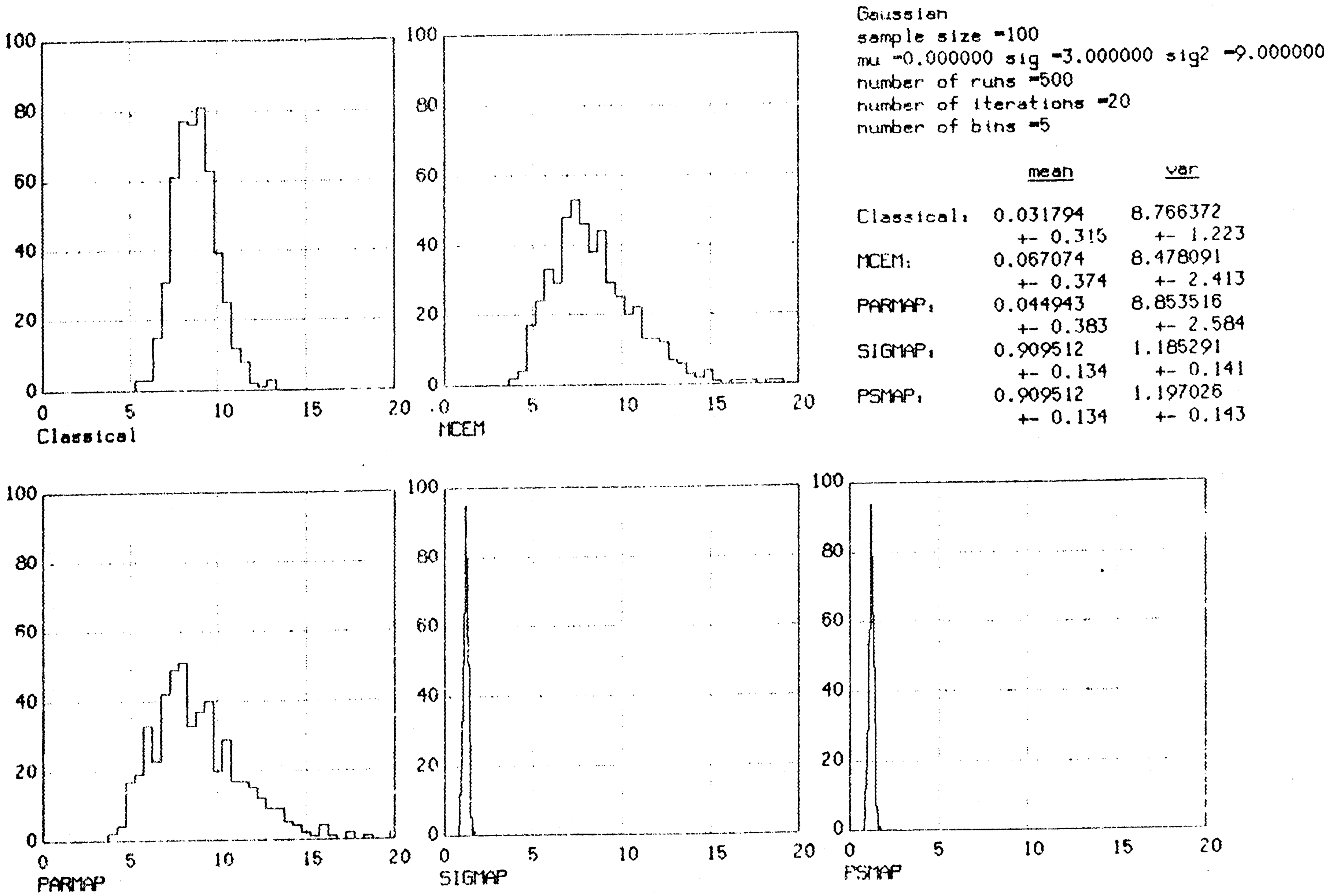
```
Gaussian
sample size =10
mu =0.000000 sig =3.000000 sig2 =9.000000
number of runs =500
number of iterations =20
number of bins =5

                mean          var

Classical: -0.009152       6.759682
              +- 0.947       +- 3.040
MCEM:       0.234471       5.721776
              +- 0.975       +- 6.039
PARMAP:     0.010694       9.919030
              +- 1.334       +- 17.52
SIGMAP:     0.887440       0.898374
              +- 0.414       +- 0.375
PSMAP:      0.887440       0.980044
              +- 0.414       +- 0.409
```

Figure 4.4 - Histograms of $\hat{\sigma}^2$ for $N = 10$, Gaussian Density



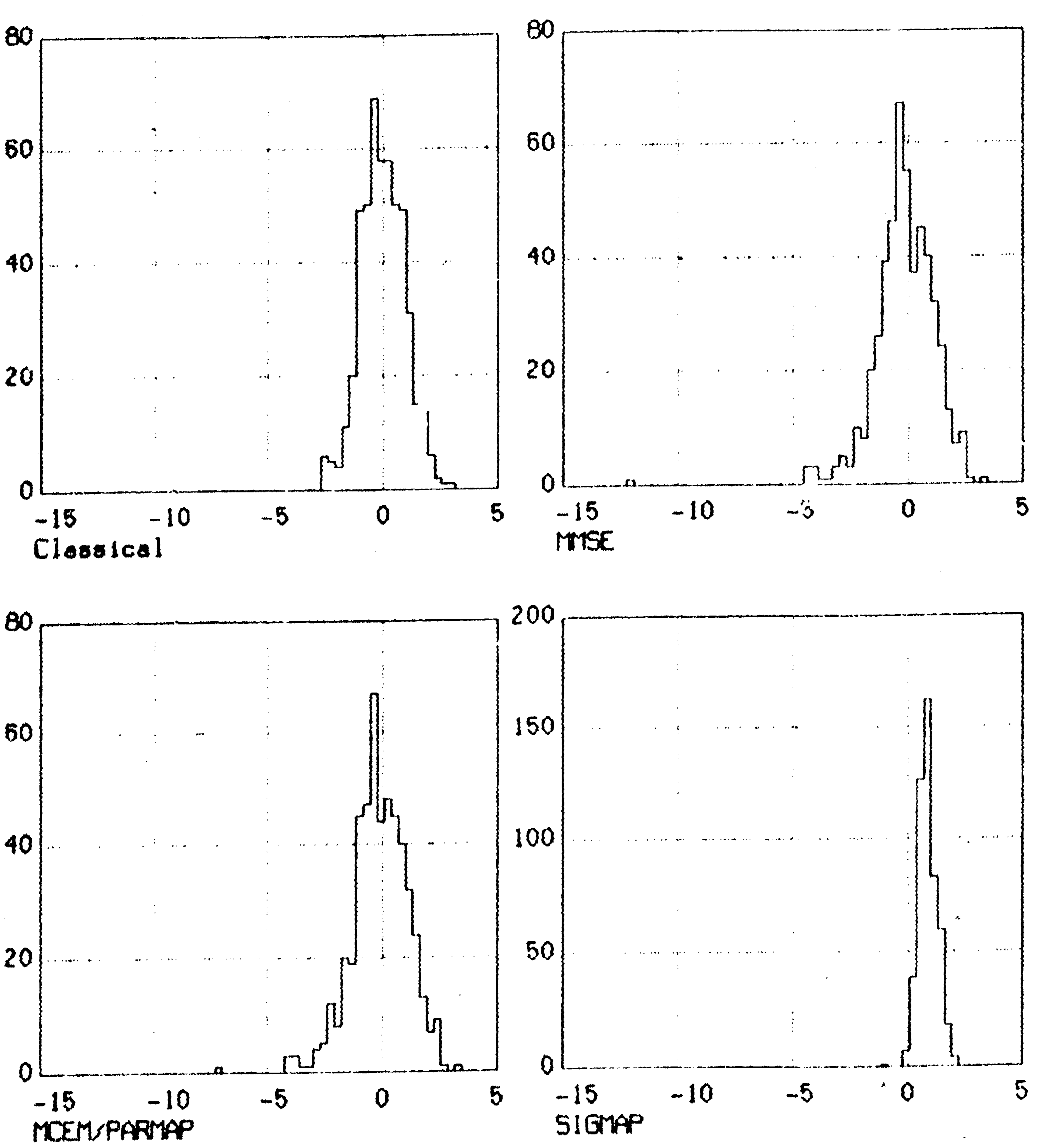
```
Gaussian
sample size =100
mu =0.000000 sig =3.000000 sig2 =9.000000
number of runs =500
number of iterations =20
number of bins =5

                 mean          var

Classical:    0.031794      8.766372
                +- 0.315      +- 1.223
MCEM:         0.067074      8.478091
                +- 0.374      +- 2.413
PARMAP:       0.044943      8.853516
                +- 0.383      +- 2.584
SIGMAP:       0.909512      1.185291
                +- 0.134      +- 0.141
PSMAP:        0.909512      1.197026
                +- 0.134      +- 0.143
```

Figure 4.5 - Histograms of $\hat{\mu}$ for $N = 100$, Gaussian Density



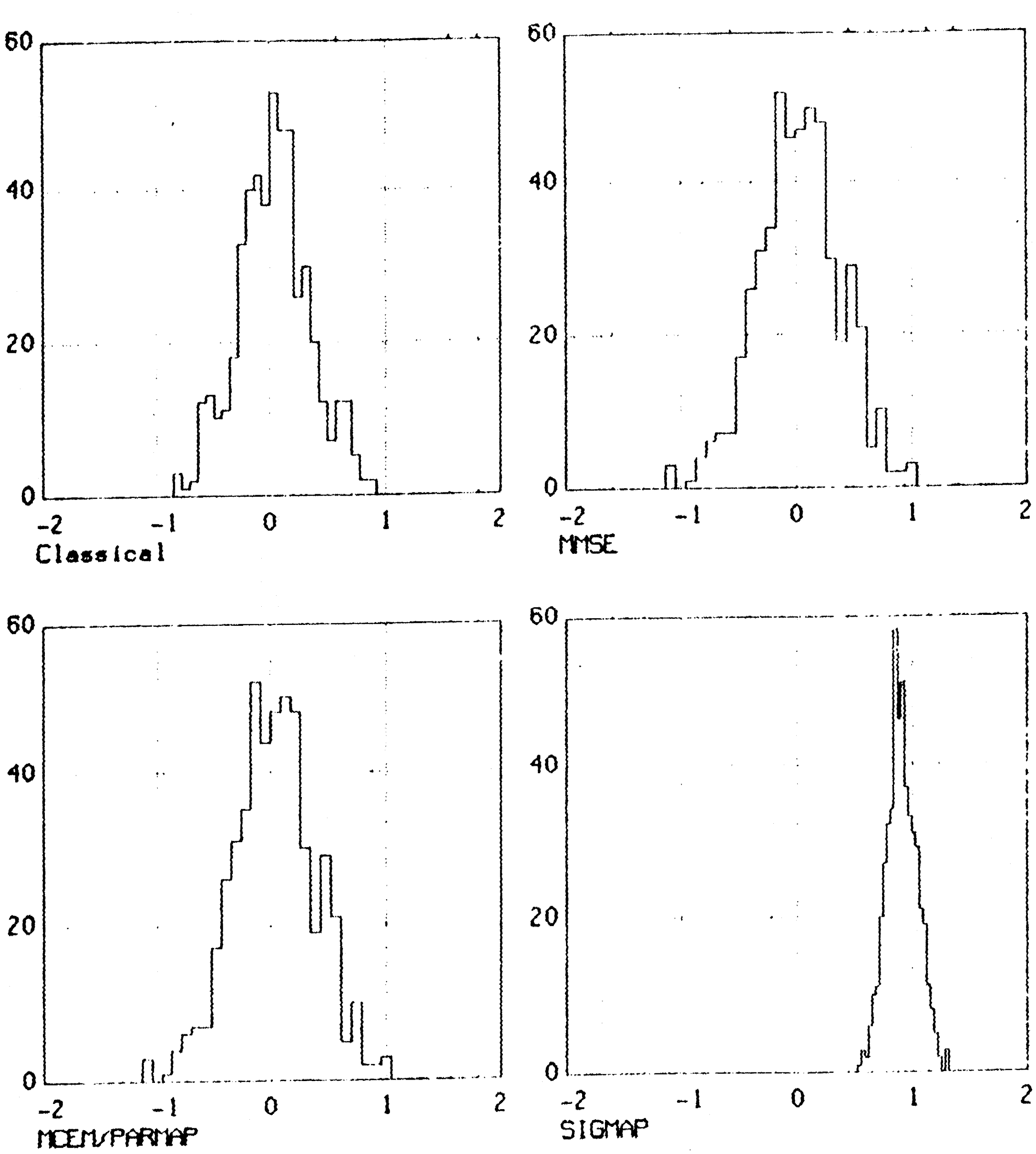
```
Gaussian
sample size =100
mu =0.000000 sig =3.000000 sig2 =9.000000
number of runs =500
number of iterations =20
number of bins =5

                 mean        var

Classical:   0.031794     8.766372
               +- 0.315     +- 1.223
MCEM:        0.067074     8.478091
               +- 0.374     +- 2.413
PARMAP:      0.044943     8.853516
               +- 0.383     +- 2.584
SIGMAP:      0.909512     1.185291
               +- 0.134     +- 0.141
PSMAP:       0.909512     1.197026
               +- 0.134     +- 0.143
```

Figure 4.6 - Histograms of $\hat{\sigma}^2$ for $N = 100$, Gaussian Density



```
Gaussian
sample size =10
mu =0.000000 sig =3.000000 sig2 =9.000000
number of runs =500
number of iterations =20
number of bins =5

                  mean          var

Classical,   -0.009152     6.759682
                +- 0.947     +- 3.040
MMSE:        -0.158196     9.000000
                +- 1.328     +- 0.000
MCEM/PARMAP -0.114788      9.000000
                +- 1.217     +- 0.000
SIG/PSMAP:    0.887440     9.000000
                +- 0.414     +- 0.000
```

Figure 4.7 - Histograms of $\hat{\mu}$ for $N=10$, Gaussian Density, Known Variance σ^2



```
Gaussian
sample size =100
mu =0.000000 sig =3.000000 sig2 =9.000000
number of runs =500
number of iterations =20
number of bins =5

                 mean          var

Classical:    0.031794      8.766372
                +- 0.315      +- 1.223
MMSE:         0.025196      9.000000
                +- 0.360      +- 0.000
MCEM/PARMAP 0.027354      9.000000
                +- 0.359      +- 0.000
SIG/PSMAP:    0.909512      9.000000
                +- 0.134      +- 0.000
```

Figure 4.8 - Histograms of $\hat{\mu}$ for $N = 100$, Gaussian Density, Known Variance σ^2

4. Conclusions

Our four iterative estimation algorithms represent a numerically robust and computationally efficient approach for estimating parameters of exponential families of densities from grouped or quantized data. Unlike the MMSE method or other Bayesian Minimum Risk estimators which require a complicated integration, these methods simply alternate between estimating the sample values using the latest parameter estimates, then reestimating the parameter values using the samples estimates. Each iteration decreases the appropriate cross-entropy function, and in the case of the MAP algorithms, also increases the appropriate likelihood function. Convergence of all four algorithms can be proven under mild conditions. MCEM and PARMAP, in particular, give nearly unbiased estimates even for small samples, and are asymptotically consistent and efficient in most problems. In the cases we have tested, MCEM's estimates of the "natural" parameters of the densities come especially close to those of MMSE.

While we have only considered Exponential and Gaussian densities in this chapter, the idea is easy to extend to any other exponential family of densities, such as Gamma, Binomial, Multinomial, Negative Binomial, etc. The only drawback in using these densities is that the algorithms may need to compute special functions which are not always included in standard mathematics subroutine libraries. To apply MCEM or PARMAP to Gamma densities, for example, one needs to be able to easily compute the derivative of a Gamma function, an operation not much more difficult than computing an Error Function, but which is not readily available. The idea can also be extended to problems of censored or truncated data, in which the data collection process discards all data outside a certain range $\bar{X}$. Here, we simply modify the probability density for the model by truncating it to the range $\bar{X}$:

$$\hat{p}(x,\phi) = \begin{cases} \dfrac{p(x,\phi)}{p(\tilde{X},\Phi)} & \text{for } x \in \tilde{X} \\ \\ 0 & \text{else} \end{cases} \tag{4.4.1}$$

and then applying our algorithms to this truncated density. Offset parameters can also be handled. Suppose, for example, that:

$$p(x \mid \phi,\alpha) = \begin{cases} \phi \exp(-\phi(x-\alpha)) & \text{for } x \geq \alpha \\ 0 & \text{else} \end{cases} \tag{4.4.2}$$

and both the scaling parameter ϕ and offset α must be estimated. By grouping the offset parameter α with the data x rather than with ϕ, this probability density can be shown to form an exponential family of densities. We can thus use our methods to fit a separable density $q(x,\alpha)q(\phi)$ to the given model density, and then use this simpler density to estimate the unknowns. Yet another possible extension would be to problems involving non-flat measurement noise. It is this flexibility and wide applicability that makes our algorithms both theoretically interesting and computationally practical.

References

1. Gunnar Kulldorff, *Estimation from Grouped and Partially Grouped Samples*, John Wiley & Sons, New York (1961).
2. Germund Dahlquist and Ake Bjorck, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N.J. (1974).
3. John C. G. Boot, *Quadratic Programming - Algorithms, Anomalies, Applications*, North-Holland Publishing Company - Rand McNally & Company, Chicago (1964).
4. Hans Künzi and Wilhelm Krelle, *Nonlinear Programming*, Blaisdell Publishing, Waltham, Mass. (1966).

Chapter 5

Applications in Optimal Signal Reconstruction

Part I - Bayesian Theory

1. Introduction

In this chapter we will consider a specific linear Gaussian system model in which a stochastic Gaussian signal $\underline{x}$ is linearly filtered and corrupted by additive Gaussian noise to form the output $\underline{y}$. After discussing the form of the probability density $p(\underline{x},\underline{y}|\underline{\phi})$ describing this system, we consider the problem of optimally reconstructing the signal and output given certain constraints on their values, and assuming that all model parameters are known. The MMSE approach is briefly presented for this problem, then our four iterative MCEM and MAP algorithms are discussed in some detail. Although these approaches generally give different estimates of the unknowns, the four iterative algorithms all share a common structure, iterating between filtering operations and a pair of projection or conditional expectation operators. Geometric convergence of all four algorithms to the unique solution is guaranteed when the constraint sets are convex. Particular attention is given to linear variety and simplex constraint sets because these problems can be theoretically analyzed in great depth. For linear variety constraint sets, we show that two different "primal" and "dual" approaches to the problem can be defined, each of which leads to a different closed-form solution. Each of these can also be solved by conjugate gradient iterative algorithms in a finite number of steps.

2. Linear Gaussian Model

The "generic" system model we will use most extensively is illustrated in figure 5.1.

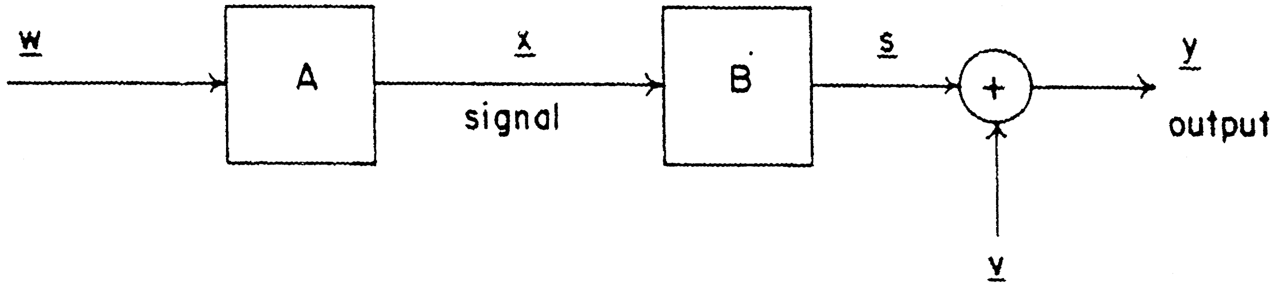


Figure 5.1 - Generic Linear Gaussian System Model

Zero mean Gaussian noise $\underline{w}$ with covariance Q passes through a linear invertible filter A^{-1} to form the N point signal $\underline{x} \in \mathbf{R}^N$. The signal is then linearly filtered by B, and independent zero mean Gaussian noise $\underline{v}$ with covariance R is added to form the M point output $\underline{y} \in \mathbf{R}^M$. Neither the signal nor the output are observed directly; instead, the incomplete observation data which is available serves only to restrict the range of feasible values for the signal, output and parameters to the sets X, Y and Φ respectively:

$$\text{Model:} \quad \begin{cases} \underline{x} = A^{-1}\underline{w} & \text{where } p(\underline{w}) = N(0,Q) \\ \underline{y} = B\underline{x} + \underline{v} & \text{where } p(\underline{v}) = N(0,R) \end{cases} \tag{5.2.1}$$

where: A is invertible ; Q>0 and R>0

$\underline{w}, \underline{x} \in \mathbf{R}^N$ and $\underline{v}, \underline{y} \in \mathbf{R}^M$

Observations: $\underline{x} \in X$, $\underline{y} \in Y$, $\phi \in \Phi$

We will usually assume that the linear filter A^{-1} is "causal", so that A and A^{-1} are lower triangular with 1's on the diagonal. We will also assume that the elements of A, B, Q and R depend at most linearly on the parameters $\underline{\phi}$. (These assumptions are convenient when we discuss parameter estimation in chapter 9, but are by no means necessary.) We will assume that the variables are real, but complex variables could be incorporated with little effort.

Because the signal $\underline{x}$ and output $\underline{y}$ are linear functions of the Gaussian variables $\underline{w}$ and $\underline{v}$, the joint probability density $p(\underline{x},\underline{y}\,|\,\underline{\phi})$ of the signal $\underline{x}$ and output $\underline{y}$ given the parameters $\underline{\phi}$ is Gaussian:

$$p(\underline{x},\underline{y}\,|\,\underline{\phi}) = N\left(\begin{pmatrix}\underline{0}\\ \underline{0}\end{pmatrix}, V_{xy} \right) \tag{5.2.2}$$

$$\text{where } V_{xy} = \begin{pmatrix} I \\ B \end{pmatrix} A^{-1}QA^{-T} \begin{pmatrix} I & B^T \end{pmatrix} + \begin{pmatrix} 0 \\ I \end{pmatrix} R \begin{pmatrix} 0 & I \end{pmatrix}$$

The covariance matrix $R=\mathrm{Var}(\underline{v})$ can be loosely viewed as the distribution of power in the noise, while the covariance matrices $A^{-1}QA^{-T}$ and $BA^{-1}QA^{-T}B^T$ can be loosely viewed as the distribution of power in the signal $\underline{x}$ and the filtered signal $\underline{z}=B\underline{x}$ respectively. Beware that when the noise power R is very small or very large relative to the filtered signal power $BA^{-1}QA^{-T}B^T$, then the covariance matrix of the Gaussian probability density becomes very nearly singular and is numerically ill-conditioned. Thus at very high or very low Signal-to-Noise Ratios (SNR) the solution to our estimation problem may be numerically ill-behaved. This phenomenon is one that will arise again in later sections, where very high SNR levels are shown to correspond to slow convergence of our iterative algorithm and high sensitivity to computation noise. Let us stress that all these phenomenon are closely linked, and that the difficulty lies in the problem formulation itself, and not solely in the iterative algorithms.

The interesting aspect of this signal model is that it is not only quite general, but also the log likelihood function $\log p(\underline{x},\underline{y}\,|\,\underline{\phi})$ has a particularly simple form. If we define the norm $\|\underline{\alpha}\|_P^2 = \underline{\alpha}^T P^{-1}\underline{\alpha}$, then taking the logarithm of equation (5.2.2) gives:

$$\log p(\underline{x},\underline{y}\,|\,\underline{\phi}) = -\tfrac{1}{2}\left\{ \|A\underline{x}\|_Q^2 + \|\underline{y}-B\underline{x}\|_R^2 + \log|2\pi Q| + \log|2\pi R| \right\} \qquad (5.2.3)$$

where we have simplified this expression slightly by using our assumption that A is lower triangular with 1's on the diagonal, so that $|A^{-1}|=|A|=1$. The interesting point is that this function is not only quadratic in the signal $\underline{x}$ and output $\underline{y}$ for fixed parameters $\underline{\phi}$, but is also quadratic in the elements of A and B for fixed $\underline{x}$, $\underline{y}$. Maximizing this function either with respect to $\underline{x}$, $\underline{y}$ or with respect to $\underline{\phi}$ therefore only requires solving linear equations. This quadratic structure also simplifies the calculation of its expectation over the sets X, Y or Φ in the MCEM, PARMAP and SIGMAP iterative algorithms, since we will only require the conditional mean and covariance of $\underline{x}$, $\underline{y}$ or $\underline{\phi}$. Calculating the expectation of $\log p(\underline{x},\underline{y}\,|\,\underline{\phi})$ over X, Y or Φ also does not change its quadratic behavior as a function of the remaining variables. It is this feature which makes this system model ideally suited for use with our iterative estimation algorithms.

3. Minimum Mean Square Error Algorithm (MMSE)

We will treat the problem of parameter identification in a later chapter; for now, let us assume that the parameters $\underline{\phi}$ are all known, so that A, B, Q and R are fixed, with A is invertible and $Q>0$, $R>0$. Our goal is to estimate the signal and output given that $\underline{x}\in X$ and $\underline{y}\in Y$. Using the ideas developed in chapters 2 and 3, there are at least five different approaches for calculating MMSE, MCEM or MAP estimates of $\underline{x}$ and $\underline{y}$. Unfortunately, except for certain special types of constraint sets to be discussed later, each approach will generate different estimates. The "best" approach, in the sense of yielding minimum variance unbiased estimates, is to solve the MMSE problem:

$$\text{MMSE:}\quad \begin{pmatrix} \hat{\underline{x}} \\ \hat{\underline{y}} \end{pmatrix} = \min_{\hat{\underline{x}} \in X,\, \hat{\underline{y}} \in Y} E_{X,Y} \left[\, \|\underline{x} - \hat{\underline{x}}\|_x^2 + \|\underline{y} - \hat{\underline{y}}\|_y^2 \, \middle| \, \underline{x} \in X, \underline{y} \in Y \right] \tag{5.3.1}$$

If the sets X and Y are convex, then the solution will be:

$$\begin{aligned} \begin{pmatrix} \hat{\underline{x}} \\ \hat{\underline{y}} \end{pmatrix} &= E_{X,Y} \left[\begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix} \middle| \, \underline{x} \in X, \underline{y} \in Y \right] \\ &= \frac{\displaystyle\int_X \int_Y \begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix} p(\underline{x}, \underline{y}) \, d\underline{x} d\underline{y}}{\displaystyle\int_X \int_Y p(\underline{x}, \underline{y}) \, d\underline{x} d\underline{y}} \end{aligned} \tag{5.3.2}$$

This can be rather difficult to calculate, however, except for certain simple types of constraint sets X and Y.

Fortunately, our four iterative algorithms are much easier to apply to this problem. The reason is that this density $p(\underline{x}, \underline{y})$ forms an exponential family of distributions which is in its "natural" form:

$$\begin{aligned} p(\underline{x}, \underline{y}) &= \left[\frac{1}{|2\pi Q|^{½} |2\pi R|^{½}} \exp\left[-½ \left(\|A\underline{x}\|_Q^2 + \|B\underline{x}\|_R^2 \right) \right] \right] \\ &\qquad \cdot \left[\exp\left(-½ \|\underline{y}\|_R^2 \right) \right] \exp\left[-\underline{y}^T R^{-1} B \underline{x} \right] \\ &= g(\underline{x}) h(\underline{y}) \exp\left[-\underline{y}^T R^{-1} B \underline{x} \right] \end{aligned} \tag{5.3.3}$$

where these terms can be defined in an obvious way. Thus we would immediately expect our four iterative algorithms to take the simple form discussed in chapter 3, section 7.2. Another important point is that this density $p(\underline{x}, \underline{y})$ is uniformly log concave in $\underline{x}$ and $\underline{y}$, a fact which will greatly simplify our convergence proofs.

4. Minimum Cross-Entropy Method (MCEM)

Because there are two unknowns, $\underline{x}$ and $\underline{y}$, there will be at least four different iterative estimation approaches we can devise using the material of chapters 2 and 3. The Minimum Cross-Entropy Method fits a separable probability density $q_X(\underline{x})q_Y(\underline{y})$ to the given density $p(\underline{x},\underline{y})$ by minimizing the cross-entropy function $H(q_X,q_Y)$:

$$\hat{q}_X,\hat{q}_Y \leftarrow \min_{q_X,q_Y} H(q_X,q_Y) \tag{5.4.1}$$

$$\leftarrow \min_{q_X,q_Y} \int_X\int_Y q_X(\underline{x})q_Y(\underline{y}) \log \frac{q_X(\underline{x})q_Y(\underline{y})}{p(\underline{x},\underline{y})}\, d\underline{x}d\underline{y}$$

To solve this problem, we iteratively minimize H with respect to q_X, then q_Y, iterating back and forth until the estimates converge:

$$\hat{q}_{X_k} \leftarrow \min_{q_X} H(q_X,\hat{q}_{Y_{k-1}}) \tag{5.4.2}$$

$$\hat{q}_{Y_k} \leftarrow \min_{q_Y} H(\hat{q}_{X_k},q_Y)$$

Substituting the formula (5.2.2) for $p(\underline{x},\underline{y})$, it is easy to show that the estimated densities will have the form:

$$\hat{q}_{X_k}(\underline{x}) = p_{X|Y}(\underline{x}\,|\,\hat{\underline{y}}_{k-1}) \tag{5.4.3}$$

$$= \begin{cases} K_x \exp\left(-\tfrac{1}{2}\,\|\underline{x}-H\hat{\underline{y}}_{k-1}\|_V^2\right) & \text{for } \underline{x}\in X \\ 0 & \text{else} \end{cases}$$

$$\text{where: } V = \left[A^TQ^{-1}A + B^TR^{-1}B\right]^{-1}$$

$$H = VB^TR^{-1}$$

and:

$$\hat{q}_{Y_k}(y) = p_{Y|X}(y|\hat{x}_k) = \begin{cases} K_y \exp\left(-\tfrac{1}{2}\|y - B\hat{x}_k\|_R^2\right) & \text{for } y \in Y \\ 0 & \text{else} \end{cases} \tag{5.4.4}$$

The estimated densities $\hat{q}_{X_k}$ and $\hat{q}_{Y_k}$ are truncated Gaussians centered at $H\hat{y}_k$ and $B\hat{x}_k$ respectively, where these centers are calculated as follows:

```
MCEM Iterative Algorithm
Guess $\hat{y}_0 \in Y$
For $k = 0, 1, \cdots$
        $\hat{x}_{k+1} = E_X\left[x \,\middle|\, \hat{q}_{X_{k+1}}\right]$
        $\hat{y}_{k+1} = E_Y\left[y \,\middle|\, \hat{q}_{Y_{k+1}}\right]$
```

Conceptually, the computation proceeds through a series of filtering and conditional expectation operators. Start with an output estimate $\hat{y}_k$. In order to compute $\hat{x}_{k+1}$ we must first compute the center $H\hat{y}_k$ of the signal density $\hat{q}_{X_{k+1}}(x)$. This operation corresponds to a standard least squares filtering operation on $\hat{y}_k$, and would be our best estimate of x given $\hat{y}_k$ *if* there were no constraints on x, i.e. if $X = \mathbf{R}^N$. In general, however, additional knowledge is available concerning the signal, so that X is a proper subset of $\mathbf{R}^N$. Thus we calculate the conditional expectation of x over the set X given the truncated Gaussian density $\hat{q}_{X_{k+1}}(x)$ centered at $H\hat{y}_k$. According to theorem 2.2.1, this estimate $\hat{x}_{k+1}$ belongs to the closed convex hull of X. Now to estimate the output, pass this signal estimate through the filter B to form an output estimate $B\hat{x}_{k+1}$. If there were no constraints on the output values, $Y = \mathbf{R}^M$, then this would be the best least squares estimate of y given $\hat{x}_{k+1}$. In general, however, more is known about the output, so that Y is a proper subset of $\mathbf{R}^M$. Thus we calculate the expectation of y over the set Y given the truncated Gaussian density $\hat{q}_{Y_{k+1}}(y)$ centered at $B\hat{x}_{k+1}$. This estimate $\hat{y}_{k-1}$ is in the closed convex hull of Y. We now iterate, refiltering the new output estimate and then recalculating the expectation of x to improve our next signal estimate. The algorithm thus iterates between a classic least squares filtering operation followed by an expectation operation to estimate $\hat{x}_{k-1}$, followed by another filtering and expectation operation to calculate $\hat{y}_{k+1}$. Each iteration decreases the cross-entropy, and thus improves the estimates. It is straightforward to show that:

$$H(\hat{q}_{X_{k+1}}, \hat{q}_{Y_{k+1}}) = \log \frac{p(\hat{x}_{k-1}, \hat{y}_k)}{p(X, \hat{y}_k)\, p(\hat{x}_{k-1}, Y)} \qquad (5.4.5)$$

and thus this combination of densities must also decrease on each pass. Finally, Appendices D and E prove that if X and Y are convex sets, then the conditional expectation operator of a truncated Gaussian is a non-expansive mapping while the filtering operation is a contraction mapping. As a result, when X and Y are convex, the MCEM algorithm is guaranteed to converge at a geometric rate to the unique solution x_*, y_* to the MCEM problem:

$$\begin{aligned} \|\hat{x}_{k+1} - x_*\|_V &\leq \nu_x \, \|\hat{y}_k - y_*\|_R \\ \|\hat{y}_{k-1} - y_*\|_R &\leq \nu_y \, \|\hat{x}_{k+1} - x_*\|_V \end{aligned} \qquad (5.4.6)$$

where ν_x, ν_y are constants less than one which are determined by the relative signal-to-noise level:

$$\begin{aligned} \nu_x &= \left(\frac{\mu_{max}}{\mu_{max}+1} \right)^{\frac{1}{2}} < 1 \\ \nu_y &= \left(\frac{\lambda_{max}}{\lambda_{max}+1} \right)^{\frac{1}{2}} < 1 \end{aligned} \qquad (5.4.7)$$

$$\text{where: } \mu_{max} = \max_{y \neq 0} \frac{y^T B A^{-1} Q A^{-T} B^T y}{y^T R y}$$

$$\lambda_{max} = \max_{x \neq 0} \frac{x^T B^T R^{-1} B x}{x^T A^T Q^{-1} A x}$$

These formulas for the convergence rate factors ν_x and ν_y simplify in the case where the noise is white, $R=\sigma^2 I$, the dimensions of x and y are equal, $N=M$, and $B=I$:

$$\nu_x = \nu_y = \left(\frac{\tau_{max}/\sigma^2}{\tau_{max}/\sigma^2+1} \right)^{1/2} \tag{5.4.8}$$

where τ_{max} = the largest eigenvalue of $A^{-1}QA^{-T}$

Beware that at high signal-to-noise levels, $\tau_{max} \gg \sigma^2$, and thus the convergence rate may be extremely slow, $\nu_x \nu_y \approx 1$. This phenomenon, as pointed out in section 2, is intimately related to the possible numerical ill-behavior of the model density $p(x,y)$ itself at high signal-to-noise levels.

If the sets X and Y are not convex, then the algorithm is no longer guaranteed to give estimates of $\hat{x}_k$ and $\hat{y}_k$ which are elements of X and Y, the convergence rate is no longer guaranteed to be geometric, and the solution is no longer guaranteed to be unique. We would still expect the measures $\hat{Q}_{X_k}$ and Qvh_k to converge to limits which are stochastically bounded and are not impulse-like. This would seem to imply that the center of these densities would have to be bounded and thus also converge, but it appears difficult to prove this formally. If, however, the estimates $\hat{x}_k$ and $\hat{y}_k$ do remain bounded, then theorem 3.9.6 guarantees that they must converge to the set of stationary points of the algorithm, and critical points of the cross-entropy.

5. Maximum A Posteriori Signal Estimation (XMAP)

A more conventional approach to estimating the signal x is to try to find its most likely value in X, given that the output y is somewhere in Y. Such a Maximum A Posteriori approach can be realized by integrating the density $p(x,y)$ over the output constraint space Y, and then maximizing the result over the signal constraint space:

$$\text{XMAP:} \quad \hat{x} - \max_{x \in X} p(x,Y) = \max_{x \in X} \int_Y p(x,y)\,dy \tag{5.5.1}$$

Unfortunately, unless Y has certain special forms, this density $p(x,Y)$ will not be Gaussian, and it will be difficult either to compute or to maximize.

Applying our iterative approach to solving this problem, however, is relatively simple. As shown in chapter 2, an approach which is equivalent to solving (5.5.1) is to iteratively minimize the cross-entropy function $\mathcal{H}(q_X, q_Y)$ in (5.4.1) but with the constraint that the signal density must be an impulse function, $\hat{q}_X(x) = \delta(x - \hat{x})$. The resulting output density estimate will have the same truncated Gaussian form as (5.4.4):

$$\hat{q}_{Y_k}(y) = p_{Y|X}(y|\hat{x}_k) \tag{5.5.2}$$

where the mean $\hat{x}_k$ is found by solving:

$$\begin{aligned} \hat{x}_{k+1} &= \max_{x \in X} E_Y\left[\log \frac{p(x,y)}{\hat{q}_{Y_k}(y)} \,\middle|\, \hat{q}_{Y_k}(y) \right] \\ &= \min_{x \in X} E_Y\left[\|x - Hy\|_V^2 \,\middle|\, \hat{x}_k \right] \\ &= \min_{x \in X} \int_Y \|x - Hy\|_V^2 \, p_{Y|X}(y|x)\, dy \end{aligned} \tag{5.5.3}$$

where the expectation operator $E_Y[\cdot|x]$ is defined by this last line. This algorithm thus effectively chooses the signal estimate $\hat{x}_{k+1}$ in X which minimizes the distance to the filtered output estimate Hy, where we average over all possible output values $y \in Y$. Because the correct signal value is unknown, this averaging process is imperfect, and thus the algorithm iterates, using improved signal density estimates to improve the averaging on the next pass. Since $\|x - Hy\|_V^2$ is only quadratic in y, and because $p_{Y|X}(y|x)$ is a truncated Gaussian, the expectation and minimum is easily calculated. The resulting algorithm has the form:

```
XMAP Iterative Algorithm
Guess $\hat{\underline{x}}_0 \in X$
For $k=0,1,\cdots$
    $\hat{\underline{y}}_k = E_Y[\underline{y} \mid \hat{\underline{x}}_k] = \int_Y \underline{y}\, p_{Y|X}(\underline{y} \mid \hat{\underline{x}}_k)\, d\underline{y}$
    $\hat{\underline{x}}_{k+1} \leftarrow \min_{\underline{x} \in X} \|\underline{x} - H\hat{\underline{y}}_k\|_V^2.$
```

First of all, to estimate the output, we pass the signal estimate $\hat{\underline{x}}_k$ through the filter B to calculate the best least squares unconstrained estimate of $\underline{y}$. To accommodate the additional knowledge that $\underline{y} \in Y$, we then calculate the conditional expectation of the output $\underline{y}$ over the set Y of the truncated Gaussian $\hat{q}_{Y_k}(\underline{y})$ centered at $B\hat{\underline{x}}_k$. To reestimate the signal, this output estimate is then filtered with the matrix H to calculate the best least squares unconstrained signal estimate, $H\hat{\underline{y}}_k$. In general, this estimate does not satisfy the known constraints on the signal, and so to estimate $\hat{\underline{x}}_{k+1}$ we find the element of X closest to $H\hat{\underline{y}}_k$. (This operation can be viewed as "projecting" the estimate $H\hat{\underline{y}}_k$ onto the constraint set X.) If the space X is not convex, there could be several such signal values; we assume that some deterministic rule is used to choose one of these. Each iteration decreases the cross-entropy and increases the likelihood function $p(\hat{\underline{x}}_k, Y)$ and thus produces a "better" signal estimate. Note in particular, that the only difference between XMAP and MCEM is that rather than choose the signal estimate by calculating the expectation over X of a Gaussian centered at $H\hat{\underline{y}}_k$, we instead simply choose the value in X closest to $H\hat{\underline{y}}_k$. (In effect, we estimate $\underline{x}$ by finding the *mode* of the density $p_{X|Y}(\underline{x} \mid \hat{\underline{y}}_k)$ rather than its *mean*.)

To prove convergence of the algorithm, note that not only $p(\underline{x}, \underline{y})$ but also $p(\underline{x})$ and $p(\underline{y})$ are uniformly log concave. Since $p(\underline{x}, Y) = p(Y \mid \underline{x})\, p(\underline{x}) \leq p(\underline{x})$, the likelihood $p(\underline{x},Y)$ must drop to zero as $\|\underline{x}\|\to\infty$. Each iteration of the algorithm increases the likelihood function $p(\hat{\underline{x}}_k,Y)$, however, and thus the estimates $\hat{\underline{x}}_k$ must remain bounded. Our convergence theorems then guarantee that the sequence of estimates $\hat{\underline{x}}_k$ must converge to the set of critical points of $p(\underline{x},Y)$ and/or local maxima on the boundary of X.

If X and Y are both convex, much more can be proven. By Prekopa's theorem (see Appendix D), $p(Y|\underline{x})$ will also be log concave, and thus $p(\underline{x},Y)=p(Y|\underline{x})p(\underline{x})$ must be uniformly log concave. Theorem 2.10.4 then guarantees that $p(\underline{x},Y)$ must have a unique global and local maximum, which can also be the only critical point. The algorithm must therefore converge to this unique XMAP solution. Appendix E proves in addition that projection operators and conditional expectation operators of truncated Gaussians are both non-expansive mappings, while the filters are contraction mappings. This leads to the conclusion that the signal and output estimates $\hat{\underline{x}}_k$, $\hat{\underline{y}}_k$ converge at a geometric rate to the unique XMAP solution $\underline{x}_*$, $\underline{y}_*$:

$$\|\hat{\underline{y}}_{k+1}-\underline{y}_*\|_R \le \nu_y \; \|\hat{\underline{x}}_{k+1}-\underline{x}_*\|_V \le \nu_x\nu_y \; \|\hat{\underline{y}}_k-\underline{y}_*\|_R \tag{5.5.4}$$

where ν_x, ν_y are exactly the same convergence factors as in the MCEM algorithm.

6. Maximum A Posteriori Output Estimation (YMAP)

Yet another approach to estimating the unknown is to try to find the most likely value of the output $\underline{y}$ in the set Y, given that the signal $\underline{x}$ is somewhere in X:

$$\text{YMAP:} \qquad \hat{\underline{y}} - \max_{\underline{y}\in Y} p(X,\underline{y}) = \max_{\underline{y}\in Y} \int_X p(\underline{x},\underline{y})\,d\underline{x} \tag{5.6.1}$$

This, of course, is identical to the XMAP algorithm except with the roles of $\underline{x}$ and $\underline{y}$ reversed. We would thus expect YMAP to behave similarly to XMAP; in particular, the function $p(X,\underline{y})$ will usually be difficult to compute and unpleasant to optimize.

One approach to solving this problem is to exploit the relationship between cross-entropy and MAP which we discussed in chapters 2 and 3. Solving the YMAP problem (5.6.1) is equivalent to minimizing the cross-entropy $H(q_X, q_Y)$ given in (5.4.1) but with the constraint that the output density estimate must be an impulse function, $\hat{q}_Y(\underline{y}) = \delta(\underline{y} - \hat{\underline{y}})$. The signal density estimate $\hat{q}_X(\underline{x})$ will be a truncated Gaussian, exactly as in the MCEM algorithm.

$$\hat{q}_{X_{k+1}}(\underline{x}) = p_{X|Y}(\underline{x}\,|\hat{\underline{y}}_k) \tag{5.6.2}$$

and the location of the output density impulse $\delta(\underline{y} - \hat{\underline{y}}_k)$ will be found by solving:

$$\begin{aligned} \hat{\underline{y}}_{k+1} &= \max_{\underline{y} \in Y} \mathrm{E}_X \left[\log \frac{p(\underline{x},\underline{y})}{\hat{q}_{X_{k+1}}(\underline{x})} \,\middle|\, \hat{q}_{X_{k+1}} \right] \\ &= \min_{\underline{y} \in Y} \mathrm{E}_X \left[\|\underline{y} - \mathrm{B}\underline{x}\|_{\mathrm{R}}^2 \,\middle|\, \hat{\underline{y}}_k \right] \\ &= \min_{\underline{y} \in Y} \int_X \|\underline{y} - \mathrm{B}\underline{x}\|_{\mathrm{R}}^2 \, p_{X|Y}(\underline{x}\,|\hat{\underline{y}}_k)\, d\underline{x} \end{aligned} \tag{5.6.3}$$

where $\mathrm{E}_X[\cdot\,|\hat{\underline{y}}_k]$ is defined by this last line. Thus we choose each output estimate $\hat{\underline{y}}_{k+1} \in Y$ to minimize the average distance to the filtered signal value $\mathrm{B}\underline{x}$. Since the "correct" value of $\hat{\underline{y}}_k$ is unknown, this averaging process is imperfect, and so the algorithm iterates, using each new output density estimate to improve the averaging on each pass. Because the expression in (5.6.3) is quadratic in both $\underline{x}$ and $\underline{y}$, it is easy to compute the expectation and to solve the minimization. The resulting algorithm takes the form:

```
YMAP Iterative Algorithm:
Guess $\hat{y}_0 \in Y$
For $k = 0, 1, \cdots$
    $\hat{x}_{k+1} = E_X \left[ x \,\middle|\, \hat{y}_k \right]$
    $\hat{y}_{k+1} - \min_{y \in Y} ||y - B\hat{x}_{k+1}||_R^2$
```

We start with an estimate of the output $\hat{y}_k$. Filter this, $H\hat{y}_k$, to get the best least squares unconstrained estimate of the signal. The additional information that $x \in X$ is accommodated by estimating $\hat{x}_{k+1}$ as the mean of a truncated Gaussian distribution over X centered at $H\hat{y}_k$. The output is then reestimated by passing this signal estimate through B, then finding the element in Y which comes closest to this. On the next pass, the new output estimate is filtered and used to find an even better signal estimate. Each iteration decreases the cross-entropy and increases the likelihood function $p(X, y)$, and thus improves the estimates.

To prove convergence, note that $p(X, y) = p(X | y) p(y) \leq p(y)$. Since $p(y)$ is uniformly log concave, $p(y) \to 0$ as $||y|| \to \infty$. Since YMAP increases $p(X, y)$ on each pass, the estimates $\hat{y}_k$ must be bounded. Since $p(Y, y)$ is continuously differentiable, the discussion in chapter 3, section 9.3 implies that $\hat{y}_k$ must therefore converge to the set of stationary points of the algorithm and critical points or local maxima of the likelihood function $p(X, y)$.

If in addition X and Y are convex, then since $p(x | y)$ is log concave, Prékopa's theorem (see Appendix D) guarantees that $p(X | y)$ is also log concave. Since $p(x)$ is uniformly log concave then $p(X, y) = p(X | y) p(y)$ must also be uniformly log concave. With Y convex, by theorem 2.10.4 $p(X, y)$ must therefore have a unique global and local maximum which can also be the only critical point. The iterative algorithm is therefore guaranteed to converge to this unique global maximum, $\underline{x}_\bullet$, $\underline{y}_\bullet$. The convergence rate can be proven to be geometric in the same way as before, with:

$$\|\hat{\underline{y}}_{k+1} - \underline{y}_\bullet\|_R \le \nu_y \ \|\hat{\underline{x}}_{k+1} - \underline{x}_\bullet\|_V \le \nu_x \nu_y \ \|\hat{\underline{y}}_k - \underline{y}_\bullet\|_R \qquad (5.6.4)$$

where ν_x, ν_y are the same convergence rate constants as in MCEM (5.4.7).

7. MAP Simultaneous Signal and Output Estimation (XYMAP)

The last estimation method we will discuss estimates the signal and output by finding the combination of values $\hat{\underline{x}}$ and $\hat{\underline{y}}$ which jointly maximize $p(\underline{x},\underline{y})$:

$$\text{XYMAP:} \qquad \hat{\underline{x}}, \hat{\underline{y}} - \max_{\underline{x} \in X, \underline{y} \in Y} p(\underline{x},\underline{y}) \qquad (5.7.1)$$

Substituting the log probability density (5.2.2) into this and simplifying yields:

$$\hat{\underline{x}}, \hat{\underline{y}} - \min_{\underline{x} \in X, \underline{y} \in Y} \left\{ \|A\underline{x}\|_Q^2 + \|\underline{y} - B\underline{x}\|_R^2 \right\} \qquad (5.7.2)$$

XYMAP is thus equivalent to a least squares minimization problem which tries to find the signal $\underline{x}$ with the least possible energy $\|A\underline{x}\|_Q^2$ for which $B\underline{x}$ also comes as close as possible to being a feasible output value $\underline{y} \in Y$. Despite the fact that this objective function is quadratic, it is usually difficult to solve this directly due to the constraint that $\underline{x} \in X$, $\underline{y} \in Y$. Iteratively minimizing this density first with respect to $\underline{x}$ and then with respect to $\underline{y}$ is often much simpler:

```
XYMAP Iterative Algorithm:
Guess $\hat{\underline{y}}_0 \in Y$
For $k = 0, 1, \cdots$
    $\hat{\underline{x}}_{k+1} - \min_{\underline{x} \in X} \|\underline{x} - H\hat{\underline{y}}_k\|_V^2$
    $\hat{\underline{y}}_{k+1} - \min_{\underline{y} \in Y} \|\underline{y} - B\hat{\underline{x}}_{k+1}\|_R^2$
```

We start with an initial estimate of the output. Filter this estimate with the matrix H to find the best least squares unconstrained estimate $H\hat{y}_k$ of the signal given $\hat{y}_k$. Since this is usually not an element of X, we use the element in X closest to $H\hat{y}_k$ as our signal estimate $\hat{x}_{k-1}$ (this corresponds to "projecting" the filtered output estimate $H\hat{y}_k$ onto the space X.) Next we process this signal estimate through B to find the best least squares unconstrained estimate of the output, $B\hat{x}_{k+1}$. Since this usually does not meet the output constraints, we use the element in Y closest to $B\hat{x}_{k+1}$ as our output estimate $\hat{y}_{k+1}$ (this corresponds to projecting $B\hat{x}_{k+1}$ onto the output space Y.) If X or Y are not convex, the projection operators may have multiple solutions; in this case, we use some deterministic rule to pick one. On the next pass, the improved output estimate is used to get a better signal estimate. Each iteration increases the likelihood function $p(\hat{x}_k, \hat{y}_k)$, and thus improves the estimates. To prove convergence, note that $p(\underline{x}, \underline{y})$ is a uniformly log concave function, and thus goes to zero as $||\underline{x}|| \to \infty$ or $||\underline{y}|| \to \infty$. Since XYMAP increases $p(\hat{x}_k, \hat{y}_k)$ on every iteration, the estimates must remain bounded, and since $p(\underline{x}, \underline{y})$ is continuously differentiable, our convergence theorems guarantee that the estimates will converge to the set of critical points or local maxima of $p(\underline{x}, \underline{y})$.

If X and Y are also convex, then each step of the iteration is guaranteed to have a unique solution, and Appendix E proves that the estimates converge at a geometric rate to the unique global maximum $\underline{x}_*$, $\underline{y}_*$ of $p(\underline{x}, \underline{y})$:

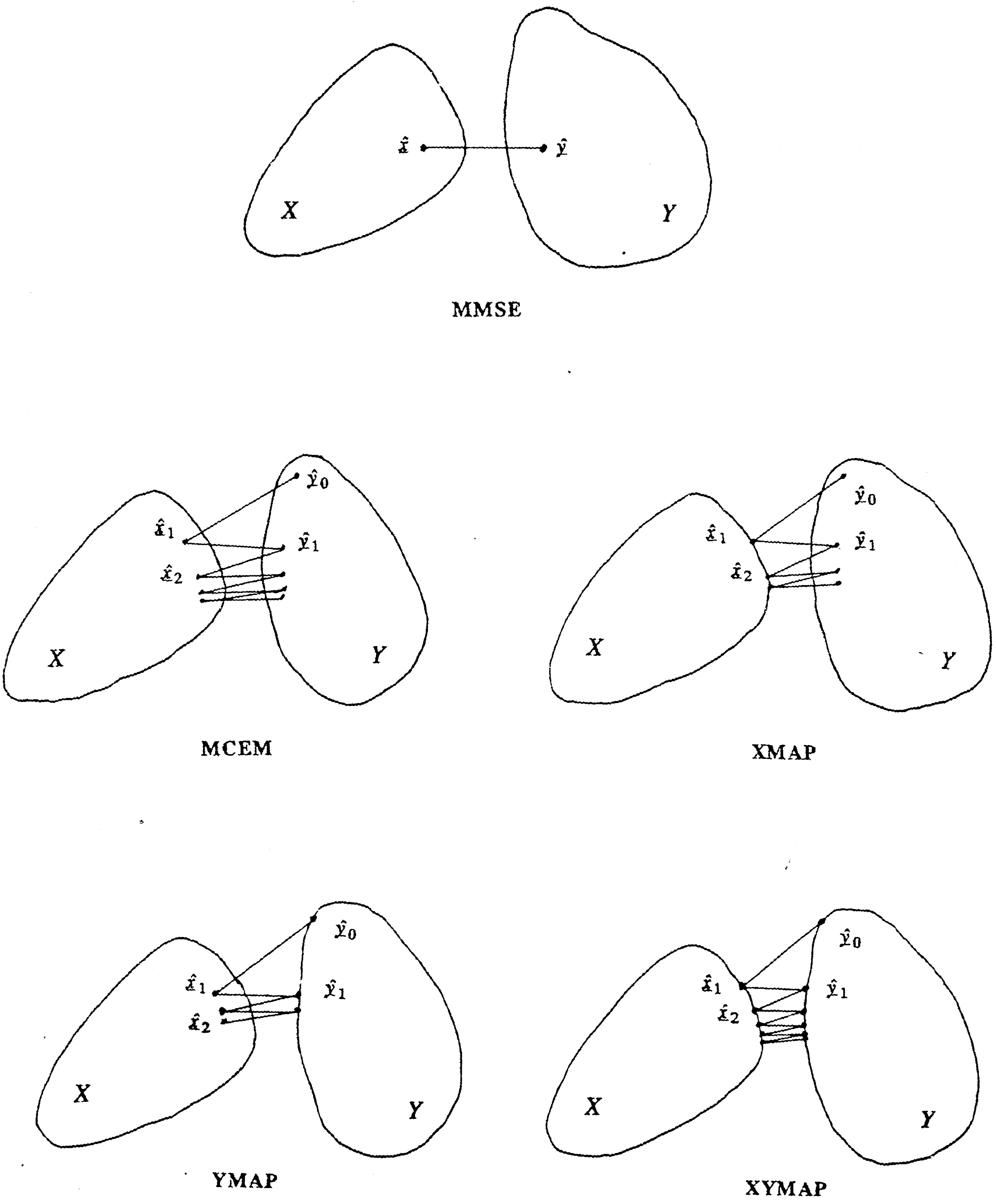
$$||\hat{y}_{k+1} - \underline{y}_*||_R \le \nu_y \; ||\hat{x}_{k+1} - \underline{x}_*||_V \le \nu_x \nu_y \; ||\hat{y}_k - \underline{y}_*||_R \tag{5.7.3}$$

where ν_x, ν_y are the same convergence rate constants (5.4.7) as in all our other algorithms.

8. Comparison of the Algorithms

Unlike the parameter estimation problems of chapter 4, it will generally not be possible to find an asymptotically consistent and efficient estimate of an unknown signal. Judging the algorithms on the basis of their asymptotic properties is thus not possible. A more workable approach is to define the MMSE estimates as the "best" we can do, and then compare our iterative algorithms to see how closely they match MMSE.

To a large extent, the behavior of our algorithms can be predicted (albeit with much hand waving) from the drawings of typical convergence patterns shown in figure 5.2. MMSE, which calculates the expectation of $\underline{x} \in X$ and $\underline{y} \in Y$, will give estimates $\hat{\underline{x}}$, $\hat{\underline{y}}$ which are in the interior of X and Y, and located near the closest meeting of these two spaces. MCEM, alternating between filtering and conditional expectation calculations, will also give estimates in the interior of X and Y located near the closest meeting of the spaces. We might therefore expect its estimates to be "near" those of MMSE. XMAP alternates between calculating the expectation of $\underline{y}$, and projecting the filtered estimate $H\hat{\underline{y}}_k$ onto the signal constraint space X. The output estimate $\hat{\underline{y}}_k$ will be in the interior of the set Y, while the signal estimate $\hat{\underline{x}}_k$ will usually be on the boundary of X as near to $H\hat{\underline{y}}_k$ as possible. Paradoxically, we would therefore expect the XMAP output estimate $\hat{\underline{y}}$ to be close to the MMSE solution, while its signal estimate (the original goal of the algorithm) will be far from the MMSE estimate. YMAP alternates between projecting the filtered signal $B\hat{\underline{x}}_k$ onto Y, then calculating the conditional expectation of $\underline{x}$ in X. The $\hat{\underline{x}}_k$ estimates will be in the interior of X while the $\hat{\underline{y}}_k$ estimate will generally be on the boundary of Y as near to $B\hat{\underline{x}}_k$ as possible. We would therefore expect YMAP's estimates of the signal $\hat{\underline{x}}$ to be near the MMSE estimate, but the output estimate (the original goal of the algorithm) will be far from the MMSE estimate. Finally, XYMAP alternates between filtering steps and projections onto X and onto Y. Both signal and output estimates will generally be on the boundaries of the constraint sets,



MMSE

MCEM

XMAP

YMAP

XYMAP

Figure 5.2 - Convergence Patterns of Reconstruction Algorithms

and the algorithm will converge to the pair of points in X and Y which are as close to each other as possible and have as little energy as possible. Both estimates of $\hat{\underline{x}}$ and $\hat{\underline{y}}$ will thus be far from the MMSE estimates. In general, therefore, we might expect MCEM to come closest to MMSE. XMAP will give good output estimates, YMAP will give good signal estimates, and XYMAP will give poor estimates of both quantities. This must be balanced against the fact that MCEM is the most difficult to calculate, as it requires two conditional expectations per pass, while XYMAP is the simplest, since it needs no conditional expectations at all.

To give some concrete basis to this handwaving, we will consider an illuminating example, which happens to have direct bearing on a phase-only reconstruction algorithm we will consider in chapter 7. Let us suppose that $\underline{x}$ and $\underline{y}$ are 2 element vectors in R^2 generated by a linear Gaussian system like the one in figure 5.1 with parameters:

$$\begin{aligned} &N = M = 2 \\ &\mathrm{A} = \mathrm{B} = \mathrm{I} \\ &\mathrm{Q} = q\mathrm{I} \quad ; \quad \mathrm{R} = r\mathrm{I} \end{aligned} \tag{5.8.1}$$

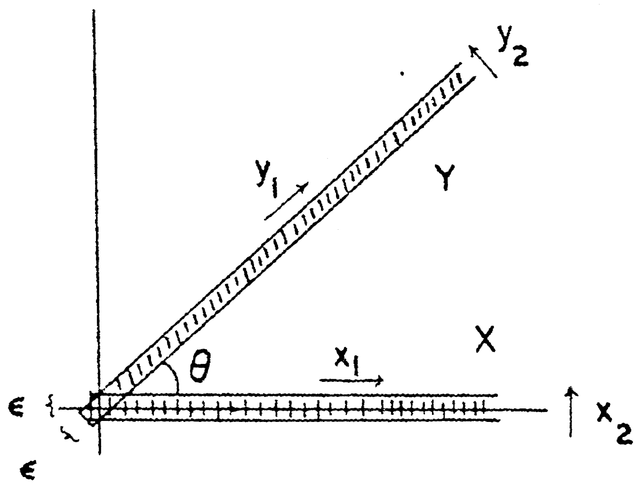
so that:

$$\mathrm{p}(\underline{x},\underline{y}) = \frac{1}{(2\pi)^2 qr} \exp\left\{ -\tfrac{1}{2} \left[\frac{1}{q}\underline{x}^{\mathrm{T}}\underline{x} + \frac{1}{r}(\underline{y}-\underline{x})^{\mathrm{T}}(\underline{y}-\underline{x}) \right] \right\} \tag{5.8.2}$$

Suppose that the observation data available indicates only that the signal $\underline{x}$ is somewhere in an infinitely long and narrow strip which starts at the origin, is oriented at an angle of 0, and has extremely narrow width $\epsilon>0$. This observation data defines the constraint space X. The output $\underline{y}$ is observed to lie on another infinitely long and narrow strip which starts at the origin, is oriented at angle $|\theta|\le\frac{\pi}{2}$, and also has width ϵ. This defines the constraint space Y.

$$X = \left\{ x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| x_1 \geq 0 \quad \text{and} \quad -\frac{\epsilon}{2} \leq x_2 \leq \frac{\epsilon}{2} \right\} \tag{5.8.3}$$

$$Y = \left\{ y = y_1 \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} + y_2 \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix} \middle| y_1 \geq 0 \quad \text{and} \quad -\frac{\epsilon}{2} \leq y_2 \leq \frac{\epsilon}{2} \right\}$$



Example - Constraint Sets X and Y

Technically, it is necessary to give sets X and Y non-zero width in order to ensure that their measures will be non-zero. (Other methods could also be used for dealing with this problem.) Computationally, we will simply set $\epsilon=0$, and estimate $\hat{x}_2=\hat{y}_2=0$.

Let $v = \frac{qr}{q+r}$ and $h = \frac{v}{r}$ (these correspond to the matrices H and V in (5.4.3).) Substituting the probability density (5.8.2) into our iterative procedure and simplifying gives the four algorithms summarized in the table below. Because the constraint spaces are simple half-lines, the projection operators are particularly easy to compute in the MAP algorithms. The expectation operators, however, are somewhat more complicated, since they involve evaluating an error function (this is a standard subroutine in most numerical software libraries.)

Method	Signal Estimate	Output Estimate
MCEM:	$\hat{x}_{1,k} = h\hat{y}_{1,k-1}\cos\theta + \xi(\alpha)$	$\hat{y}_{1,k} = \hat{x}_{1,k}\cos\theta + \zeta(\beta)$
XMAP:	$\hat{x}_{1,k} = h\hat{y}_{1,k-1}\cos\theta$	$\hat{y}_{1,k} = \hat{x}_{1,k}\cos\theta + \zeta(\beta)$
YMAP:	$\hat{x}_{1,k} = h\hat{y}_{1,k-1}\cos\theta + \xi(\alpha)$	$\hat{y}_{1,k} = \hat{x}_{1,k}\cos\theta$
XYMAP:	$\hat{x}_{1,k} = h\hat{y}_{1,k-1}\cos\theta$	$\hat{y}_{1,k} = \hat{x}_{1,k}\cos\theta$

where:

$$\begin{aligned} \xi(\alpha) &= \sqrt{v}\,\frac{\exp(-\frac{1}{2}\alpha^2)}{0.5+\mathrm{erf}(\alpha)} \qquad \text{with} \quad \alpha = \frac{1}{\sqrt{v}} h\hat{y}_{1,k-1}\cos\theta \\ \zeta(\beta) &= \sqrt{r}\,\frac{\exp(-\frac{1}{2}\beta^2)}{0.5+\mathrm{erf}(\beta)} \qquad \text{with} \quad \beta = \frac{1}{\sqrt{r}} \hat{x}_{1,k}\cos\theta \end{aligned} \tag{5.8.4}$$

and:

$$\mathrm{erf}(\omega) = \frac{1}{\sqrt{2\pi}} \int_0^{\omega} \exp(-\tfrac{1}{2}\tau^2)\, d\tau \tag{5.8.5}$$

Because this example is so simple, it is also possible to calculate the exact solutions of all four iterative algorithms, as well as MMSE:

	$\frac{\hat{x}_1}{\sqrt{v}}$	$\frac{\hat{y}_1}{\sqrt{r}}$
MMSE:	$\psi(\lambda)$	$\psi(\lambda)$
MCEM:	$\eta(\lambda)$	$\eta(\lambda)$
XMAP:	$\lambda\sqrt{r}\,\eta(\lambda^2)$	$\eta(\lambda^2)$
YMAP:	$\eta(\lambda^2)$	$\lambda\sqrt{v}\,\eta(\lambda^2)$
XYMAP:	0	0

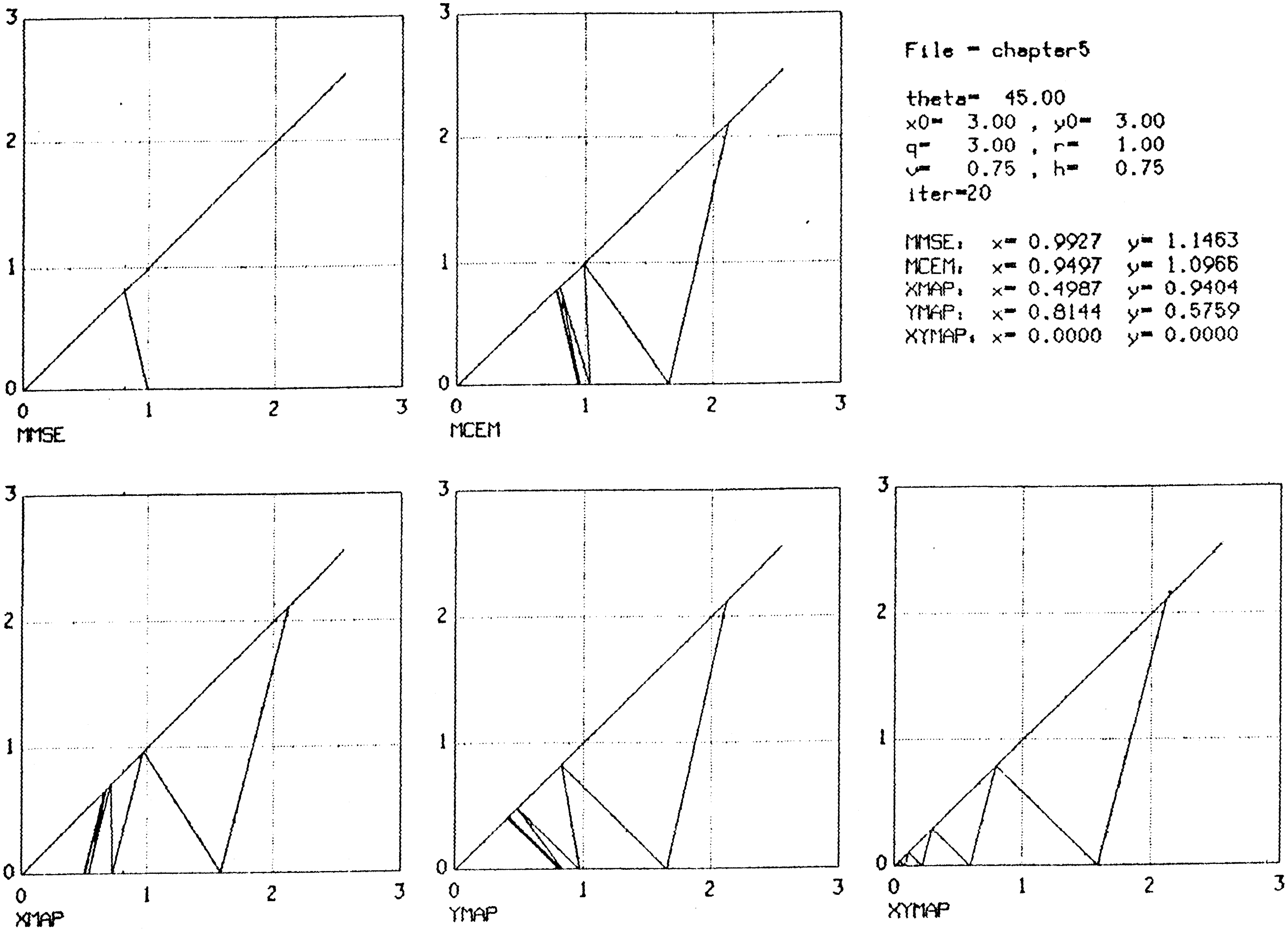
where $\lambda = \left(\frac{v}{r} \right)^{\frac{1}{2}} \cos\theta$, $\eta(\lambda)$ is the solution to:

$$\eta(\lambda) = \frac{1}{1-\lambda} \frac{\exp(-\frac{1}{2}\lambda^2\eta^2(\lambda))}{\int\limits_0^\infty \exp(-\frac{1}{2}(x-\lambda\eta(\lambda))^2)\, dx} \tag{5.8.6}$$

and $\psi(\lambda)$ is the MMSE solution:

$$\psi(\lambda) = \frac{1}{\frac{2}{\sqrt{2\pi}}(1-\lambda)\int\limits_0^\infty\int\limits_0^\infty \exp\left(-\frac{1}{2}(x^2+y^2-2\lambda xy) \right) dxdy} \tag{5.8.7}$$

Figure 5.3 also illustrates the convergence and final solutions of the five estimation algorithms for the parameter values $q=3$, $r=1$, $\theta=45°$. MMSE yields the "best" estimates in the sense that its estimates are unbiased and enjoy the pleasant symmetry $\frac{\hat{x}_1}{\sqrt{v}} = \frac{\hat{y}_1}{\sqrt{r}}$. Unfortunately, it requires evaluating a double integral. (Actually, scientific software libraries such as IMSL contain a subroutine call which evaluates this integral. In more complicated examples, however, the integral required is usually prohibitively difficult.) The MCEM algorithm requires evaluating an error function to calculate each of the conditional expectations on every pass, and thus requires more effort than any of the other iterative schemes. Its estimates, however, are the closest to the MMSE estimates, and they show the same pleasant symmetry $\frac{\hat{x}_1}{\sqrt{v}} = \frac{\hat{y}_1}{\sqrt{r}}$. Note from figure 5.3 that MCEM's convergence rate is geometric until near convergence. XYMAP is the simplest algorithm, in that each pass only solves simple projection problems. Note also the simple geometric convergence rate in figure 5.3. Unfortunately, in this problem the signal and output estimates which are as close as possible and have as little energy as possible, are $\hat{x}=\hat{y}=0$. Of all our iterative algorithms, these XYMAP



```
File = chapter5

theta=  45.00
x0=  3.00 , y0=  3.00
q=   3.00 , r=   1.00
v=   0.75 , h=   0.75
iter=20

MMSE:   x= 0.9927   y= 1.1463
MCEM:   x= 0.9497   y= 1.0966
XMAP:   x= 0.4987   y= 0.9404
YMAP:   x= 0.8144   y= 0.5759
XYMAP:  x= 0.0000   y= 0.0000
```

Figure 5.3 - Convergence of Estimates on Half-Line Constraint Sets

estimates are the farthest from the MMSE estimates. The XMAP and YMAP algorithms give intermediate performance and have intermediate difficulty, using one error function evaluation per pass. Paradoxically, as noted before, XMAP has the better output estimate and YMAP has the better signal estimate.

Our conclusion is that MMSE would be the method of choice if the computation required were not excessively difficult. The MCEM algorithm is next best, provided that the conditional expectations can be calculated easily. XMAP and YMAP are less work than MCEM, since they require only one conditional expectation calculation per pass rather than two, but their estimates can be more biased. Finally, XYMAP is the easiest to calculate, but can give estimates which have a large bias.

9. Linear Variety Constraint Sets

These optimal signal reconstruction algorithms simplify considerably when the constraint spaces X and Y are linear varieties defined by linear equality constraints on the signal and output values:

$$X = \left\{ \underline{x} \;\middle|\; G_x \underline{x} = \underline{\gamma}_x \right\} \qquad \text{where: } \begin{array}{l} G_x \text{ is a } p \times N \text{ matrix} \\ \underline{\gamma}_x \text{ is a } p \times 1 \text{ vector} \end{array}$$

$$Y = \left\{ \underline{y} \;\middle|\; G_y \underline{y} = \underline{\gamma}_y \right\} \qquad \text{where: } \begin{array}{l} G_y \text{ is a } q \times M \text{ matrix} \\ \underline{\gamma}_y \text{ is a } q \times 1 \text{ vector} \end{array}$$

We will assume that the constraints on $\underline{x}$ and $\underline{y}$ are consistent and independent so that G_x and G_y have full row rank, and so that the constraint sets are non-empty. In chapters 7 and 8 we discuss in great detail two applications of this type of model: bandlimited signal extrapolation, and reconstruction of a finite length signal from knowledge of the noisy phase or other projection of its Fourier Transform or its Short-time Fourier Transform. From a theoretical point of view, this type of model is interesting because much more powerful analysis techniques can be applied to this problem than are available for the more general case of convex constraint sets. In particular, we will prove:

a) MMSE and all four of our iterative MCEM and MAP algorithms all give the same unique estimate of $\hat{\underline{x}}$ and $\hat{\underline{y}}$, and all can be viewed as solving a constrained least squares problem.

b) Two categories of closed form solutions can be stated, corresponding to primal and dual optimization problems.

c) Two types of iterative algorithms can be stated, corresponding to the primal and dual problems, both of which are guaranteed to converge at a geometric rate to the unique globally optimum estimate.

d) The computational noise sensitivity of the algorithm can be analyzed, and can be shown to be directly related to the convergence rate of the algorithm and the ill-conditioning of the problem.

e) Each step of the iterative algorithm defines a linear mapping from X to Y and back again. The eigenvectors of the mapping form a complete orthonormal set, and the eigenvalues are all real, non-negative and less than $\nu_x \nu_y < 1$.

f) Acceleration methods are easily devised using line search extrapolation algorithms.

g) Both primal and dual problems can be transformed into problems which can be solved by PARTAN or a conjugate gradient algorithm. Each step of these methods is virtually identical to our accelerated algorithms, but convergence is achieved in a *finite* number of steps.

9.1. All Algorithms Give the Same Result

Because X and Y are linear varieties, integrating the Gaussian density $p(\underline{x},\underline{y})$ over these spaces simply yields another Gaussian. Thus $p_{X,Y}(\underline{x},\underline{y})$ is Gaussian, as is $p_{X|Y}(\underline{x}|\underline{y})$, $p_{Y|X}(\underline{y}|\underline{x})$ and all the other possible permutations. The signal and parameter density estimates will also be (non-truncated) Gaussians, $\hat{q}_X(\underline{x})=p_{X|Y}(\underline{x}|\hat{\underline{y}})$ and $\hat{q}_Y(\underline{y})=p_{Y|X}(\underline{y}|\hat{\underline{x}})$. Since the mean and the mode of a Gaussian coincide, the mean of $p_{X,Y}(\underline{x},\underline{y})$ (the MMSE estimate) is identical to the mode of $p_{X,Y}(\underline{x},\underline{y})$ (the XYMAP estimate). In addition, our iterative algorithms differ only in that some estimate $\hat{\underline{x}}$ or $\hat{\underline{y}}$ by calculating the means $E_{X|Y}[\underline{x}|\hat{\underline{y}}_k]$ or $E_{Y|X}[\underline{y}|\hat{\underline{x}}_k]$, while others effectively calculate the modes $\max_{\underline{x}\in X} p(\underline{x}|\hat{\underline{y}}_k)$ and $\max_{\underline{y}\in Y} p(\underline{y}|\hat{\underline{x}}_k)$. Since the means and modes of a Gaussian coincide, all four algorithms will generate exactly the same sequence of estimates if we start them at the same estimate, and all will converge to exactly the same answer as MMSE. In discussing linear variety constraint sets, it is therefore sufficient to focus on any one of the approaches we have discussed - we will choose XYMAP since it is the easiest to understand.

9.2. The Primal Iterative Algorithm

The XYMAP log likelihood function is:

$$\log p(\underline{x},\underline{y}) = -\frac{1}{2}\left\{ ||A\underline{x}||_Q^2 + ||\underline{y}-B\underline{x}||_R^2 + K \right\} \tag{5.9.1}$$

Because the constraint sets X and Y are closed, non-empty and convex, and because this density is uniformly log concave in $\underline{x}$ and $\underline{y}$, the XYMAP iterative algorithm is guaranteed to converge geometrically to the unique global maximum estimate at a rate $\nu_x\nu_y$ given in (5.4.7). To solve XYMAP, we iteratively maximize the density (5.9.1) with respect to $\underline{x}$ and then with respect to $\underline{y}$, using Lagrange Multiplier techniques to enforce the constraints $G_x\underline{x} = \underline{\gamma}_x$ and $G_y\underline{y} = \underline{\gamma}_y$. (See, for example, Luenberger. [1]) Given an output estimate $\hat{\underline{y}}_k$, we maximize $p(\underline{x}, \hat{\underline{y}}_k)$ over $\underline{x} \in X$ by forming the Lagrangian:

$$L_{\underline{x}_{k+1}} = \log p(\underline{x}, \hat{\underline{y}}_k) + \underline{\lambda}_x^T \left[G_x\underline{x} - \underline{\gamma}_x \right] \tag{5.9.2}$$

and then locating the stationary point of this function with respect to $\underline{\lambda}_x$ and $\underline{x}$. Similarly, given $\hat{\underline{x}}_{k+1}$, we can maximize $p(\hat{\underline{x}}_{k+1}, \underline{y})$ over $\underline{y} \in Y$ by finding the stationary point of the Lagrangian:

$$L_{\underline{y}_{k+1}} = \log p(\hat{\underline{x}}_{k+1}, \underline{y}) + \underline{\lambda}_y^T \left[G_y\underline{y} - \underline{\gamma}_y \right] \tag{5.9.3}$$

Solving these problems gives:

```
Primal Iterative Algorithm:
Guess $\hat{\underline{y}}_0 \in Y$
For $k = 0, 1, \cdots$
    $\hat{\underline{x}}_{k+1} = P_x H \hat{\underline{y}}_k + \bar{\underline{x}}$
    $\hat{\underline{y}}_{k+1} = P_y B \hat{\underline{x}}_{k+1} + \bar{\underline{y}}$
```

where:

$$\begin{aligned} P_x &= \left[I - VG_x^T(G_xVG_x^T)^{-1}G_x \right] \\ P_y &= \left[I - RG_y^T(G_yRG_y^T)^{-1}G_y \right] \\ \bar{\underline{x}} &= VG_x^T(G_xVG_x^T)^{-1}\underline{\gamma}_x \\ \bar{\underline{y}} &= RG_y^T(G_yRG_y^T)^{-1}\underline{\gamma}_y \end{aligned} \tag{5.9.4}$$

The matrix H, once again, is the filter which calculates the best unconstrained least squares estimate of $\underline{x}$ from $\underline{y}$. Matrices P_x and P_y are projection matrices (see Appendix G for a discussion of the properties of these matrices.) P_x effectively projects $H\hat{\underline{y}}_k$ onto the null space of the matrix G_x, thus removing the component which is orthogonal to the constraint set X with respect to the inner product $<\cdot,\cdot>_V$. To get a signal estimate which satisfies the constraints, we add back the vector, $\bar{\underline{x}}$, which satisfies $G_x \bar{\underline{x}} = \underline{y}_x$ and which is orthogonal to the null space of G_x. (The vector $\bar{\underline{x}}$ is thus the minimum norm $||\cdot||_V$ element in the constraint space X.) From figure 5.4 it is clear that the resulting signal estimate $\hat{\underline{x}}_{k+1}$ is the element of X which is closest to $H\hat{\underline{y}}_k$. To reestimate the output, pass this signal estimate through the filter B, then project the result onto the output constraint set Y by multiplying by P_y. The matrix P_y behaves in a similar manner as P_x, removing the component of $B\hat{\underline{x}}_{k+1}$ which is orthogonal to the constraint set Y with respect to the inner product $<\cdot,\cdot>_R$. This leaves only the component in the null space of the matrix G_y. The output estimate is then formed by adding back the vector, $\bar{\underline{y}}$, which satisfies $G_y \bar{\underline{y}} = \underline{y}_y$ and which is orthogonal to the null space of G_y.

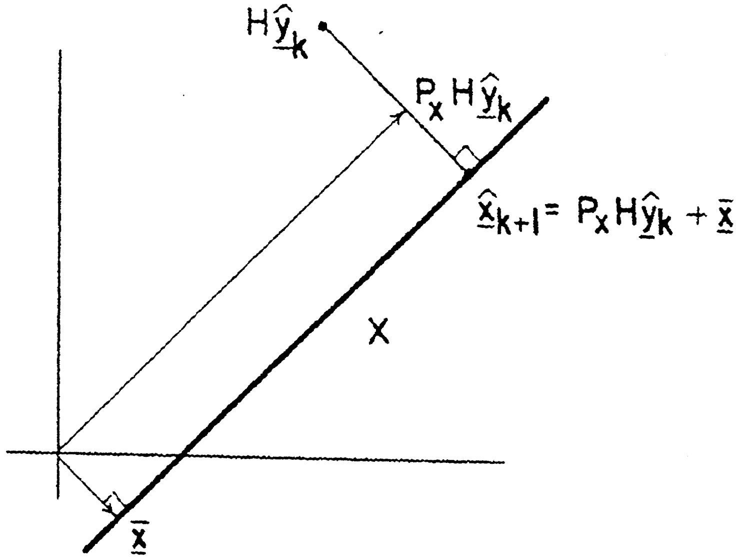

$H\hat{\underline{y}}_k$
$P_x H\hat{\underline{y}}_k$
$\hat{\underline{x}}_{k+1} = P_x H\hat{\underline{y}}_k + \bar{\underline{x}}$
X
$\bar{\underline{x}}$

Figure 5.4 - Behavior of Projection Operator P_x

Each iteration increases the likelihood $p(\hat{\underline{x}}_k, \hat{\underline{y}}_k)$ and thus improves the estimates. Since X and Y are convex, the algorithm is guaranteed to converge at a geometric rate to the unique global maximum solution $\underline{x}_*$, $\underline{y}_*$:

$$\|\hat{\underline{y}}_{k+1} - \underline{y}_* \|_R \le \nu_y \ \|\hat{\underline{x}}_{k+1} - \underline{x}_* \|_V \le \nu_x \nu_y \ \|\hat{\underline{y}}_k - \underline{y}_* \|_R \tag{5.9.5}$$

Appendix F gives slightly better upper bounds for the convergence rates ν_x, ν_y for the linear problem:

$$\begin{aligned} \nu_x &= \left\| R^{1/2} V^{-1/2} P_x H P_y \right\|_R \le \left(\frac{\mu_{max}}{\mu_{max}+1} \right)^{1/2} < 1 \\ \nu_y &= \left\| V^{1/2} R^{-1/2} P_y B P_x \right\|_V \le \left(\frac{\lambda_{max}}{\lambda_{max}+1} \right)^{1/2} < 1 \end{aligned} \tag{5.9.6}$$

9.3. Eigenvalues and Eigenvectors of the Algorithm

Each iteration of the algorithm defines a linear mapping from the signal space to the output space and back again. Some insight into the convergence behavior of our algorithm can be gained by examining the eigenstructure of this mapping. Recognizing that the solution $\underline{x}_*$, $\underline{y}_*$ is a stationary point of the algorithm, it is easy to show that:

$$\begin{aligned} \hat{\underline{x}}_{k+1} - \underline{x}_* &= P_x H P_y B (\hat{\underline{x}}_k - \underline{x}_*) \\ \hat{\underline{y}}_{k+1} - \underline{y}_* &= P_y B P_x H (\hat{\underline{y}}_k - \underline{y}_*) \end{aligned} \tag{5.9.7}$$

These equations can be put into a more symmetric form by recognizing that if $\hat{\underline{x}}_k, \underline{x}_* \epsilon X$ then $\hat{\underline{x}}_k - \underline{x}_*$ must be in the null space of G_x and must satisfy $P_x (\hat{\underline{x}}_k - \underline{x}_*) = \hat{\underline{x}}_k - \underline{x}_*$. Similarly, $\hat{\underline{y}}_k - \underline{y}_*$ must be in the null space of G_y, and so must satisfy $P_y (\hat{\underline{y}}_k - \underline{y}_*)$. This implies that:

$$\begin{aligned} \hat{\underline{x}}_{k+1} - \underline{x}_* &= P_x H P_y B P_x (\hat{\underline{x}}_k - \underline{x}_*) \\ \hat{\underline{y}}_{k+1} - \underline{y}_* &= P_y B P_x H P_y (\hat{\underline{y}}_k - \underline{y}_*) \end{aligned} \tag{5.9.8}$$

Appendix G analyzes the eigenvalues and eigenvectors of these two matrices $P_x H P_y B P_x$ and $P_y B P_x H P_x$ and proves the following results:

a) The eigenvalues of both matrices are all real, non-negative and less than $\nu_x \nu_y$.

b) The eigenvectors ϕ_i of $P_x H P_y B P_x$ form a complete orthonormal basis for $\mathbf{R}^N$ with respect to the inner product $<\cdot,\cdot>_V$. Similarly, the eigenvectors ψ_i of $P_y B P_x H P_y$ form a complete orthonormal basis for $\mathbf{R}^M$ with respect to the inner product $<\cdot,\cdot>_R$.

c) The non-zero eigenvalues λ_i of these two matrices are identical, and there are less than $\min(N-p, M-q)$ such eigenvalues. Moreover, there is a one-to-one correspondence between the eigenvectors of the two matrices corresponding to non-zero eigenvalues, defined by $\psi_i = \frac{1}{\sqrt{\lambda_i}} P_y B \phi_i$ and $\phi_i = \frac{1}{\sqrt{\lambda_i}} P_x H \psi_i$. All such eigenvectors ϕ_i are orthonormal elements of the null space of G_x, $P_x \phi_i = \phi_i$, and the eigenvectors ψ_i are orthonormal elements of the null space of G_y, $P_y \psi_i = \psi_i$.

The fact that all eigenvalues of $P_y B P_x H P_y$ and $P_x H P_y B P_x$ are real and are between 0 and $\nu_x \nu_y$ implies that the convergence rate will be underdamped, and suggests that acceleration methods ought to be very effective at improving the convergence rate. It also suggests that since $\nu_x \nu_y \rightarrow 1$ as the signal-to-noise ratio $\rightarrow \infty$, that the eigenvalues of $P_y B P_x H P_y$ and $P_x H P_y B P_x$ may be very close to 1 at high SNR, and thus the convergence rate can be very slow. Finally, note that the eigenvalues and eigenvectors of these two matrices have all the properties of the famed prolate spheroid functions; in fact, we will show that in the special case of bandlimited signal extrapolation, these eigenvectors *are* the discrete prolate spheroid functions.

9.4. Closed-Form Solution - Primal Problem

A direct closed-form solution for this problem can be easily derived. The globally optimum solution $\hat{x}$, $\hat{y}$ must be a stationary point of the algorithm, and thus must satisfy both the equations in (5.9.4). Thus:

$$\begin{pmatrix} I & -P_x H \\ -P_y B & I \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \qquad (5.9.9)$$

Adding $P_y B$ times the first row to the second gives an alternate expression:

$$\left(I - P_y B P_x H \right) \hat{y} = \left[\bar{y} + P_y B \bar{x} \right] \qquad (5.9.10)$$

This can be put into a somewhat more convenient form by the same trick used in the previous section. Because $\hat{x}$, $\hat{y}$ is a stationary point of the algorithm, it can be expressed in the form: $\hat{y} = P_y \hat{y} + \bar{y}$. Substituting this into (5.9.10) gives a more symmetric formula for $\hat{y}$:

$$\begin{aligned} \left(I - P_y B P_x H P_y \right) \hat{y} &= \left[\bar{y} + P_y B \bar{x} \right] \\ \hat{x} &= P_x H \hat{y} + \bar{x} \end{aligned} \qquad (5.9.11)$$

Similarly, a direct formula for $\hat{x}$ can be derived from (5.9.9) of the form:

$$\begin{aligned} \left(I - P_x H P_y B P_x \right) \hat{x} &= \left[\bar{x} + P_x H \bar{y} \right] \\ \hat{y} &= P_y B \hat{x} + \bar{y} \end{aligned} \qquad (5.9.12)$$

The chief problem with this closed-form solution is that in most applications the number of variables N and M is quite large, and storing and solving such a large set of simultaneous equations can be quite difficult. The iterative algorithm has the advantage that if $G_x V G_x^T$ and $G_y R G_y^T$ are diagonal or are easily diagonalizable, a situation that occurs in all the examples of chapter 7, then solving each step of the iteration is quite simple.

A much more robust approach to solving for $\hat{x}$ and $\hat{y}$ would be to return to the original constraint equation:

$$\begin{pmatrix} G_x & 0 \\ 0 & G_y \end{pmatrix} \begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix} = \begin{pmatrix} \gamma_x \\ \gamma_y \end{pmatrix} \qquad (5.9.13)$$

and to recognize that our algorithms are simply solving for a minimum norm solution to this underdetermined set of equations. Numerically robust algorithms such as Householder transforms could then be applied to solve this problem directly. This subject, however, is beyond the scope of this thesis (see, for example, Lawson and Hanson [2].)

9.5. Dual Problem - Iterative Algorithm

A rather different approach to solving this problem is to construct the dual optimization problem and solve that instead. Introduce Lagrange multipliers $\underline{\lambda}_x$ and $\underline{\lambda}_y$, and form the Lagrangian:

$$L_{xy} = \log p(\underline{x},\underline{y}) + \underline{\lambda}_x^T\left[G_x\underline{x} - \underline{\gamma}_x\right] + \underline{\lambda}_y^T\left[G_y\underline{y} - \underline{\gamma}_y\right] \tag{5.9.14}$$

The maximum of $p(\underline{x},\underline{y})$ over the domains $\underline{x} \in X$, $\underline{y} \in Y$ corresponds to the stationary point of L_{xy} with respect to $\underline{x}$, $\underline{y}$, $\underline{\lambda}_x$, and $\underline{\lambda}_y$. It is also well known (see, for example, Luenberger [1]) that this stationary point is a saddle point of L_{xy}, and that it satisfies a min-max law:

$$\min_{\underline{\lambda}_x,\underline{\lambda}_y}\left[\max_{\underline{x},\underline{y}} L_{xy}\right] \equiv \max_{\underline{x},\underline{y}}\left[\min_{\underline{\lambda}_x,\underline{\lambda}_y} L_{xy}\right] \tag{5.9.15}$$

The left hand side of this equation represents a dual algorithm for locating the solution to our problem. First maximize L_{xy} over all possible $\underline{x}$, $\underline{y}$; this gives the estimates:

$$\begin{pmatrix}\hat{\underline{x}}\\ \hat{\underline{y}}\end{pmatrix} = V_{xy}\begin{pmatrix}G_x^T & 0\\ 0 & G_y^T\end{pmatrix}\begin{pmatrix}\underline{\lambda}_x\\ \underline{\lambda}_y\end{pmatrix} \tag{5.9.16}$$

Substituting these values back into L_{xy} and simplifying leaves the dual problem:

$$\underline{\lambda}_x, \underline{\lambda}_y \leftarrow \min_{\underline{\lambda}_x,\underline{\lambda}_y} \frac{1}{2}\begin{pmatrix}\underline{\lambda}_x^T & \underline{\lambda}_y^T\end{pmatrix}\left[\begin{pmatrix}G_x & 0\\ 0 & G_y\end{pmatrix} V_{xy}\begin{pmatrix}G_x^T & 0\\ 0 & G_y^T\end{pmatrix}\right]\begin{pmatrix}\underline{\lambda}_x\\ \underline{\lambda}_y\end{pmatrix} - \begin{pmatrix}\underline{\gamma}_x^T & \underline{\gamma}_y^T\end{pmatrix}\begin{pmatrix}\underline{\lambda}_x\\ \underline{\lambda}_y\end{pmatrix}$$

$$\leftarrow \min_{\underline{\lambda}_x,\underline{\lambda}_y} \frac{1}{2}\left\|G_x^T\underline{\lambda}_x + G_y^T\underline{\lambda}_y\right\|_{V_{xy}^{-1}}^2 - \begin{pmatrix}\underline{\gamma}_x^T & \underline{\gamma}_y^T\end{pmatrix}\begin{pmatrix}\underline{\lambda}_x\\ \underline{\lambda}_y\end{pmatrix} \tag{5.9.17}$$

This is a positive definite, uniformly concave quadratic function of $\underline{\lambda}_x$, $\underline{\lambda}_y$. A formula for the solution is easily stated (see the next section), although the large number of equations, $p+q$, may make a direct solution prohibitively difficult. We therefore consider an iterative approach, minimizing first with respect to $\underline{\lambda}_x$, then with respect to $\underline{\lambda}_y$, iterating back and forth until the estimates converge. The resulting algorithm can be put into a very elegant framework if we first transform to new variables. Let V_x and V_y represent the *a priori* variances of $\underline{x}$ and $\underline{y}$ respectively:

$$\begin{aligned} V_x &= A^{-1}QA^{-T} \\ V_y &= BA^{-1}QA^{-T}B^T + R \end{aligned} \tag{5.9.18}$$

Then define:

$$\begin{aligned} \underline{\rho}_x &= A^{-1}QA^{-T}G_x^T\underline{\lambda}_x \\ \underline{\rho}_y &= \left[BA^{-1}QA^{-T}B^T + R\right]G_y\,\underline{\lambda}_y \end{aligned} \tag{5.9.19}$$

Beware that these variables $\underline{\rho}_x$ and $\underline{\rho}_y$ have dimensions N and M, and are much larger than the original p and q dimensional Lagrange multipliers $\underline{\lambda}_x$ and $\underline{\lambda}_y$. The iterative dual algorithm can now be put into the form:

```
Dual Iterative Algorithm:
Guess $\hat{\underline{\rho}}_{y_0}$
For $k=0,1,\cdots$
    $\hat{\underline{\rho}}_{x_{k+1}} = -Q_xH\hat{\underline{\rho}}_{y_k} + \bar{\underline{\rho}}_x$
    $\hat{\underline{\rho}}_{y_{k+1}} = -Q_yB\hat{\underline{\rho}}_{x_{k+1}} + \bar{\underline{\rho}}_y$
Iterate to convergence, then:
    $\hat{\underline{x}}_{k+1} = \hat{\underline{\rho}}_{x_{k+1}} + H\hat{\underline{\rho}}_{y_k}$
    $\hat{\underline{y}}_{k+1} = B\hat{\underline{\rho}}_{x_{k+1}} + \hat{\underline{\rho}}_{y_{k+1}}$
```

where:

$$Q_x = V_xG_x^T\left[G_xV_xG_x^T\right]^{-1}G_x$$

$$\begin{aligned} Q_y &= V_y G_y^T \left[G_y V_y G_y^T \right]^{-1} G_y \\ \bar{\underline{\rho}}_x &= V_x G_x^T \left[G_x V_x G_x^T \right]^{-1} \underline{\gamma}_x \\ \bar{\underline{\rho}}_y &= V_y G_y^T \left[G_y V_y G_y^T \right]^{-1} \underline{\gamma}_y \end{aligned} \tag{5.9.20}$$

and where H is the same filter matrix (5.4.3) used in our primal algorithm. Both Q_x and Q_y are projection matrices, playing a role similar to that of P_x and P_y in our primal algorithms, except that these matrices project vectors onto the *orthogonal complement* of the null spaces of G_x and G_y respectively. We start with an estimate of the "output multiplier" $\hat{\underline{\rho}}_{y_k}$. Filter this with the matrix H, then multiply by Q_x to project the result onto the orthogonal complement of the null space of G_x, with respect to an inner product $<\cdot,\cdot>_{V_x}$. Subtracting this from the minimum norm $\|\cdot\|_{V_x}$ element $\bar{\underline{\rho}}_x$ in X gives the "signal multiplier" estimate $\hat{\underline{\rho}}_{x_{k+1}}$ which points from $H\hat{\underline{\rho}}_{y_k}$ to the closest element in X (see figure 5.5).

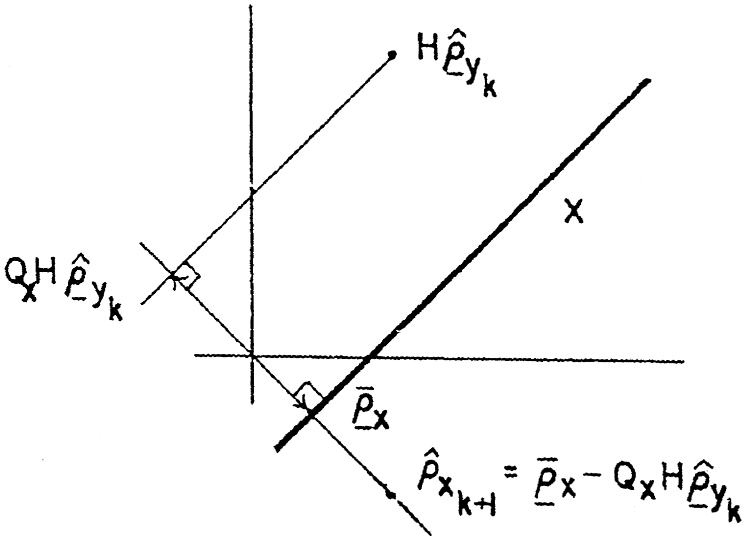

$H\hat{\underline{\rho}}_{y_k}$
X
$Q_x H \hat{\underline{\rho}}_{y_k}$
$\bar{\underline{\rho}}_x$
$\hat{\underline{\rho}}_{x_{k+1}} = \bar{\underline{\rho}}_x - Q_x H \hat{\underline{\rho}}_{y_k}$

Figure 5.5: $\hat{\underline{\rho}}_x = -Q_x H \hat{\underline{\rho}}_y + \bar{\underline{\rho}}_x$

Now to reestimate the "output multiplier", we pass the signal multiplier estimate $\hat{\underline{\rho}}_{x_{k+1}}$

through the filter B, then multiply by Q_y to project the result onto the orthogonal complement of the null space of G_y, with respect to an inner product $<\cdot,\cdot>_{V_y}$. Subtracting from the minimum norm $\|\cdot\|_{V_y}$ element $\bar{\underline{\rho}}_y$ in Y, thus estimates $\hat{\underline{\rho}}_{y_{k+1}}$ as the vector pointing from $B\hat{\underline{\rho}}_{x_{k+1}}$ to the nearest element in Y. On the next pass, this improved output multiplier estimate is used to improve the signal multiplier estimate. Each iteration decreases the quadratic objective function (5.9.17), and in fact all the convergence arguments presented for the primal algorithm apply to this dual algorithm as well. In particular, Appendix F proves that the estimates are guaranteed to converge to the unique global minimum solution $\underline{\rho}_{x*}$, $\underline{\rho}_{y*}$ at a geometric rate:

$$\| \hat{\underline{\rho}}_{y_{k+1}} - \underline{\rho}_{y*} \|_{V_y} \le \nu_y \ \| \hat{\underline{\rho}}_{x_{k+1}} - \underline{\rho}_{x*} \|_{V_x} \le \nu_x \nu_y \ \| \hat{\underline{\rho}}_{y_k} - \underline{\rho}_{y*} \|_{V_y} \tag{5.9.21}$$

where ν_x, ν_y are exactly the same convergence factors as in our primal algorithms. We can also show that:

$$\begin{aligned} \hat{\underline{\rho}}_{x_{k+1}} - \underline{\rho}_{x*} &= Q_x H Q_y B Q_x \left(\hat{\underline{\rho}}_{x_k} - \underline{\rho}_{x*} \right) \\ \hat{\underline{\rho}}_{y_{k+1}} - \underline{\rho}_{y*} &= Q_y B Q_x H Q_y \left(\hat{\underline{\rho}}_{y_k} - \underline{\rho}_{y*} \right) \end{aligned} \tag{5.9.22}$$

and the eigenvalues and eigenvectors of $Q_x H Q_y B Q_y$ and $Q_y B Q_x H Q_y$ have similar properties as listed in section 9.3. All eigenvalues are real, non-negative and less than $\nu_x \nu_y$, the eigenvectors form a complete orthonormal basis, and there are at most $\min(p, q)$ non-zero eigenvalues, and their corresponding eigenvectors are elements of the orthogonal complements of the null spaces of G_x and G_y respectively.

After the multipliers $\hat{\underline{\rho}}_{x_k}$, $\hat{\underline{\rho}}_{y_k}$ converge, the signal and output are estimated by adding appropriate multiples of the signal and output multipliers. Note that with the formulas given, $G_x \underline{x} = \underline{y}_x$ and $G_y \underline{y} = \underline{y}_y$, so that $\hat{\underline{x}}_{k+1} \in X$ and $\hat{\underline{y}}_{k+1} \in Y$.

9.6. Dual Problem - Closed-Form Solution

Several different closed-form solutions can be found for this dual problem. Since the solution must be a stationary point of the algorithm, one approach would be to simply combine equations (5.9.20):

$$\begin{pmatrix} I & Q_x H \\ Q_y B & I \end{pmatrix} \begin{pmatrix} \hat{\rho}_x \\ \hat{\rho}_y \end{pmatrix} = \begin{pmatrix} \bar{\rho}_x \\ \bar{\rho}_y \end{pmatrix} \tag{5.9.23}$$

Subtracting an appropriate multiple of the first row from the second or vice versa, and recognizing that $\hat{\rho}_x = Q_x \hat{\rho}_x + \bar{\rho}_x$ and $\hat{\rho}_y = Q_y \hat{\rho}_y + \bar{\rho}_y$, gives alternate versions:

$$\begin{aligned} \left(I - Q_x H Q_y B Q_x \right) \hat{\rho}_x &= \bar{\rho}_x - Q_x H \left[\bar{\rho}_y - Q_y B \bar{\rho}_x \right] \\ \hat{\rho}_y &= - Q_y B \hat{\rho}_x + \bar{\rho}_y \end{aligned} \tag{5.9.24}$$

or:

$$\begin{aligned} \left(I - Q_y B Q_x H Q_y \right) \hat{\rho}_y &= \bar{\rho}_y - Q_y B \left[\bar{\rho}_x - Q_x H \bar{\rho}_y \right] \\ \hat{\rho}_x &= - Q_x H \hat{\rho}_y + \bar{\rho}_x \end{aligned} \tag{5.9.25}$$

One potential difficulty with these equations is that in using the variable transformation (5.9.19), we have increased the number of variables from $p+q$ to $N+M$. A smaller number of equations would result, therefore, if we returned to the original dual problem (5.9.17) and solved it directly:

$$\left[\begin{pmatrix} G_x & 0 \\ 0 & G_y \end{pmatrix} V_{xy} \begin{pmatrix} G_x^T & 0 \\ 0 & G_x^T \end{pmatrix} \right] \begin{pmatrix} \lambda_x \\ \lambda_y \end{pmatrix} = \begin{pmatrix} \Upsilon_x \\ \Upsilon_y \end{pmatrix} \tag{5.9.26}$$

In some applications, this dual closed-form solution may be easier to solve than our primal closed-form solution.

9.7. Computational Noise Characteristics

Because all the operations in these algorithms are linear, it is straightforward to compute the effect of computational noise in our iterative algorithms. We will analyze the primal algorithm only; the dual algorithm can be treated in exactly the same way. We consider two sources of computational noise. Suppose that the correct values of $\bar{x}$ and $\bar{y}$ in (5.9.4) are given with errors $\underline{\delta}_x$ and $\underline{\delta}_y$, and suppose that on each pass $\hat{x}_k$ and $\hat{y}_k$ are calculated with errors $\underline{\epsilon}_{x_k}$ and $\underline{\epsilon}_{y_k}$. Thus each iteration calculates corrupted estimates $\tilde{x}_k$ and $\tilde{y}_k$ given by:

$$\begin{aligned} \tilde{x}_{k+1} &= P_x H \tilde{y}_k + (\bar{x}+\underline{\delta}_x) + \underline{\epsilon}_{x_{k+1}} \\ \tilde{y}_{k+1} &= P_y B \tilde{x}_k + (\bar{y}+\underline{\delta}_y) + \underline{\epsilon}_{y_{k+1}} \end{aligned} \tag{5.9.27}$$

Expanding both the noise-free iteration (5.9.4) and noisy iteration (5.9.27) recursively, using $P_y P_y = P_y$, and taking the difference:

$$(\tilde{y}_k - \hat{y}_k) = \sum_{m=0}^{k-1} (P_y B P_x H)^m \left[(\underline{\delta}_y + \underline{\epsilon}_{y_{k-m}}) + P_y B(\underline{\delta}_x + \underline{\epsilon}_{x_{k-m}}) \right] \tag{5.9.28}$$

Taking the norm of both sides:

$$\|\tilde{y}_k - \hat{y}_k\|_R \leq \sum_{m=0}^{k-1} \|P_y B P_x H\|_R^m \left\{ \|\underline{\delta}_y + \epsilon_{y_{k-m}}\|_R + \|P_y B(\underline{\delta}_x + \underline{\epsilon}_{x_{k-m}})\|_R \right\} \qquad (5.9.29)5n$$

The argument in Appeudix F proves that $\|P_y B P_x H\|_R \leq \nu_x \nu_y$, and is thus less than 1. If we assume, furthermore, that the computation noises $\underline{\delta}_x + \underline{\epsilon}_{x_k}$ and $\underline{\delta}_y + \underline{\epsilon}_{y_k}$ have about the same norm $\|\underline{\delta}_x + \underline{\epsilon}_{x_k}\|_V$ and $\|\underline{\delta}_y + \underline{\epsilon}_{y_k}\|_R$ on each pass, then as $k \to \infty$:

$$\|\tilde{y}_k - \hat{y}_k\|_R \leq \frac{1}{1-\nu_x \nu_y} \left\{ \|\underline{\delta}_y + \underline{\epsilon}_y\|_R + \|P_y B(\underline{\delta}_x + \underline{\epsilon}_x)\|_V \right\} \tag{5.9.30}$$

If the signal-to-noise level is very high, then $\nu_x \nu_y \approx 1$, and the computation noise sensitivity will be very high. Note that this is exactly the same situation in which we expect slow convergence and ill-behavior of the closed form solution. The difficulty is clearly intrinsic to the formulation of the problem.

9.8. Acceleration of the Convergence Rate

As discussed in chapter 3, section 11, these iterative algorithms tend to zig-zag along the ridge in the probability density $p(\underline{x},\underline{y})$ toward the global maximum. At high SNR, the contours of $p(\underline{x},\underline{y})$ will become very elliptical, each zig and zag will become very short, and the convergence rate will slow to a standstill. One solution, as suggested in chapter 3, is to recognize that a line connecting successive estimates will follow the ridge, and thus searching along such a line for a maximum should give an estimate that is much closer to the global maximum. Since $\log p(\underline{x},\underline{y})$ is quadratic, such a line search requires little additional computation. The resulting algorithm has the form:

```
Guess $\hat{\underline{x}}_0$, $\hat{\underline{y}}_0$

For $k=0,1,\cdots$

        $\hat{\underline{x}}_k' = P_x H \hat{\underline{y}}_k + \bar{\underline{x}}$
        $\hat{\underline{y}}_k' = P_y B \hat{\underline{x}}_k'$
        $(\hat{\underline{x}}_{k+1},\hat{\underline{y}}_{k+1}) = (1-\lambda)(\hat{\underline{x}}_k',\hat{\underline{y}}_k') + \lambda(\hat{\underline{x}}_{k-1}',\hat{\underline{y}}_{k-1}')$

            where: $\lambda = \dfrac{\hat{\underline{x}}_k'^T A^T Q^{-1} A(\hat{\underline{x}}_k' - \hat{\underline{x}}_{k-1}') + \Delta_k'^T R^{-1}(\Delta_k' - \Delta_{k-1}')}{\|A(\hat{\underline{x}}_k' - \hat{\underline{x}}_{k-1}')\|_Q^2 + \|\Delta_k' - \Delta_{k-1}'\|_R^2}$

                $\Delta_k' = \hat{\underline{y}}_k' - B\hat{\underline{x}}_k'$
```

In the above version, we use one iteration of our usual algorithm to estimate $\hat{\underline{x}}_k'$, $\hat{\underline{y}}_k'$, then search along the line connecting the last pre-extrapolation estimate $\hat{\underline{x}}_{k-1}'$, $\hat{\underline{y}}_{k-1}'$ and $\hat{\underline{x}}_k'$, $\hat{\underline{y}}_k'$ for a maximum of $\log p(\underline{x},\underline{y})$. An alternative approach, which doesn't seem to be quite as effective, would be to try searching along the line connecting $\hat{\underline{x}}_k'$, $\hat{\underline{y}}_k'$ and the last *post*-extrapolation estimate $\hat{\underline{x}}_k$, $\hat{\underline{y}}_k$ for a maximum. These methods were suggested, in a somewhat different form and with a rather different interpretation, by

Hayes and Tom [3]. Similar techniques could also be used to accelerate the dual algorithm.

9.9. PARTAN and Conjugate Gradient Methods

The interesting point about the above acceleration technique is that by using both types of acceleration mentioned above on every step, we actually arrive at a PARTAN algorithm which is guaranteed to converge in a finite number of steps! We discuss the primal problem first. Let us start by trying to solve the closed-form problem in (5.9.12). Multiplying on the left by V^{-1} gives:

$$\left[V^{-1} - V^{-1}P_x HP_y BP_x \right] \hat{\underline{x}} = V^{-1} \left[\left(I + P_x HP_y B \right) \bar{\underline{x}} + P_x H\bar{\underline{y}} \right] \qquad (5.9.31)$$

It is easy to verify that the matrix on the left equals $V^{-1} - P_x^T B^T P_y BP_x$, and is thus symmetric and positive definite. Call the matrix on the left T, and call the vector on the right $\underline{b}$. Solving $T\underline{x} = \underline{b}$ is equivalent to minimizing $\frac{1}{2}\underline{x}^T T\underline{x} - \underline{b}^T\underline{x}$, a problem than can be solved either with a PARTAN algorithm or a conjugate gradient algorithm in a finite number of steps. PARTAN is the easiest to state (see, for example, Luenberger. [1]) Start at an initial estimate $\hat{\underline{x}}_0$. Search in the direction of the gradient of $\frac{1}{2}\underline{x}^T T\underline{x} - \underline{b}^T\underline{x}$ for the minimum $\hat{\underline{x}}_1$. On every step following this, calculate the gradient $\hat{\underline{g}}_k = T\hat{\underline{x}}_k - \underline{b}$, and search along this gradient for a minimum $\hat{\underline{x}}_k''$. Next search along the line connecting $\hat{\underline{x}}_{k-1}$ and $\hat{\underline{x}}_k''$ for a minimum $\hat{\underline{x}}_{k+1}$. Repeat for $N-p$ steps (the dimension of X) and the final estimate of $\hat{\underline{x}}_{N-p}$ will be exactly the correct global minimum to the problem. This simple approach can be simplified even further by several tricks. The gradient of $\frac{1}{2}\underline{x}^T T\underline{x} - \underline{b}^T\underline{x}$ happens to be calculated by one iteration of our original algorithm; also doing a line search for the minimum of $\frac{1}{2}\underline{x}^T T\underline{x} - \underline{b}^T\underline{x}$ happens to give the same answer as a line search of the function $\log p(\underline{x},\underline{y})$. The final PARTAN algorithm takes the form:

```
Guess: $\hat{\underline{x}}_{-1} = \hat{\underline{x}}_0$
$\hat{\underline{y}}_{-1} = \hat{\underline{y}}_0 = \mathrm{P}_y \mathrm{B}\hat{\underline{x}}_0 + \bar{\underline{y}}$
For $k = 0, 1, \cdots, N-p$
    $\hat{\underline{x}}_k{}' = \mathrm{P}_x \mathrm{H}\hat{\underline{y}}_k + \bar{\underline{x}}$
    $\hat{\underline{y}}_k{}' = \mathrm{P}_y \mathrm{B}\hat{\underline{x}}_k{}' + \bar{\underline{y}}$
    $\hat{\alpha}_k = \dfrac{\hat{\underline{x}}_k{}'\mathrm{A}^{\mathrm{T}}\mathrm{Q}^{-1}\mathrm{A}(\hat{\underline{x}}_k{}' - \hat{\underline{x}}_k) + \Delta_k{}'\mathrm{R}^{-1}(\Delta_k{}' - \Delta_k)}{\|\mathrm{A}(\hat{\underline{x}}_k{}' - \hat{\underline{x}}_k)\|_{\mathrm{Q}}^2 + \|\Delta_k{}' - \Delta_k\|_{\mathrm{R}}^2}$
        where: $\Delta_k{}' = \hat{\underline{y}}_k{}' - \mathrm{B}\hat{\underline{x}}_k{}'$
               $\Delta_k = \hat{\underline{y}}_k - \mathrm{B}\hat{\underline{x}}_k$
    $(\hat{\underline{x}}_k{}'', \hat{\underline{y}}_k{}'') = (1-\hat{\alpha}_k)(\hat{\underline{x}}_k{}', \hat{\underline{y}}_k{}') + \hat{\alpha}_k(\hat{\underline{x}}_k, \hat{\underline{y}}_k)$
    $\hat{\beta}_k = \dfrac{\hat{\underline{x}}_k{}''\mathrm{A}^{\mathrm{T}}\mathrm{Q}^{-1}\mathrm{A}(\hat{\underline{x}}_k{}'' - \hat{\underline{x}}_{k-1}) + \Delta_k{}''\mathrm{R}^{-1}(\Delta_k{}'' - \Delta_{k-1})}{\|\mathrm{A}(\hat{\underline{x}}_k{}'' - \hat{\underline{x}}_{k-1})\|_{\mathrm{Q}}^2 + \|\Delta_k{}'' - \Delta_{k-1}\|_{\mathrm{R}}^2}$
    $(\hat{\underline{x}}_{k+1}, \hat{\underline{y}}_{k+1}) = (1-\hat{\beta}_k)(\hat{\underline{x}}_k{}'', \hat{\underline{y}}_k{}'') + \hat{\beta}_k(\hat{\underline{x}}_{k-1}, \hat{\underline{y}}_{k-1})$
```

This algorithm is very similar to the accelerated iterative algorithm given in the previous section; it starts with the usual filter-project-filter-project step, followed by two simple line search steps. The major (!!) difference is that this algorithm terminates in $N-p$ steps. (Actually, meticulous analysis would indicate that the only directions searched belong to the space spanned by the eigenvectors of $\mathrm{P}_x \mathrm{H} \mathrm{P}_y \mathrm{B} \mathrm{P}_x$ whose eigenvalues are non-zero. Thus the algorithm should theoretically converge in a number of steps equal to the number of non-zero eigenvectors, which is at most $\min(N-p, M-q)$.)

Another equivalent version of this algorithm is the conjugate gradient algorithm. This version also terminates in at most $N-p$ steps, but is more storage efficient. For details see Luenberger [1] ; the final computation takes the form:

```
Guess: $\hat{\underline{x}}_0$
$\hat{\underline{y}}_0 = \mathrm{P}_y \mathrm{B}\hat{\underline{x}}_0 + \bar{\underline{y}}$
```

```
$\underline{\hat{x}}_1 = \mathrm{P}_x \mathrm{H} \underline{\hat{y}}_0 + \underline{\bar{x}}$
$\underline{g}_0 = -\underline{d}_0 = \underline{\hat{x}}_0 - \underline{\hat{x}}_1$
For $k = 0, 1, \cdots, N-p$
    $\underline{e}_k = \underline{d}_k - \mathrm{P}_x \mathrm{H} \mathrm{P}_y \mathrm{B} \underline{d}_k$
    $\tau_k = \underline{d}_k^{\mathrm{T}} \mathrm{V}^{-1} \underline{e}_k$
    $\hat{\alpha}_k = \frac{\underline{g}_k^{\mathrm{T}} \mathrm{V}^{-1} \underline{g}_k}{\tau_k}$      $\left( \text{or: } \hat{\alpha}_k = -\frac{\underline{g}_k^{\mathrm{T}} \mathrm{V}^{-1} \underline{d}_k}{\tau_k} \right)$
    $\underline{\hat{x}}_{k+1} = \underline{\hat{x}}_k + \hat{\alpha}_k \underline{d}_k$
    $\underline{g}_{k+1} = \underline{g}_k + \hat{\alpha}_k \underline{e}_k$      $\left( \text{or: } \underline{g}_{k+1} = \underline{x}_{k+1} - \mathrm{P}_x \mathrm{H} \mathrm{P}_y \mathrm{B} \underline{\hat{x}}_{k+1} \right)$
    $\hat{\beta}_k = \frac{\underline{g}_{k+1}^{\mathrm{T}} \mathrm{V}^{-1} \underline{g}_{k+1}}{\underline{g}_k^{\mathrm{T}} \mathrm{V}^{-1} \underline{g}_k}$      $\left( \text{or: } \hat{\beta}_k = \frac{\underline{g}_{k+1}^{\mathrm{T}} \mathrm{V}^{-1} \underline{e}_k}{\tau_k} \right)$
    $\underline{d}_{k+1} = -\underline{g}_{k+1} + \hat{\beta}_k \underline{d}_k$
```

The alternative formulas given above in parentheses are less convenient, though theoretically equivalent methods for calculating the required values. Note that, once again, each iteration requires one filter-project-filter-project step, plus two extrapolation-like steps with factors $\hat{\alpha}_k$ and $\hat{\beta}_k$.

Although these two algorithms theoretically converge exactly in at most $\min(N-p, M-q)$ iterations, in practice more iterations are often needed to solve extremely ill-conditioned problems. If many more iterations will be needed, Luenberger suggests restarting the algorithm at intervals greater than or equal to $\min(N-p, M-q)$ in order to avoid convergence problems caused by build-up of errors. Finally, Luenberger also suggests various ways to modify these algorithms to solve problems with non-linear objective functions or non-linear constraint sets.

Another set of algorithms result if we start by trying to solve for $\underline{y}$ rather than $\underline{x}$. In this case we would have used equation (5.9.11) instead of (5.9.12) in beginning our analysis, and the algorithm would again terminate in at most $\min(N-p, M-q)$ steps. The dual problem could also be solved by a similar approach, starting either with the equations (5.9.24) or (5.9.25) using the transformed variables $\underline{\rho}_x$ and $\underline{\rho}_y$, or starting with the equations (5.9.26) using the multipliers $\underline{\lambda}_x$ and $\underline{\lambda}_y$.

9.10. Infinite Dimensional Spaces

Although we will not prove this, nearly all the results above also apply to the case when $\underline{x}$ and $\underline{y}$ are infinite dimensional vectors. Assume that the infinite dimensional linear operators Q and R satisfy $Q \geq \epsilon I$ and $R \geq \epsilon I$, for some $\epsilon > 0$, that A is a bounded and invertible linear operator, and that B is a bounded linear operator. Then each iteration of our algorithm still defines a contraction mapping on the linear varieties X, Y. The difference between successive estimates must therefore decrease at the rate $\nu_x \nu_y$, which implies that the estimates must converge at a geometric rate to the unique global optimizing solution. For a more complete discussion of infinite dimensional signal reconstruction problems, see Youla [4] or Mosca [5].

10. Linear Ineqyality Constraints

Another important special case is when the constraint sets X and Y are cones defined by sets of linear inequalities:

$$X = \left\{ \underline{x} \;\middle|\; G_x \underline{x} \leq \underline{\gamma}_x \right\} \qquad (5.10.1)$$

$$Y = \left\{ \underline{y} \;\middle|\; G_y \underline{y} \leq \underline{\gamma}_y \right\}$$

A set of linear inequality constraints, or a mixture of equality and inequality constraints, defines a "simplex" constraint set, which is a convex, closed (though possibly infinite) polytope. Since X and Y are convex, all of our iterative algorithms are guaranteed to converge linearly to the unique global solution of the MCEM, XMAP, YMAP or XYMAP problems. In general, our different estimation criteria will give different estimates of $\underline{x}$ and $\underline{y}$. Closed form solutions are difficult to derive except for the XYMAP problem, which only requires maximizing the quadratic function $\log p(\underline{x},\underline{y})$ over a simplex constraint set X, Y. The Kuhn-Tucker optimality conditions can be used to state necessary conditions for the point $(\hat{\underline{x}},\hat{\underline{y}})$ to be the global maximum of $\log p(\underline{x},\underline{y})$ (see, for example, Luenberger. [1]) A wide variety of quadratic programming algorithms have been developed for solving problems like this. Most of these are modified forms of linear programming which are able to calculate the exact solution with a finite amount of computation. The disadvantage of these algorithms is that most of them adjust only one variable at a time, and thus they can be very slow at solving large XYMAP problems. Further discussion of these techniques is beyond the scope of this thesis; for details see Boot[6] or Künzi and Krelle.[7] A dual XYMAP algorithm can also be developed; it will involve minimizing the same quadratic function of the Lagrange multipliers as in (5.9.17) but with the additional constraint that the multipliers be non-negative.

11. Summary

In this chapter we have applied our MMSE, MCEM and MAP approaches to a simple linear Gaussian system model of a stochastic signal corrupted by a linear filter and additive noise. When the parameters of this system are known, four different iterative algorithms were proposed for estimating the unknown signal and output given only incomplete knowledge of the values of these unknowns. These algorithms all iterate between filtering steps and a pair of projection and/or conditional expectation calculations. Each iteration of the algorithms improves the estimates, and if the constraint sets are convex, geometric convergence can be guaranteed to the unique global optimum solution. The MCEM algorithm appears to come closest to the MMSE estimates, but it involves more computation than the other iterative routines. XMAP and YMAP have intermediate performance at intermediate computational cost, and XYMAP is the simplest, though its estimates can be quite poor in some cases. If the constraint sets are defined by sets of linear equations, then all our methods give identical estimates. Both primal and dual approaches to the problem can be formulated; each leads to a different iterative algorithm whose convergence rate, computational noise sensitivity, and eigenvalues and eigenvectors have been analyzed in detail. Closed-form solutions for both primal and dual problems were derived. PARTAN or conjugate gradient algorithms were also presented, which solve these primal and dual problems in a finite number of steps; each step uses one pass of our usual iteration followed by two line searches. Another case that can be treated straightforwardly is when the constraint sets are defined by linear inequalities. Here the Kuhn-Tucker optimality conditions can be invoked to help find the solution, and quadratic programming algorithms are available for solving the primal or dual XYMAP problems in a finite number of steps.

References

1. David G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, Mass. (1973).
2. Charles L. Lawson and Richard J. Hanson, *Solving Least Squares Problems*, Prentice Hall, Inc., Englewood Cliffs, N.J. (1974).
3. Monson H. Hayes and Victor T. Tom, *Adaptive Acceleration of Iterative Signal Recontruction Algorithms*, Technical Note 1980-28, Lincoln Laboratory M.I.T. (to be published).
4. Dante Youla, "Generalized Image Restoration by the Method of Alternating Orthogonal Projections," *IEEE Trans. Circuits. Syst.* CAS-25(9), pp.694-702 (Sept 1978).
5. Edoardo Mosca, "On a Class of Ill-Posed Estimation Problems and a Related Gradient Iteration," *IEEE Trans. Auto. Control* AC-17(4), pp.459-465 (Aug 1972).
6. John C. G. Boot, *Quadratic Programming - Algorithms, Anomalies, Applications*, North-Holland Publishing Company - Rand McNally & Company, Chicago (1964).
7. Hans Künzi and Wilhelm Krelle, *Nonlinear Programming*, Blaisdell Publishing, Waltham, Mass. (1966).

Chapter 6

Applications in Optimal Signal Reconstruction

Part II - Fisher Theory

1. Introduction

The signal reconstruction algorithms presented in chapter 5 take a very interesting form if we let the *a priori* signal covariance Q become uniformly infinitely large, thus making the *a priori* signal density $p(\underline{x})$ asymptotically "flat". In the limit $Q = \infty$, our four Bayesian MCEM and MAP algorithms will be converted into Fisher MCEM and MAP algorithms, in which the density $p(\underline{x},\underline{y})$ in the cross-entropy expression has been replaced by $p(\underline{y}\,|\underline{x})$. In the case of XMAP and XYMAP these Fisher algorithms are equivalent to Maximum Likelihood estimation problems; the philosophical interpretation of the Fisher YMAP and MCEM algorithms is less clear. All four of these Fisher algorithms can be viewed as searching for a pair of signal and output estimates which come as close to each other as possible and yet still obey the known constraints on their values. Iterative algorithms for solving these Fisher algorithms can be derived which are similar to those of chapter 5, alternating between projections or conditional expectations on the signal and on the output constraint spaces, except that no filtering steps are used. Unfortunately, unlike the Bayesian algorithms, the Fisher estimation problems are not guaranteed to have a solution, and even if a solution exists it is not guaranteed to be unique. Our convergence rate factor $\nu_x \nu_y$ also approaches 1 in the limit as $Q \to \infty$, and thus proving convergence of the algorithms is considerably more difficult; furthermore, the convergence rate can be sublinear. Once again, linear variety constraint sets will be analyzed in great detail. We will show in this case that the convergence rate and computational noise sensitivity are a function of the "angle" between the constraint sets X and Y. Closed form solutions, iterative projection algorithms and conjugate gradient algorithms will be developed for both the primal and dual optimization approaches.

2. Limiting Behavior of the Bayesian Algorithms as $Q \rightarrow \infty$

There are many practical signal reconstruction problems in which no real *a priori* information is available about the energy in the unknown signal. One approach to this problem would be to treat the signal $\underline{x}$ as a Fisher non-random constant, and then apply Maximum Likelihood to estimate its value. A more interesting and illuminating approach, however, is to consider the case of no *a priori* information as a limiting form of the Bayesian problem, in which the *a priori* signal covariance Q becomes infinitely large, thus making the *a priori* density $p(\underline{x})$ asymptotically flat. To avoid technical problems, let us assume that the dimensions of the signal and output spaces are equal, $N=M$, and that B=I. Let the *a priori* covariance have the form $Q=\frac{1}{\alpha}Q_0$, where Q_0 is a positive definite covariance matrix, $Q_0>0$. Let $p_\alpha(\underline{x},\underline{y})$ and $p_\alpha(\underline{x})$ be the model densities corresponding to the *a priori* covariance $Q=\frac{1}{\alpha}Q_0$, and let $H_\alpha(q_X,q_\Phi)$ be the corresponding cross-entropy expression for any of our algorithms. Given any $\alpha>0$, let $\hat{q}_{X,\alpha}(\underline{x})$ and $\hat{q}_{\Phi,\alpha}(\underline{\phi})$ be the density estimates corresponding to the global minimum of the cross-entropy expression for any of our algorithms. If we let $N_\Lambda(\underline{m},V)$ represent a Ga:ussian density with mean $\underline{m}$ and variance V which has been truncated to the set Λ, then tre four estimation approaches discussed in chapter 5 yield density estimates of the form:

Bayesian	$\hat{q}_{X,\alpha}$	$\hat{q}_{\Phi,\alpha}$
MCEM:	$N_X(H_\alpha\hat{\underline{y}}_\alpha, V_\alpha)$	$N_Y(\hat{\underline{x}}_\alpha, R)$
XMAP:	$\delta(\underline{x} - \hat{\underline{x}}_\alpha)$	$N_Y(\hat{\underline{x}}_\alpha, R)$
YMAP:	$N_X(H_\alpha\hat{\underline{y}}_\alpha, V_\alpha)$	$\delta(\underline{y} - \hat{\underline{y}}_\alpha)$
XYMAP:	$\delta(\underline{x} - \hat{\underline{x}}_\alpha)$	$\delta(\underline{y} - \hat{\underline{y}}_\alpha)$

where H_α, V_α are the matrices corresponding to $Q = \frac{1}{\alpha} Q_0$. If X and Y are convex, then these global minimizing solutions are guaranteed to exist and to be unique for all four algorithms. Even if X, Y are not convex, the MAP algorithms are still guaranteed to have at least one global minimizing solution (we conjecture that MCEM will also always have at least one global minimizing solution.)

As discussed in section 8 of chapter 2, as $\alpha \to 0$ and the *a priori* density $p_\alpha(\underline{x})$ becomes asymptotically flat, we would expect our Bayesian density estimates $\hat{q}_{X,\alpha}(\underline{x})$, $\hat{q}_{Y,\alpha}(\underline{y})$ to asymptotically minimize the "Fisher" cross-entropy expression $H_{ML}(q_X, q_Y)$. Remember that the Fisher cross-entropy is formed by replacing $p_\alpha(\underline{x}, \underline{y})$ in the Bayesian cross-entropy expression $H_\alpha(q_X, q_Y)$ by $p(\underline{y} | \underline{x})$, and that it satisfies:

$$H_\alpha(q_X, q_Y) = H_{ML}(q_X, q_Y) - \int_X q_X(\underline{x}) \log p_\alpha(\underline{x})\, d\underline{x} \tag{6.2.1}$$

Note that $p(\underline{y} | \underline{x})$ and thus $H_{ML}(q_X, q_Y)$ are independent of α. Since the integral of $p(\underline{y} | \underline{x})$ over $X \times Y$ is not necessarily finite, H_{ML} is not necessarily bounded below, and may not have a global minimizer. If, however, the Fisher cross-entropy does achieve its global minimum at a pair of densities $\hat{q}_X$, $\hat{q}_Y$, then this pair must satisfy:

$$\hat{q}_X \leftarrow \min_{q_X} H_{ML}(q_X, \hat{q}_Y) \tag{6.2.2}$$

$$\hat{q}_Y = \min_{q_Y} H_{ML}(\hat{q}_X, q_Y)$$

With some algebra, it is easy to show that these global minimizing Fisher estimates must have the form:

Fisher	$\hat{q}_X(\underline{x})$	$\hat{q}_Y(\underline{y})$
MCEM:	$N_X(\hat{\underline{y}}, \mathrm{R})$	$N_Y(\hat{\underline{x}}, \mathrm{R})$
XMAP:	$\delta(\underline{x} - \hat{\underline{x}})$	$N_Y(\hat{\underline{x}}, \mathrm{R})$
YMAP:	$N_X(\hat{\underline{y}}, \mathrm{R})$	$\delta(\underline{y} - \hat{\underline{y}})$
XYMAP:	$\delta(\underline{x} - \hat{\underline{x}})$	$\delta(\underline{y} - \hat{\underline{y}})$

Two questions remain: does the Fisher problem indeed have a solution, and how does the Bayesian solution $\hat{q}_{X,\alpha}$, $\hat{q}_{Y,\alpha}$ behave as $\alpha \to 0$? To answer these, let us return to equation (6.2.1) linking the Bayesian and Fisher cross-entropies. Substituting our model density into this expression yields:

$$H_\alpha(q_X, q_Y) = H_{ML}(q_X, q_Y) + \frac{\alpha}{2} \int_X \|\mathrm{A}\underline{x}\|^2_{\mathrm{Q}_0}\, q_X(\underline{x})\, d\underline{x} + \frac{N}{2} \log \left| \frac{1}{\alpha} \mathrm{Q}_0 \right| \tag{6.2.3}$$

The "Bayesian" cross-entropy is thus equal to the Fisher cross-entropy, plus a term proportional to α measuring the average signal energy for the density q_X, plus another term which is independent of the densities q_X, q_Y. From this relationship we would expect that if α is very small, then the estimates which minimize the Bayesian cross-entropy H_α ought to be very similar to the estimates which minimize the Fisher cross-entropy H_{ML}. This intuition can be proven formally. Because the densities $\hat{q}_{X,\alpha}$, $\hat{q}_{Y,\alpha}$ and $\hat{q}_X$, $\hat{q}_Y$ always have the form given in the preceding tables, to prove that the Bayesian estimates converge to the Fisher estimates, we need only prove that the centers of the densities converge. Appendix H proves:

Theorem 6.1 Let X, Y be closed and measurable sets. Assume that for any of our four algorithms that for all $\alpha>0$, $H_\alpha(q_X, q_Y)$ achieves its global minimum at some value $\hat{q}_{X,\alpha}$, $\hat{q}_{Y,\alpha}$ with means $\hat{x}_\alpha$, $\hat{y}_\alpha$. Also assume that $H_{ML}(q_X, q_Y)$ achieves its global minimum at some value $\hat{q}_X$, $\hat{q}_Y$ with means $\hat{x}$, $\hat{y}$. Let $\Psi = \{(\hat{q}_X, \hat{q}_Y)\}$ be the set of all density pairs which minimize H_{ML}. Then:

a) For $\alpha>0$, the average signal energy of any of our Bayesian estimates is less than the average signal energy of any Fisher estimate:

$$||A\hat{x}_\alpha||^2_{Q_0} \le \int_X ||Ax||^2_{Q_0}\, \hat{q}_{X,\alpha}(x)\, dx \le \inf_{\hat{q}_X \in \Psi} \int_X ||Ax||^2_{Q_0}\, \hat{q}_X(x)\, dx$$

b) Let $\alpha_0 > \alpha_1 > \cdots$ be any monotonically decreasing sequence such that $\lim_{i\to\infty} \alpha_i = 0$. Then the means $\hat{x}_{\alpha_i}$, $\hat{y}_{\alpha_i}$ of the Bayesian density estimates remain bounded as $i\to\infty$, and every limit point $\hat{x}$, $\hat{y}$ of the sequence corresponds to the means of a pair of densities $\hat{q}_X$, $\hat{q}_Y$ which minimize the Fisher cross-entropy and have minimal signal energy:

$$\hat{q}_X, \hat{q}_Y \leftarrow \min_{q_X, q_Y} H_{ML}(q_X, q_Y)$$

and:

$$\int_X ||Ax||^2_{Q_0}\, \hat{q}_X(x)\, dx = \inf_{\hat{q}_X' \in \Psi} \int_X ||Ax||^2_{Q_0}\, \hat{q}_X'(x)\, dx$$

c) If there is no finite global minimizing solution to the Fisher problem, then the Bayesian estimates diverge:

$$\int_X ||Ax||^2_{Q_0}\, \hat{q}_{X,\alpha_i}(x)\, dx \to \infty \quad \text{as } \alpha_i \to 0$$

The Bayesian algorithm therefore always generates estimates with less average signal energy than that of the Fisher algorithm's estimates. As $\alpha\to 0$ and our *a priori* estimate of the signal covariance Q goes to infinity, the signal energy of our Bayesian MCEM and MAP algorithms will gradually increase, and the estimates will converge to a minimal signal energy solution to the corresponding Fisher problem. If the Fisher problem has no finite minimizing solution, then the Bayesian algorithm will diverge as $\alpha\to 0$, with $\|\hat{\underline{x}}_\alpha\|, \|\hat{\underline{y}}_\alpha\| \to \infty$.

3. The Fisher Algorithms

Let us examine the Fisher versions of our estimation problems. XYMAP is the easiest to analyze. The XYMAP cross-entropy expression satisfies:

$$H_{ML}(q_X, q_Y) = -\log p(\hat{\underline{y}}\,|\hat{\underline{x}}) \tag{6.3.1}$$

Thus minimizing this Fisher cross-entropy function is equivalent to solving:

$$\text{Fisher XYMAP:} \quad \hat{\underline{x}}, \hat{\underline{y}} \leftarrow \max_{\underline{x}\in X, \underline{y}\in Y} p(\underline{y}\,|\underline{x})$$
$$\leftarrow \min_{\underline{x}\in X, \underline{y}\in Y} \|\underline{y} - \underline{x}\|_R^2 \tag{6.3.2}$$

The Fisher XYMAP problem therefore tries to find the pair of signal and output values which not only meet the known constraints $\underline{x}\in X$ and $\underline{y}\in Y$, but which also come as close to each other as possible. The signal estimate $\hat{\underline{x}}$ will meet all the signal constraints and come "close" to meeting the output constraints, while the output estimate $\hat{\underline{y}}$ will meet all the output constraints, while coming "close" to meeting the signal constraints. In particular, if there exists some vector $\hat{\underline{x}}$ which meets both signal and output constraints, $\underline{x}\in X$ and $\underline{y}\in Y$, then it will be the solution to the XYMAP Fisher algorithm.

Problems which are appropriately treated by an algorithm like this arise in a wide variety of applications in which a signal must be constructed which meets two different sets of constraints stated in different domains. For example, we might want to design a

Finite Impulse Response (FIR) filter which meets certain impulse response constraints and also certain frequency response constraints. Direct calculation of the filter coefficients is often very difficult [1], so we might try to put this problem into the form of a Fisher XYMAP problem. Let $\underline{x}$ and $\underline{y}$ be filter coefficient sets, let X be the set of time constraints, and let Y be the set of frequency constraints. If there exists a unique filter $\underline{x}_f$ which meets all these constraints, then the solution to (6.3.2) will exactly satisfy $\hat{\underline{x}} = \hat{\underline{y}} = \underline{x}_f$. If no such solution exists, then the filter $\hat{\underline{x}}$ will meet all the time constraints and come as close as possible to meeting all the frequency constraints, while the filter $\hat{\underline{y}}$ will meet all the frequency constraints and come as close as possible to meeting all the time constraints.

The Fisher version of XMAP is also relatively easy to characterize. We can easily show that for this algorithm:

$$\min_{q_X} H_{ML}(q_X, q_Y) = -\log p(Y | \underline{x}) \tag{6.3.3}$$

and thus minimizing this Fisher cross-entropy is equivalent to solving the Maximum Likelihood problem:

$$\textbf{Fisher XMAP:} \qquad \hat{\underline{x}} - \max_{\underline{x} \in X} p(Y | \underline{x}) \tag{6.3.4}$$

This algorithm effectively searches for the signal value $\hat{\underline{x}}$ which comes as close as possible to the set Y.

The Fisher versions of YMAP and MCEM can not be stated in terms of traditional Maximum Likelihood estimation problems. We will therefore treat these two cases simply as limiting forms of the Bayesian problem when all *a priori* knowledge is absent.

4. Iterative Fisher Algorithms

Starting with the Fisher form of the cross-entropy expression for our four algorithms, we can iteratively solve for the global minimizer of H_{ML} by minimizing first with respect to the signal density q_X, then with respect to the output density q_Y, iterating back and forth until the estimates converge. Using precisely the same derivation used in chapter 5, the resulting signal and output density estimates can be shown to have the form:

$$\hat{q}_{X_k}(\underline{x}) = \begin{cases} N_X(\hat{\underline{y}}_{k-1}, \mathrm{R}) & \text{for MCEM, YMAP} \\ \delta(\underline{x} - \hat{\underline{x}}_k) & \text{for XMAP, XYMAP} \end{cases} \tag{6.4.1}$$

$$\hat{q}_{Y_k}(\underline{y}) = \begin{cases} N_Y(\hat{\underline{x}}_k, \mathrm{R}) & \text{for MCEM, XMAP} \\ \delta(\underline{y} - \hat{\underline{y}}_k) & \text{for YMAP, XYMAP} \end{cases}$$

where the centers of these densities $\hat{\underline{x}}_k$, $\hat{\underline{y}}_k$ are iteratively calculated as follows:

	Signal Estimates	Output Estimates
MCEM:	$\hat{\underline{x}}_{k+1} = \mathrm{E}_X[\underline{x} \mid \hat{\underline{y}}_k]$	$\hat{\underline{y}}_{k+1} = \mathrm{E}_Y[\underline{y} \mid \hat{\underline{x}}_{k+1}]$
XMAP:	$\hat{\underline{x}}_{k+1} - \min_{\underline{x} \in X} \lVert \underline{x} - \hat{\underline{y}}_k \rVert_{\mathrm{R}}^2$	$\hat{\underline{y}}_{k+1} = \mathrm{E}_Y[\underline{y} \mid \hat{\underline{x}}_{k+1}]$
YMAP:	$\hat{\underline{x}}_{k+1} = \mathrm{E}_X\{\underline{x} \mid \hat{\underline{y}}_k]$	$\hat{\underline{y}}_{k+1} = \min_{\underline{y} \in Y} \lVert \underline{y} - \hat{\underline{x}}_{k+1} \rVert_{\mathrm{R}}^2$
XYMAP:	$\hat{\underline{x}}_{k+1} - \min_{\underline{x} \in X} \lVert \underline{x} - \hat{\underline{y}}_k \rVert_{\mathrm{R}}^2$	$\hat{\underline{y}}_{k+1} = \min_{\underline{y} \in Y} \lVert \underline{y} - \hat{\underline{x}}_{k+1} \rVert_{\mathrm{R}}^2$

The XYMAP procedure alternates between projecting the estimate $\hat{\underline{y}}_k$ onto the constraint set X to estimate $\hat{\underline{x}}_{k+1}$, then projecting this signal estimate onto the constraint set Y to estimate $\hat{\underline{y}}_{k+1}$. Each pass therefore simply alternates between moving $\underline{x}$ closer to

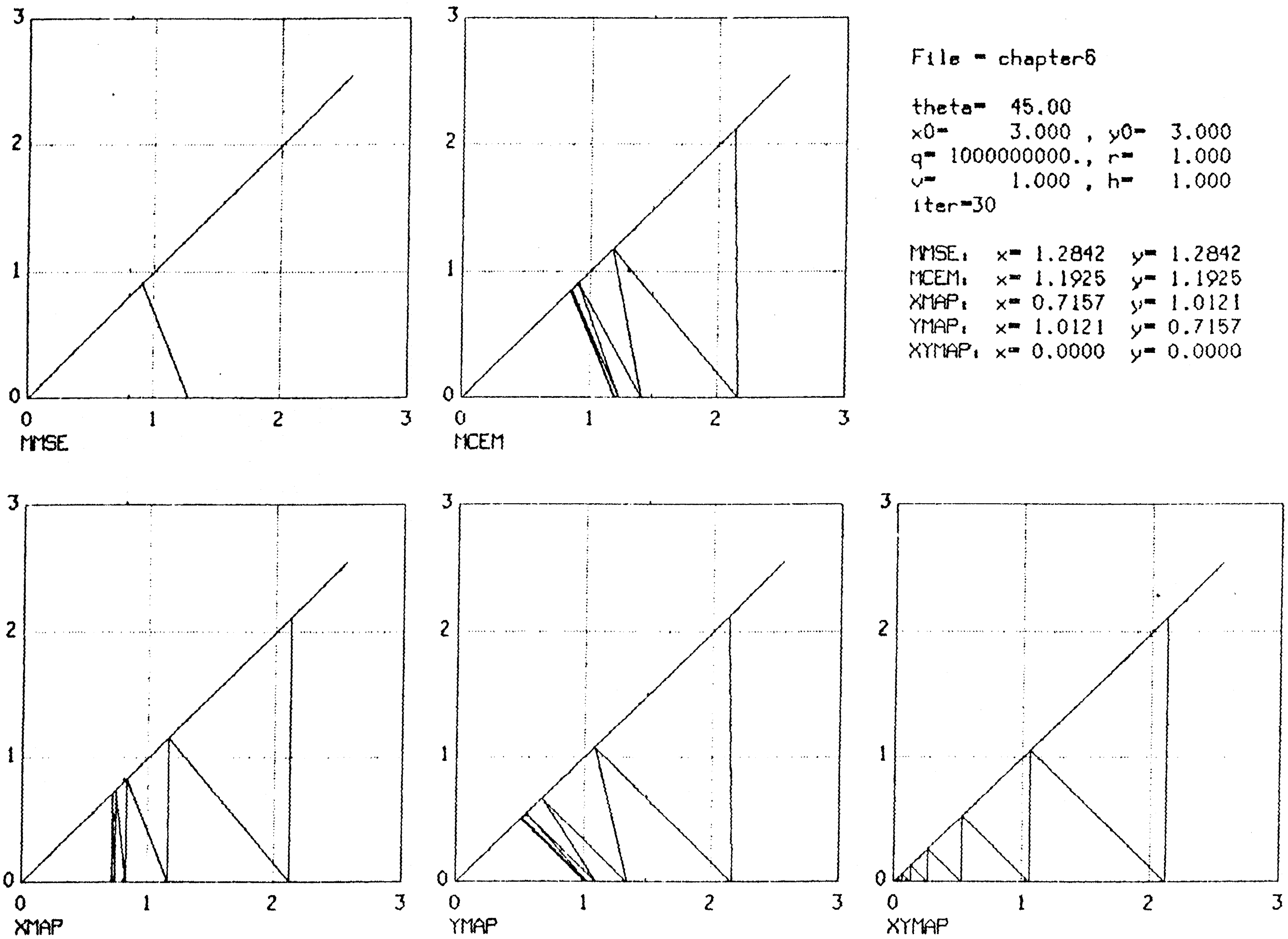
$\underline{y}$ and moving $\underline{y}$ closer to $\underline{x}$. The MCEM, XMAP and YMAP algorithms are similar, except that one or both projection operations are replaced by a conditional expectation operation, using the truncated Gaussians $N_X(\hat{y}_k, R)$ or $N_Y(\hat{x}_k, R)$. Using a conditional expectation operator instead of a projection operator is usually more complicated, but we would expect the resulting estimate to come closer on average to the actual value of the unknown. Each pass of any of our algorithms strictly decreases the appropriate cross-entropy expression unless a stationary point has already been reached. The XMAP and XYMAP methods also strictly increase the likelihood functions $p(Y | \hat{x}_k)$ and $p(\hat{y}_k | \hat{x}_k)$ respectively on each pass. In the case of XYMAP, this implies that:

$$\| \hat{y}_{k+1} - \hat{x}_{k+1} \|_R^2 < \| \hat{y}_k - \hat{x}_{k+1} \|_R^2 < \| \hat{y}_k - \hat{x}_k \|_R^2 \qquad (6.4.2)$$

so that the distance between the estimates strictly decreases on each pass.

5. An Example

Figure 6.1 shows the behavior of our four iterative algorithms for the same example used in chapter 5, section 8. Here we have set the *a priori* signal covariance to $Q=10^9$; all the other parameters have the same values as in chapter 5. For comparison, the figure also shows the limiting estimates of MMSE as $\alpha \to 0$. MCEM alternates between a pair of conditional expectations on X and Y to calculate $\hat{x}_k, \hat{y}_k$. Note that of all the algorithms, MCEM again comes closest to the MMSE estimates, and also shows the same symmetry as MMSE, setting $\hat{x} = \hat{y}$. The convergence rate is roughly linear. The XYMAP algorithm uses projections instead of conditional expectations to calculate its estimates. As a result, the estimates converge at a geometric rate to the pair of signal and output values which are as close to each other as possible, namely $\hat{x} = \hat{y} = \underline{0}$. Unfortunately, these XYMAP estimates are the worst of all our algorithms. XMAP and YMAP alternate between a conditional expectation and a projection



```
File - chapter6

theta=  45.00
x0=      3.000 , y0=  3.000
q= 1000000000., r=   1.000
v=       1.000 , h=   1.000
iter=30

MMSE,  x= 1.2842   y= 1.2842
MCEM,  x= 1.1925   y= 1.1925
XMAP,  x= 0.7157   y= 1.0121
YMAP,  x= 1.0121   y= 0.7157
XYMAP, x= 0.0000   y= 0.0000
```

Figure 6.1 - Convergence of Fisher Algorithm for Example

operation. Note again the peculiar fact that XMAP generates better output estimates, while YMAP generates better signal estimates.

6. Convergence of the Fisher Algorithms

Proving convergence of the estimates to a global minimizing solution is considerably more difficult for these Fisher problems that it was for the Bayesian problems. In general, our convergence theorems guarantee that if our estimates remain bounded, then they must converge to the set of stationary points and local minima of the appropriate cross-entropy function. However, in the Fisher problem there is no guarantee that the estimates will remain bounded, and there may not even be a finite global minimizing solution. Consider the example shown in figure 6.2, where the set X is a convex set in $\mathbf{R}^2$ bounded by a hyperbola, while Y is a (convex) half plane in $\mathbf{R}^2$. Clearly X and Y approach each other more and more closely as we go farther and farther to the right, but though the infimum of the distance between the sets is zero, no finite elements of X and Y attain this minimum distance. All of our iterative algorithms will generate estimates $\hat{x}_k$, $\hat{y}_k$ which diverge at a very slow rate toward the right.

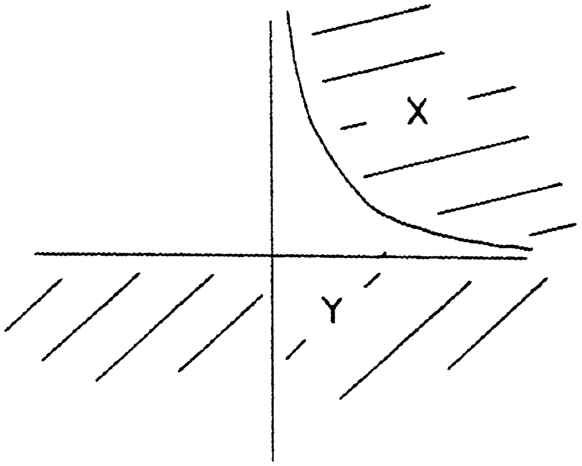


Figure 6.2 - Hyperbola Constraint Sets

When X and Y are convex, closed and non-empty, then our convergence analysis can be strengthened considerably. The key property, as in chapter 5, is that projection operators on convex sets, as well as expectation operators of truncated Gaussians on convex sets, are both non-expansive and uniformly continuous mappings. Appendix I uses this fact together with the log concavity of the model density $p(\underline{y}\,|\underline{x})$ to prove the following theorem:

Theorem 6.2 Let X and Y be convex, closed and non-empty. Start at any initial estimate $\hat{\underline{x}}_0 \in X$, $\hat{\underline{y}}_0 \in Y$. Then the sequence of estimates $(\hat{\underline{x}}_k, \hat{\underline{y}}_k)$ generated by any of our four algorithms will converge to a finite global minimizing solution of the appropriate Fisher cross-entropy function if and only if such a solution exists. Furthermore, the distance from the estimates to *any* solution $\hat{\underline{x}}$, $\hat{\underline{y}}$ decreases on each iteration:

$$\| \hat{\underline{y}}_{k+1} - \hat{\underline{y}} \|_R \le \| \hat{\underline{x}}_{k+1} - \hat{\underline{x}} \|_R \le \| \hat{\underline{y}}_k - \hat{\underline{y}} \|_R \qquad (6.6.1)$$

(In the case of XYMAP, this distance strictly decreases on each iteration.) If the Fisher problem has no finite minimizing solution, then the sequence $(\hat{x}_k, \hat{y}_k)$ is unbounded and diverges. □

The following corollary is sometimes useful:

Corollary 6.2 If X, Y are bounded, or if either X or Y is bounded and the other is convex, then all four Fisher algorithms are guaranteed to converge to a finite solution. □

Let us leave the topic of convex constraint sets with one final warning. Although we have proven that the iteration converges, it is not necessarily true that the convergence rate is linear. For example, in the problem illustrated in figure 6.3, the constraint set X is a disk of radius 1, while Y is a half plane. Clearly the unique solution to the Fisher problem is $\hat{x} = \hat{y} = (\,1\ 0\,)$.

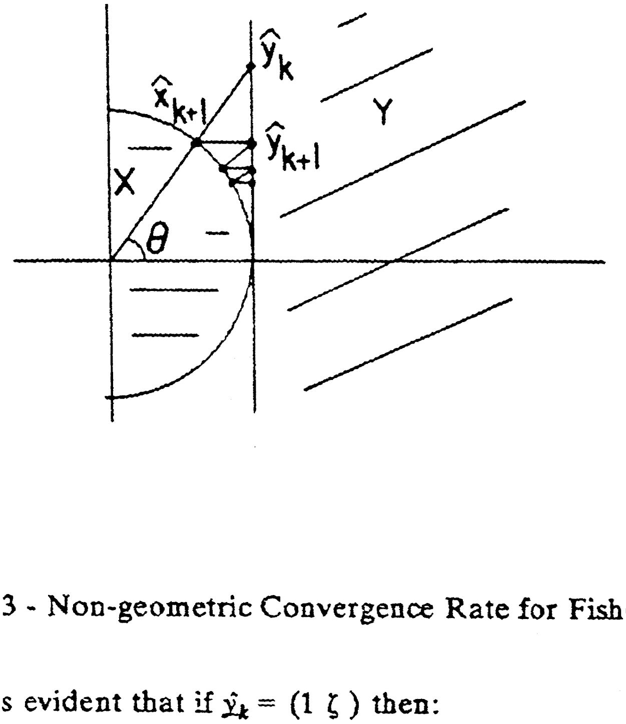


Figure 6.3 - Non-geometric Convergence Rate for Fisher Problem

From the figure, it is evident that if $\hat{y}_k = (1\ \zeta\,)$ then:

$$\|\hat{\underline{y}}_{k+1}-\hat{\underline{y}}\| = \cos(\theta_k)\ \|\hat{\underline{y}}_k-\hat{\underline{y}}\|$$

As $\hat{\underline{y}}_k \rightarrow \hat{\underline{y}}$, then $\cos(\theta_k) \rightarrow 1$ and the convergence rate will be slower than linear. We will see in the next section, however, that if the constraints are linear equalities, then the convergence rate is guaranteed to be geometric.

7. Linear Equality Constraints

When the constraint sets X and Y are linear varieties defined by the linear equations $G_x \underline{x} = \underline{y}_x$ and $G_y \underline{y} = \underline{y}_y$, then as in chapter 5, our algorithms are particularly easy to analyze. Nearly all the results proven in chapter 5 for linear equality constraints hold even when $Q = \infty$. The major change is that the convergence rate factor $\nu_x \nu_y$ equals 1, and that the set of equations defining the closed-form solution is not necessarily invertible. Despite this, if the constraint spaces are finite dimensional, we will see that both the primal and dual problems always have a (possibly non-unique) solution, and both the primal and dual iterative algorithms still converge at a geometric rate. Conjugate gradient algorithms converging in a finite number of steps can also be devised. Unfortunately, if the solution is not unique, then the noise sensitivity of our algorithms is infinite. The derivation of the dual algorithm is also quite a bit messier than in chapter 5, since the required limits may not exist. Finally, when the spaces are infinite dimensional, many of our previous results will no longer apply because new forms of degeneracy can arise. We will indicate why these difficulties occur, though we will not treat the matter in great detail (see Youla [2] or Mosca [3] for a treatment of the infinite dimensional case.) Much of the material in this section has appeared in one form or another in the literature (see, for example, [4,5,6,7]), though our eigenvalue analysis, the symmetric closed-form solutions, and the dual algorithm appear to be new.

7.1. Primal Iterative Algorithm

We proved in section 2 that the Bayesian estimation methods of chapter 5 must asymptotically converge to the solution to the corresponding Fisher estimation problem as the *a priori* signal density becomes flat. We will show later that the Fisher problem does indeed have a solution in the case of linear variety constraint spaces. Our previous proof then guarantees that as $\alpha \to 0$ our Bayesian algorithms will converge to the solution to the Fisher problem with the smallest signal energy $\|A\underline{x}\|^2_{Q_0}$. Since all our Bayesian algorithms were identical, however, this suggests that all of our Fisher estimation algorithms will also be identical. We therefore only need to consider the iterative XYMAP algorithm:

$$\hat{\underline{x}}, \hat{\underline{y}} \leftarrow \min_{\underline{x} \in X, \underline{y} \in Y} \| \underline{y} - \underline{x} \|^2_R \tag{6.7.1}$$

The primal iterative algorithm solves this problem by minimizing with respect to $\underline{x}$ and then $\underline{y}$; using Lagrange multipliers in the same manner as in chapter 5 to calculate the resulting estimates yields:

```
Primal Iterative Algorithm:
Guess $\hat{\underline{y}}_0 \in Y$
For $k = 0, 1, \cdots$
    $\hat{\underline{x}}_{k+1} = P_x \hat{\underline{y}}_k + \bar{\underline{x}}$                (6.7.2)
    $\hat{\underline{y}}_{k+1} = P_y \hat{\underline{x}}_{k+1} + \bar{\underline{y}}$
```

where:

$$\begin{aligned} P_x &= \left[I - RG_x^T(G_x R G_x^T)^{-1} G_x \right] \\ P_y &= \left[I - RG_y^T(G_y R G_y^T)^{-1} G_y \right] \\ \bar{\underline{x}} &= RG_x^T(G_x R G_x^T)^{-1} \underline{y}_x \end{aligned} \tag{6.7.3}$$

$$\bar{y} = RG_y^T(G_y RG_y^T)^{-1}y_y$$

Note the similarity between these equations and those of the Bayesian XYMAP problem in chapter 5; the only change is that by setting $\alpha=0$ and $B=I$, we get $H=I$ and $V=R$. Once again, P_x and P_y are projection matrices. To estimate the signal, we multiply the latest output estimate $\hat{y}_k$ by P_x to remove the component orthogonal to X, then add back an offset $\bar{x}$ which is the minimum norm $||\cdot||_R$ element in X (and is thus orthogonal to the null space of G_x.) The output is then reestimated by multiplying by P_y to remove the component of $\hat{x}_{k+1}$ orthogonal to Y, then adding back an offset $\bar{y}$ which is the minimum norm $||\cdot||_R$ element in Y (and is thus orthogonal to the null space of G_y.)

The only problem with this algorithm, as we will see in section 7.4, is that if the null spaces of the constraint matrices G_x and G_y overlap, then although the algorithm converges at a geometric rate, the solution it converges to will depend on the initial starting estimate $\hat{y}_0$.

7.2. Primal Algorithm - Closed-form Solution

A closed-form solution can be calculated by recognizing that any global minimizing solution $\hat{x}, \hat{y}$ must be a stationary point of the algorithm, and thus must satisfy:

$$\begin{pmatrix} I & -P_x \\ -P_y & I \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \qquad (6\ 7.4)$$

Adding P_y times the first row to the second or P_x times the second row to the first, and using $\hat{x} = P_x\hat{x} + \bar{x}$ and $\hat{y} = P_y\hat{y} + \bar{y}$ gives alternate forms:

$$\begin{aligned} \left(I - P_xP_yP_x\right)\hat{x} &= \bar{x} + P_x\left(\bar{y} + P_y\bar{x}\right) \\ \hat{y} &= P_y\hat{x} + \bar{y} \end{aligned} \qquad (6.7.5)$$

or:

$$\left(I - P_y P_x P_y\right)\hat{\underline{y}} = \bar{\underline{y}} + P_y\left(\bar{\underline{x}} + P_x \bar{\underline{y}}\right) \tag{6.7.6}$$

$$\hat{\underline{x}} = P_x \hat{\underline{y}} + \bar{\underline{x}}$$

The only problem with these closed-form solutions is that if the null spaces of G_x and G_y overlap, then although equations (6.7.5) and (6.7.6) can still be solved, the matrices in the equations are non-invertible, and there will be many different solutions to the problem.

7.3. Primal Algorithm - Eigenstructure

The proof given in Appendix J that the iterative algorithm converges and that the formulas (6.7.5) and (6.7.6) can always be solved relies heavily on a careful analysis of the eigenstructure of $P_x P_y P_x$ and $P_y P_x P_y$. This analysis is quite similar to that given in chapter 5, although the problem has several new features. Let N_x and N_y be the null spaces of the G_x and G_y matrices, and let $N_x^{\perp}$ and $N_y^{\perp}$ be their orthogonal complements:

$$N_x^{\perp} = \left\{ \underline{v} \;\middle|\; \underline{v} \in \mathbf{R}^N \quad \text{and} \quad <\underline{v},\underline{w}>_{\mathbf{R}} = 0 \;\text{ for all } \underline{w} \in N_x \right\} \tag{6.7.7}$$

and $N_y^{\perp}$ is defined similarly. Appendix J now proves:

a) All eigenvalues λ_i of $P_x P_y P_x$ are real, non-negative and less than or equal to 1. The eigenvectors $\underline{\psi}_i$ form a complete orthonormal basis with respect to the inner product $<\cdot,\cdot>_V$. Similar statements also hold for the eigenvalues of $P_y P_x P_y$ and its eigenvectors $\underline{\phi}_i$.

b) The matrices $P_x P_y P_x$ and $P_y P_x P_y$ have exactly the same non-zero eigenvalues λ_i, and there is a one-to-one correspondence between the eigenvectors corresponding to these non-zero eigenvalues:

$$\begin{aligned} \phi_i &= \frac{1}{\sqrt{\lambda_i}} \mathrm{P}_y \psi_i \\ \psi_i &= \frac{1}{\sqrt{\lambda_i}} \mathrm{P}_x \phi_i \end{aligned} \tag{6.7.8}$$

These eigenvectors ψ_i corresponding to non-zero λ_i are elements of the null space N_x of G_x, so that $\mathrm{P}_x \psi_i = \psi_i$, and the eigenvectors ϕ_i are elements of the null space N_y of G_y, so that $\mathrm{P}_y \phi_i = \phi_i$. This implies that there can be at most $\min(N-p, N-q)$ non-zero eigenvalues. If the eigenvectors ψ_i corresponding to non-zero eigenvalues are chosen to be orthonormal, then the eigenvectors ϕ_i constructed from (6.7.8) will also be orthonormal.

c) All elements of the intersection $N_x \cap N_y$ are eigenvectors of P_x and P_y, and thus also of $\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x$ and $\mathrm{P}_y \mathrm{P}_x \mathrm{P}_y$, with eigenvalue of one. All other eigenvectors of $\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x$ and $\mathrm{P}_y \mathrm{P}_x \mathrm{P}_y$ are orthogonal to $N_x \cap N_y$ and have eigenvalues strictly less than one.

d) There can be at most $\min(p, q, N-p, N-q)$ non-zero eigenvalues strictly less than 1. (See also section 7.11.)

7.4. Primal Algorithm - Convergence Rate

We can analyze the convergence rate of this algorithm by methods similar to those we used before. First of all, we recognize that the estimates $\hat{x}_k$, $\hat{y}_k$ can be written in the form:

$$\begin{aligned} \hat{x}_k &= \mathrm{P}_x \hat{x}_k + \bar{x} \\ \hat{y}_k &= \mathrm{P}_y \hat{y}_k + \bar{y} \end{aligned} \tag{6.7.9}$$

Substituting these expressions into our primal iteration, and using some algebra gives:

$$\begin{aligned} \hat{x}_{k+1} - \hat{x}_k &= \mathrm{P}_x \mathrm{P}_y (\hat{y}_k - \hat{y}_{k-1}) \\ \hat{y}_{k+1} - \hat{y}_k &= \mathrm{P}_y \mathrm{P}_x (\hat{x}_{k+1} - \hat{x}_k) \end{aligned} \tag{6.7.10}$$

Analyzing the convergence properties of this algorithm is more difficult than in chapter 5, because with $\alpha=0$ our previous analysis would give $\nu_x \nu_y = 1$, which only implies that:

$$\|\hat{\underline{y}}_{k+1} - \hat{\underline{y}}_k\|_R^2 \le \|\hat{\underline{x}}_{k+1} - \hat{\underline{x}}_k\|_R^2 \le \|\hat{\underline{y}}_k - \hat{\underline{y}}_{k-1}\|_R^2 \qquad (6.7.11)$$

This is no longer sufficient to prove convergence. A more careful analysis must consider two different cases.

Case 1: $N_x \cap N_y = \{\underline{0}\}$

The simplest case is when the intersection of the null spaces of G_x and G_y is the zero vector, $N_x \cap N_y = \{\underline{0}\}$. In finite dimensional spaces $\mathbf{R}^N$, this is equivalent to:

$$\mathbf{R}^N = (N_x \cap N_y)^\perp = \text{row space of } \begin{pmatrix} G_x \\ G_y \end{pmatrix} \qquad (6.7.12)$$

and thus $N_x \cap N_y = \{\underline{0}\}$ is equivalent to requiring that the rank of $\begin{pmatrix} G_x \\ G_y \end{pmatrix}$ equals N, the dimension of $\underline{x}$ and $\underline{y}$. Since the null space $N_x \cap N_y$ is empty, according to property (c) of section 7.3 all the eigenvalues of $P_x P_y P_x$ must be strictly less than 1. If there are a finite number of non-zero eigenvalues (a situation that will occur if $\min(N-p, N-q) < \infty$), then one of these eigenvalues λ_{max} must be the largest, and it will be strictly less than one, $\lambda_{max} < 1$. In this case, Appendix J proves that the matrices in (6.7.5) and (6.7.6) are invertible (since their smallest eigenvalue will be $1-\lambda_{max} > 0$) and thus there is a unique solution to the Fisher problem. Furthermore,

$$\|P_x P_y\|_R^2 = \|P_y P_x\|_R^2 = \lambda_{max} < 1 \qquad (6.7.13)$$

and the iterative algorithm will converge at a geometric rate to the unique solution $(\hat{\underline{x}}, \hat{\underline{y}})$:

$$\|\hat{\underline{y}}_{k+1} - \hat{\underline{y}}\|_R^2 \le \|P_y P_x\|_R^2 \, \|\hat{\underline{x}}_{k+1} - \hat{\underline{x}}\|_R^2 = \lambda_{max} \, \|\hat{\underline{x}}_{k+1} - \hat{\underline{x}}\|_R^2 \qquad (6.7.14)$$

$$\|\hat{\underline{x}}_{k+1} - \hat{\underline{x}}\|_R^2 \le \|P_x P_y\|_R^2 \, \|\hat{\underline{y}}_k - \hat{\underline{y}}\|_R^2 = \lambda_{max} \, \|\hat{\underline{y}}_k - \hat{\underline{y}}\|_R^2 \qquad (6.7.15)$$

Case 2: $N_x \cap N_y \neq \{0\}$

If the null spaces of G_x and G_y overlap, $N_x \cap N_y \neq \{\underline{0}\}$, or equivalently in finite dimensions, if rank $\begin{pmatrix} G_x \\ G_y \end{pmatrix} < N$, then the analysis is more difficult. The problem, as noted in property (c) of section 7.3 is that every element of $N_x \cap N_y$ is an eigenvector of $P_x P_y P_x$ and $P_y P_x P_y$ with eigenvalue of 1. Thus the matrices $(I - P_x P_y P_x)$ and $(I - P_y P_x P_y)$ will have a non-trivial null space $N_x \cap N_y$, and will not be invertible. Thus although Appendix J proves that the equations still have a solution, that solution is not unique, since $(\hat{\underline{x}} + \underline{v}, \hat{\underline{y}} + \underline{v})$ will also be a solution for any $\underline{v} \in N_x \cap N_y$.

Proving convergence of the iterative algorithm is also more difficult in this case because $\|P_y P_x\|_R^2 = \|P_x P_y\|_R^2$ will be equal to the largest eigenvalue of $P_x P_y P_x$ and $P_y P_x P_y$, which is one. To prove convergence, we will have to decompose the space R^N into a direct sum of $N_x \cap N_y$ and its orthogonal complement:

$$R^N = (N_x \cap N_y) \oplus (N_x \cap N_y)^{\perp} \tag{6.7.16}$$

Then we can show that the matrices $P_y P_x$ and $P_x P_y$ map the space $(N_x \cap N_y)$ onto itself, and also map $(N_x \cap N_y)^{\perp}$ onto itself. Assuming that the number of non-zero eigenvalues less than 1 is finite (a condition that is guaranteed if $\min(p, q, N-p, N-q) < \infty$), the maximum eigenvalue λ_{max} of $P_x P_y P_x$ and $P_y P_x P_y$ on the set $(N_x \cap N_y)^{\perp}$ will be strictly less than 1. Let us decompose our initial guess $\hat{\underline{y}}_0$ into a component $\underline{v}_0$ in $N_x \cap N_y$ and a component $\tilde{\underline{y}}_0$ orthogonal to $N_x \cap N_y$:

$$\hat{\underline{y}}_0 = \tilde{\underline{y}}_0 + \underline{v}_0 \qquad \text{where } \tilde{\underline{y}}_0 \in (N_x \cap N_y)^{\perp} \quad \underline{v}_0 \in (N_x \cap N_y) \tag{6.7.17}$$

Then Appendix J shows that if there is no quantization noise in the calculation, then the algorithm converges at a geometric rate to the solution $(\hat{\underline{x}}_{min} + \underline{v}_0, \hat{\underline{y}}_{min} + \underline{v}_0)$:

$$\|\hat{\underline{y}}_{k+1}-(\hat{\underline{y}}_{min}+\underline{v}_0)\|_R^2 \le \lambda_{max}\,\|\hat{\underline{x}}_{k+1}-(\hat{\underline{x}}_{min}+\underline{v}_0)\|_R^2 \tag{6.7.18}$$
$$\le \lambda_{max}^2\,\|\hat{\underline{y}}_k-(\hat{\underline{y}}_{min}+\underline{v}_0)\|_R^2$$

where $\hat{\underline{x}}_{\text{min}}$ and $\hat{\underline{y}}_{\text{min}}$ are orthogonal to $N_x \cap N_y$ and are thus the solutions to the Fisher XYMAP problem with the smallest norm $\|\cdot\|_R$.

$$\begin{aligned}\|\hat{\underline{x}}_{min}\|_R^2 &\le \|\hat{\underline{x}}\|_R^2\\ \|\hat{\underline{y}}_{min}\|_R^2 &\le \|\hat{\underline{y}}\|_R^2\end{aligned} \qquad \text{for any other solution } \hat{\underline{x}},\hat{\underline{y}} \tag{6.7.19}$$

and $(\hat{\underline{x}}_{min}+\underline{v}_0,\hat{\underline{y}}_{min}+\underline{v}_0)$ is the solution to the Fisher problem which is "closest" to $\hat{\underline{y}}_0$. Thus if we wish to calculate the minimum norm solution $\hat{\underline{x}}_{min}$, $\hat{\underline{y}}_{min}$, we need only start with an initial output estimate $\hat{\underline{y}}_0$ which is orthogonal to $N_x \cap N_y$; the easiest way to do this is to choose $\hat{\underline{y}} = \overline{\underline{y}}$, the minimum norm element in Y.

To summarize: the iterative algorithm converges at a geometric rate λ_{max} to the nearest solution to the Fisher problem, where λ_{max} is the largest eigenvalue of $P_x P_y P_x$ and $P_y P_x P_y$ which is less than 1. If $N_x \cap N_y = \{\underline{0}\}$ then the Fisher problem has a unique solution.

7.5. Primal Algorithm - Noise Sensitivity

Beware that for practical computation, this proof of convergence must be treated with some caution. If our computation uses finite arithmetic, so that quantization errors occur, then the same noise analysis used in chapter 5 suggests that the error between our computed estimate $\tilde{\underline{y}}_k$ and the exact estimate $\hat{\underline{y}}_k$ will be:

$$\tilde{\underline{y}}_k - \hat{\underline{y}}_k = \sum_{m=0}^{k-1} (P_y P_x)^m \left[(\underline{\delta}_y + \underline{\epsilon}_{y_{k-m}}) + P_y(\underline{\delta}_x + \underline{\epsilon}_{x_{k-m}}) \right] \tag{6.7.20}$$

Let us decompose the errors into components belonging to and orthogonal to $N_x \cap N_y$:

$$(\underline{\delta}_y + \underline{\epsilon}_{y_k}) + P_y(\underline{\delta}_x + \underline{\epsilon}_{x_k}) = \underline{\Delta}_k + \underline{\Delta}_k^{\perp} \tag{6.7.21}$$
$$\text{where } \underline{\Delta}_k \in (N_x \cap N_y) \quad \text{and} \quad \underline{\Delta}_k^{\perp} \in (N_x \cap N_y)^{\perp}$$

Then Δ_k is an eigenvector of P_y and P_x with eigenvalue 1. Liberally using the fact that $P_x P_x = P_x$ and $P_y P_y = P_y$ gives:

$$\tilde{\underline{y}}_k - \hat{\underline{y}}_k = \sum_{m=0}^{k-1} \Delta_{k-m} + \Delta_k^{\perp} + \sum_{m=1}^{k-1} \left[(P_y P_x)(P_x P_y) \right]^{m-1} (P_y P_x)\, \Delta_{k-m}^{\perp} \qquad (6.7.22)$$

Appendix J now shows that if the average energy of the errors $\|\Delta_k\|_R^2$ and $\|\Delta_k^{\perp}\|_R^2$ is about $\bar{\Delta}$ and $\bar{\Delta}^{\perp}$ respectively, then:

$$\|\tilde{\underline{y}}_k - \hat{\underline{y}}_k\|_R^2 \leq k\bar{\Delta} + \left(1 + \frac{\lambda_{max}(1-\lambda_{max}^{2k-1})}{1-\lambda_{max}^2} \right) \bar{\Delta}^{\perp} \qquad (6.7.23)$$

The iteration is unable to affect any component in the null space $N_x \cap N_y$; thus the components of the computation error in $N_x \cap N_y$ simply accumulate, leading to a linearly growing error term (the first term in (6.7.23).) Components of the computation noise orthogonal to $N_x \cap N_y$, however, are reduced by further iterations at a rate λ_{max}^2. Thus the error that accumulates orthogonal to $N_x \cap N_y$ (the second term in (6.7.23)) is bounded above and proportional to $\frac{1}{1-\lambda_{max}^2}$. If $N_x \cap N_y \neq \{\underline{0}\}$ so that the solution is unique, then the first term does not exist, and the computational error is bounded. If $N_x \cap N_y \neq \{\underline{0}\}$, however, the noise sensitivity is infinite.

This infinite noise sensitivity is not necessarily fatal. Adding any vector in $N_x \cap N_y$ to a solution simply gives another solution; therefore since the linearly growing error term is in $N_x \cap N_y$, it changes which solution the algorithm is heading toward, but the estimates will still converge geometrically to the set of solutions. Of course, if the error term in $N_x \cap N_y$ grows infinitely large as $k \to \infty$, then the estimates will be unbounded. Conceptually, the problem results from the algorithm's attempt to solve the perturbed problem:

$$(I - P_y P_x P_y)\, \underline{y} = \left[(\bar{\underline{y}} + \underline{\delta}_y) + P_y \left((\bar{\underline{x}} + \underline{\delta}_x) + P_x (\bar{\underline{y}} + \underline{\delta}_y) \right) \right] \qquad (6.7.24)$$

where the perturbation of the right hand side has a component orthogonal to the range of $I - P_y P_x P_y$, and thus the equation no longer has any solution at all.

7.6. Primal Problem - Infinite Dimensions

Much of this analysis can be carried over to infinite dimensional Hilbert Space (see Youla [2] †), but new complications arise. If the number of non-zero eigenvalues of $P_x P_y P_x$ and $P_y P_x P_y$ is finite, then we pointed out that the problem always has a solution, and the convergence rate is geometric because the largest eigenvalue λ_{max} of $P_x P_y P_x$ and $P_y P_x P_y$ on the space orthogonal to $N_x \cap N_y$ must be strictly less than 1. If $\min(N-p, N-q) = \infty$ and these matrices have an infinite number of non-zero eigenvalues, however, then the eigenvalues of $P_x P_y P_x$ in the space $(N_x \cap N_y)^{\perp}$ can get arbitrarily close to 1, and the supremum of these, λ_{max}, may in fact equal 1. The iteration will still converge to a solution, since it is strictly non-expansive, but even if $N_x \cap N_y = \{0\}$, the convergence rate can be slower than geometric, and the noise sensitivity can be infinitely large.

7.7. Primal Problem - Geometric Angle Interpretation

Youla also pointed out an interesting geometric interpretation of the quantity $\lambda_{max} = ||P_y P_x||_R^2 = ||P_x P_y||_R^2 \leq 1$, which is the factor which determines the convergence rate and noise sensitivity of the algorithm. In the simple example given in section 5, the magnitude of the estimates generated by the XYMAP algorithm satisfy:

$$||\hat{y}_{k+1}||_R = \cos\theta \ ||\hat{x}_{k+1}||_R = \cos^2\theta \ ||\hat{y}_k||_R \qquad (6.7.25)$$

Thus the estimates converge to the limiting value $\hat{x} = \hat{y} = 0$ at a rate which is determined

† Note that while Youla's arguments are perfectly valid, his error bounds and convergence rate limits could be made much tighter by using our analysis.

by the cosine of the angle between the line X and the line Y. This idea was exploited in depth by Youla. Let us define the angle between two arbitrary vectors $\underline{x}$ and $\underline{y}$ by using the inner product $<\underline{x},\underline{y}>_R$:

$$\cos\theta(\underline{x},\underline{y}) = \frac{|<\underline{x},\underline{y}>_R|}{\|\underline{x}\|_R \|\underline{y}\|_R} \tag{6.7.26}$$

By Schwartz's inequality:

$$\cos\theta(\underline{x},\underline{y}) \le 1 \tag{6.7.27}$$

To define the angle between two linear varieties X and Y, note that X and Y are cosets of the linear subspaces N_x and N_y. Since they differ from N_x and N_y only by a constant offset, X and Y are "parallel" to N_x and N_y respectively. The angle between X and Y can thus be defined as the angle between N_x and N_y, which in turn is defined as the smallest angle between any two elements $\underline{x} \in N_x$, $\underline{y} \in N_y$:

$$\theta(X,Y) = \inf_{\underline{x} \in N_x, \underline{y} \in N_y} \theta(\underline{x},\underline{y}) \tag{6.7.28}$$

or equivalently, since $\cos\theta$ is a monotonic decreasing function of θ in the interval $0 \le \theta \le \frac{\pi}{2}$:

$$\cos\theta(X,Y) = \sup_{\underline{x} \in N_x, \underline{y} \in N_y} \frac{|<\underline{x},\underline{y}>_R|}{\|\underline{x}\|_R \|\underline{y}\|_R} \tag{6.7.29}$$

Youla then showed that:

<u>Lemma:</u> $\cos\theta(X,Y) = \|P_y P_x\|_R = \|P_x P_y\|_R = \lambda_{max}^{\frac{1}{2}}$

The proof is given in Appendix J. Our previous discussion of the properties of our iterative algorithm can now be restated using this concept of angles. If the angle between X and Y is greater than zero, $\cos\theta(X,Y)<1$, then $N_x \cap N_y = \{\underline{0}\}$, the problem will have a unique solution, the iteration will converge at a rate $\cos^2\theta(X,Y)$ and the noise sensitivity will be bounded. If, however, $\cos\theta(X,Y)=1$, then the algorithm still converges, but the noise sensitivity is infinite.

7.8. Dual Algorithm

Because the Fisher XYMAP objective function (6.7.1) is only a positive semi-definite, rather than a positive definite quadratic funetion, deriving a dual algorithm via the Lagrange multiplier saddlepoint theorem is no longer always possible. We will instead derive a dual algorithm for this problem by taking the limit as $\alpha \to 0$ in our Bayesian dual algorithm of chapter 5, then verifying that the method actually gives the same answer as the primal algorithm. We stress at the outset that this approach is unconventional, and in fact does not entirely work without substantial fudging.

Let $Q = \frac{1}{\alpha} Q_0$ in our Bayesian dual objective function (5.9.17). Solving for $\underline{\lambda}_x$, $\underline{\lambda}_y$, letting $\hat{Q} = A^{-1}Q_0A^{-T}$, then transforming to the variables $\underline{\rho}_x$, $\underline{\rho}_y$, and finally letting $\alpha \to 0$, gives:

$$\begin{pmatrix} \hat{\underline{\rho}}_x \\ \hat{\underline{\rho}}_y \end{pmatrix} = \begin{pmatrix} \hat{Q}G_x^T & 0 \\ 0 & \hat{Q}G_y^T \end{pmatrix} \begin{pmatrix} G_x\hat{Q}G_x^T & G_x\hat{Q}G_y^T \\ G_y\hat{Q}G_x^T & G_y\hat{Q}G_y^T \end{pmatrix}^{-1} \begin{pmatrix} \underline{y}_x \\ \underline{y}_y \end{pmatrix} \tag{6.7.30}$$

$$\begin{pmatrix} \hat{\underline{x}} \\ \hat{\underline{y}} \end{pmatrix} = \begin{pmatrix} I & I \\ I & I \end{pmatrix} \begin{pmatrix} \hat{\underline{\rho}}_x \\ \hat{\underline{\rho}}_y \end{pmatrix}$$

Beware that in deriving these equations, the estimated Lagrange multipliers $\hat{\underline{\lambda}}_x$, $\hat{\underline{\lambda}}_y$ tend to $\underline{0}$ as $\alpha \to 0$; this effect is canceled out, however, in transforming to the variables $\hat{\underline{\rho}}_x$, $\hat{\underline{\rho}}_y$. Next note that the matrix R has disappeared, and its role has been taken over, apparently, by the *a priori* signal covariance matrix $\hat{Q} = A^{-1}Q_0A^{-T}$. This is rather peculiar, since in the primal algorithm it is the matrix Q which disappears, not R. Another anomaly is that the formula deriving $\hat{\underline{x}}$, $\hat{\underline{y}}$ from $\hat{\underline{\rho}}_x$, $\hat{\underline{\rho}}_y$ always sets $\hat{\underline{x}} = \hat{\underline{y}}$. Finally, note that the matrix on the right side of the equation for $\hat{\underline{\rho}}_x$, $\hat{\underline{\rho}}_y$ is not always invertible.

Blithely ignoring these difficulties, we will state a dual iterative algorithm for computing $\hat{\lambda}_x$, $\hat{\lambda}_y$ and $\hat{x}$, $\hat{y}$. If the constraint sets X and Y overlap, so that there exists a single vector $\hat{x}=\hat{y}$ meeting both the signal and output constraints, then we will show that the formula in (6.7.30) gives the solution $(\hat{x}_{min};\hat{y}_{min})$ to the Fisher XYMAP problem (6.7.1) with the least energy $\|Ax\|^2_{Q_0}$. Furthermore, the iterative dual algorithm will converge at a geometric rate to this solution. Thus the dual algorithm gives exactly the same answer as our Bayesian algorithm in the limit $\alpha \rightarrow 0$. This is also in sharp contrast to the primal algorithm, which will only converge to the nearest solution. Unfortunately, in the dual algorithm if there is no single vector meeting all the constraints, then we will show that the formula in (6.7.30) has no solution at all. It is just a fortunate stroke of luck, therefore, that if we set $\tilde{Q}=R$ then the dual algorithm we present will still converge at a geometric rate to the minimal energy solution to the Fisher XYMAP problem. We also present a dual closed-form solution for $\hat{\lambda}_x$, $\hat{\lambda}_y$ and $\hat{x}$, $\hat{y}$ which works under all circumstances.

We derive our dual algorithm by setting $B=I$ and letting $\alpha \rightarrow 0$ in our Bayesian dual iterative algorithm. Noting that $H \rightarrow I$, $V_x \rightarrow \frac{1}{\alpha}\tilde{Q}$, $V_y \rightarrow \frac{1}{\alpha}\tilde{Q}$, we get:

Dual Iterative Algorithm:

Guess $\hat{\mu}_{y_0}$

For $k=0,1,\cdots$

$$\hat{\mu}_{x_{k+1}} = -\mathrm{Q}_x \hat{\mu}_{y_k} + \bar{\mu}_x \qquad (6.7.31)$$

$$\hat{\mu}_{y_{k+1}} = -\mathrm{Q}_y \hat{\mu}_{x_{k+1}} + \bar{\mu}_y$$

After sufficient iterations calculate:

$$\hat{x}_{k+1} = \hat{\mu}_{x_{k+1}} + \hat{\mu}_{y_k}$$

$$\hat{y}_{k+1} = \hat{\mu}_{x_{k+1}} + \hat{\mu}_{y_{k+1}}$$

where:

$$\begin{aligned} \mathrm{Q}_x &= \hat{\mathrm{Q}}\mathrm{G}_x \left(\mathrm{G}_x \hat{\mathrm{Q}} \mathrm{G}_x^{\mathrm{T}}\right)^{-1} \mathrm{G}_x^{\mathrm{T}} \\ \mathrm{Q}_y &= \hat{\mathrm{Q}}\mathrm{G}_y \left(\mathrm{G}_y \hat{\mathrm{Q}} \mathrm{G}_y^{\mathrm{T}}\right)^{-1} \mathrm{G}_y^{\mathrm{T}} \\ \bar{\mu}_x &= \hat{\mathrm{Q}}\mathrm{G}_x \left(\mathrm{G}_x \hat{\mathrm{Q}} \mathrm{G}_x^{\mathrm{T}}\right)^{-1} \Upsilon_x \\ \bar{\mu}_y &= \hat{\mathrm{Q}}\mathrm{G}_y \left(\mathrm{G}_y \hat{\mathrm{Q}} \mathrm{G}_y^{\mathrm{T}}\right)^{-1} \Upsilon_y \end{aligned} \qquad (6.7.32)$$

The structure of this algorithm is virtually identical to that of the primal algorithm. Q_x and Q_y are projection matrices onto the *orthogonal complements* $N_x^{\perp}$ and $N_y^{\perp}$ of the null spaces of G_x and G_y. (In the primal algorithm, P_x and P_y project onto the null spaces N_x and N_y.) Starting with an output multiplier estimate $\hat{\mu}_{y_0}$, we estimate the signal multiplier by projecting $\hat{\mu}_{y_k}$ onto the orthogonal complement $N_x^{\perp}$, then adding an offset $\bar{\mu}_x$. The output multiplier is then reestimated by projecting $\hat{\mu}_{x_{k+1}}$ back onto the orthogonal complement null space $N_y^{\perp}$ and adding an offset $\bar{\mu}_y$. After sufficient iteration, the signal and output are estimated by combining the appropriate multipliers.

It is not hard to show that $\mathrm{G}_x \hat{x}_k = \Upsilon_x$ and $\mathrm{G}_y \hat{y}_k = \Upsilon_y$, so that $\hat{x}_k \in X$ and $\hat{y}_k \in Y$ as claimed. More importantly, rearranging the equations of the iteration gives:

$$\begin{aligned} \hat{x}_{k+1} &= (I-Q_x)\hat{y}_k + \bar{\mu}_x \\ \hat{y}_{k+1} &= (I-Q_y)\hat{x}_{k+1} + \bar{\mu}_y \end{aligned} \tag{6.7.33}$$

This is precisely the primal algorithm that would result if we tried to solve:

$$\hat{x}, \hat{y} = \min_{x \in X, y \in Y} \| y - x \|_{\tilde{Q}}^2 \tag{6.7.34}$$

Thus despite the flaky derivation, this dual algorithm still solves the right type of problem; the only difficulty is that we seem to have switched to a different norm. We could fix this simply by setting $\tilde{Q}=R$ in our dual algorithm; then the dual algorithm would exactly solve the same problem as our primal algorithm. Moreover, the exact correspondence between primal and dual, expressed in (6.7.33) guarantees that the final signal and output estimates generated by this dual algorithm must converge at a geometric rate to a solution to the Fisher problem.

The importance of the dual algorithm is that we do not actually compute the signal and output estimates until the algorithm terminates. The projection operators used are "orthogonal" to those used in the primal algorithm, projecting onto the orthogonal complements of the null spaces of G_x and G_y, rather than onto the null spaces themselves. More importantly, by only computing estimates of the multipliers $\hat{\mu}_{x_k}$, $\hat{\mu}_{y_k}$, we are dealing with a problem whose dimensions can be significantly smaller than the primal problem, and which therefore may be much more convenient to solve. In analyzing the dual algorithm, therefore, it is important to consider the convergence properties of the multipliers $\hat{\mu}_{x_k}$, $\hat{\mu}_{y_k}$ themselves, rather than just the signal and output estimates.

The approach we use is quite similar to that used in our primal algorithm. Note, first of all, that the signal multiplier $\hat{\mu}_{x_k}$ and the offset $\bar{\mu}_x$ are both elements of the orthogonal complement null space $N_x^{\perp}$, while the output multiplier $\hat{\mu}_{y_k}$ and the offset $\bar{\mu}_y$

are both elements of the orthogonal complement null space $N_y^{\perp}$. This implies that $Q_x \hat{\underline{\rho}}_x = \hat{\underline{\rho}}_x$ and $Q_y \hat{\underline{\rho}}_y = \hat{\underline{\rho}}_y$. Combining this with the iteration equations gives:

$$\begin{aligned} \hat{\underline{\rho}}_{x_{k+1}} - \hat{\underline{\rho}}_{x_k} &= -Q_x Q_y \left(\hat{\underline{\rho}}_{y_k} - \hat{\underline{\rho}}_{y_{k-1}} \right) \\ \hat{\underline{\rho}}_{y_{k+1}} - \hat{\underline{\rho}}_{y_k} &= -Q_y Q_x \left(\hat{\underline{\rho}}_{x_{k+1}} - \hat{\underline{\rho}}_{x_k} \right) \end{aligned} \tag{6.7.35}$$

The fact that the projection operators are contraction mappings then guarantees that:

$$\| \hat{\underline{\rho}}_{y_{k+1}} - \hat{\underline{\rho}}_{y_k} \|_{\hat{Q}} \leq \| \hat{\underline{\rho}}_{x_{k+1}} - \hat{\underline{\rho}}_{x_k} \|_{\hat{Q}} \leq \| \hat{\underline{\rho}}_{y_k} - \hat{\underline{\rho}}_{y_{k-1}} \|_{\hat{Q}} \tag{6.7.36}$$

This, however, is not sufficient to prove convergence of the multipliers. In fact, close analysis will show that in certain cases the multiplier estimates actually grow infinitely large, despite the fact that the corresponding signal and output estimates converge at a geometric rate. We will divide the analysis into two cases:

Case 1: $N_x^{\perp} \cap N_y^{\perp} = \{\underline{0}\}$

Suppose that the orthogonal complement null spaces $N_x^{\perp}$ and $N_y^{\perp}$ of G_x and G_y do not intersect except at $\underline{0}$. This is equivalent to saying that the constraint matrix has full row rank:

$$\operatorname{rank} \begin{pmatrix} G_x \\ G_y \end{pmatrix} = p + q \tag{6.7.37}$$

In particular, since this matrix has dimension $(p+q) \times N$, we will need $p + q \leq N$. This also implies that:

$$0 = \| G_x^T \underline{\lambda}_x + G_y^T \underline{\lambda}_y \|_{\hat{Q}^{-1}}^2 = \begin{pmatrix} \underline{\lambda}_x^T & \underline{\lambda}_y^T \end{pmatrix} \begin{pmatrix} G_x \hat{Q} G_x^T & G_y \hat{Q} G_x^T \\ G_y \hat{Q} G_x^T & G_y \hat{Q} G_y^T \end{pmatrix} \begin{pmatrix} \underline{\lambda}_x \\ \underline{\lambda}_y \end{pmatrix} \tag{6.7.38}$$

if and only if $\underline{\lambda}_x = \underline{0}$ and $\underline{\lambda}_y = \underline{0}$. Thus this matrix on the right hand side must be positive definite and invertible, and the closed form solution suggested in (6.7.30) for $\hat{\underline{\rho}}_x$ and $\hat{\underline{\rho}}_y$ really can be solved.

Now to explain the anomalies of the closed-form solution (6.7.30). Because $\begin{pmatrix} G_x \\ G_y \end{pmatrix}$ has full row rank, its range must include all of $\mathbf{R}^{p+q}$, and its null space will have dimension $N-p-q$. Thus there exists at least one vector $\hat{\underline{x}} = \hat{\underline{y}}$ such that:

$$\begin{pmatrix} G_x \\ G_y \end{pmatrix} \hat{\underline{x}} = \begin{pmatrix} \underline{\Upsilon}_x \\ \underline{\Upsilon}_y \end{pmatrix} \tag{6.7.39}$$

This $\hat{\underline{x}}$ satisfies both the signal and output constraints simultaneously, $\hat{\underline{x}} \in X$ and $\hat{\underline{x}} \in Y$; also at this solution $\hat{\underline{x}} = \hat{\underline{y}}$ the Fisher XYMAP objective function is zero, so that $\hat{\underline{x}} = \hat{\underline{y}}$ is not only a solution to the "wrong" Fisher problem (6.7.34) with norm $||\cdot||_{\hat{Q}}$, but is also a solution to the original Fisher XYMAP problem (6.7.1) with norm $||\cdot||_{R}$.

If the null space of $\begin{pmatrix} G_x \\ G_y \end{pmatrix}$ is nontrivial, then there will be many possible solutions to the Fisher XYMAP problem. Clearly, if $\underline{v}$ is any element of $N_x \cap N_y$ so that $\begin{pmatrix} G_x \\ G_y \end{pmatrix} \underline{v} = \underline{0}$, then the vector $\hat{\underline{x}} + \underline{v}$ is still an element of both X and Y, and must also solve the XYMAP problem. As proved in Appendix J, however, the dual algorithm always calculates the solution with the minimal signal energy $||A\hat{\underline{x}}||^2_{Q_0}$. (This is unlike the primal algorithm, which must be initialized at an output estimate $\hat{\underline{y}}_0 \in (N_x \cap N_y)^{\perp}$ in order to find the minimal energy solution.)

Taking the norms of all sides in equation (6.7.35) gives:

$$\begin{aligned} ||\hat{\underline{\mu}}_{x_{k+1}} - \hat{\underline{\mu}}_{x_k}||^2_{\hat{Q}} &\leq ||Q_x Q_y||^2_{\hat{Q}} \, ||\hat{\underline{\mu}}_{y_k} - \hat{\underline{\mu}}_{y_{k-1}}||^2_{\hat{Q}} \\ ||\hat{\underline{\mu}}_{y_{k+1}} - \hat{\underline{\mu}}_{y_k}||^2_{\hat{Q}} &\leq ||Q_y Q_x||^2_{\hat{Q}} \, ||\hat{\underline{\mu}}_{x_{k+1}} - \hat{\underline{\mu}}_{x_k}||^2_{\hat{Q}} \end{aligned} \tag{6.7.40}$$

Appendix J proves that when $N_x^{\perp} \cap N_y^{\perp} = \{\underline{0}\}$, then the convergence factors are strictly less than one: $\lambda_{max} = ||Q_x Q_y||^2_{\hat{Q}} = ||Q_y Q_x||^2_{\hat{Q}} < 1$. Thus the multiplier estimates (and the corresponding signal and output estimates) converge at a geometric rate λ_{max} to the unique solution to the dual problem.

Closed-form solutions for the multipliers $\hat{\underline{\rho}}_x$ and $\hat{\underline{\rho}}_y$ can be derived by the usual methods. Recognizing that the solution must be a stationary point of the iterative algorithm gives:

$$\begin{pmatrix} I & Q_x \\ Q_y & I \end{pmatrix} \begin{pmatrix} \hat{\underline{\rho}}_x \\ \hat{\underline{\rho}}_y \end{pmatrix} = \begin{pmatrix} \bar{\underline{\rho}}_x \\ \bar{\underline{\rho}}_y \end{pmatrix} \tag{6.7.41}$$

Subtracting Q_x times the second row from the first, or Q_y times the first row from the second, and recognizing that $Q_x \hat{\underline{\rho}}_x = \hat{\underline{\rho}}_x$ and $Q_y \hat{\underline{\rho}}_y = \hat{\underline{\rho}}_y$ gives the alternative formulas:

$$\begin{aligned} \left(I - Q_x Q_y Q_x \right) \hat{\underline{\rho}}_x &= \bar{\underline{\rho}}_x - Q_x \bar{\underline{\rho}}_y \\ \hat{\underline{\rho}}_y &= -Q_y \hat{\underline{\rho}}_x + \bar{\underline{\rho}}_y \end{aligned} \tag{6.7.42}$$

or:

$$\begin{aligned} \left(I - Q_y Q_x Q_y \right) \hat{\underline{\rho}}_y &= \bar{\underline{\rho}}_y - Q_y \bar{\underline{\rho}}_x \\ \hat{\underline{\rho}}_x &= -Q_x \hat{\underline{\rho}}_y + \bar{\underline{\rho}}_x \end{aligned} \tag{6.7.43}$$

<u>Case 2:</u> $N_x^{\perp} \cap N_y^{\perp} \neq \{\underline{0}\}$

When $N_x^{\perp}$ and $N_y^{\perp}$ have a nontrivial intersection, then the analysis is much more difficult. This condition implies that:

$$\operatorname{rank} \begin{pmatrix} G_x \\ G_y \end{pmatrix} < p + q \tag{6.7.44}$$

which means that the matrix:

$$\begin{pmatrix} G_x \hat{Q} G_x^T & G_x \hat{Q} G_y^T \\ G_y \hat{Q} G_x^T & G_y \hat{Q} G_y^T \end{pmatrix} = \begin{pmatrix} G_x \\ G_y \end{pmatrix} \hat{Q} \begin{pmatrix} G_x^T \, G_y^T \end{pmatrix} \geq 0 \tag{6.7.45}$$

will only be positive semidefinite and will not be invertible. Thus if $N_x^{\perp} \cap N_y^{\perp} \neq \underline{0}$, then the formula in (6.7.30) for $\hat{\underline{\rho}}_x$, $\hat{\underline{\rho}}_y$, which uses the inverse of this matrix, may not be solvable. The derivation of our dual algorithm when $N_x^{\perp} \cap N_y^{\perp} \neq \{\underline{0}\}$ is therefore completely bogus, and it is just lucky that it still solves the correct problem, provided we set $\hat{Q}=R$.

Analyzing the convergence behavior of the dual algorithm and finding a correct closed-form solution when $N_x^{\perp} \cap N_y^{\perp} \neq \{\underline{0}\}$ is rather complicated, so we leave the details to Appendix J. This Appendix proves:

<u>Theorem 6.3</u> The following procedure gives a closed-form solution to our dual problem in all circumstances (provided we set $\hat{Q}=R$).

a) Compute a best least squares solution $\hat{\underline{\rho}}_{x_0}$ to:

$$\left(I - Q_x Q_y Q_x \right) \underline{\rho}_x = \bar{\underline{\rho}}_x - Q_x \bar{\underline{\rho}}_y \tag{6.7.46}$$

with respect to the inner product $<\cdot,\cdot>_{\hat{Q}}$. (There may be many solutions - any of them will do.)

b) Compute an output multiplier estimate and a second signal estimate $\hat{\underline{\rho}}_{x_1}$:

$$\begin{aligned} \hat{\underline{\rho}}_{y_1} &= -Q_y \hat{\underline{\rho}}_{x_0} + \bar{\underline{\rho}}_y \\ \hat{\underline{\rho}}_{x_1} &= -Q_x \hat{\underline{\rho}}_{y_1} + \bar{\underline{\rho}}_y \end{aligned} \tag{6.7.47}$$

c) The signal and output estimates are then given by:

$$\begin{aligned} \hat{\underline{x}}_{min} &= \hat{\underline{\rho}}_{x_1} + \hat{\underline{\rho}}_{y_1} \\ \hat{\underline{y}}_{min} &= \hat{\underline{\rho}}_{x_0} + \hat{\underline{\rho}}_{y_1} \end{aligned} \tag{6.7.48}$$

These signal and output estimates will be the minimal energy solution to the Fisher XYMAP problem:

$$\hat{\underline{x}}, \hat{\underline{y}} - \min_{\underline{x} \in X, \underline{y} \in Y} \| \underline{y} - \underline{x} \|_{\hat{Q}}^2 \tag{6.7.49}$$

and:

$$\begin{aligned} \|\hat{\underline{x}}_{min}\|_{\hat{Q}}^2 &\leq \|\hat{\underline{x}}\|_{\hat{Q}}^2 \\ \|\hat{\underline{y}}_{min}\|_{\hat{Q}}^2 &\leq \|\hat{\underline{y}}\|_{\hat{Q}}^2 \end{aligned} \qquad \text{where } \hat{\underline{x}}, \hat{\underline{y}} \text{ are any other solution} \tag{6.7.50}$$

By the "best least squares solution" in part (a), we mean that if there is no single vector $\hat{\underline{x}} = \hat{\underline{y}}$ which will satisfy all the signal and output constraints simultaneously, then equation (6.7.46) will not have any solution at all. Thus we choose $\hat{\underline{\rho}}_{x_0}$ to minimize the equation error:

$$\hat{\underline{\rho}}_{x_0} - \min_{\underline{\rho}_x} \left\| (I - Q_x Q_y Q_x)\underline{\rho}_x - (\bar{\underline{\rho}}_x - Q_x \bar{\underline{\rho}}_y) \right\|_{\bar{Q}}^2 . \tag{6.7.51}$$

Any solution to this minimization may be used. A different, but equivalent closed-form solution would be:

a) Find a best least square solution to:

$$\left(I - Q_y Q_x Q_y \right) \hat{\underline{\rho}}_{y_0} = \bar{\underline{\rho}}_y - Q_y \bar{\underline{\rho}}_x \tag{6.7.52}$$

b)
$$\begin{aligned} \hat{\underline{\rho}}_{x_1} &= -Q_x \hat{\underline{\rho}}_{y_0} + \bar{\underline{\rho}}_x \\ \hat{\underline{\rho}}_{y_1} &= -Q_y \hat{\underline{\rho}}_{x_1} + \bar{\underline{\rho}}_y \end{aligned} \tag{6.7.53}$$

c)
$$\begin{aligned} \hat{\underline{x}}_{min} &= \hat{\underline{\rho}}_{x_1} + \hat{\underline{\rho}}_{y_0} \\ \hat{\underline{y}}_{min} &= \hat{\underline{\rho}}_{x_1} + \hat{\underline{\rho}}_{y_1} \end{aligned} \tag{6.7.54}$$

This set of estimates $\hat{\underline{x}}$, $\hat{\underline{y}}$ is identical to that generated by the previous algorithm.

Appendix J also analyzes the convergence properties of the dual iterative algorithm. In the general case, $N_x^{\perp} \cap N_y^{\perp} \neq \{\underline{0}\}$, it shows that if the number of non-zero eigenvalues of $Q_x Q_y Q_x$ and $Q_y Q_x Q_y$ is finite, (a condition which will occur if $\min(p,q) < \infty$), then the multiplier estimates $\hat{\underline{\rho}}_{x_k}$, $\hat{\underline{\rho}}_{y_k}$ are the sum of a constant, plus a linear ramp $k(\hat{\underline{y}}_{min} - \hat{\underline{x}}_{min})$ which grows on each iteration, plus another term which decays at a geometric rate given by the largest eigenvalue λ_{max} of $Q_x Q_y Q_x$ which is not one. The multipliers will thus converge in the usual sense only if the solution to the Fisher problem satisfies $\hat{\underline{x}} = \hat{\underline{y}}$. Otherwise, the multiplier estimates grow by a fixed amount on each iteration. Fortunately, the linear ramp term cancels out when we compute the signal and output estimates. Thus if $\hat{x}_{min}$, $\hat{y}_{min}$ is the minimal energy solution to (6.7.34), then:

$$\begin{aligned} \| (\hat{y}_{k+1} - \hat{x}_{k+1}) - (\hat{y}_{min} - \hat{x}_{min}) \|_Q^2 &\leq \lambda_{max} \| (\hat{y}_k - \hat{x}_{k+1}) - (\hat{y}_{min} - \hat{x}_{min}) \|_Q^2 \\ &\leq \lambda_{max}^2 \| (\hat{y}_k - \hat{x}_k) - (\hat{y}_{min} - \hat{x}_{min}) \|_Q^2 \end{aligned} \qquad (6.7.55)$$

7.9. Dual Algorithm - Eigenstructure, Noise Sensitivity, Geometric Angle

Because Q_x and Q_y are projection matrices just like P_x and P_y, the eigenstructures of $Q_x Q_y Q_x$ and $Q_y Q_x Q_y$ will have exactly the same properties as $P_x P_y P_x$ and $P_y P_x P_y$. In particular, all eigenvalues are real, non-negative and less than or equal to one, and the eigenvectors form a complete orthonormal basis. The only difference is that Q_x and Q_y project onto the orthogonal complements of the null spaces $N_x^\perp$ and $N_y^\perp$, unlike P_x and P_y which project onto N_x and N_y. Every vector in $N_x^\perp \cap N_y^\perp$ will be an eigenvector of $Q_x Q_y Q_x$ with eigenvalue of one; all the other eigenvectors will have eigenvalue strictly less than 1, and are orthogonal to $N_x^\perp \cap N_y^\perp$. The number of non-zero eigenvalues is no larger than $\min(p,q)$, and there is a one-to-one mapping between the eigenvectors ξ_i and η_i of $Q_x Q_y Q_x$ and $Q_y Q_x Q_y$ corresponding to the same non-zero eigenvalue. The number of non-zero eigenvalues strictly less than 1 is smaller than $\min(p,q,N-p,N-q)$.

7.10. Dual Algorithm - Noise Sensitivity

The noise sensitivity analysis of the dual algorithm looks very similar to that of the primal algorithm, and so we will not include the details. Once again, the actual multiplier estimates are corrupted by computation noise which linearly accumulates in the space $N_x^\perp \cap N_y^\perp$, plus additional computation noise orthogonal to $N_x^\perp \cap N_y^\perp$ which is reduced by further iterations at a rate λ_{max}^2. The major difference is that in computing the signal and output estimates from the multipliers, this linear error ramp in $N_x^\perp \cap N_y^\perp$ almost exactly cancels out, leaving only a bounded error term orthogonal to $N_x^\perp \cap N_y^\perp$. Thus the noise sensitivity of the dual algorithm's signal and output estimates is finite, though still proportional to $\frac{1}{1-\lambda_{max}^2}$. (The noise sensitivity of the multiplier estimates, however, is infinite if $N_x^\perp \cap N_y^\perp \neq \{\underline{0}\}$.)

7.11. Link Between Primal and Dual Algorithms

As equation (6.7.33) strongly suggests, there is a strong link between the dual and primal problems. Set $\acute{Q}=R$; then the primal algorithm's projection matrices P_x, P_y are related to the dual algorithm's projection matrices Q_x, Q_y by:

$$\begin{aligned} P_x &= I - Q_x \\ P_y &= I - Q_y \end{aligned} \tag{6.7.56}$$

Appendix J proves the following property:

The non-zero eigenvalues of $Q_x Q_y Q_x$ and $Q_y Q_x Q_y$ which are strictly less than 1, are identical to the non-zero eigenvalues of $P_y P_x P_y$ and $P_x P_y P_x$ which are strictly less than 1. Furthermore, if $\{\underline{\psi}_i\}$ are an orthonormal set of eigenvectors of $P_x P_y P_x$ with non-zero eigenvalues less than 1, then the vectors:

$$\begin{aligned} \underline{\phi}_i &= \frac{1}{\sqrt{\lambda_i}} P_y P_x \underline{\psi}_i \\ \underline{\eta}_i &= \frac{1}{\sqrt{1-\lambda_i}} Q_y P_x \underline{\psi}_i \\ \underline{\xi}_i &= \frac{1}{\sqrt{1-\lambda_i}} Q_x P_y \underline{\phi}_i \end{aligned} \tag{6.7.57}$$

are orthonormal eigenvectors of the other matrices with exactly the same eigenvalue:

$$P_y P_x P_y \underline{\phi}_i = \lambda_i \underline{\phi}_i$$

$$Q_y Q_x Q_y \eta_i = \lambda_i \eta_i \tag{6.7.58}$$

$$Q_x Q_y Q_x \xi_i = \lambda_i \xi_i$$

Since $\psi_i \in N_x$, $\phi_i \in N_y$, $\eta_i \in N_y^{\perp}$, $\xi_i \in N_x^{\perp}$, the total number of non-zero eigenvalues strictly less than 1 must be smaller than the dimensions of all these spaces, $\min(p, q, N-p, N-q)$.

The only difference between the primal and dual algorithms, therefore, is in the arrangement of eigenvectors with eigenvalues of exactly zero or one. Since the convergence rate is determined by the largest eigenvalue λ_{max} less than one, we would expect both the primal and dual algorithms to converge at exacfly the same rate.

The case $N_x \cap N_y = \{0\}$ and $N_x^{\perp} \cap N_y^{\perp} = \{0\}$ is particularly interesting. This situation occurs only if the number of constraints equals the number of points, $p + q = N$, and the matrix $\begin{pmatrix} G_x \\ G_y \end{pmatrix}$ is invertible. Thus the Fisher solution will not only be unique, but will also satisfy $\hat{x} = \hat{y}$. Both the primal and dual problems will be well behaved. Furthermore, the reasoning above guarantees that:

$$\cos\theta = \lambda_{max}^{\frac{1}{2}} = \|P_y P_x\|_R = \|P_x P_y\|_R = \|Q_y Q_x\|_R = \|Q_x Q_y\|_R < 1 \tag{6.7.59}$$

In this case, the constraint equations could actually be solved directly:

$$\begin{pmatrix} G_x \\ G_y \end{pmatrix} \hat{x} = \begin{pmatrix} \mathbf{y}_x \\ \mathbf{y}_y \end{pmatrix} \tag{6.7.60}$$

7.12. Estimating the Convergence Rate

The convergence rate λ_{max} of these algorithms can be estimated by a variety of methods. For example, the derivation in Appendix J suggests:

$$\lambda_{max} \geq \frac{\|\hat{y}_{k+1} - \hat{y}_k\|_R^2}{\|\hat{x}_{k+1} - \hat{x}_k\|_R^2} \tag{6.7.61}$$

and:

$$\lambda_{max}^{2} \geq \frac{\| \hat{y}_{k+1} - \hat{y}_{k} \|_{R}^{2}}{\| \hat{y}_{k} - \hat{y}_{k-1} \|_{R}^{2}} \tag{6.7.62}$$

An *a priori* estimate of the convergence rate can also be found from the primal closed form solution:

$$\begin{aligned} \| \hat{y} \| &= \left\| (I-P_y P_x P_y)^{-1} (\bar{y} + P_y(\bar{x} + P_x \bar{y})) \right\|_R \\ &\leq \frac{1}{1-\lambda_{max}} \| \bar{y} + P_y(\bar{x} + P_x \bar{y}) \|_R \end{aligned} \tag{6.7.63}$$

or:

$$\lambda_{max} \geq 1 - \frac{\| \bar{y} + P_y(\bar{x} + P_x \bar{y}) \|_R}{\| \hat{y} \|_R} \tag{6.7.64}$$

This last estimate is usually very conservative (λ_{max} is usually much closer to one than this would suggest) but it will be very useful in getting a rough estimate of how "well-posed" various problems are.

7.13. Acceleration Techniques, Conjugate Gradient Algorithms

Exactly the same acceleration techniques which solve the Bayesian algorithms in chapter 5 can be used to accelerate these Fisher algorithms. The only change is that with B=I and $\alpha=0$, then H=I and V=R, so that the calculation required simplifies. For example, the line search acceleration formula for PARTAN will take the form:

$$\hat{\alpha}_k = \frac{< \Delta_{k+1}, \Delta_{k+1} - \Delta_k >_R}{\| \Delta_{k+1} - \Delta_k \|_R^2} \tag{6.7.65}$$

The interested reader can easily write down the appropriate PARTAN or conjugate gradient algorithms for solving either the primal or the dual problems. Note that if the global optimizing solution does not satisfy $\hat{x} = \hat{y}$, then exact dual closed form solutions do not exists, and the dual conjugate gradient methods will diverge. Conjugate gradient will always solve the primal problem, however, since the primal problem always has a solution, even when the matrices are non-invertible.

8. Linear Inequality Constraints

The case when the constraint sets are defined by linear inequalities $G_x x \leq \gamma_x$ and $G_y y \leq \gamma_y$ can also be analyzed with relative ease. If the constraint sets are non-empty, then the Fisher XYMAP algorithm is guaranteed to have a finite solution (see Künzi and Krelle [8] or Goldstein [9] chapter 3). (We would conjecture that MCEM, XMAP and YMAP will also always have a solution.) Lagrange multipliers can be used together with the Kuhn-Tucker conditions to state necessary conditions for each required maximization. Because the constraint sets are convex, closed and non-empty, the set of solutions will be convex and closed, and in the case of XYMAP, any two distinct global maxima (x_1,y_1) and (x_2,y_2) will satisfy:

$$y_1 - x_1 = y_2 - x_2 \tag{6.8.1}$$

All four iterative algorithms will converge to a finite global optimum solution if and only if such a solution exists, and the distance from the estimates $\hat{x}_k$, $\hat{y}_k$ to any such solution is decreasing. Various quadratic programming algorithms, such as Wolfe's, Beale's or Dantzig's algorithms [10, 8] are available for solving problems such as this using a finite amount of computation. The chief drawback of these algorithms is that they are based on the simplex method of linear programming, and thus adjust only one variable at a time. Total computation time could therefore be worse than our iterative algorithm, which though it never terminates, improves all the variable estimates simultaneously and can thus potentially give reasonably good answers relatively quickly.

9. Summary

In this chapter we have examined the structure of our four iterative signal reconstruction algorithms when we assume no *a priori* knowledge of the signal covariance. We first showed that as our *a priori* signal covariance estimate $Q \to \infty$, then our four Bayesian algorithms asymptotically locate the minimal signal energy solution to the corresponding "Fisher" estimation problem. These four "Fisher" estimation algorithms can be interpreted as trying to find a pair of signal and output estimates which come as close to each other as possible. The iterative algorithm which solves these problems uses projection or expectation operators to estimate the signal from the output, and then the output from the signal, iterating back and forth until the estimates converge. No filtering step is used because without *a priori* knowledge of the difference in the statistical behavior of the signal and noise, filtering is not possible. When the constraint sets are convex, we showed that the iterative algorithms will converge to a global minimizing solution if and only if such a solution exists.

When the constraints are defined by linear equalities, then all four algorithms are identical. The primal algorithm is the most straightforward; the iteration always converges at a geometric rate to the nearest solution, the closed-form solution can always be calculated, and if the solution is unique then the noise sensitivity will be bounded (though it may be quite large.) The dual algorithm has the same form as the primal algorithm, but it uses projection operators which are orthogonal to the primal algorithm's projection operators. The effective dimension of the dual algorithm is $\min(p,q)$, which is usually different than the the effective dimension $\min(N-p,N-q)$ of the primal algorithm. The dual signal and output estimates will always converge at a geometric rate to the minimal energy solution regardless of how we initialize the algorithm, and the noise sensitivity will be finite. However, if the solution does not satisfy $\hat{x}=\hat{y}$, then the multiplier estimates actually grow by a fixed amount on each iteration, and their noise sensitivity will be infinite. If the solution does not satisfy $\hat{x}=\hat{y}$, then the dual closed-form solution will not have an exact solution, although signal and output estimates can still be constructed from any best least squares solution to the formulas. PARTAN and conjugate gradient algorithms could also be applied to solve either the primal or dual problems in a finite number of steps. These algorithms use one pass of our original algorithm followed by two line search acceleration steps. The primal and dual algorithms are actually very closely linked; the eigenvalues which are non-zero and strictly less than one are identical in the two problems, and the convergence rates will be the same. The difference is that the computational effort of one may be very much smaller than the other. We ended by briefly treating linear inequality constraints, mentioning that the Fisher XYMAP algorithm will always have a solution in this case, and pointing out the availability of quadratic programming algorithms to help solve the problem.

References

1. Lawrence R. Rabiner and Bernard Gold, *Theory and Applications of Digital Signal Processing*, Prentice Hall Inc., Englewood Cliffs, N.J. (1975).
2. Dante Youla, "Generalized Image Restoration by the Method of Alternating Orthogonal Projections," *IEEE Trans. Circuits. Syst.* CAS-25(9), pp.694-702 (Sept 1978).
3. Edoardo Mosca, "On a Class of Ill-Posed Estimation Problems and a Related Gradient Iteration," *IEEE Trans. Auto. Control* AC-17(4), pp.459-465 (Aug 1972).
4. Anil K. Jain and Surendra Ranganath, "Extrapolation Algorithms for Discrete Signals with Application in Spectral Estimation," *IEEE Trans. Acoust., Speech, Sig. Proc.* ASSP-29(4), pp.830-845 (Aug 1981).
5. James A. Cadzow, *Inversion of Signal Operations*, Proc. 1979 IEEE Int. Conf. Acoust., Speech, Sig. Proc., Wash. D.C. (1979).
6. Victor T. Tom, Thomas F. Quatieri, Monson H. Hayes, and James H. McClellan, "Convergence of Iterative Nonexpansive Signal Reconstruction Algorithms," *IEEE Trans. Acoust., Speech, Sig. Proc.* ASSP-29(5), pp.1052-1058 (Oct 1981).
7. Ronald W. Schafer, Russell M. Mersereau, and Mark A. Richards, "Constrained Iterative Restoration Algorithms," *Proc. IEEE* 69(4), pp.432-450 (April 1981).
8. Hans Künzi and Wilhelm Krelle, *Nonlinear Programming*, Blaisdell Publishing, Waltham, Mass. (1966).
9. A.A. Goldstein, *Constructive Real Analysis*, Harper and Row, New York (1967).
10. John C. G. Boot, *Quadratic Programming - Algorithms, Anomalies, Applications*, North-Holland Publishing Company - Rand McNally & Company, Chicago (1964).

Chapter 7

Applications of Optimal Signal Reconstruction

Part III - Time and Frequency Constraints

1. Introduction

In the last two chapters we have applied our MCEM and MAP estimation methods to the problem of optimally reconstructing a Gaussian signal corrupted by Gaussian noise, when we are only given constraints which the signal and output are known to satisfy. In the next 2 chapters we will apply these algorithms to a variety of specific applications. This chapter deals with several applications in which a signal must be reconstructed given noisy information about its behavior in both the time and frequency domains. We first consider a general model for this class of problems, and show that the algorithms take a particularly simple form if the signal covariance matrix is diagonal and the noise is white. All the algorithms start with a signal estimate, take its Discrete Fourier transform, then find the "nearest" output estimate whose frequency samples obey the known frequency constraints. The output estimate is then inverse Discrete Fourier transformed, filtered, and the "nearest" signal is found which satisfies the known time domain constraints.

When the constraint sets are linear, all four iterative algorithms give the same estimates, and we can develop both primal and dual iterative algorithms, closed-form solutions, and conjugate gradient algorithms. The best known application of this type is extrapolating a band-limited signal. We show that our analysis in the last two chapters not only covers most of the known properties of this algorithm, as presented by Papoulis [1] and Jain and Ranganath [2] , but also suggests some new ones. Another linear equality constraint application is reconstructing a finite length signal from knowledge of its phase modulo π, a problem originally considered by Hayes [3] . Several examples are given illustrating the various Bayesian and Fisher algorithms. We also point out that this problem is inherently ill-conditioned as the signal length grows large.

Next we consider some applications involving linear inequality constraints. Reconstruction of a finite signal from knowledge of the phase modulo 2π is considered, and we show that although the MCEM algorithm is somewhat more complicated than the others, it appears to generate the best signal reconstruction. We also compare our algorithms with that of Hayes, Lim and Oppenheim [4,5,3] who originally proposed this problem, derived an iterative algorithm to solve it, and proved conditions for the uniqueness of the solution. A second linear inequality constraint application we consider is designing multidimensional Finite Impulse Response Filters to meet arbitrary time and frequency domain constraints.

Finally, we consider problems in which the constraint sets are non-convex. Our success with these, however, has not been very good. We consider the problem of reconstructing a multi-dimensional signal from the magnitude of its spectrum. Our XYMAP algorithm for this problem is identical to that proposed by Fienup, [6] Gerchberg and Saxton, [7] and Hayes, Lim and Oppenheim. [4,3] Unfortunately, although Hayes has proven that reconstruction from the magnitude is theoretically possible if the finite sequence is irreducible, in practice the problem appears to have a very large number of local minima and critical points. Convergence to the true global minimizer is therefore virtually impossible unless the initial starting point is very close to the true solution.

Section A - Reconstruction from Time and Frequency Constraints

2. The Model

The most interesting applications of our optimal signal reconstruction algorithms, of course, are those in which the minimizations or conditional expectations required at each step are easy to compute. The simplest case in which this will happen is when the covariance matrices are diagonal and the constraints on X and Y are related by an orthogonal transform. The most appealing and useful situation of this type is when we must reconstruct a Gaussian signal $\underline{x}$ and noisy Gaussian output $\underline{y}$, given only a set of constraints on the time domain behavior of $\underline{x}$, and a set of constraints on the frequency domain behavior of $\underline{y}$. In orcer to state this model concisely, let us define the $N \times N$ Discrete Fourier Transform (DFT) matrix $\mathbf{W}_N$ by:

$$\left[\mathbf{W}_N\right]_{k,l} = \frac{1}{\sqrt{N}} \exp\left[-j2\pi \frac{kl}{N} \right] \tag{7.2.1}$$

It is well known [8,9] that the eigenvalues of $\mathbf{W}_N$ are ± 1, $\pm j$ (to prove this, note that $\mathbf{W}_N^4 = \mathbf{I}$.) Its determinant is therefore $|\mathbf{W}_N| = 1$, and its inverse is $\mathbf{W}_N^{-1} = \mathbf{W}_N^H = \mathbf{W}_N^3$. Let the vector $\underline{x}$ represent an N point signal $\underline{x} = \left(x(0) \cdots x(N-1) \right)^T$, and let $X(\omega_l)$ be its Discrete Fourier Transform (DFT):

$$X(\omega_l) = \sum_{n=0}^{N-1} x(n)\, e^{-j\omega_l n} \qquad \text{for } \omega_l = \frac{2\pi l}{N} \tag{7.2.2}$$

It is easy to see that:

$$\underline{x}_f = \mathbf{W}_N \underline{x} = \frac{1}{\sqrt{N}} \begin{pmatrix} X(\omega_0) \\ \vdots \\ X(\omega_{N-1}) \end{pmatrix} \qquad \text{where: } \omega_l = \frac{2\pi l}{N} \tag{7.2.3}$$

is the DFT of $\underline{x}$. Note that the signal $\underline{x}$ can be recovered from the DFT vector $\underline{x}_f$ by:

$$\underline{x} = \mathbf{W}_N^H \underline{x}_f \tag{7.2.4}$$

Because $\mathbf{W}_N^H \mathbf{W}_N = \mathbf{W}_N \mathbf{W}_N^H = \mathbf{I}$, Parseval's theorem also follows easily:

$$\begin{aligned}\sum_{n=0}^{N-1} |x(n)|^2 &= \underline{x}^H \underline{x} \\ &= \underline{x}_f^H \mathbf{W}_N \mathbf{W}_N^H \underline{x}_f \\ &= \underline{x}_f^H \underline{x}_f \\ &= \frac{1}{N} \sum_{l=0}^{N-1} |X(\omega_l)|^2\end{aligned} \tag{7.2.5}$$

The model we will consider is the simplest example in which noisy time and frequency domain constraints are given:

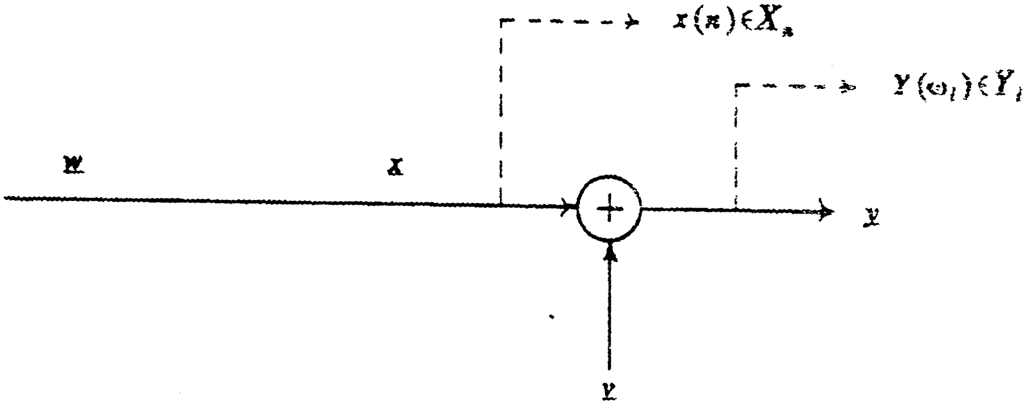


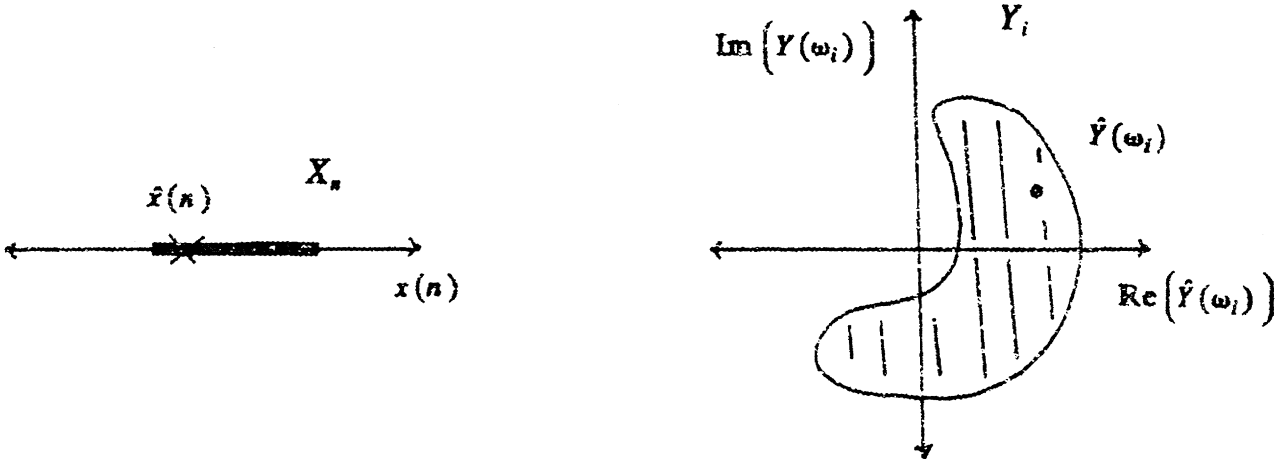
Figure 7.2.1 - Time/Frequency Constraint Model

$$\begin{array}{lll} \text{Model:} & \underline{x} = \underline{w} & \text{where: } p(\underline{w}) = N(\underline{0},Q) \\ & \underline{y} = \underline{x} + \underline{v} & \text{where: } p(\underline{v}) = N(\underline{0},R) \end{array} \qquad (7.2.6)$$

$$\text{where: } Q = \text{diag}(\,q(0) \cdots q(N-1)\,)$$
$$R = \text{diag}(\,r \cdots r\,)$$

$$\begin{array}{lll} \text{Observations:} & x(n) \in X_n & \text{where: } X = X_0 \times X_1 \times \cdots \times X_{N-1} \\ & Y(\omega_i) \in Y_i & \text{where: } Y = Y_0 \times Y_1 \times \cdots \times Y_{N-1} \end{array}$$

The signal $x(n)$ is a white Gaussian random sequence with zero mean and time varying variance $q(n)$. All signal samples are thus assumed to be stochastically independent of each other. The output $\underline{y}$ is formed by adding white Gaussian noise with variance r to $\underline{x}$. The observation information specifies only that each sample $x(n)$ is known to lie in some range X_n, and each sample $Y(\omega_i)$ of the Discrete Fourier Transform of the output is known to lie in a set Y_i. These sets X_n and Y_i are assumed to be independent of the value of any other samples of $\underline{x}$ or $\underline{y}$.


$\hat{x}(n)$
X_n
$x(n)$
$\text{Im}\left(Y(\omega_i)\right)$
Y_i
$\hat{Y}(\omega_i)$
$\text{Re}\left(\hat{Y}(\omega_i)\right)$

Signal Constraint

Output Constraint

Figure 7.2.2 - Typical Constraint Sets

Because the signals are assumed to be real, their DFT's must be conjugate symmetric, and so we will assume that $Y_{N-i} = Y_i^* = \{Y^*(\omega_i) \mid Y(\omega_i) \in Y_i\}$. (If complex signals are considered, this restriction is unnecessary.) Given this data, our goal is to try to reconstruct the most likely value of the signal $\hat{x}$ and the output $\hat{y}$.

3. Separability of the Conditional Densities

The most important feature of this model, for our purposes, is that the conditional densities $p_{X|Y}(x|\hat{y}) = N_X(H\hat{y},V)$ and $p_{Y|X}(y|\hat{x}) = N_Y(\hat{x},R)$ are separable. This fact will allow us to estimate each time domain sample $x(n)$ independently of all the other samples of x, and each frequency domain sample $Y(\omega_l)$ independently of all the other samples of $Y(\omega)$.

Note first that the covariance matrices R and Q are diagonal, and so the covariance matrix V and filter H are also diagonal:

$$V = (Q^{-1}+R^{-1})^{-1} = \begin{pmatrix} v(0) & 0 \\ & \cdot \\ 0 & v(N-1) \end{pmatrix} \qquad \text{where } v(n) = \frac{q(n)r}{q(n)+r}$$

$$H = \begin{pmatrix} h(0) & 0 \\ & \cdot \\ 0 & h(N-1) \end{pmatrix} \qquad \text{where } h(n) = \frac{q(n)}{q(n)+r} \tag{7.3.1}$$

Furthermore, because the signal constraint set is the cartesian product of the sets $X_0, \cdots, X_{N-1}$, the conditional probability of each sample $x(n)$ will depend only on the corresponding output sample $\hat{y}(n)$, and will be independent of all other samples of $\hat{y}$. Thus the conditional probability density $p_{X|Y}(x|\hat{y}) = N_X(H\hat{y},V)$ can be written as a product of truncated Gaussian densities over each sample value $x(n)$ and centered at $h(n)\hat{y}(n)$:

$$p_{X|Y}(x|\hat{y}) = \prod_{i=0}^{N-1} p_{X_n}(x(n)|\hat{y}(n)) \tag{7.3.2}$$

where:

$$p_{X_n}(x(n) \mid \hat{y}(n)) = N_{X_n}\Big(h(n)\hat{y}(n), v(n)\Big)$$

$$= \begin{cases} K_{x_n} \exp\left[-\frac{1}{2v(n)} \Big(x(n)-h(n)\hat{y}(n)\Big)^2 \right] & \text{for } x(n) \epsilon X_n \\ 0 & \text{else} \end{cases}$$

We can decompose $p_{Y|X}(\underline{y} \mid \hat{\underline{x}})$ in a similar way, except that we must first transform to the frequency domain. Let us define the DFT variables $\underline{x}_f$, $\underline{y}_f$ by:

$$\underline{x}_f = W_N \underline{x} = \frac{1}{\sqrt{N}} \begin{pmatrix} X(\omega_0) \\ \vdots \\ X(\omega_{N-1}) \end{pmatrix} \qquad (7.3.3)$$

$$\underline{y}_f = W_N \underline{y} = \frac{1}{\sqrt{N}} \begin{pmatrix} Y(\omega_0) \\ \vdots \\ Y(\omega_{N-1}) \end{pmatrix}$$

Then:

$$p(\underline{y}_f \mid \underline{x}_f) = p(\underline{y} \mid \underline{x}) \left| \frac{\partial \underline{y}}{\partial \underline{y}_f} \right| = p(\underline{y} \mid \underline{x}) \, |W_N^{-1}| = p(\underline{y} \mid \underline{x}) \qquad (7.3.4)$$

or because R is a multiple r of an identity matrix:

$$p(\underline{y}_f \mid \underline{x}_f) = N_Y(\underline{x}_f, \mathbf{W}_N r \mathbf{W}_N^H) = N_Y(\underline{x}_f, r\mathbf{I}) \qquad (7.3.5)$$

Now because the covariance R is diagonal, and because the constraint set Y decomposes into an independent set of constraints on the components of $\underline{y}_f$, the probability density $p(\underline{y}_f \mid \underline{x}_f)$ can be decomposed into a product of densities over the individual components of $\underline{y}_f$:

$$p_{Y|X}(\underline{y}_f \mid \underline{x}_f) = \prod_{l=0}^{N-1} p_{Y_l}\left(\frac{1}{\sqrt{N}} Y(\omega_l) \,\middle|\, \frac{1}{\sqrt{N}} \hat{X}(\omega_l) \right) \qquad (7.3.6)$$

where:

$$p_{Y_l}\left(\frac{1}{\sqrt{N}} Y(\omega_l) \,\middle|\, \frac{1}{\sqrt{N}} \hat{X}(\omega_l) \right) = N_{Y_l}\left(\frac{1}{\sqrt{N}} \hat{X}(\omega_l), r \right)$$

$$\begin{cases} K_{y_i}\exp\left[-\frac{1}{2r}\frac{1}{N}\left|Y(\omega_i)-\hat{X}(\omega_i)\right|^2\right] & \text{for } Y(\omega_i)\in Y_i \\ 0 & \text{else} \end{cases}$$

Each output frequency component $Y(\omega_i)$ has a truncated complex Gaussian distribution centered at $\hat{X}(\omega_i)$, and is thus stochastically independent of any other samples of the signal or output spectrum.

4. MCEM and MAP Estimation Algorithms

The four Bayesian MCEM and MAP estimation algorithms which we developed for this problem in chapter 5 iteratively calculate signal and output estimates as follows:

	Signal Estimate	Output Estimate
MCEM:	$\hat{x}_{k+1} = E_X[x \mid H\hat{y}_k]$	$\hat{y}_{k+1} = E_Y[y \mid \hat{x}_{k+1}]$
XMAP:	$\hat{x}_{k+1} - \min_{x\in X} \lVert x - H\hat{y}_k \rVert_V^2$	$\hat{y}_{k+1} = E_Y[y \mid \hat{x}_{k+1}]$
YMAP:	$\hat{x}_{k+1} = E_X[x \mid H\hat{y}_k]$	$\hat{y}_{k+1} - \min_{y\in Y} \lVert y - \hat{x}_{k+1} \rVert_R^2$
XYMAP:	$\hat{x}_{k+1} - \min_{x\in X} \lVert x - H\hat{y}_k \rVert_V^2$	$\hat{y}_{k+1} - \min_{y\in Y} \lVert y - \hat{x}_{k+1} \rVert_R^2$

where the conditional expectations are with respect to the conditional densities $N_X(H\hat{y}_k, V)$ and $N_Y(\hat{x}_{k+1}, R)$. Because of the separability of the conditional probability densities and the constraint sets, these formulas can be dramatically simplified. In the signal estimation step, the expectation of each component $x(n)$ of x can be computed independently using only the corresponding output component $\hat{y}_k(n)$. The minimum of $\lVert x - H\hat{y}_k \rVert^2$ can also be computed by expanding the norm as:

$$\lVert x - H\hat{y}_k \rVert_R^2 = \sum_{n=0}^{N-1} \frac{1}{v(n)} \left| x(n) - h(n)\hat{y}_k(n) \right|^2 \qquad (7.4.1)$$

and then minimizing with respect to each component $x(n)$. The output can be estimated in a similar way if we first transform to the variables $\underline{x}_f$, $\underline{y}_f$ in (7.3.3). The conditional expectation of $\underline{y}$ is just the inverse DFT of the expectation of $\underline{y}_f$, which in turn can be computed over each component $\frac{1}{\sqrt{N}} Y(\omega_i)$ independently. Similarly, the norm $||\underline{y} - \hat{\underline{x}}_{k+1}||^2$ can be computed in the frequency domain by Parseval's theorem:

$$||\underline{y} - \hat{\underline{x}}_{k+1}||_R^2 = \frac{1}{r} \sum_{n=0}^{N-1} \left(y(n) - \hat{x}_{k+1}(n) \right)^2 \tag{7.4.2}$$
$$= \frac{1}{r} \sum_{n=0}^{N-1} \frac{1}{N} \left| Y(\omega_i) - \hat{X}_{k+1}(\omega_i) \right|^2$$

Thus we need only minimize over each output frequency sample $Y(\omega_i)$ independently. Putting this all together, our four algorithms take the form shown in table 7.4.1. All four algorithms share the same structure. We start with an estimate of the output $\hat{Y}_k(\omega_i) \in Y_i$. Inverse Discrete Fourier Transform to calculate the time domain value $\hat{y}_k(n)$, filter by multiplying by $h(n) = \frac{q(n)}{q(n)+r}$, (a "time domain Weiner-Hopf filter"), then apply a projection or conditional expectation operator to choose the best estimate of each signal sample $\hat{x}_{k+1}(n)$. To reestimate the output, we Discrete Fourier Transform the signal, $\hat{X}_{k+1}(\omega_i)$, and apply a projection or conditional expectation operator to choose the best estimate of each output frequency sample $\hat{Y}_{k+1}(\omega_i)$ given $\hat{X}_{k+1}(\omega_i)$. This improved output estimate is used on the next pass to improve the next signal estimate. The algorithm thus alternates between filtering, forcing the time domain constraints, and then forcing the frequency domain constraints. Each iteration strictly decreases the cross-entropy, and the MAP methods also strictly increase the corresponding likelihood function on each pass. Each iteration therefore generates "better" signal and output estimates. If the estimates remain bounded (the MAP estimates always remain bounded) then they converge to a critical point or local minimum of the cross-entropy function (and a critical point or local maximum of the likelihood function.) If each constraint set X_n and Y_i is convex, then the entire constraint sets X and Y will be convex and geometric convergence is guaranteed to the unique global optimizing solution $\hat{x}$, $\hat{y}$:

$$\sum_{n=0}^{N-1} \frac{1}{r} \left| \hat{y}_k(n) - \hat{y}(n) \right|^2 \leq v_y \sum_{n=0}^{N-1} \frac{1}{v(n)} \left| \hat{x}_{k+1}(n) - \hat{x}(n) \right|^2 \tag{7.4.3}$$

$$\leq v_x v_y \sum_{l=0}^{N-1} \frac{1}{r} \left| \hat{y}_k(n) - \hat{y}(n) \right|^2$$

$$\text{where: } v_x = v_y = \max_n \left(\frac{q(n)}{q(n)+r} \right)$$

Geometric convergence will be guaranteed even if N is infinite.

Table 7.4.1 - Time/Frequency Constraint Iterative Algorithms

	Signal Estimate	Output Estimate
MCEM:	$\hat{x}_{k+1}(n) = \mathrm{E}_{X_n}\left[x(n)\,\middle\vert\,h(n)\hat{y}_k(n)\right]$	$\frac{1}{\sqrt{N}}\hat{Y}_{k+1}(\omega_l) = \mathrm{E}_{Y_l}\left[\frac{1}{\sqrt{N}}Y(\omega_l)\,\middle\vert\,\frac{1}{\sqrt{N}}\hat{X}_{k+1}(\omega_l)\right]$
XMAP:	$\hat{x}_{k+1}(n) \leftarrow \min_{x(n)\in X_n}\left\lvert x(n)-h(n)\hat{y}_k(n)\right\rvert^2$	$\frac{1}{\sqrt{N}}\hat{Y}_{k+1}(\omega_l) = \mathrm{E}_{Y_l}\left[\frac{1}{\sqrt{N}}Y(\omega_l)\,\middle\vert\,\frac{1}{\sqrt{N}}\hat{X}_{k+1}(\omega_i)\right]$
YMAP:	$\hat{x}_{k+1}(n) = \mathrm{E}_{X_n}\left[x(n)\,\middle\vert\,h(n)\hat{y}_k(n)\right]$	$\frac{1}{\sqrt{N}}\hat{Y}_{k+1}(\omega_l) \leftarrow \min_{Y(\omega_l)\in Y_l}\frac{1}{N}\left\lvert Y(\omega_l)-\hat{X}_{k+1}(\omega_l)\right\rvert^2$
XYMAP:	$\hat{x}_{k+1}(n) \leftarrow \min_{x(n)\in X_n}\left\lvert x(n)-h(n)\hat{y}_k(n)\right\rvert^2$	$\frac{1}{\sqrt{N}}\hat{Y}_{k+1}(\omega_i) \leftarrow \min_{Y(\omega_i)\in Y_i}\frac{1}{N}\left\lvert Y(\omega_i)-\hat{X}_{k+1}(\omega_i)\right\rvert^2$

where the conditional expectations are calculated with respect to the conditional densities:

$$p_{X_n}\left(x(n)\,\middle\vert\,\hat{y}_k(n)\right) = N_{X_n}\left(h(n)\hat{y}_k(n),\,v(n)\right)$$

$$p_{Y_l}\left(\frac{1}{\sqrt{N}}Y(\omega_l)\,\middle\vert\,\frac{1}{\sqrt{N}}\hat{X}_{k+1}(\omega_l)\right) = N_{Y_l}\left(\frac{1}{\sqrt{N}}\hat{X}_{k+1}(\omega_l)\,,\,r\right)$$

where: **Bayesian:** $h(n) = \frac{q(n)}{q(n)+r}$, $v(n) = \frac{q(n)r}{q(n)+r}$ **Fisher:** $h(n)=1$, $v(n)=r$

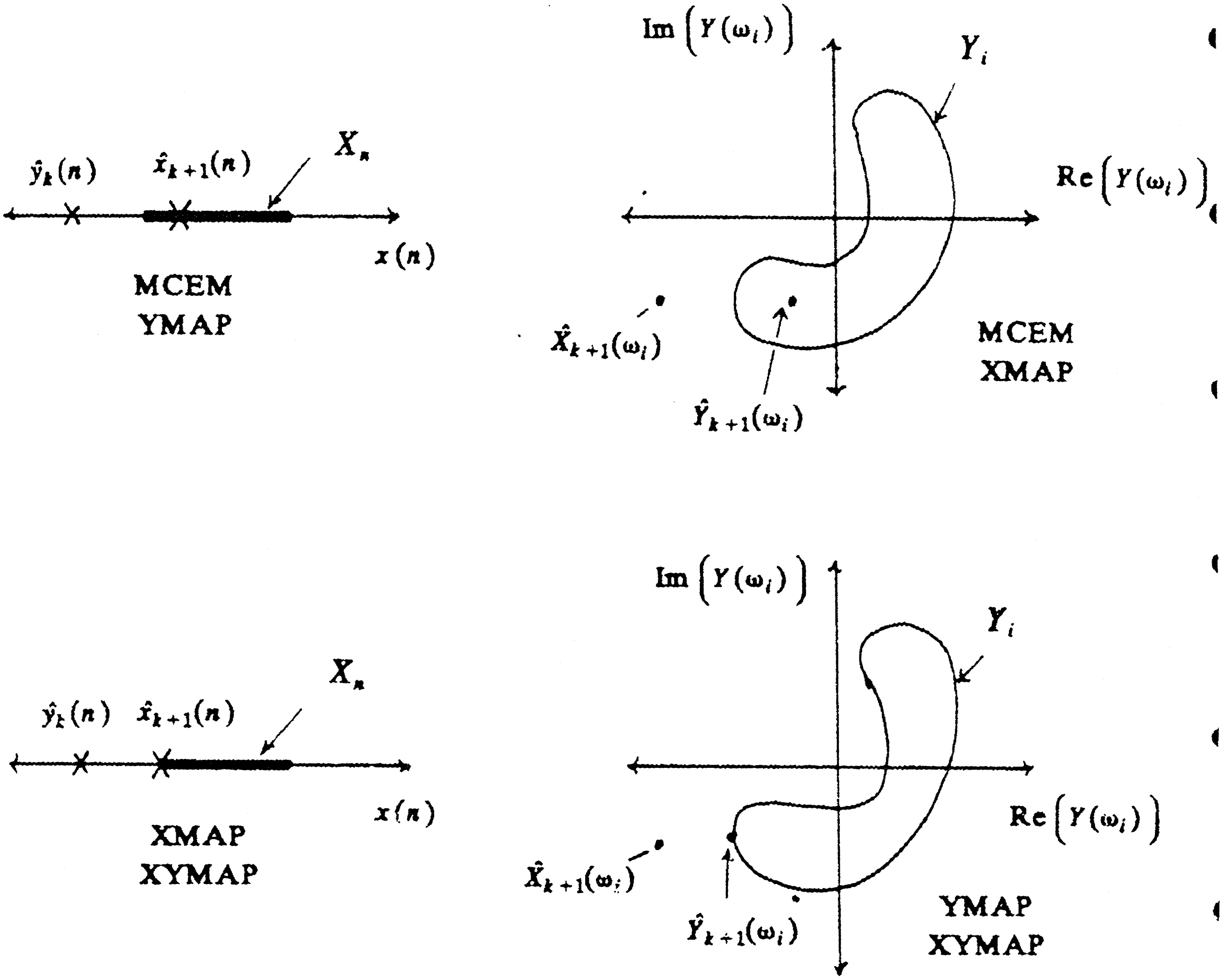

$\operatorname{Im}\left(Y(\omega_i)\right)$
Y_i
X_n
$\hat{y}_k(n)$
$\hat{x}_{k+1}(n)$
$\operatorname{Re}\left(Y(\omega_i)\right)$
$x(n)$
MCEM
YMAP
$\hat{X}_{k+1}(\omega_i)$
MCEM
XMAP
$\hat{Y}_{k+1}(\omega_i)$
$\operatorname{Im}\left(Y(\omega_i)\right)$
Y_i
X_n
$\hat{y}_k(n)$
$\hat{x}_{k+1}(n)$
$x(n)$
$\operatorname{Re}\left(Y(\omega_i)\right)$
XMAP
XYMAP
$\hat{X}_{k+1}(\omega_i)$
$\hat{Y}_{k+1}(\omega_i)$
YMAP
XYMAP

Figure 7.4.1 - Iterative Estimates

Note that although we have stated the algorithm using one-dimensional signals, the idea can be extended in an obvious way to multi-dimensional signals. Finally, note that although the conditional densities are separable, in general the original model density $p_{\underline{X},\underline{Y}}(\underline{x},\underline{y})$ will not be separable. Thus calculating the MMSE estimate will usually be extremely difficult, since it will require a 2N dimensional integration.

5. Fisher MCEM, MAP Algorithms

In many applications we really do not know the *a priori* covariance $q(n)$ of each signal sample. We will treat this Fisher estimation problem by assuming $q(n) = \frac{1}{\alpha} q_0(n)$, then letting $\alpha \to 0$, so that the *a priori* density $p(\underline{x})$ becomes asymptotically flat. As $\alpha \to 0$, our four Bayesian problems will asymptotically approach the minimum signal energy solution to the corresponding Fisher problem, provided of course that such a solution exists. In the limit as $\alpha = 0$, the resulting iterative algorithms have exactly the same form as in table 7.1, but with the filter $h(n)=1$ and the conditional signal covariance $v(n)=r$. The Fisher algorithm thus simply alternates between forcing the time domain constraints and forcing the frequency domain constraints. Each iteration strictly decreases the appropriate Fisher cross-entropy function, and XMAP and XYMAP strictly increase the appropriate likelihood functions. If the estimates remain bounded, they must converge to a critical point or local minimizer of the cross-entropy. If the constraint sets X_n, Y_i are convex, then the algorithms are always guaranteed to converge to a global minimizing solution if and only if such a solution exists. The convergence rate, however, may be sublinear, and the solution may be non-unique.

6. Alternative Time and Frequency Constraint Model

Our choice of setting time domain constraints on $\underline{x}$ and frequency domain constraints on $\underline{y}$ was clearly arbitrary, and could easily be reversed. Thus an alternative time-frequency constraint model would be:

$$\begin{array}{llll} \text{Model:} & \underline{x} = \underline{w} & \text{where: } p(\underline{w}) = N(\underline{0},Q) & \\ & \underline{y} = \underline{x} + \underline{v} & \text{where: } p(\underline{v}) = N(\underline{0},R) & (7.6.1) \\ & & \text{where: } [Q]_{i,j} = q(i-j) & \\ & & \qquad\quad [R]_{i,j} = r\delta_{i,j} & \\ \text{Observations:} & X(\omega_i) \in X_i & & \\ & y(n) \in Y_n & & \end{array}$$

We start with a stationary Gaussian signal with covariance Q. Note that Q is a cyclic matrix, so that $Q = W_N^H Q_f W_N$ where Q_f is diagonal with samples $Q(\omega_i)$ of the DFT of $q(n)$ on the diagonal. White Gaussian noise with variance r is added to form the output. Constraints are given for the value of each signal spectrum sample $X(\omega_i)$ and each output time sample $y(n)$. The algorithm for estimating $\underline{x}$ and $\underline{y}$ now looks exactly like our previous algorithm, but with the time and frequency domains reversed. To estimate the signal, we Fourier transform the output estimate, $\hat{Y}_k(\omega_i)$, filter by multiplying by $\dfrac{Q(\omega_i)}{Q(\omega_i)+r}$ (the Weiner-Hopf filter), then use a projection or conditional expectation to estimate $\hat{X}_{k+1}(\omega_i)$. Inverse transform, then use a projection or conditional expectation to estimate $\hat{y}_{k+1}(n)$ from $\hat{x}_{k+1}(n)$. Each iteration thus alternates between forcing the time constraints, filtering in the frequency domain, and forcing the frequency constraints. Each iteration decreases the cross-entropy and improves the estimates. The Fisher version of this algorithm is actually identical to the Fisher algorithm for the previous model, except with the roles of $\underline{x}$ and $\underline{y}$ reversed.

Section B - Linear Equality Constraints

7. Reconstruction from Linear Equality Constraint Sets

The first set of applications we will consider are models involving constraints defined solely by linear equations. We will only treat the model in section 2; modifications to treat the model in section 6 simply involve swapping the time and frequency domains. Because each signal sample is real, there are only two different types of linear variety signal constraint sets. X_n could be zero-dimensional, consisting of a single point, $X_n = \{x(n)\}$, so that this sample is known exactly. The other possibility is that X_n is one-dimensional, consisting of the entire real line, $X_n = \mathbf{R}$, so that the sample's value is completely unknown:

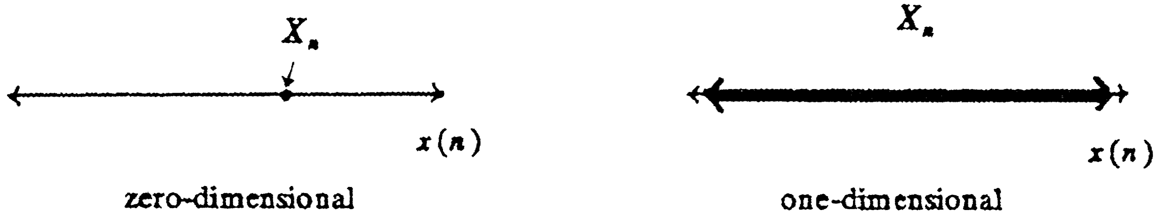


Figure 7.7.1 - Linear Variety Signal Constraint Sets

Because the output frequency samples are complex, there are three different types of linear variety output constraint sets. Y_i could be zero-dimensional, consisting of a single point, $Y_i = \{Y(\omega_i)\}$, so that the frequency sample is known exactly. Y_i could also be one-dimensional, consisting of a line $Y_i = \left\{ Y(\omega_i) \;\middle|\; \mathrm{Im}\left(Y(\omega_i)e^{-j\phi_i}\right) = \bar{Y}_i \right\}$ where ϕ_i is the angle of the line, and $\bar{Y}_i$ is its distance from the origin. Note that in this case, all samples on this line have values $(v_i + j\bar{Y}_i)e^{j\phi_i}$ for some $v_i \in \mathbf{R}$. Lastly, the constraint set could be two-dimensional, $Y_i = \mathbf{C}$, so that the frequency sample $Y(\omega_i)$ is completely unknown. (Beware that if the signals are real, we will require that $Y_{N-i} = Y_i^*$ since the transform must be conjugate symmetric.)

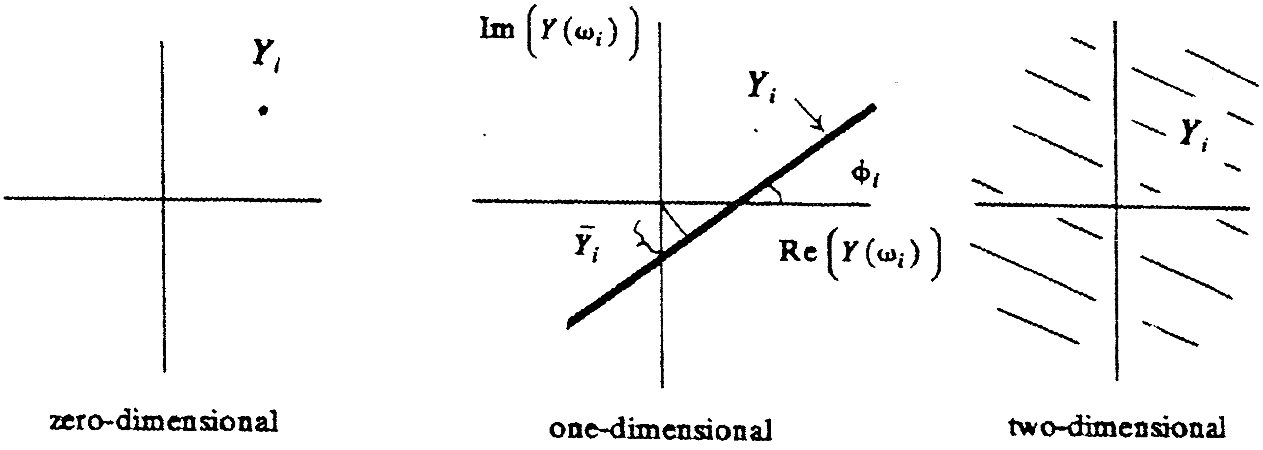


Figure 7.7.2 - Linear Variety Output Constraint Sets

Because all these constraint sets are defined by linear equalities, all our results about linear variety constraint sets can be directly applied. First of all, all four of our MCEM and MAP estimation approaches will give results identical to MMSE; we therefore focus exclusively on the XYMAP algorithm. To derive the iterative primal algorithm, we could write down the equations $G_x x = \chi_x$ and $G_y y = \chi_y$, defining all the constraint sets above, then plug into our general algorithm of chapter 5. An easier approach, however, is to solve the original minimization problem directly:

$$\begin{aligned} \hat{x}_{k+1}(n) &= \min_{x(n)\in X_n} \left| x(n) - \frac{q(n)}{q(n)+r}\hat{y}_k(n) \right|^2 \\ \hat{Y}_{k+1}(\omega_i) &= \min_{Y(\omega_i)\in Y_i} \left| Y(\omega_i) - \hat{X}_{k+1}(\omega_i) \right|^2 \end{aligned} \tag{7.7.1}$$

This gives the following Bayesian algorithm:

Primal Iterative Algorithm (Linear Equality Constraints)

Guess $\hat{Y}_0(\omega_i)\in Y_i$

For $k=0,1,\cdots$

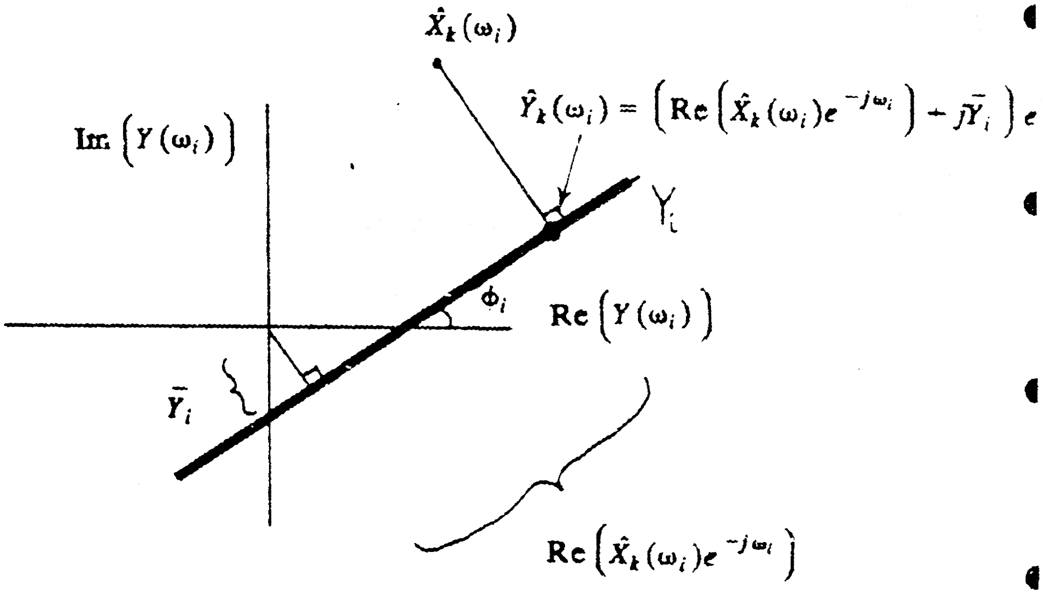
$$\hat{x}_{k+1}(n) = \begin{cases} x(n) & \text{if } X_n=\{x(n)\} \\ \dfrac{q(n)}{q(n)+r}\hat{y}_k(n) & \text{if } X_n=\mathbf{R} \end{cases} \tag{7.7.2}$$

$$\hat{Y}_{k+1}(\omega_i) = \begin{cases} Y(\omega_i) & \text{if } Y_i=\{Y(\omega_i)\} \\ \left(\mathrm{Re}\left(\hat{X}_{k+1}(\omega_i)e^{-j\phi_i}\right)+j\bar{Y}_i\right)e^{j\phi_i} & \text{if } Y_i \text{ is a line} \\ \hat{X}_{k+1}(\omega_i) & \text{if } Y_i=\mathbf{C} \end{cases}$$

The Fisher algorithm is identical, except that $q(n)=\infty$ so that the filtering step is omitted. We start with an estimate of the output $\hat{y}_0$. A good choice is usually to pick the minimal norm element in Y, since in the Fisher algorithm, this ensures convergence to the minimum norm solution if the answer should not be unique.

$$\hat{Y}_0(\omega_i) = \begin{cases} Y(\omega_i) & \text{if } Y_i=\{Y(\omega_i)\} \\ j\bar{Y}_i e^{j\phi_i} & \text{if } Y_i \text{ is a line} \\ 0 & \text{if } Y_i=\mathbf{C} \end{cases} \tag{7.7.3}$$

Given this output estimate, we inverse Fourier transform to get the time domain sequence $\hat{y}_k(n)$, and multiply by the "time domain Wiener-Hopf filter" $\frac{q(n)}{q(n)+r}$ (this is skipped in the Fisher algorithm since $q(n)=\infty$ and $\frac{q(n)}{q(n)+r}=1$). We then set the samples where $x(n)$ is known to their correct values and use the result as the new signal estimate. Now to reestimate the output, Fourier transform the signal, $\hat{X}_{k+1}(\omega_i)$, then find the output spectrum values which comes closest to this. If the output spectrum sample is known, we use its correct value; if it is completely unknown, we set it equal to $\hat{X}_{k+1}(\omega_i)$. If the output sample is only known to lie on a given line in the complex plane, then we retain the component of $\hat{X}_{k+1}(\omega_i)$ parallel to the line, $\mathrm{Re}\left(\hat{X}_{k+1}(\omega_i)e^{-j\phi_i}\right)e^{j\phi_i}$, and add it to the known component $j\bar{Y}_i e^{j\phi_i}$. Figure 7.7.3 illustrates this calculation:



$\hat{X}_k(\omega_i)$
$\hat{Y}_k(\omega_i) = \left(\mathrm{Re}\left(\hat{X}_k(\omega_i)e^{-j\omega_i}\right) + j\bar{Y}_i\right)e$
$\mathrm{Im}\left(Y(\omega_i)\right)$
Y_i
ϕ_i
$\mathrm{Re}\left(Y(\omega_i)\right)$
$\bar{Y}_i$
$\mathrm{Re}\left(\hat{X}_k(\omega_i)e^{-j\omega_i}\right)$

Figure 7.7.3 - Estimating $\hat{Y}(\omega_i)$ when Constraint Set is a Line

Each iteration strictly decreases the likelihood function $\frac{1}{2}\|\hat{x}_k\|_Q^2 + \frac{1}{2}\|\hat{y}_k - \hat{x}_k\|_R^2$, and the Bayesian algorithm's estimates (computed with finite $q(n)$) are guaranteed to converge to the unique global optimizing solution at a rate given in (7.4.3). This geometric convergence rate is guaranteed even if $N=\infty$.

Analyzing the convergence rate of the Fisher algorithm $(q(n)=\infty)$ is, as usual, slightly more difficult. The number of signal constraints p and output constraints q is given by:

p = number of zero-dimensional sets X_n

q = 2 * number of zero-dimensional sets Y_i for $0<\omega_i<\pi$

+ number of one-dimensional sets Y_i for $0<\omega_i<\pi$

+ 1 if Y_0 is zero-dimensional

+ 1 if N is even and $Y_{N/2}$ is zero-dimensional

Counting the output constraints is complicated by the conjugate symmetry of the constraint sets, $Y_{N-i}=Y_i^*$. Now our previous results guarantee that a global minimizing solution to the Fisher problem always exists, and the iterative algorithm above will converge to the nearest global minimizing solution. Assuming no computation noise, if we start at the minimum norm element in Y, as suggested in (7.7.3), then the algorithm converges to the minimum energy solution $(\hat{x}_{min},\hat{y}_{min})$. If $\min(p,q,N-p,N-q)<\infty$, then the number of non-zero eigenvalues which are strictly less than one must be finite. The supremum of these, λ_{max}, will therefore be strictly less than 1, and the algorithm will converge at the rate λ_{max} to the limit $\hat{x},\hat{y}$:

$$\sum_{n=0}^{N-1}\left|\hat{y}_{k+1}(n)-\hat{y}(n)\right|^2 \le \lambda_{max}\sum_{n=0}^{N-1}\left|\hat{x}_{k+1}(n)-\hat{x}(n)\right|^2 \tag{7.7.4}$$
$$\le \lambda_{max}^2\sum_{n=0}^{N-1}\left|\hat{y}_k(n)-\hat{y}(n)\right|^2$$

Finally, the solution will be non-unique if the intersection $N_x \cap N_y \ne \{0\}$, or in other words, if there exists a non-zero signal $v(n)$ such that:

$$\begin{aligned} v(n) &= 0 && \text{wherever the sample } x(n) \text{ is known} \\ V(\omega_i) &= 0 && \text{wherever the sample } Y(\omega_i) \text{ is known exactly} \\ \mathrm{Im}\left\{V(\omega_i)e^{-j\phi_i}\right\} &= 0 && \text{wherever the sample } Y(\omega_i) \text{ is known to lie on a line} \end{aligned} \tag{7.7.5}$$

In particular, if there are fewer constraints than points, $p+q<N$, then there will be an $N-p-q$ dimensional subspace of such signals $v(n)$. Then if $(\hat{x},\hat{y})$ is any solution to the Fisher problem, then $(\hat{x}+\underline{v},\hat{y}-\underline{v})$ will be another solution. If the solution is non-unique, the computation noise can accumulate at a constant rate as the iteration progresses, possibly growing infinitely large. If the solution is unique, the computational noise sensitivity is proportional to $\frac{1}{1-\lambda_{max}^2}$.

8. Dual Algorithm

The dual algorithm takes a form similar to that of the primal algorithm, except that the projection operators are orthogonal to those used in the primal algorithm. This can change the dimensionality of the problem quite dramatically. Substituting the model density and constraint sets into the dual algorithm in chapter 5 gives the following iteration:

Dual Iterative Algorithm:

Guess $\hat{\rho}_{y_0}(\omega_i)$

For $k=0,1,\cdots$

$$\hat{\rho}_{x_{k+1}}(n) = \begin{cases} x(n) - \dfrac{q(n)}{q(n)+r}\hat{\rho}_{y_k}(n) & \text{if } X_n = \{x(n)\} \\ 0 & \text{if } X_n = \mathbf{R} \end{cases}$$

$$\hat{\rho}_{y_{k+1}}(\omega_i) = \begin{cases} Y(\omega_i) - \hat{\rho}_{x_{k+1}}(\omega_i) & \text{if } Y_i = \{Y(\omega_i)\} \\ j\left(\bar{Y}_i - \mathrm{Im}\left(\hat{X}_{k+1}(\omega_i)e^{-j\phi_i}\right)\right)e^{j\phi_i} & \text{if } Y_i \text{ is a line} \\ 0 & \text{if } Y_i = \mathbf{C} \end{cases}$$

Iterate sufficiently, then:

$$\hat{x}_{k-1}(n) = \hat{\rho}_{x_{k+1}}(n) + \frac{q(n)}{q(n)+r}\hat{\rho}_{y_k}(n)$$
$$\hat{y}_{k+1}(n) = \hat{\rho}_{x_{k+1}}(n) + \hat{\rho}_{y_{k+1}}(n)$$

The Fisher algorithm is identical, except that the *a priori* signal covariance $q(n)$ is infinite, and so the filtering step can be skipped. Start with an estimate of the Fourier transform of the output multiplier; the initial estimate is arbitrary, so we might just as well choose $\hat{\rho}_{y_0}(\omega_i) = 0$. Inverse Fourier transform to find the time domain vector $\hat{\rho}_{y_k}(n)$. To estimate the signal multiplier, wherever the signal sample is unknown, set $\hat{\rho}_{x_{k+1}}(n)$ to zero; wherever $x(n)$ is known, subtract a filtered output multiplier estimate $\frac{q(n)}{q(n)+r}\hat{\rho}_{y_k}(n)$ from $x(n)$ to estimate $\hat{\rho}_{x_{k+1}}(n)$. Now to reestimate the output multiplier, Fourier transform the signal multiplier, giving $\hat{\rho}_{x_{k+1}}(\omega_i)$. Wherever the frequency sample $Y(\omega_i)$ is unknown, set $\hat{\rho}_{y_{k+1}}(\omega_i)$ to zero. At frequencies where $Y(\omega_i)$ is known, estimate $\hat{\rho}_{y_{k+1}}(\omega_i)$ by subtracting $\hat{\rho}_{x_{k+1}}(\omega_i)$ from $Y(\omega_i)$. Finally, wherever $Y(\omega_i)$ is known to lie on the line $\mathrm{Im}\left(Y(\omega_i)e^{j\phi_i}\right) = \bar{Y}_i$, throw away all but the component of $\hat{\rho}_{x_{k+1}}(\omega_i)$ which is *perpendicular* to the line, and subtract from the known component $\bar{Y}_i$ of $Y(\omega_i)$ (see figure 7.8.1).

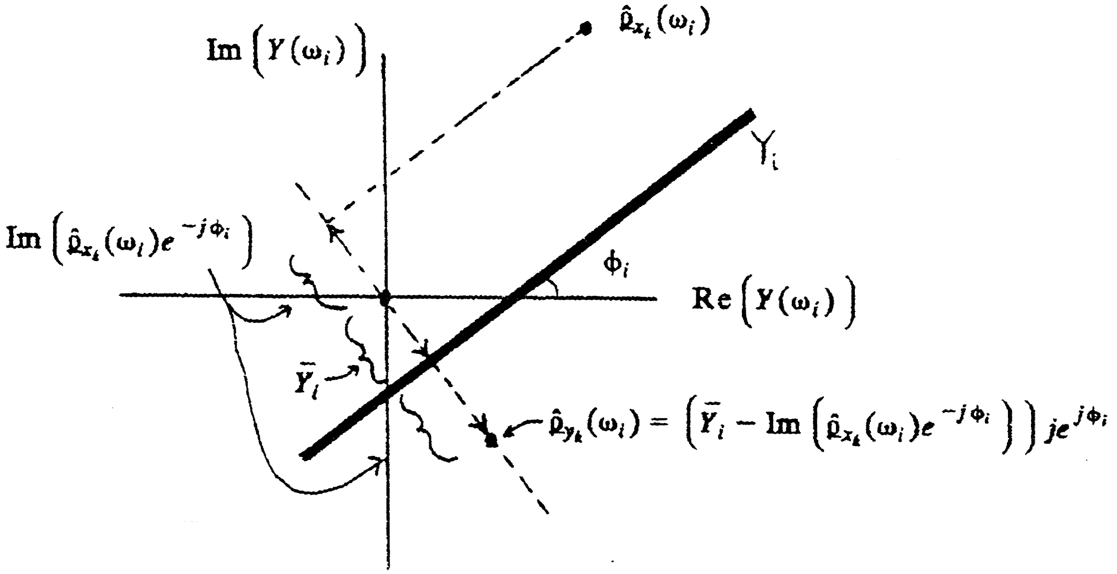

$\hat{\rho}_{x_k}(\omega_i)$
$\mathrm{Im}\left(Y(\omega_i)\right)$
Y_i
$\mathrm{Im}\left(\hat{\rho}_{x_k}(\omega_i)e^{-j\phi_i}\right)$
ϕ_i
$\mathrm{Re}\left(Y(\omega_i)\right)$
$\bar{Y}_i$
$\hat{\rho}_{y_k}(\omega_i) = \left(\bar{Y}_i - \mathrm{Im}\left(\hat{\rho}_{x_k}(\omega_i)e^{-j\phi_i}\right)\right) je^{j\phi_i}$

Figure 7.3.1 - Calculation of $\hat{\rho}_y(\omega_i)$ when Constraint Set is a Line

After enough iterations, signal and output estimates can be calculated by adding the signal and output multiplier estimates. Each iteration gives better estimates, and the Bayesian algorithm is guaranteed to converge to the unique global optimum solution at the same geometric rate as the primal algorithm. The Fisher algorithm ($q(n)=\infty$) will converge to the minimal energy estimates ($\hat{x}_{min},\hat{y}_{min}$) regardless of the initial estimate $\hat{\rho}_{y_0}$. If the signal and output estimates are not equal, however, the signal and output multipliers $\hat{\rho}_{x_k}$, $\hat{\rho}_{y_k}$ will only converge to a linear ramp, growing proportionally to $k(\hat{y}_{min}-\hat{x}_{min})$. The signal and output estimates $\hat{x}_k$, $\hat{y}_k$ still converge geometrically, however. If min($p,q,N-p,N-q$) is finite, then there are a finite number of non-zero eigenvalues less than 1. The largest of these, λ_{max}, will be strictly less than one, and the Fisher algorithm will therefore converge at the rate λ_{max}.

The major difference between the dual and primal algorithms is that the dimensions of the problems can be quite different. In the primal algorithm, $N-p$ signal components and $N-q$ output components are unknown and must be estimated. In the dual algorithm, p signal multiplier components are unknown and q output multiplier components are unknown. In some applications, most notably the band-limited extrapolation problem we discuss below, there may be a large computational advantage to choosing the algorithm with the fewest unknowns.

9. Closed-Form Solutions

Formulas for $\underline{x}$, $\underline{y}$, $\underline{\rho}_x$ and $\underline{\rho}_y$ can be stated by constructing the equations defining the constraint sets, $\mathrm{G}_x\underline{x} = \underline{\gamma}_x$ and $\mathrm{G}_y\underline{y} = \underline{\gamma}_y$, and substituting these into our formulas in chapters 5 and 6. The chief point of interest is that the number of equations to be solved can be reduced by solving only for the unknown components of these vectors. Since $N-p$ components of $\underline{x}$, $N-q$ components of $\underline{y}$, p components of $\underline{\rho}_x$ and q components of $\underline{\rho}_y$ are unknown, we can minimize the computational difficulty by solving for the variable with the fewest unknowns.

We will state the primal formula for $\underline{x}$ and the dual formula for $\underline{\rho}_x$; the derivation of these is not very illuminating, and thus we will skip the intermediate steps. The hardest part is keeping track of the indexing. Let us define:

$n_1, \ldots, n_p$	known signal samples
$n_{p+1}, \ldots, n_N$	unknown signal samples
$i_1, \ldots, i_s$	known frequency samples
$i_{s+1}, \ldots, i_t$	frequency samples on line
$i_{t+1}, \ldots, i_N$	unknown frequency samples

Then the primal solution for the unknown components of $\underline{x}$ is:

$$\mathbf{M}_x \begin{pmatrix} x(n_{p+1}) \\ \vdots \\ x(n_N) \end{pmatrix} = \begin{pmatrix} \hat{x}_1(n_{p+1}) \\ \vdots \\ \hat{x}_1(n_N) \end{pmatrix} \tag{7.9.1}$$

where $\hat{x}_1$ is the signal estimate generated by one pass of our iterative algorithm starting at:

$$\hat{x}_0 = \begin{cases} x(n) & n = n_1, \ldots, n_p \\ 0 & \text{else} \end{cases} \tag{7.9.2}$$

and the matrix $\mathbf{M}_x$ is given by:

$$\left[\mathbf{M}_x\right]_{l,j} = \delta_{l,m} - \frac{q(n_l)}{q(n_l)+r}\frac{1}{N}\left[\sum_{i=i_t+1}^{i_N} e^{j\omega_i(n_l-n_m)} + \frac{1}{2}\sum_{i=i_s+1}^{i_t}\left(e^{j\omega_i(n_l-n_m)} + e^{j(\omega_i(n_l+n_m)+2\varphi_i)}\right)\right] \tag{7.9.3}$$

The rows and columns of $\mathbf{M}_x$ have indices $l, m = p+1, \ldots, N$.

Similarly, the dual closed form solution for the unknown components of $\hat{\rho}_x$ can be put into the form:

$$\mathbf{M}_{\rho_x}\begin{pmatrix} \rho_x(n_1) \\ \vdots \\ \rho_x(n_p) \end{pmatrix} = \begin{pmatrix} x(n_1)-\bar{\rho}_y(n_1) \\ \vdots \\ x(n_p)-\bar{\rho}_y(n_p) \end{pmatrix} \tag{7.9.4}$$

where $\bar{\rho}_y$ is the inverse DFT of the known frequency components:

$$\bar{\rho}_y(\omega_i) = \begin{cases} Y(\omega_i) & \text{if } Y(\omega_i) \text{ known} \\ j\,\bar{Y}_i\, e^{j\phi_i} & \text{if } Y(\omega_i) \text{ on line} \\ 0 & \text{if } Y(\omega_i) \text{ unknown} \end{cases} \tag{7.9.5}$$

and the matrix $\mathbf{M}_{\rho_x}$ looks similar to $\mathbf{M}_x$, but with the indexing modified:

$$\left[\mathbf{M}_{\rho_x}\right]_{l,m} = \delta_{l,m} - \frac{q(n_l)}{q(n_l)+r}\frac{1}{N}\left[\sum_{l=i_1}^{i_s} e^{j\omega_i(n_l-n_m)} + \frac{1}{2}\sum_{i=i_s+1}^{i_t}\left(e^{j\omega_i(n_l-n_m)} - e^{j(\omega_i(n_l+n_m)+2\phi_i)}\right)\right] \tag{7.9.6}$$

The rows and columns of M_{ρ_x} have indices $l,m=1,\ldots,p$. Beware that if $\hat{y}_{min}\neq\hat{x}_{min}$, then this dual formula will have no solution, and a best least squares solution for $\hat{\rho}_x$ must be found instead. The remaining components of $\hat{\rho}_x$ are zero. To calculate signal and output estimates, run one pass of the dual algorithm, starting with $\hat{\rho}_x$, then add the signal and output multipliers together. Formulas for $\underline{y}$ and ρ_y also can be derived, but they are somewhat messier.

The Lagrange multiplier solution used in the dual algorithm can be interpreted as finding a minimum norm solution to the original constraint equations:

$$\begin{pmatrix} G_x & 0 \\ 0 & G_y \end{pmatrix} \begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix} = \begin{pmatrix} \gamma_x \\ \gamma_y \end{pmatrix} \tag{7.9.7}$$

where these equations always have at least one solution because we have assumed that G_x and G_y have full row rank. A more numerically robust procedure for calculating $\underline{x}$ and $\underline{y}$, therefore, would use a Singular Value Decomposition to solve (7.9.7) directly via orthogonal transformations. This topic, however, is beyond the scope of this thesis (see, for example, Lawson and Hanson [10]).

10. Conjugate Gradient Algorithms

Solving these closed-form solutions directly usually requires a lot of storage and computation. The iterative primal and dual algorithms also tend to converge rather slowly. A more efficient approach, therefore, is to use either a PARTAN or conjugate gradient algorithm to solve any of our four formulas for $\underline{x}$, $\underline{y}$, ρ_x or ρ_y. Each step of the algorithm looks like one step of our primal or dual iteration followed by two line search extrapolation steps. Convergence, however, is guaranteed in a number of steps equal to the number of unknowns. Common sense would therefore suggest using the procedure to estimate the variable $\underline{x}$, $\underline{y}$, $\underline{\rho}_x$, or $\underline{\rho}_y$ which has the fewest unknown components. We will not repeat the details of the algorithm, since we have already stated it in chapters 5 and 6. We will note, however, that there are two strategies that could be used. The key step in the conjugate gradient algorithm for solving:

$$(I-P_x H P_y P_x)\hat{\underline{x}} = \bar{\underline{x}} + P_x H(\bar{\underline{y}} + P_y \bar{\underline{x}}) \quad . \tag{7.10.1}$$

for example, is multiplying the direction vector $\underline{d}_k$ by the matrix $(I-P_x H P_y P_x)$ to get $\underline{e}_k$. This could be implemented by performing the indicated series of projections and filtering steps on $\underline{d}_k$, thus making it unnecessary to compute or store this matrix directly. Alternatively, in some cases it may be easier to exploit the fact that p components of $\underline{x}$ are already known, so that the conjugate gradient algorithm only searches over directions $\underline{d}_k$, $\underline{e}_k$, $\underline{g}_k$ having $N-p$ non-zero points located at sample values where $x(n)$ is unknown. We can therefore reduce the dimensionality of the problem by removing the p unnecessary rows and columns from the matrix $(I-P_x H P_y P_x)$, thus leaving a smaller number of equations to be solved. This is exactly the same trick used to derive the reduced primal (7.9.1) and dual (7.9.4) formulas. Now it will be necessary on each step to multiply the search direction $\underline{d}_k$ by the matrix M_x or M_{ρ_x}, and thus this matrix will have to be stored or be computed on each pass. In applications where this matrix M_x or M_{ρ_x} has a simple form, this "brute force" approach can be easier and faster than actually computing all the FFT's and projections.

11. Eigenvalues/Eigenvectors of Fisher Problem

Finally, we interpret the eigenvalue/eigenvector properties discussed in chapter 6 for the Fisher problem. Let $\{\underline{\psi}_m\}$ be an orthonormal set of eigenvectors of $P_x P_y P_x$ with non-zero eigenvalues $\lambda_m > 0$. Thus $\underline{\psi}_m \in N_x$, which implies that $\psi_m(n)=0$ wherever the signal $x(n)$ is known. Let us define $\underline{\phi}_m = \frac{1}{\sqrt{\lambda_m}} P_y \underline{\psi}_m$, or in other words:

$$\phi_m(\omega_i) = \begin{cases} 0 & \text{if } Y_m = \{Y(\omega_m)\} \\ \frac{1}{\sqrt{\lambda_m}} \operatorname{Re}\left(\psi_m(\omega_i) e^{-j\phi_i}\right) e^{j\phi_i} & \text{if } Y_m \text{ is a line} \\ \frac{1}{\sqrt{\lambda_m}} \psi_m(\omega_i) & \text{if } Y_m = \mathrm{C} \end{cases} \qquad (7.11.1)$$

These vectors ϕ_m, which are simply projections of ψ_m onto the output constraint null space, are orthonormal eigenvectors of $\mathrm{P}_y \mathrm{P}_x \mathrm{P}_y$ with eigenvalue λ_m. Projecting ϕ_m back onto the signal constraint null space by setting its inverse transform to zero wherever $x(n)$ is known, gives back the original vector $\psi_i = \frac{1}{\sqrt{\lambda_i}} \mathrm{P}_x \phi_i$.

Now consider the subset $\{\psi_m\}$ of orthonormal eigenvectors of $\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x$ with non-zero eigenvalues which are strictly less than 1, $0<\lambda_m<1$. Project ψ_m onto the orthogonal complement null space of the output constraint, $\eta_m = \frac{1}{\sqrt{1-\lambda_m}} \mathrm{Q}_y \psi_m$ by setting the frequency components of ψ_m to zero wherever $Y(\omega_i)$ is *unknown*. Then the vectors η_m are an orthonormal set of eigenvectors of the dual iteration matrix $\mathrm{Q}_y \mathrm{Q}_x \mathrm{Q}_y$ with eigenvalue λ_m. Furthermore, setting $\eta_m(n)$ to zero wherever $x(n)$ is unknown, $\xi_m = \frac{1}{\sqrt{1-\lambda_m}} \mathrm{Q}_x \eta_m$, gives an orthonormal set of eigenvectors of the dual iteration matrix $\mathrm{Q}_x \mathrm{Q}_y \mathrm{Q}_x$ with eigenvalue λ_m.

12. Special Case - Bandlimited Extrapolation

The best known example of an optimal signal reconstruction problem with linear equality time and frequency constraints is the problem of extrapolating a finite segment of a bandlimited sequence. The usual statement of this problem sets $N=\infty$, assumes that $x(0), \ldots, x(p-1)$ are known, and assumes that the signal is low pass, so that all frequency samples beyond a certain point are zero, $Y(\omega_i)=0$ for $|\omega_i|>\omega_c$:

$$X_n = \begin{cases} \mathrm{R} & \text{for } n=0, \ldots, p-1 \\ \{\underline{0}\} & \text{else} \end{cases} \tag{7.12.1}$$

$$Y_i = \begin{cases} \mathrm{C} & \text{for } |\omega_i| \leq \omega_c \\ \{\underline{0}\} & \text{else} \end{cases}$$

Using these constraint sets, we can now apply our primal and dual algorithms above, state four different closed form solutions for $\underline{x}$, $\underline{y}$, $\underline{\rho}_x$, and $\underline{\rho}_y$, and list the usual variety properties which the eigenvalues and eigenvectors satisfy.

Due to its importance, this problem has been studied by numerous researchers. The original results for this problem, developed in the 50's and 60's, concerned extrapolating continuous-time bandlimited signals. The theory for continuous-time signals has many similarities to that for discrete time signals, but there are some major differences. The most important difference is that a bandlimited continuous-time signal is analytic, and thus in theory the unique extrapolation can be calculated simply by representing the infinite length signal by its Taylor series expansion about any point in the known interval:

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \tfrac{1}{2}f''(x_0)(x-x_0)^2 + \cdots \tag{7.12.2}$$

This procedure, however, is excruciatingly sensitive to errors in the measurement of the derivatives. Slepian, *et. al.*[11] and others at Bell Labs [12,13,14] studied this problem in a search for a more robust extrapolation procedure. They analyzed the eigenfunctions of the problem, calling them the "prolate spheroidal wave functions", and showed that the eigenvalues were between 0 and 1, that the eigenfunctions formed a complete orthonormal basis, and that the eigenfunction decomposition could be used to construct the correct extrapolation. Papoulis [1] suggested an iterative procedure for solving the problem, which is the continuous-time analog of our primal algorithm, and used the properties of prolate spheroidal wave functions to prove that the algorithm converges.

Gerchberg [15] suggested a similar algorithm for reconstructing the high frequency components of a finite length signal which has been low pass filtered.

As digital computers became more available, attention shifted to the discrete-time problem of extrapolating a bandlimited sequence. Unlike the continuous-time problem, the extrapolation of a discrete bandlimited sequence is generally non-unique. Sabri and Steenaart [16] studied the discrete-time problem and presented a discrete-time solution corresponding to our primal closed-form solution for $\underline{x}$. Cadzow [17] reconsidered the problem and by sampling the continuous time problem derived a new closed-form solution, corresponding to our dual closed-form solution for $\underline{\rho}_x$. Jain and Ranganath [2] present an excellent analysis of the problem which comes closest to our approach (it was published half a year after this thesis proposal was submitted.) They use both Bayesian and Fisher approaches to the problem and focus attention on the minimum norm solution to the primal problem, which they define as the unique best extrapolation of the sequence. They also present our dual closed-form solution, a conjugate gradient algorithm (we borrowed this idea from them), a detailed discussion of the properties of the eigenvectors and eigenvalues (the "discrete prolate spheroidal functions"), consider both clutter and noise, and add a comparison of periodic versus non-periodic (finite N versus infinite N) extrapolation. Our major improvements to this paper are recognizing that the iterative algorithm is most naturally developed by starting with the problem of minimizing $||\underline{y}-\underline{x}||^2$, and also recognizing that virtually all the properties that have been derived for this problem can be extended to the general problem of reconstruction from linear equality constraints. Our dual iterative algorithm and its interpretation in terms of Lagrange multipliers also appears to be new.

We will not repeat the presentation of our results for this particular problem, but will merely point out some features of the problem which are unique. The primal iterative algorithm simply alternates between filtering, forcing the known signal samples to their correct values, and then low pass filtering the result. The dual iterative algorithm alternates between filtering, truncating the output multiplier to the length of the given signal and subtracting it from the given segment, then high pass filtering and negating the result. Both of these algorithms decrease the objective function on each iteration and are guaranteed to converge to a solution. The problem with these simple iterative algorithms is that if N is very large, then calculating the necessary FFT's will be cumbersome. Also, since any computation will have to take N to be finite rather than infinite, aliasing errors will be introduced. Another problem is that the total number of known signal samples, p, and known frequency samples, q, is usually much smaller than the number of points, $p+q<N$. In theory, this does not affect the Bayesian algorithms ($q(n)<\infty$), since these will always converge at a geometric rate to the unique global minimizing solution. In the Fisher algorithms, however, with $q(n)=\infty$, the extrapolation will be non-unique. The primal algorithm must therefore be initialized to start at the minimum energy initial spectral estimate, $\hat{Y}_0(\omega_i)=0$, to ensure convergence to the minimum energy extrapolation. The dual algorithm, on the other hand, will always give the minimal energy solution. The total number of non-zero eigenvalues less than 1 will be bounded above by $p<\infty$; this ensures that both the primal and dual Fisher problems will converge at a rate of $\lambda_{max}<1$.

An *a priori* estimate of the convergence rate for the Fisher problem with $q(n)=\infty$ is given by:

$$\lambda_{max} \geq 1 - \frac{\| \underline{\bar{y}} + \mathrm{P}_y(\underline{\bar{x}} + \mathrm{P}_x \underline{\bar{y}}) \|}{\| \underline{\hat{y}} \|} \approx 1 - \left(\frac{\sum_{n=0}^{p-1} \hat{x}^2(n)}{\sum_{n=-\infty}^{n=\infty} \hat{x}^2(n)} \right)^{1/2} \tag{7.12.3}$$

Thus the largest eigenvalue less than 1 is bounded below by the ratio of energy in the known segment to the energy of the total reconstructed signal. If the reconstructed signal tails are large compared to the known signal segment, then λ_{max} must be very close to 1, which will make the problem ill-conditioned, slow the convergence rate, and make the problem very sensitive to computation noise. Our Bayesian algorithms try to cure this ill-conditioning by using filtering. However, the resultant signal tail estimate will have less energy than the Fisher algorithm's estimates (see chapter 6, section 2). Furthermore, the closer the eigenvalue λ_{max} is to one and the more ill-conditioned the Fisher problem becomes, the more drastic is the effect of the filter.

Closed-form solutions for both the primal and dual problems are easy to state. With some algebra, setting $N=\infty$ in the primal closed form solution for $\hat{x}$ gives:

$$\mathbf{M}_x \underline{x} = \hat{x}_1 \tag{7.12.4}$$

where:

$\hat{x}_1$ = result of one pass of primal algorithm, starting at:

$$\hat{x}_0(n) = \begin{cases} x(n) & n=0, \ldots, p-1 \\ 0 & \text{else} \end{cases}$$

and:

$$\left[\mathbf{M}_x\right]_{l,m} = \begin{cases} \delta_{l,m} & \text{if } 0 \le l < p \quad \text{or} \quad 0 \le m < p \\ \delta_{l,m} - \left(\dfrac{q(l)}{q(l)+r}\right) \dfrac{\sin(\omega_c(l-m))}{\pi(l-m)} & \text{else} \end{cases}$$

Unfortunately, since $p+q<N$, the solution for $\hat{x}$ is non-unique, and the matrix on the left will be non-invertible.

The closed-form solution for $\underline{\rho}_x$ can also be easily computed. Recognizing that all the components of $\underline{\rho}_x$ will be zero except for $n=0, \ldots, p-1$, it is easy to simplify this dual-form solution so that it involves only a $p \times p$ matrix:

$$M_{\rho_x}\begin{pmatrix}\rho_x(0)\\ \vdots\\ \rho_x(p-1)\end{pmatrix} = \begin{pmatrix}x(0)\\ \vdots\\ x(p-1)\end{pmatrix} \tag{7.12.5}$$

$$\text{where:} \quad \left[M_{\rho_x}\right]_{l,m} = \left(\frac{r}{q(l)+r}\right)\delta_{l,m} + \left(\frac{q(l)}{q(l)+r}\right)\frac{\sin(\omega_c(l-m))}{\pi(l-m)}$$

Because $p+q<N$, there will be many solutions satisfying $\hat{\underline{x}}=\hat{\underline{y}}$, and thus this equation for $\hat{\underline{\rho}}_x$ can always be solved. Moreover, it can be shown that since $q \geq p$ then $\hat{\underline{\rho}}_x$ must be unique and thus this matrix M_{ρ_x} will be invertible. Furthermore, M_{ρ_x} is Toeplitz, and so we can solve for $\hat{\underline{\rho}}_x$ by using the Levinson-Trench algorithm [18,19,20]. Finally, since $\hat{\underline{\rho}}_x$ is finite length, we will not have any aliasing errors. Clearly this dual problem is the method cf choice for the bandlimited extrapolation problem.

The conjugate gradient algorithm is also easily implemented for this dual problem. The key step on each iteration will be computing the vector $\underline{e}_k = (I-Q_x H Q_y Q_x)\underline{d}_k$. Although this could be done by performing the indicated series of high pass, filter and truncation operations, a much faster approach would be to recognize that only components $n=0,\ldots,p-1$ of all the vectors $\underline{g}_k$, $\underline{\rho}_{x_k}$, $\underline{d}_k$ and $\underline{e}_k$ will be non-zero. Eliminating the unnecessary rows and columns of $(I-Q_x H Q_y Q_x)$ leaves the matrix M_{ρ_x} in (7.12.5). Thus we only need to calculate $\underline{e}_k = M_{\rho_x}\underline{d}_k$ on each step, an operation requiring only a small amount of calculation.

The eigenvectors of this problem are the discrete prolate spheroid vectors. Our previous analysis guarantees that they form an orthonormal basis and at most p of them have non-zero eigenvalues less than 1. Furthermore, not only are the eigenvectors $\underline{\psi}_i$ of $P_x P_y P_x$ orthonormal, but so are the low pass vectors $\underline{\phi}_i = \frac{1}{\sqrt{\lambda_i}}P_y\underline{\psi}_i$, the high pass vectors $\underline{\eta}_i = \frac{1}{\sqrt{1-\lambda_i}}(I-P_y)\underline{\psi}_i$, and the truncated low pass vectors

$$\xi_i = \frac{1}{\sqrt{1-\lambda_i}}(I-P_x)\phi_i .$$

We could also extend our analysis to include arbitrary signal Q and noise R covariance matrices. Jain and Ranganath [2] consider one particular choice. The only difficulty will be that calculating the projection matrices and filter will be more difficult, and the matrix M_{p_x} in the dual algorithm will have a more complicated form.

We conclude by mentioning that there is a much simpler approach available for the problem of extrapolating a band-limited finite signal segment, if we are willing to approximate the bandpass characteristic by an autoregressive-moving-average (ARMA) filter response. Musicus has developed an extrapolation/interpolation/filtering algorithm for noisy ARMA models which is based on the linear equality constraint algorithms developed in this thesis. This method iteratively filters the observed data in the frequency domain, linearly predicts the signal tails, uses these tails as its estimate of the noisy observation tails, then repeats the process. Convergence is usually achieved in a very small number of iterations.

13. Special Case - Phase-Only Reconstruction Modulo π

Another application of recent interest is reconstructing a finite length sequence given noisy samples of the phase of its transform. Hayes [3, 4] proved that it is possible to uniquely reconstruct a finite length one- or multi-dimensional signal from samples of its phase modulo π or 2π together with the value of its first non-zero point, provided only that the signal has no symmetric factors (a situation which has measure-zero probability if the sequences are random.) Hayes, Lim and Oppenheim suggested two closed-form solutions for the signal given the phase modulo π or 2π, and they also suggested an iterative procedure for solving the problem given the phase modulo 2π. This procedure iterates between forcing the time domain constraints, and then replacing the phase of the Fourier Transform with its correct value. Tom, Hayes, Quatieri, and McClellan [21] use non-expansive mapping theory to prove that if there exists a unique sequence meeting all the given constraints, then this algorithm converges to that solution. (Otherwise their algorithm can diverge.) Quatieri and Oppenheim [22] considered using a similar algorithm for reconstructing a minimum phase signal from its phase, and Espy [23] investigated the noise sensitivity of the problem.

There is a much simpler approach to this problem than that used by Hayes, *et.al.*, which also suggests iterative algorithms for reconstructing a finite signal from the phase modulo π, rather than 2π. Knowledge of the phase ϕ_i modulo π of the transform sample $Y(\omega_i)$ is equivalent to knowing that the transform lies on a line in the complex plane:

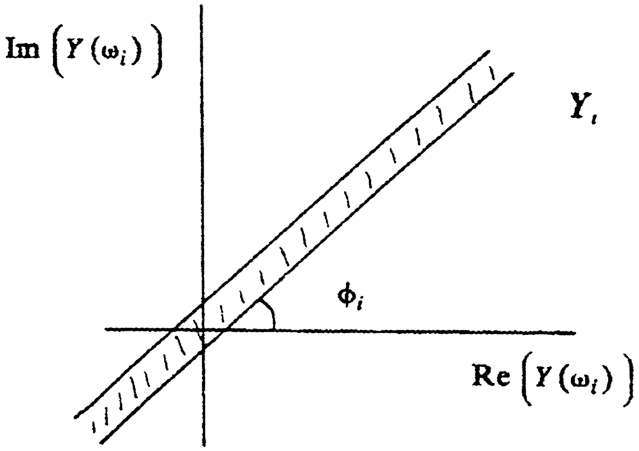

$\mathrm{Im}\left(Y(\omega_i)\right)$
Y_i
ϕ_i
$\mathrm{Re}\left(Y(\omega_i)\right)$

Figure 7.14.1 - Constraint Corresponding to Knowing Phase Modulo π

This, however, is simply a one-dimensional linear equality constraint on the transform sample, with angle ϕ_i and offset $\overline{Y}_i = 0$. The time domain constraints, knowing certain signal points and being ignorant of others, are also linear equality constraints. Therefore, we can directly apply our general linear equality constraint analysis to the phase-only reconstruction problem. In particular, we get both primal and dual iterative algorithms, closed-form solutions for $\underline{x}$, $\underline{y}$, $\underline{\rho}_x$ and $\underline{\rho}_y$, and conjugate gradient algorithms converging in a finite number of steps.

Suppose the signal is constrained to be finite length, $x(0), \ldots, x(p-1)$, and also suppose that the first point $x(0)$ is non-zero and is known. Also suppose that the phase ϕ_i modulo π of the noisy output transform sample $Y(\omega_i)$ is known at each sample of the DFT. We now use our XYMAP algorithm to to find the minimal energy finite length signal which comes as close as possible to having the correct phase. The primal iterative algorithm for solving this has the form:

<u>Primal Phase-Only Reconstruction Modulo π</u>

Guess $\hat{Y}_0(\omega_i)=0$

For $k=0,1,\cdots$

$$\hat{x}_{k+1}(n) = \begin{cases} x(0) & \text{for } n=0 \\ \dfrac{q(n)}{q(n)+r}\,\hat{y}_k(n) & \text{for } n=1,\ldots,p \\ 0 & \text{else} \end{cases} \tag{7.13.1}$$

$$\hat{Y}_{k+1}(\omega_i) = \mathrm{Re}\left(\hat{X}_{k+1}(\omega_i)e^{-j\phi_i}\right)e^{j\phi_i}$$

The signal is estimated by truncating the filtered output sequence to the right length and setting $x(0)$ to its known value. The output is then reestimated by projecting the DFT $\hat{X}_{k+1}(\omega_i)$ onto the correct phase angle, thus yielding the output with the correct phase which comes as close as possible to the signal.

The dual algorithm uses projection operators orthogonal to those of the primal algorithm:

Dual Phase-Only Reconstruction Modulo π

Guess $\hat{\rho}_{y_0}(\omega_i)=0$

For $k=0,1,\cdots$

$$\hat{\rho}_{x_{k+1}}(n) = \begin{cases} x(0) - \dfrac{q(0)}{q(0)+r}\hat{\rho}_{y_k}(0) & \text{for } n=0 \\ 0 & \text{for } n=1,\ldots,p-1 \\ -\dfrac{q(n)}{q(n)+r}\hat{\rho}_{y_k}(n) & \text{else} \end{cases}$$

$$\hat{\rho}_{y_k}(\omega_i) = j\,\mathrm{Im}\left(\hat{\rho}_{x_{k+1}}(\omega_i)e^{-j\phi_i}\right)e^{j\phi_i}$$

Iterate sufficiently, then:

$$\hat{x}_{k+1}(n) = \hat{\rho}_{x_{k+1}}(n) + \frac{q(n)}{q(n)+r}\hat{\rho}_{y_k}(n)$$

$$\hat{y}_{k+1}(n) = \hat{\rho}_{x_{k+1}}(n) + \hat{\rho}_{y_{k+1}}(n)$$

We start with an arbitrary output multiplier estimate. Set it to zero wherever $x(n)$ is unknown and subtract the filtered output multiplier from the known signal values elsewhere.. To reestimate the output, project the signal multiplier transform onto the phase angle orthogonal to that of $Y(\omega_i)$. Iterate sufficiently, then compute the signal and output estimates by adding the multipliers.

The Bayesian versions of both algorithms are guaranteed to converge to the global optimizing XYMAP solution at a geometric rate. Because p is finite, the Fisher primal algorithm, $q(n)=\infty$, will converge to the nearest solution at a geometric rate, $\lambda_{max}<1$. Starting at the minimum norm estimate $\hat{Y}_0(\omega_i)=0$ guarantees convergence to the minimum norm solution. The Fisher dual algorithm will also converge at the same geometric rate λ_{max}, although if no finite length signal exists with the given phase (a situation that may occur if there are more than p+1 phase samples) then the multipliers actually grow proportionally to $k(\hat{\underline{y}}_{min}-\hat{\underline{x}}_{min})$ on each iteration. The signal and output estimates, however, will always converge to the minimum norm solution.

The convergence rate of the Bayesian algorithm is given by (7.4.3). A lower bound for the convergence rate λ_{max} for the Fisher algorithm is given by:

$$\lambda_{max} \geq 1 - \frac{\|\bar{\underline{x}}+P_x(\bar{\underline{y}}+P_y\bar{\underline{x}})\|}{\|\hat{\underline{x}}\|} \approx 1 - \frac{|x(0)|}{\left(\sum_{n=0}^{p} x^2(n)\right)^{1/2}} \tag{7.13.2}$$

Thus a lower bound for the convergence rate is determined by the ratio of energy in the known signal point $x(0)$ to the total energy in the reconstructed signal. This suggests that as the number of points p to be reconstructed becomes large, then the phase-only reconstruction problem inevitably becomes ill-behaved. (Espy [23] observed this same phenomenon experimentally.) The Bayesian approach cures this ill-behavior by using filtering; however the Bayesian reconstructions will have much less signal energy than the Fisher reconstructions, and when the eigenvalue λ_{max} is very close to one, very small filtering levels, $h(n)\approx\lambda_{max}$, cause drastic changes in the reconstruction.

Closed form solutions can be stated for both the primal and dual problems; these can be derived from the general formulas in section 9. The primal equations are:

$$\mathbf{M}_x \begin{pmatrix} x(1) \\ \vdots \\ x(p-1) \end{pmatrix} = \begin{pmatrix} \hat{x}_1(1) \\ \vdots \\ \hat{x}_1(p-1) \end{pmatrix} \tag{7.13.3}$$

where:

$$\hat{x}_1(n) = \frac{1}{N} x(0) \sum_{i=0}^{N-1} \cos\phi_i \; e^{j(\omega_i n + \phi_i)} \qquad .$$

$$\left[\mathbf{M}_x\right]_{l,m} = \delta_{l,m} - \frac{q(n_l)}{q(n_l)+r} \frac{1}{2N} \sum_{i=0}^{N-1} \left(e^{j\omega_i(l-m)} + e^{j(\omega_i(l+m)+2\phi_i} \right)$$

$$\text{for } l,m = 1, \ldots, p-1$$

and the dual equations are:

$$\mathbf{M}_{\rho_x} \begin{pmatrix} \hat{\rho}_x(0) \\ \hat{\rho}_x(p) \\ \vdots \\ \hat{\rho}_x(N-1) \end{pmatrix} = \begin{pmatrix} x(0) \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{7.13.4}$$

$$\left[\mathbf{M}_{\rho_x}\right]_{l,m} = \delta_{l,m} - \frac{q(l)}{q(l)+r} \frac{1}{2N} \sum_{i=0}^{N-1}$$

$$\text{for } l,m = 0, p, \ldots, N-1$$

(Note the unusual indexing of the rows and columns of $\mathbf{M}_{\rho_x}$.)

If exactly $p-1$ phase samples are given, and it is known that there exists a finite length signal with the given phase, so that $\hat{x} = \hat{y}$, then we could simply solve the constraint equations

$$\begin{pmatrix} \mathbf{G}_x \\ \mathbf{G}_y \end{pmatrix} \hat{x} = \begin{pmatrix} \Upsilon_x \\ \Upsilon_y \end{pmatrix} \tag{7.13.5}$$

directly for the solution. After some algebra, this gives:

$$\bar{\mathbf{G}} \begin{pmatrix} x(1) \\ \vdots \\ x(p) \end{pmatrix} = -x(0) \begin{pmatrix} \sin\phi_1 \\ \vdots \\ \sin\phi_q \end{pmatrix} \tag{7.13.6}$$

$$\text{where } \left[\bar{\mathbf{G}}\right]_{i,n} = \sin(\omega_i n + \phi_i)$$

This is the equation used by Hayes [3]. If more than $p-1$ phase samples are known, then a best least squares approximation could also be calculated directly from (7.13.6) by a Singular Value Decomposition algorithm using orthonormalizing transformations (see Lawson and Hanson [10].

Conjugate gradient methods are convenient to use with this problem. The primal problem is usually easier to solve, because if there are more phase samples than unknown signal points, then there will be no exact solution to the dual problem. All of the usual eigenvalue/eigenvector properties can also be applied to this problem.

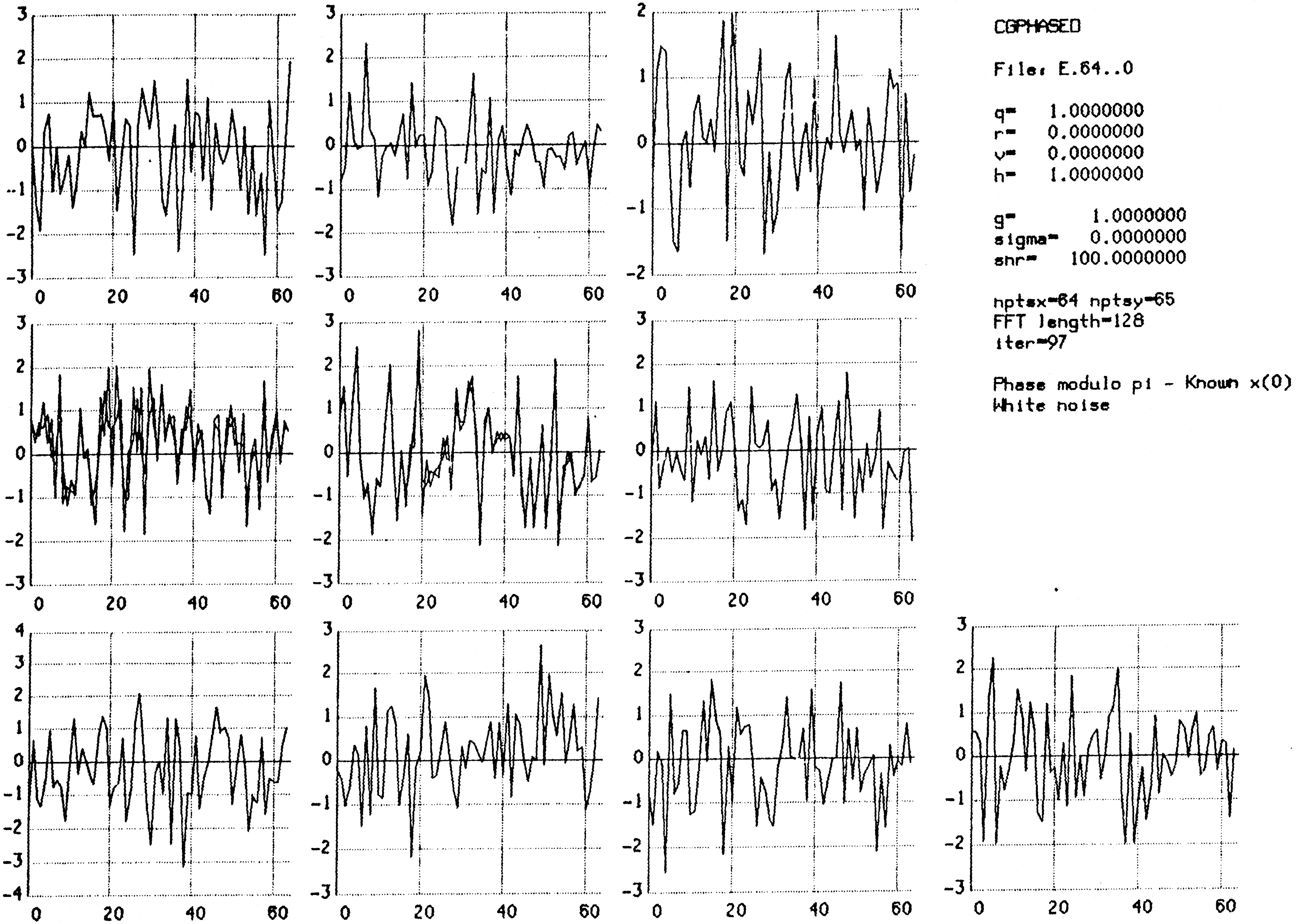
We have programmed the iterative projection algorithm, the PARTAN algorithm and the conjugate gradient algorithms for this problem in order to demonstrate the relative performance of our schemes. Figure 7.13.2 shows the convergence of our primal iterative algorithm for reconstructing ten white noise sequences of 64 points each using the exact phase of every sample of a 128 point DFT, together with the first point $x(0)$. Figure 7.13.2a shows the ten reconstructed sequences $\hat{x}_{min},\hat{y}_{min}$ from our conjugate gradient algorithm superimposed on top of the original sequence. Note that even with double precision arithmetic (64 bits precision), 2 out of 10 sequences could not be reconstructed. These two sequences appear to be very "close" to having symmetric zero-phase factors; there is so little deviation between the reconstructed signal $\hat{x}$ and output $\hat{y}$, and the slope of the objective function "valley" is so shallow, that none of our algorithms were capable of driving the estimates to their true solution. To illustrate the convergence rate of our various primal algorithms, we plotted the error $||\hat{x}_k - x_*||^2$ between the original signal x_* and the reconstructed signal $\hat{x}_k$. Figure 7.13.2b shows the convergence of our primal iterative algorithm without using any acceleration. Note that even after 1000 iterations convergence is still nowhere in sight. From the formula

$$||\hat{x}_{1000} - x_* ||^2 \le \lambda_{max}^{2000} ||\hat{x}_0 - x_* ||^2$$

we would estimate λ_{max} for this problem to be greater than .9999! Figure 7.13.2c shows the convergence rate of our conjugate gradient algorithm, which in theory is guaranteed to converge in 63 iterations, the number of unknown signal points. Note that the first 55 or so iterations seem to have virtually no effect on the error. All of the improvement comes in the last few iterations. Note, also, that somewhat more than 63 iterations were needed to achieve convergence. Figure 7.13.2d shows the value of the objective function $\frac{1}{2}||\hat{x}_k ||_Q^2 + \frac{1}{2}||\hat{y}_k - \hat{x}_k ||_R^2$ as the conjugate gradient algorithm iterates. Note that, unlike the actual reconstruction error, it is driven to a small value in the first few iterations; the remaining 50 iterations or so cause relatively little change in the objective function, although they make a very large change in the signal estimate. We also tried the PARTAN algorithm, which is also guaranteed to converge in 63 iterations. We do not show its behavior because its estimates are identical to those of conjugate gradient, but the computation appears to be slightly less numerically robust.

To summarize: this reconstruction problem is fantastically ill-behaved. In fact single precision arithmetic was not sufficient to achieve convergence. Because some eigenvalues λ_i are greater than .9999 but still less than 1, the rate of convergence is minute, and the noise sensitivity, proportional to $\frac{1}{1-\lambda_{max}^2}$ is enormous. Worse yet, most of the energy in the reconstructed signal seems to be generated by eigenvectors associated with these few eigenvalues. As p becomes larger and larger, these eigenvalues get even closer to 1; reconstructing 256 point signals, for example, was impossible with our algorithms.

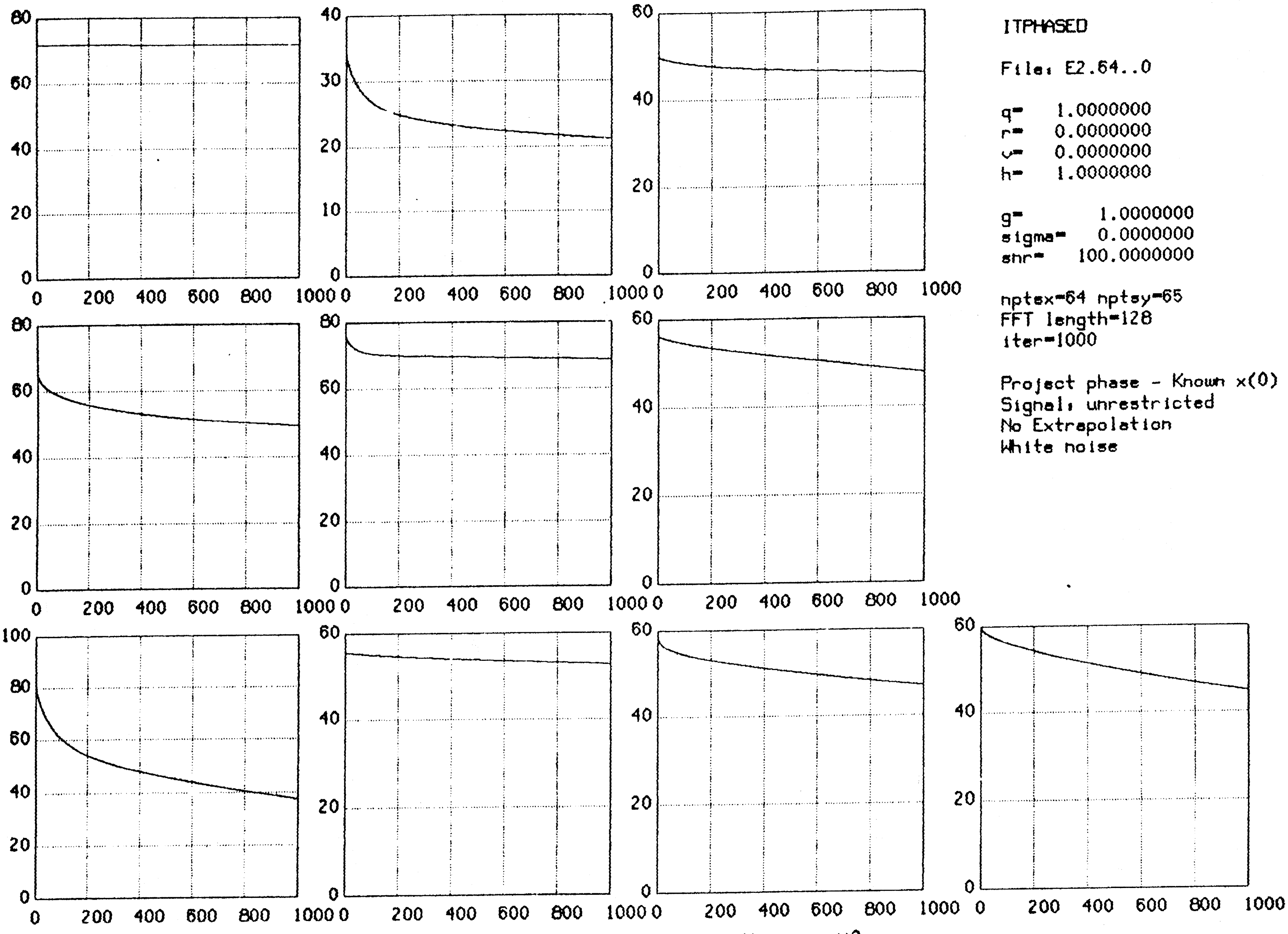
To illustrate the effect of filtering in a problem which is this badly behaved, we added a 128 point white noise sequence to our 64 point signal x_*, thus independently corrupting every frequency sample of the spectrum $X_{\bullet}(\omega_i)$ for a total signal-to-noise ratio of 40dB. We then tried to reconstruct the sequence from the phase of the noisy transform. Figures 7.13.3a and 7.13.3b show the reconstructed estimates generated by the primal Bayesian and Fisher conjugate gradient algorithms. The Bayesian algorithm uses a filter $h(n)=\frac{q(n)}{q(n)+r}\approx .9999$ on each pass; the Fisher algorithm sets $h(n)=1$. Despite the apparent insignificance of this filtering, the Bayesian estimates have drastically less energy than the Fisher estimates. One of the Fisher estimates, on the other hand, has a large high frequency oscillation caused by small amounts of noise being amplified by the high gain of the iteration. The reason the effect of the filtering is so drastic is that with $h(n)=h$ constant, the eigenvalues of $(I-P_x HP_y P_x)^{-1}$ are $\frac{1}{(1-h\lambda_i)}$. Since $\lambda_{max}>.9999$, the largest of these eigenvalues in the Fisher case will be well over 10000; in the Bayesian case with $h=.9999$, the largest of its eigenvalues will only be 5000, a difference of at least a factor of two. Note, finally, that the error between the reconstructed and the original signals in the Fisher algorithm is about 1000 times greater than the noise energy that was originally introduced.



```
CGPHASED

File: E.64..0

q=    1.0000000
r=    0.0000000
v=    0.0000000
h=    1.0000000

g=          1.0000000
sigma=      0.0000000
snr=      100.0000000

nptsx=64 nptsy=65
FFT length=128
iter=97

Phase modulo pi - Known x(0)
White noise
```

Figure 7.13.2a - 10 Reconstructed and Original Sequences,

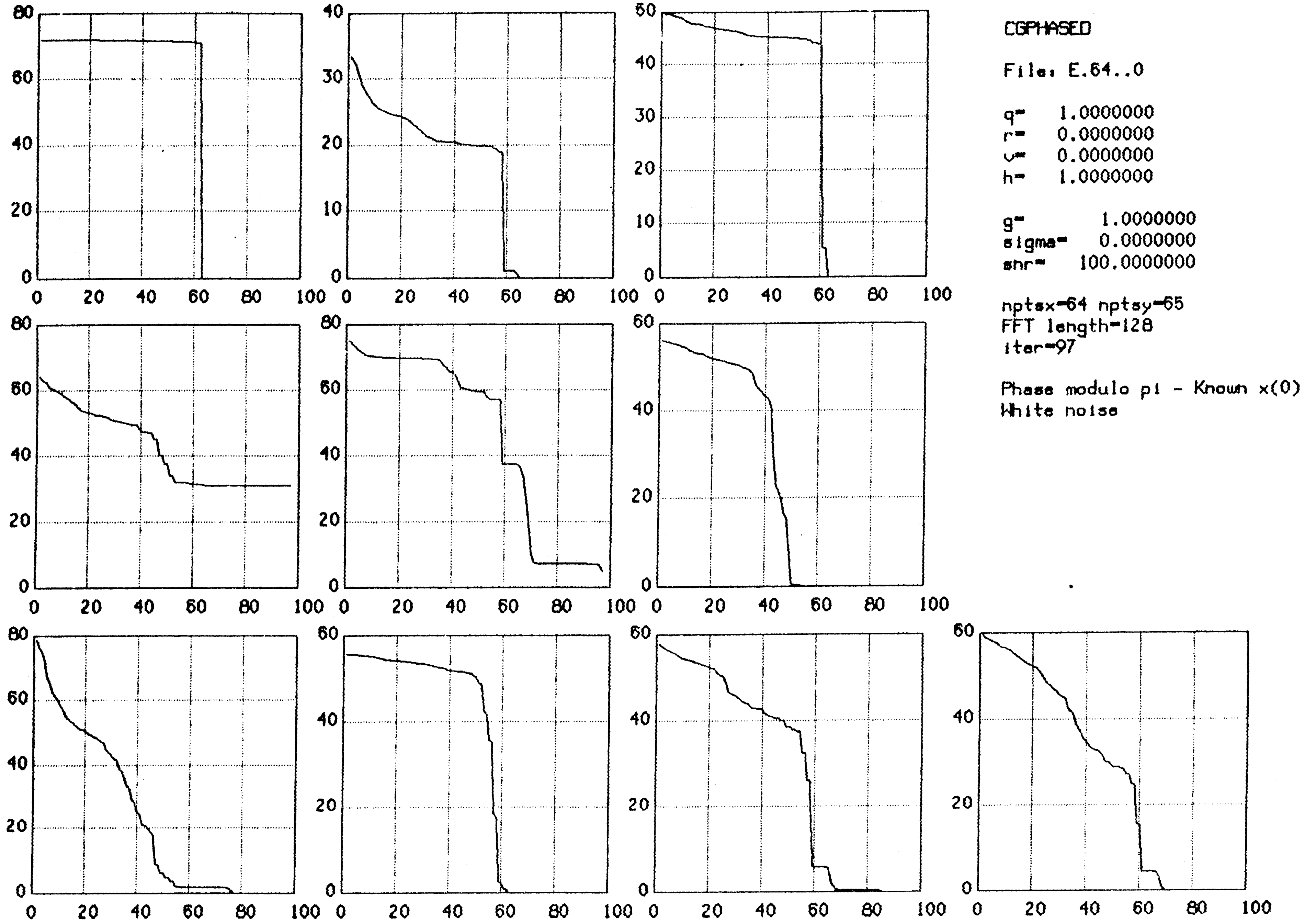
Conjugate Gradient, SNR=infinite, 64 points, 128 point FFT



```
ITPHASED

File: E2.64..0

q=    1.0000000
r=    0.0000000
v=    0.0000000
h=    1.0000000

g=          1.0000000
sigma=    0.0000000
snr=    100.0000000

nptex=64 nptsy=65
FFT length=128
iter=1000

Project phase - Known x(0)
Signal: unrestricted
No Extrapolation
White noise
```

Figure 7.13.2b - Reconstruction Error $||\hat{x}_k - x_*||^2$ vs Iteration,

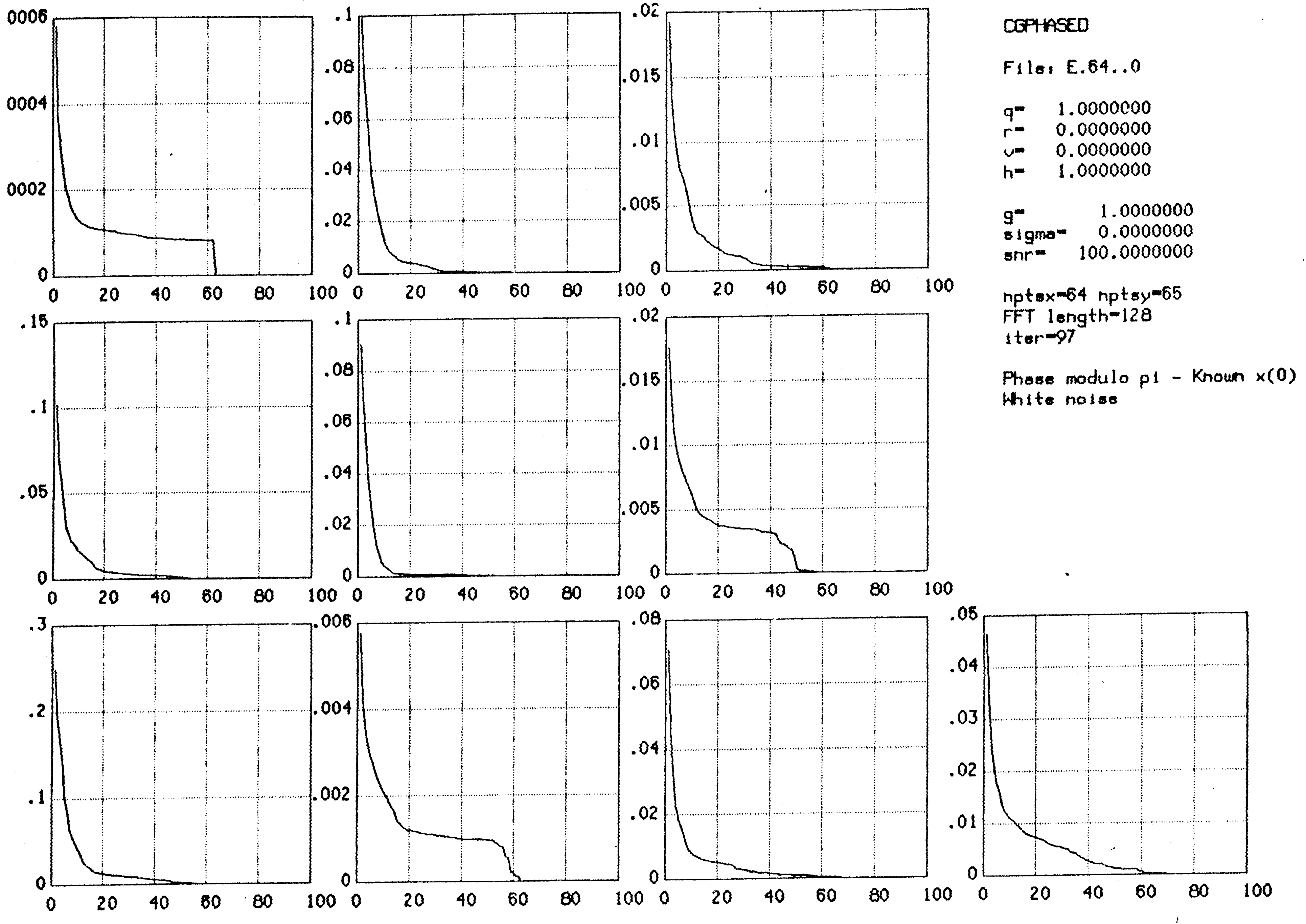
Primal Iterative Algorithm



```
CGPHASED

File: E.64..0

q=    1.0000000
r=    0.0000000
v=    0.0000000
h=    1.0000000

g=        1.0000000
sigma=    0.0000000
shr=    100.0000000

nptsx=64 nptsy=65
FFT length=128
iter=97

Phase modulo pi - Known x(0)
White noise
```

Figure 7.13.2c - Reconstruction Error $||\hat{x}_k - x_*||^2$ vs Iteration,

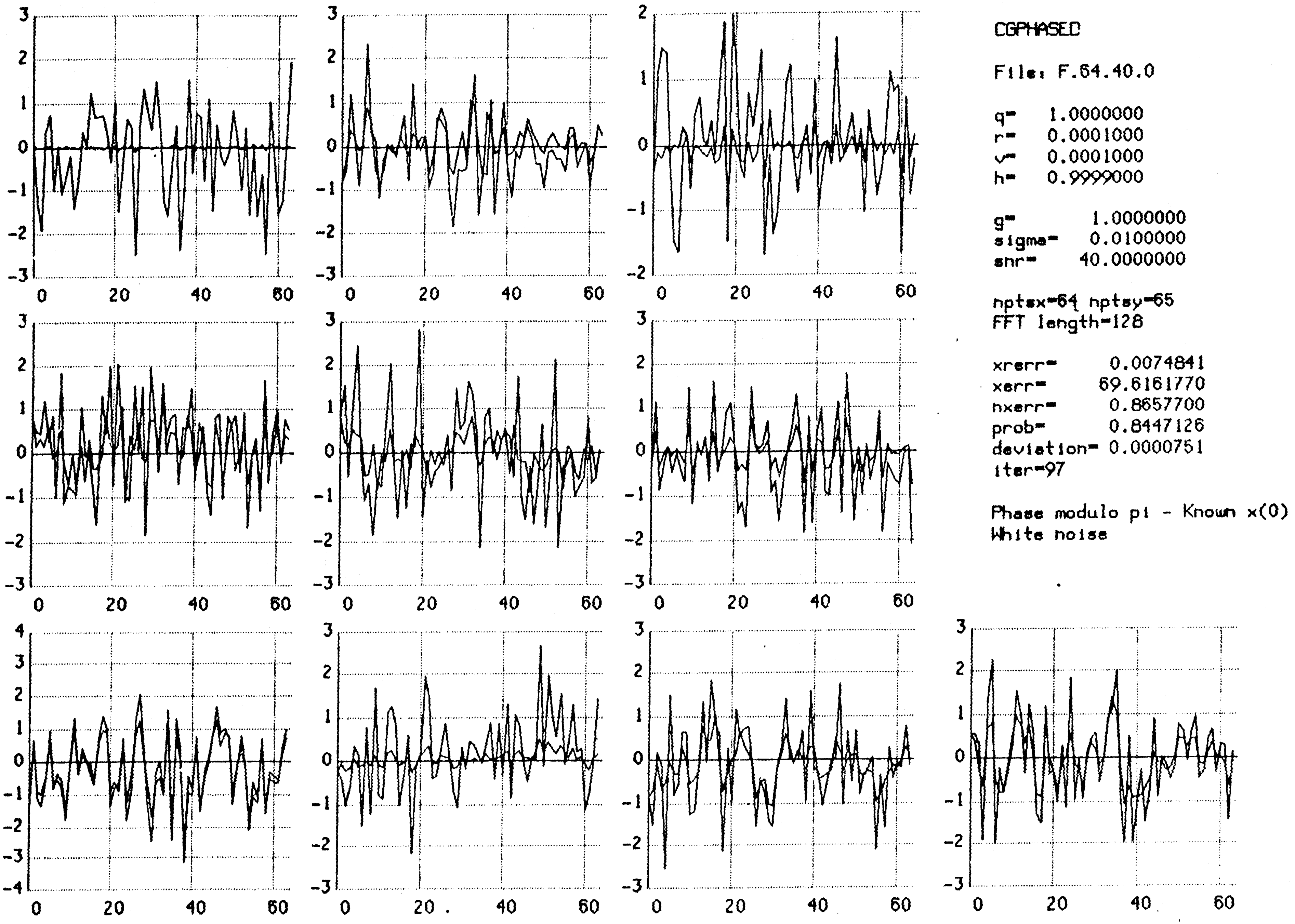
Conjugate Gradient



```
CGPHASED

File: E.64..0

q=   1.0000000
r=   0.0000000
v=   0.0000000
h=   1.0000000

g=        1.0000000
sigma=    0.0000000
snr=    100.0000000

nptsx=64 nptsy=65
FFT length=128
iter=97

Phase modulo pi - Known x(0)
White noise
```

Figure 7.13.2d - Objective Function $||\hat{y}_k - \hat{x}_k||^2$ vs. Iteration,

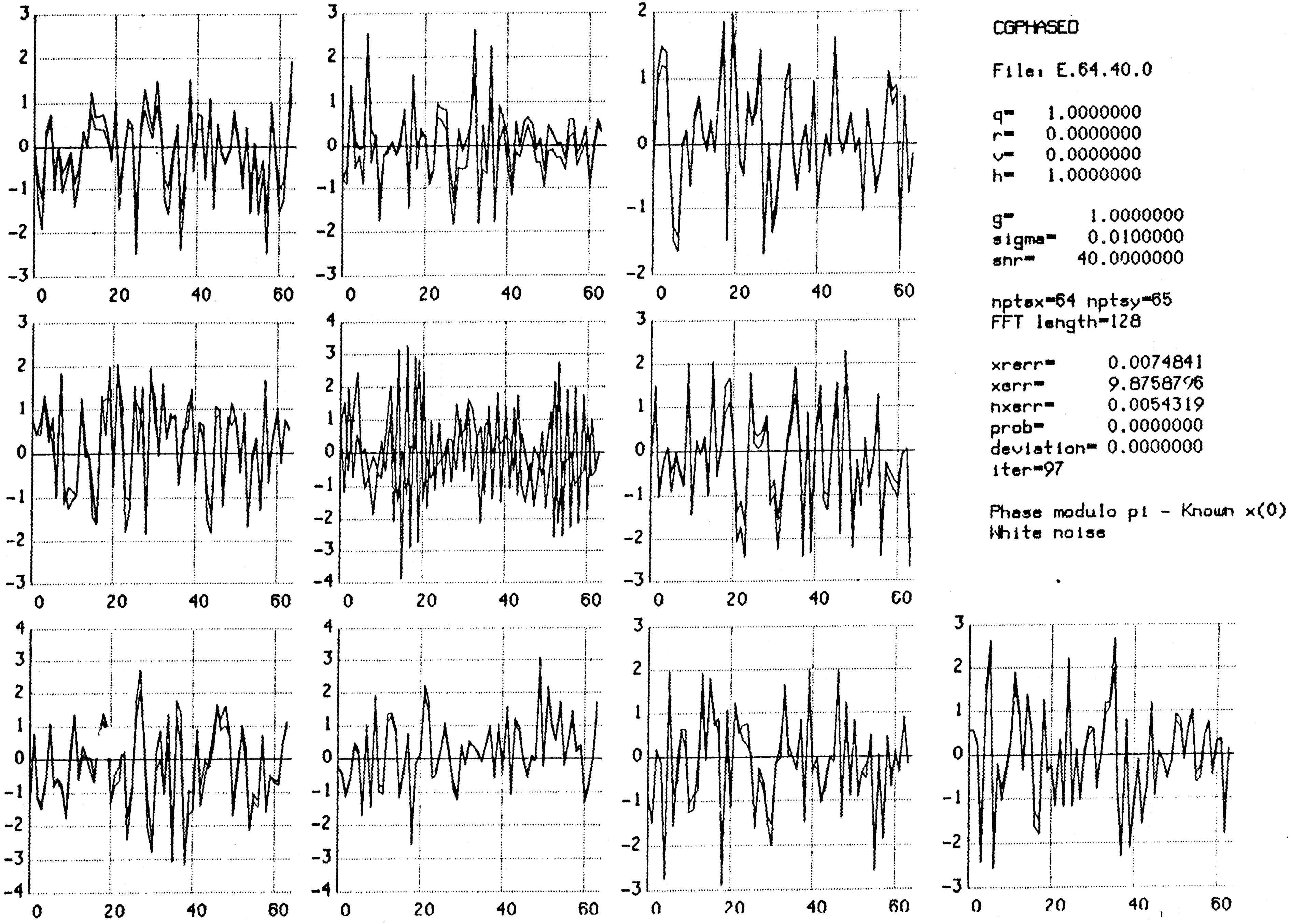
Conjugate Gradient



```
CGPHASED

File: F.64.40.0

q=    1.0000000
r=    0.0001000
v=    0.0001000
h=    0.9999000

g=          1.0000000
sigma=     0.0100000
shr=      40.0000000

nptsx=64 nptsy=65
FFT length=128

xrerr=      0.0074841
xerr=      69.6161770
nxerr=      0.8657700
prob=       0.8447126
deviation= 0.0000751
iter=97

Phase modulo pi - Known x(0)
White noise
```

Figure 7.13.3a - 10 Reconstructed and Original Sequences,

Conjugate Gradient, Bayesian, SNR = 40dB, 64 points, 128 point FFT



```
CGPHASED

File: E.64.40.0

q=     1.0000000
r=     0.0000000
v=     0.0000000
h=     1.0000000

g=          1.0000000
sigma=      0.0100000
snr=       40.0000000

nptsx=64 nptsy=65
FFT length=128

xrerr=       0.0074841
xerr=        9.8758796
nxerr=       0.0054319
prob=        0.0000000
deviation=   0.0000000
iter=97

Phase modulo pi - Known x(0)
White noise
```

Figure 7.13.3b - 10 Reconstructed and Original Sequences,

Coniugate Gradient, Fisher, SNR=40dB, 64 points, 128 point FFT

Section C - Convex Constraint Sets

14. Reconstruction of Finite Signal From Noisy Phase Modulo 2π

When the constraint sets are convex, but not linear varieties, then the four MCEM and MAP algorithms will give different results. The chief example we will consider involving convex constraint sets is reconstructing a finite length signal from knowledge of its phase modulo 2π. The examples in section 13 indicate that when the phase is only known modulo π, reconstruction of 64 point sequences becomes difficult even at signal-to-noise ratios of 40dB. With additional information, however, it is often possible to improve the reconstructions.

We first consider the case where each output spectrum sample $Y(\omega_i)$ is known to lie on an infinitesimally thin strip in the complex plane radiating outward from the origin at angle ϕ_i. The signal $x(0), \ldots, x(p-1)$ is assumed to be finite, and the sample $x(0)$ is assumed to be known.

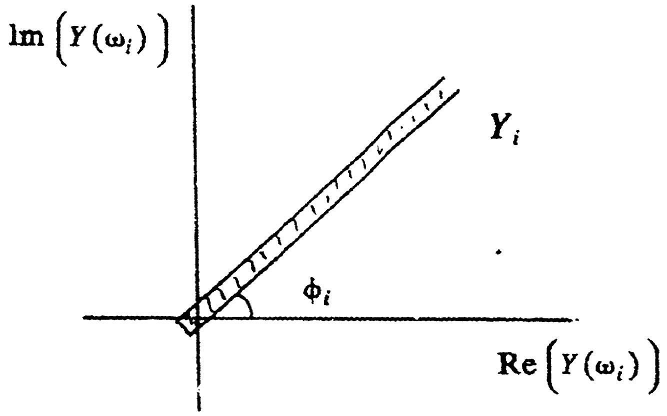

$\mathrm{Im}\left(Y(\omega_i)\right)$
Y_i
ϕ_i
$\mathrm{Re}\left(Y(\omega_i)\right)$

Figure 7.13.1 - Phase Modulo 2π Constraint Set

This type of output constraint set was used in the examples in chapter 5 and 6. Because the signal constraint space is still defined by linear equalities, our four iterative algorithms only give two distinct methods for solving the problem: MCEM/XMAP and YMAP/XYMAP. Substituting the constraint set in the above figure into our iterative algorithms in table 7.4.1 and using the formula in chapter 5 for the expectation of a Gaussian random variable on a thin strip, we get the following algorithm:

MCEM/XMAP Iterative Reconstruction

For $k=0,1,\cdots$

$$\hat{x}_{k+1}(n) = \begin{cases} x(0) & \text{for } n=0 \\ \dfrac{q(n)}{q(n)+r}\,\hat{y}_k(n) & \text{for } n=1,\ldots,p-1 \\ 0 & \text{else} \end{cases} \tag{7.14.1}$$

$$\frac{1}{\sqrt{N}}\hat{Y}_{k+1}(\omega_i) = \left[\hat{\chi}_i + \sqrt{r}\,\frac{\exp\left(-\frac{1}{2r}\hat{\chi}_i^2\right)}{\sqrt{2\pi}\left(\frac{1}{2}+\mathrm{erf}\left(\frac{\hat{\chi}_i}{\sqrt{r}}\right)\right)} \right] e^{j\phi_i}$$

$$\text{where: } \hat{\chi}_i = \frac{1}{\sqrt{N}} \mathrm{Re}\left(X(\omega_i) e^{-j\phi_i}\right)$$

$$\mathrm{erf}(v) = \frac{1}{\sqrt{2\pi}} \int_0^v \exp(-\tfrac{1}{2}x^2)\, dx$$

and:

YMAP/XYMAP Iterative Algorithm

For $k = 0, 1, \cdots$

$$\hat{x}_{k+1}(n) = \begin{cases} x(0) & \text{for } n = 0 \\ \dfrac{q(n)}{q(n)+r}\, \hat{y}_k(n) & \text{for } n = 1, \ldots, p-1 \\ 0 & \text{else} \end{cases} \qquad (7.14.2)$$

$$\frac{1}{\sqrt{N}} \hat{Y}_{k+1}(\omega_i) = \mathrm{Max}\left(\hat{\chi}_i, 0\right) e^{j\phi_i}$$

In both algorithms, we start with an estimate of the output spectrum. Inverse Fourier transform, then impose the known time domain constraints, setting the known sample to its correct value and truncating the signal to the correct length. The output is then reestimated in the YMAP/XYMAP algorithm by Fourier transforming the signal, then projecting each spectral sample onto the correct phase angle. The MCEM/XMAP algorithm is similar, but it adds an extra correction factor which depends on the magnitude of the projection $\hat{\chi}_i$ relative to the standard deviation of the noise. This increases the energy in the output estimate. Figure 7.14.2 explains the origin of this correction term by showing the conditional probability of $Y(\omega_i)$ along the Y_i constraint strip.

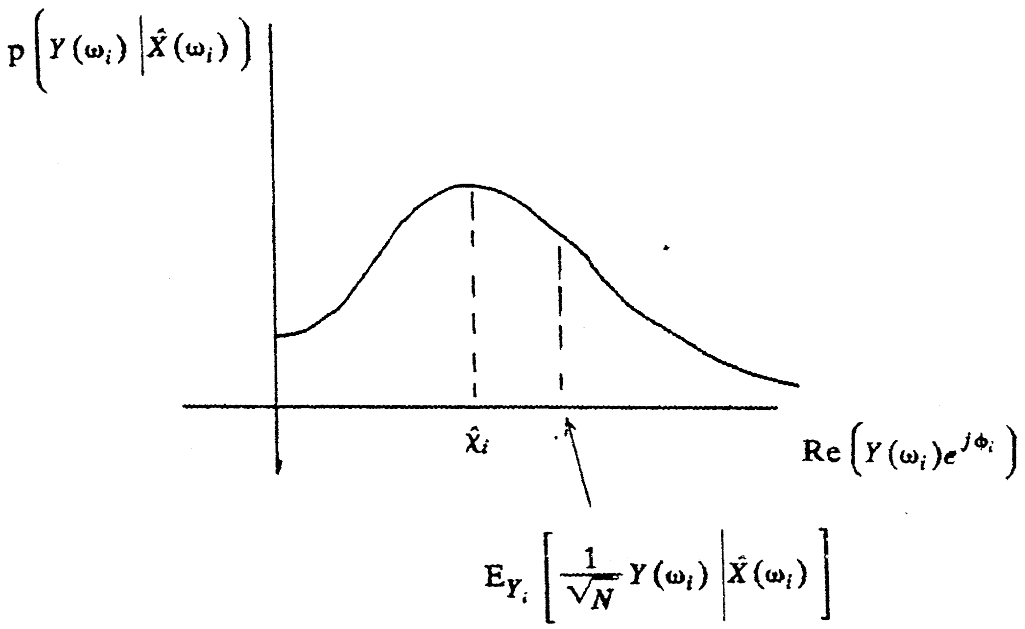


Figure 7.14.2 - Conditional Output Density

This conditional density is a truncated Gaussian. YMAP/XYMAP estimates $Y(\omega_i)$ by choosing the mode of this density, $\hat{\chi}_i$. MCEM/XMAP chooses the mean of the density, which will be located at a larger value of $Y(\omega_i)$. Each iteration of either algorithm strictly decreases the cross-entropy at each stage, and strictly increases the corresponding likelihood function. Geometric convergence of the Bayesian algorithms $(q(n)<\infty)$ is guaranteed to the unique global optimizing solution at a rate $v_x = v_y = \max_n \frac{q(n)}{q(n)+r}$. The Fisher algorithms $(q(n)=\infty)$ are guaranteed to be bounded and converge to a global optimizing solution if and only if a global optimizing solution exists. Note that this does not require that a finite signal exists which has the given phase; all it requires is that there be a finite length signal and an output sequence with the correct phase which come as close to each other as possible. Because the constraint sets are simplexes defined by linear inequalities, YMAP/XYMAP is guaranteed to have a minimizing solution, and thus will always converge (see Goldstein [24] or Künzi and Krelle. [25])

It is interesting to compare these algorithms with the Hayes, Lim, Oppenheim, Quatieri algorithm, which was derived by a more intuitive and ad hoc approach to the problem. We can state their algorithm in the following form:

Hayes, Lim, Oppenheim, Quatieri Phase-Only Reconstruction Modulo 2π

Guess $\hat{Y}_0(\omega_i)$

For $k=0,1,\cdots$

$$\hat{x}_{k+1}(n) = \begin{cases} x(0) & \text{if } n=0 \\ \hat{y}_k(n) & \text{if } n=1,\ldots,p-1 \\ 0 & \text{else} \end{cases} \tag{7.14.3}$$

$$\hat{Y}_{k+1}(\omega_i) = \left|\hat{X}_{k+1}(\omega_i)\right| e^{j\phi_i}$$

Starting with an initial output estimate, truncate the sequence to the correct length and set the known signal value to its correct value. Now take the DFT, retain the magnitude of the signal transform, but set the output transform phase to the known value. Now the output sequence is no longer finite length, so we iterate, truncating the signal and forcing the correct phase until it converges. Tom, Hayes, Quatieri, McClellan [21] used the fact that $||\hat{y}_k - \hat{x}_k||^2$ is strictly decreasing in this algorithm to prove convergence if a finite length signal exists with the given phase. Figure 7.14.3 compares the output spectrum estimates generated by our iterative algorithms.

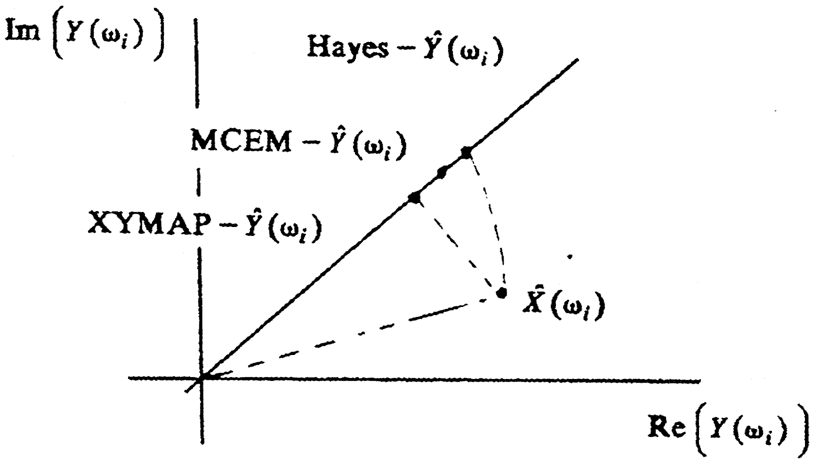

$\mathrm{Im}\left(Y(\omega_i)\right)$
$\text{Hayes} - \hat{Y}(\omega_i)$
$\text{MCEM} - \hat{Y}(\omega_i)$
$\text{XYMAP} - \hat{Y}(\omega_i)$
$\hat{X}(\omega_i)$
$\mathrm{Re}\left(Y(\omega_i)\right)$

Figure 7.14.3 - Comparison of Output Spectrum Estim:tes

Given noise-free phase, all of our methods will converge to the same estimate (with $r=0$, MCEM/XMAP will be identical to YMAP/XYMAP.) We therefore tried comparing the algorithms on noisy sequences. The convergence rate for 64 point sequences was very slow, so we chose ten 32 point sequences instead; except for the slow convergence rate and increased ill-behavior, the algorithms act similarly for 64 point sequences. Figure 7.14.4 shows the reconstructions generated by our three Fisher algorithms ($q(n)=\infty$) with 64 noise samples added to the 32 point sequence, for an SNR=30dB. The reconstructed signals are shown, superimposed on the original, and also the reconstruction error $||\hat{x}_k - x_*||^2$ is graphed as a function of iteration number. We used a fixed over-relaxation procedure to accelerate convergence in these algorithms; no real attempt was made to optimize the convergence rate. It is significant, however, that while the conjugate gradient modulo π algorithm would converge in 31 iterations, these algorithms require several hundred iterations even with acceleration. YMAP/XYMAP performs the worst. Surprisingly, in 8 out of 10 sequences, Hayes' algorithm appears to give slightly better reconstructions than MCEM; the reconstruction error seems to be about 80% or so of MCEM. This is difficult to explain. Figure 7.14.5 repeats the exercise at 20dB for the three algorithms. Now the relative performance is reversed; MCEM/XMAP gives far better reconstructions than Hayes, with reconstruction errors 2 to 3 times lower. At 10 dB (figures not shown) no method works well. The Bayesian MCEM/XMAP and YMAP/XYMAP algorithms are not shown because at 30dB and below they filter out most of the signal in their attempt to filter out the noise.

To summarize, the method based on careful probabilistic analysis, MCEM/XMAP, is best when the reconstruction difficulty is dominated by the stochastic behavior of the noise. At higher SNR, the Hayes, *et al* approach appears to be slightly better and is simpler to boot. Beware, however, that applying the line search acceleration that they suggest will change the solution towards which their algorithm converges. (In fact, it pushes it closer to our XYMAP solution.)

MCEM/XMAP achieves its superior low SNR performance by exploiting the fact that the noise is known to be Gaussian. If the phase measurements have been distorted by quantization error, for example, rather than by Gaussian noise, then MCEM/XMAP performs little better than Hayes. Other assumptions could be tried for modeling flat quantization noise in our Gaussian setting. For example, we could try modeling the constraint set as a *wedge* radiating out from the origin at angle ϕ_i.

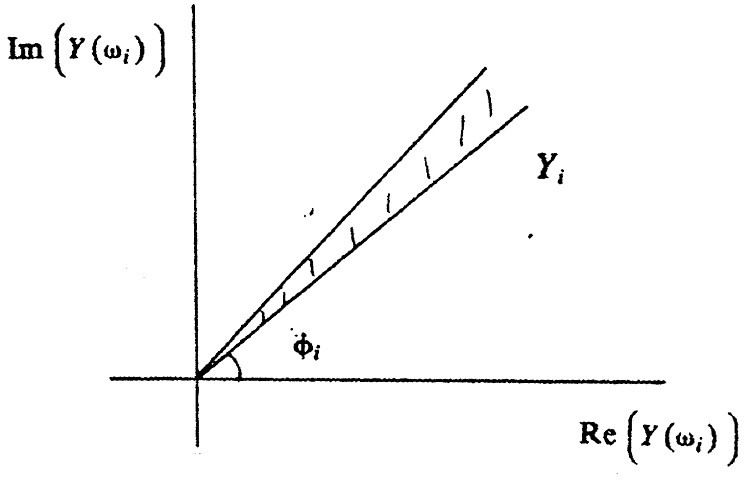

$\mathrm{Im}\left\{Y(\omega_i)\right\}$
Y_i
ϕ_i
$\mathrm{Re}\left\{Y(\omega_i)\right\}$

Figure 7.14.6 - Wedge Constraint Set

Since the width of the wedge is proportional to the distance from the origin, if we approximate the wedge as being very narrow, then the MCEM/XMAP estimate would be:

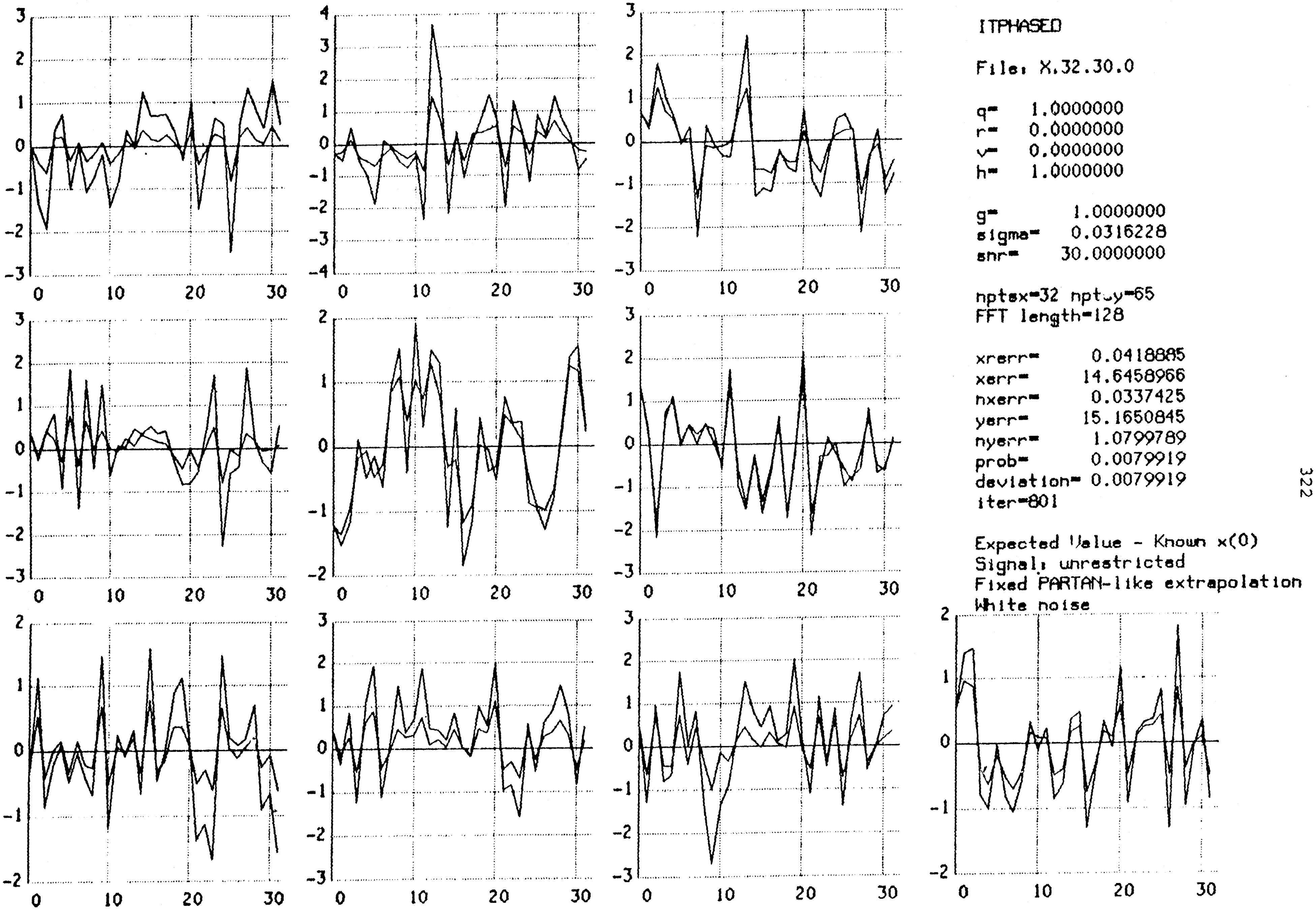
$$\hat{Y}_{k+1}(\omega_i) = \mathrm{E}_{Y_i}\left[Y(\omega_i)\,|\,\hat{X}(\omega_i)\right] \tag{7.14.4}$$

$$= \left[\hat{\chi}_i + \sqrt{r}\,\frac{1}{\hat{\chi}_i + \sqrt{r}\,\dfrac{\exp\left(-\dfrac{1}{2r}\hat{\chi}_i^2\right)}{\sqrt{2\pi}\left(\dfrac{1}{2} + \mathrm{erf}\left(\dfrac{\hat{\chi}_i}{\sqrt{r}}\right)\right)}}\right] e^{j\phi_i}$$

This calculation would bias the output spectrum estimate toward larger values than in our thin strip approximation. Experimenting with flat phase noise, however, has not shown any conclusive advantage to the scheme. It also has severe numerical difficulties.

The basic idea can be extended to problems in which we know both the phase of the spectrum and the phase of each signal point (i.e. the sign of each point.) Experiments with our algorithms has shown similar performance as in our example above;

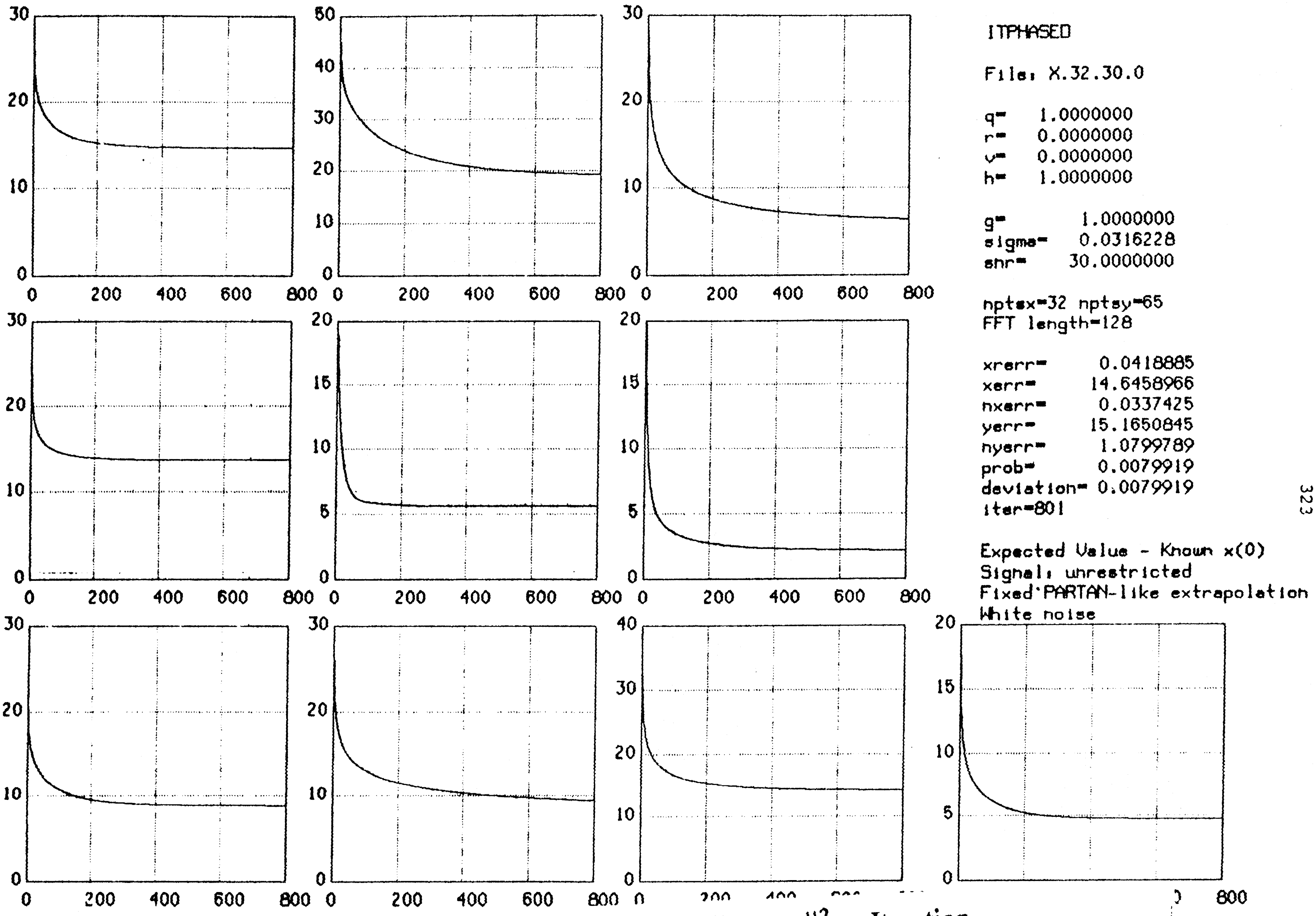
MCEM using expectations in both time and frequency domains works far better than anything else at 20dB, while using the phase substitution idea of the Hayes algorithm in both the time and frequency domains works somewhat better at 30dB.



```
ITPHASED

File: X.32.30.0

q=    1.0000000
r=    0.0000000
v=    0.0000000
h=    1.0000000

g=          1.0000000
sigma=      0.0316228
snr=       30.0000000

nptsx=32 nptуy=65
FFT length=128

xrerr=       0.0418885
xerr=       14.6458966
nxerr=       0.0337425
yerr=       15.1650845
nyerr=       1.0799789
prob=        0.0079919
deviation=   0.0079919
iter=801

Expected Value - Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.4a - 10 Reconstructed and Original Sequences,

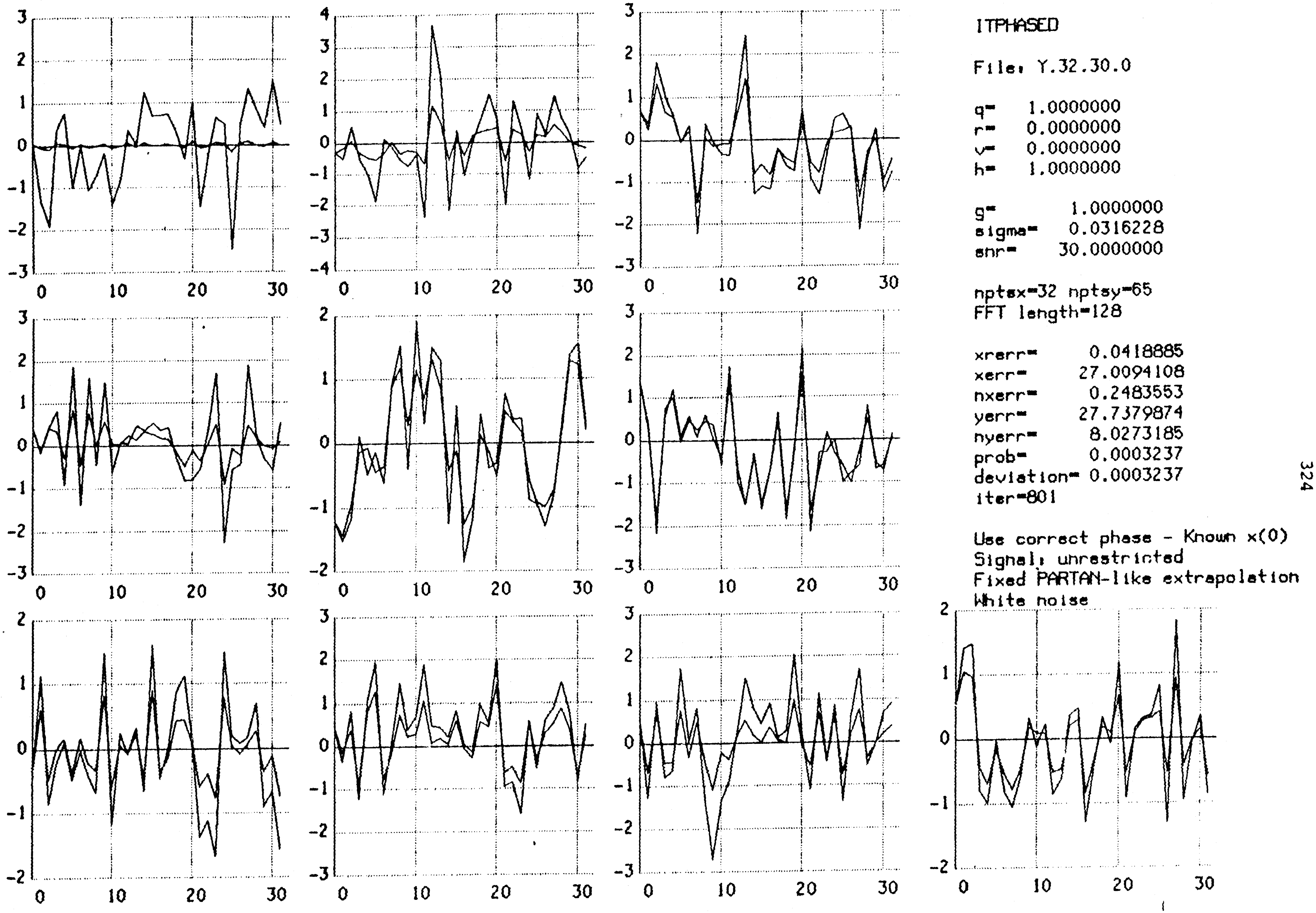
MCEM/XMAP, 30dB



```
ITPHASED

File: X.32.30.0

q=    1.0000000
r=    0.0000000
v=    0.0000000
h=    1.0000000

g=          1.0000000
sigma=     0.0316228
snr=     30.0000000

nptsx=32 nptsy=65
FFT length=128

xrerr=      0.0418885
xerr=      14.6458966
nxerr=      0.0337425
yerr=      15.1650845
nyerr=      1.0799789
prob=       0.0079919
deviation= 0.0079919
iter=801

Expected Value - Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.4b - Reconstruction Error $||\hat{x}_k - x_*||^2$ vs Iteration,

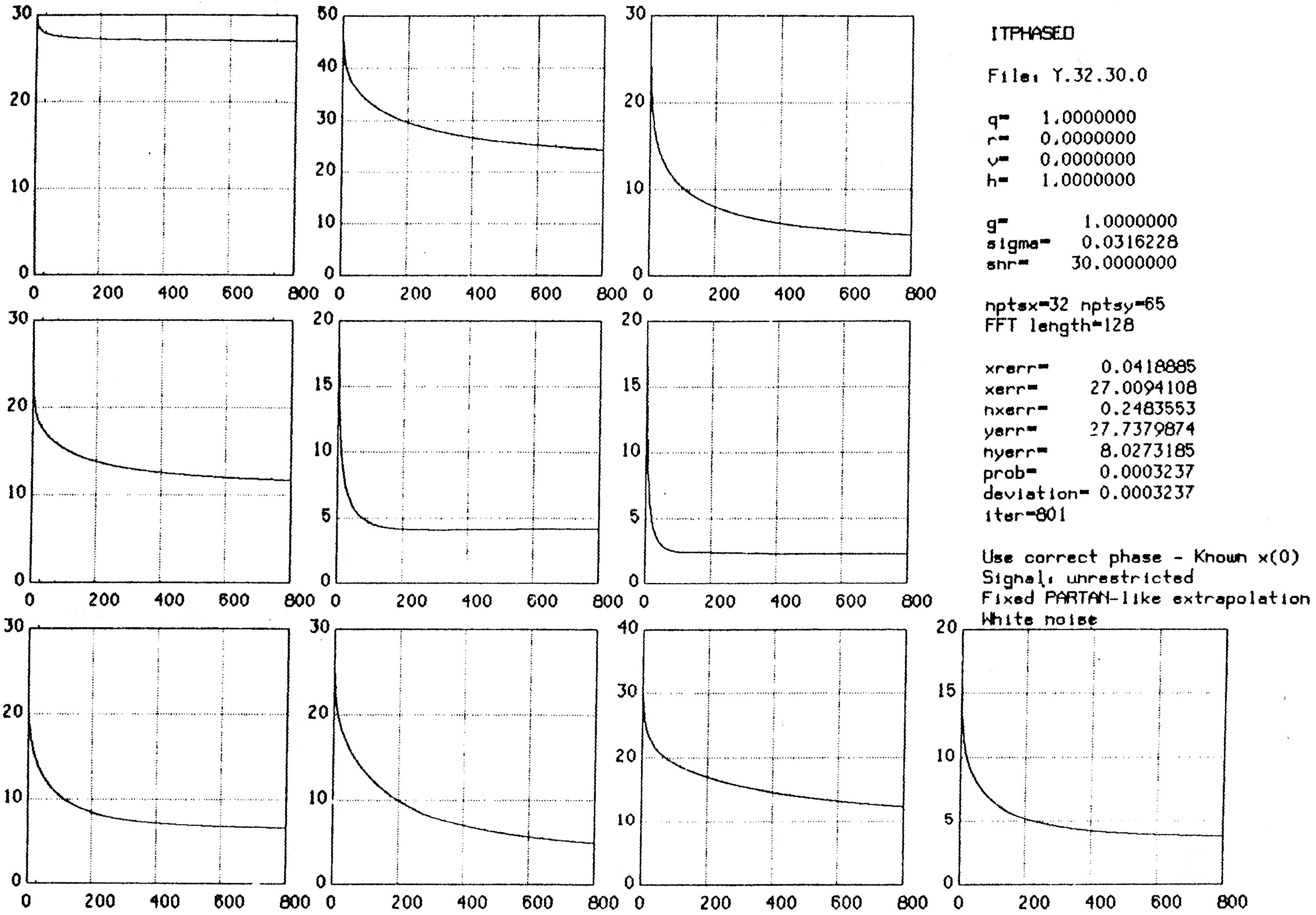
MCEM/XMAP, 30dB



```
ITPHASED

File: Y.32.30.0

q=    1.0000000
r=    0.0000000
v=    0.0000000
h=    1.0000000

g=        1.0000000
sigma=    0.0316228
snr=     30.0000000

nptsx=32 nptsy=65
FFT length=128

xrerr=      0.0418885
xerr=      27.0094108
nxerr=      0.2483553
yerr=      27.7379874
nyerr=      8.0273185
prob=       0.0003237
deviation=  0.0003237
iter=801

Use correct phase - Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.4c - 10 Reconstructed and Original Sequences,

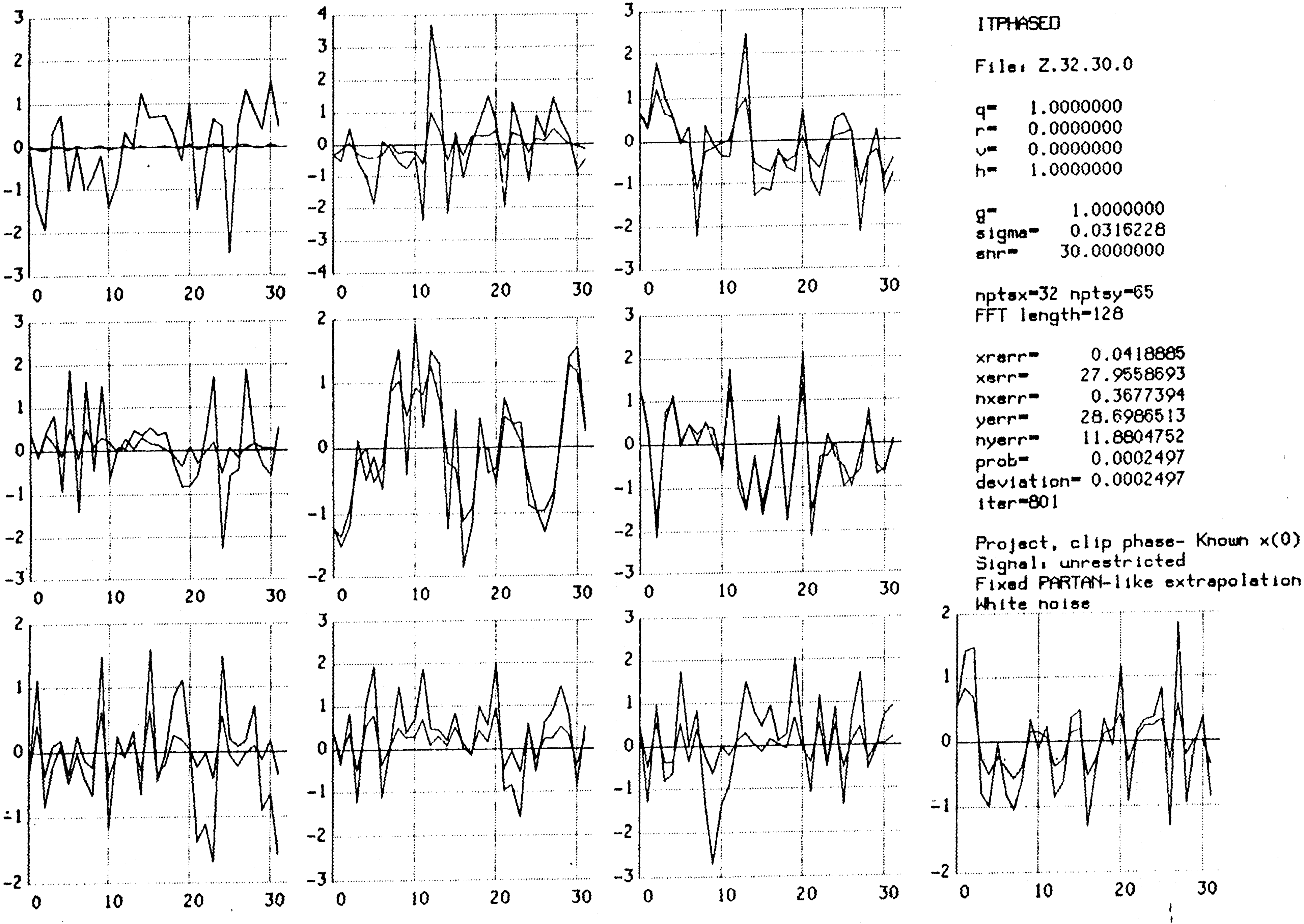
Force Phase, 30dB



```
ITPHASED

File: Y.32.30.0

q=    1.0000000
r=    0.0000000
v=    0.0000000
h=    1.0000000

g=          1.0000000
sigma=    0.0316228
snr=     30.0000000

nptsx=32 nptsy=65
FFT length=128

xrerr=       0.0418885
xerr=       27.0094108
nxerr=       0.2483553
yerr=       27.7379874
nyerr=       8.0273185
prob=        0.0003237
deviation=  0.0003237
iter=801

Use correct phase - Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.4d - Reconstruction Error $||\hat{x}_k - x_*||^2$ vs Iteration,

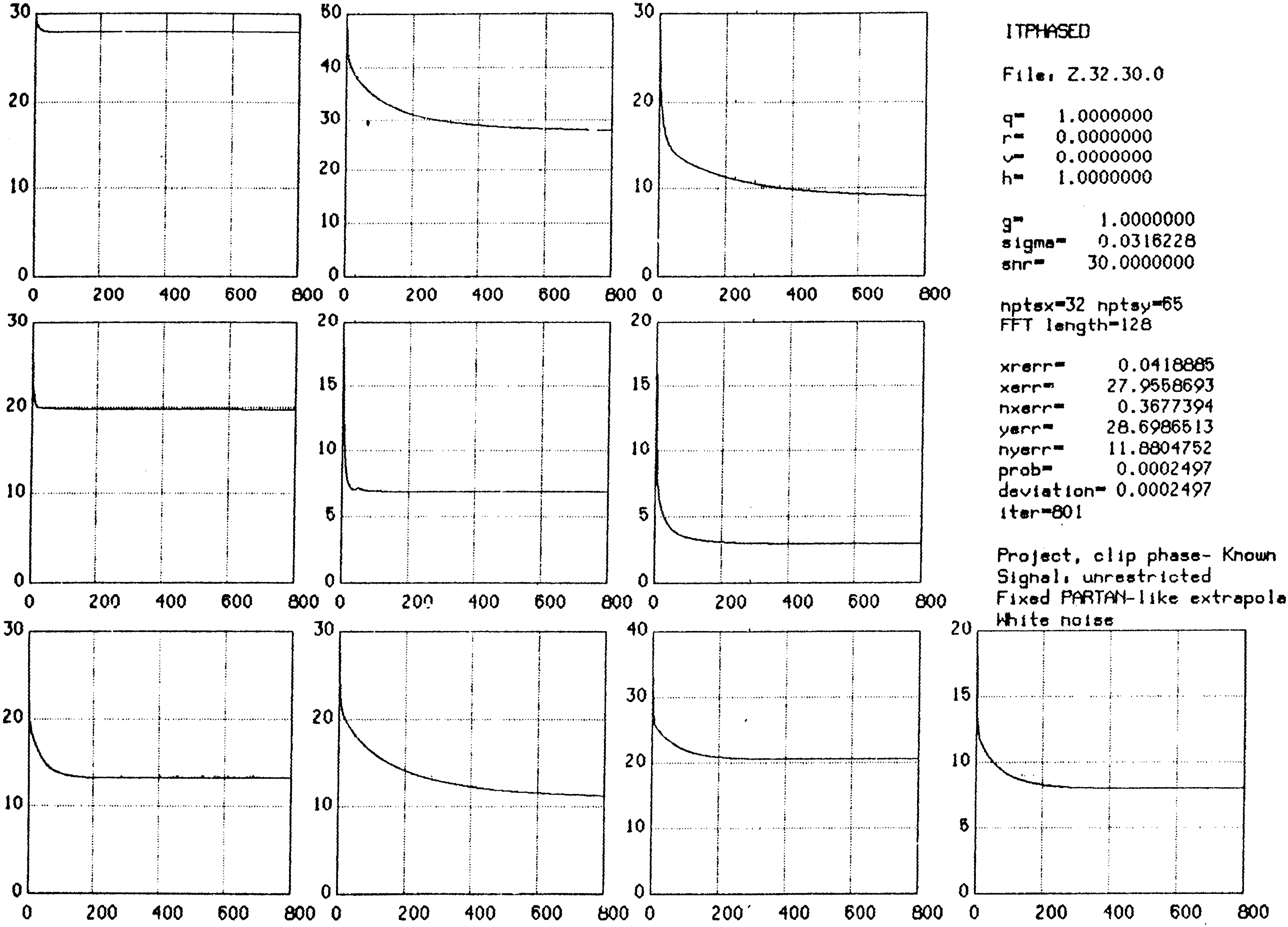
Force Pha 30dB



```
ITPHASED

File: Z.32.30.0

q=    1.0000000
r=    0.0000000
v=    0.0000000
h=    1.0000000

g=          1.0000000
sigma=      0.0316228
snr=       30.0000000

nptsx=32 nptsy=65
FFT length=128

xrerr=       0.0418885
xerr=       27.9558693
nxerr=       0.3677394
yerr=       28.6986513
nyerr=      11.8804752
prob=        0.0002497
deviation=   0.0002497
iter=801

Project, clip phase- Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.4e - 10 Reconstructed and Original Sequences,

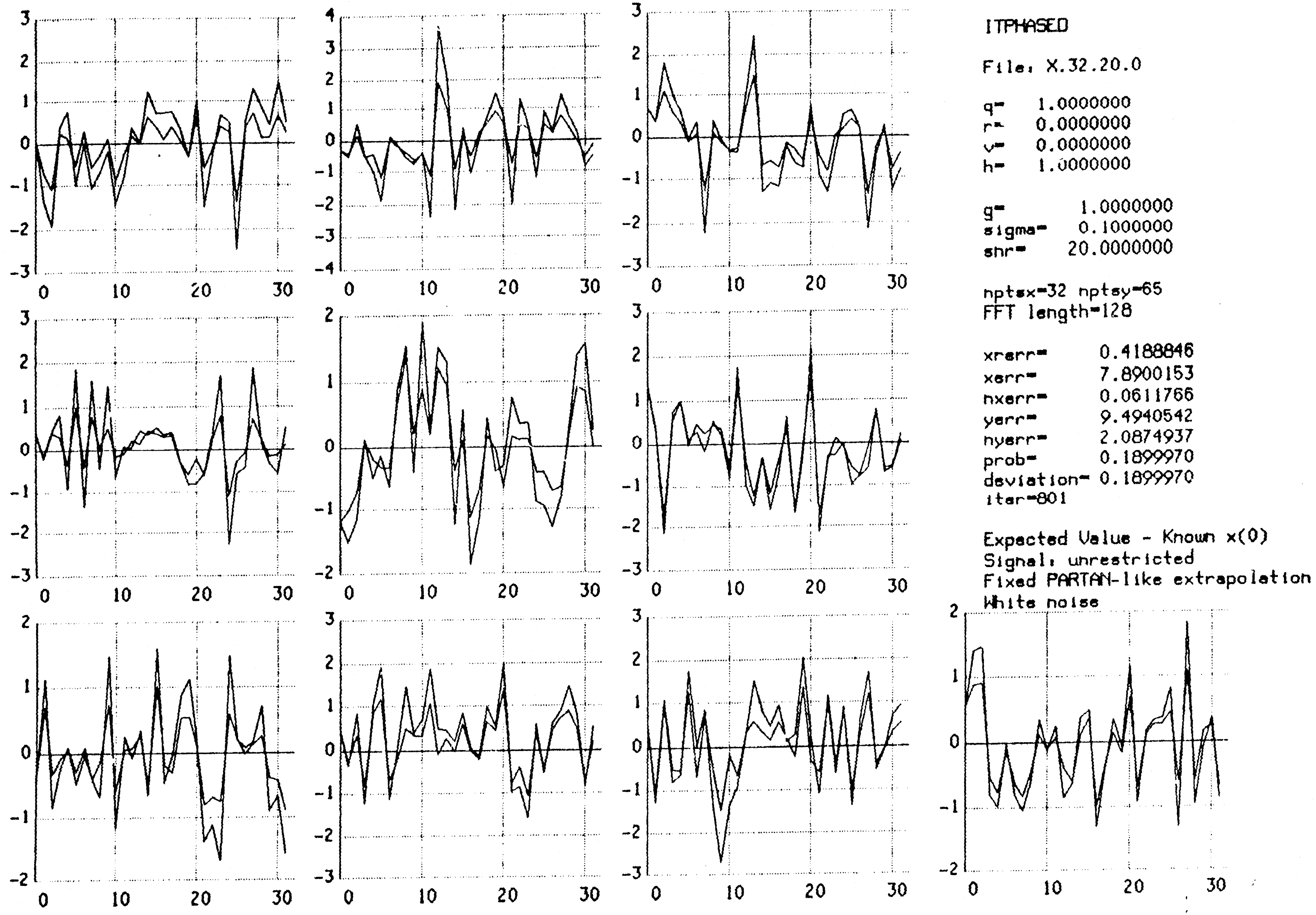
YMAP/XYMAP, 30dB



```
ITPHASED

File: Z.32.30.0

q=     1.0000000
r=     0.0000000
v=     0.0000000
h=     1.0000000

g=          1.0000000
sigma=      0.0316228
snr=       30.0000000

nptsx=32 nptsy=65
FFT length=128

xrerr=        0.0418885
xerr=        27.9558693
nxerr=        0.3677394
yerr=        28.6986513
nyerr=       11.8804752
prob=         0.0002497
deviation=    0.0002497
iter=801

Project, clip phase- Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.4f - Reconstruction Error $||\hat{x}_k - \underline{x}_*||^2$ vs Iteration,

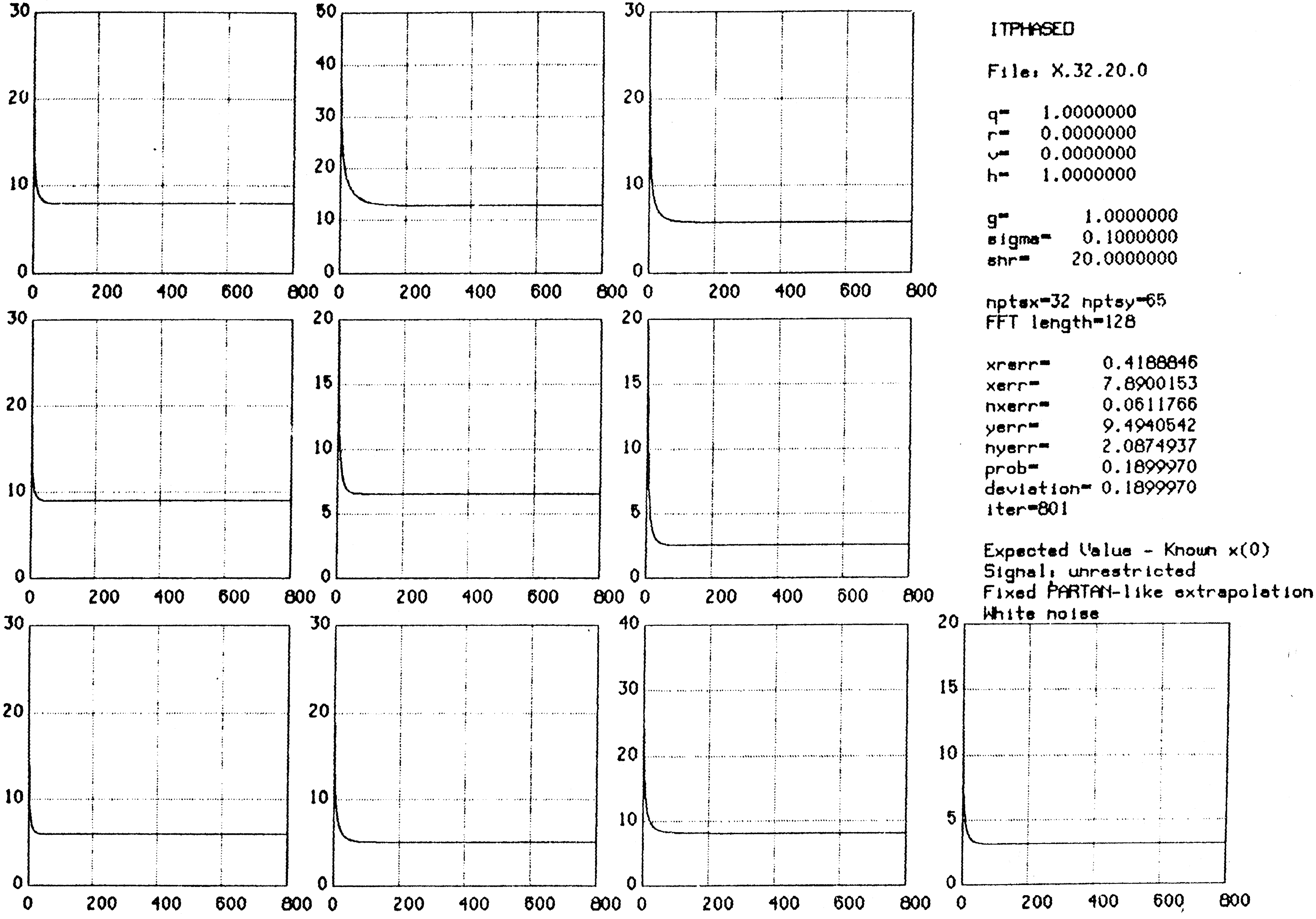
YMAP/XYMAP, 30dB



```
ITPHASED

File: X.32.20.0

q=    1.0000000
r=    0.0000000
v=    0.0000000
h=    1.0000000

g=          1.0000000
sigma=    0.1000000
shr=     20.0000000

nptsx=32 nptsy=65
FFT length=128

xrerr=       0.4188846
xerr=        7.8900153
nxerr=       0.0611766
yerr=        9.4940542
nyerr=       2.0874937
prob=        0.1899970
deviation= 0.1899970
iter=801

Expected Value - Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.5a - 10 Reconstructed and Original Sequences,

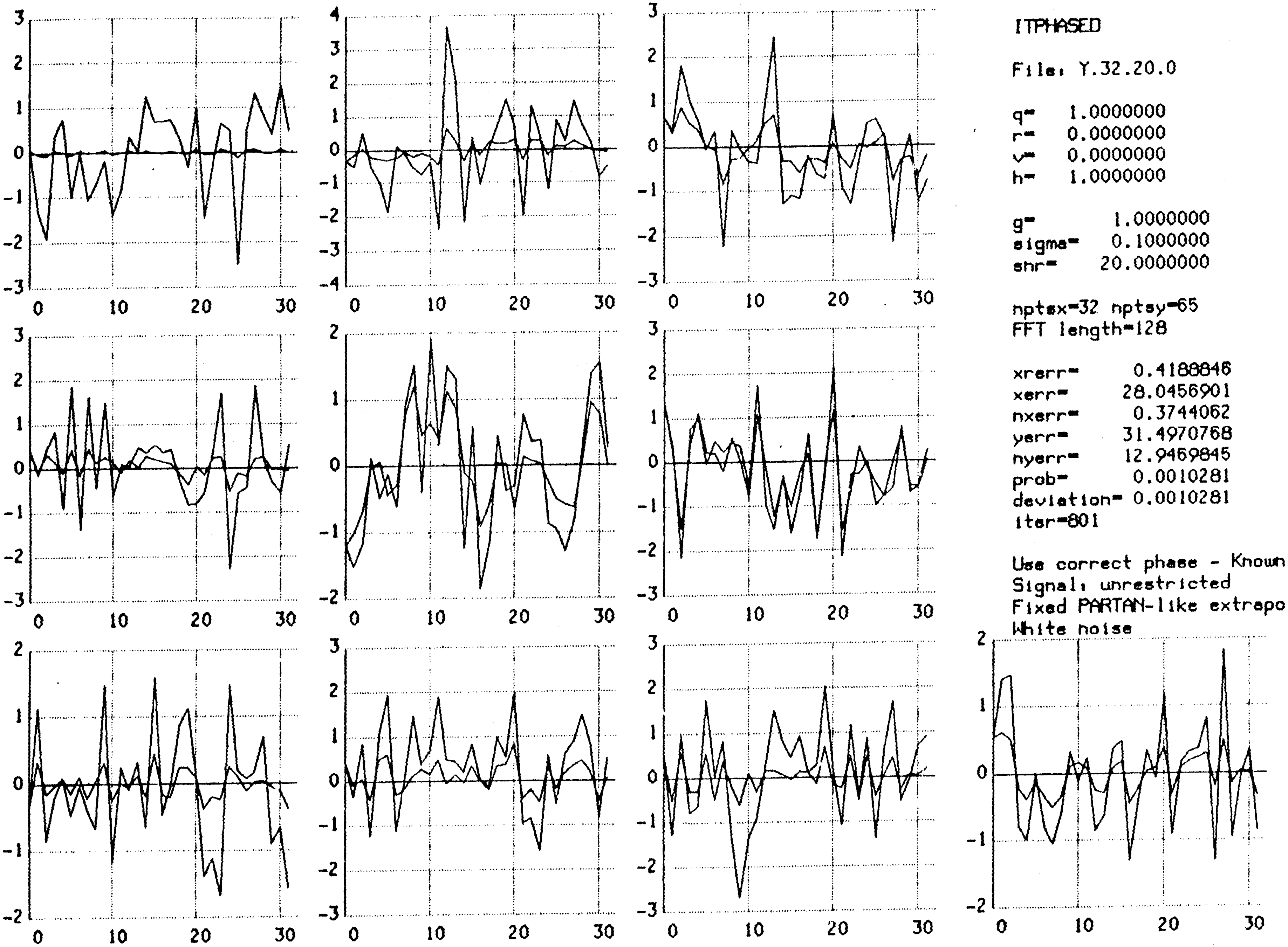
MCEM/XMAP, 20dB



```
ITPHASED

File: X.32.20.0

q=     1.0000000
r=     0.0000000
v=     0.0000000
h=     1.0000000

g=         1.0000000
sigma=     0.1000000
shr=      20.0000000

nptsx=32 nptsy=65
FFT length=128

xrerr=        0.4188846
xerr=         7.8900153
nxerr=        0.0611766
yerr=         9.4940542
nyerr=        2.0874937
prob=         0.1899970
deviation=  0.1899970
iter=801

Expected Value - Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.5b - Reconstruction Error $||\hat{x}_k - x_*||^2$ vs Iteration,

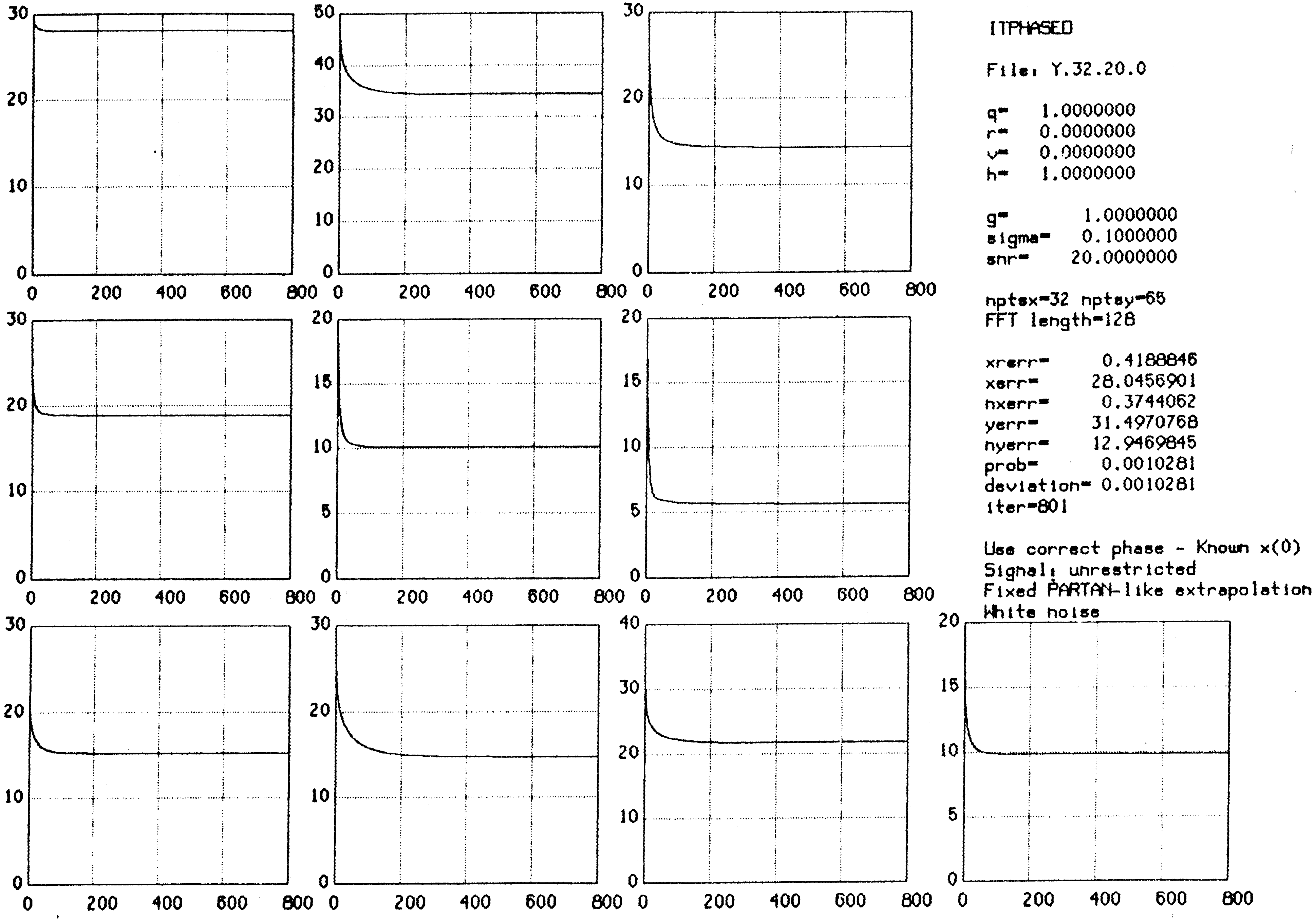
MCEM/XMAP 20dB



```
ITPHASED

File: Y.32.20.0

q=    1.0000000
r=    0.0000000
v=    0.0000000
h=    1.0000000

g=          1.0000000
sigma=      0.1000000
shr=       20.0000000

nptsx=32 nptsy=65
FFT length=128

xrerr=       0.4188846
xerr=       28.0456901
nxerr=       0.3744062
yerr=       31.4970768
nyerr=      12.9469845
prob=        0.0010281
deviation=  0.0010281
iter=801

Use correct phase - Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.5c - 10 Reconstructed and Original Sequences,

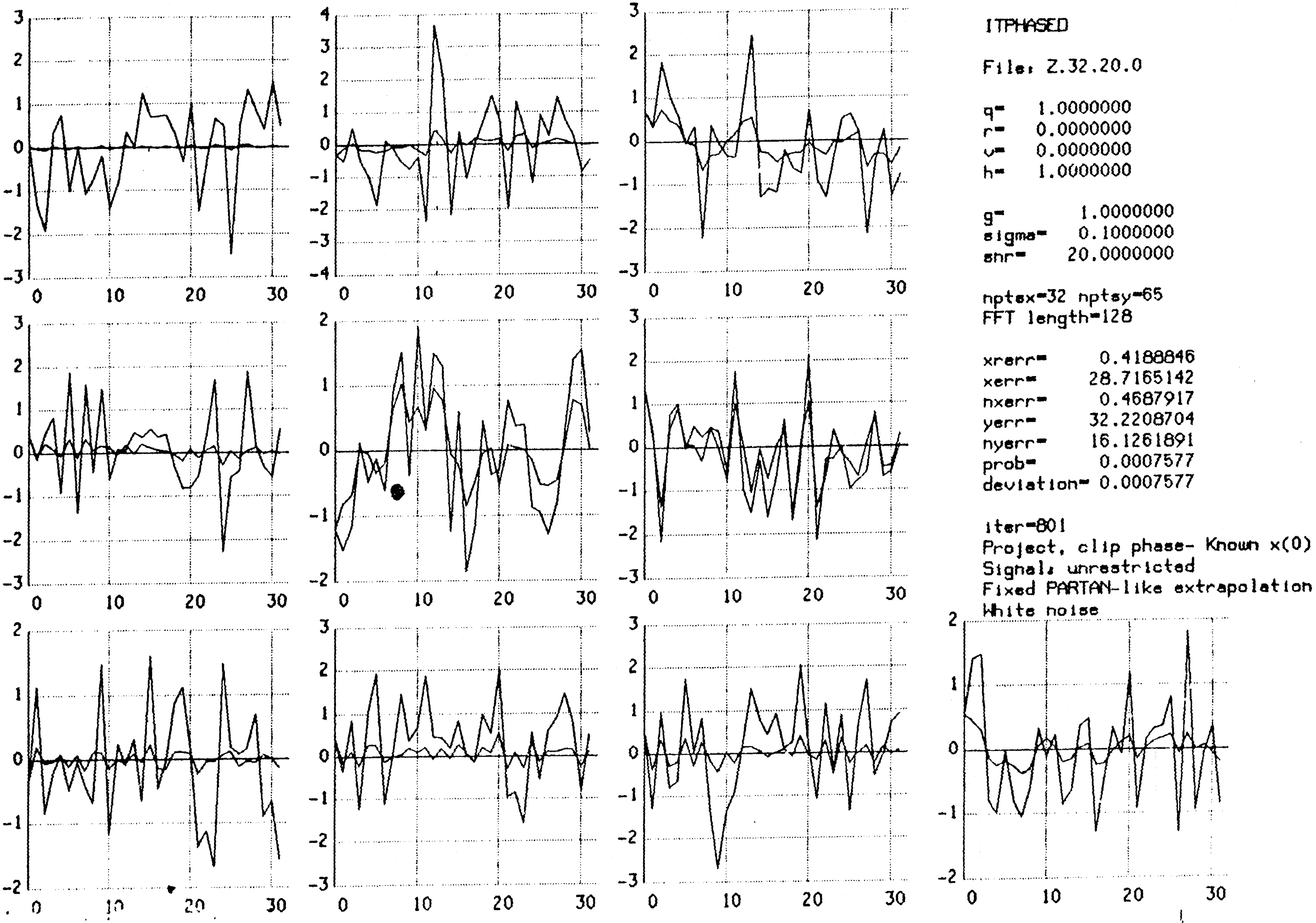
Force Phase, 20dB



```
ITPHASED

File: Y.32.20.0

q=     1.0000000
r=     0.0000000
v=     0.0000000
h=     1.0000000

g=          1.0000000
sigma=      0.1000000
snr=       20.0000000

nptsx=32 nptsy=65
FFT length=128

xrerr=       0.4188846
xerr=       28.0456901
nxerr=       0.3744062
yerr=       31.4970768
nyerr=      12.9469845
prob=        0.0010281
deviation=  0.0010281
iter=801

Use correct phase - Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.5d - Reconstruction Error $||\hat{x}_k - x_*||^2$ vs Iteration,

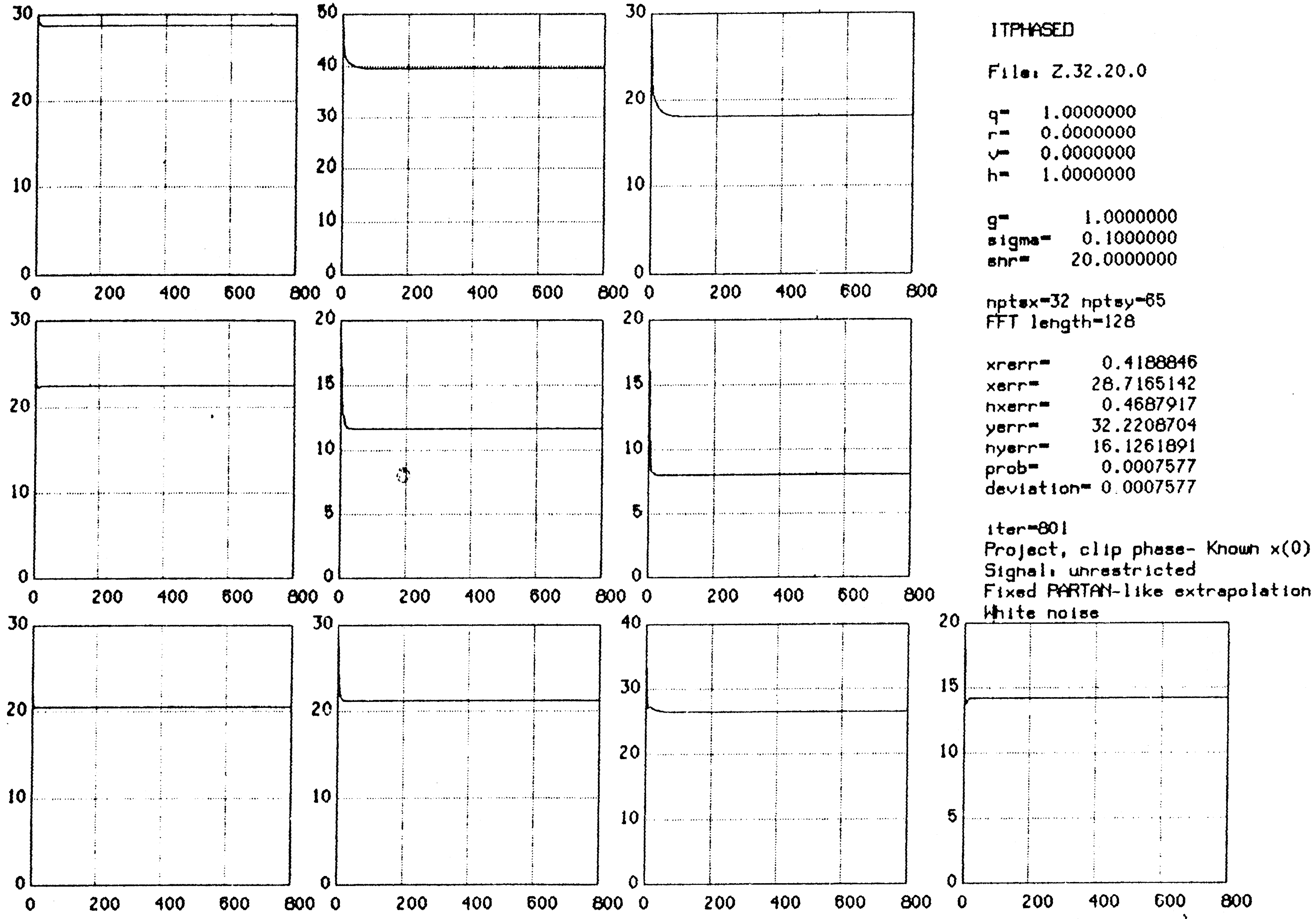
Force Phase 20dB



```
ITPHASED

File: Z.32.20.0

q=     1.0000000
r=     0.0000000
v=     0.0000000
h=     1.0000000

g=          1.0000000
sigma=      0.1000000
snr=       20.0000000

nptsx=32 nptsy=65
FFT length=128

xrerr=       0.4188846
xerr=       28.7165142
nxerr=       0.4687917
yerr=       32.2208704
nyerr=      16.1261891
prob=        0.0007577
deviation=  0.0007577

iter=801
Project, clip phase- Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.5e - 10 Reconstructed and Original Sequences,

YMAP/XYMAP, 20dB



```
ITPHASED

File: Z.32.20.0

q=    1.0000000
r=    0.0000000
v=    0.0000000
h=    1.0000000

g=          1.0000000
sigma=    0.1000000
snr=     20.0000000

nptsx=32 nptsy=65
FFT length=128

xrerr=       0.4188846
xerr=       28.7165142
nxerr=       0.4687917
yerr=       32.2208704
nyerr=      16.1261891
prob=        0.0007577
deviation= 0.0007577

iter=801
Project, clip phase- Known x(0)
Signal: unrestricted
Fixed PARTAN-like extrapolation
White noise
```

Figure 7.14.5f - Reconstruction Error $||\hat{x}_k - x_*||^2$ vs Iteration,

YMAP/XYMAP. 20dB

15. Multidimensional Finite Impulse Response Filter Design

Another interesting application of iterative signal reconstruction is to the design of one- or multi-dimensional Finite Impulse Response (FIR) filters meeting arbitrary time and frequency constraints. Problems like this arise when it is necessary to try to control the response of the filter over the entire frequency range of the filter, even in the transition bands, and also control the range of coefficient values which are permitted. For a very thorough discussion of this problem, see Rabiner and Gold [26]. The most general algorithm for FIR filter design, suggested by Rabiner [27,28], uses a linear programming algorithm, varying only one coefficient of the filter at a time in an attempt to drive the filter towards the desired form. Convergence is slow, but the algorithm terminates in a finite number of steps with the correct solution. The best known algorithm for one-dimensional Chebyshev FIR filter design is Remez exchange, as used in the Parks-McClellan algorithm. This approach builds the best high, low or bandpass filter design by using an iterative multipoint adjustment scheme to decrease the worst frequency domain errors in the filter. By exploiting the properties of finite one-dimensional polynomials, the resulting algorithm converges to an optimal solution in a finite number of iterations. Unfortunately, Remez exchange does not easily generalize to multiple dimensions, and cannot handle arbitrary time and frequency constraints.

The algorithm we present uses no intelligence in the FIR filter design process, and therefore is applicable to arbitrary time and frequency constraints in arbitrary dimensions. The penalty paid for this flexibility is that the filters are optimal in a least squares, not Chebyshev, sense, and the convergence rate is only geometric, so that some stopping criterion must be invoked.

Suppose we are given two types of constraints the filter must satisfy; each filter coefficient must fit within the set X_n, and each frequency sample of the DFT of the filter must fit within a set Y_i. The sets Y_i therefore define the desired frequency response together with the allowable tolerance we are willing to accept at each point. For a zero-phase FIR filter, for example, we may constrain the lower and upper bounds of both the coefficients and the magnitude of the frequency response, as in figure 7.15.1.

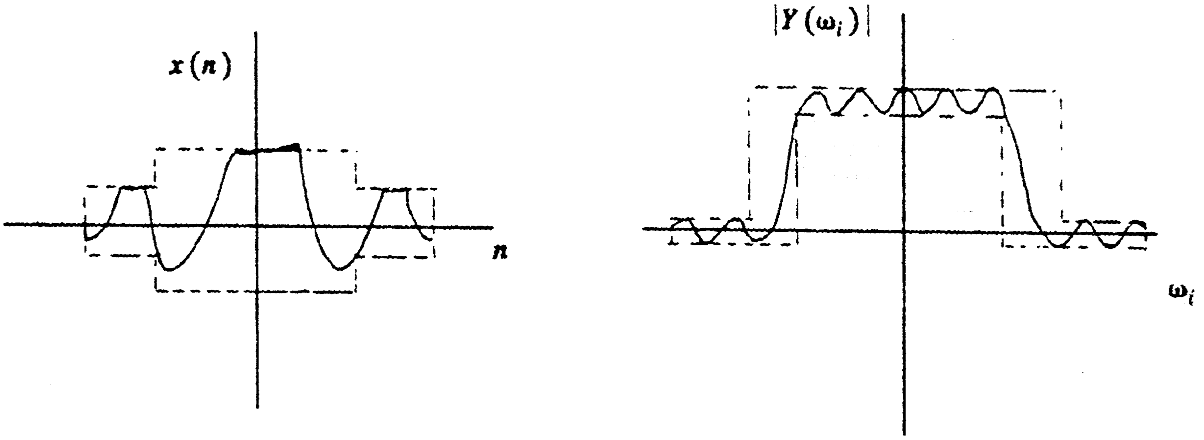


Figure 7.15.1 - Possible FIR Filter Constraints

Let x be an FIR filter meeting all the time domain constraints, and let y be a (non-finite length) filter meeting all the frequency constraints. Let us measure the "error" between the response of filters x and y by passing unit variance stationary white Gaussian noise $w(n)$ through both filters and measuring the mean square error between their outputs:

$$
\begin{aligned}
E_0 &= \mathrm{E}\left[\frac{1}{N} \sum_{n=0}^{N-1} \left(w(n)*y(n) - w(n)*x(n) \right)^2 \right] \qquad (7.15.1) \\
&= \mathrm{E}\left[\frac{1}{N} \sum_{n=0}^{N-1} \left(w(n)*(y(n)-x(n)) \right)^2 \right]
\end{aligned}
$$

$$= \sum_{n=0}^{N-1} \left(y(n) - x(n) \right)^2$$

$$= \|\underline{y} - \underline{x}\|^2$$

We now choose the filter $\underline{x}$ meeting the time domain constraints and the filter $\underline{y}$ meeting the frequency domain constraints which minimize this average error energy:

$$\hat{\underline{x}}, \hat{\underline{y}} - \min_{\underline{x} \in X, \underline{y} \in Y} \|\underline{y} - \underline{x}\|^2 \tag{7.15.2}$$

This, of course, is just the Fisher XYMAP objective function. We can therefore use the Fisher XYMAP algorithm to design the filter:

For $k = 0, 1, \cdots$

$$\begin{aligned} \hat{x}_{k+1}(n) &= \min_{x(n) \in X_n} \left(y(n) - x(n) \right)^2 \\ \hat{Y}_{k+1}(\omega_i) &- \min_{Y(\omega_i) \in Y_i} \left| Y(\omega_i) - \hat{X}_{k+1}(\omega_i) \right|^2 \end{aligned} \tag{7.15.3}$$

We simply alternate between clipping the filter $\hat{y}_k(n)$ to fit within the known constraints, giving the estimate $\hat{x}_{k+1}(n)$, then clipping $\hat{X}_{k+1}(\omega_i)$ to fit within the known frequency constraints, giving $\hat{Y}_{k+1}(\omega_i)$. Each iteration decreases the objective function (7.15.2) and if the constraint sets X_n and Y_i are convex, convergence to a global optimizing solution is guaranteed if such a solution exists. If all the sets Y_i are bounded, then Y will be compact, and thus convergence will be guaranteed.

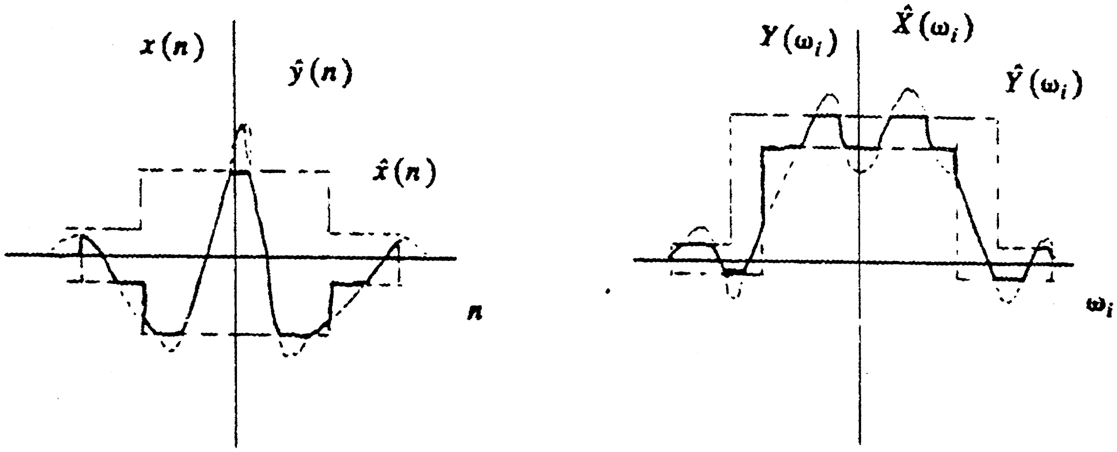
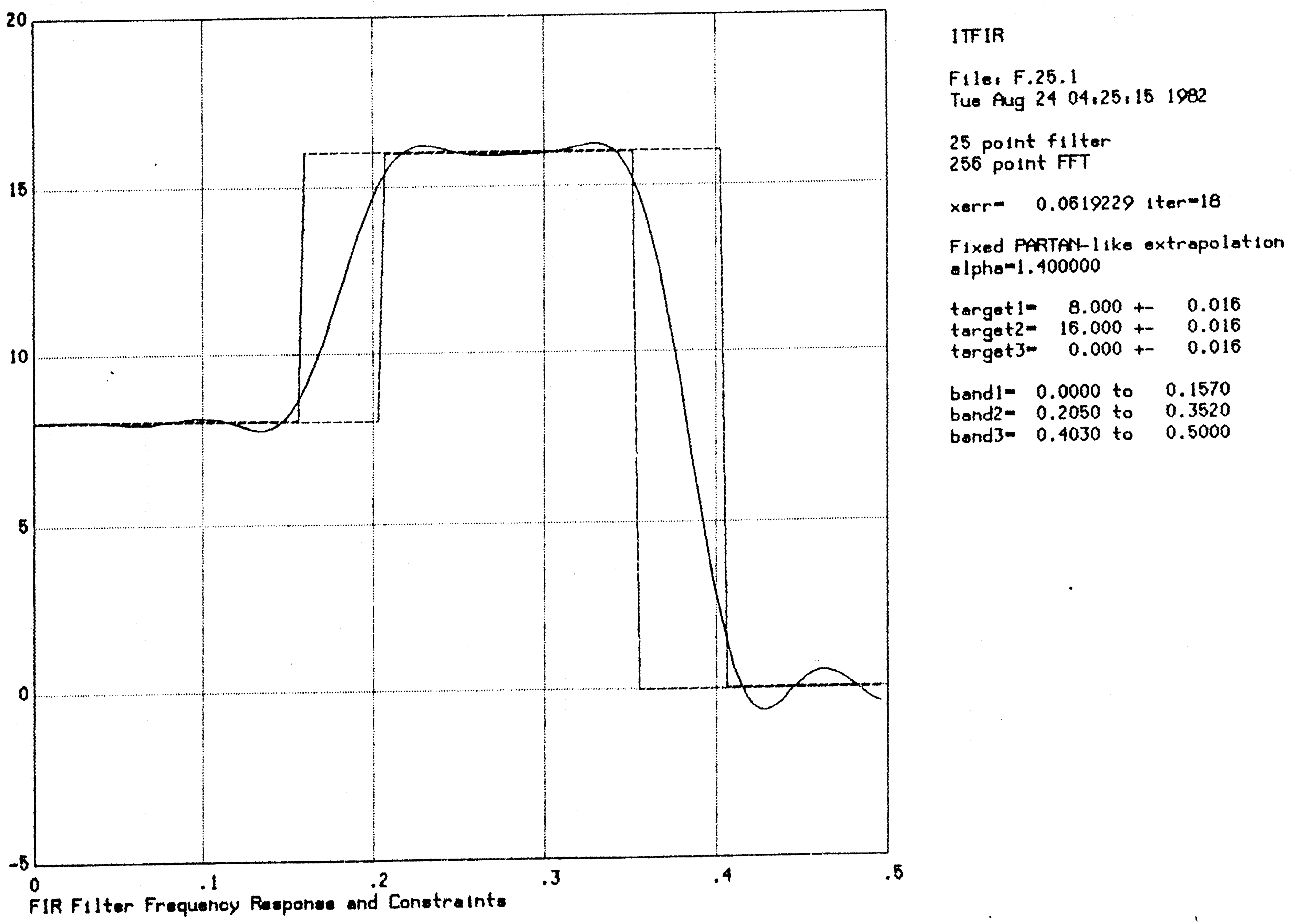


Figure 7.15.2 - Meat-Cleaver FIR Design

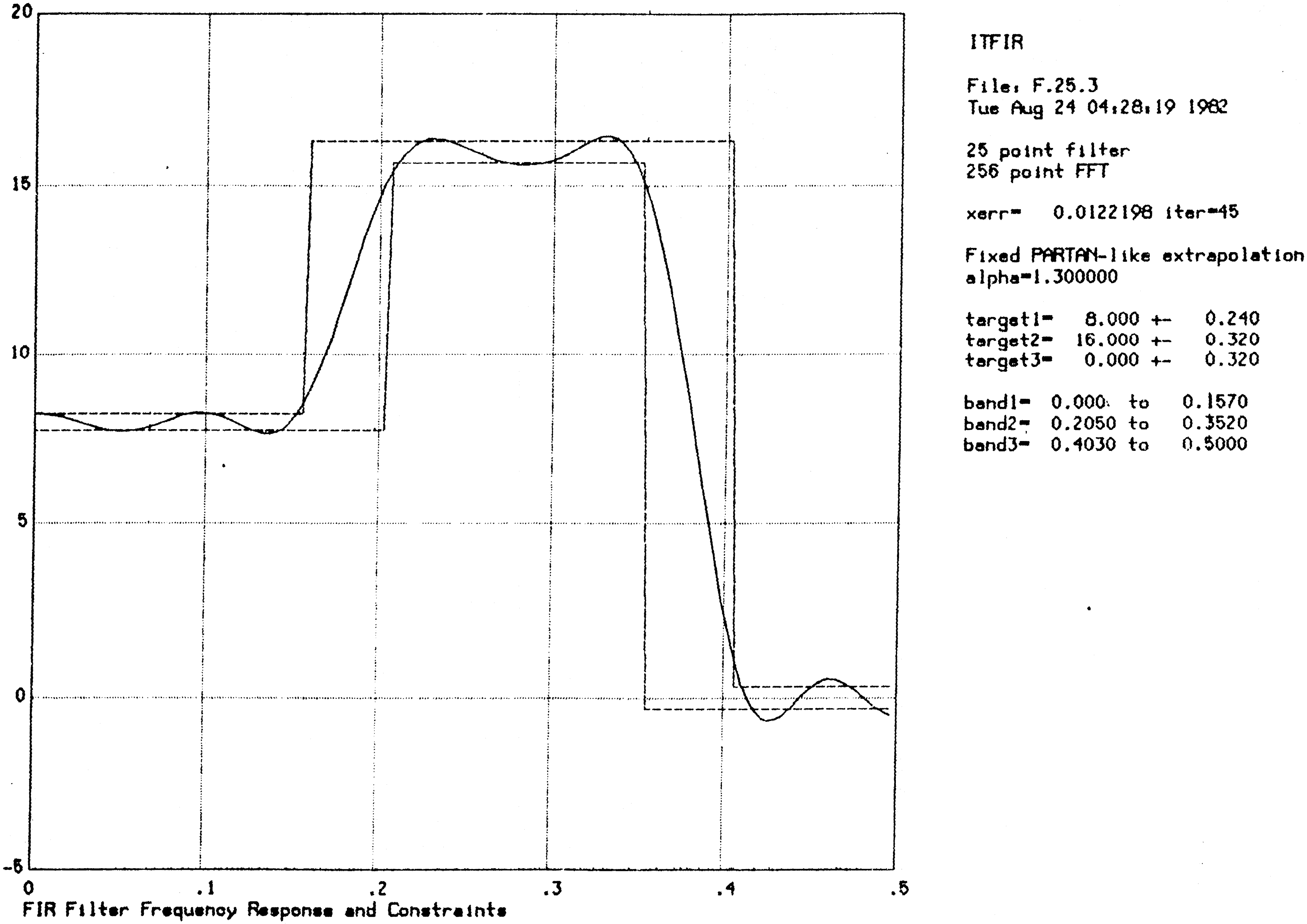
If a filter exists which satisfies both the time and frequency constraints, $\hat{x} = \hat{y}$, then the algorithm is guaranteed to find such a solution. Otherwise, it finds a pair of filters $(\hat{x}, \hat{y})$ such that $\hat{x}$ meets the time constraints and "comes as close as possible" to meeting the frequency constraints, while $\hat{y}$ meets the frequency constraints and "comes as close as possible" to meeting the time constraints. Since the algorithm uses no intelligence about FIR filter design, it will work on arbitrary dimensional filters meeting arbitrary convex constraints. Of course, the convergence rate may be sublinear, and an infinite number of iterations might be needed. A least squares criterion is also not always appropriate, since a Chebyshev design will yield less peak error. Unlike Remez exchange, the method does not suggest the tightest possible constraints that could be used. Finally, by relying on finite length FFT's, we are only constraining *samples* of the filter's response to fit inside the constraints. Peaks of the filter response could occur between samples, and thereby violate the constraints. Using very dense FFT's will solve this problem, but especially in multiple dimensions, large FFT's can be prohibitively costly. On the other hand, the advantage of our algorithm is that most Chebyshev or Remez exchange algorithms cannot handle time domain constraints, and are limited to one-dimensional filters.

We illustrate a typical FIR design procedure. (The following is intended to illustrate the behavior of this algorithm, and no attempt is made to present a practical design methodology.) Our goal is a 25 coefficient zero-phase two-band low pass filter with twice the gain in the second band than in the first. We define the edges of the bands, set transition regions, and adjust the tolerance for the deviation we are willing to allow in each band. Figure 7.15.3a shows the resulting FIR filter after about 15 iterations, with extremely tight tolerances used in the bands. The dashed lines indicate the frequency constraints, and the solid line the FIR filter response. Note that the ripple exceeds the tolerance levels and the transition regions of the filter are too wide. This is the best that a least squares 25 point filter can do. We slowly relaxed the tolerances in each band and continued iterating. After about 45 iterations, we achieved the filter shown in figure 7.15.3b. Note that the ripple is much larger, though still not within the constraints, but the transition regions have become narrower. Continuing to slowly relax the tolerances in each band, we finally ended at the filter shown in figure 7.15.3c, which exactly meets both the time constraint (25 points) and the frequency constraints. Note that this is a classic Chebyshev "equiripple" filter design. Figure 7.15.3d shows the FIR filter coefficients. We tried adding time domain constraints such as requiring all coefficients to be less than 5; unfortunately, these types of constraints distort the resulting filter so badly that it is not at all possible to match the desired response. Time domain constraints that might be quite useful, however, would be to force small coefficients of the filter to zero, then adjust the remaining coefficients to optimize the response. This could significantly save on the number of multiplications that would be necessary to implement the filter.



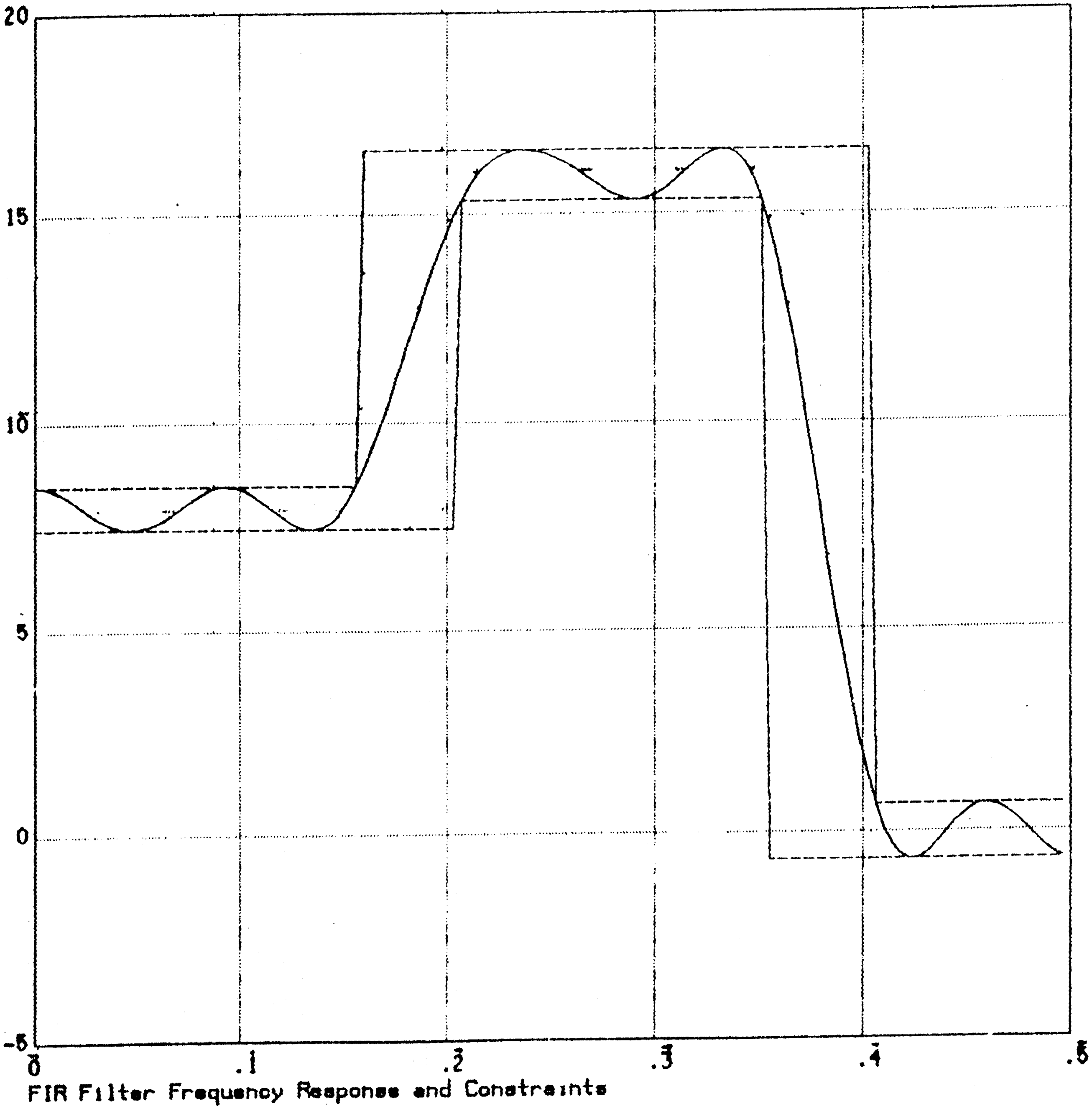
```
ITFIR

File: F.25.1
Tue Aug 24 04:25:15 1982

25 point filter
256 point FFT

xerr=   0.0619229 iter=18

Fixed PARTAN-like extrapolation
alpha=1.400000

target1=   8.000 +-   0.016
target2=  16.000 +-   0.016
target3=   0.000 +-   0.016

band1=  0.0000 to   0.1570
band2=  0.2050 to   0.3520
band3=  0.4030 to   0.5000
```

Figure 7.15.3a - 25 Point FIR Filter - Frequency Response



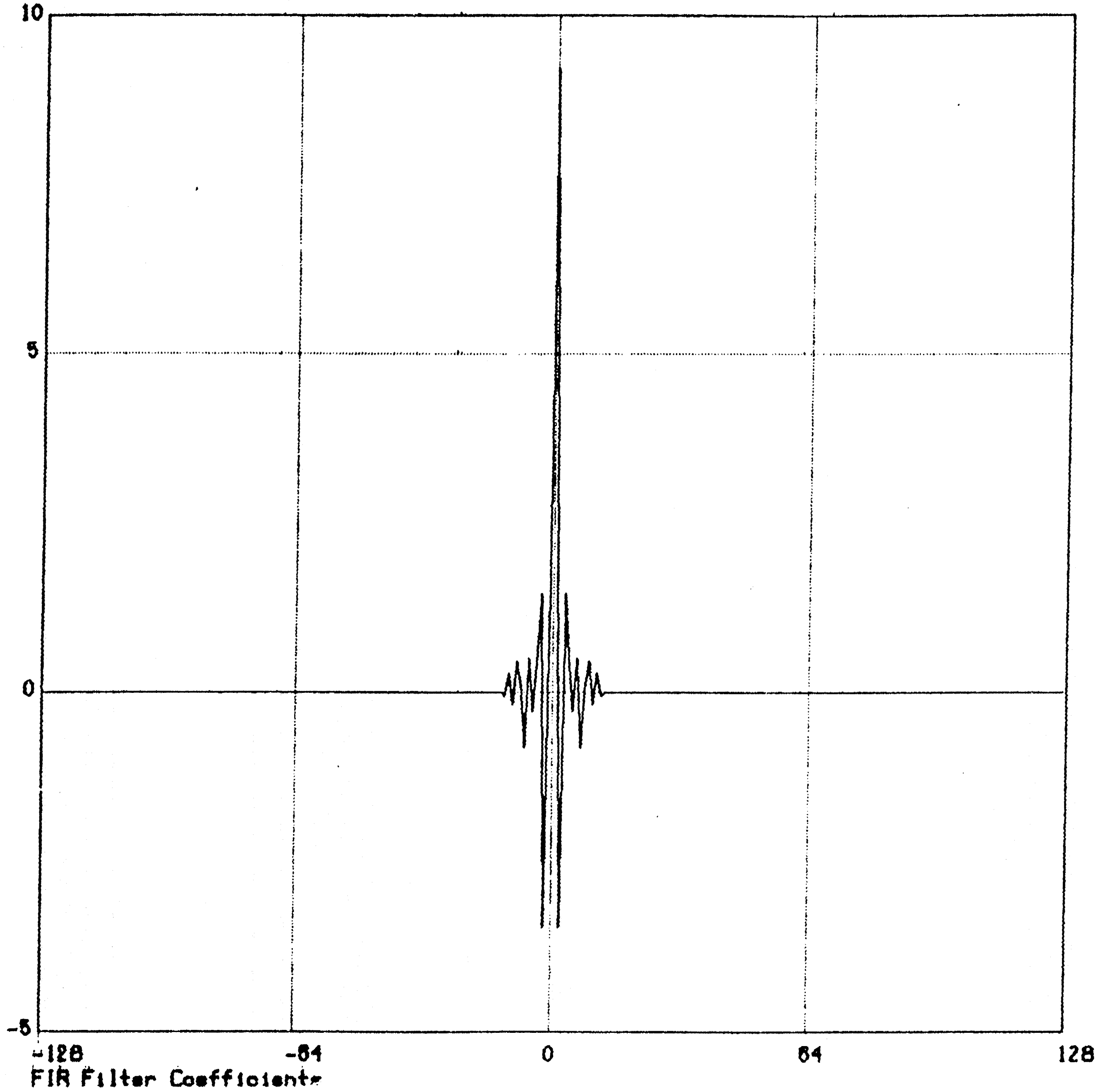
```
ITFIR

File: F.25.3
Tue Aug 24 04:28:19 1982

25 point filter
256 point FFT

xerr=   0.0122198 iter=45

Fixed PARTAN-like extrapolation
alpha=1.300000

target1=   8.000 +-   0.240
target2=  16.000 +-   0.320
target3=   0.000 +-   0.320

band1=  0.000  to   0.1570
band2=  0.2050 to   0.3520
band3=  0.4030 to   0.5000
```

Figure 7.15.3b - 25 Point FIR Filter - Frequency Response



```
ITFIR

File: F.25.8
Tue Aug 24 04:49:11 1982

25 point filter
256 point FFT

xerr=   0.0000000 iter=268

Fixed PARTAN-like extrapolation
alpha=1.300000

target1=   8.000 +-    0.502
target2=  16.000 +-    0.662
target3=   0.000 +-    0.662

band1=  0.0000 to   0.1570
band2=  0.2050 to   0.3520
band3=  0.4030 to   0.5000
```

Figure 7.15.3c - 25 Point FIR Filter - Frequency Response



```
ITFIR

File: F.25.8
Tue Aug 24 04:49:11 1982

25 point filter
256 point FFT

xerr=   0.0000000 iter=268

Fixed PARTAN-like extrapolation
alpha=1.300000

target1=   8.000 +-   0.502
target2=  16.000 +-   0.662
target3=   0.000 +-   0.662

band1=  0.0000 to   0.1570
band2=  0.2050 to   0.3520
band3=  0.4030 to   0.5000
```

Figure 7.15.3d - 25 Point FIR Filter - Time Domain

Section D - Non-Convex Constraint Sets

16. Magnitude-Only Signal Reconstruction

We have not had as much success with constraint sets which are not convex. The difficulty is that our convergence proofs only guarantee that if the estimates remain bounded then they will converge to a critical point of the objective function. There is no guarantee of convergence to a global optimizing solution, and the critical point we converge to could be quite far from the global optimizing solution. The only example we will discuss of this type is reconstruction of a finite length signal from the noisy magnitude of its Fourier Transform. Let us assume that $\underline{x}$ is known to be a finite multidimensional sequence, and that the magnitudes $|Y(\omega_i)|$ of the samples of the Fourier Transform of $\underline{y}$ are known.

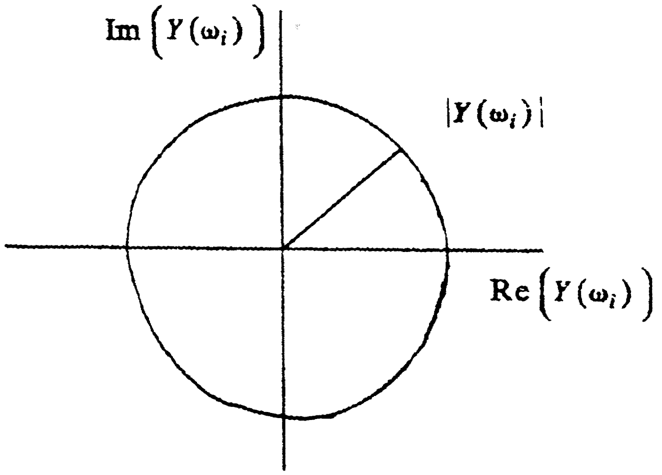

$\mathrm{Im}\left(Y(\omega_i)\right)$
$|Y(\omega_i)|$
$\mathrm{Re}\left(Y(\omega_i)\right)$

Figure 7.16.1 - Magnitude Constraint Set

Note that while the signal constraint is linear, the magnitude constraint on $Y(\omega_i)$ is not convex. Hayes [3] has proven that a finite sequence is uniquely defined by its magnitude up to translation and rotation of its factors. We can easily apply our iterative algorithms to this problem; the only tricky part is evaluat..g the conditional expectation of $Y(\omega_i)$ given $\hat{X}(\omega_i)$, an operation which requires using modified zero and first-order Bessel functions $I_0(z)$ and $I_1(z)$ (these routines are available in many scientific subroutine libraries.)

MCEM/XMAP Magnitude-Only Reconstruction

Guess $\hat{Y}_0(\omega_i)$

For $k=0,1,\cdots$

$$\hat{x}_k(n) = \begin{cases} \hat{y}_k(n) & \text{wherever } x(n) \text{ is unknown} \\ 0 & \text{else} \end{cases} \tag{7.16.1}$$

$$\hat{Y}_{k+1}(\omega_i) = |Y(\omega_i)| \frac{I_1\left(\frac{1}{r}|\hat{X}_{k+1}(\omega_i)Y(\omega_i)|\right)}{I_0\left(\frac{1}{r}|\hat{X}_{k+1}(\omega_i)Y(\omega_i)|\right)} e^{j\angle\hat{X}_{k+1}(\omega_i)}$$

where:

$$I_n(z) = e^{-j\frac{1}{2}n\pi} J_n(ze^{j\frac{1}{2}\pi})$$

$$J_n(z) = \frac{1}{j^n\pi}\int_0^{\pi} e^{jz\cos\phi}\cos(n\phi)\,d\phi$$

and:

YMAP/XYMAP Magnitude-Only Reconstruction

Guess $\hat{Y}_0(\omega_i)$

For $k=0,1,\cdots$

$$\hat{x}_k(n) = \begin{cases} \hat{y}_k(n) & \text{wherever } x(n) \text{ is non-zero} \\ 0 & \text{else} \end{cases} \tag{7.16.2}$$

$$\hat{Y}_{k+1}(\omega_i) = |Y(\omega_i)| \, e^{j\angle\hat{X}_{k+1}(\omega_i)}$$

This last YMAP/XYMAP algorithm is identical to the one proposed by Fienup [6] and Hayes [3] and is in the same category as Gerchberg and Saxton [7]. Corollary 6.2 in chapter 6 guarantees that since the constraint set Y is bounded and closed, and X is convex, then the estimates must remain bounded and must converge to a critical point of the objective function. If initialized at a spectrum $Y(\omega_i)$ that is close to the true signal, this algorithm has a good chance to converge to the correct solution. In general, however, due to the non-convexity of the magnitude constraints, there are an enormous number of local minima of the objective function which are not at all similar to the global minimizing solution. If initialized randomly, the iteration inevitably locates one of these local minima, and cannot find the true solution. We have not tried the MCEM algorithm yet, and so can not report on its performance. Fienup suggested jiggling the iteration by switching extrapolation methods or varying the procedure in other ways. However, we had no success with this.

An approach which was marginally better than that above was to restate the constraints in the following form. Let M_i be the magnitude of $\frac{1}{N}|Y(\omega_i)|^2$. Then require:

$$X = \left\{ x(n) \;\middle|\; x(n) \text{ a finite sequence, and } \sum_n x^2(n) = \sum_i M_i \right\} \tag{7.16.3}$$

$$Y = \left\{ Y(\omega_i) \;\middle|\; \frac{1}{N}|Y(\omega_i)|^2 \le M_i \right\}$$

Here we constrain the output spectrum samples to be less than or equal to their known magnitude, and indirectly enforce the fact that they must be equal by requiring the total signal energy to add up to the maximum possible. Thus we have replaced $N/2$ non-convex constraints by a single non-convex constraint. Unfortunately, the resulting estimation XYMAP algorithm still has local minima, and still tends to converge to one of these.

1. Summary

The examples presented in this chapter represent only a few of the many possible applications of the algorithms presented in the thesis. The advantage of our approach is that the MCEM, XMAP, YMAP and XYMAP algorithms are easily computed, easily understood, applicable to many problems involving multiple constraints stated in different domains, and their convergence behavior is dominated by the fact that we are minimizing a relatively simple and often concave objective function. This one unified approach not only yields most of the well known iterative projection methods, such as bandlimited extrapolation, but also suggests improved algorithms in problems such as phase-only signal reconstruction modulo 2π, in which the known noise statistics are used to improve the estimates.

In general, our signal reconstruction algorithms work best when the constraint sets are defined by linear equalities, or are convex. These applications, bandlimited extrapolation, phase-only recontruction, and multi-dimensional FIR filter design, are the most straightforward and are guaranteed to converge. Problems involving nonlinear constraints, such as magnitude-only reconstruction, are significantly more difficult due to the presence of local minima.

References

1. Athanasios Papoulis, "A New Algorithm in Spectral Analysis and Bandlimited Signal Extrapolation," *IEEE Trans. Circuits Syst.* CAS-22(9), pp.735-742 (Sept 1975).
2. Anil K. Jain and Surendra Ranganath, "Extrapolation Algorithms for Discrete Signals with Application in Spectral Estimation," *IEEE Trans. Acoust., Speech, Sig. Proc.* ASSP-29(4), pp.830-845 (Aug 1981).
3. Monson H. Hayes III., *Signal Reconstruction from Phase or Magnitude,* M.I.T. PhD Thesis (June 1981).
4. Monty H. Hayes, Jae S. Lim, and Alan V. Oppenheim, "Signal Reconstruction from Phase or Magnitude," *IEEE Trans. Acoust., Speech, and Signal Processing* ASSP-28(6), pp.672-680 (Dec 1980).
5. Monson H. Hayes and Victor T. Tom, *Adaptive Acceleration of Iterative Signal Recontruction Algorithms,* Technical Note 1980-28, Lincoln Laboratory M.I.T. (to be published).
6. J.R. Fienup, "Reconstruction of an Object from the Modulus of its Fourier Transform," *Optics Letters* 3(1), pp.27-29 (July 1978).
7. R.W. Gerchberg and W.O. Saxton, "A Practical Algorithm for the Determination of Phase From Image and Diffraction Plane Pictures," *Optik* 35, pp.237-246 (1972).
8. James H. McClellan and Thomas W. Parks, "Eigenvalue and Eigenvector Decomposition of the Discrete Fourier Transform," *IEEE Trans. Audio Electr.* AU-20(1), pp.66-74 (March 1972).
9. James H. McClellan, "Comments on 'Eigenvector and Eigenvalue Decomposition of the Discrete Fourier Transform'," *IEEE Trans. Audio Electr.* AU-21, p.65 (Feb 1973).
10. Charles L. Lawson and Richard J. Hanson, *Solving Least Squares Problems,* Prentice Hall, Inc., Englewood Cliffs, N.J. (1974).
11. D. Slepian, H.O. Pollak, and H.J. Landau, "Prolate Spheroidal Wave Functions," *Bell Syst. Tech. J.* **40**, pp.43-84 (Jan 1961).
12. I.W. Sandberg, "On the Properties of Some Systems That Distort Signals I," *Bell Syst. Tech. J.* **42**, pp.2033-2046 (Sept 1963).
13. I.W. Sandberg, "On the Properties of Some Systems That Distort Signals II," *Bell Syst. Tech. J.* **43**, pp.91-112 (Jan 1964).
14. H.J. Landau, "On the Recovery of a Band-Limited Signal, After Instantaneous Companding and Subsequent Band-Limiting," *Bell Syst. Tech. J.* **39**, pp.351-364 (March 1960).
15. R. W. Gerchberg, "Super-resolution through Error Energy Reduction," *Optica Acta* **21**, pp.709-720 (1974).
16. M. Shaker Sabri and Willem Steenaart, "An Approach to Band-Limited Signal Extrapolation: The Extrapolation Matrix," *IEEE Trans. Circ. Sys.* CAS-25(2), pp.74-78 (Feb 1978).

17. J. A. Cadzow, "An Extrapolation Procedure for Band-Limited Signals," *IEEE Trans Acoust., Speech, and Signal Processing* ASSP-27(1), pp.4-12 (Feb 1979).

18. N. Levinson, "The Weiner RMS Error Criterion in Filter Design and Prediction," *J. Math. Phys* 25, pp.261-278 (Jan 1947).

19. William F. Trench, "An Algorithm for the Inversion of Finite Toeplitz Matrices," *J. Soc. Indust. Appl. Math* 12, pp.515-522 (Sept 1964).

20. S. Zohar, "The Solution of a Toeplitz Set of Linear Equations," *J. Ass. Comput. Mach.* 21(2), pp.272-276 (April 1974).

21. Victor T. Tom, Thomas F. Quatieri, Monson H. Hayes, and James H. McClellan, "Convergence of Iterative Nonexpansive Signal Reconstruction Algorithms," *IEEE Trans. Acoust., Speech, Sig. Proc.* ASSP-29(5), pp.1052-1058 (Oct 1981).

22. Tom F. Quatieri and Alan V. Oppenheim, "Iterative Techniques for Minimum Phase Signal Reconstruction from Phase or Magnitude," *IEEE Trans. Acoust., Speech, and Sig. Proc.* ASSP-29(6), pp.1187-1193 (Dec 1981).

23. Carol Espy, *S.M. Thesis MIT*, 1981.

24. A.A. Goldstein, *Constructive Real Analysis*, Harper and Row, New York (1967).

25. Hans Künzi and Wilhelm Krelle, *Nonlinear Programming*, Blaisdell Publishing, Waltham, Mass. (1966).

26. Lawrence R. Rabiner and Bernard Gold, *Theory and Applications of Digital Signal Processing*, Prentice Hall Inc., Englewood Cliffs, N.J. (1975).

27. Lawrence R. Rabiner, "The Design of Finite Impulse Response Digital Filters Using Linear Programming Techniques," *Bell System Tech. J.*, pp.1177-1198 (July-Aug 1972).

28. Lawrence R. Rabiner, "Linear Program Design of Finite Impulse Response (FIR) Digital Filters," *IEEE Trans. Audio Electro.* **20**(4), pp.280-288 (Oct 1972).

Chapter 8

Applications of Optimal Signal Reconstruction

Part IV - Short Time Fourier Transforms, MEM, and Penalty Functions

1. Introduction

In this chapter we consider more esoteric applications of our optimal signal reconstruction theory. We first propose a generalization of Short Time Fourier Transforms (STFT), in which we divide the transform operation into an "orthonormal" sectioning operator followed by an array of Fourier Transforms. This more general viewpoint indicates the proper methods for handling boundary conditions in the transforms, suggests more general windowing procedures, demonstrates that the inverse STFT is a projection operator, and yields a Parseval-like theorem relating the energy of the signal in the time and Short Time Fourier domains. This allows us to directly apply all our previous theory to problems involving reconstruction of signals from a mixture of constraints on its behavior in the time and Short Time Fourier domains. This new framework also provides a structure for evaluating algorithms such as that of Nawab [1,2] for reconstructing a signal trom the magnitude of its Short Time Fourier Transform.

The remaining algorithms do not strictly belong to the stochastic framework which we have developed in this thesis. We discuss them because of their intrinsic interest, and because their structure is very similar to that of XYMAP Both of these problems iteratively minimize a least squares objective function involving several unknowns, where the given constraints on each unknown are stated in different domains. In addition, the same convergence proofs used for XYMAP can also be applied to these algorithms.

The first of these applications was motivated by a "ping-pong" algorithm developed by Lim and Malik [3,4] for solving multidimensional MEM problems. This algorithm tries to extend a set of multidimensional correlations given that the infinite correlation set is the inverse Fourier transform of one over a finite polynomial. We provide a more systematic approach to this problem, and show that the problem actually involves constraints stated in three different domains. The resulting reconstruction algorithm then resembles that of Malik and Lim, except that a quartic equation must be solved at each frequency sample to estimate the spectrum. Malik and Lim's algorithm appears as a form of our algorithm in which a non-optimal decision is made at each iteration. Since we have not yet programmed our algorithm, however, it is too early to conclude which is the better approach.

Finally, we suggest a new method for constructing penalty functions for constrained minimization problems. By introducing an extra variable belonging to the constraint set, we can convert the original constrained minimization into an iterative sequence of unconstrained minimizations and projection operators.

2. Short Time Fourier Transforms†

Stationary signals are particularly convenient to model or filter because, among other reasons, the Fourier Transform of the data is a white noise sequence in which each frequency sample is stochastically independent of all other frequency samples. Thus by transforming to the Fourier domain, we no longer need to consider the interaction between adjacent samples, and can process each frequency component

† The application of this thesis to reconstruction of signals using Short Time Fourier Transforms was suggested by discussions with Hamid Nawab.

independently. Unfortunately, when the data is non-stationary, as in speech or images, its Fourier Transform can no longer be interpreted in such a simple way.

A common engineering approach to this difficulty is to try to exploit the "local" stationarity of speech or images by sectioning the data with windows, then processing each frame as if the data it contained were stationary. After modifying each frame, the sections must be stitched back together in some reasonable fashion to reconstruct the complete signal. A desirable feature of this sectioning and resynthesis procedure is that if no processing is performed on the data, then the procedure should exactly return the signal we started with. This requirement of being an identity transformation is typically used in the development of this topic to specify constraints on the shape of the windows used to section the data. The overall procedure of sectioning the data and Fourier transforming each frame is called a Short Time Fourier Transform (STFT), and the inverse procedure of inverse transforming each segment and then stitching the frames back together is called an Inverse Short Time Fourier Transform (STFT^{-1}).

2.1. A Development of the Short Time Fourier Transform

We will present a somewhat unusual development of this subject, one which not only provides an interesting and useful generalization of the procedure, but which also highlights the implications of requiring the analysis and resynthesis procedures to be an identity system. Let $\{B_i\}$ be a set of linear operators on the data $\underline{x}$. These "window" matrices could be diagonal with elements $b_i(n)$ on the diagonal, in which case multiplying B_i times $\underline{x}$ would be equivalent to multiplying the signal $x(n)$ by a windowing function $b_i(n)$. This interpretation is not essential, however, and B_i could have any arbitrary form, subject to certain constraints we develop below. Let $\underline{y}_i = B_i\underline{x}$ be the i^{th} "windowed frame of data", and let $\underline{y}$ be the collection of all these data segments:

$$\underline{y} = \begin{pmatrix} \underline{y}_1 \\ \vdots \\ \underline{y}_L \end{pmatrix} = \begin{pmatrix} \mathrm{B}_1 \\ \vdots \\ \mathrm{B}_L \end{pmatrix} \underline{x} \tag{8.2.1}$$

Let B be the block matrix on the right. The matrix B can thus be intepreted as a sectioning operator which converts a one-dimensional signal $\underline{x}$ into an array of signal frames $\underline{y}$. Let B_i have size $M_i \times N$, so that $\underline{y}_i$ is a vector of length M_i. Let $M = \sum_{i=1}^{L} M_i$ so that $\underline{y}$ is an M element vector, and B is an $M \times N$ matrix.

In order that there be a resynthesis procedure which can reconstruct the signal $\underline{x}$ given its sectioned representation $\underline{y}$, it is necessary and sufficient that the mapping B be one-to-one. This means that B must have full column rank so that its null space only contains the zero vector. With this constraint, there must exist a linear operator F which will map the sectioned data $\underline{y}$ back onto the original $\underline{x}$:

$$\underline{x} = \mathrm{F}\underline{y} = \begin{pmatrix} \mathrm{F}_1 & \cdots & \mathrm{F}_L \end{pmatrix} \begin{pmatrix} \underline{y}_1 \\ \vdots \\ \underline{y}_L \end{pmatrix} = \sum_{i=1}^{L} \mathrm{F}_i \mathrm{B}_i \underline{x} = \underline{x} \tag{8.2.2}$$

which means that the blocks forming the operator F must satisfy:

$$\sum_{i=1}^{L} \mathrm{F}_i \mathrm{B}_i = \mathrm{I} \tag{8.2.3}$$

If the total number of points in the sectioned data vector $\underline{y}$ is larger than the number of points in the original signal, $M > N$, then there will be many linear operators F which properly map the range of B back onto its domain. One choice for F which is particularly convenient to use is the left pseudo-inverse of B:

$$\mathrm{F} = \mathrm{B}^{\dagger} = (\mathrm{B}^H\mathrm{B})^{-1}\mathrm{B}^H = \left(\sum_{i=1}^{L} \mathrm{B}_i^H \mathrm{B}_i \right)^{-1} \mathrm{B}^H \tag{8.2.4}$$

This operator is well defined because B has full column rank, and therefore $\mathrm{B}^H\mathrm{B}$ is invertible. The pseudo-inverse $F = B^{\dagger}$ not only properly maps the range of B onto its domain, but also maps the orthogonal complement of the range to zero:

$$\begin{aligned} B^{\dagger}\underline{y} &= \underline{x} && \text{if } \underline{x} = B\underline{y} \\ B^{\dagger}\underline{y} &= \underline{0} && \text{if } \underline{y} \perp B\underline{x} \text{ for all } \underline{x} \end{aligned} \tag{8.2.5}$$

Another way of saying the same thing is that $B^{\dagger}\underline{y}$ is the signal $\hat{\underline{x}}$ such that $B\hat{\underline{x}}$ comes as close as possible to $\underline{y}$:

$$B^{\dagger}\underline{y} = \hat{\underline{x}} - \min_{\underline{x}} \|\underline{y} - B\underline{x}\|^2 \tag{8.2.6}$$

The pseudo-inverse has several useful properties:

a) $B^{\dagger}B = I$. Thus sectioning the data (multiplying by B) followed by resynthesizing the signal by piecing together frames (multiplying by $B^{\dagger}$) is an identity operation.

b) $BB^{\dagger}$ is a projection matrix. It is an identity on $R(B)$, and its null space is the orthogonal complement $R^{\perp}(B)$:

$$\begin{aligned} BB^{\dagger}\underline{y} &= \underline{y} && \text{if } \underline{y} = B\underline{x} \text{ for some } \underline{x} \\ BB^{\dagger}\underline{y} &= \underline{0} && \text{if } \underline{y} \perp B\underline{x} \text{ for all } \underline{x} \end{aligned} \tag{8.2.7}$$

Thus if we decompose any M long vector $\underline{y}$ into $\underline{y} = \underline{y}_1 + \underline{y}_2$, where $\underline{y}_1$ is a component in $R(B)$ and $\underline{y}_2$ is orthogonal to $R(B)$, then $BB^{\dagger}\underline{y} = \underline{y}_1$.

It would be more elegant if the analysis and synthesis procedures were symmetric, so that the synthesis operator $F = B^{\dagger}$ were just the Hermitian of the analysis operator, $B^{\dagger} = B^H$. We will call such a matrix orthonormal, since $B^H B = I$ implies that the columns of B must be orthonormal. Fortunately, it is easy to convert any one-to-one windowing matrix $\tilde{B}$ into orthonormal form. Let $(\tilde{B}^H\tilde{B})^{1/2}$ be a "square root" of $\tilde{B}^H\tilde{B}$, so that $(\tilde{B}^H\tilde{B}) = (\tilde{B}^H\tilde{B})^{1/2}(\tilde{B}^H\tilde{B})^{H/2}$. Then an orthonormal windowing matrix B can be defined in terms of $\tilde{B}$ as follows:

$$B = \tilde{B}(\tilde{B}^H\tilde{B})^{-H/2} \tag{8.2.8}$$

Note that:

$$B^H B = (\tilde{B}^H\tilde{B})^{-1/2}\tilde{B}^H\tilde{B}(\tilde{B}^H\tilde{B})^{-H/2} = I \tag{8.2.9}$$

so that the pseudo-inverse $B^\dagger$ is just the Hermitian of B. As an example, suppose that all the window matrices $\tilde{B}_i$ are diagonal with entries $b_i(n)$ on the diagonal. Then the blocks B_i in the orthonormalized window matrix $B_i = \tilde{B}(\tilde{B}^H\tilde{B})^{-H/2}$ would also be diagonal, with elements $\dfrac{b_i(n)}{\left(\sum_{i=1}^{L} b_i^2(n)\right)^{1/2}}$.

Given an orthonormal windowing matrix B, we now define the Short Time Fourier Transform (STFT) of the signal $\underline{x}$ as the vector $\underline{y}_f$ of Fourier Transforms $\underline{y}_{f_i}$ of each windowed segment $\underline{y}_i$:

$$\mathrm{STFT}(\underline{x}) = \underline{y}_f = \begin{pmatrix} \underline{y}_{f_1} \\ \vdots \\ \underline{y}_{f_L} \end{pmatrix} = \begin{pmatrix} W_{M_1}\underline{y}_1 \\ \vdots \\ W_{M_L}\underline{y}_L \end{pmatrix} = \begin{pmatrix} W_{M_1}B_1 \\ \vdots \\ W_{M_L}B_L \end{pmatrix} \underline{x} \tag{8.2.10}$$

Let W_B be the matrix on the right. Note that:

$$\underline{y}_{f_j} = \frac{1}{\sqrt{M_j}} \begin{pmatrix} Y_j(\omega_0) \\ \vdots \\ Y_j(\omega_{M_j-1}) \end{pmatrix} \qquad \text{where } \omega_i = \frac{2\pi i}{M_j} \tag{8.2.11}$$

where $Y_j(\omega_i)$ is the DFT of the j^{th} frame $\underline{y}_i$. Beware that the definition of the Short Time Fourier Transform depends intimately on the choice of the orthonormal windowing matrix B; different windows lead to different Short Time Fourier Transforms.

The inverse Short Time Fourier Transform (STFT^{-1}) is defined as the synthesis of the inverse transform of each segment:

$$\mathrm{STFT}^{-1}(\underline{y}_f) = B^H \begin{pmatrix} W^H_{M_1}\underline{y}_{f_1} \\ \vdots \\ W^H_{M_L}\underline{y}_{f_L} \end{pmatrix} = \begin{pmatrix} B_1^H W^H_{M_1} & \cdots & B_L^H W^H_{M_L} \end{pmatrix} \underline{y}_f \tag{8.2.12}$$

Note that the inverse STFT matrix operator is simply $\mathbf{W}_B^H$, the Hermitian of the forward STFT matrix operator. In fact, because the DFT matrices $\mathbf{W}_{M_j}$ are orthonormal, $\mathbf{W}_{M_j}^H\mathbf{W}_{M_j} = \mathbf{W}_{M_j}\mathbf{W}_{M_j}^H = \mathbf{I}$, the inverse STFT matrix $\mathbf{W}_B^H$ is the pseudo-inverse of $\mathbf{W}_B$ and satisfies the following properties:

a) $\mathbf{W}_B^H\mathbf{W}_B = \mathbf{I}$

b) $\mathbf{W}_B\mathbf{W}_B^H$ is a projection matrix which is an identity on $R(\mathbf{W}_B)$ and has null space $R^{\perp}(\mathbf{W}_B)$.

Thus $\mathbf{W}_B^H$ maps an arbitrary vector $\underline{y}$ into the signal $\underline{x}$ whose STFT comes as close as possible to $\underline{y}$. These properties also lead to a Parseval's relationship for Short Time Fourier Transforms.

<u>Theorem 8.2.1</u> Let $\underline{y} = \mathbf{B}\underline{x}$. Then:

$$\sum_{j=1}^{L}\sum_{i=0}^{M_j-1}\frac{1}{M_j}\left|Y_j(\omega_i)\right|^2 = \sum_{j=1}^{L}\sum_{i=0}^{M_j-1}\left|y_j(n)\right|^2 = \sum_{n=0}^{N-1}\left|x(n)\right|^2 \qquad (8.2.13)$$

<u>Proof:</u>

$$\begin{aligned}\underline{y}_f^H\underline{y}_f &= \sum_{j=1}^{L}\underline{y}_j^H\mathbf{W}_{M_j}^H\mathbf{W}_{M_j}\underline{y}_j \\ &= \sum_{j=1}^{L}\underline{y}_j^H\underline{y}_j \\ &= \underline{y}^H\underline{y}\end{aligned}$$

and:

$$\begin{aligned}\underline{y}^H\underline{y} &= \underline{x}^H\mathbf{B}^H\mathbf{B}\underline{x} \\ &= \underline{x}^H\underline{x}\end{aligned}$$

Expanding these inner products in terms of their elements gives the theorem. □

2.2. Reconstruction from Time and STFT Domain Constraints

Because of all these properties, it is easy to apply our optimal signal reconstruction algorithms to problems involving a mixture of constraints in the time domain and the Short Time Fourier Domain. Consider the signal model shown in figure 8.2.1:

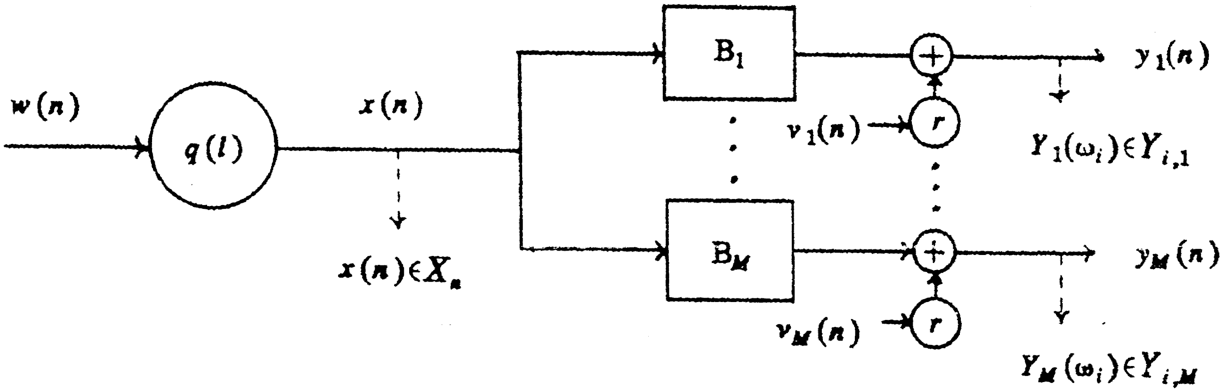


Figure 8.2.1 - Short Time Fourier Transform Reconstruction Model

Model: $\underline{x} = \underline{w}$ where $p(\underline{w}) = N(\underline{0},Q)$

$\underline{y} = \underline{x} + \underline{v}$ where $p(\underline{v}) = N(\underline{0},R)$

$$\begin{aligned} \text{where } [Q]_{l,m} &= q(l)\delta_{l,m} \\ [R]_{l,m} &= r\delta_{l,m} \\ B &= \text{orthonormal sectioning operator} \end{aligned} \tag{8.2.14}$$

Observations: $x(n) \in X_n$

$Y_j(\omega_i) \in Y_{j,i}$

The signal is a white Gaussian sequence with zero mean and time varying variance $q(n)$. Each signal sample $x(n)$ is thus stochastically independent of all other signal samples. The signal is cut into sections by the orthonormal windowing matrix B, and white Gaussian noise with variance r is added to every element of every frame of data.

The available observation information indicates that each $x(n)$ is in some range X_n, and each sample of the spectrum of each frame of the output data, $Y_j(\omega_i)$, is in some range $Y_{j,i}$. These constraint sets are independent of any other signal or output samples. Given this model, our goal is to estimate the signal $\underline{x}$ and the noisy sectioned output $\underline{y}$.

The probability density for this model has the form:

$$\log p(\underline{x},\underline{y}) = -\frac{1}{2}\left\{ \|\underline{x}\|_Q^2 + \|\underline{y}-B\underline{x}\|_R^2 + \text{constant} \right\} \tag{8.2.15}$$

which is exactly the form assumed in chapter 5. Plugging this into our algorithms, using the fact that the conditional probability densities $p_{X|Y}(\underline{x}|\hat{\underline{y}})$ and $p_{Y|X}(\underline{y}|\hat{\underline{x}})$ will be separable, and also using the Parseval relationship in (8.2.13), we get the reconstruction algorithm shown in table 8.2.1. These algorithms are virtually identical to those suggested in chapter 7, except that we have substituted Short Time Fourier Transforms for Fourier Transforms. Start with an estimate of the Short Time Fourier Transform $\hat{Y}_{j,0}(\omega_i)$. Inverse transform each frame, giving $\hat{\underline{y}}_{j,k}$, then stitch together frames by multiplying by B^H, thus generating the nearest time domain signal $\hat{y}_k(n)$ corresponding to this STFT. Filter by multiplying by $h(n) = \frac{q(n)}{q(n)+r}$ (the "time domain Weiner-Hopf filter"), and apply a projection or conditional expectation operator to estimate $\hat{x}_k(n) \in X_n$. To reestimate the output, form the Short Time Fourier Transform of the signal, $\hat{X}_{j,k+1}(\omega_i)$, and apply a projection or conditional expectation operator to estimate the output Short Time Fourier Transform, $\hat{Y}_{j,k+1}(\omega_i)$. Each iteration strictly decreases the cross-entropy, and the MAP methods also strictly increase the likelihood function. Each iteration thus improves the estimates. If the estimates remain bounded, convergence is guaranteed to a critical point of the cross-entropy. If the constraint sets

Table 8.2.1 - Time/STFT Constraint Iterative Algorithms

	Signal Estimate	Output Estimate	STFT^{-1}
MCEM:	$\hat{x}_{k+1}(n) = \mathrm{E}_{X_n}\left[x(n) \middle\vert h(n)\hat{y}_k(n)\right]$	$\frac{1}{\sqrt{N}}\hat{Y}_{j,k+1}(\omega_l) = \mathrm{E}_{Y_{j,l}}\left[\frac{1}{\sqrt{N}}Y_j(\omega_l) \middle\vert \frac{1}{\sqrt{N}}\hat{X}_{j,k+1}(\omega_l)\right]$	$\hat{\mathbf{y}}_{k+1} = \sum_{j=1}^{L}\mathrm{B}_j^{\mathrm{H}}\,\mathrm{DFT}^{-1}\left(\hat{Y}_j(\omega_l)\right)$
XMAP:	$\hat{x}_{k+1}(n) \leftarrow \min_{x(n)\in X_n}\left\vert x(n) - h(n)\hat{y}_k(n)\right\vert^2$	$\frac{1}{\sqrt{N}}\hat{Y}_{j,k+1}(\omega_l) = \mathrm{E}_{Y_{j,l}}\left[\frac{1}{\sqrt{N}}Y_j(\omega_l) \middle\vert \frac{1}{\sqrt{N}}\hat{X}_{j,k+1}(\omega_l)\right]$	$\hat{\mathbf{y}}_{k+1} = \sum_{j=1}^{L}\mathrm{B}_j^{\mathrm{H}}\,\mathrm{DFT}^{-1}\left(\hat{Y}_j(\omega_l)\right)$
YMAP:	$\hat{x}_{k+1}(n) = \mathrm{E}_{X_n}\left[x(n) \middle\vert h(n)\hat{y}_k(n)\right]$	$\frac{1}{\sqrt{N}}\hat{Y}_{j,k+1}(\omega_l) \leftarrow \min_{Y_j(\omega_l)\in Y_{j,l}}\frac{1}{N}\left\vert Y_j(\omega_l) - \hat{X}_{j,k+1}(\omega_l)\right\vert^2$	$\hat{\mathbf{y}}_{k+1} = \sum_{j=1}^{L}\mathrm{B}_j^{\mathrm{H}}\,\mathrm{DFT}^{-1}\left(\hat{Y}_j(\omega_l)\right)$
YMAP:	$\hat{x}_{k+1}(n) \leftarrow \min_{x(n)\in X_n}\left\vert x(n) - h(n)\hat{y}_k(n)\right\vert^2$	$\frac{1}{\sqrt{N}}\hat{Y}_{j,k+1}(\omega_l) \leftarrow \min_{Y_j(\omega_l)\in Y_{j,l}}\frac{1}{N}\left\vert Y_j(\omega_l) - \hat{X}_{j,k+1}(\omega_l)\right\vert^2$	$\hat{\mathbf{y}}_{k+1} = \sum_{j=1}^{L}\mathrm{B}_j^{\mathrm{H}}\,\mathrm{DFT}^{-1}\left(\hat{Y}_j(\omega_l)\right)$

where the conditional expectations are calculated with respect to the conditional densities:

$$\mathrm{p}_{X_n}\left(x(n) \middle\vert \hat{y}_k(n)\right) = N_{X_n}\left(h(n)\hat{y}_k(n), v(n)\right)$$

$$\mathrm{p}_{Y_{j,l}}\left(\frac{1}{\sqrt{N}}Y_j(\omega_l) \middle\vert \frac{1}{\sqrt{N}}\hat{X}_{j,k+1}(\omega_l)\right) = N_{Y_{j,l}}\left(\frac{1}{\sqrt{N}}\hat{X}_{j,k+1}(\omega_l)\ ,\ r\right)$$

where: Bayesian: $h(n) = \frac{q(n)}{q(n)+r}$ Fisher: $h(n)=1$

$$v(n) = \frac{q(n)r}{q(n)+r} \qquad v(n)=r$$

and where the Short Time Fourier Transforms are calculated using W_B.

X_n and Y_{ji} are all convex, geometric convergence is guaranteed to the unique global minimizing solution. In the Fisher case ($q(n)=\infty$) if X_n, Y_{ji} are convex, convergence is guaranteed to a global minimizing solution if and only if such a solution exists. The convergence rate is geometric in the Bayesian case, but can be sublinear in the Fisher case.

In most speech or image processing applications using Short Time Fourier Transforms, storing the entire sectioned output $\underline{y}$ is usually impossible (in fact, even storing the entire original signal $\underline{x}$ is usually difficult.) In implementing this algorithm, therefore, it is a good idea not to store the output $\underline{y}$, but to only store the output's inverse STFT $\hat{\underline{x}}$. Furthermore, we can estimate each output frame separately, add in its contribution to $\hat{\underline{x}}$, and then reuse the storage space to compute the next frame:

```
$\hat{\underline{x}} \leftarrow \underline{0}$
For $j=1, \ldots, L$
    Estimate $\hat{Y}_j(\omega_i)$ from $\hat{X}_j(\omega_i)$ using table 8.2.1
    $\hat{\underline{x}} \leftarrow \hat{\underline{x}} + \mathrm{B}_j^H \mathrm{DFT}^{-1}\left(\hat{Y}_j(\omega)\right)$
```

This frame by frame computation suggests an interesting variation on this algorithm. Rather than minimizing with respect to the entire signal $\underline{x}$ and then with respect to the entire segmented output $\underline{y}$, we could instead minimize with respect to a frame of output, the corresponding frame of signal, the next frame of output, etc. In the usual application, the window matrices extract small overlapping segments of $\underline{x}$. Because of the overlap, improving the estimate of one output frame should be immediately useful in estimating the next output frame. By updating the signal estimate $\hat{\underline{x}}$ as each new output frame is calculated, the next output frame calculation will reflect the improvement due to the last. The complete reorganized frame by frame iteration will then take the form: (we stripped off the iteration index k to simplify the notation)

```
Guess $\hat{\underline{y}}$
For $k=0,1,\cdots$
    For $j=1,\ldots,L$
            $\hat{X}_j(\omega) = \mathrm{DFT}\left(\mathrm{B}_j\hat{x}\right)$
            Estimate $\hat{Y}_j(\omega)$ from $\hat{X}_j(\omega)$ using table 8.2.1
            $\hat{\underline{y}} = \hat{\underline{y}} + \mathrm{B}_j^{\mathrm{H}}\left[\mathrm{DFT}^{-1}\left(\hat{Y}_j(\omega)\right) - \mathrm{B}_j\hat{\underline{y}}\right]$
            Estimate $\hat{x}$ from $\hat{\underline{y}}$ using table 8.2.1
```

After the j^{th} frame is reestimated, we update its inverse Short Time Fourier Transform $\underline{y}$ by windowing and subtracting off the old j^{th} frame, $\mathrm{B}_j^{\mathrm{H}}\mathrm{B}_j\underline{y}$, then adding in the windowed new frame, $\mathrm{B}_j^{\mathrm{H}}\mathrm{DFT}^{-1}(\hat{Y}_j(\omega))$. The signal $\hat{x}$ is reestimated, then we move on to the next frame, compute the DFT of the $(j+1)^{th}$ windowed segment of x and reestimate the corresponding frame of $\hat{Y}_{j+1}(\omega)$. Reestimating $\hat{x}$ from $\hat{\underline{y}}$ on the j^{th} step above is usually simplified by the fact that only the j^{th} frame of $\hat{\underline{y}}$ has changed since the last iteration. Thus, in the usual case where each frame only involves a few signal points, only a small section of $\hat{x}$ will have to be recomputed after each new output frame. Since the only difference between this iteration and our previous iteration is the order in which we minimize the cross-entropy, the convergence properties of the two approaches should be identical. In most applications, however, the convergence rate of the frame by frame approach should be faster. (The difference between these approaches is very similar to the difference between the Jacobi and Gauss-Seidel methods of solving linear equations. See, for example, Dahlquist and Bjorck [5] .)

Another possible improvement, called the Aitken double sweep method [6], would be to cycle forwards and backwards through the frames of data, estimating frames 1 through L, then reestimating the frames in reverse order, L through 1. This scheme, however, may be inconvenient when working with sequentially organized data files or pipelined processors.

All the algorithms we suggested in the previous chapter can be modified for use with Short Time Fourier Transforms. We leave the details to the reader.

An alternative model for this type of problem results when we switch the time and frequency domains. Now assume that $\underline{x}$ is an array of vectors $\underline{x}_1, \ldots, \underline{x}_L$ with lengths $N_1, \ldots, N_L$ and that the M point output $\underline{y}$ is formed by stitching the segments together and adding white noise:

$$\text{Model:} \qquad \underline{x}_j = \underline{w}_j \qquad\qquad \text{for } j=1, \ldots, L$$

$$\underline{y} = \sum_{j=1}^{L} \mathrm{B}_j^{\mathrm{H}} \underline{x}_j + \underline{v}$$

$$\text{where } \mathrm{p}(\underline{w}_j) = N(\underline{0}, \mathrm{Q}_j)$$

$$\mathrm{p}(\underline{v}) = N(\underline{0}, \mathrm{R})$$

$$\text{Observations:} \qquad X_j(\omega_i) \in \mathcal{X}_{j,i}$$

$$y(n) \in \mathcal{Y}_n$$

and each vector $\underline{x}_j = \underline{w}_j$ is assumed to be one cycle of a stationary periodic zero mean Gaussian sequence with periodic covariance $q(n)$. Thus each covariance matrix Q_j is circulant

$$\left[\mathrm{Q}_j\right]_{m,n} = q(m-n) \tag{8.2.16}$$

and $\mathrm{W}_{N_j} \mathrm{Q}_j \mathrm{W}_{N_j}^{\mathrm{H}}$ is a diagonal matrix whose diagonal elements are samples $Q_j(\omega_i)$ of the power spectrum of $q(n)$. This type of model is excellent for problems involving signals such as speech which are "locally stationary" but not "globally stationary". All of our iterative algorithms can now be applied to this problem. We will not go into the details, but the general form is:

For $k = 0,1,\cdots$

$\underline{y}_j$ = Filter and Short Time Fourier Transform $\underline{y}$ by computing $W_B H\underline{y}$.

Estimate $\hat{X}_j(\omega)$ from $\hat{\underline{y}}_j(\omega)$ by projection or conditional expectation.

Inverse Short Time Fourier Transform the signal, $\hat{\underline{x}} = \sum_j B_j DFT^{-1}(\hat{X}_j(\omega))$

Estimate $y(n)$ from $\underline{x}(n)$ by projection or conditional expectation.

Frame by frame iterations are also easily devised. The tricky part of this computation is the filtering step. Using the filter formula in chapter 5 gives:

$$H\underline{y} = \left[Q^{-1} + BR^{-1}B^H \right]^{-1} BR^{-1}\underline{y} \qquad (8.2.17)$$

$$\text{where } Q = \begin{pmatrix} Q_1 & & 0 \\ & \cdot & \\ 0 & & Q_L \end{pmatrix}$$

This is a huge matrix, since it is the size of $\underline{x}$, the total number of points in all sections. An equivalent but more convenient formula is:

$$H\underline{y} = QB \left[\sum_{j=1}^{L} B_j^H Q_j B_j + R \right]^{-1} \underline{y} \qquad (8.2.18)$$

Note that the matrix in brackets is the non-stationary (i.e. non-circulant) covariance matrix of the output $\underline{y} = \sum_j B_j \underline{x}_j + \underline{v}$. Unfortunately, in applications such as speech, this may still be a huge matrix. When the B_j matrix corresponds to windowing, then this matrix is band diagonal, but the bandwidth is the width of the window, which unfortunately, can be quite wide. The best approach would be to use an iterative block Gauss-Seidel approach to solving the filter equations [7]. This would be particularly convenient in the frame by frame version of the algorithm where only one window section of $\underline{y}$ will change at a time, so that the last filtered estimate will still be close to the new value except in the vicinity of the new section. The important point is that, unlike the frequency/time domain problem in chapter 5, section 6, the exact filter equations are *not* easily implemented by a frequency domain Weiner-Hopf filter.

3. Multidimensional Maximum Entropy Method Power Spectrum Estimation

A well known problem in spectral estimation is to find the power spectrum of a stationary process given only a finite set of correlations of the signal. Since there are infinitely many power spectra with these same correlations, we need some criterion for selecting one of these as our estimate. One approach is to find the power spectrum which is maximally non-committal with respect to the unknown correlations by choosing the spectrum with the largest entropy. Let $R(\underline{n})$ be the correlations of our multidimensional signal $x(\underline{n})$, where $\underline{n} = (n_1, n_2, \cdots)$ is the coordinate of the sample. Let $P(\underline{\omega})$ be the multidimensional Fourier Transform of $R(\underline{n})$:

$$P(\underline{\omega}) = \sum_{\underline{n}} R(\underline{n}) e^{j\underline{\omega}\underline{n}} \tag{8.3.1}$$

where $\underline{\omega} = (\omega_1, \omega_2, \cdots)$ is the frequency coordinate of the spectrum. Suppose the correlations $R(\underline{n})$ are known within some set of values $\underline{n} \in \Lambda$ (this set usually includes $\underline{0}$, and is usually symmetric, $\underline{n} \in \Lambda \Rightarrow -\underline{n} \in \Lambda$.) The Maximum Entropy Method (MEM) then estimates $P(\underline{\omega})$ by solving:

$$P(\underline{\omega}) = \max_{P(\underline{\omega})} \int \log P(\underline{\omega})\, d\underline{\omega} \tag{8.3.2}$$

$$\text{subject to } \left(\frac{1}{2\pi}\right)^N \int P(\underline{\omega}) e^{-j\underline{\omega}\underline{n}}\, d\underline{\omega} = R(\underline{n}) \text{ for } \underline{n} \in \Lambda$$

Using Lagrange Multipliers in the usual way to solve this constrained maximization gives:

$$\hat{P}(\underline{\omega}) = \sum_{\underline{n}} R(\underline{n}) e^{j\underline{\omega}\underline{n}} = \frac{1}{\sum_{\underline{n} \in \Lambda} \lambda_{\underline{n}} e^{j\underline{\omega}\underline{n}}} \tag{8.3.3}$$

Thus the power spectrum, which is the Fourier transform of an infinite set of correlations $R(\mathbf{n})$ whose values on the set Λ are given, must also be a finite all-pole polynomial with nonzero coefficients constrained to the set Λ. In one dimension, when Λ is a contiguous symmetric interval centered at the origin, we can solve for the polynomial coefficients by factoring the pole polynomial $\lambda(\omega) = \sum_{\mathbf{n}\in\Lambda} \lambda_{\mathbf{n}} e^{j\omega\mathbf{n}} = A(\omega)A^*(\omega)$, and then using an autoregressive modeling technique. In multiple dimensions, however, polynomials can not necessarily be factored, and so this approach for calculating the $\lambda_{\mathbf{n}}$ coefficients fails. Some sort of iterative search must therefore be used to find the appropriate polynomial coefficients $\lambda_{\mathbf{n}}$ such that the corresponding power spectrum has the specified correlations.

Numerous algorithms have been suggested for this problem (see, for example, Lang [8]), but the one of interest to us at present is a conceptually simple "ping-pong" approach suggested by Lim and Malik [3, 4]. Start by guessing the coefficients of a finite polynomial $\lambda(\underline{\omega}) = \sum_{\mathbf{n}\in\Lambda} \lambda_{\mathbf{n}} e^{j\underline{\omega}\mathbf{n}}$ whose spectrum is positive, $\lambda(\underline{\omega})>0$. Compute the corresponding power spectrum in (8.3.3) and inverse transform to find the set of correlations corresponding to the polynomial. Force the known correlations $R(\mathbf{n})$ for $\mathbf{n}\in\Lambda$ to their correct values, then transform to get the corresponding power spectrum $P(\underline{\omega})$. Inverse transform $\frac{1}{P(\underline{\omega})}$ to reestimate the polynomial $\lambda(\underline{\omega})$; this inverse transform will no longer be finite in extent, so truncate it to the set Λ and start all over.

$$\hat{R}_{k+1}(\mathbf{n}) = \begin{cases} R(\mathbf{n}) & \text{for } \mathbf{n}\in\Lambda \\ \mathrm{DFT}^{-1}\left(\dfrac{1}{\sum\limits_{\mathbf{n}\in\Lambda} \hat{\lambda}_k(\mathbf{n})e^{j\underline{\omega}\mathbf{n}}}\right) & \text{else} \end{cases} \tag{8.3.4}$$

$$\hat{\lambda}_{k+1}(\mathbf{n}) = \begin{cases} \mathrm{DFT}^{-1}\left(\dfrac{1}{\sum\limits_{\mathbf{n}} \hat{R}_{k+1}(\mathbf{n})e^{j\underline{\omega}\mathbf{n}}}\right) & \text{for } \mathbf{n}\in\Lambda \\ 0 & \text{else} \end{cases}$$

Unfortunately, it is also necessary to ensure that the power spectra of $\hat{R}_{k+1}(n)$ and $\hat{\lambda}_{k+1}(n)$ remain positive everywhere, something which the above simplified algorithm does not do. Malik and Lim's solution was to use cautious adaptive under-relaxation of the algorithm, testing at every stage to make sure that the estimates generated do not have negative spectral values. Directly calculating the entire set of correlations $R(n)$ on every pass of their algorithm was also avoided by only calculating the corrections to be made to the correlations inside Λ.

There is a more systematic approach to the problem. (Beware that we have not yet tested the algorithm that follows, and so there is no guarantee that this idea will work at all.) There are really three different types of constraints to be satisfied in this algorithm by three conceptually separate variables:

$$\begin{array}{ll} \hat{R}(n) = R(n) & \text{for } n \in \Lambda \\ \hat{\lambda}(n) = 0 & \text{for } n \notin \Lambda \\ \hat{P}(\omega) \geq 0 & \text{for all } \omega \end{array} \qquad (8.3.5)$$

Our goal is to find a set of correlations $\hat{R}(n)$, a finite polynomial $\hat{\lambda}(n)$, and a power spectrum $\hat{P}(\omega)$ satisfying these constraints such that:

$$\hat{P}(\omega) = \sum_{n} \hat{R}(n) e^{j\omega n} = \frac{1}{\sum\limits_{n \in \Lambda} \hat{\lambda}(n) e^{j\omega n}} \qquad (8.3.6)$$

Let $R(\omega)$ be the transform of $R(n)$, let $\hat{\lambda}(\omega)$ be the transform of $\hat{\lambda}(n)$, and let $\hat{P}(n)$ and $\hat{P}^{-1}(n)$ be the inverse transforms of $\hat{P}(\omega)$ and $\frac{1}{\hat{P}(\omega)}$ respectively.

A convenient objective function whose global minimum occurs at the solution (8.3.6) is given by:

$$\hat{R}, \hat{\lambda}, \hat{P} - \min_{R,\lambda,P} \sum_{\omega} \alpha \left| P(\omega) - R(\omega) \right|^2 + \beta \left| \frac{1}{P(\omega)} - \lambda(\omega) \right|^2 \qquad (8.3.7)$$

where α, β are arbitrary weights, and where the minimization is understood to be constrained to correlations, polynomials, and power spectra satisfying the constraints of (8.3.5). Iteratively minimizing with respect to R, λ, and P gives:

For $k=0,1,\cdots$

$$\hat{R}_{k+1}(n) = \begin{cases} R(n) & \text{for } n \in \Lambda \\ \hat{P}_k(n) & \text{else} \end{cases} \tag{8.3.8}$$

$$\hat{\lambda}_{k+1}(n) = \begin{cases} \hat{P}_k^{-1}(n) & \text{for } n \in \Lambda \\ 0 & \text{else} \end{cases}$$

$$\hat{P}_{k+1}(\omega) = \min_{P(\omega)>0} \alpha \left| P(\omega) - \hat{R}_{k+1}(\omega) \right|^2 + \beta \left| \frac{1}{P(\omega)} - \hat{\lambda}_{k+1}(\omega) \right|^2$$

Start with a model power spectrum $\hat{P}_0(\omega)$. To estimate the correlations $\hat{R}_{k+1}(n)$, inverse transform and force the values where $R(n)$ is known to their correct values. To estimate $\hat{\lambda}_{k+1}(n)$, inverse transform $\frac{1}{\hat{P}_k(\omega)}$ and truncate it to the right size. Now we reestimate the positive model spectrum by locating the value which comes as close as possible to these correlation and polynomial estimates. Each iteration thus alternates between moving the correlations $R(n)$ and polynomial $\hat{\lambda}(n)$ closer to the given power spectrum and its inverse, then readjusting the power spectrum to bring it closer to these new correlation and polynomial estimates. Each iteration decreases the objective function, and if the estimates remain bounded, convergence is guaranteed to a critical point of the objective function. Unfortunately, although the constraint sets in (8.3.5) are convex, the objective function (8.3.7) is not convex, and therefore there may be many local minima and critical points to confuse the iteration's search for the global minimum.

Note that this algorithm is similar in spirit to that of Malik and Lim. One major difference is that we make no attempt to force the correlations $R(n)$ and the polynomial $\lambda(n)$ to have positive spectra. Malik and Lim's algorithm can be viewed as adding this constraint to the problem (8.3.7) and also not solving for the optimal value of $P(\omega)$ on each step, but instead simply setting $\hat{P}_{k+1}(\omega) = \hat{R}_{k+1}(\omega)$ and $\hat{P}_{k+2}(\omega) = \dfrac{1}{\hat{\lambda}_{k+2}(\omega)}$ on alternate passes.

The most difficult step in this algorithm is calculating the output spectrum estimate $\hat{P}_{k+1}(\omega)$. A direct solution could be found by setting the derivatives of the equation for $P(\omega)$ to zero. This yields the fourth order polynomial:

$$P^2(\omega) - R(\omega)P^3(\omega) + \lambda(\omega)P(\omega) - 1 = 0 \tag{8.3.9}$$

for every frequency component. Formulas for solving equations like this are well known, and are given in any Mathematics Handbook [9] . These formulas, however, are rather complicated, and so another approach might be to simply search for the best value of $P(\omega)$. The objective function in (8.3.7) tends to ∞ as $P(\omega)\to 0$ or $\hat{P}(\omega)\to\infty$; thus somewhere in between these extremes must be a minimizing solution for $\hat{P}_{k+1}(\omega)$ which could be located by any simple search routine. (A good place to start the search is at the last estimate $\hat{P}_k(\omega)$, since then the new estimate will be at least as good as the old one, and the objective function will be sure to decrease on every step.)

4. Penalty Functions for Constrained Minimizations

One of the most popular techniques for solving constrained optimization problems of the form:

$$\hat{x} = \min_{x \in X} L(x) \tag{8.4.1}$$

is to add some multiple μ_k of a penalty function $P(x)$ which penalizes the distance from x to the set X, and then solve the resulting unconstrained minimization problem:

$$\hat{x} - \min_{x} L(x) + \mu_k P(x) \tag{8.4.2}$$

(see, for example, Luenberger [6] .) If μ_k is large, there will be a large penalty attached to choosing x outside X, and thus the unconstrained minimum will actually either be inside the constraint set or very close to the boundary. In the limit as $\mu_k \to \infty$, the unconstrained minimum will be located inside the constraint set X.

One of the problems with this technique is finding an appropriate penalty function which can be easily used inside a gradient optimization search routine. One possible choice which we would suggest would be to use a penalty function which measures the minimum distance to the set; for example:

$$P(x) = d(x,X) = \min_{y \in X} \|y - x\|^2 \tag{8.4.3}$$

Substituting this back into (8.4.2) gives:

$$\begin{aligned} \hat{x} &- \min_{x} L(x) + \mu_k \left\{ \min_{y \in X} \|y - x\|^2 \right\} \\ &- \min_{y \in X ; x} \left[L(x) + \mu_k \|y - x\|^2 \right] \end{aligned} \tag{8.4.4}$$

We have now reduced our original problem (8.4.1) to an optimization problem which looks quite similar to XYMAP. The minimum of this new objective function (8.4.4) could now be located by minimizing over all x without any constraints, then minimizing over y in the constraint set, $y \in X$, iterating back and forth while slowly increasing the penalty weight μ_k.

$$\begin{aligned} \hat{x}_{k+1} &- \min_{x} \left[L(x) + \mu_k \|x - y\|^2 \right] \\ \hat{y}_{k+1} &- \min_{y \in X} \|y - x\|^2 \end{aligned} \tag{8.4.5}$$

Minimizing over x is a standard unconstrained minimization problem; minimizing over y simply involves projecting x back inside the constraint set. As $\mu_k \to \infty$, the penalty for choosing $\hat{x}_{k+1}$ far from the nearest element in X becomes infinite, and thus the unconstrained estimates $\hat{x}_k$ must converge toward the constraint set X.

The chief difficulty we would expect with this approach is that as $\mu_k \to \infty$, the problem becomes ill-behaved and convergence of x and y will be very slow unless we can somehow vary x and y together.

References

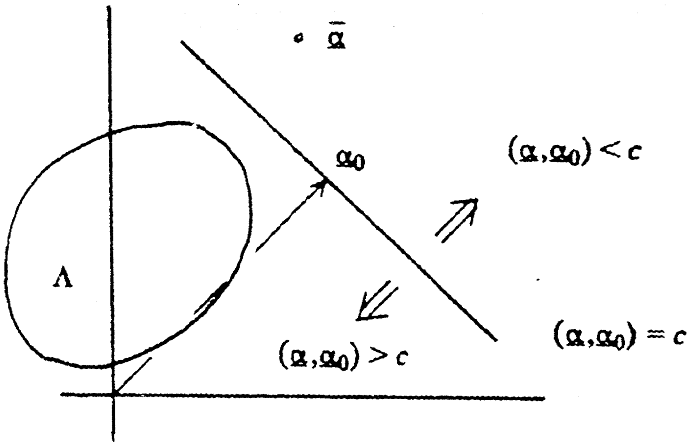
1. S.H. Nawab, *Signal Estimation from Short-Time Spectral Magnitude*, Ph.D. Thesis, M.I.T., Dept. of EECS (May 1982).
2. S.H. Nawab, T.F. Quatieri, and J.S. Lim, "Signal Reconstruction from Short-Time Fourier Transform Magnitude," *IEEE Trans. Acoust. Speech and Sig. Proc.* (submitted July 1982).
3. Naveed Malik, *One and Two Dimensional Maximum Entropy Spectral Estimation*, M.I.T. ScD Thesis (Nov 1981).
4. Jae S. Lim and Naveed A. Malik, "A New Algorithm for Two-Dimensional Maximum Entropy Power Spectrum Estimation," *IEEE Trans. Acoust. Speech, Sig. Proc.* ASSP-29(3), pp.401-413 (June 1981).
5. Germund Dahlquist and Ake Bjorck, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N.J. (1974).
6. David G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, Mass. (1973).
7. Richard S. Varga, *Matrix Iterative Analysis*, Prentice Hall Inc., Englewood Cliffs, N.J. (1962).
8. Stephen W. Lang, *Spectral Estimation for Sensor Arrays*, M.I.T. PhD Thesis (Aug 1981).
9. Milton Abramowitz and Irene A. Stegun, *Handbook of Mathematical Functions*, National Bureau of Standards (Dec 1972).

Appendix A
Proofs of Theorems in Chapter 2

1. Expectations on Convex Sets

Proof of Theorem 2.2.1 Let $p_{\Lambda}(\alpha|z)$ be the *a posteriori* probability of α given z over the domain $\Lambda \subseteq \mathbf{R}^N$. Suppose that $\bar{\alpha} = E_{\Lambda|z}[\alpha|z]$ exists. Let the closed convex hull of Λ be $\bar{\Lambda}$, and let (α,β) be the inner product on $\mathbf{R}^N$. If $\bar{\alpha} \notin \bar{\Lambda}$, then by the geometric Hahn-Banach Theorem [1] there exists a separating hyperplane, described by the equation $(\alpha,\alpha_0)=c$, such that:

$$(\bar{\alpha},\alpha_0) < c < (\alpha,\alpha_0) \qquad \text{for all } \alpha \in \bar{\Lambda} \tag{A.1.1}$$



Separating Hyperplane Theorem

Since the expectation operator is linear, we can compute the conditional expectation of both sides of (A.1.1):

$$(\bar{\alpha},\alpha_0) < c < E_{\Lambda|z}\left[(\alpha,\alpha_0) \,\middle|\, z \right] = (\bar{\alpha},\alpha_0) \tag{A.1.2}$$

But this is a contradiction; thus $\bar{\alpha}$ must belong to the closed convex hull $\bar{\Lambda}$. □

2. Properties of Cross-Entropy

In this section, we prove theorems 2.4.1-5.

Proof of Theorem 2.4.1: We first note that $x \log \frac{x}{y}$ is strictly convex for $x > 0$. To prove this, note that

$$\frac{\partial^2}{\partial x^2}\left[x \log \frac{x}{y} \right] = 1 > 0 \tag{A.2.1}$$

In particular, for any $\alpha \in \Lambda$:

$$\begin{aligned} \Big(\lambda q_1(\alpha) + (1-\lambda) q_2(\alpha) \Big) \log \frac{\lambda q_1(\alpha) + (1-\lambda) q_2(\alpha)}{p(\alpha)} & \\ \leq \lambda q_1(\alpha) \log \frac{q_1(\alpha)}{p(\alpha)} + (1-\lambda)\, q_2(\alpha) \log \frac{q_2(\alpha)}{p(\alpha)} & \end{aligned} \tag{A.2.2}$$

for all $\lambda \in (0,1)$ with equality if and only if $q_1(\alpha) = q_2(\alpha)$. Integrating both sides over Λ gives the desired result. □

Proof of Theorem 2.4.2: Let $\bar{\Lambda}$ be any measurable set. We use the inequality $\log x \leq x - 1$ where equality holds if and only if $x = 1$:

$$\begin{aligned} \int_{\bar{\Lambda}} q(\alpha) \log \frac{q(\alpha)}{p(\alpha)}\, d\alpha &= Q(\bar{\Lambda}) \left\{ \int_{\bar{\Lambda}} \frac{q(\alpha)}{Q(\bar{\Lambda})} \left[\log \frac{q(\alpha)/Q(\bar{\Lambda})}{p(\alpha)/P(\bar{\Lambda})} + \log \frac{Q(\bar{\Lambda})}{P(\bar{\Lambda})} \right] \right\} \\ &\geq Q(\bar{\Lambda}) \left\{ \int_{\bar{\Lambda}} \frac{q(\alpha)}{Q(\bar{\Lambda})} \left[1 - \frac{p(\alpha)/P(\bar{\Lambda})}{q(\alpha)/Q(\bar{\Lambda})} \right] + \log \frac{Q(\bar{\Lambda})}{P(\bar{\Lambda})} \right\} \\ &= Q(\bar{\Lambda}) \log \frac{Q(\bar{\Lambda})}{P(\bar{\Lambda})} \end{aligned} \tag{A.2.3}$$

and equality holds in the second line if and only if $\frac{q(\alpha)}{Q(\bar{\Lambda})} = \frac{p(\alpha)}{P(\bar{\Lambda})}$ almost everywhere in $\hat{\Lambda}$. □

Proof of Theorem 2.4.3: Applying theorem 2.4.2 to each set in the partition $P = \{\Lambda_i\}$:

$$\int_{\Lambda} q(\alpha) \log \frac{q(\alpha)}{p(\alpha)}\, d\alpha = \sum_i \int_{\Lambda_i} q(\alpha) \log \frac{q(\alpha)}{p(\alpha)}\, d\alpha$$

$$\geq \sum_i Q(\Lambda_i) \log \frac{Q(\Lambda_i)}{P(\Lambda_i)} \qquad \text{(A.2.4)}$$

Proof of Theorem 2.4.4: Applying theorem 2.4.2 to each subset Λ_{ij} of the set Λ_i:

$$\sum_i \left[\sum_j Q(\Lambda_{ij}) \log \frac{Q(\Lambda_{ij})}{P(\Lambda_{ij})} \right] \geq \sum_i Q(\bigcup_j \Lambda_{ij}) \log \frac{Q(\bigcup_j \Lambda_{ij})}{P(\bigcup_j \Lambda_{ij})}$$

$$= \sum_i Q(\Lambda_i) \log \frac{Q(\Lambda_i)}{P(\Lambda_i)} \qquad \text{(A.2.5)}$$

Proof of Theorem 2.4.5: See Pinsker [?] or Gallager (chapter 1, [3]).

References

1. David G. Luenberger, *Optimization By Vector Space Methods*, John Wiley & Sons Inc., New York (1969).
2. Pinsker, *Information and Information Stability of Random Variables*, Holden Day, San Francisco (1964). translated by A. Feinstein
3. Robert Gallager, *Information Theory and Reliable Communication*, John Wiley & Sons, New York (1968).

Appendix B

Proofs of Convergence of Iterative Algorithms in Chapter 3

1. General Convergence Theorems

Proof of Theorem 3.9.1

In this appendix we will prove the convergence theorems of chapter 3. Assume that the function $F(\underline{\alpha};\underline{\beta})$ is continuous for all $\underline{\alpha}\in\Lambda$, $\underline{\beta}\in\Phi$. Let $(\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k)$ be the sequence of estimates generated by minimizing $F(\underline{\alpha};\underline{\beta})$ with respect to each variable in turn. Assume that the set of estimates $\{(\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k)\}$ remain within a compact subset of $\Lambda\times\Phi$ (i.e. for finite dimensional spaces, the sequence is bounded.) Because $(\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k)$ is an infinite sequence contained within a compact set, there exists at least one infinite subsequence $(\hat{\underline{\alpha}}'_k, \hat{\underline{\beta}}'_k)$ of the sequence $(\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k)$ which converges to some limit point $(\hat{\underline{\alpha}}_*, \hat{\underline{\beta}}_*)\in\Lambda\times\Phi$:

$$(\hat{\underline{\alpha}}'_k, \hat{\underline{\beta}}'_k) \to (\hat{\underline{\alpha}}_*, \hat{\underline{\beta}}_*) \quad \text{as } k\to\infty \tag{B.1.1}$$

Because $F(\underline{\alpha};\underline{\beta})$ is a continuous function, it is bounded below on every compact subset of $\Lambda\times\Phi$, and thus $F(\hat{\underline{\alpha}}'_k, \hat{\underline{\beta}}'_k)$ must converge monotonically downward to a lower limit F_* as $k\to\infty$. By construction of the iterative algorithm:

$$\begin{aligned} F(\underline{\alpha}\,;\hat{\underline{\beta}}'_k) &\geq F(\hat{\underline{\alpha}}'_{k+1}\,;\hat{\underline{\beta}}'_{k+1}) && \text{for all } \underline{\alpha}\in\Lambda \\ F(\hat{\underline{\alpha}}'_k\,;\underline{\beta}) &\geq F(\hat{\underline{\alpha}}'_k\,;\hat{\underline{\beta}}'_k) && \text{for all } \underline{\beta}\in\Phi \end{aligned} \tag{B.1.2}$$

Because the estimates $(\hat{\underline{\alpha}}'_k, \hat{\underline{\beta}}'_k)$ and $(\hat{\underline{\alpha}}'_{k+1}, \hat{\underline{\beta}}'_{k+1})$ converge to $(\hat{\underline{\alpha}}_*, \hat{\underline{\beta}}_*)$, taking the limit as $k\to\infty$ in (B.1.2) gives:

$$\begin{aligned} F(\underline{\alpha}\,;\hat{\underline{\beta}}_*) &\geq F(\hat{\underline{\alpha}}_*\,;\hat{\underline{\beta}}_*) && \text{for all } \underline{\alpha}\in\Lambda \\ F(\hat{\underline{\alpha}}_*\,;\underline{\beta}) &\geq F(\hat{\underline{\alpha}}_*\,;\hat{\underline{\beta}}_*) && \text{for all } \underline{\beta}\in\Phi \end{aligned} \tag{B.1.3}$$

We have thus proven:

Lemma B.1 Under the conditions stated above, the infinite sequence of estimates $(\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k)$ generated by our iterative algorithm must have at least one limit point $(\hat{\underline{\alpha}}_*, \hat{\underline{\beta}}_*)$. Furthermore, every limit point of this sequence must be an element of the set $\Lambda_\infty \times \Phi_\infty$ defined in (3.9.2). □

Because $F(\underline{\alpha};\underline{\beta})$ is continuous and $\Lambda \times \Phi$ is closed, $\Lambda_\infty \times \Phi_\infty$ must also be closed. Let us define the distance $d((\underline{\alpha}, \underline{\beta}) ; \Lambda_\infty \times \Phi_\infty)$ from a point $(\underline{\alpha}, \underline{\beta}) \in \Lambda \times \Phi$ to the set $\Lambda_\infty \times \Phi_\infty$ by:

$$d((\underline{\alpha}, \underline{\beta}) ; \Lambda_\infty \times \Phi_\infty) = \min_{(\underline{\alpha}_*, \underline{\beta}_*) \in \Lambda_\infty \times \Phi_\infty} \left\{ \left\| \underline{\alpha} - \underline{\alpha}_* \right\|_\Lambda + \left\| \underline{\beta} - \underline{\beta}_* \right\|_\Phi \right\} \tag{B.1.4}$$

where: $\|\cdot\|_\Lambda$ and $\|\cdot\|_\Phi$ are norms on the spaces Λ and Φ respectively

This distance function is continuous in $\underline{\alpha}, \underline{\beta}$ (see Hoffman, [1] page 87). Now suppose the theorem were false. Then there would exist an infinite subsequence $(\hat{\underline{\alpha}}'_k, \hat{\underline{\beta}}'_k)$ of $(\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k)$ and an $\epsilon > 0$ such that:

$$d((\hat{\underline{\alpha}}'_k, \hat{\underline{\beta}}'_k) ; \Lambda_\infty \times \Phi_\infty) \geq \epsilon \quad \text{for all } k \tag{B.1.5}$$

Because the sequence $\{(\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k)\}$ remains within a compact space, there would be at least one infinite subsequence $(\hat{\underline{\alpha}}^*_k, \hat{\underline{\beta}}^*_k)$ of $(\hat{\underline{\alpha}}'_k, \hat{\underline{\beta}}'_k)$ which would converge to a limit point $(\underline{\alpha}_*, \underline{\beta}_*)$:

$$(\hat{\underline{\alpha}}^*_k, \hat{\underline{\beta}}^*_k) \rightarrow (\underline{\alpha}_*, \underline{\beta}_*) \tag{B.1.6}$$

But since the distance measure $d((\underline{\alpha}, \underline{\beta}) ; \Lambda_\infty \times \Phi_\infty)$ is continuous:

$$\lim_{k \to \infty} d((\hat{\underline{\alpha}}^*_k, \hat{\underline{\beta}}^*_k) ; \Lambda_\infty \times \Phi_\infty) = d((\underline{\alpha}_*, \underline{\beta}_*) ; \Lambda_\infty \times \Phi_\infty) \geq \epsilon > 0 \tag{B.1.7}$$

Thus we would have found an infinite converging subsequence of estimates $(\hat{\underline{\alpha}}^*_k, \hat{\underline{\beta}}^*_k)$ whose limit point $(\underline{\alpha}_*, \underline{\beta}_*)$ could not possibly be a member of $\Lambda_\infty \times \Phi_\infty$. This, however, would contradict lemma B.1. The sequence $(\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k)$ must therefore converge to the set $\Lambda_\infty \times \Phi_\infty$.

Note that since $F(\hat{\alpha}_k\,;\hat{\beta}_k)$ is continuous and converges downward to F_* as $k\to\infty$, that for any convergent subsequence $\{(\hat{\alpha}'_k,\hat{\beta}'_k)\}\subseteq\{(\hat{\alpha}_k,\hat{\beta}_k)\}$ with limit point (α_*,β_*):

$$F(\alpha_*\,;\beta_*) = \lim_{k\to\infty} F(\hat{\alpha}'_k\,;\hat{\beta}'_k) = \lim_{k\to\infty} F(\hat{\alpha}_k\,;\hat{\beta}_k) = F_* \tag{B.1.8}$$

Thus if the sequence $(\hat{\alpha}_k,\hat{\beta}_k)$ has multiple limit points, they must all correspond to the same value of $F(\alpha;\beta)=F_*$. □

Proof of Theorem 3.9.2 Assume that the function $F(\alpha;\beta)$ has a continuous first derivative in α and β for all $\alpha,\beta\in\Lambda\times\Phi$. Applying equation (2.10.2) to the equation (3.9.2) defining the limit set $\Lambda_\infty\times\Phi_\infty$, we can show that the derivative of F must either be zero at every point $(\alpha_*,\beta_*)\in\Lambda_\infty\times\Phi_\infty$, or else if (α_*,β_*) is on the boundary of $\Lambda\times\Phi$, then the derivative of F must be inwardly normal to the boundary. In the latter case, the point (α_*,β_*) would usually be a local minimum on the boundary of $\Lambda_\infty\times\Phi_\infty$. □

Proof of Theorem 3.9.3 Assume that $F(\alpha;\beta)$ is continuously differentiable and convex on $\Lambda\times\Phi$, and that $\Lambda\times\Phi$ is a convex and closed set. Suppose that the sequence of estimates $(\hat{\alpha}_k,\hat{\beta}_k)$ is compact. Theorem 3.9.1 then guarantees that the set $\Lambda_\infty\times\Phi_\infty$ is nonempty and contains all the limit points of the sequence. To show that every element in $\Lambda_\infty\times\Phi_\infty$ is in fact a finite global minimizer of F over $\Lambda\times\Phi$, let (α_*,β_*) be any element in $\Lambda_\infty\times\Phi_\infty$ which is not a global minimizer. Then there must be some element $(\alpha',\beta')\in\Lambda\times\Phi$ such that

$$F(\alpha'\,;\beta') < F(\alpha_*\,;\beta_*) \tag{B.1.9}$$

Because F is convex, property (1.5.4) guarantees that:

$$\left(\frac{\partial F(\alpha_*\,;\beta_*)^T}{\partial\alpha} \quad \frac{\partial F(\alpha_*\,;\beta_*)^T}{\partial\beta}\right)\begin{pmatrix}\alpha'-\alpha_*\\ \beta'-\beta_*\end{pmatrix} \le F(\alpha'\,;\beta') - F(\alpha_*\,;\beta_*) \tag{B.1.10}$$

unique finite global minimum of F. Thus if the iteration sequence is compact, it will converge to the unique global minimizer. If, on the other hand, F has a finite global minimum, and the spaces are finite dimensional, then theorem 2.10.3 guarantees that every level set is bounded and thus compact. This then guarantees that the sequence $\{\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k\}$ is compact, and thus is guaranteed .o converge to the unique global minimum in $\Lambda_x \times \Phi_x$. If F is uniformly convex, then convergence to the unique global minimizer is guaranteed since all level sets are bounded.

Finally, let $g: \mathbf{R} \to \mathbf{R}$ be a monotonically strictly increasing function. Suppose that all the conditions of theorem 3.9.3 hold, except that instead of requiring $F(\underline{\alpha};\underline{\beta})$ to be convex, we require $g(F(\underline{\alpha};\underline{\beta}))$ to be convex. Then we can apply exactly the proof above to the function $g(F(\underline{\alpha};\underline{\beta}))$ to show that the iteration converges to the global minimum of $g(F(\underline{\alpha};\underline{\beta}))$. Since g is monotonic, however, this is equivalent to converging to the global minimum of $F(\underline{\alpha};\underline{\beta})$.

2. Convergence of PARMAP

The only non-trivial part of the proof of the convergence of PARMAP is the application of theorem 3.9.2. Suppose that $p(\underline{X}, \underline{\phi})$ and $\tilde{H}(\underline{\psi}, \underline{\phi})$ are continuously differentiable in $\underline{\phi}, \underline{\psi} \in \Phi$. Theorem 3.9.2 guarantees that for every point $(\hat{\underline{\psi}}, \hat{\underline{\phi}}) \in \Phi_x \times \Phi_x$, and for all tangent vectors $\underline{h}_{\psi}$ and $\underline{h}_{\phi}$ of Φ at $(\hat{\underline{\psi}}, \hat{\underline{\phi}})$

$$\frac{\partial \tilde{H}(\hat{\underline{\psi}}, \hat{\underline{\phi}})^T}{\partial \underline{\psi}} \underline{h}_{\psi} \geq 0 \qquad \text{(B.2.1)}$$
$$\frac{\partial \tilde{H}(\hat{\underline{\psi}}, \hat{\underline{\phi}})^T}{\partial \underline{\phi}} \underline{h}_{\phi} \geq 0$$

However, the solution to $\min_{\underline{\psi} \in \Phi} \tilde{H}(\underline{\psi}, \hat{\underline{\phi}})$ is $\hat{\underline{\psi}} = \hat{\underline{\phi}}$. Combining this with the fact that $\tilde{H}(\hat{\underline{\phi}}, \hat{\underline{\phi}}) = -\log p(\underline{X}, \hat{\underline{\phi}})$:

$$\frac{\partial \log p(X,\hat{\phi})^{T}}{\partial \phi} h_{\phi} = \left(\frac{\partial \bar{H}(\hat{\psi},\hat{\phi})^{T}}{\partial \psi} + \frac{\partial \bar{H}(\hat{\psi},\hat{\phi})^{T}}{\partial \phi} \right) h_{\phi}$$

$$\geq 0 \qquad \text{(B.2.2)}$$

for all tangent vectors h_{ϕ} of Φ at $\hat{\phi}$.

To prove that the difference between successive signal density estimates is asymptotically zero, note that:

$$H(\hat{q}_{X_k},\hat{\phi}_k) - H(\hat{q}_{X_{k+1}},\hat{\phi}_k) = \int_{X} p_{X|\Phi}(x|\hat{\phi}_k) \log \frac{p_{X|\Phi}(x|\hat{\phi}_k)}{p_{X|\Phi}(x|\hat{\phi}_{k+1})} dx \qquad \text{(B.2.3)}$$

Let $\tilde{X}$ be any measurable subset of X, and partition X into sets $\tilde{X}$ and $\tilde{X}^{C} = X - \tilde{X}$. Applying theorem 2.4.3:

$$H(\hat{q}_{X_k},\hat{\phi}_k) - H(\hat{q}_{X_{k+1}},\hat{\phi}_k)$$

$$\geq p_{X|\Phi}(\tilde{X}|\hat{\phi}_k) \log \frac{p_{X|\Phi}(\tilde{X}|\hat{\phi}_k)}{p_{X|\Phi}(\tilde{X}|\hat{\phi}_{k+1})} + p_{X|\Phi}(\tilde{X}^{C}|\hat{\phi}_k) \log \frac{p_{X|\Phi}(\tilde{X}^{C}|\hat{\phi}_k)}{p_{X|\Phi}(\tilde{X}^{C}|\hat{\phi}_{k+1})}$$

$$\geq 0 \qquad \text{(B.2.4)}$$

As $k \to \infty$, the left hand side goes to zero, which implies that $p_{X|\Phi}(\tilde{X}|\hat{\phi}_k) - p_{X|\Phi}(\tilde{X}|\hat{\phi}_{k+1}) \to 0$ as $k \to \infty$.

3. Convergence of MCEM

Proof of Theorem 3.9.4

a) To prove that the sequence of densities $\hat{q}_{X_k}$, $\hat{q}_{\Phi_k}$ is well defined, note simply that $\hat{q}_{X_k}(x) \geq 0$, $\hat{q}_{\Phi_k}(\phi) \geq 0$ for all $x \in X$, $\phi \in \Phi$, and that the normalization constants $c_{X_{k+1}}$, $c_{\Phi_{k+1}}$ are all strictly positive and less than or equal to one:

$$\infty > H(\hat{q}_{X_0},\hat{q}_{\Phi_0}) \geq H(\hat{q}_{X_k},\hat{q}_{\Phi_k}) = -\log c_{\Phi_k} \geq H_{*} \geq -\log p(X,\Phi) \geq 0$$

and similarly for c_{x_k}.

b) This was proven in chaper 3, section 2.

c) Let $\bar{\Phi}$ be any measurable subset of Φ, and $\bar{\Phi}^C = \Phi - \bar{\Phi}$. (Thus $\bar{\Phi} \cap \bar{\Phi}^C = \varnothing$, $\bar{\Phi} \cup \bar{\Phi}^C = \Phi$.) By theorem 2.4.3,

$$\begin{aligned}
\infty &> H(\hat{q}_{X_{k+1}},\hat{q}_{\Phi_k}) - H(\hat{q}_{X_{k+1}},\hat{q}_{\Phi_{k+1}}) \\
&= \int_{\Phi} \hat{q}_{\Phi_k}(\phi) \log \frac{\hat{q}_{\Phi_k}(\phi)}{\hat{q}_{\Phi_{k+1}}(\phi)} \, d\phi \geq 0 && \text{(B.3.1)} \\
&\geq \hat{Q}_{\Phi_k}(\bar{\Phi}) \log \frac{\hat{Q}_{\Phi_k}(\bar{\Phi})}{\hat{Q}_{\Phi_{k+1}}(\bar{\Phi})} + \hat{Q}_{\Phi_k}(\bar{\Phi}^C) \log \frac{\hat{Q}_{\Phi_k}(\bar{\Phi}^C)}{\hat{Q}_{\Phi_{k+1}}(\bar{\Phi}^C)} \\
&\geq 0 && \text{(B.3.2)}
\end{aligned}$$

By assumption $\hat{Q}_{\Phi_0}(\bar{\Phi})>0$ for any measurable subset $\bar{\Phi}$ of Φ. But then for $k=0$, the second to last line in (B.3.1) can be finite only if $\hat{Q}_{\Phi_1}(\bar{\Phi})>0$. Applying the argument recursively, $\hat{Q}_{\Phi_k}(\bar{\Phi})>0$ for all k and for every measurable subset $\bar{\Phi}$ of Φ. A similar argument shows that $\hat{Q}_{X_k}(\bar{X})>0$ for all measurable subsets $\bar{X}$ of X.

d) As $k \to \infty$, then $H(\hat{q}_{X_{k+1}},\hat{q}_{\Phi_k}) - H(\hat{q}_{X_{k+1}},\hat{q}_{\Phi_{k+1}}) \to 0$. But from (B.3.1) this implies that:

$$\lim_{k \to \infty} \hat{Q}_{\Phi_k}(\bar{\Phi}) - \hat{Q}_{\Phi_{k+1}}(\bar{\Phi}) = 0 \qquad \text{(B.3.3)}$$

for any measurable subset $\bar{\Phi}$ of Φ. The proof that $\left(\hat{Q}_{X_k}(\bar{X}) - \hat{Q}_{X_{k+1}}(\bar{X})\right) \to 0$ as $k \to \infty$ is similar.

e) To derive this upper bound on the measures $\hat{Q}_{X_k}$, $\hat{Q}_{\Phi_k}$, partition the space $X \times \Phi$ into a measurable region Ψ and its complement Ψ^C ($\Psi \cap \Psi^C = \varnothing$ and $\Psi \cup \Psi^C = X \times \Phi$.) By theorem 2.4.3:

$$H(\hat{Q}_{X_k},\hat{Q}_{\Phi_k}) \geq \delta_k \log \frac{\delta_k}{\epsilon} + (1-\delta_k) \log \frac{1-\delta_k}{1-\epsilon} - \log p(X,\Phi)$$

$$\text{where: } \delta_k = \int_{\Psi} \hat{q}_{X_k}(\mathbf{x}) \hat{q}_{\Phi_k}(\boldsymbol{\phi}) \, d\mathbf{x} d\boldsymbol{\phi} \tag{B.3.4}$$

$$\epsilon = \frac{1}{p(X,\Phi)} \int_{\Psi} p(\mathbf{x},\boldsymbol{\phi}) \, d\mathbf{x} d\boldsymbol{\phi}$$

But:

$$\begin{aligned} &H(\hat{q}_{X_0},\hat{q}_{\Phi_0}) \geq H(\hat{q}_{X_k},\hat{q}_{\Phi_k}) \\ &\delta_k \log \delta_k + (1-\delta_k) \log (1-\delta_k) \geq -\log 2 \\ &-(1-\delta_k) \log (1-\epsilon) \geq 0 \end{aligned} \tag{B.3.5}$$

Combining all these with (B.3.4) gives:

$$\delta_k \leq \frac{H(\hat{q}_{X_0},\hat{q}_{\Phi_0}) + \log p(X,\Phi) + \log 2}{-\log \epsilon} \tag{B.3.6}$$

This formula provides an upper bound on the measure δ_k assigned by the separable density to the set Ψ as a function of the measure ϵ assigned to the set by the original density $p(\mathbf{x},\boldsymbol{\phi})$. Note that this bound applies to any measures with cross-entropy less than $H(q_{X_0},q_{\Phi_0})$. Also note that we could choose:

$$\Psi = \left\{ (\mathbf{x},\boldsymbol{\phi}) \in X \times \Phi \;\middle|\; \|\mathbf{x}\| \leq T \text{ and } \|\boldsymbol{\phi}\| \leq T \right\} \tag{B.3.7}$$

$$\Psi^C = \left\{ (\mathbf{x},\boldsymbol{\phi}) \in X \times \Phi \;\middle|\; (\mathbf{x},\boldsymbol{\phi}) \notin \Psi \right\}$$

Since $p(x,\phi)$ is assumed to be integrable over $X \times \Phi$, as $T \to \infty$ we must have $\epsilon \to 0$. Thus by choosing T large enough, we can make the upper bound on the tail probability δ_k arbitrarily small for all k. Thus $\hat{q}_{X_k}(x)$, $\hat{q}_{\Phi_k}(\phi)$ must be stochastically bounded.

f) By the Helly Selection Theorem (see [2 volume 2] or [3]) every stochastically bounded sequence of measures $\hat{Q}_{X_k}$, $\hat{Q}_{\Phi_k}$ has at least one convergent subsequence with a limit $\overline{Q}_X, \overline{Q}_\Phi$, and every such limit must be a proper measure.

g) Let X_i, Φ_i be any finite partition of X, Φ. Let $\hat{Q}_{X_{k_n}}$, $\hat{Q}_{\Phi_{k_n}}$ be a convergent subsequence with limit $\overline{Q}_X$, $\overline{Q}_\Phi$. By theorem 2.4.3:

$$H(\hat{Q}_{X_{k_n}},\hat{Q}_{\Phi_{k_n}}) \geq \sum_{i,j} \hat{Q}_{X_{k_n}}(X_i)\hat{Q}_{\Phi_{k_n}}(\Phi_j) \log \frac{\hat{Q}_{X_{k_n}}(X_i)\hat{Q}_{\Phi_{k_n}}(\Phi_j)}{P(X_i,\Phi_j)} \tag{B.3.8}$$

Taking the limit as $k_n \to \infty$,

$$\hat{Q}_{X_{k_n}}(X_i) \to \overline{Q}_X(X_i) \quad \text{and} \quad \hat{Q}_{\Phi_{k_n}}(\Phi_j) \to \overline{Q}_\Phi(\Phi_j) \tag{B.3.9}$$

and:

$$\liminf_{k_n \to \infty} H(\hat{Q}_{X_{k_n}},\hat{Q}_{\Phi_{k_n}}) = \liminf_{k \to \infty} H(\hat{Q}_{X_k},\hat{Q}_{\Phi_k}) = H_\bullet \tag{B.3.10}$$

Thus taking the limit as $k \to \infty$ of both sides of (B.3.8) gives:

$$H_\bullet \geq \sum_{i,j} \overline{Q}_X(X_i)\overline{Q}_\Phi(\Phi_j) \log \frac{\overline{Q}_X(X_i)\overline{Q}_\Phi(\Phi_j)}{P(X_i,\Phi_j)} \tag{B.3.11}$$

Since this is true for all partitions P of $X \times \Phi$:

$$H_\bullet \geq \sup_P \sum_{i,j} \overline{Q}_X(X_i)\overline{Q}_\Phi(\Phi_j) \log \frac{\overline{Q}_X(X_i)\overline{Q}_\Phi(\Phi_j)}{P(X_i,\Phi_j)} \tag{B.3.12}$$

But because $\underline{\alpha}_*$ must satisfy $\min_{\underline{\alpha} \in \Lambda} F(\underline{\alpha} ; \underline{\beta}_*) = F(\underline{\alpha}_* ; \underline{\beta}_*)$, and because F is continuously differentiable, theorem 3.9.2 guarantees that

$$\frac{\partial F(\underline{\alpha}_* ; \underline{\beta}_*)^{\mathrm{T}}}{\partial \underline{\alpha}} \underline{h} \geq 0 \tag{B.1.11}$$

for all sequentially tangent vectors $\underline{h}$ of Λ at $\underline{\alpha}_*$. Because Λ is convex, the vector $\underline{h}_\alpha = \underline{\alpha}' - \underline{\alpha}_*$ must be a tangent vector. To see this, substitute $\hat{\underline{\alpha}}_k = t_k \underline{\alpha} + (1 - t_k) \underline{\alpha}_*$ and $t_k = (\frac{1}{2})^k$ into the definition of tangent vectors in (2.10.3), giving:

$$\lim_{k \to \infty} \frac{\hat{\underline{\alpha}}_k - \underline{\alpha}_*}{t_k} = \underline{\alpha}' - \underline{\alpha}_* = \underline{h}_\alpha \qquad \text{and} \qquad \lim_{k \to \infty} t_k = 0 \tag{B.1.12}$$

Thus:

$$\frac{\partial F(\underline{\alpha}_* ; \underline{\beta}_*)}{\partial \underline{\alpha}} (\underline{\alpha}' - \underline{\alpha}_*) \geq 0 \tag{B.1.13}$$

Similarly, because $\underline{\beta}_*$ must satisfy $\min_{\underline{\beta} \in \Phi} F(\underline{\alpha}_* ; \underline{\beta}) = F(\underline{\alpha}_* ; \underline{\beta}_*)$, we can show that

$$\frac{\partial F(\underline{\alpha}_* ; \underline{\beta}_*)}{\partial \underline{\beta}} (\underline{\beta}' - \underline{\beta}_*) \geq 0 \tag{B.1.14}$$

Combining (B.1.10), (B.1.13) and (B.1.14) gives:

$$0 \leq F(\underline{\alpha}' ; \underline{\beta}') - F(\underline{\alpha}_* ; \underline{\beta}_*) \tag{B.1.15}$$

But this contradicts our assumption in (B.1.9) that $(\underline{\alpha}_*, \underline{\beta}_*)$ was not a global minimizer of F over $\Lambda \times \Phi$. Thus every element of $\Lambda_\infty \times \Phi_\infty$ is a global minimizer; since every finite global minimizer must belong to $\Lambda_\infty \times \Phi_\infty$, this set must in fact be exactly the closed, convex set of global minimizers of F. The iterative algorithm therefore converges to the set of global minimizers.

If F is strictly convex and continuously differentiable, then it can have at most one global minimizer on $\Lambda \times \Phi$. If the sequence of estimates $(\hat{\underline{\alpha}}_k, \hat{\underline{\beta}}_k)$ is compact, then we have shown that it must converge to a non-empty set $\Lambda_\infty \times \Phi_\infty$, which will contain the

$$= H(\overline{Q}_X, \overline{Q}_\Phi)$$

The proof of the other assertions follows by noting that $\hat{Q}_{X_{k_n}} Q_\Phi$ converges to $\overline{Q}_X Q_\Phi$, and $Q_x \hat{Q}_{\Phi_{k_n}}$ converges to $Q_x \overline{Q}_\Phi$.

h) The upper bound given in e) holds for any measure whose cross-entropy is lower than $H(\hat{q}_{X_0}, \hat{q}_{\Phi_0})$.

Proof of Theorem 3.9.5

a) Follows from theorem 3.9.4.

b) By direct calculation, minimizing $H(q_X, \hat{q}_{\Phi_k})$ over q_X gives:

$$\begin{aligned} \log \hat{q}_{X_{k+1}}(x_i) &= \sum_{j=1}^{M} \hat{q}_{\Phi_k}(\phi_j) \log \frac{p(x_i, \phi_j)}{\hat{q}_{\Phi_k}(\phi_j)} - \log c_{x_{k+1}} \\ &= \sum_{j=1}^{N} \hat{q}_{\Phi_k}(\phi_j) \log p(x_i, \phi_j) - \log c'_{x_{k+1}} \end{aligned} \qquad (B.3.13)$$

where $c_{x_{k+1}}$, $c'_{x_{k+1}}$ are normalizing constants:

$$c'_{x_{k+1}} = \sum_{i=1}^{N} \exp\left[\sum_{j=1}^{M} \hat{q}_{\Phi_k}(\phi_j) \log p(x_i, \phi_j) \right] \qquad (B.3.14)$$

Using the inequality

$$\epsilon \leq p(x_i, \phi_j) \leq 1 \qquad \text{for all } i, j \qquad (B.3.15)$$

in equations (B.3.13) and (B.3.14) gives:

$$c'_{x_{k+1}} \leq N \qquad (B.3.16)$$

and

$$\hat{q}_{X_{k+1}}(x_i) \geq \frac{1}{N}\epsilon \tag{B.3.17}$$

The inequality $\hat{q}_{\Phi_{k+1}}(\phi_j) \geq \frac{1}{M}\epsilon$ can be proven similarly.

c) Proof is identical to property d) of theorem 3.9.4.

d) Because the constraint spaces have a finite number of points, the space of all probability densities q_X and q_Φ is finite dimensional. Because of the constraints (3.9.16), the space is also bounded and closed. Therefore the sequence $\hat{q}_{X_k}$, $\hat{q}_{\Phi_k}$ is compact, and the Bolzano-Weierstrass theorem guarantees that there is at least one convergence subsequence $\hat{q}_{X_{k_i}}$, $\hat{q}_{\Phi_{k_i}}$ with limit $\bar{q}_X$, $\bar{q}_\Phi$.

e) Since the inequalities of property b) are satisfied for all k, they must be satisfied in the limit as well.

f) The cross-entropy is an analytic function of q_X and q_Φ for all strictly positive densities. Since the limit is strictly positive, $H(q_X, q_\Phi)$ is continuous in a neighborhood about $(\bar{q}_X, \bar{q}_\Phi)$. Let q_X, q_Φ be any densities. Taking the limit in the following inequalities:

$$H(\hat{q}_{X_{k_i}}, q_\Phi) \geq H(\hat{q}_{X_{k_i}}, \hat{q}_{\Phi_{k_i}}) \tag{B.3.18}$$

$$H(q_X, \hat{q}_{\Phi_{k_i}}) \geq H(\hat{q}_{X_{k_i+1}}, \hat{q}_{\Phi_{k_i}})$$

as $i \to \infty$ then gives:

$$H(\bar{q}_X, q_\Phi) \geq H(\bar{q}_X, \bar{q}_\Phi) \tag{B.3.19}$$

$$H(q_X, \bar{q}_\Phi) \geq H(\bar{q}_X, \bar{q}_\Phi)$$

Since q_X, q_Φ were arbitrary, the result follows.

g) Caculating the derivative of the Lagrangian:

$$\frac{\partial L_{x,\phi}(\bar{q}_X,\bar{q}_\Phi)}{\partial q_X(x_i)} = 1 + \sum_{j=1}^{M} \bar{q}_\Phi(\phi_j) \log \frac{\bar{q}_X(x_i)\bar{q}_\Phi(\phi_j)}{p(x_i,\phi_j)} + \lambda_x \tag{B.3.20}$$

$$= 1 - \log \bar{c}_x + \lambda_x$$

and similarly:

$$\frac{\partial H(\bar{q}_X,\bar{q}_\Phi)}{\partial q_\Phi(\phi_j)} = 1 - \log \bar{c}_\phi + \lambda_\phi \tag{B.3.21}$$

where we have used the fact that $\bar{q}_X$, $\bar{q}_\Phi$ form a stationary point of the iterative algorithm, and therefore satisfy:

$$\log \bar{q}_X(x_i) = \sum_{j=1}^{M} \bar{q}_\Phi(\phi_j) \log \frac{p(x_i,\phi_j)}{\bar{q}_\Phi(\phi_j)} - \log \bar{c}_x \tag{B.3.22}$$

$$\log \bar{q}_\Phi(\phi_j) = \sum_{i=1}^{N} \bar{q}_X(x_i) \log \frac{p(x_i,\phi_j)}{\bar{q}_X(x_i)} - \log \bar{c}_\phi$$

Choosing Lagrange multiplier values $\lambda_x = -1 + \log \bar{c}_x$ and $\lambda_\phi = -1 + \log \bar{c}_\phi$ then proves the result.

Proof of Lemma 3.9.6.2 Substituting formulas (3.9.19) into the cross-entropy expression gives:

$$H(\alpha,\beta) = -\log c_x - \log c_\phi + \alpha^T \overline{t(x)} + \beta^T \overline{\pi(\phi)} - \overline{t(x)}^T\, \overline{\pi(\phi)} \tag{B.3.23}$$

$$\text{where } \overline{t(x)} = E_X\left[\, t(x) \;\middle|\; q_X \,\right] = \frac{\partial}{\partial \alpha} \log c_x$$

$$\overline{\pi(\phi)} = E_\Phi\left[\, \pi(\phi) \;\middle|\; q_\Phi \,\right] = \frac{\partial}{\partial \beta} \log c_\phi$$

Since c_x and c_ϕ are analytic functions of α, β in the interior of the natural parameter space, the cross-entropy $H(\alpha,\beta)$ also must be analytic in the interior.

Proof of Theorem 3.9.6 Let $\tilde{\Lambda}_\alpha \times \tilde{\Lambda}_\beta$ be a compact set contained entirely within the interior of $\Lambda_\alpha \times \Lambda_\beta$ which contains all the estimates $(\hat{\underline{\alpha}}_t, \hat{\underline{\beta}}_t)$. Since $H(\alpha, \underline{\beta})$ is analytic in the interior of $\Lambda_\alpha \times \Lambda_\beta$, it is analytic at all points in $\tilde{\Lambda}_\alpha \times \tilde{\Lambda}_\beta$. Since the sequence $(\hat{\underline{\alpha}}_t, \hat{\underline{\beta}}_t)$ remains within the compact set $\tilde{\Lambda}_\alpha \times \tilde{\Lambda}_\beta$, theorem 3.9.1 immediately applies, guaranteeing convergence tu stationary points of the algorithm. Theorem 3.9.2 also applies; since any limit point of the sequence is inside $\tilde{\Lambda}_\alpha \times \tilde{\Lambda}_\beta$, it is in the interior of $\Lambda_\alpha \times \Lambda_\beta$ and therefore must be a critical point of the cross-entropy. Finally, if $\hat{\underline{\beta}}(\alpha)$ is the value of $\underline{\beta}$ which minimizes $H(\alpha, \underline{\beta})$ for any $\underline{\alpha}$, then at any limit point $\hat{\underline{\alpha}}$:

$$\begin{aligned} \frac{dF(\hat{\underline{\alpha}})}{d\underline{\alpha}} &= \frac{d}{d\underline{\alpha}} \left[\min_{\underline{\beta}} H(\hat{\underline{\alpha}}, \underline{\beta}) \right] \\ &= \left[\frac{\partial}{\partial \underline{\alpha}} H(\hat{\underline{\alpha}}, \underline{\beta}) + \frac{\partial \hat{\underline{\beta}}(\hat{\underline{\alpha}})^T}{\partial \underline{\alpha}} \frac{\partial}{\partial \underline{\beta}} H(\hat{\underline{\alpha}}, \underline{\beta}) \right]_{\underline{\beta} = \hat{\underline{\beta}}(\hat{\underline{\alpha}})} \\ &= \left[0 + \mathrm{R}_t \cdot 0 \right] \\ &= 0 \end{aligned} \tag{B.3.24}$$

The proof that $\frac{dG(\hat{\underline{\beta}})}{d\underline{\beta}} = 0$ at any limit $\hat{\underline{\beta}}$ is similar.

References

1. Kenneth Hoffman, *Analysis in Euclidean Space*, Prentice Hall Inc., Englewood Cliffs, N.J. (1975).
2. William Feller, *An Introduction to Probability Theory and Its Applications*, John Wiley & Sons, New York (1966).
3. Patrick Billingsley, *Convergence of Probability Measures*, John Wiley & Sons, New York (1968).

Appendix C

Proofs of Theorems in Chapter 4

1. Exponential Density

MCEM

We first show that the marginal density p(X,ϕ) is strictly log concave provided that some U_i is finite:

$$\frac{\partial^2 \log p(X,\phi)}{\partial\phi^2} = -\sum_{i=1}^{N} \frac{(U_i-L_i)^2}{4\sinh^2\left(\frac{1}{2}(U_i-L_i)\right)} \leq 0 \tag{C.1.1}$$

where equality holds if and only if all $U_i=\infty$. Log concavity follows from the discussion in chapter 2, section 10.2. Now to prove that the MCEM algorithm converges. By direct substitution, it is straightforward to show that

$$\hat{q}_{X_i}(x_i) = \begin{cases} c_{x_i}\exp\left(-\hat{\phi}x_i\right) & \text{for } L_i \leq x_i \leq U_i \\ 0 & \text{else} \end{cases} \tag{C.1.2}$$

$$\hat{q}_{\Phi}(\phi) = c_{\phi}\phi^N \exp\left[-\phi\left(\epsilon_0+\sum_{i=1}^{N}\hat{x}_i\right)\right]$$

where $\hat{\phi}$ and $\hat{x}_i$ are parameters of these densities. Restricting the minimization of the cross-entropy to densities of this form (C.1.2) will not change the sequence of estimates generated nor the final solution. We can then view the MCEM algorithm as iteratively minimizing $H(\hat{\phi},\hat{x}) \equiv H(\hat{q}_X,\hat{q}_\Phi)$ with respect to $\hat{\phi}$ and $\hat{x}$:

$$\begin{aligned} \hat{\phi}_{k+1} &- \min_{\phi\geq 0} H(\phi,\hat{x}_k) \\ \hat{x}_{k+1} &- \min_{L_i\leq x_i\leq U_i} H(\hat{\phi}_{k+1},x) \end{aligned} \tag{C.1.3}$$

By direct calculation we can show that:

$$F(\hat{\underline{x}}) \equiv \min_{\phi\geq 0} H(\phi,\hat{\underline{x}}) = \min_{q_X} H(q_X,\hat{q}_\Phi) = K - \log\left[\,\hat{\phi}\, p(X,\hat{\phi})\,\right] \qquad \text{(C.1.4)}$$

$$\text{where: } \hat{\phi} = E_\Phi\left[\,\phi\,\middle|\,\hat{q}_\Phi\,\right] = \frac{N+1}{\epsilon_0+\sum_{i=1}^{N}\hat{x}_i}$$

where K is a fixed constant. Each MCEM iteration decreases $H(\hat{q}_{X_{k+1}},\hat{q}_{\Phi_k})$, and thus must also increase $\hat{\phi}_k\, p(X,\hat{\phi}_k)$. Note that $\phi\, p(\phi) \to 0$ as $\phi \to 0$ or $\phi \to \infty$. But:

$$\phi\, p(X,\phi) = p(X\,|\phi)\left[\,\phi\, p(\phi)\,\right] \leq \phi\, p(\phi) \qquad \text{(C.1.5)}$$

Since $\hat{\phi}_k\, p(X,\hat{\phi}_k)$ increases on each iteration, $\hat{\phi}_k\, p(\hat{\phi}_k)$ must be bounded away from zero, which in turn implies that $\hat{\phi}_k$ is bounded above and bounded away from zero. From equation (4.2.13) for $\hat{x}_{i,k}$, we can show that $\hat{x}_{i,k} \leq L_i + \frac{1}{\hat{\phi}_k}$ and thus $\hat{x}_{i,k}$ must also be bounded above. Since the sequence $\hat{\underline{x}}_k$, $\hat{\phi}_k$ is finite dimensional and bounded, by the Bolzano-Weierstrass theorem it must be compact and have at least one convergent subsequence with limit point $\underline{x}_*$, ϕ_*. Since $\hat{\phi}_k$ is bounded away from zero, $\phi_* > 0$ also. Thus ϕ_* is in the interior of the natural parameter space $\Phi = [0,\infty)$, and theorem 3.9.6 guarantees that the limit point $\underline{x}_*$, ϕ_* is a stationary point of the algorithm, a critical point of $H(\phi,\underline{x})$ and a critical point of $F(\underline{x})$. This last statement, however, implies:

$$\begin{aligned} 0 &= \frac{dF(\underline{x}_*)}{d\underline{x}} \\ &= -\left[\frac{\partial \log \phi_*\, p(X,\phi_*)}{\partial \phi}\right]\left[\frac{\partial \phi_*}{\partial \underline{x}}\right] \qquad \text{(C.1.6)} \\ &\quad \text{where: } \phi_* = \frac{N+1}{\epsilon_0+\sum_{i=1}^{N} x_{i*}} \end{aligned}$$

But $\frac{\partial\phi}{\partial x_i} = -\frac{\phi^2}{N+1}$ is nonzero, and thus:

$$\frac{\partial\left[\phi_* \, \mathrm{p}(X,\phi_*)\right]}{\partial\phi} = 0 \qquad \text{(C.1.7)}$$

and ϕ_* is a critical point of $\phi\,\mathrm{p}(X,\phi)$. However, we showed in (C.1.1) that $\mathrm{p}(X,\phi)$ is log concave. Thus $\phi\,\mathrm{p}(X,\phi)$ is strictly log concave and its derivative can be zero at only one point. Thus the limit ϕ_* must be unique, and since x_* is related to ϕ_* by equation (4.2.13), x_* must also be unique. MCEM must therefore have a unique solution, and the iterative algorithm is guaranteed to find it.

An alternate convergence proof can also be stated in terms of contraction mappings (see chapters 4 and 12 in Ortega and Rheinboldt [1].) The MCEM algorithm can be viewed as defining a mapping T_x from the parameter space Φ to a sum of sample values, and a mapping T_x back again:

$$T_\phi\left(\sum_{i=1}^{N} x_i\right) = \frac{N+1}{\epsilon_0 + \sum_{i=1}^{N} x_i} \qquad \text{(C.1.8)}$$

$$T_x(\phi) = \sum_{i=1}^{N}\left[L_i + \frac{1}{\phi}\left[1 - \frac{\delta_i e^{-\delta_i}}{1-e^{-\delta_i}}\right]\right]$$

The MCEM algorithm simply calculates

$$\hat{\phi}_{k+1} = T_\phi\left(\sum_{i=1}^{N} \hat{x}_{i,k}\right) \qquad \text{(C.1.9)}$$

$$\sum_{i=1}^{N} \hat{x}_{i,k+1} = T_x(\hat{\phi}_{k+1})$$

Let x' and x'' be any two sets of samples values. Since T_ϕ and T_x are both continuously differentiable mappings, there must be an intermediate sample value $\tilde{x} = \lambda x' + (1-\lambda)x''$ for some $\lambda \in (0,1)$ such that:

$$\left|T_x(T_\phi(\sum x_i')) - T_x(T_\phi(\sum x_i''))\right| \le \left|\frac{\partial T_x(T_\phi(\sum \tilde{x}_i))}{\partial(\sum x_i)}\right| \left|\sum x_i' - \sum x_i''\right| \qquad \text{(C.1.10)}$$

However,

$$\frac{\partial T_x(T_\phi(\sum x_i))}{\partial(\sum x_i)} = \frac{\partial T_x(\tilde{\phi})}{\partial\phi}\,\frac{\partial T_\phi(\sum \tilde{x}_i)}{\partial(\sum x_i)}$$

$$= \left[\sum_{i=1}^{N}\frac{1}{\tilde{\phi}^2}\left[-1+\frac{\delta_i^2}{4\sinh^2(\frac{1}{2}\delta_i)}\right]\right]\left[-\frac{N+1}{\left(\epsilon_0+\sum_{i=1}^{N}\tilde{x}_i\right)^2}\right]$$

where: $\delta_i = \tilde{\phi}(U_i - L_i)$ and $\tilde{\phi} = T_\phi(\sum \tilde{x}_i)$

$$= \frac{1}{N+1}\sum_{i=1}^{N}\left[1-\frac{\delta_i^2}{2\sinh^2(\frac{1}{2}\delta_i)}\right] \qquad \text{(C.1.11)}$$

It is easy to show that $\frac{\delta_i^2}{4\sinh^2(\frac{1}{2}\delta_i)}$ is a monotonically strictly decreasing function for $\delta_i \geq 0$ which equals 1 at $\delta_i = 0$ and equals 0 at $\delta_i = \infty$. Thus:

$$\frac{\partial T_x(T_\phi(\sum \tilde{x}_i))}{\partial(\sum x_i)} \leq \frac{N}{N+1} \qquad \text{(C.1.12)}$$

where equality holds if and only if every $\delta_i = \infty$, or equivalently, if and only if every $U_i = \infty$. If we let $\underline{x}' = \hat{\underline{x}}_{k-1}$ and $\underline{x}'' = \hat{\underline{x}}_k$, then combining (C.1.10) and (C.1.12) gives:

$$\left|\sum \hat{x}_{i,k+1} - \sum \hat{x}_{i,k}\right| \leq \frac{N}{N+1}\left|\sum \hat{x}_{i,k} - \sum \hat{x}_{i,k-1}\right| \qquad \text{(C.1.13)}$$

which is equivalent to:

$$\left|\frac{1}{\hat{\phi}_{k+1}} - \frac{1}{\hat{\phi}}\right| \leq \frac{N}{N+1}\left|\frac{1}{\hat{\phi}_k} - \frac{1}{\hat{\phi}_{k-1}}\right| \qquad \text{(C.1.14)}$$

The MCEM algorithm thus defines a strict contraction mapping on the values of $\frac{1}{\phi}$. The Contraction Mapping theorem (see, for example, page 120 of Ortega and Rheinboldt [1]) then guarantees that the algorithm must converge to its unique fixed point, the unique solution to MCEM.

PARMAP

To prove that PARMAP converges, assume that at least one $U_i = \infty$. We know that $p(X, \phi) \to 0$ as $\phi \to 0$ or $\phi \to \infty$. Thus, since the PARMAP algorithm increases the likelihood function $p(X, \hat{\phi}_k)$ on each step, $\hat{\phi}_k$ must remain bounded above and bounded away from zero. The discussion in chapter 3, section 9.3 then guarantees that the sequence $\hat{\phi}_k$ must have at least one convergent subsequence with limit point ϕ_*, where ϕ_* is a critical point of $p(X, \phi)$. But as argued in equation (C.1.1), $p(X, \phi)$ is strictly log concave, and thus has a single critical point occurring at the global maximum. If at least one $U_i < \infty$, the PARMAP algorithm is therefore guaranteed to converge to the unique global maximum of $p(X, \phi)$.

The PARMAP convergence rate can be analyzed in exactly the same way as MCEM. The only change is that in the PARMAP algorithm the mapping T_ϕ from the sum of signal values to the parameter space Φ is given by

$$T_\phi(\sum x_i) = \frac{N}{\epsilon_0 + \sum x_i} \tag{C.1.15}$$

Using the same argument as before, with this change, gives:

$$\left| \frac{1}{\hat{\phi}_{k+1}} - \frac{1}{\hat{\phi}_k} \right| \leq \left| \frac{1}{\hat{\phi}_k} - \frac{1}{\hat{\phi}_{k-1}} \right| \tag{C.1.16}$$

with equality if and only if $U_i = \infty$ for all i. If any U_i is finite, PARMAP thus defines a strictly non-expansive mapping on the values of $\frac{1}{\hat{\phi}_k}$.

Note that if all $U_i = \infty$, then the PARMAP solution is $\hat{\phi}_{MAP} = 0$, and the PARMAP algorithm's estimate $\hat{\phi}_k$ converges to zero at the rate:

$$\frac{\hat{\phi}_{k+1}}{\hat{\phi}_k} = \frac{1}{\left[1 + \frac{\hat{\phi}_k}{N} (\epsilon_0 + \sum_{i=1}^{N} L_i) \right]} \tag{C.1.17}$$

Maximum Likelihood Algorithms

The proof of convergence of the ML versions of these algorithms exactly follow the proof for the Bayesian version. The ML version of MCEM converges to the unique global minimum of the cross-entropy because the contraction mapping in (C.1.14) still applies; the only problem is that if all $L_i = 0$ then

$$\frac{1}{\hat{\phi}_{k+1}} \leq \frac{N}{N+1} \frac{1}{\hat{\phi}_k} \tag{C.1.18}$$

and $\lim_{k \to \infty} \hat{\phi}_k = \infty$, the MCEM estimate.

The PARML algorithm converges if at least one $L_i > 0$ and one $U_i < \infty$. The proof relies on the fact that $p(X \mid \phi)$ is log concave, strictly log concave if any $U_i < \infty$, and $p(X \mid \phi) \to 0$ as $\phi \to \infty$ if any $U_i < \infty$. These conditions imply that $\hat{\phi}_k$ must remain bounded if any U_i is finite. The three special cases discussed in section 2.7 follow by simple algebra.

2. Gaussian Density - Unknown Variance

MCEM

In the Gaussian example of section 3, the MCEM algorithm generates density estimates of the form:

$$\hat{q}_{X_i}(x_i) = \begin{cases} c_{x_i} \exp\left[-\frac{\hat{s}}{2} (x_i - \hat{\mu})^2 \right] & \text{for } L_i \leq x_i \leq U_i \\ 0 & \text{else} \end{cases}$$

$$\hat{q}_{\Phi}(\mu \mid s) = \left(\frac{s(N+\epsilon)}{2\pi} \right)^{1/2} \exp\left(-\frac{s}{2} \left(\mu - \frac{1}{N+\epsilon} \sum_{i=1}^{N} \hat{x}_i \right)^2 \right) \tag{C.2.1}$$

$$\hat{q}_{\Phi}(s) = K s^{N/2} \exp(-\hat{V} s) \qquad \text{for } s \geq 0$$

Restricting the MCEM minimization to the space of densities of this form will not change the sequence of estimates generated, nor the final solution. With this restriction, we can view the cross-entropy as a function of the coefficients of these densities:

$$H(\hat{s},\hat{\mu},\hat{V},\hat{x}) \equiv H(\hat{q}_X,\hat{q}_\Phi) \qquad \text{(C.2.2)}$$

and we can view the MCEM algorithm as iteratively minimizing H with respect to $\hat{s},\hat{\mu}$ and then $\hat{V},\hat{x}$.

$$\begin{aligned} \hat{\mu}_{k+1}, \hat{s}_{k+1} &\leftarrow \min_{\mu,s} H(\mu, s, \hat{V}_k, \hat{x}_k) \qquad \text{(C.2.3)} \\ \hat{V}_{k+1}, \hat{x}_{k+1} &\leftarrow \min_{V,x} H(\hat{\mu}_{k+1}, \hat{s}_{k+1}, V, x) \end{aligned}$$

After a considerable amount of algebra, we can show that:

$$F(\hat{V},\hat{x}) \equiv \min_{\mu,s} H(\mu,s,\hat{V},\hat{x}) = \min_{q_X} H(q_X,\hat{q}_\Phi) = K - \log \hat{s}^{\frac{1}{2}} \mathrm{p}(X,\hat{\mu},\hat{s}) \qquad \text{(C.2.4)}$$

where K is a fixed constant and $\hat{\mu}$, $\hat{s}$ is the solution to $\min_{\mu,s} H(\mu,s,\hat{v},\hat{x})$:

$$\begin{aligned} \hat{\mu} &= \frac{1}{N+\epsilon} \sum_{i=1}^{N} \hat{x}_i \qquad \text{(C.2.5)} \\ \hat{s} &= \frac{N+2}{\hat{V}} \end{aligned}$$

From (C.2.3), each iteration of MCEM must decrease the value of $F(\hat{V}_k,\hat{x}_k)$, and thus must also increase the value of $\hat{s}_k^{\frac{1}{2}} \mathrm{p}(X,\hat{\mu}_k,\hat{s}_k)$. Note, however, that $s^{\frac{1}{2}}\mathrm{p}(\mu,s) \to 0$ as $s \to 0$ or $s \to \infty$ or $\mu \to \pm\infty$. But:

$$s^{\frac{1}{2}} \mathrm{p}(X,\mu,s) = \mathrm{p}(X \mid \mu,s)\, s^{\frac{1}{2}} \mathrm{p}(\mu,s) \le s^{\frac{1}{2}} \mathrm{p}(\mu,s) \qquad \text{(C.2.6)}$$

Since MCEM always increases the left hand side, the right hand side must be bounded away from zero, which implies that $\hat{\mu}_k$ must be bounded for all k, and $\hat{s}_k$ must be bounded above and bounded away from zero, $\hat{s}_k \ge \xi > 0$ for some ξ. By the Bolzano-Weierstrass theorem, there must exist a convergent subsequence of $\hat{\mu}_k$, $\hat{s}_k$ with limit point μ_*, s_*. Clearly $s_* \ge \xi > 0$. Let x_*, V_* be the corresponding estimates of x, V.

This limit point is in the interior of the natural parameter space, and thus theorem 3.9.6 guarantees that every limit point of the iteration must be a stationary point of the algorithm, a critical point of the cross-entropy, and a critical point of $F(\underline{x},V)$. If we let $\hat{\mu}(\underline{x},V)$, $\hat{s}(\underline{x},V)$ be the solution to $\min\limits_{\mu,s} H(\mu,s,\underline{x},V)$ with the formulas given by (C.2.5), then:

$$0 = \begin{pmatrix} \dfrac{\partial F(\underline{x}_*,V_*)}{\partial \underline{x}} \\ \dfrac{\partial F(\underline{x}_*,V_*)}{\partial V} \end{pmatrix}$$

$$= \begin{pmatrix} \dfrac{\partial\hat{\mu}(\underline{x}_*,V_*)}{\partial \underline{x}} & \dfrac{\partial\hat{s}(\underline{x}_*,V_*)}{\partial \underline{x}} \\ \dfrac{\partial\hat{\mu}(\underline{x}_*,V_*)}{\partial V} & \dfrac{\partial\hat{s}(\underline{x}_*,V_*)}{\partial V} \end{pmatrix} \begin{pmatrix} \dfrac{\partial}{\partial\hat{\mu}}\left[K - \log \hat{s}^{\frac{1}{2}} \mathrm{p}(X,\hat{\mu},\hat{s})\right] \\ \dfrac{\partial}{\partial\hat{s}}\left[K - \log \hat{s}^{\frac{1}{2}} \mathrm{p}(X,\hat{\mu},\hat{s})\right] \end{pmatrix}_{\mu_*,s_*,\underline{x}_*,V_*}$$

$$= \begin{pmatrix} \dfrac{1}{N+\epsilon}\mathrm{I} & 0 \\ 0 & -\dfrac{N+2}{\hat{V}^2} \end{pmatrix} \begin{pmatrix} \dfrac{\partial}{\partial\hat{\mu}}\left[K - \log \hat{s}^{\frac{1}{2}} \mathrm{p}(X,\hat{\mu},\hat{s})\right] \\ \dfrac{\partial}{\partial\hat{s}}\left[K - \log \hat{s}^{\frac{1}{2}} \mathrm{p}(X,\hat{\mu},\hat{s})\right] \end{pmatrix}_{\mu_*,s_*,\underline{x}_*,V_*} \qquad \text{(C.2.7)}$$

which implies that every limit point μ_*, s_* is also a critical point of $s\, \mathrm{p}(X,\mu,s)$.

PARMAP

To prove that PARMAP converges, note that $\mathrm{p}(\mu,s) \to 0$ as $\mu \to \pm\infty$ or $s \to 0$ or $s \to \infty$. Since:

$$\mathrm{p}(X,\hat{\mu}_k,\hat{s}_k) = \mathrm{p}(X \mid \hat{\mu}_k,\hat{s}_k)\,\mathrm{p}(\hat{\mu}_k,\hat{s}_k) \le \mathrm{p}(\hat{\mu}_k,\hat{s}_k) \qquad \text{(C.2.8)}$$

and since each iteration of PARMAP increases the likelihood function $\mathrm{p}(X,\hat{\mu}_k,\hat{s}_k)$, then $\mathrm{p}(\hat{\mu}_k,\hat{s}_k)$ must be bounded below, and thus $\hat{\mu}_k$ must remain bounded and $\hat{s}_k$ must be bounded above and bounded away from zero. The Bolzano-Weierstrass theorem then guarantees that there must be a convergent subsequence of $\hat{\mu}_k$, $\hat{s}_k$ with limit point

μ_*, s_*. Furthermore, since $\hat{s}_k$ is bounded away from zero, $s_*>0$ also. Since the probability densities are continuously differentiable, the discussion in chapter 3, section 9.3 guarantees that every such limit point will be a stationary point of the algorithm, and a critical point of the likelihood function $p(X, \mu, s)$.

SIGMAP

Because

$$\Sigma^{-1} = I - \frac{1}{N+\epsilon}\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}\begin{pmatrix}1 & \cdots & 1\end{pmatrix} \qquad \text{(C.2.9)}$$

is positive definite, $\Sigma^{-1}>0$, the SIGMAP likelihood function:

$$p(x,\Phi) = \begin{cases} \dfrac{K}{\left(\eta + \frac{1}{2}x^T\Sigma^{-1}x\right)^{N/2+1}} & \text{for } L_i \le x_i \le U_i, \quad i=1,\ldots,N \\ 0 & \text{else} \end{cases} \qquad \text{(C.2.10)}$$

is strictly log concave. It therefore goes to zero as any $x_i \to \infty$, and has only a single critical point, located at the global maximum. Since each iteration of the SIGMAP algorithm increases $p(\hat{x}_k, \Phi)$, all estimates $\hat{x}_k$ must remain bounded. The argument given in chapter 3, section 9.3 then guarantees that the sequence converges to the set of critical points of the density. Since there is only one such critical point, the SIGMAP algorithm converges to the unique global maximum of $p(x, \Phi)$.

To show that the convergence rate of the algorithm is at least linear, define the mappings T_{x_i} from the parameter space to the sample space, and T_μ back again by:

$$T_{x_i}(\mu) = \begin{cases} L_i & \text{for } \mu < L_i \\ \mu & \text{for } L_i \le \mu \le U_i \\ U_i\,. & \text{for } U_i < \mu \end{cases} \qquad \text{(C.2.11)}$$

$$T_\mu\left(\sum_{i=1}^{N} x_i\right) = \frac{1}{N+\epsilon}\sum_{i=1}^{N} x_i$$

Let μ', μ'' be any two values of μ. Suppose without loss of generality that $\mu' \geq \mu''$. Then:

$$0 \leq T_{x_i}(\mu') - T_{x_i}(\mu'') \leq \mu' - \mu'' \qquad \text{(C.2.12)}$$

and thus:

$$0 \leq \sum_{i=1}^{N} \left(T_{x_i}(\mu') - T_{x_i}(\mu'') \right) \leq N(\mu' - \mu'') \qquad \text{(C.2.13)}$$

and:

$$0 \leq T_{\mu}\left(\sum_{i=1}^{N} T_{x_i}(\mu') \right) - T_{\mu}\left(\sum_{i=1}^{N} T_{x_i}(\mu'') \right) \leq \frac{N}{N+\epsilon}\,(\mu' - \mu'') \qquad \text{(C.2.14)}$$

or:

$$\left| T_{\mu}\left(\sum_{i=1}^{N} T_{x_i}(\mu') \right) - T_{\mu}\left(\sum_{i=1}^{N} T_{x_i}(\mu'') \right) \right| \leq \frac{N}{N+\epsilon} \left| \mu' - \mu'' \right| \qquad \text{(C.2.15)}$$

Thus, regardless of whether μ' or μ'' is larger, SIGMAP defines a contraction mapping on the parameter space; if we let $\mu' = \hat{\mu}_k$ and $\mu'' = \hat{\mu}_{k-1}$ then:

$$\left| \hat{\mu}_{k+1} - \hat{\mu}_k \right| \leq \frac{N}{N+\epsilon} \left| \hat{\mu}_k - \hat{\mu}_{k-1} \right| \qquad \text{(C.2.16)}$$

PSMAP

Since the PSMAP algorithm is identical to SIGMAP except for the estimate of $\hat{\sigma}^2$ at the end, the proof of convergence is also identical.

Maximum Likelihood Algorithms

To prove convergence of the PARML algorithm, we need only show that the estimates $\hat{\mu}_k$ and $\hat{\sigma}_k^2$ remain bounded. Now:

$$\begin{aligned} p(X \mid \hat{\mu},\hat{\sigma}^2) &= \prod_{i=1}^{N} p(X_i \mid \hat{\mu},\hat{\sigma}^2) \\ &= \prod_{i=1}^{N} \left[\operatorname{erf}\left(\frac{U_i - \hat{\mu}}{\hat{\sigma}} \right) - \operatorname{erf}\left(\frac{L_i - \hat{\mu}}{\hat{\sigma}} \right) \right] \end{aligned} \qquad \text{(C.2.17)}$$

If any interval $[L_i, U_i]$ is finite, $-\infty < L_i < U_i < \infty$, then $p(X_i \mid \hat{\mu}, \hat{\sigma}^2) \to 0$ as $\hat{\mu} \to \pm\infty$ or $\hat{\sigma}^2 \to \infty$. Since $p(X_j \mid \hat{\mu}, \hat{\sigma}^2) \leq 1$ for all j,

$$p(X \mid \hat{\mu}, \hat{\sigma}^2) \leq p(X_i \mid \hat{\mu}, \hat{\sigma}^2) \qquad \text{(C.2.18)}$$

and thus $p(X \mid \hat{\mu}, \hat{\sigma}^2) \to 0$ as $\hat{\mu} \to \pm\infty$ or $\hat{\sigma}^2 \to \infty$. Since each iteration of PARML increases $p(X \mid \hat{\mu}_k, \hat{\sigma}_k^2)$, if any interval $[L_i, U_i]$ is finite then both $\hat{\mu}_k$ and $\hat{\sigma}_k^2$ remain bounded, and the convergence theorems can be applied to show that the iteration must converge to the set of stationary points of the algorithm and critical points of the likelihood function. Conversely, if all intervals $[L_i, U_i]$ are infinite, with $L_i = -\infty$ or $U_i = \infty$ or both for all i, then $\hat{\sigma}_k^2 \to \infty$.

The proof that the MCEM algorithm converges follows by noting that:

$$F(\hat{x}) \equiv \min_{\mu,s} H(\mu, s, \hat{x}) = K - \log\left[\hat{s}^{*} p(X \mid \hat{\mu}, \hat{s})\right] \qquad \text{(C.2.19)}$$

$$\text{where: } \hat{s}, \hat{\mu} \leftarrow \min_{\mu,s} H(\mu, s, \hat{x})$$

The same reasoning used for PARMAP then shows that if any L_i and any U_j are finite, then $s^{*} p(X \mid \mu, s) \to 0$ as $\mu \to \pm\infty$ or $s \to 0$ (i.e. $\sigma^2 \to \infty$.) Since each iteration of the Maximum Likelihood version of MCEM increases $\hat{s}_k^{*} p(X \mid \hat{\mu}_k, \hat{s}_k)$, the parameter estimates must remain bounded if any L_i and any U_j are finite. MCEM thus converges to the set of stationary points of the algorithm and critical points of the cross-entropy and of $s^{*} p(X \mid \mu, s)$.

To prove that PSML converges, note that:

$$\max_{s} p(x \mid \mu, s) = \frac{K}{\left(\sum_{i=1}^{N} (x_i - \mu)^2\right)^{N/2}} \qquad \text{(C.2.20)}$$

The PSML algorithm maximizes this function with respect to x and μ on each step. Let $L^{+} = \max(L_i)$ and let i_{+} be the interval for which this maximum occurs. Also, let

$U^-=\min(U_i)$ and let i_- be the interval in which this minimum occurs. For $\mu>U^-$,

$$\max_s p(x \mid \mu,s) \le \frac{K}{|U^- - \mu|^N} \tag{C.2.21}$$

and thus the density function drops to zero as $\mu\to\infty$. Similarly, for $\mu<L^+$:

$$\max_s p(x \mid \mu,s) \le \frac{K}{|L^+ - \mu|^N} \tag{C.2.22}$$

and thus the density function drops to zero as $\mu\to-\infty$. Since PSML must always increase the value of the function (C.2.20), $\hat{\mu}_k$ must therefore remain bounded. This in turn implies that $\hat{x}_{i,k}$ and $\hat{\sigma}^2$ will also remain bounded. Our convergence theorems can then be applied to show that the algorithm is guaranteed to converge to the set of stationary points of the algorithm and critical points of the function (C.2.20). This function, however, is log concave and thus the only critical points are global maxima of the density. If $L^+>U^-$, then we can also show that the global maximum of the density is unique and lies between U^- and L^+. If $L^+\le U^-$, then every value $\hat{\mu}$ between U^- and L^+ is a global maximum solution to the problem.

3. Gaussian Density - Known Variance

PARMAP

To prove that the PARMAP iteration converges when σ^2 is known, note that $p(\mu)\to 0$ as $\mu\to\pm\infty$. Thus, since:

$$0 < p(X,\mu) = p(X \mid \mu)p(\mu) \le p(\mu) \tag{C.3.1}$$

then $p(X,\mu)\to 0$ as $\mu\to\pm\infty$ also. Since the PARMAP algorithm strictly increases $p(X,\hat{\mu}_k)$ on each iteration, $\hat{\mu}_k$ must be bounded above and below for all k. By the reasoning in chapter 3, section 9.3, the PARMAP algorithm is therefore guaranteed to converge to the set of stationary points of the algorithm, and critical points of the density.

To prove that PARMAP has a unique critical point, note that $p(x \mid \mu)$ is log concave, and X and Φ are convex. Prékopa's theorem (see Appendix D) then guarantees that $p(X \mid \mu)$ will also be log concave. Since $p(\mu)$ is uniformly log concave for $\epsilon>0$, the PARMAP likelihood function $p(X,\mu) = p(X \mid \mu)p(\mu)$ must also be uniformly log concave. Theorem 2.10.4 then guarantees that $p(X,\mu)$ has exactly one critical point, located at the unique global maximum. The iterative PARMAP algorithm must therefore converge to the unique global maximum solution to $\max_{\mu} p(X,\mu)$.

It is also possible to analyze the convergence rate of the algorithm. Each iteration of the algorithm defines a mapping T_μ from the sum of sample values to the parameter space, and a mapping T_x back again:

$$T_x(\mu) = \sum_{i=1}^{N} \mathrm{E}_{X_i}\left[x_i \mid \mu \right] \tag{C.3.2}$$
$$T_\mu(\sum x_i) = \frac{1}{N+\epsilon} \sum x_i$$

One iteration of the algorithm can be written as:

$$\hat{\mu}_{k+1} = T_\mu\left(\sum_{i=1}^{N} \hat{x}_{i,k} \right) \tag{C.3.3}$$
$$\sum_{i=1}^{N} \hat{x}_{i,k+1} = T_x(\hat{\mu}_{k+1})$$

Let μ', μ'' be any two parameter values. Since the functions T_μ and T_x are continuously differentiable, there must exist an intermediate value $\tilde{\mu} = \lambda\mu' + (1-\lambda)\mu''$ for some $0<\lambda<1$ such that:

$$\left| T_\mu(T_x(\mu')) - T_\mu(T_x(\mu'')) \right| \le \left| \frac{\partial T_\mu(T_x(\tilde{\mu}))}{\partial \mu} \right| \left| \mu' - \mu'' \right| \tag{C.3.4}$$

But:

$$\frac{\partial T_{\mu}(T_x(\hat{\mu}))}{\partial \mu} = \frac{1}{N+\epsilon} \sum_{i=1}^{N} \frac{\partial}{\partial \mu} \mathrm{E}_{X_i}\left[x_i \,\middle|\, \hat{\mu} \right]$$

$$= \frac{1}{N+\epsilon} \sum_{i=1}^{N} \frac{1}{\sigma^2} \mathrm{Var}_{X_i}\left[x_i \,\middle|\, \hat{\mu} \right] \qquad \text{(C.3.5)}$$

Our argument in Appendix D, however, guarantees that:

$$\mathrm{Var}_{X_i}\left[x_i \,\middle|\, \hat{\mu} \right] \leq \sigma^2 \qquad \text{(C.3.6)}$$

with equality if and only if $L_i = -\infty$ and $U_i = +\infty$. Thus

$$\left| T_{\mu}(T_x(\mu')) - T_{\mu}(T_x(\mu'')) \right| \leq \frac{N}{N+\epsilon} \left| \mu' - \mu'' \right| \qquad \text{(C.3.7)}$$

Each iteration of PARMAP therefore defines a contraction mapping. By the Contraction Mapping theorem (see, for example, page 120 of Ortega and Rheinboldt [1]) the algorithm must therefore converge to the unique stationary point ("fixed point") of the algorithm. In particular, choosing $\mu' = \hat{\mu}_k$ and $\mu'' = \hat{\mu}_{k-1}$:

$$\left| \hat{\mu}_{k+1} - \hat{\mu}_k \right| \leq \frac{N}{N+\epsilon} \left| \hat{\mu}_k - \hat{\mu}_{k-1} \right| \qquad \text{(C.3.8)}$$

Note that this convergence rate $\frac{N}{N+\epsilon}$ is extremely conservative; if any L_i is finite and any U_i is finite, then the supremum of $\frac{\partial}{\partial \mu} T_{\mu}(T_x(\hat{\mu}))$ in (C.3.5) will be much smaller than $\frac{N}{N+\epsilon}$.

MCEM

Since the MCEM algorithm is identical in this case to PARMAP, geometric convergence to the unique solution of MCEM is guaranteed by the fact that each iteration is a contraction mapping (C.3.8). Yet another way to see the same result is to note that:

$$F(\hat{x}_k) = \min_{q_X} H(q_X, \hat{q}_{\Phi_k}) = K - \log p(X, \hat{\mu}_k) \qquad \text{(C.3.9)}$$

Thus minimizing cross-entropy is exactly equivalent in this case to maximizing the PARMAP density $p(X,\mu)$.

SIGMAP

The SIGMAP likelihood function is:

$$p(x,\Phi) = \int_{\Phi} p(x,\mu)\, d\mu = K \exp\left[-\frac{1}{2\sigma^2} x^T \Sigma^{-1} x \right] \tag{C.3.10}$$

$$\text{where: } \Sigma^{-1} = I - \frac{1}{N+\epsilon} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} (\, 1 \ \cdots \ 1)$$

Since Σ^{-1} is positive definite, $p(x,\Phi) \to 0$ as any component $x_i \to \pm\infty$. Thus, since the SIGMAP algorithm always increases the likelihood $p(\hat{x}_k,\Phi)$, the estimate $\hat{x}_k$ must be bounded for all k. By the argument in chapter 3, section 9.3, the estimate must converge to the set of stationary points of the algorithm, and critical points of $p(x,\Phi)$. Since $\Sigma^{-1}>0$, however, $p(x,\Phi)$ is uniformly log concave, and thus must have a unique global maximum and critical point. The SIGMAP algorithm therefore converges to the unique global maximum of $p(x,\Phi)$.

PSMAP

Since the PSMAP algorithm is identical to the SIGMAP algorithm in this problem, the above proof is sufficient to show that PSMAP must have only one global maximum to which the algorithm converges. Another way to prove the same result is to note that in this problem:

$$p(x,\Phi) = K \max_{\mu} p(x,\mu) \tag{C.3.11}$$

for some constant K. Thus SIGMAP and PSMAP must give exactly the same estimates.

References

1. J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York (1970).

Appendix D
Projection Operators Onto Convex Subsets

In this Appendix we will prove that the projection operator onto a convex (possibly infinite dimensional) set Y is a non-expansive and continuous mapping. We will also prove that the distance $d(x,Y)$ from a point x to a closed set Y is a continuous function. If the set Y is convex, then we will show that the distance function is also a convex function.

Let H be any (possibly infinite dimensional) inner product vector space (Hilbert space) with inner product $<x,y>$, and let $||x||$ be the corresponding norm:

$$||x|| = \sqrt{<x,x>} \tag{D.1}$$

Let Y be a closed and non-empty subset of H. Then we will define the projection operator $K()$ as the function which maps each element $x \in H$ to the nearest possible element $y = K(x)$ in Y:

$$K(x) = \min_{y \in Y} ||y-x||^2 \tag{D.2}$$

We'll start with the following projection theorem:

<u>Theorem D.1</u> [Luenberger [1] p. 69] Let x be a vector in H, and let Y be a closed, non-empty and convex subset of H. Then there is a unique vector $\hat{y} \in Y$ such that:

$$||x-\hat{y}|| \leq ||x-y|| \qquad \text{for all } y \in Y \tag{D.3}$$

Furthermore, a necessary and sufficient condition that $\hat{y}$ be the unique minimizing vector is that:

$$< x-\hat{y} \, , \, y-\hat{y} > \leq 0 \qquad \text{for all } y \in Y \tag{D.4}$$

With this theorem we can then prove:

Theorem D.2 If $Y \subseteq H$ is a closed, non-empty and convex set, then the projection operator $K():H \rightarrow Y$ is a non-expansive and uniformly continuous mapping:

$$\left\|K(x_1)-K(x_2)\right\| \leq \left\|x_1-x_2\right\| \qquad \text{for all } x_1,x_2 \in H \tag{D.5}$$

Proof: Let $y_1=K(x_1)$ and $y_2=K(x_2)$. Because the norm is defined in terms of an inner product:

$$0 \leq \left\|(x_1-x_2)-(y_1-y_2)\right\|^2 = \left\|x_1-x_2\right\|^2 + \left\|y_1-y_2\right\|^2 - 2<x_1-x_2,y_1-y_2> \tag{D.6}$$

But from theorem D.1:

$$\begin{aligned} &< x_1-y_1 , y_2-y_1 > \leq 0 \\ &< x_2-y_2 , y_1-y_2 > \leq 0 \end{aligned} \tag{D.7}$$

Subtracting these yields:

$$< (x_1-x_2)-(y_1-y_2) , y_1-y_2 > \geq 0 \tag{D.8}$$

or:

$$< x_1-x_2 , y_1-y_2 > \geq ||y_1-y_2||^2 \tag{D.9}$$

Thus, by combining (D.9) and (D.6):

$$\begin{aligned} ||x_1-x_2||^2 &\geq 2< x_1-x_2 , y_1-y_2 > - ||y_1-y_2||^2 \\ &\geq ||y_1-y_2||^2 \\ &= \left\|K(x_1)-K(x_2)\right\|^2 \end{aligned} \tag{D.10}$$

Uniform continuity of $K()$ follows immediately, since as $x_2 \rightarrow x_1$, equation (D.10) implies that $K(x_2) \rightarrow K(x_1)$.

Theorem D.3 Let Y be a closed and non-empty set (not necessarily convex.) Then the distance function:

$$d(x,Y) = \min_{y \in Y} ||y - x||^2 \tag{D.11}$$

is uniformly continuous.

Proof: Let $x_1, x_2 \in H$, and let $y_1, y_2 \in Y$ be their corresponding projections onto Y. By the triangle rule:

$$\begin{aligned} ||x_2-x_1||^2 + ||x_1-y_1||^2 &\geq ||(x_2-x_1)+(x_1-y_1)||^2 \\ &= ||x_2-y_1||^2 \end{aligned} \tag{D.12}$$

But since y_2 is the projection of x_2 onto Y:

$$||x_2-y_1||^2 \geq ||x_2-y_2||^2 \tag{D.13}$$

Thus combining these:

$$\begin{aligned} ||x_2-x_1||^2 &\geq ||x_2-y_2||^2 - ||x_1-y_1||^2 \\ &= d^2(x_2,Y) - d^2(x_1,Y) \end{aligned} \tag{D.14}$$

Similarly:

$$||x_1-x_2||^2 + ||x_2-y_2||^2 \geq ||x_1-y_2||^2 \geq ||x_1-y_1||^2 \tag{D.15}$$

and thus:

$$||x_1-x_2||^2 \geq d^2(x_1,Y) - d^2(x_2,Y) \tag{D.16}$$

Combining (D.16) and (D.14):

$$\left| d^2(x_1,Y) - d^2(x_2,Y) \right| \leq ||x_1-x_2||^2 \tag{D.17}$$

Thus as $x_1 \rightarrow x_2$, $d(x_1,Y) \rightarrow d(x_2,Y)$ and the distance function is uniformly continuous.

Theorem D.4 Let $Y \subseteq H$ be closed, non-empty and also convex. Then the distance function:

$$d(x,Y) = \min_{y \in Y} ||y - x||^2 \tag{D.18}$$

is a convex and uniformly continuous function of x.

Proof: Choose any $x_1, x_2 \in H$, and define y_1, y_2 as their projection onto Y:

$$y_1 = K(x_1) \quad \text{and} \quad y_2 = K(x_2) \tag{D.19}$$

Because Y is convex, the point $\lambda y_1 + (1-\lambda) y_2$ is an element of Y for any $0 \leq \lambda \leq 1$. Then because the norm is a convex function:

$$\begin{aligned} d(\lambda x_1 + (1-\lambda) x_2, Y) &= \min_{y \in Y} \left\| y - (\lambda x_1 + (1-\lambda) x_2) \right\| \\ &\leq \left\| (\lambda y_1 + (1-\lambda) y_2) - (\lambda x_1 + (1-\lambda) x_2) \right\| \\ &\leq \lambda \|y_1 - x_1\| + (1-\lambda) \|y_2 - x_2\| \\ &= \lambda d(x_1, Y) + (1-\lambda) d(x_2, Y) \end{aligned} \tag{D.20}$$

and thus $d(x, Y)$ is convex. Uniform continuity follows from Theorem D.3.

References

1. David G. Luenberger, *Optimization By Vector Space Methods*, John Wiley & Sons Inc., New York (1969).

Appendix E

Log Concave Functions

Conditional Expectations as Non-expansive Mappings

In this Appendix we will summarize some theorems on log concave functions which were developed by Davidovic, *et al*[1] Prékopa, [2,3] and others, [4,5,6] and show how these theorems can be used to analyze the convergence of MCEM and PARMAP-like algorithms such as XMAP and YMAP in chapter 5. We also present some conjectures about upper bounds on the variance of log concave densities and convexity of the cross-entropy; we used to call these "theorems" until we discovered some minor technical flaws in the proofs. Lack of time prevented us from fixing these, though we believe the results are correct.

A function $f()$ defined on R^N is said to be logarithmic concave if it is non-negative and if for every pair of vectors $x_1, x_2 \in \mathrm{R}^N$:

$$f(\lambda x_1 + (1-\lambda)x_2) \geq f(x_1)^\lambda f(x_2)^{1-\lambda} \qquad \text{for all } 0<\lambda<1 \tag{E.1}$$

Strict and uniform log concavity can be defined in much the same way as for concave functions. If the function $f()$ is logarithmic concave in R^N, then it can be written as $f(x)=e^{Q(x)}$ where $Q(x)$ is concave in the entire space and the value $-\infty$ is also allowed for the function $Q(x)$. The set $\{x \,|f(x)>0\}$ is convex, and $f()$ is continuous in the interior of the set. Many probability densities are log concave. For example, the Gaussian density

$$N(m,\mathrm{V}) = \frac{1}{|2\pi \mathrm{V}|^{½}} \exp\left\{ -½(x-m)^{\mathrm{T}}\mathrm{V}^{-1}(x-m) \right\} \tag{E.2}$$

with $\mathrm{V}>0$ is log concave, as is the Wishart density, the multivariate beta density, the

Dirichlet distribution (see Prékopa), the gamma distribution (see Davidovic) and many others.

Log concave functions have a variety of interesting properties. The following have been proved in the literature:

1) Let $f(x)$, $g(x)$ be log concave in $\mathbf{R}^N$. Then:

   a) The product $f(x)g(x)$ is log concave in $x \in \mathbf{R}^N$.

   b) The convolution $h(y) = \int_{\mathbf{R}^N} f(x)g(y-x)\,dx$ is log concave in $y \in \mathbf{R}^N$ (see Davidovic, Prékopa)

2) [Prékopa's theorem] Let $f(x,y)$ be a function of $N+M$ variables when $x \in \mathbf{R}^N$, $y \in \mathbf{R}^M$. Suppose $f()$ is log concave in $\mathbf{R}^{N+M}$. Let $\mathbf{K}$ be a convex subset of $\mathbf{R}^N$. Then the function of y defined by:

$$h(y) = \int_{\mathbf{K}} f(x,y)\,dx$$

is log concave for all $y \in \mathbf{R}^M$.

The following is not as well known, though it is easy to prove:

3) Let $f(x,y)$ be a log concave function of $N+M$ variables as in (2) above, and let $\mathbf{K}$ be a convex subset of $\mathbf{R}^N$. Then the function of x defined by:

$$h(x) = \sup_{y \in \mathbf{K}} f(x,y)$$

is log concave for all $x \in \mathbf{R}^N$.

<u>Proof of Property 3:</u>

Let x_1, $x_2 \in \mathbf{R}^N$, and let $y_i - \sup_{y \in \mathbf{K}} f(x_i,y)$ for $i=1,2$. Define $(\hat{x},\hat{y})$ to be any point on the line connecting (x_1,y_1) and (x_2,y_2):

$$(\tilde{x},\tilde{y}) = \lambda(x_1,y_2) + (1-\lambda)(x_2,y_2) \tag{E.3}$$

Then:

$$\begin{aligned} h(\tilde{x}) &= \sup_{y \in K} f(\tilde{x},y) \\ &\geq f(\tilde{x},\tilde{y}) \\ &\geq f(x_1,y_1)^{\lambda} f(x_2,y_2)^{1-\lambda} \\ &= h(x_1)^{\lambda} h(x_2)^{1-\lambda} \end{aligned} \tag{E.4}$$

and so $h(x)$ is log concave also. In fact, if $f()$ is strictly or uniformly log concave, then $h(x)$ will also be strictly or uniformly concave, respectively. □

Property (2) immediately implies, for example, that if our system model probability density $p(x|\phi)$ is log concave in x,ϕ and the constraint sets X,Φ are convex, then the probability density

$$p(X|\phi) = \int_{X} p(x|\phi)\, dx \tag{E.5}$$

will also be log concave in $\phi \in \Phi$.

An interesting theorem closely related to these properties of log concave functions concerns the behavior of the mean of a truncated Gaussian as we move the center of the density. (See Kanter and Proppe [7] for a more complete discussion.) We first show that:

Theorem E.1: Let x be a Gaussian variable with density $N(m,V)$. Then if x is restricted to lie within a convex set $X \subseteq R^N$, its constrained variance must be smaller than the unconstrained variance V:

$$\mathrm{Var}_X\left[x \,\middle|\, m\right] \leq V \tag{E.6}$$

where:

$$\mathrm{Var}_X\left[x \,\middle|\, m\right] = \frac{\int_X (x-\hat{x})(x-\hat{x})^T\, p(x \mid m)\, dx}{\int_X p(x \mid m)\, dx} \tag{E.7}$$

$$\hat{x} = E_X\left[x \,\middle|\, m\right] = \frac{\int_X x\, p(x \mid m)\, dx}{\int_X p(x \mid m)\, dx} \tag{E.8}$$

Proof: The Gaussian density $p(x \mid m)$ is log concave. Therefore, by property 2 above, if X is convex, the the marginal density:

$$p(X \mid m) = \int_X p(x \mid m)\, dx \tag{E.9}$$

is also log concave in m. Because this density is twice differentiable in m, the second derivative of $\log p(X \mid m)$ must therefore be negative semidefinite:

$$0 \geq \frac{\partial^2}{\partial m^2} \log p(X \mid m) \tag{E.10}$$

Calculating this derivative gives:

$$0 \geq V^{-1}\mathrm{Var}_X[x \mid m]V^{-1} - V^{-1} \tag{E.11}$$

Multiplying both sides of (E.11) on the left and the right by V gives the desired result.

□

Next we show that:

Theorem E.2: Let x be a Gaussian variable with density $N(m, V)$, and let X be a convex subset of $\mathbf{R}^N$. Then the conditional expectation mapping $K_x : \mathbf{R}^N \rightarrow \mathbf{R}^N$ defined by:

$$K_x(\boldsymbol{m}) = \mathrm{E}_X[x\,|\boldsymbol{m}] = \frac{\displaystyle\int_X x\ \mathrm{p}(x\,|\boldsymbol{m})\ dx}{\displaystyle\int_X \mathrm{p}(x\,|\boldsymbol{m})\ dx} \tag{E.12}$$

is a non-expansive and uniformly continuous mapping under the norm $||\cdot||_\mathrm{V}$:

$$||\ K_x(\boldsymbol{m}_1)-K_x(\boldsymbol{m}_2)\ ||_\mathrm{V} \leq ||\ \boldsymbol{m}_1-\boldsymbol{m}_2\ ||_\mathrm{V} \tag{E.13}$$

Proof: Note that $K_x(\boldsymbol{m})$ is continuously differentiable in $\boldsymbol{m}$. Therefore, by the mean value theorem (see, for example, [8] chapters 1,3), there exists an $\bar{\boldsymbol{m}} = \lambda\boldsymbol{m}_1+(1-\lambda)\boldsymbol{m}_2$ on the line connecting $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ such that:

$$\left|\left| K_x(\boldsymbol{m}_1)-K_x(\boldsymbol{m}_2) \right|\right|_\mathrm{V}^2 = \left|\left| \frac{d}{d\boldsymbol{m}}K_x(\bar{\boldsymbol{m}}) \right|\right|_\mathrm{V}^2 \left|\left| \boldsymbol{m}_1-\boldsymbol{m}_2 \right|\right|_\mathrm{V}^2 \tag{E.14}$$

We now prove that $\left|\left| \frac{d}{d\boldsymbol{m}}K_x(\bar{\boldsymbol{m}}) \right|\right|_\mathrm{V} \leq 1$ for all $\bar{\boldsymbol{m}} \in \mathbf{R}^N$, which will be sufficient to prove (E.13). Substituting the Gaussian probability density into the formula for $K_x()$ in (E.12) and differentiating with respect to $\boldsymbol{m}$ gives:

$$\frac{d}{d\boldsymbol{m}}K_x(\boldsymbol{m}) = \mathrm{Var}_X[x\,|\boldsymbol{m}]\ \mathrm{V}^{-1} \tag{E.15}$$

Let $\mathrm{V}^{1/2}$ be any "square root" of V and let $\mathrm{V}^{T/2}$ be its transpose, $\mathrm{V}^{1/2}\mathrm{V}^{T/2}=\mathrm{V}$. Then:

$$\begin{aligned}\left|\left| \frac{d}{d\boldsymbol{m}}\ K_x(\boldsymbol{m}) \right|\right|_\mathrm{V}^2 &= \max_{\underline{y}\neq\underline{0}} \frac{\underline{y}^\mathrm{T}\mathrm{V}^{-1}\mathrm{Var}_X[x\,|\boldsymbol{m}]\mathrm{V}^{-1}\mathrm{Var}_X[x\,|\boldsymbol{m}]\mathrm{V}^{-1}\underline{y}}{\underline{y}^\mathrm{T}\mathrm{V}^{-1}\underline{y}} \\ &= \max_{\boldsymbol{z}\neq 0} \frac{\boldsymbol{z}^\mathrm{T}\left(\mathrm{V}^{-1/2}\mathrm{Var}_X[x\,|\boldsymbol{m}]\mathrm{V}^{-T/2}\right)^2\boldsymbol{z}}{\boldsymbol{z}^\mathrm{T}\boldsymbol{z}}\end{aligned} \tag{E.16}$$

$$\text{where: } \boldsymbol{z} = \mathrm{V}^{-1/2}\underline{y}$$

But multiplying (E.6) on the left and right by $\mathrm{V}^{-1/2}$ and $\mathrm{V}^{-T/2}$ respectively gives:

$$\mathrm{V}^{-T/2}\mathrm{Var}_X[x\,|\boldsymbol{m}]\mathrm{V}^{-1/2} \leq \mathrm{I} \tag{E.17}$$

so that all eigenvalues of this positive semi-definite matrix on the left must be less than 1. The same must hold for the matrix squared, which implies in (E.16) that:

$$\left\| \frac{d}{d\underline{m}} K_x(\underline{\tilde{m}}) \right\|_V^2 \leq 1 \qquad \text{for all } \underline{\tilde{m}} \in \mathbf{R}^N \tag{E.18}$$

Combining this with (E.14) proves that the conditional expectation operator is non-expansive. It is also easy to show from (E.14) that it is uniformly continuous. □

This last theorem implies that if X and Y are convex, then our MCEM, XMAP, and YMAP algorithms in chapter 5 satisfy:

$$\left\| \mathrm{E}_Y[\underline{y} \mid \mathrm{B}\underline{x}_1] - \mathrm{E}_Y[\underline{y} \mid \mathrm{B}\underline{x}_2] \right\|_R \leq \left\| \mathrm{B}\underline{x}_1 - \mathrm{B}\underline{x}_2 \right\|_R \tag{E.19}$$

$$\left\| \mathrm{E}_X[\underline{x} \mid \mathrm{H}\underline{y}_1] - \mathrm{E}_X[\underline{x} \mid \mathrm{H}\underline{y}_2] \right\|_V \leq \left\| \mathrm{H}\underline{y}_1 - \mathrm{H}\underline{y}_2 \right\|_V$$

Thus the conditional expectation operators on convex sets resemble projection operators on convex sets in that they are both non-expansive mappings. We therefore should expect that since our four iterative signal reconstruction algorithms differ only in the substitution of conditional expectations for projection operators, that all four iterative algorithms will have similar convergence characteristics.

****

The following results are stated as conjectures because at the last minute we found some technical problems with their proofs. Nevertheless, we believe these to be true, though some additional technical conditions on the existence of certain integrals may have to be added.

Conjecture E.1 Suppose $p(\underline{x})$ has a continuous second derivative and that it is log concave on a closed, convex set X. Then for all vectors $\underline{v}$:

$$\underline{v}^{T}\mathrm{Cov}_{X}[\underline{x}]\,\underline{v} \le \inf_{M\in\Psi} \underline{v}^{T}M\underline{v} \tag{E.20}$$

where Ψ is the set of symmetric positive definite matrices M defined by:

$$\Psi = \left\{ M \,\middle|\, M = M^{T} \text{ and } M \ge \left[\frac{\partial^2}{\partial \underline{x}^2} \log p(\underline{x}) \right]^{-1} \text{ for all } \underline{x} \in X \right\} \tag{E.21}$$

Conjecture E.2 Suppose that $p(\underline{x},\underline{\phi})$ is a "natural" exponential family of densities:

$$p(\underline{x},\underline{\phi}) = g(\underline{x})h(\underline{\phi})\exp\left[\underline{\phi}^{T}\underline{x}\right] \tag{E.22}$$

which is also strictly log concave on a convex and closed domain $X \times \Phi$. Then the cross-entropy expression $H(\underline{\tau},\underline{\rho})$ given in chapter 3 is positive definite everywhere:

$$\frac{\partial^2 H(\underline{\tau},\underline{\rho})}{\partial(\underline{\tau},\underline{\rho})^2} = \begin{pmatrix} R_{\tau}^{-1} & -I \\ -I & R_{\pi}^{-1} \end{pmatrix} > 0 \tag{E.23}$$

and thus $H(\underline{\tau},\underline{\rho})$ is strictly convex. □

References

1. Ju.S. Davidovic, B.I. Korenbljum, and B.I. Hacet, *A Property of Logarithmically Concave Functions*, 1969.
2. András Prékopa, "Logarithmic Concave Measures With Application to Stochastic Programming," *(Szeged) Acta Sci. Math.* **32**, pp.301-316 (1971).
3. András Prékopa, "On Logarithmic Concave Measures and Functions," *(Szeged) Acta Sci. Math.* **34**, pp.335-343 (1973).
4. V.A. Tomilenko, "Integrals of Logarithmically Concave Functions," *Math. Notes of the Academy of Sciences of the USSR* (Russian Original) **20**(6), pp.1030-1031 (Dec 1976 (Jan 1977)).
5. S. Dancs and B. Uhrin, "On a Class of Integral Inequalities and Their Measure-Theoretic Consequences," *J. Math. Anal. Appl.* **74**, pp.388-400 (1980).
6. Somesh Das Gupta, "Brunn-Minkowski Inequality and Its Aftermath," *J. Multivariate Anal.* **10**, pp.296-318 (1980).
7. Marek Kanter and Harold Proppe, *Reduction of Variance for Gaussian Densities via Restriction to Convex Sets*, 1977.
8. A.A. Goldstein, *Constructive Real Analysis*, Harper and Row, New York (1967).

Appendix F

Convergence Rate of Signal Reconstruction Algorithms in Chapter 5

In this Appendix we will prove that when the constraint sets X, Y are convex, then the iterative algorithms of chapter 5 are guaranteed to converge to the unique global maximum solution at a geometric rate. We will also derive an upper bound for this convergence rate. Let us define the operators $K_x()$ and $K_y()$ which map the signal and output space onto the constraint sets X and Y by either a projection operation or a conditional expectation operation:

$$K_x(\mathrm{H}\underline{y}) = \begin{cases} \min\limits_{x \in X} \|\underline{x} - \mathrm{H}\underline{y}\|_{\mathrm{V}}^2 & \text{for XMAP, XYMAP} \qquad \text{(F.1a)} \\ \mathrm{E}_X[\underline{x}\,|\mathrm{H}\underline{y}] & \text{for MCEM, YMAP} \qquad \text{(F.1b)} \end{cases}$$

$$K_y(\mathrm{B}\underline{x}) = \begin{cases} \min\limits_{y \in Y} \|\underline{y} - \mathrm{B}\underline{x}\|_{\mathrm{R}}^2 & \text{for YMAP, XYMAP} \qquad \text{(F.1c)} \\ \mathrm{E}_Y[\underline{y}\,|\mathrm{B}\underline{x}] & \text{for MCEM, XMAP} \qquad \text{(F.1d)} \end{cases}$$

where the conditional expectation operator $\mathrm{E}_X[\underline{x}\,|\mathrm{H}\underline{y}]$ finds the mean of a Gaussian density $N(\mathrm{H}\underline{y},\mathrm{V})$ truncated to the set X, and the conditional expectation operator $\mathrm{E}_Y[\underline{y}\,|\mathrm{B}\underline{x}]$ finds the mean of a Gaussian density $N(\mathrm{B}\underline{x},\mathrm{R})$ truncated to the set Y. With this notation, all four of our algorithms can be written in the form:

$$\begin{aligned} \hat{\underline{x}}_{k+1} &= K_x(\mathrm{H}\hat{\underline{y}}_k) \\ \hat{\underline{y}}_{k+1} &= K_y(\mathrm{B}\hat{\underline{x}}_{k+1}) \end{aligned} \qquad \text{(F.2)}$$

Because the constraint sets are convex, theorem D.2 in Appendix D guarantees that the projection operators (F.1a) and (F.1c) are non-expansive mappings. Theorem E.3 in Appendix E similarly guarantees that conditional expectation operators (F.1b) and (F.1d) are non-expansive mappings. Thus for any of our four iterative algorithms, if

$\underline{x}_1,\underline{x}_2$ are any elements of X and $\underline{y}_1,\underline{y}_2$ are any elements of Y, then:

$$\left\| K_x(\underline{x}_1)-K_x(\underline{x}_2) \right\|_V \leq \left\| \underline{x}_1-\underline{x}_2 \right\|_V \tag{F.3}$$

$$\left\| K_y(\underline{y}_1)-K_y(\underline{y}_2) \right\|_R \leq \left\| \underline{y}_1-\underline{y}_2 \right\|_R \tag{F.4}$$

We will now use these expressions to show that a single pass of any of our four iterative algorithms defines a contraction mapping on the constraint sets. This result will immediately lead to an upper bound on the convergence rate of the algorithms.

First some preliminary mathematics. Define the spectral radius matrix norm $||\cdot||_V$ by:

$$||T||_V = \max_{\underline{x}\neq\underline{0}} \frac{||T\underline{x}||_V}{||\underline{x}||_V} = \max_{\underline{x}\neq\underline{0}} \frac{\underline{x}^T T^T V^{-1} T\underline{x}}{\underline{x}^T V^{-1}\underline{x}} \tag{F.5}$$

Clearly:

$$||T\underline{x}||_V \leq ||T||_V \; ||\underline{x}||_V \qquad \text{for all } \underline{x} \tag{F.6}$$

The spectral radius norm $||T||_R$ can be defined similarly. We will also need to factor V and R; because V and R are both positive definite and symmetric matrices, it is possible to find matrices $V^{1/2}$ and $R^{1/2}$ such that:

$$V = V^{1/2}V^{T/2} \qquad \text{and} \qquad R = R^{1/2}R^{T/2} \tag{F.7}$$

where the notation $V^{T/2}$ represents the transpose of $V^{1/2}$. We will also use $V^{-1/2}$ and $V^{-T/2}$ to represent the inverses of $V^{1/2}$ and $V^{T/2}$.

On to the derivation. Let $\underline{y}_1$, $\underline{y}_2$ be any two elements in Y. Starting at $\underline{y}_i$, our iterative signal reconstruction algorithms will calculate a new signal estimate of $\underline{x}_i = K_x(H\underline{y}_i)$ for $i=1,2$. Using the non-expansive mapping property of $K_x()$ in (F.3), using the definition of the norms $||\cdot||_V$ and $||\cdot||_R$, and using the factorizations (F.7) gives:

$$
\begin{aligned}
\| \underline{x}_1 - \underline{x}_2 \|_V^2 &= \left\| K_x(H\underline{y}_1) - K_x(H\underline{y}_2) \right\|_V^2 \\
&\leq \left\| H\underline{y}_1 - H\underline{y}_2 \right\|_V^2 \\
&= (\underline{y}_1 - \underline{y}_2)^T H^T V^{-1} H (\underline{y}_1 - \underline{y}_2) \\
&\leq \left\{ \max_{\underline{y} \neq \underline{0}} \frac{\underline{y}^T H^T V^{-1} H \underline{y}}{\underline{y}^T R^{-1} \underline{y}} \right\} (\underline{y}_1 - \underline{y}_2)^T R^{-1} (\underline{y}_1 - \underline{y}_2) \\
&= \left\{ \max_{\underline{y} \neq \underline{0}} \frac{\underline{y}^T R^{-1} B V B^T R^{-1} \underline{y}}{\underline{y}^T R^{-1} \underline{y}} \right\} \| \underline{y}_1 - \underline{y}_2 \|_R^2 \\
&= \left\{ \max_{\underline{y} \neq \underline{0}} \frac{\underline{y}^T \left[R^{-1} - [BA^{-1}QA^{-T}B^T + R]^{-1} \right] \underline{y}}{\underline{y}^T R^{-1} \underline{y}} \right\} \| \underline{y}_1 - \underline{y}_2 \|_R^2 \\
&= \max_{\underline{u} \neq \underline{0}} \left\{ 1 - \frac{\underline{u}^T \left[R^{-1/2} B A^{-1} Q A^{-T} B^T R^{-T/2} + I \right]^{-1} \underline{u}}{\underline{u}^T \underline{u}} \right\} \| \underline{y}_1 - \underline{y}_2 \|_R^2
\end{aligned}
$$

where: $\underline{u} = R^{-1/2} \underline{y}$

$$
= \max_{\underline{u} \neq \underline{0}} \left\{ 1 - \frac{\underline{u}^T \underline{u}}{\underline{u}^T \left[R^{-1/2} B A^{-1} Q A^{-T} B^T R^{-T/2} + I \right] \underline{u}} \right\} \| \underline{y}_1 - \underline{y}_2 \|_R^2
$$

$$
= \max_{\underline{y} \neq \underline{0}} \left\{ 1 - \frac{\underline{y}^T R \underline{y}}{\underline{y}^T \left[B A^{-1} Q A^{-T} B^T + R \right] \underline{y}} \right\} \| \underline{y}_1 - \underline{y}_2 \|_R^2
$$

where: $\underline{y} = R^{1/2} \underline{u}$

$$
= \frac{\mu_{max}}{\mu_{max} + 1} \| \underline{y}_1 - \underline{y}_2 \|_R^2 \qquad \text{(F.8)}
$$

where $\mu_{max} = \max_{\underline{u} \neq \underline{0}} \dfrac{\underline{u}^T B A^{-1} Q A^{-T} B^T \underline{u}}{\underline{u}^T R \underline{u}}$

Similarly, let $\underline{x}_1$, $\underline{x}_2$ be any two elements in X, and let $\underline{y}_1$, $\underline{y}_2$ be the corresponding output estimates generated by our iterative algorithm, $\underline{y}_i = K_y(\mathrm{B}\underline{x}_i)$. Using the non-expansive mapping property of $K_y()$ in (F.8):

$$\begin{aligned}
\|\underline{y}_1-\underline{y}_2\|_{\mathrm{R}}^2 &= \left\| K_y(\mathrm{B}\underline{x}_1) - K_y(\mathrm{B}\underline{x}_2) \right\|_{\mathrm{R}}^2 \\
&\leq \left\| \mathrm{B}\underline{x}_1 - \mathrm{B}\underline{x}_2 \right\|_{\mathrm{R}}^2 \\
&= (\underline{x}_1-\underline{x}_2)^{\mathrm{T}}\mathrm{B}^{\mathrm{T}}\mathrm{R}^{-1}\mathrm{B}(\underline{x}_1-\underline{x}_2) \\
&\leq \left\{ \max_{\underline{x}\neq\underline{0}} \frac{\underline{v}^{\mathrm{T}}\mathrm{B}^{\mathrm{T}}\mathrm{R}^{-1}\mathrm{B}\underline{v}}{\underline{v}^{\mathrm{T}}\mathrm{V}^{-1}\underline{v}} \right\} (\underline{x}_1-\underline{x}_2)^{\mathrm{T}}\mathrm{V}^{-1}(\underline{x}_1-\underline{x}_2) \\
&= \left\{ \max_{\underline{x}\neq\underline{0}} \frac{\underline{v}^{\mathrm{T}}\mathrm{B}^{\mathrm{T}}\mathrm{R}^{-1}\mathrm{B}\underline{v}}{\underline{v}^{\mathrm{T}}\mathrm{B}^{\mathrm{T}}\mathrm{R}^{-1}\mathrm{B}\underline{v} + \underline{v}^{\mathrm{T}}\mathrm{A}^{\mathrm{T}}\mathrm{Q}^{-1}\mathrm{A}\underline{v}} \right\} \|\underline{x}_1-\underline{x}_2\|_{\mathrm{V}}^2 \\
&= \frac{\lambda_{max}}{\lambda_{max}+1} \|\underline{x}_1-\underline{x}_2\|_{\mathrm{V}}^2 \qquad \text{(F.9)}
\end{aligned}$$

$$\text{where } \lambda_{max} = \max_{\underline{x}\neq\underline{0}} \frac{\underline{v}^{\mathrm{T}}\mathrm{B}^{\mathrm{T}}\mathrm{R}^{-1}\mathrm{B}\underline{v}}{\underline{v}^{\mathrm{T}}\mathrm{A}^{\mathrm{T}}\mathrm{Q}^{-1}\mathrm{A}\underline{v}}$$

To summarize:

$$\left\| K_x(\mathrm{H}\underline{y}_1) - K_x(\mathrm{H}\underline{y}_2) \right\|_{\mathrm{V}} \leq \nu_x \; \|\underline{y}_1-\underline{y}_2\|_{\mathrm{R}} \qquad \text{(F.10)}$$

$$\left\| K_y(\mathrm{B}\underline{x}_1) - K_y(\mathrm{B}\underline{x}_2) \right\|_{\mathrm{R}} \leq \nu_y \; \|\underline{x}_1-\underline{x}_2\|_{\mathrm{V}} \qquad \text{(F.11)}$$

$$\begin{aligned}
\text{where: } \nu_x &= \left(\frac{\mu_{max}}{\mu_{max}+1} \right)^{1/2} < 1 \\
\nu_y &= \left(\frac{\lambda_{max}}{\lambda_{max}+1} \right)^{1/2} < 1
\end{aligned}$$

Let $\underline{x}_1$, $\underline{y}_1$ be the estimates $\hat{\underline{x}}_k$ and $\hat{\underline{y}}_{k-1}$, and let $\underline{x}_2$, $\underline{y}_2$ be the estimates $\hat{\underline{x}}_{k+1}$, $\hat{\underline{y}}_k$. Substituting these values into (F.10) and (F.11) and using (F.2), these equations take the form:

$$\begin{aligned}
\|\hat{\underline{x}}_{k+1}-\hat{\underline{x}}_k\|_{\mathrm{V}} &\leq \nu_x \; \|\hat{\underline{y}}_k-\hat{\underline{y}}_{k-1}\|_{\mathrm{R}} \qquad \text{(F.12)} \\
\|\hat{\underline{y}}_{k+1}-\hat{\underline{y}}_k\|_{\mathrm{R}} &\leq \nu_y \; \|\hat{\underline{x}}_{k+1}-\hat{\underline{x}}_k\|_{\mathrm{V}}
\end{aligned}$$

Since ν_x, $\nu_y<1$, the distance between successive estimates drops at least at a geometric rate. To show that these relationships guarantee convergence, let L be a (large) number and let n, m be any integers such that $n \geq m \geq L$. Then:

$$\begin{aligned} \|\hat{\underline{y}}_n - \hat{\underline{y}}_m\|_R &\leq \sum_{k=m}^{n-1} \|\hat{\underline{y}}_{k+1} - \hat{\underline{y}}_k\|_R \\ &\leq \sum_{k=m}^{n-1} (\nu_x \nu_y)^k \|\hat{\underline{y}}_1 - \hat{\underline{y}}_0\|_R \\ &\leq \frac{(\nu_x \nu_y)^L}{1-\nu_x \nu_y} \|\hat{\underline{y}}_1 - \hat{\underline{y}}_0\|_R \end{aligned} \qquad \text{(F.13)}$$

By choosing L sufficiently large, we can therefore make $\|\hat{\underline{y}}_n - \hat{\underline{y}}_m\|_R$ arbitrarily small for all $n, m \geq L$. The sequence $\hat{\underline{y}}_n$ is therefore a Cauchy sequence, and must converge to a unique limit $\hat{\underline{y}}_n \rightarrow \underline{y}_*$ as $n \rightarrow \infty$. Similarly, we can show that $\hat{\underline{x}}_n \rightarrow \underline{x}_*$ as $n \rightarrow \infty$. Let $\underline{x}_1$, $\underline{y}_1$ be the estimates $\hat{\underline{x}}_{k+1}$, $\hat{\underline{y}}_k$, and let $\underline{x}_2$, $\underline{y}_2$ be the global maximum solution $\underline{x}_*$, $\underline{y}_*$. Then since the global maximum must be a stationary point of the algorithm, $\underline{x}_* = K_x(\mathrm{H}\underline{y}_*)$ and $\underline{y}_* = K_y(\mathrm{B}\underline{x}_*)$, and equations (F.10) and (F.11) guarantee that:

$$\left\| \hat{\underline{y}}_{k+1} - \underline{y}_* \right\|_R \leq \nu_y \left\| \hat{\underline{x}}_{k+1} - \underline{x}_* \right\|_V \leq \nu_y \nu_x \left\| \hat{\underline{y}}_k - \underline{y}_* \right\| \qquad \text{(F.14)}$$

Thus the estimates converge to this unique limit point at the geometric rate $\nu_x \nu_y$. Finally, note that this convergence proof holds even if $\underline{x}$ and $\underline{y}$ are vectors in an infinite dimensional Hilbert space, provided only that $\nu_x, \nu_y < 1$.

Linear Variety Constraint Sets

When the constraint sets X and Y are linear varieties, the operators $K_x()$ and $K_y()$ become simple projection matrices P_x and P_y. In this case, somewhat tighter convergence bounds can be derived by exactly the same reasoning used above:

$$\begin{aligned} \nu_x &= \left\| \mathrm{R}^{1/2}\mathrm{V}^{-1/2}\mathrm{P}_x \mathrm{H} \mathrm{P}_y \right\|_R \leq \left(\frac{\mu_{max}}{\mu_{max}+1} \right)^{1/2} \\ \nu_y &= \left\| \mathrm{V}^{1/2}\mathrm{R}^{-1/2}\mathrm{P}_y \mathrm{B} \mathrm{P}_x \right\|_V \leq \left(\frac{\lambda_{max}}{\lambda_{max}+1} \right)^{1/2} \end{aligned} \qquad \text{(F.15)}$$

Also it is easy to show that:

$$\left\| P_x H P_y B \right\|_V \le \nu_x \nu_y \qquad\qquad (F.16)$$
$$\left\| P_y B P_x H \right\|_R \le \nu_x \nu_y$$

The proof that the dual algorithm also defines a contraction mapping on each step looks quite similar to the proof above. Let $\underline{\rho}_{x_1}$, $\underline{\rho}_{x_2}$ be any arbitrary signal multipiier values, and let $\underline{\rho}_{y_1}$, $\underline{\rho}_{y_2}$ be the corresponding output multiplier estimates, $\underline{\rho}_{y_i} = -Q_y B \underline{\rho}_{x_i} + \bar{\underline{\rho}}_y$. Let $V_y = BA^{-1}QA^{-T}B^T + R$ and $V_x = A^{-1}QA^{-T}$. Then:

$$\begin{aligned}
\| \underline{\rho}_{y_1} - \underline{\rho}_{y_2} \|^2_{V_y} &= \left\| Q_y B(\underline{\rho}_{x_1} - \underline{\rho}_{x_2}) \right\|^2_{V_y} \\
&\le \left\| B(\underline{\rho}_{x_1} - \underline{\rho}_{x_2}) \right\|^2_{V_y} \\
&= \left\{ \max_{\underline{v} \ne \underline{0}} \frac{\underline{v}^T B^T V_y^{-1} B \underline{v}}{\underline{v}^T V_x^{-1} \underline{v}} \right\} (\underline{\rho}_{x_1} - \underline{\rho}_{x_2})^T V_x^{-1} (\underline{\rho}_{x_1} - \underline{\rho}_{x_2}) \\
&= \left\{ \max_{\underline{v} \ne \underline{0}} \frac{\underline{v}^T B^T \left[R^{-1} - R^{-1} B V B^T R^{-1} \right] B \underline{v}}{\underline{v}^T V_x^{-1} \underline{v}} \right\} \| \underline{\rho}_{x_1} - \underline{\rho}_{x_2} \|^2_{V_x} \\
&= \left\{ \max_{\underline{v} \ne \underline{0}} \frac{\underline{v}^T \left[V_x^{-1} - V_x^{-1} V V_x^{-1} \right] \underline{v}}{\underline{v}^T V_x^{-1} \underline{v}} \right\} \| \underline{\rho}_{x_1} - \underline{\rho}_{x_2} \|^2_{V_x} \\
&= \left\{ \max_{\underline{v} \ne \underline{0}} \frac{\underline{v}^T [V_x - V] \underline{v}}{\underline{v}^T V_x \underline{v}} \right\} \| \underline{\rho}_{x_1} - \underline{\rho}_{x_2} \|^2_{V_x} \\
&= \max_{\underline{v} \ne \underline{0}} \left\{ 1 - \frac{\underline{v}^T V_x^{-1/2} V V_x^{-T/2} \underline{v}}{\underline{v}^T \underline{v}} \right\} \| \underline{\rho}_{x_1} - \underline{\rho}_{x_2} \|^2_{V_x} \\
&= \max_{\underline{v} \ne \underline{0}} \left\{ 1 - \frac{\underline{v}^T \underline{v}}{\underline{v}^T \left(V_x^{-1/2} V V_x^{-T/2} \right)^{-1} \underline{v}} \right\} \| \underline{\rho}_{x_1} - \underline{\rho}_{x_2} \|^2_{V_x} \\
&= \max_{\underline{v} \ne \underline{0}} \left\{ 1 - \frac{\underline{v}^T V_x^{-1} \underline{v}}{\underline{v}^T V^{-1} \underline{v}} \right\} \| \underline{\rho}_{x_1} - \underline{\rho}_{x_2} \|^2_{V_x}
\end{aligned}$$

$$= \left\{ \max_{\underline{v} \neq \underline{0}} \frac{\underline{v}^T B^T R^{-1} B \underline{v}}{\underline{v}^T [B^T R^{-1} B + A^T Q^{-1} A] \underline{v}} \right\} \| \underline{p}_{x_1} - \underline{p}_{x_2} \|^2_{V_x}$$

$$= \frac{\lambda_{max}}{\lambda_{max}+1} \| \underline{p}_{x_1} - \underline{p}_{x_2} \|^2_{V_x} \qquad \text{(F.17)}$$

where this is the same convergence rate constant as in the primal algorithm (F.9).

Similarly, let $\underline{p}_{y_1}$, $\underline{p}_{y_2}$ be any arbitrary output multipliers, and let $\underline{p}_{x_1}$, $\underline{p}_{x_2}$ be the corresponding signal multiplier estimates, $\underline{p}_{x_i} = -Q_x H \underline{p}_{y_i} + \bar{\underline{p}}_x$. Then:

$$\begin{aligned}
\| \underline{p}_{x_1} - \underline{p}_{x_2} \|^2_{V_x} &= \left\| Q_x H(\underline{p}_{y_1} - \underline{p}_{y_2}) \right\|^2_{V_x} \\
&\leq \left\| H(\underline{p}_{y_1} - \underline{p}_{y_2}) \right\|^2_{V_x} \\
&= (\underline{p}_{y_1} - \underline{p}_{y_2})^T H^T V_x^{-1} H (\underline{p}_{y_1} - \underline{p}_{y_2}) \\
&\leq \left\{ \max_{\underline{v} \neq \underline{0}} \frac{\underline{v}^T H^T V_x^{-1} H \underline{v}}{\underline{v}^T V_y^{-1} \underline{v}} \right\} (\underline{p}_{y_1} - \underline{p}_{y_2})^T V_y^{-1} (\underline{p}_{y_1} - \underline{p}_{y_2}) \\
&= \left\{ \max_{\underline{v} \neq \underline{0}} \frac{\underline{v}^T R^{-1} B V V_x^{-1} V B^T R^{-1} \underline{v}}{\underline{v}^T V_y^{-1} \underline{v}} \right\} \| \underline{p}_{y_1} - \underline{p}_{y_2} \|^2_{V_y} \\
&= \left\{ \max_{\underline{v} \neq \underline{0}} \frac{\underline{v}^T \left[R^{-1} B^T V B R^{-1} - R^{-1} B^T V B R^{-1} B^T V B R^{-1} \right] \underline{v}}{\underline{v}^T V_y^{-1} \underline{v}} \right\} \| \underline{p}_{y_1} - \underline{p}_{y_2} \|^2_{V_y} \\
&= \left\{ \max_{\underline{v} \neq \underline{0}} \frac{\underline{v}^T \left[(R^{-1} - V_y^{-1}) - (R^{-1} - V_y^{-1}) R (R^{-1} - V_y^{-1}) \right] \underline{v}}{\underline{v}^T V_y^{-1} \underline{v}} \right\} \| \underline{p}_{y_1} - \underline{p}_{y_3} \|^2_{V_y} \\
&= \left\{ \max_{\underline{v} \neq \underline{0}} \frac{\underline{v}^T [V_y - R] \underline{v}}{\underline{v}^T V_y \underline{v}} \right\} \| \underline{p}_{y_1} - \underline{p}_{y_2} \|^2_{V_y} \\
&= \frac{\mu_{max}}{\mu_{max}+1} \| \underline{p}_{y_1} - \underline{p}_{y_2} \|^2_{V_y} \qquad \text{(F.18)}
\end{aligned}$$

where this is the same convergence rate constant as in the primal algorithm (F.8). These relationships imply:

$$\left\| \underline{\hat{\rho}}_{y_{k+1}} - \underline{\hat{\rho}}_{y_k} \right\|_{V_y} \le \nu_y \left\| \underline{\hat{\rho}}_{x_{k+1}} - \underline{\hat{\rho}}_{x_k} \right\|_{V_x} \le \nu_x \nu_y \left\| \underline{\hat{\rho}}_{y_k} - \underline{\hat{\rho}}_{y_{k-1}} \right\|_{V_y} \tag{F.19}$$

which implies that the dual algorithm converges at a geometric rate to the unique global minimum $\underline{\rho}_{x*}$, $\underline{\rho}_{y*}$. This also implies that:

$$\left\| \underline{\hat{\rho}}_{y_{k+1}} - \underline{\rho}_{y*} \right\|_{V_y} \le \nu_y \left\| \underline{\hat{\rho}}_{x_{k+1}} - \underline{\rho}_{x*} \right\|_{V_x} \le \nu_x \nu_y \left\| \underline{\hat{\rho}}_{y_k} - \underline{\rho}_{y*} \right\|_{V_y} \tag{F.20}$$

Appendix G

Introduction to Projection Matrices

Eigenstructure of Linear Signal Reconstruction Algorithms

1. Inner Products, Norms and Projection Matrices

In this section we will discuss basic properties of projection operators on linear subspaces. More material on this subject can be found in Youla [1] or Künzi and Krelle. [2] Many of the results below are well-known, and their proofs are therefore omitted.

Let x, y be elements of $\mathbf{R}^N$, and assume we are given an inner product on $\mathbf{R}^N$ defined by:

$$<x,y>_V = x^T V^{-1} y \tag{G.1.1}$$

where the matrix V is positive definite and symmetric. This inner product can be used to define a vector norm:

$$||x||_V = <x,x>_V^{\frac{1}{2}} \tag{G.1.2}$$

and can also be used to define a spectral radius matrix norm:

$$||T||_V = \max_{x \neq 0} \frac{||Tx||_V}{||x||_V} \tag{G.1.3}$$

This matrix norm satisfies the inequality:

$$||Tx||_V \leq ||T||_V \; ||x||_V \qquad \text{for all } x \in \mathbf{R}^n \tag{G.1.4}$$

We will call two vectors orthogonal, $x_1 \perp x_2$ if $<x_1,x_2>_V=0$, and will call a vector orthogonal to a set Λ if it is orthogonal to every element of that set. The orthogonal complement $\Lambda^{\perp}$ of the set Λ is defined as the set of all vectors which are orthogonal to Λ.

Let $g_1, \cdots, g_p$ be a set of linearly independent vectors in $\mathbf{R}^N$. Let $\Lambda \subseteq \mathbf{R}^N$ be a linear subspace defined by a set of linear equality constraints:

$$\Lambda = \left\{ x \;\middle|\; g_i^T x = 0 \quad \text{for } i=1, \cdots, p \right\} \tag{G.1.5}$$

Equivalently, we can arrange these vectors $g_1^T, \cdots, g_p^T$ as rows in a matrix G and abbreviate this definition of Λ by:

$$\Lambda = \left\{ x \;\middle|\; Gx = \underline{0} \right\} \tag{G.1.6}$$

G is a $p \times N$ matrix; because $g_1, \cdots, g_p$ are linearly independent, G has full row rank. The set Λ is thus the null space of the matrix G and has dimension $N-p$.

The orthogonal complement of Λ with respect to our norm is the set of all vectors orthogonal to Λ. This set is spanned by the linearly independent vectors Vg_i:

$$\Lambda^{\perp} = \left\{ x \;\middle|\; x = \sum_{i=1}^{p} \lambda_i V g_i \quad \text{for some } \lambda_1, \cdots \lambda_p \right\} \tag{G.1.7}$$

or equivalently:

$$\Lambda^{\perp} = \left\{ x \;\middle|\; x = VG^T\lambda \quad \text{for some } \lambda \in \mathbf{R}^p \right\} \tag{G.1.8}$$

Clearly $\Lambda^{\perp}$ has dimension p. The only point common to both Λ and $\Lambda^{\perp}$ is the origin, but together they span all of $\mathbf{R}^N$. In other words, it is possible to write any $x \in \mathbf{R}^N$ uniquely in the form:

$$x = x_{\Lambda} + x_{\Lambda^{\perp}} \qquad \text{where } x_{\Lambda} \in \Lambda \quad \text{and} \quad x_{\Lambda^{\perp}} \in \Lambda^{\perp} \tag{G.1.9}$$

From the orthogonality of Λ and $\Lambda^{\perp}$, it follows that:

$$<x_{\Lambda}, x_{\Lambda^{\perp}}>_V = 0 \tag{G.1.10}$$

In particular, if $x \in \Lambda$ then $x_{\Lambda} = x$ and $x_{\Lambda^{\perp}} = 0$; on the other hand, if $x \in \Lambda^{\perp}$ then $x_{\Lambda} = 0$ and $x_{\Lambda^{\perp}} = x$.

We call $\underline{x}_\Lambda$ the projection of $\underline{x}$ onto Λ under the norm $<\underline{x},\underline{y}>_V=\underline{x}^T V^{-1}\underline{y}$, and call $\underline{x}_\Lambda^\perp$ the projection of $\underline{x}$ onto $\Lambda^\perp$. This projection $\underline{x}_\Lambda$ is the unique vector which satisfies:

$$\underline{x}_\Lambda \in \Lambda \quad \text{and} \quad < \underline{x}_\Lambda \, , \underline{x} - \underline{x}_\Lambda >_V = 0 \tag{G.1.11}$$

Further, it has the property that it is the vector in Λ which is the unique closest element to $\underline{x}$:

$$||\underline{x} - \underline{x}_\Lambda||_V < ||\underline{x} - \tilde{\underline{x}}\,||_V \qquad \text{for all } \tilde{\underline{x}} \neq \underline{x}_\Lambda \text{ in } \Lambda \tag{G.1.12}$$

(These results form the Classical Projection Theorem; for a proof, see Luenberger [3] p.51.) To calculate the value of $\underline{x}_\Lambda$, we therefore only need to solve the problem:

$$\underline{x}_\Lambda \leftarrow \min_{\tilde{\underline{x}} \in \Lambda} ||\underline{x} - \tilde{\underline{x}}\,||_V^2 \qquad \text{subject to } G\tilde{\underline{x}} = \underline{0} \tag{G.1.13}$$

This problem can be solved by straightforward Lagrange multiplier techniques:

$$\begin{aligned} &\underline{x}_\Lambda = P\underline{x} \\ &\underline{x}_{\Lambda^\perp} = \underline{x} - \underline{x}_\Lambda = (I-P)\underline{x} \\ &\qquad\qquad \text{where} \quad P = \Big[I - VG^T(GVG^T)^{-1}G \Big] \end{aligned} \tag{G.1.14}$$

The $p \times p$ matrix GVG^T is non-singular because all the rows of G are linearly independent and V>0. Hence its inverse $(GVG^T)^{-1}$ exists, and the matrix P is well defined. The matrix P is called a projection matrix since it calculates the projection of $\underline{x}$ onto Λ under the given inner product. The matrix I−P is also a projection matrix, since it calculates the projection of $\underline{x}$ onto $\Lambda^\perp$ under the given inner product. Note the following properties:

$$\begin{aligned} &\underline{x}_\Lambda = P\underline{x} \in \Lambda \\ &\underline{x}_{\Lambda^\perp} = (I-P)\underline{x} \in \Lambda^\perp \\ &< P\underline{x} \, , \underline{x} - P\underline{x} >_V = 0 \end{aligned} \tag{G.1.15}$$

Also, if $\underline{x} \in \Lambda$ then $P\underline{x}=\underline{x}$, and if $\underline{x} \in \Lambda^\perp$ then $P\underline{x}=0$. This implies that PP=P. Furthermore, the only eigenvalues of P are 0 and 1; all the elements of Λ are eigenvectors of P with eigenvalue of 1, and all the elements of $\Lambda^{\perp}$ are eigenvectors of P with eigenvalue of 0. The range of P is the null space of G; the null space of P is the orthogonal complement of the null space of G. Finally, note that:

$$\begin{aligned} \|x\|_V^2 &= \|Px + (I-P)x\|_V^2 \\ &= \|Px\|_V^2 + 2 < Px\ ,\ (I-P)x >_V + \|(I-P)x\|_V^2 \\ &= \|Px\|_V^2 + \|(I-P)x\|_V^2 \end{aligned} \tag{G.1.16}$$

and thus:

$$\|Px\|_V \leq \|x\|_V \tag{G.1.17}$$

with equality if and only if $x \in \Lambda$. This implies that:

$$\|Px_1 - Px_2\|_V = \|P(x_1 - x_2)\|_V \leq \|x_1 - x_2\|_V \tag{G.1.18}$$

and thus P is a non-expansive mapping (compare with Theorem D.2 in Appendix D.)

2. Eigenvalues and Eigenvectors of Linear Variety Iteration

When the constraint sets of our iterative signal reconstruction algorithms of chapter 5 are linear varieties, then the convergence behavior of our algorithms can be analyzed in much greater detail than in Appendix F. We showed in chapter 5 that the error between the estimates $\hat{x}_k$, $\hat{y}_k$ and the solution to the problem $x_\bullet$, $y_\bullet$ satisfies:

$$\begin{aligned} \hat{x}_{k+1} - x_\bullet &= P_x H P_y 3 P_x (\hat{x}_k - x_\bullet) \\ \hat{y}_{k+1} - y_\bullet &= P_y B P_x H P_y (\hat{y}_k - y_\bullet) \end{aligned} \tag{G.2.1}$$

Analyzing the eigenvalues and eigenvectors of these two matrices $P_x H P_y B P_x$ and $P_y B P_x H P_y$ should thus be very informative about the convergence properties of our algorithms.

Let $R^{1/2}$ and $V^{1/2}$ be square roots of R and V. Then:

$$\begin{aligned} R^{-1/2} P_y B P_x H P_y R^{1/2} &= \left(R^{-1/2} P_y B V^{1/2} \right) \\ &\cdot \left[I - V^{T/2} G_x^T (G_x V G_x^T)^{-1} G_x V^{1/2} \right] \left(V^{T/2} B^T P_y^T R^{-1/2} \right) \end{aligned} \tag{G.2.2}$$

The matrix in brackets in the center of the right hand side is a symmetric projection matrix with eigenvalues 0 and 1. Therefore $R^{-1/2}P_yBP_xHP_yR^{1/2}$ must be a symmetric positive semi-definite matrix. Its eigenvalues λ_i will thus all be real and non-negative, and its eigenvectors $\tilde{\psi}_i$ must form a complete orthonormal basis for $\mathbf{R}^M$, $\tilde{\psi}_i^T\tilde{\psi}_j = \delta_{ij}$. But if we let $\psi_i = R^{1/2}\tilde{\psi}_i$, then:

$$P_yBP_xHP_y\psi_i = R^{1/2}\left(R^{-1/2}P_yBP_xHP_yR^{1/2}\tilde{\psi}_i\right) = \lambda_i R^{1/2}\tilde{\psi}_i = \lambda_i\psi_i \tag{G.2.3}$$

and:

$$<\psi_i,\psi_j>_R = \psi_i^T R^{-1}\psi_j = \tilde{\psi}_i^T\tilde{\psi}_j = \delta_{ij} \tag{G.2.4}$$

Thus the vectors ψ_i are eigenvectors of $P_yBP_xHP_y$ with the same real, non-negative eigenvalues λ_i, and they form a complete orthonormal basis for $\mathbf{R}^M$ under the inner product $<\cdot,\cdot>_R$.

The arguments in Appendix F can be used to prove that:

$$\left\| P_yBP_xHP_y \right\|_R \le \nu_x\nu_y \tag{G.2.5}$$

Expanding this gives:

$$\begin{aligned}\left\| P_yBP_xHP_y \right\|_R^2 &= \max_{\underline{v}\neq\underline{0}} \frac{\underline{v}^T P_y^T H^T P_x^T B^T P_y^T R^{-1} P_y B P_x H P_y \underline{v}}{\underline{v}^T R^{-1}\underline{v}} \\ &= \max_{\underline{u}\neq\underline{0}} \frac{\underline{u}^T\left(R^{-1/2}P_yBP_xHP_yR^{1/2}\right)^2\underline{u}}{\underline{u}^T\underline{u}}\end{aligned} \tag{G.2.6}$$

$$\text{where: } \underline{u} = R^{1/2}\underline{v}$$

The norm $\left\| P_yBP_xHP_y \right\|_R^2$ is thus equal to the maximum eigenvalue of $R^{-1/2}P_yBP_xHP_yR^{1/2}$ squared, which in turn is equal to the maximum eigenvalue of $P_yBP_xHP_y$ squared. Combining (G.2.5) and (G.2.6) thus shows that the eigenvalues of $P_yBP_xHP_y$ are not only real and non-negative, but are also bounded above by $\nu_x\nu_y$.

The eigenvalues and eigenvectors of the matrix $P_x HP_y BP_x$ can be analyzed in exactly the same way; the eigenvectors can be shown to form a complete orthonormal basis of $\mathbf{R}^N$ under the inner product $<\cdot,\cdot>_V$, and the eigenvalues are all real, non-negative, and bounded above by $\nu_x \nu_y$.

As might be expected, the eigenstructures of these two matrices are closely related. Let us look at the non-zero eigenvalues λ_i of $P_y BP_x HP_y$, and their corresponding eigenvectors $\boldsymbol{\psi}_i$. Since:

$$P_y BP_x HP_y \boldsymbol{\psi}_i = \lambda_i \boldsymbol{\psi}_i \neq \underline{0} \tag{G.2.7}$$

all the eigenvectors $\boldsymbol{\psi}_i$ must be in the range of P_y, which implies that they are in the $M-q$ dimensional null space of G_y. This implies that $P_y \boldsymbol{\psi}_i = \boldsymbol{\psi}_i$. Define the vectors $\boldsymbol{\phi}_i$ by:

$$\boldsymbol{\phi}_i = \frac{1}{\sqrt{\lambda_i}} P_x HP_y \boldsymbol{\psi}_i = \frac{1}{\sqrt{\lambda_i}} P_x H \boldsymbol{\psi}_i \tag{G.2.8}$$

Now since $P_x P_x = P_x$:

$$\begin{aligned}
\left(P_x HP_y BP_x\right) \boldsymbol{\phi}_i &= \left(P_x HP_y BP_x\right) \left(\frac{1}{\sqrt{\lambda_i}} P_x HP_y \boldsymbol{\psi}_i\right) \\
&= \frac{1}{\sqrt{\lambda_i}} P_x H \left(P_y BP_x HP_y \boldsymbol{\psi}_i\right) \\
&= \lambda_i \left(\frac{1}{\sqrt{\lambda_i}} P_x H \boldsymbol{\psi}_i\right) \\
&= \lambda_i \boldsymbol{\phi}_i
\end{aligned} \tag{G.2.9}$$

Thus $\boldsymbol{\phi}_i$ is an eigenvector of $P_x HP_y BP_x$ with the same non-zero eigenvalue λ_i as $\boldsymbol{\psi}_i$. Furthermore:

$$\begin{aligned}
<\boldsymbol{\phi}_i, \boldsymbol{\phi}_j>_V &= \frac{1}{\sqrt{\lambda_i \lambda_j}} \boldsymbol{\phi}_i^T V^{-1} \boldsymbol{\phi}_j \\
&= \frac{1}{\sqrt{\lambda_i \lambda_j}} \boldsymbol{\psi}_i^T P_y^T H^T P_x^T V^{-1} P_x HP_y \boldsymbol{\psi}_j
\end{aligned}$$

$$= \frac{1}{\sqrt{\lambda_i \lambda_j}} \psi_i^T \mathbf{R}^{-1} \left(\mathbf{P}_y \mathbf{B} \mathbf{P}_x \mathbf{H} \mathbf{P}_y \psi_j \right)$$

$$= \frac{\lambda_j}{\sqrt{\lambda_i \lambda_j}} < \psi_i , \psi_j >_{\mathbf{R}}$$

$$= \delta_{ij} \tag{G.2.10}$$

and thus the ϕ_i form an orthonormal set with respect to the inner product $<\cdot,\cdot>_{\mathbf{V}}$. Since:

$$\left(\mathbf{P}_x \mathbf{H} \mathbf{P}_y \mathbf{B} \mathbf{P}_x \right) \phi_i = \lambda_i \, \phi_i \neq \underline{0} \tag{G.2.11}$$

the eigenvectors ϕ_i must all be in the range of $\mathbf{P}_x$, which implies that $\mathbf{P}_x \phi_i = \phi_i$, and that all the ϕ_i are in the $N - p$ dimensional null space of $\mathbf{G}_x$. Also, note that:

$$\psi_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{P}_y \mathbf{B} \mathbf{P}_x \, \phi_i \tag{G.2.12}$$

To summarize, the non-zero eigenvalues of $\mathbf{P}_y \mathbf{B} \mathbf{P}_x \mathbf{H} \mathbf{P}_y$ and $\mathbf{P}_x \mathbf{H} \mathbf{P}_y \mathbf{B} \mathbf{P}_x$ are identical, there is a one-to-one mapping between the eigenvectors ψ_i and ϕ_i corresponding to each non-zero eigenvalue, and these eigenvectors are elements of the null spaces of $\mathbf{G}_y$ and $\mathbf{G}_x$ respectively. This last fact also guarantees that there can be at most $\min(N - p, M - q)$ non-zero eigenvalues.

Finally, we can also show that the matrices $\mathbf{P}_y \mathbf{B} \mathbf{P}_x \mathbf{H}$ and $\mathbf{P}_x \mathbf{H} \mathbf{P}_y \mathbf{B}$ have exactly the same eigenvalues of $\mathbf{P}_y \mathbf{B} \mathbf{P}_x \mathbf{H} \mathbf{P}_y$ and $\mathbf{P}_x \mathbf{H} \mathbf{P}_y \mathbf{B} \mathbf{P}_x$ respectively. The eigenvectors corresponding to the non-zero eigenvalues must also be the same; however, the eigenvectors corresponding to the zero eigenvalues may be quite different.

References

1. Dante Youla, "Generalized Image Restoration by the Method of Alternating Orthogonal Projections," *IEEE Trans. Circuits. Syst.* CAS-25(9), pp.694-702 (Sept 1978).
2. Hans Künzi and Wilhelm Krelle, *Nonlinear Programming*, Blaisdell Publishing, Waltham, Mass. (1966).
3. David G. Luenberger, *Optimization By Vector Space Methods*, John Wiley & Sons Inc., New York (1969).

Appendix H
Asymptotic Behavior of Signal Reconstruction Algorithms With Flat *A Priori* Signal Density

In this Appendix we will prove that the four "Bayesian" MCEM and MAP algorithms in chapter 5 asymptotically locate the solution to the corresponding "Fisher" problem with the minimum average signal energy. In each of our algorithms, for all $\alpha>0$ assume that the cross-entropy expression $H_\alpha(q_X, q_Y)$ achieves its global minimum at some densities $\hat{q}_{X,\alpha}$, $\hat{q}_{Y,\alpha}$. (Recall that the MAP algorithms are guaranteed to have such a solution, and the MCEM algorithm is guaranteed to have such a solution if X, Y are convex.) Also assume that the Fisher cross-entropy $H_{ML}(q_X, q_Y)$ achieves its global minimum at a pair of densities with finite average signal energy, and let $\Psi=\{(\hat{q}_X, \hat{q}_Y)\}$ be the set of all such global minimizers of H_{ML}. All global minimizers must be stationary points of our algorithm, and therefore must have the form given in table H.2 below. The minimum Bayesian cross-entropy is bounded above by:

$$\begin{aligned} H_\alpha(\hat{q}_{X,\alpha}, \hat{q}_{Y,\alpha}) &\leq H_\alpha(\hat{q}_X, \hat{q}_Y) \\ &= H_{ML}(\hat{q}_X, \hat{q}_Y) + \frac{\alpha}{2}\int_X \|Ax\|^2_{\hat{Q}_0}\, \hat{q}_X(x)dx + \frac{N}{2}\log\left|\frac{1}{\alpha}Q_0\right| \end{aligned} \tag{H.1}$$

where $\hat{q}_X$, $\hat{q}_Y$ is any element of Ψ. H_α is also bounded below by:

$$\begin{aligned} H_\alpha(\hat{q}_{X,\alpha}, \hat{q}_{Y,\alpha}) &= H_{ML}(\hat{q}_{X,\alpha}, \hat{q}_{Y,\alpha}) + \frac{\alpha}{2}\int_X \|Ax\|^2_{\hat{Q}_0}\, \hat{q}_{X,\alpha}(x)\, dx + \frac{N}{2}\log\left|\frac{1}{\alpha}Q_0\right| \\ &\geq H_{ML}(\hat{q}_{X,\alpha}, \hat{q}_{Y,\alpha}) + \frac{N}{2}\log\left|\frac{1}{\alpha}Q_0\right| \\ &\geq H_{ML}(\hat{q}_X, \hat{q}_Y) + \frac{N}{2}\log\left|\frac{1}{\alpha}Q_0\right| \end{aligned} \tag{H.2}$$

Thus:

$$H_{ML}(\hat{q}_X,\hat{q}_Y) \le H_\alpha(\hat{q}_{X,\alpha},\hat{q}_{Y,\alpha}) - \frac{N}{2}\log\left|\frac{1}{\alpha}Q_0\right| \qquad \text{(H.3)}$$

$$\le H_{ML}(\hat{q}_X,\hat{q}_Y) + \frac{\alpha}{2}\int_X \|Ax\|^2_{Q_0}\,\hat{q}_X(x)\,dx$$

Taking the limit as $\alpha\to 0$ gives:

$$\lim_{\alpha\to 0}\left\{H_\alpha(\hat{q}_{X,\alpha},\hat{q}_{Y,\alpha}) - \frac{N}{2}\log\left|\frac{1}{\alpha}Q_0\right|\right\} = H_{ML}(\hat{q}_X,\hat{q}_Y) \qquad \text{(H.4)}$$

Thus the minimum Bayesian cross-entropy, adjusted by $\frac{N}{2}\log\left|\frac{1}{\alpha}Q_0\right|$, asymptotically approaches the minimum Fisher cross-entropy from above. Now:

$$H_\alpha(\hat{q}_{X,\alpha},\hat{q}_{Y,\alpha}) \le H_\alpha(\hat{q}_X,\hat{q}_Y) \qquad \text{(H.5)}$$

which using (6.2.1) means:

$$H_{ML}(\hat{q}_{X,\alpha},\hat{q}_{Y,\alpha}) + \frac{N}{2}\log\left|\frac{1}{\alpha}Q_0\right| + \frac{\alpha}{2}\int_X \|Ax\|^2_{Q_0}\,\hat{q}_{X,\alpha}(x)\,dx \qquad \text{(H.6)}$$

$$\le H_{ML}(\hat{q}_X,\hat{q}_Y) + \frac{N}{2}\log\left|\frac{1}{\alpha}Q_0\right| + \frac{\alpha}{2}\int_X \|Ax\|^2_{Q_0}\,\hat{q}_X(x)\,dx$$

But:

$$H_{ML}(\hat{q}_X,\hat{q}_Y) \le H_{ML}(\hat{q}_{X,\alpha},\hat{q}_{Y,\alpha}) \qquad \text{(H.7)}$$

and thus combining (H.6) and (H.7):

$$\int_X \|Ax\|^2_{Q_0}\,\hat{q}_{X,\alpha}(x)\,dx \le \int_X \|Ax\|^2_{Q_0}\,\hat{q}_X(x)\,dx \qquad \text{(H.8)}$$

for all $\alpha>0$ and for any Fisher global minimizing density $\hat{q}_X \in \Psi$. This implies that:

$$\int_X \|Ax\|^2_{Q_0}\,\hat{q}_{X,\alpha}(x)\,dx \le \inf_{\hat{q}_X\in\Psi}\int_X \|Ax\|^2_{Q_0}\,\hat{q}_X(x)\,dx \qquad \text{(H.9)}$$

Thus the average signal energy of the Bayesian estimate is always smaller than that of any of the Fisher estimates. Also, if we let $\hat{x}_\alpha$ be the mean of $\hat{q}_{X,\alpha}$, then:

$$\|A\hat{x}_\alpha\|^2_{Q_0} \leq \int_X \|Ax\|^2_{Q_0}\, \hat{q}_{X,\alpha}(x)\, dx \tag{H.10}$$

which, in combination with (H.9), implies that the Bayesian signal mean estimate $\hat{x}_\alpha$ must be bounded for all α.

We now prove that the Bayesian estimates $\hat{q}_{X,\alpha}$, $\hat{q}_{Y,\alpha}$ approach the Fisher estimates $\hat{q}_X$, $\hat{q}_Y$ with minimal signal energy as $\alpha \to 0$. The Bayesian signal and output density estimates will have the form:

	$\hat{q}_{X,\alpha}$	$\hat{q}_{\Phi,\alpha}$
MCEM:	$N_X(H_\alpha\hat{y}_\alpha, V_\alpha)$	$N_Y(\hat{x}_\alpha, R)$
XMAP:	$\delta(x - \hat{x}_\alpha)$	$N_Y(\hat{x}_\alpha, R)$
YMAP:	$N_X(H_\alpha\hat{y}_\alpha, V_\alpha)$	$\delta(y - \hat{y}_\alpha)$
XYMAP:	$\delta(x - \hat{x}_\alpha)$	$\delta(y - \hat{y}_\alpha)$

Table H.1 - Bayesian Density Estimates

Let $\alpha_1 > \alpha_2 > \cdots$ be any monotonically decreasing sequence of values of α which satisfy $\lim_{i \to \infty} \alpha_i = 0$. Let $\hat{q}_{X,\alpha_i}$, $\hat{q}_{Y,\alpha_i}$ be the corresponding density estimates and let $\hat{x}_{\alpha_i}$ be the signal mean. Since $\hat{x}_{\alpha_i}$ is finite dimensional and bounded, by the Bolzano-Weierstrass theorem there must be a convergent subsequence $\{\hat{x}_{\alpha_i'}\} \subseteq \{\hat{x}_{\alpha_i}\}$ with limit point $\hat{x}'$. Let $\hat{q}_{X,\alpha_i'}$, $\hat{q}_{Y,\alpha_i'}$ be the corresponding sequence of densities, and let $\hat{y}_{\alpha_i'}$ be the corresponding output estimates. Since $\hat{x}_\alpha'$ remains bounded, so must $\hat{y}_\alpha'$, and thus we can further trim the sequence α_i, if necessary, so that $\hat{y}_{\alpha_i'}$ also converges to a limit $\hat{y}'$. Let $\hat{q}_X'$, $\hat{q}_Y'$ be the densities corresponding to $\alpha=0$ and centers $\hat{x}'$, $\hat{y}'$; since $H_\alpha \to I$ and $V_\alpha \to R$ as $\alpha \to 0$, these densities will have the form:

	$\hat{q}_X'(x)$	$\hat{q}_Y'(y)$
MCEM:	$N_X(\hat{y}', R)$	$N_Y(\hat{x}', R)$
XMAP:	$\delta(x - \hat{x}')$	$N_Y(\hat{x}', R)$
YMAP:	$N_X(\hat{y}', R)$	$\delta(y - \hat{y}')$
XYMAP:	$\delta(x - \hat{x}')$	$\delta(y - \hat{y}')$

Table H.2 - Limiting Density

It is straightforward to show that in all four cases, $H_{ML}(\hat{q}_{X,\alpha}, \hat{q}_{Y,\alpha})$ will be an analytic function of the covariance $Q = \frac{1}{\alpha} Q_0$, and of the centers $\hat{x}_\alpha$ and $\hat{y}_\alpha$. Thus:

$$\lim_{i \to \infty} H_{ML}(\hat{q}_{X,\alpha_i}', \hat{q}_{Y,\alpha_i}') = H_{ML}(\hat{q}_X', \hat{q}_Y') \tag{H.11}$$

Also the average signal energy will be an analytic function of α, $\hat{x}_\alpha$ and $\hat{y}_\alpha$, and so inequality (H.9) must still be satisfied at the limit $\hat{x}'$, $\hat{y}'$:

$$\begin{aligned} \lim_{i \to \infty} \int_X \|Ax\|_{Q_0}^2 \, \hat{q}_{X,\alpha_i}'(x) \, dx &= \int_X \|Ax\|_{Q_0}^2 \, \hat{q}_X'(x) \, dx \\ &\leq \inf_{q_X \in \Psi} \int_X \|Ax\|_{Q_0}^2 \, \hat{q}_X(x) \, dx \end{aligned} \tag{H.12}$$

Also:

$$\begin{aligned} \lim_{i \to \infty} H_\alpha(\hat{q}_{X,\alpha_i}', \hat{q}_{Y,\alpha_i}') - \frac{N}{2} \log \left| \frac{1}{\alpha} Q_0 \right| &= \lim_{\alpha \to 0} H_\alpha(\hat{q}_{X,\alpha}, \hat{q}_{Y,\alpha}) - \frac{N}{2} \log \left| \frac{1}{\alpha} Q_0 \right| \\ &= \inf_{q_X, q_Y} H_{ML}(q_X, q_Y) \end{aligned} \tag{H.13}$$

Combining (6.2.1), (H.11), (H.12), and (H.13):

$$H_{ML}(\hat{q}_X{}'\hat{q}_Y{}') = \inf_{q_X,q_Y} H_{ML}(q_X,q_Y) \tag{H.14}$$

Thus every limit $\hat{q}_X{}'$, $\hat{q}_Y{}'$ of the Bayesian estimates must be a solution of the Fisher cross-entropy minimization problem; because of (H.12), it must also be a solution with the least possible average signal energy.

A converse to this result can also be proved. If the average signal energy $\int_X \|Ax\|^2_{\hat{Q}_0} \hat{q}_{X,\alpha}(x)\,dx$ of the Bayesian estimator remains bounded as $\alpha \to 0$, then the corresponding Fisher problem must achieve its global minimum at the density corresponding to any limit of the sequence. To prove this, suppose that there is some sequence of estimates $\hat{x}_{\alpha_i}$, $\hat{y}_{\alpha_i}$ which is bounded as $\alpha_i \to 0$. Since the sequence must have at least one limit point, we can find a convergent subsequence $\hat{x}_{\alpha_i}{}'$, $\hat{y}_{\alpha_i}{}'$ with limit $\hat{x}'$, $\hat{y}'$. Let $\hat{q}_X{}'$, $\hat{q}_Y{}'$ be the densities with the form in table H.2 corresponding to $\hat{x}'$, $\hat{y}'$. Suppose that $\hat{q}_X{}'$, $\hat{q}_Y{}'$ is not a global minimizer of the Fisher problem. Then there is some pair of densities $\hat{q}_X{}''$, $\hat{q}_Y{}''$ with the form given in table H.2 such that:

$$\epsilon = H_{ML}(\hat{q}_X{}',\hat{q}_Y{}') - H_{ML}(\hat{q}_X{}'',\hat{q}_Y{}'') > 0 \tag{H.15}$$

(We could find such a pair of truncated Gaussians by applying one pass of our iterative algorithm to any pair of densities with lower cross-entropy than $\hat{q}_X{}'$, $\hat{q}_Y{}'$.) Because all the cross-entropy expressions and the average signal energy are analytic functions of α, $\hat{x}$ and $\hat{y}$, we can choose an $\alpha_0 > 0$ such that:

$$\left| H_{ML}(\hat{q}_{X,\alpha_i}{}'\hat{q}_{Y,\alpha_i}{}') - H_{ML}(\hat{q}_X{}',\hat{q}_Y{}') \right| \le \frac{\epsilon}{4} \qquad \text{for all } \alpha_i \le \alpha_0 \tag{H.16}$$

and:

$$\frac{\alpha_i}{2} \left| \int_X \|Ax\|^2_{\hat{Q}_0} \hat{q}_{X,\alpha_i}{}'(x)\,dx - \int_X \|Ax\|^2_{\hat{Q}_0} \hat{q}_X{}''(x)\,dx \right| \le \frac{\epsilon}{4} \tag{H.17}$$

Then for $\alpha \leq \alpha_0$:

$$
\begin{aligned}
&H_{\alpha_i}(\hat{q}_{X,\alpha_i}{}', \hat{q}_{Y,\alpha_i}{}') \qquad\qquad \text{(H.18)} \\
&\quad = H_{ML}(\hat{q}_{X,\alpha_i}{}', \hat{q}_{Y,\alpha_i}{}') + \frac{\alpha_i}{2} \int_X \|\mathrm{A}x\|^2_{\hat{Q}_0} \hat{q}_{X,\alpha_i}{}'(x)\, dx + \frac{N}{2} \log \left| \frac{1}{\alpha} \mathrm{Q}_0 \right| \\
&\quad \geq \frac{\epsilon}{2} + H_{ML}(\hat{q}_X{}'', \hat{q}_Y{}'') + \frac{\alpha_i}{2} \int_X \|\mathrm{A}x\|^2_{\hat{Q}_0} qhx''(x)\, dx + \frac{N}{2} \log \left| \frac{1}{\alpha} \mathrm{Q}_0 \right| \\
&\quad \geq H_{\alpha_i}(\hat{q}_X{}'', \hat{q}_Y{}'') + \frac{\epsilon}{2}
\end{aligned}
$$

But this contradicts the fact that $\hat{q}_{X,\alpha_i}{}'$, $\hat{q}_{Y,\alpha_i}{}'$ is a global minimizer of H_{α_i}. Thus the limit $\hat{q}_X{}'$, $\hat{q}_Y{}'$ of the Bayesian estimates must be a global minimizing solution to the Fisher problem. Reversing the direction of the result guarantees that if the Fisher problem has no global minimizing solution, then the Bayesian estimates can not have bounded signal energy as $\alpha \rightarrow 0$.

Appendix I

Convergence of Fisher Signal Reconstruction Algorithms of Chapter 6

1. Convergence to Stationary Point

In this Appendix we prove theorem 6.2, showing that when the constraint sets are convex then the Fisher signal reconstruction algorithms of chapter 6 are guaranteed to converge to a finite global minimizing solution to the Fisher estimation problem if and only if a finite global minimizer exists. Assume that X and Y are convex and measurable sets. Define the operators $K_x()$ and $K_y()$ corresponding to each step of our algorithms by:

$$K_x(\hat{y}) = \begin{cases} \mathrm{E}_X[x \mid \hat{y}\,] & \text{for MCEM, YMAP} \qquad \text{(I.1.1a)} \\ \min\limits_{x \in X} \|x - \hat{y}\,\|_R^2 & \text{for XMAP, XYMAP} \qquad \text{(I.1.1b)} \end{cases}$$

$$K_y(\hat{x}) = \begin{cases} \mathrm{E}_Y[y \mid \hat{x}\,] & \text{for MCEM, XMAP} \qquad \text{(I.1.1c)} \\ \min\limits_{y \in Y} \|y - \hat{x}\,\|_R^2 & \text{for YMAP, XYMAP} \qquad \text{(I.1.1d)} \end{cases}$$

where $\mathrm{E}_X[x \mid \hat{y}\,]$ is the conditional expectation of $x \in X$ for a truncated Gaussian $N_X(\hat{y},\mathrm{R})$ centered at $\hat{y}$. Similarly, $\mathrm{E}_Y[y \mid \hat{x}\,]$ is the conditional expectation of $y \in Y$ of a truncated Gaussian $N_Y(\hat{x},\mathrm{R})$ centered at $\hat{x}$. By theorem D.1 of appendix D, the minimization problems in (I.1.1b) and (I.1.1d) have unique solutions because X and Y are convex. Each iteration of our algorithms can now be written in the form:

$$\begin{aligned} \hat{x}_{k+1} &= K_x(\hat{y}_k) \\ \hat{y}_{k+1} &= K_y(\hat{x}_{k+1}) \end{aligned} \qquad \text{(I.1.2)}$$

Theorem D.2 of Appendix D guarantees that if X and Y are convex sets then the conditional expectation operators (I.1.1a) and (I.1.1c) must be non-expansive mappings. Similarly, theorem E.3 of Appendix E guarantees that the projection mappings (I.1.1b) and (I.1.1d) must be non-expansive. Thus in all four of our algorithms, for any $\underline{x}_1, \underline{x}_2 \in X$:

$$\| K_y(\underline{x}_1) - K_y(\underline{x}_2) \|_{\mathbf{R}} \leq \| \underline{x}_1 - \underline{x}_2 \|_{\mathbf{R}} \tag{I.1.3}$$

and for any $\underline{y}_1, \underline{y}_2 \in Y$:

$$\| K_x(\underline{y}_1) - K_x(\underline{y}_2) \|_{\mathbf{R}} \leq \| \underline{y}_1 - \underline{y}_2 \|_{\mathbf{R}} \tag{I.1.4}$$

Letting $\underline{x}_1 = \hat{\underline{x}}_{k+1}$, $\underline{x}_2 = \hat{\underline{x}}_k$, $\underline{y}_1 = \hat{\underline{y}}_k$, $\underline{y}_2 = \hat{\underline{y}}_{k-1}$, and using (I.1.2) gives:

$$\| \hat{\underline{y}}_{k+1} - \hat{\underline{y}}_k \|_{\mathbf{R}} \leq \| \hat{\underline{x}}_{k+1} - \hat{\underline{x}}_k \|_{\mathbf{R}} \leq \| \hat{\underline{y}}_k - \hat{\underline{y}}_{k-1} \|_{\mathbf{R}} \tag{I.1.5}$$

Thus the step size in our algorithms decreases on each step.

Now suppose that the estimates $\hat{\underline{x}}_k$ and $\hat{\underline{y}}_k$ remain bounded. By the Bolzano-Weierstrass theorem, if $\underline{x}$ and $\underline{y}$ are finite dimensional, then there must be a convergent subsequence $\{\hat{\underline{x}}_k{}', \hat{\underline{y}}'\} \subseteq \{\hat{\underline{x}}_k, \hat{\underline{y}}_k\}$ with limit points $\hat{\underline{x}}'$, $\hat{\underline{y}}'$. Since all four Fisher cross-entropy expressions are analytic functions of $\hat{\underline{x}}$ and $\hat{\underline{y}}$, the convergence theorems of Appendix B guarantee that $\hat{\underline{x}}'$, $\hat{\underline{y}}'$ must be a stationary point of the algorithm and a critical point or local minimum of the appropriate Fisher cross-entropy expression:

$$\begin{aligned} \hat{\underline{x}}' &= K_x(\hat{\underline{y}}') \\ \hat{\underline{y}}' &= K_y(\hat{\underline{x}}') \end{aligned} \tag{I.1.6}$$

Letting $\underline{x}_1 = \hat{\underline{x}}'$, $\underline{x}_2 = \hat{\underline{x}}_k{}'$, $\underline{y}_1 = \hat{\underline{y}}'$, $\underline{y}_2 = \hat{\underline{y}}_k$, and using (I.1.2) and (I.1.6) recursively gives:

$$\| \hat{\underline{y}}_{k+1}{}' - \hat{\underline{y}}' \|_{\mathbf{R}} \leq \| \hat{\underline{x}}_{k+1}{}' - \hat{\underline{x}}' \|_{\mathbf{R}} \leq \| \hat{\underline{y}}_k{}' - \hat{\underline{y}}' \|_{\mathbf{R}} \tag{I.1.7}$$

Thus the distance from the estimates to the limit point $\hat{\underline{x}}'$, $\hat{\underline{y}}'$ must decrease on every step. Since $\hat{\underline{x}}_k{}', \hat{\underline{y}}_k{}' \rightarrow \hat{\underline{x}}', \hat{\underline{y}}'$, for any $\epsilon > 0$ we can find an L sufficiently large that:

$$\| \hat{\underline{x}}_k{}' - \hat{\underline{x}}' \|_{\mathbf{R}} \leq \epsilon \quad \text{and} \quad \| \hat{\underline{y}}_k{}' - \hat{\underline{y}}' \|_{\mathbf{R}} \leq \epsilon \quad \text{for all } k \geq L \tag{I.1.8}$$

But $\{\hat{x}_k{}', \hat{y}_k{}'\}$ is a subsequence of the set $\{\hat{x}_k, \hat{y}_k\}$; thus the estimates $\hat{x}_L{}'$, $\hat{y}_L{}'$ correspond to some element $\hat{x}_K$, $\hat{y}_K$ of the original sequence:

$$\|\hat{x}_K - \hat{x}'\|_R \le \epsilon \quad \text{and} \quad \|\hat{y}_K - \hat{y}'\|_R \le \epsilon \tag{I.1.9}$$

But now applying our non-expansive argument recursively:

$$\|\hat{x}_k - \hat{x}'\|_R \le \epsilon \quad \text{and} \quad \|\hat{y}_k - \hat{y}'\|_R \le \epsilon \quad \text{for all } k \ge K \tag{I.1.10}$$

Since ϵ is arbitrarily small, the sequence $(\hat{x}_k, \hat{y}_k)$ must converge to the limit $(\hat{x}', \hat{y}')$, and this can be the only limit point of the sequence.

We have thus proven that if the estimates remain bounded, then they converge to a stationary point of the algorithm. Let us now prove that if a stationary point of the algorithm exists, then the estimates are bounded and thus converge. Suppose $\hat{x}'$, $\hat{y}'$ is a limit point of the algorithm satisfying (I.1.6). Letting $x_1=\hat{x}'$, $x_2=\hat{x}_k$, $y_1=\hat{y}'$, $y_2=\hat{y}_k$ as in (I.1.7) we again get:

$$\|\hat{y}_{k+1} - \hat{y}'\|_R \le \|\hat{x}_{k+1} - \hat{x}'\|_R \le \|\hat{y}_k - \hat{y}'\|_R \tag{I.1.11}$$

Applying this recursively gives:

$$\|\hat{y}_k - \hat{y}'\|_R \le \|\hat{x}_k - \hat{x}'\|_R \le \|\hat{y}_0 - \hat{y}'\|_R \quad \text{for all } k \tag{I.1.12}$$

Thus the estimates $\hat{x}_k$, $\hat{y}_k$ remain within a fixed ball about $\hat{x}'$, $\hat{y}'$ with radius $\|\hat{y}_0 - \hat{y}'\|_R$, and are therefore bounded. Convergence follows from our previous argument.

2. All Stationary Points are Global Minimizers

To complete the argument, we need only show in all our algorithms that the only possible stationary points are the global minimizing solutions. Let us first transform variables to:

$$\tilde{x} = R^{-\frac{1}{2}} x \tag{I.2.1}$$

$$\bar{\underline{y}} = -R^{-½}\underline{y}$$

The transformed probability density is then in its "natural" exponential form:

$$p(\bar{\underline{y}}\,|\bar{\underline{x}}) = \left\{ \frac{1}{(2\pi)^{N/2}} \exp(-½\bar{\underline{y}}^T\bar{\underline{y}}) \right\} \left\{ \exp(-½\bar{\underline{x}}^T\bar{\underline{x}}) \right\} \exp(\bar{\underline{y}}^T\bar{\underline{x}}) \qquad (I.2.2)$$

$$= g(\bar{\underline{y}})h(\bar{\underline{x}})\exp(\bar{\underline{y}}^T\bar{\underline{x}})$$

where $g(\bar{\underline{y}})$ and $h(\bar{\underline{x}})$ are defined in an obvious way. This density is also log concave and the second derivatives of $\log g(\bar{\underline{y}})$ and $\log h(\bar{\underline{x}})$ are both negative definite. The derivation in section 3 of Appendix E then guarantees that the cross-entropy function of all of our estimation methods can be transformed into concave functions. This guarantees that any stationary point will have to be a global optimizing solution, and also guarantees that the sets of global optimizers $(\hat{\underline{x}},\hat{\underline{y}})$ must form a closed, convex set. As a result, our algorithm will be bounded and converge to a finite global minimizing solution in the closed convex set of such solutions, if and only if such a solution exists. Otherwise the densities must be unbounded and diverge.

3. Alternative Corollaries

By strengthening the assumptions, we can strengthen the conclusion of theorem 6.1.

Theorem I.1 Let X, Y be convex, closed and non-empty sets. Assume that there exist subsets $X' \subseteq X$ and $Y' \subseteq Y$ such that either X' or Y' is bounded, and such that the operator $K_x()$ maps Y' onto X', and the operator $K_y()$ maps X' onto Y'. Then all four iterative algorithms are guaranteed to converge to a finite global minimum solution to the corresponding Fisher problem.

Proof: Because $K_x()$ and $K_y()$ are continuous and finite valued mappings, and X' and

Y' are their respective ranges, if X' is bounded then Y' must be bounded as well (see, for example, Ortega and Rheinboldt [1] p.404) The sequence of estimates $(\hat{x}_k, \hat{y}_k) \in X' \times Y'$ is thus bounded, and theorem 6.2 applies immediately.

Corollary 6.2 If X and Y are convex, closed and non-empty, and if either X or Y are bounded, then all four iterative algorithms are guaranteed to converge to a finite global minimum solution.

Proof: Let $X'=X$ and $Y'=Y$, and apply theorem I.1.

4. XYMAP

Because XYMAP simply involves minimizing a quadratic function over some domain, several additional results can be proven for this algorithm. In particular:

Theorem I.2 Let X and Y be convex, closed and non-empty. Then not only is the set of global maximizers to the XYMAP problem closed and convex (though possibly empty), but also if $(\hat{x}_1, \hat{y}_1)$ and $(\hat{x}_2, \hat{y}_2)$ are two global maximum solutions, then:

$$\hat{y}_1 - \hat{x}_1 = \hat{y}_2 - \hat{x}_2 \tag{I.4.1}$$

Proof: The closure and convexity of the set of global maximum solutions is guaranteed by theorem 2.10.2. To prove (I.4.1), first note that since $\hat{x}_1$, $\hat{y}_1$ and $\hat{x}_2$, $\hat{y}_2$ are both global XYMAP minimizers, then $\|\hat{y}_1 - \hat{x}_1\|_R = \|\hat{y}_2 - \hat{x}_2\|_R$. Now define the interpolation point $(\bar{x}, \bar{y})$ by:

$$(\bar{x}, \bar{y}) = \lambda(\hat{x}_1, \hat{y}_1) + (1-\lambda)(\hat{x}_2, \hat{y}_2) \qquad \text{for } 0<\lambda<1 \tag{I.4.2}$$

Because the norm is a strictly convex function of its argument:

$$\|\bar{y} - \bar{x}\|_R = \left\| \lambda(\hat{y}_1 - \hat{x}_1) + (1-\lambda)(\hat{y}_2 - \hat{x}_2) \right\|_R$$

$$\leq \lambda \|\hat{y}_1-\hat{x}_1\|_R + (1-\lambda)\|\hat{y}_2-\hat{x}_2\|_R \qquad (I.4.3)$$

$$= \|\hat{y}_1-\hat{x}_1\|_R$$

but since $(\hat{x}_1,\hat{y}_1)$ is a global minimizing XYMAP solution, $\|\bar{y}-\bar{x}\|_R$ can't be smaller than $\|\hat{y}_1-\hat{x}_1\|_R$, and thus equality must hold in the relation above. However, because the norm is strictly convex, equality can hold if and only if:

$$\hat{y}_1-\hat{x}_1 = \hat{y}_2-\hat{x}_2 \qquad (I.4.4)$$

Note that, as guaranteed by theorem 2.10.2, all points on the line connecting $(\hat{x}_1,\hat{y}_1)$ and $(\hat{x}_2,\hat{y}_2)$ will be global maximizers. □

We conjecture that this result is also true for our other algorithms. We can also prove:

Theorem I.3 Let X and Y be convex, closed and non-empty. Then the XYMAP iterative estimates satisfy:

$$\|\hat{y}_{k+1}-\hat{y}_k\|_R < \|\hat{x}_{k+1}-\hat{x}_k\|_R \qquad \text{for } k=0,1,\ldots \qquad (I.4.5)$$

provided that $\hat{x}_{k+1}\neq\hat{x}_k$ (this could only happen if the iteration has already converged to a limit.) Similarly:

$$\|\hat{x}_{k+1}-\hat{x}_k\|_R < \|\hat{y}_k-\hat{y}_{k-1}\|_R \qquad \text{for } k=0,1,\ldots \qquad (I.4.6)$$

provided that $\hat{y}_k\neq\hat{y}_{k-1}$.

Proof: We will only prove the first of these relationships; the second can be proved in an identical manner by swapping the roles of x and y. By theorem D.2 in Appendix D, letting $x_1=\hat{x}_{k+1}$ and $x_2=\hat{x}_k$:

$$\|\hat{y}_{k+1}-\hat{y}_k\|_R = \left\| K_y(\hat{x}_{k+1}) - K_y(\hat{x}_k) \right\|_R \leq \|\hat{x}_{k+1}-\hat{x}_k\|_R \qquad (I.4.7)$$

Close examination of the proof of the non-expansive mapping theorem in Appendix D indicates that equality holds in the relation above only if:

$$0 = \left\| (\hat{y}_{k+1} - \hat{y}_k) - (\hat{x}_{k+1} - \hat{x}_k) \right\|_R \tag{I.4.8}$$

This will only be true if:

$$\hat{y}_{k+1} - \hat{x}_{k+1} = \hat{y}_k - \hat{x}_k \tag{I.4.9}$$

But by construction of the iterative algorithm:

$$\|\hat{y}_{k+1} - \hat{x}_{k+1}\|_R \le \|\hat{y}_k - \hat{x}_{k+1}\|_R \le \|\hat{y}_k - \hat{x}_k\|_R \tag{I.4.10}$$

Since (I.4.9) is true, however, these norms must actually be equal to each other. In particular:

$$\|\hat{y}_k - \hat{x}_{k+1}\|_R = \|\hat{y}_k - \hat{x}_k\|_R \tag{I.4.11}$$

But Theorem D.1 in Appendix D guarantees that because the set X is convex, the projection of $\hat{y}_k$ onto X, i.e. $\hat{x}_{k+1}$, is the ***unique*** element of X which minimizes $\|\hat{y}_k - x\|_R$. Therefore; (I.4.11) can only be possible if:

$$\hat{x}_k = \hat{x}_{k+1} \tag{I.4.12}$$

But then $\hat{y}_{k+1} = \hat{y}_k$, and in fact:

$$\begin{aligned} \hat{x}_{k+m} &= \hat{x}_k \\ \hat{y}_{k+m} &= \hat{y}_k \qquad \text{for all } m > 0 \end{aligned} \tag{I.4.13}$$

and thus the sequence has converged. Therefore $\|\hat{y}_{k+1} - \hat{y}_k\|_R$ must be strictly less than $\|\hat{x}_{k+1} - \hat{x}_k\|_R$ unless the sequence has already converged.

We would conjecture that this strict non-expansiveness is true for all four algorithms; to prove this, however, would require strengthening Prékopa's result in Appendix E to cover strictly concave functions.

References

1. J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York (1970).

Appendix J

Linear Equality Constraints in the Fisher XYMAP Problem

1. Eigenstructure of $P_x P_y P_x$ and $P_y P_x P_y$

The eigenstructure of the matrices $P_x P_y P_x$ and $P_y P_x P_y$ can be analyzed in exactly the same way we analyzed $P_x H P_y B P_x$ and $P_y B P_x H P_y$ for the Bayesian problem in Appendix G. In fact, if we let $B = I$ and take $\alpha \rightarrow 0$, most of the results carry over directly. In the following, therefore, we will only prove results which are unique to the Fisher problem. Unless otherwise indicated, we will assume that all spaces are finite dimensional.

Property A

The range of P_x is the null space N_x of G_x, $P_x \underline{v} \in N_x$. The range of P_y is the null space N_y of G_y, $P_y \underline{v} \in N_y$. The null space of P_x is $N_x^{\perp}$ and the null space of P_y is $N_y^{\perp}$, where these orthogonal complement sets are formed with respect to the inner product $<\cdot,\cdot>_R$.

Property B

The matrix $R^{-\frac{1}{2}} P_x P_y P_x R^{\frac{1}{2}}$ is symmetric and positive semi-definite; its eigenvalues λ_i are thus real and non-negative, and its eigenvectors $\underline{\tilde{\psi}}_i$ form a complete orthonormal set.

Property C

The matrix $P_x P_y P_x$ has the same real and non-negative eigenvalues λ_i as $R^{-\frac{1}{2}} P_x P_y P_x R^{\frac{1}{2}}$, its eigenvectors are $\underline{\psi}_i = R^{\frac{1}{2}} \underline{\tilde{\psi}}_i$, and they form a complete orthonormal set with respect to the inner product $<\cdot,\cdot>_{\mathrm{R}}$.

Property D

Since projection matrices are non-expansive mappings (see Appendix G, section 1) then

$$\|\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x \underline{v}\|_{\mathrm{R}} \leq \|\mathrm{P}_y \mathrm{P}_x \underline{v}\|_{\mathrm{R}} \leq \|\mathrm{P}_x \underline{v}\|_{\mathrm{R}} \leq \|\underline{v}\|_{\mathrm{R}} \tag{J.1.1}$$

and thus all eigenvalues of $\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x$ must be less than or equal to one.

Property E

If $\underline{v}$ is an element of the null space of G_x, $\underline{v} \in N_x$, then $\mathrm{P}_x \underline{v} = \underline{v}$. Similarly, if $\underline{v}$ is an element of the null space of G_y, $\underline{v} \in N_y$, then $\mathrm{P}_y \underline{v} = \underline{v}$. Thus if the intersection $N_x \cap N_y$ is non-trivial, then every element $\underline{v} \in N_x \cap N_y$ must be an eigenvector of $\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x$ with eigenvalue 1:

$$\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x \underline{v} = \mathrm{P}_x \mathrm{P}_y \underline{v} = \mathrm{P}_x \underline{v} = \underline{v} \qquad \text{for all } \underline{v} \in N_x \cap N_y \tag{J.1.2}$$

Any eigenvectors of $\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x$ not in $N_x \cap N_y$ must have eigenvalue strictly less than 1. Furthermore, by property C, any eigenvector not in $N_x \cap N_y$ will be orthogonal to $N_x \cap N_y$. The eigenvectors of $\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x$ thus split into two groups; those in $N_x \cap N_y$ have eigenvalue 1, and those orthogonal to $N_x \cap N_y$ have eigenvalue strictly less than 1.

Property F

All eigenvectors $\underline{\psi}_i$ of $\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x$ corresponding to non-zero eigenvalues λ_i must satisfy:

$$\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x \underline{\psi} = \lambda_i \underline{\psi}_i \neq \underline{0} \tag{J.1.3}$$

and thus $\underline{\psi}_i$ must be in the range of P_x, or in other words, $\underline{\psi}_i \in N_x$.

Property G

Exactly the same type of statements can be made about the eigenvalues λ_i and eigenvectors ϕ_i of $P_y P_x P_y$. Moreover, $P_y P_x P_y$ has exactly the same eigenvalues as $P_x P_y P_x$, and its eigenvectors ϕ_i corresponding to non-zero eigenvalues λ_i can be put into one-to-one correspondence with those of $P_x P_y P_x$:

$$\phi_i = \frac{1}{\sqrt{\lambda_i}} P_y \psi_i \tag{J.1.4}$$

$$\psi_i = \frac{1}{\sqrt{\lambda_i}} P_x \phi_i$$

These eigenvectors ϕ_i are all elements of N_y. Thus there can be at most $\min(N-p, N-q)$ non-zero eigenvalues of $P_x P_y P_x$ and $P_y P_x P_y$. All elements in the intersection $N_x \cap N_y$ are eigenvectors of $P_y P_x P_y$ with eigenvalue 1. All other eigenvectors of $P_y P_x P_y$ are orthogonal to $N_x \cap N_y$ and have eigenvalue strictly less than 1.

Property H

$\|P_x P_y\|_R = \|P_y P_x\|_R =$ the square root of the maximum eigenvalue of $P_x P_y P_x$ and $P_y P_x P_y$. To prove this, use the fact that $R^{-1}P_x = P_x^T R^{-1}$, $R^{-1}P_y = P_y^T R^{-1}$, and $P_x P_x = P_x$:

$$\begin{aligned} \|P_x P_y\|_R^2 &= \max_{\underline{y}} \frac{\underline{y}^T P_y^T P_x^T R^{-1} P_x P_y \underline{y}}{\underline{y}^T R^{-1} \underline{y}} \\ &= \max_{\underline{y}} \frac{\underline{y}^T R^{-1} P_y P_x P_y \underline{y}}{\underline{y}^T R^{-1} \underline{y}} \\ &= \max_{\underline{y}} \frac{\underline{z}^T R^{-\frac{1}{2}} P_y P_x P_y R^{\frac{1}{2}} \underline{z}}{\underline{z}^T \underline{z}} \qquad \text{where: } \underline{z} = R^{-\frac{1}{2}} \underline{y} \end{aligned} \tag{J.1.5}$$

which is the largest eigenvalue of $R^{-\frac{1}{2}} P_y P_x P_y R^{\frac{1}{2}}$, which in turn is the largest eigenvalue of $P_y P_x P_y$ and $P_x P_y P_x$. Exactly the same result can be proven for $\|P_y P_x\|_R^2$.

Property I

Eigenvectors $\underline{\psi}_i$ and $\underline{\phi}_i$ of $P_xP_yP_x$ and $P_yP_xP_y$ corresponding to non-zero eigenvalues λ_i are also eigenvectors of P_xP_y and P_yP_x respectively with the same eigenvalues. This is because $\underline{\psi}_i \in N_x$ and $\underline{\phi}_i \in N_y$, and thus $P_x\underline{\psi}_i = \underline{\psi}_i$ and $P_y\underline{\phi}_i = \underline{\phi}_i$. This implies that:

$$\begin{aligned} P_xP_y\underline{\psi}_i &= P_xP_yP_x\underline{\psi}_i = \lambda_i\underline{\psi}_i \\ P_yP_x\underline{\phi}_i &= P_yP_xP_y\underline{\phi}_i = \lambda_i\underline{\phi}_i \end{aligned} \tag{J.1.6}$$

Property J

Both P_x and P_y map $(N_x \cap N_y)^\perp$ onto $(N_x \cap N_y)^\perp$ and map $N_x \cap N_y$ onto $N_x \cap N_y$. To prove this, suppose $\underline{v} \in N_x \cap N_y$. Then $P_x\underline{v} = \underline{v} \in N_x \cap N_y$ and $P_y\underline{v} = \underline{v} \in N_x \cap N_y$. Now suppose $\underline{v} \in (N_x \cap N_y)^\perp$. Then $\underline{v}$ can be written uniquely in the form:

$$\underline{v} = \underline{v}_1 + \underline{v}_2 \qquad \text{where } \underline{v}_1 \in N_x \quad \text{and} \quad \underline{v}_2 \in N_x^\perp \tag{J.1.7}$$

Because $\underline{v}_2 \perp N_x$, it must also be true that $\underline{v}_2 \perp N_x \cap N_y$. But then $\underline{v} \perp N_x \cap N_y$ is only possible if $\underline{v}_1 \perp N_x \cap N_y$. Since $P_x\underline{v} = \underline{v}_1$, then $P_x\underline{v} \perp N_x \cap N_y$. We can prove $P_y\underline{v} \perp N_x \cap N_y$ similarly.

Property K

The null space of the matrices $(I - P_xP_yP_x)$ and $(I - P_yP_xP_y)$ is $N_x \cap N_y$, while the range of these matrices is $(N_x \cap N_y)^\perp$. To prove this, let $\underline{\psi}_i$ be any eigenvector of $P_xP_yP_x$ with eigenvalue λ_i; then it is also an eigenvector of $(I - P_xP_yP_x)$ with eigenvalue $1-\lambda_i$. The null space of $(I - P_xP_yP_x)$ is therefore spanned by the set of eigenvectors $\underline{\psi}_i$ with eigenvalue $\lambda_i = 1$, which is simply $N_x \cap N_y$. Since the eigenvectors $\underline{\psi}_i$ form a complete orthonormal basis, the range of $(I - P_xP_yP_x)$ must be:

$$\text{Range}(I - P_xP_yP_x) = \text{span}\left\{ (I - P_xP_yP_x)\underline{\psi}_i \right\} = \text{span}\left\{ (1-\lambda_i)\underline{\psi}_i \right\} \tag{J.1.8}$$

The only vectors missing from the range will be those in the subspace spanned by eigenvectors ψ_i with eigenvalue $\lambda_i = 1$, which is all the vectors in $N_x \cap N_y$. The range must therefore be $(N_x \cap N_y)^\perp$.

Property L

All the eigenvectors ψ_i in the set spanning $(N_x \cap N_y)^\perp$ have eigenvalue λ_i less than 1. If there are a finite number of these eigenvectors with non-zero eigenvalue, then there must be one with the largest such eigenvalue $\lambda_{max} < 1$. Then:

$$\begin{aligned} \|\mathrm{P}_y \mathrm{P}_x \underline{v}\|_{\mathrm{R}}^2 &\le \lambda_{max} \|\underline{v}\|_{\mathrm{R}}^2 \\ \|\mathrm{P}_x \mathrm{P}_y \underline{v}\|_{\mathrm{R}}^2 &\le \lambda_{max} \|\underline{v}\|_{\mathrm{R}}^2 \end{aligned} \qquad \text{for all } \underline{v} \in (N_x \cap N_y)^\perp \tag{J.1.9}$$

To prove this, write $\underline{v}$ as a linear combination of the eigenvectors ψ_i in $(N_x \cap N_y)^\perp$.

$$\underline{v} = \sum_{\psi_i \in (N_x \cap N_y)^\perp} v_i \psi_i \tag{J.1.10}$$

Then:

$$\mathrm{P}_x \mathrm{P}_y \mathrm{P}_x \underline{v} = \sum v_i \lambda_i \psi_i$$

and thus:

$$\begin{aligned} \|\mathrm{P}_y \mathrm{P}_x \underline{v}\|_{\mathrm{R}}^2 &= \underline{v}^{\mathrm{T}} \mathrm{P}_x^{\mathrm{T}} \mathrm{P}_y^{\mathrm{T}} \mathrm{R}^{-1} \mathrm{P}_y \mathrm{P}_x \underline{v} \\ &= \underline{v}^{\mathrm{T}} \mathrm{R}^{-1} \mathrm{P}_x \mathrm{P}_y \mathrm{P}_x \underline{v} \\ &= \left(\sum v_i \psi_i \right) \mathrm{R}^{-1} \left(\sum v_i \lambda_i \psi_i \right) \\ &= \sum v_i^2 \lambda_i \\ &\le \lambda_{max} \sum v_i^2 \\ &= \lambda_{max} \|\underline{v}\|_{\mathrm{R}}^2 \end{aligned} \tag{J.1.11}$$

We can prove

$$\|\mathrm{P}_x \mathrm{P}_y \underline{v}\|_{\mathrm{R}}^2 \le \lambda_{max} \|\underline{v}\|_{\mathrm{R}}^2 \tag{J.1.12}$$

similarly. This property thus implies that $\mathrm{P}_x \mathrm{P}_y$ and $\mathrm{P}_y \mathrm{P}_x$ are contraction mappings on the set $(N_x \cap N_y)^{\perp}$ and identities on $N_x \cap N_y$.

Property M

The matrices $Q_- = (I-P_x)$ and $Q_y = (I-P_y)$ are projection matrices which are "orthogonal" to P_x and P_y in the sense that they project vectors onto the orthogonal complements $N_x^{\perp}$ and $N_y^{\perp}$ of the null spaces N_x and N_y. The matrices $Q_x Q_y Q_x$ and $Q_y Q_x Q_y$ has a similar eigenstructure as $P_y P_x P_y$ and $P_x P_y P_x$; all their eigenvalues are real, non-negative and bounded above by one, and their eigenvectors form a complete orthonormal basis. In addition, the non-zero eigenvalues of these matrices which are strictly less than one are identical, and the corresponding eigenvectors can all be put into a one-to-one correspondence. To see this, let $\{\psi_i\}$ be a set of orthonormal eigenvectors of $P_x P_y P_x$ with eigenvalues λ_i, where $0<\lambda_i<1$. Note that $\psi_i \in N_x$, so that $P_x \psi_i = \psi_i$. Define:

$$
\begin{aligned}
\phi_i &= \frac{1}{\sqrt{\lambda_i}} P_y P_x \psi_i \\
\eta_i &= \frac{1}{\sqrt{1-\lambda_i}} Q_y P_x \psi_i \qquad\qquad \text{(J.1.13)} \\
\xi_i &= \frac{1}{\sqrt{1-\lambda_i}} Q_x P_y \phi_i
\end{aligned}
$$

Then the sets $\{\phi_i\}$, $\{\eta_i\}$, $\{\xi_i\}$ are orthonormal and are eigenvectors of $P_y P_x P_y$, $Q_y Q_x Q_y$ and $Q_x Q_y Q_x$ respectively with eigenvalue λ_i:

$$
\begin{aligned}
P_y P_x P_y \phi_i &= \lambda_i \phi_i \\
Q_y Q_x Q_y \eta_i &= \lambda_i \eta_i \qquad\qquad \text{(J.1.14)} \\
Q_x Q_y Q_x \xi_i &= \lambda_i \xi_i
\end{aligned}
$$

We prove the result for the vectors η_i; the proof for the others is similar.

$$
Q_y Q_x Q_y \eta_i = \frac{1}{\sqrt{1-\lambda_i}} (I-P_y)(I-P_x)(I-P_y)\psi_i
$$

$$= \frac{1}{\sqrt{1-\lambda_i}} (\mathrm{I}-\mathrm{P}_y)(\mathrm{I}-\mathrm{P}_x-\mathrm{P}_y+\mathrm{P}_x\mathrm{P}_y)\psi_i$$

$$= \frac{1}{\sqrt{1-\lambda_i}} (\lambda_i\mathrm{I}-\lambda_i\mathrm{P}_y-\mathrm{P}_y+\mathrm{P}_y\mathrm{P}_y)\psi_i$$

$$= \frac{1}{\sqrt{1-\lambda_i}} \lambda_i(\mathrm{I}-\mathrm{P}_y)\psi_i$$

$$= \lambda_i\eta_i$$

Also:

$$<\eta_j,\eta_i>_{\mathrm{R}} = \frac{1}{\sqrt{1-\lambda_i}\sqrt{1-\lambda_j}} \psi^{\mathrm{T}}\mathrm{P}_x^{\mathrm{T}}(\mathrm{I}-\mathrm{P}_y)^{\mathrm{T}}\mathrm{R}^{-1}(\mathrm{I}-\mathrm{P}_y)\mathrm{P}_x\psi_i$$

$$= \frac{1}{\sqrt{1-\lambda_i}\sqrt{1-\lambda_j}} \psi_j^{\mathrm{T}}\mathrm{R}^{-1}\mathrm{P}_x(\mathrm{I}-\mathrm{P}_y)\mathrm{P}_x\psi_i$$

$$= \frac{1}{\sqrt{1-\lambda_i}\sqrt{1-\lambda_j}} \left[<\psi_j,\psi_i>_{\mathrm{R}} - \lambda_i <\psi_j,\psi_i>_{\mathrm{R}}\right]$$

$$= \frac{1}{\sqrt{1-\lambda_i}\sqrt{1-\lambda_j}} \left[(1-\lambda_i)\,\delta_{ij}\right]$$

$$= \delta_{ij}$$

2. Primal Algorithm

<u>Theorem J.1</u> Suppose the constraint sets X and Y are defined by linear equations:

$$X = \left\{\mathbf{x} \;\middle|\; \mathrm{G}_x\mathbf{x} = \mathbf{y}_x\right\} \quad \text{and} \quad Y = \left\{\mathbf{y} \;\middle|\; \mathrm{G}_y\mathbf{y} = \mathbf{y}_y\right\} \tag{J.2.1}$$

where G_x and G_y have full row rank. Then the Fisher XYMAP problem:

$$\hat{\mathbf{x}}, \hat{\mathbf{y}} \leftarrow \min_{\mathbf{x}\in X,\mathbf{y}\in Y} \|\hat{\mathbf{y}} - \hat{\mathbf{x}}\|_{\mathrm{R}}^2 \tag{J.2.2}$$

has a solution given by:

$$\left(\mathrm{I} - \mathrm{P}_x\mathrm{P}_y\mathrm{P}_x\right)\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathrm{P}_x(\bar{\mathbf{y}}+\mathrm{P}_y\bar{\mathbf{x}}) \tag{J.2.3}$$

$$\hat{\mathbf{y}} = \mathrm{P}_y\hat{\mathbf{x}} + \bar{\mathbf{y}}$$

or:

$$\left(I - P_y P_x P_y\right)\hat{y} = \bar{y} + P_y(\bar{x} + P_x\bar{y}) \qquad \text{(J.2.4)}$$
$$\hat{x} = \bar{x} + P_x\bar{y}$$

The solution to (J.2.3) or (J.2.4) will be unique if the null spaces N_x and N_y of G_x and G_y respectively do not intersect, $N_x \cap N_y = \{\underline{0}\}$. Otherwise, there will be many solutions; one of these $\hat{x}_{min}$, $\hat{y}_{min}$ will have minimal energy, and the rest $(\hat{x}, \hat{y})$ can be written as:

$$\begin{aligned} \hat{x} &= \hat{x}_{min} + \underline{v} \\ \hat{y} &= \hat{y}_{min} + \underline{v} \end{aligned} \qquad \text{for some } \underline{v} \in N_x \cap N_y \qquad \text{(J.2.5)}$$

and:

$$\|\hat{x}_{min}\|_R^2 \le \|\hat{x}\|_R^2 \qquad \text{(J.2.6)}$$

$$\|\hat{y}_{min}\|_R^2 \le \|\hat{y}\|_R^2$$

Proof: The derivation of the formulas (J.2.3) and (J.2.4) was presented in chapter 6, and we will not repeat it here. Let us prove that the closed-form formula (J.2.3) for $\hat{x}$ can always be solved. Note that:

$$P_x\bar{x} = \left[I - RG_x^T\left(G_x RG_x^T\right)^{-1} G_x \right]\left[RG_x^T\left(G_x RG_x^T\right)^{-1} \underline{y}_x \right] = 0 \qquad \text{(J.2.7)}$$

Thus $\bar{x}$ is in the null space of P_x, so $\bar{x} \in N_x^{\perp}$. Similarly, $\bar{y} \in N_y^{\perp}$. Since $\bar{y} \perp N_y$ and $\bar{x} \perp N_x$, we must have $\bar{y} \perp N_x \cap N_y$ and $\bar{x} \perp N_x \cap N_y$ also, and thus by property J above:

$$\begin{array}{ll} \bar{x} & \perp\ N_x \cap N_y \\ P_x\bar{y} & \perp\ N_x \cap N_y \\ P_y\bar{x} & \perp\ N_x \cap N_y \\ P_x P_y\bar{x} & \perp\ N_x \cap N_y \end{array} \qquad \text{(J.2.8)}$$

The right hand side of (J.2.3) is thus an element of $(N_x \cap N_y)^{\perp}$. But property K above says that the range of $(I-P_xP_yP_x)$ is exactly $(N_x \cap N_y)^{\perp}$. Thus we can always find a solution $\hat{x}_0$. Since the null space of $(I-P_xP_yP_x)$ is $N_x \cap N_y$, every solution must have the form:

$$\hat{x} = \hat{x}_0 + v \qquad \text{where } v \in N_x \cap N_y \tag{J.2.9}$$

The corresponding output estimate will be:

$$\begin{aligned} \hat{y} &= P_y\hat{x} + \bar{y} \\ &= \hat{y}_0 + v \\ &\quad \text{where: } \hat{y}_0 = P_y\hat{x}_0 + \bar{y} \end{aligned} \tag{J.2.10}$$

The set of solutions forms a linear variety; of these, there is a unique estimate $(\hat{x}_{min}, \hat{y}_{min})$ such that $\hat{x}_{min} \perp N_x \cap N_y$. But then $\hat{y}_{min} = P_y\hat{x}_{min} + \bar{y} \perp N_x \cap N_y$ also. All the other solutions $(\hat{x}, \hat{y})$ can be written:

$$\begin{aligned} \hat{x} &= \hat{x}_{min} + v \\ \hat{y} &= \hat{y}_{min} + v \end{aligned} \qquad \text{for some } v \in N_x \cap N_y \tag{J.2.11}$$

Note that:

$$\begin{aligned} \|\hat{x}\|_R^2 &= \|\hat{x}_{min}\|_R^2 + \|v\|_R^2 \geq \|\hat{x}_{min}\|_R^2 \\ \|\hat{y}\|_R^2 &= \|\hat{y}_{min}\|_R^2 + \|v\|_R^2 \geq \|\hat{y}_{min}\|_R^2 \end{aligned} \tag{J.2.12}$$

so that all these other solutions have more energy than $(\hat{x}_{min}, \hat{y}_{min})$. Proving the result for the other version (J.2.4) of the closed-form formula is straightforward. □

Theorem J.2 Suppose that the constraint sets X, Y are defined by linear equalities as in (J.2.1). Suppose also that the matrices $P_xP_yP_x$ and $P_yP_xP_y$ have a finite number of non-zero eigenvalues (This will be true if $\min(N-p, N-q) < \infty$). Then the iterative projection algorithm:

$$\begin{aligned} \hat{x}_{k+1} &= P_x\hat{y}_k + \bar{x} \\ \hat{y}_{k+1} &= P_y\hat{x}_{k+1} + \bar{y} \end{aligned} \tag{J.2.13}$$

is guaranteed to converge at a geometric rate to a solution $(\underline{\hat{x}},\underline{\hat{y}})$ to the Fisher XYMAP problem (J.2.2):

$$\| \underline{\hat{y}}_{k+1} - \underline{\hat{y}}_k \|_{\mathrm{R}}^2 \le \lambda_{max} \| \underline{\hat{x}}_{k+1} - \underline{\hat{x}}_k \|_{\mathrm{R}}^2 \le \lambda_{max}^2 \| \underline{\hat{y}}_k - \underline{\hat{x}}_k \|_{\mathrm{R}}^2 \tag{J.2.14}$$

If we decompose the initial estimate $\underline{\hat{y}}_0$ into a component $\underline{y}_0$ in $N_x \cap N_y$ and a component $\underline{\bar{y}}_0$ orthogonal to $N_x \cap N_y$

$$\underline{\hat{y}}_0 = \underline{\bar{y}}_0 + \underline{y}_0 \qquad \text{where } \underline{\hat{y}}_0 \perp N_x \cap N_y \tag{J.2.15}$$
$$\underline{y}_0 \in N_x \cap N_y$$

then the iterative algorithm converges to the "nearest" solution:

$$\begin{aligned} \underline{\hat{x}} &= \underline{\hat{x}}_{min} + \underline{y}_0 \\ \underline{\hat{y}} &= \underline{\hat{y}}_{min} + \underline{y}_0 \end{aligned} \tag{J.2.16}$$

Thus if we choose $\underline{\hat{y}}_0 \perp N_x \cap N_y$, the convergent solution will be $(\underline{\hat{x}}_{min},\underline{\hat{y}}_{min})$.

Proof: Let $\underline{\hat{y}}$ be the global minimum solution $\underline{\hat{y}} = \underline{\hat{y}}_{min} + \underline{y}_0$. Then since both $\underline{\bar{y}}_0$ and $\underline{\hat{y}}_{min}$ are orthogonal to $N_x \cap N_y$:

$$\underline{\hat{y}}_0 - \underline{\hat{y}} = \underline{\bar{y}}_0 - \underline{\hat{y}}_{min} \perp N_x \cap N_y \tag{J.2.17}$$

Now recognizing that $\underline{\hat{x}}_{k+1} = \mathrm{P}_x \underline{\hat{x}}_{k+1} + \underline{\bar{x}}$,

$$\underline{\hat{y}}_{k+1} = \mathrm{P}_y \mathrm{P}_x \underline{\hat{x}}_{k+1} + (\underline{\bar{y}} + \mathrm{P}_y \underline{\bar{x}}) \tag{J.2.18}$$

Also, the global minimum $(\underline{\hat{x}},\underline{\hat{y}})$ must be a stationary point of the algorithm, with $\underline{\hat{x}} = \mathrm{P}_x \underline{\hat{x}} + \underline{\bar{x}}$, and thus:

$$\underline{\hat{y}} = \mathrm{P}_y \mathrm{P}_x \underline{\hat{x}} + (\underline{\bar{y}} + \mathrm{P}_y \underline{\bar{x}}) \tag{J.2.19}$$

Subtracting (J.2.19) from (J.2.18) yields:

$$\underline{\hat{y}}_{k+1} - \underline{\hat{y}} = \mathrm{P}_y \mathrm{P}_x (\underline{\hat{x}}_{k+1} - \underline{\hat{x}}) \tag{J.2.20}$$

Similarly, we can show that:

$$\underline{\hat{x}}_{k+1} - \underline{\hat{x}} = \mathrm{P}_x \mathrm{P}_y (\underline{\hat{y}}_k - \underline{\hat{y}}) \tag{J.2.21}$$

Applying property J recursively, it is easy to show that:

$$\begin{array}{ll} \hat{y}_k - \hat{y} \perp N_x \cap N_y & \\ \hat{x}_k - \hat{x} \perp N_x \cap N_y & \text{for all } k \end{array} \qquad \text{(J.2.22)}$$

Thus by property L above:

$$\begin{aligned} \|\hat{y}_{k+1} - \hat{y}\|_R^2 &= \|P_y P_x (\hat{x}_{k+1} - \hat{x})\|_R^2 \\ &\le \lambda_{max} \|\hat{x}_{k+1} - \hat{x}\|_R^2 \\ &= \lambda_{max} \|P_x P_y (\hat{y}_k - \hat{y})\|_R^2 \\ &\le \lambda_{max}^2 \|\hat{y}_k - \hat{y}\|_R^2 \end{aligned} \qquad \text{(J.2.23)}$$

Thus as $k \to \infty$, $\|\hat{y}_k - \hat{y}\|_R^2 \to 0$, and $\|\hat{x}_k - \hat{x}\|_R^2 \to 0$, and thus $(\hat{x}_k, \hat{y}_k)$ converges geometrically to the global minimum solution $(\hat{x}, \hat{y})$. If we choose $\hat{y}_0$ to be orthogonal to $N_x \cap N_y$, then $y_0 = 0$ and the convergent solution will be $(\hat{x}_{min}, \hat{y}_{min})$, which is the minimum norm solution to the Fisher problem (J.2.2). Finally, note that since $\bar{y}$ is orthogonal to N_y, it is the vector in Y with the smallest norm. Thus if we choose $\hat{y}_0$ to be the element in Y with the minimum norm, we are guaranteed that it will be orthogonal to $N_x \cap N_y$, and thus the convergent solution will be $(\hat{x}_{min}, \hat{y}_{min})$.

Primal Algorithm - Noise Sensitivity

We can analyze the noise sensitivity of this algorithm in exactly the same way as in chapter 5. Combining the derivation in Appendix G with the decomposition of the errors into components Δ_k in $N_x \cap N_y$ and component $\Delta_k^{\perp}$ orthogonal to $N_x \cap N_y$, we can show that the error between the computed estimate $\tilde{y}_k$ and the "correct" estimate $\hat{y}_k$ is:

$$\tilde{y}_k - \hat{y}_k = \sum_{m=0}^{k-1} (P_y P_x)^m (\Delta_{k-m} + \Delta_{k-m}^{\perp}) \qquad \text{(J.2.24)}$$

But since $\Delta_k \in N_x \cap N_y$, then $P_x \Delta_k = P_y \Delta_k = \Delta_k$. Thus:

$$\tilde{y}_k - \hat{y}_k = \sum_{m=0}^{k-1} \Delta_{k-m} + \sum_{m=0}^{k-1} (P_y P_x)^m \Delta_{k-m}^{\perp}$$

$$= \sum_{m=0}^{k-1} \Delta_{k-m} + \left[\Delta_k^{\perp} + \sum_{m=1}^{k-1} \left[(P_y P_x)(P_x P_y) \right]^{m-1} (P_y P_x) \Delta_{k-m}^{\perp} \right] \qquad (J.2.25)$$

where we have liberally used $P_x P_x = P_x$ and $P_y P_y = P_y$ in the last line. But by property J, since $\Delta_{k-m}^{\perp} \in (N_x \cap N_y)^{\perp}$, then $P_y P_x \Delta_{k-m}^{\perp} \in (N_x \cap N_y)^{\perp}$, and in fact every term in the last summation in (J.2.25) is an element of $(N_x \cap N_y)^{\perp}$. Thus by property L:

$$\| \tilde{y}_k - \hat{y}_k \|_R^2 \leq \sum_{m=0}^{k-1} \| \Delta_k \|_R^2 + \| \Delta_k^{\perp} \|_R^2 + \sum_{m=1}^{k-1} \lambda_{max}^{2m-1} \| \Delta_{k-m}^{\perp} \|_R^2 \qquad (J.2.26)$$

If all the errors $\| \Delta_k \|_R^2$ are about equal to $\bar{\Delta}$, and all the errors $\| \Delta_{k-m} \|_R^2$ are about equal to $\bar{\Delta}^{\perp}$, then:

$$\| \tilde{y}_k - \hat{y}_k \|_R^2 \leq k \bar{\Delta} + \left(1 + \frac{\lambda_{max}(1 - \lambda_{max}^{2k-2})}{1 - \lambda_{max}^2} \right) \bar{\Delta}^{\perp} \qquad (J.2.27)$$

Primal Algorithm - Interpretation of $\| P_y P_x \|_R$ as a Cosine

Finally, we present Youla's proof [1] that:

Theorem J.3

$$\cos \theta(X, Y) = \| P_y P_x \|_R = \| P_x P_y \|_R$$

where $\cos \theta(X, Y)$ is the angle between the constraint sets X and Y defined in (J.2.1).

Proof: Assume that $R(P_x) \neq \{\emptyset\}$ and $R(P_y) \neq \{\emptyset\}$; otherwise, the result is trivial since P_x or P_y would be the zero matrix. For any $y \in R(P_y)$ we have

$$\begin{aligned} \sup_{x \in R(P_x)} \frac{|\langle x, y \rangle_R|}{\| x \|_R \| y \|_R} &= \sup_{x \in R(P_x)} \frac{|\langle P_x x, y \rangle_R|}{\| x \|_R \| y \|_R} \\ &= \sup_{x \in R(P_y)} \frac{|\langle x, P_x y \rangle_R|}{\| x \|_R \| y \|_R} \qquad (J.2.28) \\ &\leq \frac{\| P_x y \|_R}{\| y \|_R} \end{aligned}$$

where the last line is due to Schwartz' inequality, $|\langle x, P_x y \rangle_R| \leq \| x \|_R \| P_x y \|_R$. Equality will be achieved only if we choose x linearly proportional to $P_x y$. Since

$\underline{y} \in R(P_y)$, we have $\underline{y} = P_y \underline{y}$ and thus:

$$\cos\theta(X,Y) = \sup_{\underline{y} \in R(P_y)} \frac{\|P_x P_y \underline{y}\|_R}{\|\underline{y}\|_R} \tag{J.2.29}$$

To show that this is simply the norm $\|P_x P_y\|_R$, note first that from the definition of $\|P_x P_y\|_R$ that the supremum on the right hand side of (J.2.29) cannot exceed $\|P_x P_y\|_R$. Next, note that we can always decompose a vector $\underline{y}$ into $\underline{y} = \underline{y} + \underline{y}^{\perp}$ with $\underline{y} \in R(P_y)$, $\underline{y}^{\perp} \in R(P_y)^{\perp} = N(P_y)$. Then:

$$\begin{aligned} \frac{\|P_x P_y \underline{y}\|_R}{\|\underline{y}\|_R} &= \frac{\|P_x P_y \underline{y}\|_R}{\|\underline{y} + \underline{y}^{\perp}\|_R} \\ &\leq \frac{\|P_x P_y \underline{y}\|_R}{\left(\|\underline{y}\|_R^2 + \|\underline{y}^{\perp}\|_R^2\right)^{\frac{1}{2}}} \\ &\leq \frac{\|P_x P_y\|_R}{\|\underline{y}\|_R} \end{aligned} \tag{J.2.30}$$

Thus:

$$\sup_{\underline{y} \in R(P_y)} \frac{\|P_x P_y \underline{y}\|_R}{\|\underline{y}\|_R} \leq \|P_x P_y\|_R \leq \sup_{\underline{y} \in R(P_y)} \frac{\|P_x P_y \underline{y}\|_R}{\|\underline{y}\|_R} \tag{J.2.31}$$

and the lemma is proven. □

3. Dual Problem

Because Q_x and Q_y are projection matrices onto the spaces $N_x^{\perp}$ and $N_y^{\perp}$ with respect to the inner product $<\cdot,\cdot>_{\hat{Q}}$, exactly the same conclusions can be drawn about the eigenstructure of $Q_x Q_y Q_x$ and $Q_y Q_x Q_y$ as about $P_x P_y P_x$ and $P_y P_x P_y$ in section 1 of this Appendix. We need only replace P_x, P_y, N_x, N_y and R everywhere in section 1 with Q_x, Q_y, $N_x^{\perp}$, $N_y^{\perp}$ and $\hat{Q}$, and the same properties will continue to hold.

We now prove that the dual closed-form solution exists and that the dual iterative algorithm converges at a geometric rate. First we factor the offsets $\bar{\underline{\mu}}_x$ and $\bar{\underline{\mu}}_y$ into the form:

$$\bar{\rho}_x = \bar{\rho}_x^0 + \bar{\rho}_x^\perp \quad \text{where } \bar{\rho}_x^0 \in N_x^\perp \cap N_y^\perp \quad \text{and} \quad \bar{\rho}_x^\perp \perp N_x^\perp \cap N_y^\perp \tag{J.3.1}$$

$$\bar{\rho}_y = \bar{\rho}_y^0 + \bar{\rho}_y^\perp \quad \text{where } \bar{\rho}_y^0 \in N_x^\perp \cap N_y^\perp \quad \text{and} \quad \bar{\rho}_y^\perp \perp N_x^\perp \cap N_y^\perp$$

Now we prove:

Theorem J.4 Let $\hat{\rho}_x$ be any solution to:

$$\left(I - Q_x Q_y Q_x \right) \hat{\rho}_x = \bar{\rho}_x^\perp - Q_x \bar{\rho}_y^\perp \tag{J.3.2}$$

Let $\hat{\rho}_y$ be the corresponding output multiplier estimate:

$$\hat{\rho}_y = -Q_y \hat{\rho}_x + \bar{\rho}_y^\perp \tag{J.3.3}$$

then construct signal and output estimates by:

$$\hat{x}_{min} = \hat{\rho}_x + \hat{\rho}_y + \bar{\rho}_x^0 \tag{J.3.4}$$

$$\hat{y}_{min} = \hat{\rho}_x + \hat{\rho}_y + \bar{\rho}_y^0$$

Then $(\hat{x}_{min}, \hat{y}_{min})$ not only solves:

$$\hat{x}_{min}, \hat{y}_{min} - \min_{x \in X, y \in Y} \| y - x \|_Q^2 \tag{J.3.5}$$

but is also the solution with minimal energy; thus if $\hat{x}$, $\hat{y}$ are any other solution to this Fisher problem:

$$\|\hat{x}\|_Q^2 \geq \|\hat{x}_{min}\|_Q^2 \tag{J.3.6}$$

$$\|\hat{y}\|_Q^2 \geq \|\hat{y}_{min}\|_Q^2$$

Proof: We first need to show that a solution for $\hat{\rho}_x$ in (J.3.2) exists. By property K of section 1, the range of $(I - Q_x Q_y Q_x)$ is $(N_x^\perp \cap N_y^\perp)^\perp$. Now $\bar{\rho}_x^\perp \in (N_x^\perp \cap N_y^\perp)^\perp$. Also $\bar{\rho}_y^\perp \in (N_x^\perp \cap N_y^\perp)^\perp$. By property J of section 1, $Q_x \bar{\rho}_y^\perp \in (N_x^\perp \cap N_y^\perp)^\perp$ also. Thus the right hand side of (J.3.2) is an element of $(N_x^\perp \cap N_y^\perp)^\perp$, the range of $(I - Q_x Q_y Q_x)$, and thus at least one solution $\hat{\rho}_x$ exists. Since the null space of $(I - Q_x Q_y Q_x)$ is $N_x^\perp \cap N_y^\perp$, any other solution $\hat{\rho}_x'$ will have the form:

$$\hat{\rho}_x' = \hat{\rho}_x + \underline{v} \qquad \text{where } \underline{v} \in N_x^{\perp} \cap N_y^{\perp} \tag{J.3.7}$$

Note that $\bar{\rho}_x \in N_x^{\perp}$ which implies that both components $\bar{\rho}_x^0$ and $\bar{\rho}_x^{\perp}$ are elements of $N_x^{\perp}$. Furthermore, since the range of Q_x is $N_x^{\perp}$, then $Q_x\bar{\rho}_y^{\perp} \in N_x^{\perp}$. Thus the right hand side of equation (J.3.2) also is an element of $N_x^{\perp}$. To show that the solution $\hat{\rho}_x \in N_x^{\perp}$ also, note that if $\underline{v} \in N_x$, then:

$$(I - Q_xQ_yQ_x)\underline{v} = \underline{v} \in N_x \tag{J.3.8}$$

and if $\underline{v} \in N_x^{\perp}$ then:

$$(I - Q_xQ_yQ_x)\underline{v} = \underline{v} - Q_x(Q_yQ_x\underline{v}) \in N_x^{\perp} \tag{J.3.9}$$

$(I - Q_xQ_yQ_x)$ thus maps N_x into N_x and $N_x^{\perp}$ into $N_x^{\perp}$; since the right hand side of (J.3.2) is an element of $N_x^{\perp}$, the solution $\hat{\rho}_x$ must be in $N_x^{\perp}$. The corresponding output multiplier estimate $\hat{\rho}_y$ will be an element of $N_y^{\perp}$ since $\bar{y}^{\perp} \in N_y^{\perp}$ and the range of Q_y is $N_y^{\perp}$.

Now since $\hat{\rho}_x \in N_x^{\perp}$, then $Q_x\hat{\rho}_x = \hat{\rho}_x$. Also since $\hat{\rho}_y \in N_y^{\perp}$ then $Q_y\hat{\rho}_y = \hat{\rho}_y$. Thus by rearranging equations (J.3.2) we get:

$$\begin{aligned} \hat{\rho}_x &= Q_xQ_yQ_x\hat{\rho}_x + \bar{\rho}_x^{\perp} - Q_x\bar{\rho}_y^{\perp} \\ &= Q_x(Q_y\hat{\rho}_x - \bar{\rho}_y^{\perp}) + \bar{\rho}_x^{\perp} \\ &= -Q_x\hat{\rho}_y + \bar{\rho}_x^{\perp} \end{aligned} \tag{J.3.10}$$

The solution $\hat{\rho}_x$, $\hat{\rho}_y$ thus satisfies:

$$\begin{aligned} \hat{\rho}_x &= -Q_x\hat{\rho}_y + \bar{\rho}_x^{\perp} \\ \hat{\rho}_y &= -Q_y\hat{\rho}_x + \bar{\rho}_y^{\perp} \end{aligned} \tag{J.3.11}$$

All other solutions $(\hat{\rho}_x', \hat{\rho}_y')$ will have the form:

$$\begin{aligned} \hat{\rho}_x' &= \hat{\rho}_x + \underline{v} \\ \hat{\rho}_y' &= -Q_y\hat{\rho}_x' + \bar{\rho}_y^{\perp} = \hat{\rho}_y - \underline{v} \end{aligned} \qquad \text{where } \underline{v} \in N_x^{\perp} \cap N_y^{\perp} \tag{J.3.12}$$

Now let us analyze the solution $\hat{x}_{min}$, $\hat{y}_{min}$:

$$
\begin{aligned}
G_x \hat{\underline{x}}_{min} &= G_x\,(\hat{\underline{\rho}}_x + \hat{\underline{\rho}}_y + \bar{\underline{\rho}}_x^0) \\
&= -G_x Q_x \hat{\underline{\rho}}_y + G_x \bar{\underline{\rho}}_x^{\perp} + G_x \hat{\underline{\rho}}_y + G_x \bar{\underline{\rho}}_x^0 \\
&= G_x\,(\bar{\underline{\rho}}_x^{\perp} + \bar{\underline{\rho}}_x^0) \\
&= G_x \bar{\underline{\rho}}_x \\
&= \underline{y}_x
\end{aligned}
\tag{J.3.13}
$$

Similarly, we can show that $G_y \hat{\underline{y}}_{min} = \underline{y}_y$. Thus $\hat{\underline{x}}_{min} \in X$ and $\hat{\underline{y}}_{min} \in Y$. Note that starting from any of the other solutions $(\hat{\underline{\rho}}_x', \hat{\underline{\rho}}_y')$ in (J.3.12) would give exactly the same estimate of $\hat{\underline{x}}_{min}, \hat{\underline{y}}_{min}$. Moreover:

$$
\hat{\underline{y}}_{min} - \hat{\underline{x}}_{min} = \bar{\underline{\rho}}_y^0 - \bar{\underline{\rho}}_x^0 \in N_x^{\perp} \cap N_y^{\perp}
\tag{J.3.14}
$$

Let $\underline{x}$, $\underline{y}$ be any other elements of X and Y. These can always be written in the form:

$$
\begin{aligned}
\underline{x} &= \hat{\underline{x}}_{min} + \underline{v}_x \qquad \text{where } \underline{v}_x \in N_x \\
\underline{y} &= \hat{\underline{y}}_{min} + \underline{v}_y \qquad \text{where } \underline{v}_y \in N_y
\end{aligned}
\tag{J.3.15}
$$

Note that since $\underline{v}_x \perp N_x^{\perp}$ and $\underline{v}_y \perp N_y^{\perp}$, then also $\underline{v}_x \perp N_x^{\perp} \cap N_y^{\perp}$ and $\underline{v}_y \perp N_x^{\perp} \cap N_y^{\perp}$. Now:

$$
\| \underline{y} - \underline{x} \|_Q^2 = \| \hat{\underline{y}}_{min} - \hat{\underline{x}}_{min} + \underline{v}_y - \underline{v}_x \|_Q^2
\tag{J.3.16}
$$

But $\hat{\underline{y}}_{min} - \hat{\underline{x}}_{min} \in N_x^{\perp} \cap N_y^{\perp}$ while $\underline{v}_y - \underline{v}_x \perp N_x^{\perp} \cap N_y^{\perp}$; thus $\hat{\underline{y}}_{min} - \hat{\underline{x}}_{min} \perp \underline{v}_y - \underline{v}_x$ and:

$$
\| \underline{y} - \underline{x} \|_Q^2 = \| \hat{\underline{y}}_{min} - \hat{\underline{x}}_{min} \|_Q^2 + \| \underline{v}_y - \underline{v}_x \|_Q^2 \geq \| \hat{\underline{y}}_{min} - \hat{\underline{x}}_{min} \|_Q^2
\tag{J.3.17}
$$

Thus $\hat{\underline{x}}_{min}, \hat{\underline{y}}_{min}$ are indeed global minimizing solutions to the Fisher XYMAP problem. Any other global minimizing solution $(\hat{\underline{x}}', \hat{\underline{y}}')$ must have the form $\underline{v}_x = \underline{v}_y$, or:

$$
\begin{aligned}
\hat{\underline{x}}' &= \hat{\underline{x}}_{min} + \underline{v} \\
\hat{\underline{y}}' &= \hat{\underline{y}}_{min} + \underline{v}
\end{aligned}
\qquad \text{where } \underline{v} \in N_x \cap N_y
\tag{J.3.18}
$$

However:

$$
\begin{aligned}
\hat{\underline{\rho}}_x \in N_x^{\perp} \quad &\Rightarrow \quad \hat{\underline{\rho}}_x \perp N_x \cap N_y \\
\hat{\underline{\rho}}_y \in N_y^{\perp} \quad &\Rightarrow \quad \hat{\underline{\rho}}_y \perp N_x \cap N_y \\
\bar{\underline{\rho}}_x^0 \in N_x^{\perp} \cap N_y^{\perp} \quad &\Rightarrow \quad \bar{\underline{\rho}}_x^0 \perp N_x \cap N_y
\end{aligned}
\tag{J.3.19}
$$

Thus:

$$\hat{x}_{min} = \hat{\rho}_x + \hat{\rho}_y + \bar{\rho}_x^0 \perp N_x \cap N_y \tag{J.3.20}$$

Similarly, we can show:

$$\hat{y}_{min} \perp N_x \cap N_y \tag{J.3.21}$$

These relationships (J.3.20) and (J.3.21), however, guarantee that $\hat{x}_{min}$, $\hat{y}_{min}$ are the minimal energy global minimum solution, since any other solution $(\hat{x}, \hat{y})$ in (J.3.18) will satisfy:

$$\begin{aligned} \|\hat{x}\|_Q^2 &= \|\hat{x}_{min}\|_Q^2 + \|x\|_Q^2 \geq \|\hat{x}\|_Q^2 \\ \|\hat{y}\|_Q^2 &= \|\hat{y}_{min}\|_Q^2 + \|y\|_Q^2 \geq \|\hat{y}\|_Q^2 \qquad \square \end{aligned} \tag{J.3.22}$$

This closed-form solution is unfortunately not computationally practical because it requires splitting $\bar{\rho}_x$ and $\bar{\rho}_y$ into components. A more convenient solution is given by the following:

<u>Theorem J.5</u> Solve:

$$\left(I - Q_x Q_y Q_x \right) \rho_x = \bar{\rho}_x - Q_x \bar{\rho}_y \tag{J.3.23}$$

for a best least squares solution $\hat{\rho}_{x_0}$ with respect to the inner product $<\cdot,\cdot>_Q$. Then compute:

$$\begin{aligned} \hat{\rho}_{y_1} &= -Q_y \hat{\rho}_{x_0} + \bar{\rho}_y \\ \hat{\rho}_{x_1} &= -Q_x \hat{\rho}_{y_1} + \bar{\rho}_x \end{aligned} \tag{J.3.24}$$

The minimal energy global minimizing solution is then given by:

$$\begin{aligned} \hat{x}_{min} &= \hat{\rho}_{x_1} + \hat{\rho}_{y_1} \\ \hat{y}_{min} &= \hat{\rho}_{x_0} + \hat{\rho}_{y_1} \end{aligned} \tag{J.3.25}$$

Moreover, equation (J.3.23) will have an exact solution for $\hat{\rho}_x$ if and only if

$$\hat{x}_{min} = \hat{y}_{min} \tag{J.3.26}$$

<u>Proof:</u> A best least squares solution for $\hat{\rho}_x$ is found by:

$$\hat{\rho}_x - \min_{\rho_x} \left\| (I-Q_xQ_yQ_x)\rho_x - (\bar{\rho}_x - Q_x\bar{\rho}_y) \right\|^2_{\hat{Q}} \tag{J.3.27}$$

$$- \min_{\rho_x} \left\| \left\{ (I-Q_xQ_yQ_x)\rho_x - (\bar{\rho}_x^{\perp} - Q_x\bar{\rho}_y^{\perp}) \right\} - \left\{ \bar{\rho}_x^0 - Q_x\bar{\rho}_y^0 \right\} \right\|^2_{\hat{Q}}$$

The first term inside the norm is an element of $(N_x^{\perp} \cap N_y^{\perp})^{\perp}$ while the second term is in $N_x^{\perp} \cap N_y^{\perp}$. Thus:

$$\hat{\rho}_x - \min_{\rho_x} \left\| (I-Q_xQ_yQ_x)\rho_x - (\bar{\rho}_x^{\perp} - Q_x\bar{\rho}_y^{\perp}) \right\|^2_{\hat{Q}} + \left\| \bar{\rho}_x^0 - Q_x\bar{\rho}_y^0 \right\|^2_{\hat{Q}} \tag{J.3.28}$$

The second term is independent of $\hat{\rho}_{x_0}$. Thus the least squares solution ρ_x will be exactly the solution $\hat{\rho}_x$ found by our previous theorem (J.3.2). Now:

$$\begin{aligned} \hat{\rho}_{y_1} &= -Q_y\hat{\rho}_{x_0} + \bar{\rho}_y \\ &= \hat{\rho}_y + \bar{\rho}_y^0 \end{aligned} \tag{J.3.29}$$

where $\hat{\rho}_y$ is the solution in our previous theorem and:

$$\begin{aligned} \hat{\rho}_{x_1} &= -Q_x\hat{\rho}_{y_1} + \bar{\rho}_x \\ &= \hat{\rho}_x + \bar{\rho}_x^0 - \bar{\rho}_y^0 \end{aligned} \tag{J.3.30}$$

Thus:

$$\begin{aligned} \hat{x}_{min} &= \hat{\rho}_{x_1} + \hat{\rho}_{y_1} = \hat{\rho}_x + \hat{\rho}_y + \bar{\rho}_x^0 \\ \hat{y}_{min} &= \hat{\rho}_{x_0} + \hat{\rho}_{y_1} = \hat{\rho}_x + \hat{\rho}_y + \bar{\rho}_y^0 \end{aligned} \tag{J.3.31}$$

and thus $\hat{x}_{min}, \hat{y}_{min}$ are the same solutions calculated in the last theorem (J.3.4).

Finally, we note that because $\bar{\rho}_y^0 \in N_x^{\perp} \cap N_y^{\perp}$, then $Q_y\bar{\rho}_y^0 = \bar{\rho}_y^0$. Thus:

$$\hat{y}_{min} - \hat{x}_{min} = \bar{\rho}_y^0 - \bar{\rho}_x^0 = -(\bar{\rho}_x^0 - Q_y\bar{\rho}_y^0) \tag{J.3.32}$$

Thus if $\hat{x}_{min} = \hat{y}_{min}$, then the second term in (J.3.28) is zero, and the solution $\hat{\rho}_x$ actually solves the equation (J.3.23) exactly. Finally, note that we could prove a similar result starting with the formula for $\hat{\rho}_y$ instead of $\hat{\rho}_x$; the corresponding procedure would look the same as above except with the roles of the signal and output reversed everywhere. □

Next we prove that the iterative algorithm converges. Let us split our initial estimate $\hat{\underline{\rho}}_{y_0}$ into:

$$\hat{\underline{\rho}}_{y_0} = \hat{\underline{\rho}}_{y_0}^{0} + \hat{\underline{\rho}}_{y_0}^{\perp} \qquad \text{where } \hat{\underline{\rho}}_{y_0}^{0} \in N_x^{\perp} \cap N_y^{\perp} \quad \text{and} \quad \hat{\underline{\rho}}_{y_0}^{\perp} \in (N_x^{\perp} \cap N_y^{\perp})^{\perp} \tag{J.3.33}$$

Also split the eigenvectors of $Q_x Q_y Q_x$ into a group spanning $N_x^{\perp} \cap N_y^{\perp}$, each having eigenvalue of 1, and a group spanning $(N_x^{\perp} \cap N_y^{\perp})^{\perp}$, each having eigenvalue strictly less than 1. If there are a finite number of such non-zero eigenvalues which are less than 1, then one of them must have the largest value $\lambda_{max} < 1$. Then by property L:

$$\begin{aligned} \|Q_x Q_y \underline{v}\|_Q^2 &\le \lambda_{max} \|\underline{v}\|_Q^2 \\ \|Q_y Q_x \underline{v}\|_Q^2 &\le \lambda_{max} \|\underline{v}\|_Q^2 \end{aligned} \qquad \text{for all } \underline{v} \in (N_x \cap N_y)^{\perp} \tag{J.3.34}$$

Let $\hat{\underline{\rho}}_{x_{min}}$, $\hat{\underline{\rho}}_{y_{min}}$ be the solution to the closed-form problem (J.3.23) with minimum energy. We then prove:

<u>Theorem J.6</u> The multiplier estimates $\hat{\underline{\rho}}_{x_k}$, $\hat{\underline{\rho}}_{y_k}$ "converge" at a geometric rate to a linearly ramping estimate in the sense that:

$$\begin{aligned} \hat{\underline{\rho}}_{y_k} &= \left[\hat{\underline{\rho}}_{y_{min}} + \hat{\underline{\rho}}_{y_0}^{0} + k\,(\hat{\underline{y}}_{min} - \hat{\underline{x}}_{min}) \right] + \Delta_{y_k} \\ \hat{\underline{\rho}}_{x_k} &= \left[\hat{\underline{\rho}}_{x_{min}} - \hat{\underline{\rho}}_{y_0}^{0} + \bar{\underline{\rho}}_{y}^{0} - k\,(\hat{\underline{y}}_{min} - \hat{\underline{x}}_{min}) \right] + \Delta_{x_k} \end{aligned} \tag{J.3.35}$$

where Δ_{x_k} and Δ_{y_k} decay at a geometric rate:

$$\|\Delta_{y_{k+1}}\|_Q^2 \le \lambda_{max} \|\Delta_{x_{k+1}}\|_Q^2 \le \lambda_{max}^2 \|\Delta_{y_k}\|_Q^2 \tag{J.3.36}$$

The corresponding signal and output estimates $\hat{\underline{x}}_k$, $\hat{\underline{y}}_k$ converge at the rate λ_{max} to the minimum energy global minimizing solution $\hat{\underline{x}}_{min}$, $\hat{\underline{y}}_{min}$ in the sense that:

$$\|(\hat{\underline{y}}_{k+1} - \hat{\underline{x}}_{k+1}) - (\hat{\underline{y}}_{min} - \hat{\underline{x}}_{min})\|_Q^2 \tag{J.3.37}$$

$$\leq \lambda_{max} \, \| \, (\hat{y}_k - \hat{x}_{k+1}) - (\hat{y}_{min} - \hat{x}_{min}) \, \|^2_{\hat{Q}}$$
$$\leq \lambda^2_{max} \, \| \, (\hat{y}_k - \hat{x}_k) - (\hat{y}_{min} - \hat{x}_{min}) \, \|^2_{\hat{Q}}$$

Proof: Start with the relationships:

$$\begin{aligned} \hat{\rho}_{x_{k+1}} &= -Q_x \hat{\rho}_{y_k} + \bar{\rho}_x \\ \hat{\rho}_{y_{k+1}} &= -Q_y \hat{\rho}_{x_{k+1}} + \bar{\rho}_y \end{aligned} \tag{J.3.38}$$

and:

$$\begin{aligned} \hat{\rho}_{x_{min}} &= -Q_x \hat{\rho}_{y_{min}} + \bar{\rho}_x^{\perp} \\ \hat{\rho}_{y_{min}} &= -Q_y \hat{\rho}_{x_{min}} + \bar{\rho}_y^{\perp} \end{aligned} \tag{J.3.39}$$

Also from (J.3.32):

$$\hat{y}_{min} - \hat{x}_{min} = \bar{\rho}_y^0 - \bar{\rho}_x^0 \tag{J.3.40}$$

Then:

$$\begin{aligned} \Delta_{y_{k+1}} &= \hat{\rho}_{y_{k+1}} - \Big(\hat{\rho}_{y_{min}} + \hat{\rho}^0_{y_0} + (k+1)(\hat{y}_{min} - \hat{x}_{min}) \Big) \\ &= \Big(-Q_y \hat{\rho}_{x_{k+1}} + \bar{\rho}_y \Big) - \Big((-Q_y \hat{\rho}_{x_{min}} + \bar{\rho}_y^{\perp}) + \hat{\rho}_y^0 + (k+1)(\hat{y}_{min} - \hat{x}_{min}) \Big) \\ &= -Q_y(\hat{\rho}_{x_{k+1}} - \hat{\rho}_{x_{min}}) + \bar{\rho}_y^0 - \hat{\rho}^0_{y_0} - (k+1)(\hat{y}_{min} - \hat{x}_{min}) \end{aligned} \tag{J.3.41}$$

But $\hat{\rho}^0_{y_0}$, $\bar{\rho}^0_y$, $\bar{\rho}^0_x \in N_x^{\perp} \cap N_y^{\perp}$, and thus $Q_y \hat{\rho}^0_{y_0} = \hat{\rho}^0_{y_0}$, $Q_y \bar{\rho}^0_y = \bar{\rho}^0_y$, and $Q_y(\hat{y}_{min} - \hat{x}_{min}) = (\hat{y}_{min} - \hat{x}_{min})$. Thus:

$$\begin{aligned} \Delta_{y_{k+1}} &= -Q_y \Big(\hat{\rho}_{x_{k+1}} - (\hat{\rho}_{x_{min}} - \hat{\rho}^0_{y_0} + \bar{\rho}^0_y - (k+1)(\hat{y}_{min} - \hat{x}_{min})) \Big) \\ &= -Q_y Q_x \Big(\hat{\rho}_{x_{k+1}} - (\hat{\rho}_{x_{min}} - \hat{\rho}^0_{y_0} + \bar{\rho}^0_y - (k+1)(\hat{y}_{min} - \hat{x}_{min})) \Big) \\ &= -Q_y Q_x \Delta_{x_{k+1}} \end{aligned} \tag{J.3.42}$$

where this second to last line follows because every term on the right side is an element of $N_x^{\perp}$, and thus is an eigenvector of Q_x with eigenvalue of 1. Applying similar arguments show that:

$$\Delta_{x_{k+1}} = -\mathrm{Q}_x \mathrm{Q}_y \Delta_{y_k} \tag{J.3.43}$$

Now:

$$\Delta_{y_0} = \hat{\mu}_{y_0} - (\hat{\mu}_{y_{min}} + \hat{\mu}^0_{y_0}) = \hat{\mu}^{\perp}_{y_0} - \hat{\mu}_{y_{min}} \in (N_x^{\perp} \cap N_y^{\perp})^{\perp} \tag{J.3.44}$$

Since both $\mathrm{Q}_x \mathrm{Q}_y$ and $\mathrm{Q}_y \mathrm{Q}_x$ map the space $(N_x^{\perp} \cap N_y^{\perp})^{\perp}$ into $(N_x^{\perp} \cap N_y^{\perp})^{\perp}$, we must have

$$\Delta_{x_k}, \Delta_{y_k} \in (N_x^{\perp} \cap N_y^{\perp})^{\perp} \qquad \text{for all } k \tag{J.3.45}$$

But then:

$$\begin{aligned} \|\Delta_{y_{k+1}}\|^2_{\hat{Q}} &= \|\mathrm{Q}_y \mathrm{Q}_x \Delta_{x_{k+1}}\|^2_{\hat{Q}} \\ &\le \lambda_{max} \|\Delta_{x_{k+1}}\|^2_{\hat{Q}} \\ &= \lambda_{max} \|\mathrm{Q}_x \mathrm{Q}_y \Delta_{y_k}\|^2_{\hat{Q}} \\ &\le \lambda^2_{max} \|\Delta_{y_k}\|^2_{\hat{Q}} \end{aligned} \tag{J.3.46}$$

Now also:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= \hat{\mu}_{x_{k+1}} + \hat{\mu}_{y_k} \\ &= \hat{\mu}_{x_{min}} + \hat{\mu}_{y_{min}} + \bar{\mu}^0_x + \Delta_{y_k} + \Delta_{x_{k+1}} \\ &= \hat{\mathbf{x}}_{min} + \Delta_{x_{k+1}} + \Delta_{y_k} \end{aligned} \tag{J.3.47}$$

and similarly:

$$\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{y}}_{min} + \Delta_{y_{k+1}} + \Delta_{x_{k+1}} \tag{J.3.48}$$

Notice that the leftover piece of the initial output multiplier estimate $\hat{\mu}^0_{y_0}$ has disappeared together with the linear ramp term $k(\hat{\mathbf{y}}_{min} - \hat{\mathbf{x}}_{min})$. Thus:

$$\begin{aligned} \|(\hat{\mathbf{y}}_{k+1} - \hat{\mathbf{x}}_{k+1}) - (\hat{\mathbf{y}}_{min} - \hat{\mathbf{x}}_{min})\|^2_{\hat{Q}} &= \|\Delta_{y_{k+1}} - \Delta_{y_k}\|^2_{\hat{Q}} \\ &= \|\mathrm{Q}_y \mathrm{Q}_x (\Delta_{x_{k+1}} - \Delta_{x_k})\|^2_{\hat{Q}} \\ &\le \lambda_{max} \|\Delta_{x_{k+1}} - \Delta_{x_k}\|^2_{\hat{Q}} \\ &= \lambda_{max} \|(\hat{\mathbf{y}}_k - \hat{\mathbf{x}}_{k+1}) - (\hat{\mathbf{y}}_{min} - \hat{\mathbf{x}}_{min})\|^2_{\hat{Q}} \end{aligned} \tag{J.3.49}$$

where the second to last line follows because $\Delta_{x_{k+1}} - \Delta_{x_k} \in (N_x^{\perp} \cap N_y^{\perp})^{\perp}$. Similarly:

$$\|(\hat{\mathbf{y}}_k - \hat{\mathbf{x}}_{k+1}) - (\hat{\mathbf{y}}_{min} - \hat{\mathbf{x}}_{min})\|^2_{\hat{Q}} \le \lambda_{max} \|(\hat{\mathbf{y}}_k - \hat{\mathbf{x}}_k) - (\hat{\mathbf{y}}_{min} - \hat{\mathbf{x}}_{min})\|^2_{\hat{Q}} \tag{J.3.50}$$

References

1. Dante Youla, "Generalized Image Restoration by the Method of Alternating Orthogonal Projections," *IEEE Trans. Circuits. Syst.* CAS-25(9), pp.694-702 (Sept 1978).

Biography

Born to a typically middle class family in Chicago, Bruce Musicus began his life in the usual messy way. Refusing to talk until 3 (why start talking until you can do it right?) he then launched into an intensive analysis of arithmetic and reading (writing was never his specialty.) By four, he had developed the life style that would serve him for the rest of his career: up at 10:00 AM, eat, sit in a chair and read until 8:00 PM, eat, then sleep. At 4½ his life took a new direction; using bribes of chinese checkers, dominoes and cookies, his grandmother enticed him to practice the piano for 5 hours a day. Combining perfect pitch with the patience to sit endlessly moving only his fingers, Bruce endured a short career as a concert pianist. By five, however, having grown pudgy from too many cookies and a sedentary lifestyle, he secretly plotted to get away from it all. "I'll go to Harvard College", he thought. "I'll study Engineering, Applied Math and Physics, and graduate Summa Cum Laude, Phi Beta Kappa and win the Tau Beta Pi Engineering award!", he chortled. "I'll get an NSF graduate fellowship and go to MIT graduate school to study computer architecture!" And so it came to be. After 2 years of MIT, however, his ardor waned. Then it hit him! "I'll do Digital Signal Processing!" His life brightened! Heavenly choirs sang with angelic sweetness, women swooned with passion at his feet. One Master's thesis, one Electrical Engineering degree, and one PhD thesis later, he stands before you, MIT's newest faculty member, dazed, bewildered, confronting the future before him and pondering that immortal question, "What next?"