Signal Enhancement for Automatic Recognition of Noisy Speech

Shawn M. Verbout

RLE Technical Report No. 584

May 1994

Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139-4307

Signal Enhancement for Automatic Recognition of Noisy Speech

by

Shawn M. Verbout

Submitted to the Department of Electrical Engineering and Computer Science
on March 1, 1994, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

The problem of how to increase the robustness of an automatic speech recognition system with respect to additive environment noise is addressed. A two-stage approach is adopted for improving recognition performance in such an environment, whereby: (1) the noisy input waveform is filtered in order to enhance the underlying speech, and (2) the enhanced waveform is processed by the recognizer in an attempt to decode the utterance. Two sequential/adaptive algorithms are derived for use in the initial stage of the two-stage system; one of these algorithms is designed to operate in an environment with white noise, and the other in an environment with colored noise. A suite of experiments is conducted to evaluate the performance of the recognizer in the presence of noise, with and without the use of a front-end speech enhancement stage. Recognition accuracy is evaluated at the phone and word levels, and at a wide range of signal-to-noise ratios for both white and colored additive noise.

Thesis Supervisor: Alan V. Oppenheim
Title: Distinguished Professor of Electrical Engineering

Acknowledgments

Many people have influenced my personal and professional development in recent years, and I shall recognize them here. I would first like to thank my thesis advisor, Alan Oppenheim, for providing me with wise and steady guidance in my research, and for subjecting my work to his uncompromising standards of quality. I have great respect for Al, and I look forward to working with him as I continue my research in signal processing.

I also want to thank Victor Zue and Mike Phillips of the Spoken Language Systems Group at MIT. Victor was very generous in making available to me his group's computer resources and speech recognition software, and Mike was good enough to explain to me (not an easy job by any means) exactly how everything worked.

I extend my thanks to all the members of the Digital Signal Processing Group, who have consistently provided a supportive and stimulating environment in which to work. All of these people are quite brilliant and talented; it is an invaluable experience just to be around them for any length of time. I want to extend a special thanks to Steven Isabelle, a DSPG veteran who has regularly brought common sense and rational perspective into my way of thinking.

From the time I arrived at MIT for full-time graduate study, I have been supported financially through the Staff Associate Program at MIT Lincoln Laboratory. The person who took the necessary steps to make this support possible was Jerry Morse, my group leader at Lincoln. Recently, money has become a scarce resource at the laboratory, but in spite of financial constraints Jerry thought enough of me to invest in my future, and I am very grateful to him for it. I also want to thank my supervisor at Lincoln, Les Novak, for giving me the freedom to pursue promising new ideas, and for letting me know with candor when an idea was neither promising nor new. And I thank the rest of the crew in Group 47, who always succeed in rejuvenating me during the summer months with many laughs and memorable moments.

I want to express my appreciation for Mike Burl, whom I consider to be a highly skilled and efficient thinker, for changing the way that I approach technical problems. I learned a great deal by observing him and working with him at Lincoln Laboratory. Mike and his wife Maureen (another former Lincolnite) have been my good friends for many years. Another very good friend whom I met while at Lincoln, and one with whom I continue to work, write, and learn during the summer months, is Bill Irving. I want to thank Bill and his wife Anneli for the many good times, conversations, and dinners that we have shared.

Finally, on a quite personal note, I am deeply indebted to Stephen Cullinan. He did something many years ago that I think about every day, and it is something I will surely never forget: he saved my life. During that precarious time, however, it was my parents who provided the true life support, and it is they who remain my best friends on earth to this day. To John and Melanie Verbout, I express my warmest gratitude and love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Basic Problem: Recognizing Speech in an Adverse Environment

As the technology in automatic speech recognition (ASR) becomes increasingly advanced, greater consideration is given to speech as a viable means of interaction between humans and machines. However, as more ASR systems become incorporated into real-world applications, greater attention must be devoted to making these systems robust with respect to changes in their operating environments. Although it is true that many modern speech recognizers can perform well in certain adverse environments when trained extensively under actual environment conditions, even the most sophisticated ASR systems today are extremely sensitive to variations in these conditions, and must be carefully retrained for each new setting in which they are deployed.

In any given setting, a variety of environmental factors can interfere with obtaining quality measurements of spoken language. Among the many potential sources of environmental interference, three sources that are commonly addressed in the field of robust speech recognition are: (1) additive noise, (2) channel distortion, and (3) articulatory phenomena. To gain an appreciation for how each of these three factors can influence speech measurements, let us consider the scenario in which an ASR

system is deployed in a typical business office.

In the office environment, a spoken command intended for the recognizer can be corrupted by many different types of additive noise. Examples of noise sources in the workplace include background conversations, keyboard clicking, mechanical air circulation, and in general any movement of people, chairs, desk drawers, books, and paper. A noise source may produce sudden and transient interference, as in the case of a slamming door or a ringing telephone, or it may produce steady and persistent interference, as in the case of a humming computer fan or a buzzing overhead light.

In addition to undergoing corruption by additive noise, a speech signal can be distorted as it propagates from the vocal chords of the speaker to the sensor of the recognizer. Examples of such distortion include spectral shaping by the vocal tract, spectral shaping by the recording microphone, and reverberation due to acoustic reflections from walls, floors, and various objects in the room. Characteristics of the distorted speech signal received by the recognizer will also be affected by *variable* factors such as physiological differences between speakers, differences in the geometric relationship between the speaker and the microphone, and changes in room acoustics brought about by such events as opening a window or rearranging office furniture.

Finally, in addition to being altered by the effects of noise and distortion, speech can be influenced heavily by changes in the articulatory patterns of a speaker. For example, if a speaker attempts to compensate for the ambient noise in the room, or is simply uncomfortable because of the awkwardness of interacting with the recognizer, then the resulting speech may have very unnatural rhythm and exhibit significant spectral tilt. Dealing with this kind of environmental influence is extremely difficult because it is inherently a part of the speech production process, and therefore varies as rapidly as the speech itself.

Clearly, each of these environmental factors can affect the critical attributes of speech used by a speech recognizer, and hence can cause severe degradation in recognition performance. Though all of these factors eventually must be managed to achieve robustness in speech recognition, in this thesis we shall address only the problem of compensating for additive environmental noise.

1.2 An Approach to Robustness Against Environmental Noise

To develop a preliminary solution for dealing with a noisy operating environment, we consider the situation depicted in Figure 1-1. In this figure, we see that the speaker has produced an utterance intended to be processed by the recognizer. The environment contains many sources of noise, however, and the utterance is corrupted before it reaches the recording microphone. In general, such corruption will be unanticipated by the speech recognition system, since the system has been trained to recognize speech that is free of extraneous noise. Because the conditions in the operating environment are very different from those of the training environment, it is likely that the performance of the recognizer will be quite poor.

To boost the noise immunity of the recognizer, we introduce the augmented two-stage system shown in Figure 1-2. In the first stage of this modified system, the input waveform is filtered in order to enhance the underlying speech; in the second stage, the enhanced waveform is simply processed by the recognizer as before. If, in this proposed two-stage system, the front-end speech enhancement component yields a reasonably accurate representation of the original utterance, then recognition performance can be expected to improve. This basic two-stage approach to improving the robustness of the recognizer is the approach we shall pursue in the sequel.

1.3 Previous Work in Robust Speech Recognition

With the recent appearance of ASR systems in many practical applications, the issue of environmental robustness (and, in particular, the issue of recognition of noisy speech) has attracted increasing attention in the research literature. Much of the research in robust speech recognition has been done within the two-stage framework described in Section 1.2, whereby a suitable speech enhancement subsystem is used as a preprocessor for the recognizer.

Early development of algorithms for enhancing noisy and distorted speech was

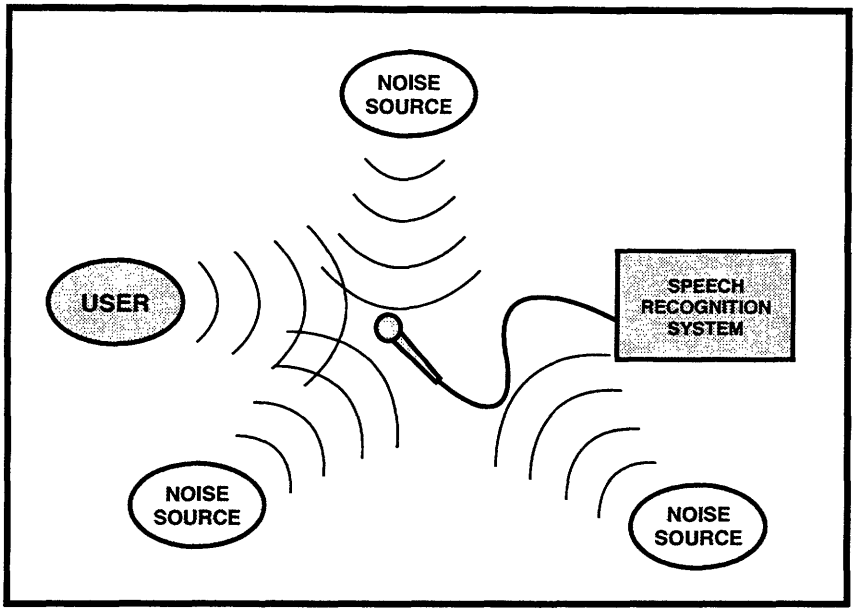


Figure 1-1: Depiction of an adverse environment for automatic speech recognition in which multiple noise sources interfere with the utterance of a system user.

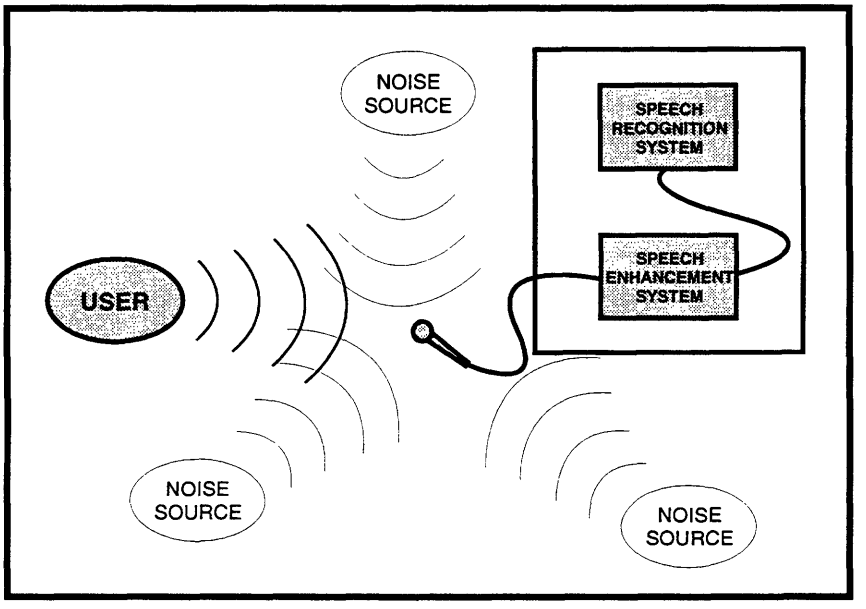


Figure 1-2: Illustration of the noise-reduction strategy in which a front-end speech enhancement system is used.

based largely on classical theory in signal processing and estimation; a representative survey of such classical methods is given in the book by Lim [18]. An important development in this early work was the speech enhancement algorithm proposed by Lim and Oppenheim [19]. In this frame-based algorithm, estimates of both the underlying speech and the autoregressive speech parameters are iteratively refined until a termination criterion is satisfied. At each iteration of the algorithm, the speech signal is first estimated with a Weiner filter using the current values of the parameter estimates; then, the parameters are re-estimated using the current value of the speech signal estimate. In a later paper by Feder, Oppenheim, and Weinstein [7], this iterative estimation technique was identified as a particular implementation of the EM (Expectation-Maximization) algorithm [4], which is now a widely used mathematical tool for computing maximum likelihood estimates. Hansen and Clements [13] later extended the work of Lim and Oppenheim by imposing spectral constraints both within and across frame boundaries to ensure optimum quality for many different types of speech sounds.

Recently, there have been several notable developments in speech-enhancement preprocessing algorithms for recognition systems. Ephraim *et al* [5] introduced an approach for enhancing speech using a Hidden Markov Model (HMM). In this method, both the speech and noise are modeled locally as autoregressive processes, and the underlying utterance is modeled with the use of an HMM. The speech and noise parameters are computed iteratively with the EM algorithm such that the likelihood function over the entire utterance is eventually maximized. This approach has been shown to yield significant gains in SNR; however, since it involves a re-estimation of the entire speech waveform at every iteration, it is extremely computationally expensive. Other research more closely tied to the speech recognition problem has been contributed by Acero [1]. The basic idea put forth by Acero is that the speech recorded in the operating environment should be transformed so that it assumes the critical attributes of the speech used to train the recognizer. To accomplish this goal, Acero has applied the classical techniques of spectral subtraction and spectral normalization to provide *joint* compensation for the effects of environmental noise and

distortion. When used in conjunction with the SPHINX recognition system [16], these techniques yielded substantial improvement in recognition accuracy over independent compensation strategies developed in previous research.

In addition to recently developed methods based on optimizing mathematical criteria, a number of novel speech enhancement techniques based on the properties of human hearing have also been proposed. Ghitza [9] developed a computational model of the peripheral auditory system known as the Ensemble Interval Histogram (EIH) to be used as a front end for a speech recognition system. The EIH has been shown to be remarkably robust with respect to additive noise, and has produced extremely accurate estimates of speech spectra in SNR levels as low as 0 dB. A substantial gain in performance was reported when the EIH was employed in a simple speech recognition task. More recently, Tsoukalas *et al* [40] proposed an enhancement technique based on the theory of psychoacoustics that yields a significant gain in SNR and simultaneously preserves the intelligibility of the underlying speech. This technique, which exploits the masking property of the auditory system, first identifies the audible noise components in the corrupted signal and then suppresses these components through adaptive nonlinear spectral modification. This perceptually-based method was found to perform well in SNR levels of 10 dB or higher, and was found to be superior to the classical spectral subtraction technique in very severe noise environments. Another approach based on noise masking has recently been studied in the cepstral domain by Mellor and Varga [22].

A number of other potentially useful techniques have appeared in the signal processing literature (e.g., [3, 6, 12, 17, 43]), and many more will undoubtedly be developed in the near term as research in robust speech recognition continues to expand rapidly.

1.4 Purpose and Scope of this Thesis

The long-term objective of research in this area is to develop the signal processing and speech recognition technology required to design an ASR system that performs

at a consistently high level even in adverse conditions. In the present work, however, we focus exclusively on improving the robustness of an existing ASR system in the presence of stationary additive noise by using a signal-enhancement preprocessing algorithm as discussed in Section 1.2. The preprocessing algorithms used in the experiments described herein have evolved directly from previous work done by Oppenheim *et al* [24, 25] in the context of active noise cancellation; each algorithm employs an autoregressive model for the speech, and incorporates prior knowledge about the statistics of the noise. The speech recognition system used in the experiments is the SUMMIT system, which has been developed by Zue and other researchers from the Spoken Language Systems Group at the MIT Laboratory for Computer Science [45, 46, 47]. We seek to evaluate the recognition performance of the SUMMIT system under noisy conditions, with and without the use of one of the preprocessing algorithms. Since the SUMMIT system has not previously been tested with respect to its noise immunity, these studies will establish baseline levels of performance from which to direct future research.

1.5 Thesis Organization and Development

The material presented in the sequel is organized in the following way:

In Chapter 2, we address the basic problem of how to enhance speech that has been corrupted by additive environmental noise. First, we introduce stochastic models for both the speech and noise processes, and formulate objectives for a speech enhancement algorithm. We then derive two separate solutions to the enhancement problem that take the form of sequential, time-adaptive algorithms; one of these algorithms is designed for the case in which the corrupting noise is temporally uncorrelated, or *white*, and the other for the case in which the noise is temporally correlated, or *colored.*

In Chapter 3, we describe how the enhancement algorithms were incorporated in a set of baseline speech recognition experiments. These experiments were designed to evaluate the performance of an ASR system, with and without the use of a front-end

speech enhancement component, in a variety of environment conditions. Recognition accuracy was tested at different linguistic levels (specifically, the *phone* and *word* levels), and at different input SNR values for both the white-noise and colored-noise cases. We provide a general overview of each experiment that was conducted, and then describe in detail the various components that comprised each experiment.

In Chapter 4, we present and discuss the numerical results generated in these recognition performance tests. First, we examine the phone and word accuracy rates achieved by the recognizer as a function of the input SNR for the case in which white noise was added to the speech; we then analyze an analogous set of results for the colored-noise case. To aid in the interpretation of the results, we introduce a further series of experiments designed to evaluate the performance of each speech enhancement algorithm operating in isolation. In particular, these auxiliary tests are intended to measure the gain in speech quality afforded by each algorithm as a function of the input SNR.

Finally, in Chapter 5, we briefly summarize the work done in the thesis, put forth conclusions regarding the most significant results obtained, and give directions for future research in the area of robust speech recognition.

Chapter 2

Algorithms for Enhancing Noisy Speech

In this chapter, we address the basic problem of how to enhance speech that has been corrupted by additive environmental noise. First, we introduce stochastic models for both the speech and noise processes, and formulate objectives for a speech enhancement algorithm. We then derive two separate solutions to the enhancement problem that take the form of sequential, time-adaptive algorithms; one of these algorithms is designed for the case in which the corrupting noise is white, and the other for the case in which the noise is colored. We show the results of applying each algorithm to actual speech waveforms corrupted by computer-generated noise.

2.1 The Speech Enhancement Problem: Definitions and Models

Before considering the details of designing a speech enhancement system, we must first impose some probabilistic structure on the speech and noise processes observed in the recognition environment. In Figure 2-1, we identify the various signals entering at the front end of the augmented ASR system. The waveform $s(t)$ represents the speech produced by the system user, and the waveform $v(t)$ represents the additive

disturbance generated within the environment. The composite waveform $z(t)$ received at the microphone is therefore given by

$$z(t) = s(t) + v(t). \tag{2.1}$$

The signal enhancement algorithm processes this corrupted speech waveform and generates the output signal $\hat{s}(t)$, which is intended to be an accurate representation of the original speech.

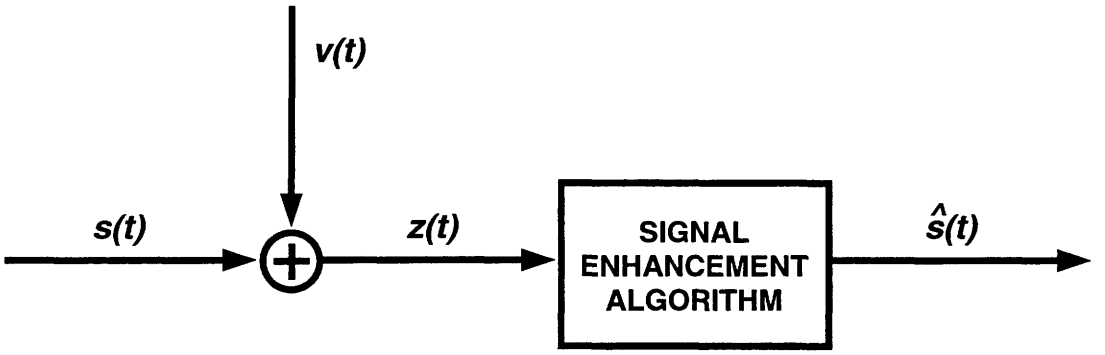


Figure 2-1: Definition of waveforms at the front end of the augmented speech recognition system.

In general, the signal $s(t)$ will be nonstationary, since the time attributes of speech vary dramatically even over the duration of a single word. However, since the first- and second-order statistics of relatively short sections of a speech signal (i.e., sections that are 15 to 30 milliseconds long) are approximately constant, speech is typically modeled over brief time intervals as the steady-state output of a linear time-invariant system driven by white noise [31]. Here we shall assume that $s(t)$ has the autoregressive form

$$s(t) = -\sum_{k=1}^{p} \alpha_k s(t-k) + u(t), \tag{2.2}$$

where p is the order of the speech process over the interval, $\alpha_1, \alpha_2, \ldots, \alpha_p$ are the autoregressive parameters of the speech, and $u(t)$ is a white Gaussian process with

$$E\{u(t)\} = 0 \quad \text{and} \quad E\left\{u^2(t)\right\} = \sigma_u^2. \tag{2.3}$$

In many instances, acoustic noise, like speech, is approximately stationary over brief time intervals, and may also admit a linear representation such as the one given above. Thus, we shall assume that $v(t)$ also has an autoregressive form given by

$$v(t) = -\sum_{k=1}^{q} \beta_k v(t-k) + w(t), \tag{2.4}$$

where q is the order of the noise process, $\beta_1, \beta_2, \ldots, \beta_q$ are the autoregressive parameters of the noise, and $w(t)$ is a white Gaussian process with

$$E\{w(t)\} = 0 \quad \text{and} \quad E\left\{w^2(t)\right\} = \sigma_w^2. \tag{2.5}$$

In the remaining part of this chapter we develop two speech enhancement algorithms, each of which is tailored to a specific case for the additive disturbance $v(t)$. In Section 2.2, we consider the degenerate case in which all of the autoregressive parameters are zero, and $v(t)$ is a white noise process with

$$E\{v(t)\} = 0 \quad \text{and} \quad E\left\{v^2(t)\right\} = \sigma_v^2. \tag{2.6}$$

In Section 2.3, we consider a case in which the autoregressive parameters are nonzero, so that $v(t)$ is in fact highly correlated in time.

For the algorithm derivations presented in Sections 2.2 and 2.3, we assume that the speech and noise processes are statistically independent. Furthermore, we restrict our attention to enhancement algorithms that use only past and present observations to produce a minimum mean-square error (MMSE) estimate of the current speech signal value. Thus, at time t, we wish to compute the estimate $\hat{s}(t)$ that satisfies

$$\hat{s}(t) = \underset{\xi}{\operatorname{argmin}}\, E\left\{(s(t)-\xi)^2 \middle| z(0), z(1), \ldots, z(t)\right\}, \tag{2.7}$$

where the argmin operator yields the argument of the expectation at which the minimum value occurs. For the above MMSE criterion, it is well known that the optimal

estimate $\hat{s}(t)$ is the conditional mean given by

$$\hat{s}(t) = E\{s(t)|z(0), z(1), \ldots, z(t)\}. \tag{2.8}$$

In the following sections, we develop techniques for computing such a conditional mean estimate sequentially in time.

2.2 Enhancing Speech in Additive White Noise

Since our underlying speech signal model is autoregressive, and the corrupting noise is additive, we can conveniently represent our speech measurement model using the classical linear state-space equations from dynamical system theory. Specifically, at time t we construct a state vector $\mathbf{x}(t)$ that consists of the present speech sample together with the previous p consecutive speech samples, defined by

$$\mathbf{x}(t) = \begin{bmatrix} s(t) & s(t-1) & \cdots & s(t-p) \end{bmatrix}^T. \tag{2.9}$$

Using this state vector definition, we can express our speech measurement model as

$$\mathbf{x}(t+1) = \mathbf{F}\mathbf{x}(t) + \mathbf{g}u(t) \tag{2.10}$$

$$z(t) = \mathbf{g}^T\mathbf{x}(t) + v(t), \tag{2.11}$$

where $\mathbf{F}$ is a $(p+1) \times (p+1)$ state transition matrix, given by

$$\mathbf{F} = \begin{bmatrix} -\alpha_1 & -\alpha_2 & \cdots & \cdots & -\alpha_p & 0 \\ 1 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 1 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & 1 & 0 \end{bmatrix}, \tag{2.12}$$

and $\mathbf{g}$ is a $(p+1) \times 1$ elementary vector, given by

$$\mathbf{g} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T . \tag{2.13}$$

Observe that the matrix $\mathbf{F}$ consists mainly of zeroes, with the exception of its first row, whose elements are the speech signal parameters, and its subdiagonal, whose elements are all ones. Thus, $\mathbf{F}$ is structured to effect the autoregression and to delay the signal samples in the state vector by one time unit; the vector $\mathbf{g}$, on the other hand, is designed to incorporate only the current signal sample (i.e., the first element) of the state vector into the measurement.

To relate the above state-space representation to our original speech enhancement problem, we express the conditional mean estimate from Equation (2.8) as

$$\begin{aligned} \widehat{s}(t) &= E\left\{ \mathbf{g}^T \mathbf{x}(t) \middle| z(0), z(1), \ldots, z(t) \right\} && (2.14) \\ &= \mathbf{g}^T E\left\{ \mathbf{x}(t) \middle| z(0), z(1), \ldots, z(t) \right\}, && (2.15) \end{aligned}$$

where the first equality follows from the definitions of $\mathbf{g}$ and $\mathbf{x}(t)$, and the second follows from the linearity of the expectation operator. Now, based on the assumption that the speech parameters $\alpha_1, \alpha_2, \ldots, \alpha_p, \sigma_u^2$ and the noise parameter σ_v^2 are all precisely known, the conditional expectation of the state vector appearing in Equation (2.15) can be computed efficiently using a Kalman filter that is designed specifically for the measurement model of Equations (2.10) and (2.11). In the remaining portion of this chapter, we shall pursue a solution based on the Kalman filtering algorithm.

To prepare for our discussion on sequential state estimation, we adopt a standard notation from the literature on Kalman filtering theory whereby the optimal estimate of the state vector at time t based on measurements up to time τ is denoted by $\widehat{\mathbf{x}}(t|\tau)$, and the error covariance associated with this estimate is denoted by $\mathbf{P}(t|\tau)$. For the special case of interest to us in which $\tau = t$, we have by definition

$$\widehat{\mathbf{x}}(t|t) = E\left\{ \mathbf{x}(t) \middle| z(0), z(1), \ldots, z(t) \right\} \tag{2.16}$$

$$\mathbf{P}(t|t) = E\left\{ (\hat{\mathbf{x}}(t|t) - \mathbf{x}(t)) (\hat{\mathbf{x}}(t|t) - \mathbf{x}(t))^T \Big| z(0), z(1), \ldots, z(t) \right\}. \quad (2.17)$$

Using the classical Kalman filter formulation, we can compute both the state estimate $\hat{\mathbf{x}}(t|t)$ and its error covariance $\mathbf{P}(t|t)$ recursively in time. These recursive computations are typically decomposed into two sets of updates that are performed sequentially at each time instant, namely: (1) the *time update equations*, in which the previous state estimate and its error covariance are propagated forward to the current time based on knowledge of the underlying system dynamics and *a priori* noise statistics, and (2) the *measurement update equations*, in which the propagated state estimate and error covariance are adjusted to reflect the new information in the current measurement [2]. These two sets of updates are specified as follows:

TIME UPDATE EQUATIONS

$$\hat{\mathbf{x}}(t|t-1) = \mathbf{F}\hat{\mathbf{x}}(t-1|t-1) \quad (2.18)$$

$$\mathbf{P}(t|t-1) = \mathbf{F}\mathbf{P}(t-1|t-1)\mathbf{F}^T + \sigma_u^2 \mathbf{g}\mathbf{g}^T \quad (2.19)$$

MEASUREMENT UPDATE EQUATIONS

$$\hat{\mathbf{x}}(t|t) = \hat{\mathbf{x}}(t|t-1) + \mathbf{k}(t)\left(z(t) - \mathbf{g}^T\hat{\mathbf{x}}(t|t-1)\right) \quad (2.20)$$

$$\mathbf{P}(t|t) = \mathbf{P}(t|t-1) - \mathbf{k}(t)\mathbf{g}^T\mathbf{P}(t|t-1) \quad (2.21)$$

In the measurement update equations above, the vector $\mathbf{k}(t)$, which defines the direction of the adjustment made to the propagated state estimate $\mathbf{x}(t|t-1)$, is commonly known as the *Kalman gain*, and is given by

$$\mathbf{k}(t) = \frac{\mathbf{P}(t|t-1)\mathbf{g}}{\mathbf{g}^T\mathbf{P}(t|t-1)\mathbf{g} + \sigma_v^2}. \quad (2.22)$$

In this case, the Kalman gain is simply a scaled version of the first column of the propagated error covariance matrix $\mathbf{P}(t|t-1)$. Thus, the gain vector $\mathbf{k}(t)$ consists of the cross-correlations between the error in the estimate of $s(t)$ and the errors in the estimates of $s(t-1), s(t-2), \ldots, s(t-p)$ based on measurements received up

to time $t-1$. The strength of these cross-correlations indicate the extent to which the elements of the vector $\mathbf{x}(t|t-1)$ can be corrected using the signal portion of the measurement received at time t. In addition, each element of the Kalman gain vector is scaled such that it is inversely proportional to the average power in the measurement noise. Thus, for a case in which noise is dominant in the measurement, only small corrections to the projected state estimate should be made; on the other hand, for a case in which the noise power is very small, the measurements will be rich in signal information, and therefore much larger adjustments to the state estimate are called for.

Recall that the Kalman filtering equations given above require exact knowledge of all parameters in our measurement model. In an actual speech processing environment, however, we will almost certainly have no prior information about the speech parameters at any time. In fact, it is reasonable to assume that we can, under suitably stable environment conditions, accurately measure only the statistics of the background noise. Thus, in the sequel we shall assume that only the noise parameters are known, and that the speech parameters must be estimated jointly with the speech signal values.

As one might expect by examining the linear signal model of Equation (2.2), the signal parameters can be determined uniquely if the second-order signal statistics are precisely known. To find the relationship between the signal parameters and the signal correlation function, we first define a parameter vector $\boldsymbol{\alpha}$ as

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_p \end{bmatrix}^T \tag{2.23}$$

and then rewrite Equation (2.2) in vector form as

$$\mathbf{x}^T(t) \begin{bmatrix} 1 \\ \boldsymbol{\alpha} \end{bmatrix} = u(t). \tag{2.24}$$

This equivalent expression for the signal model follows from a simple algebraic rearrangement of the original equation and use of the definition of the state vector $\mathbf{x}(t)$.

If we now multiply both sides of this new vector equation by $\mathbf{x}(t)$ and apply the expectation operator, we obtain the expression

$$E\left\{\mathbf{x}(t)\mathbf{x}^T(t)\right\}\begin{bmatrix} 1 \\ \boldsymbol{\alpha} \end{bmatrix} = E\left\{\mathbf{x}(t)u(t)\right\}. \tag{2.25}$$

By assumption, the random variable $u(t)$ is zero-mean and is statistically independent of all signal values $s(\tau)$ for $\tau < t$; hence, in particular we have

$$E\left\{s(t-k)u(t)\right\} = 0 \quad \text{for } k = 1, 2, \ldots, p. \tag{2.26}$$

Moreover, by using these orthogonality conditions in conjunction with our autoregressive signal model, we see that the correlation between $u(t)$ and the current signal value is given by

$$\begin{aligned} E\left\{s(t)u(t)\right\} &= E\left\{\left(-\sum_{k=1}^{p}\alpha_k s(t-k) + u(t)\right)u(t)\right\} && (2.27)\\ &= E\left\{u^2(t)\right\} && (2.28)\\ &= \sigma_u^2. && (2.29) \end{aligned}$$

Upon substituting the above correlation values into Equation (2.25), we obtain the new expression

$$\mathbf{R}\begin{bmatrix} 1 \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \sigma_u^2 \\ \mathbf{0} \end{bmatrix}, \tag{2.30}$$

where $\mathbf{R}$ is the $(p+1) \times (p+1)$ signal correlation matrix defined by

$$\mathbf{R} = E\left\{\mathbf{x}(t)\mathbf{x}^T(t)\right\}. \tag{2.31}$$

Equation (2.30) reveals the relationship between the signal parameters and the second-order signal statistics. The scalar equations that comprise this matrix-vector expression are commonly known as the *Yule-Walker equations* [14], and they will form the

basis for our parameter estimation algorithm.

Clearly, if the correlation matrix $\mathbf{R}$ is known, then Equation (2.30) can be used to solve explicitly for the parameters $\boldsymbol{\alpha}$ and σ_u^2. However, as we have already mentioned, the signal statistics will not be known *a priori*, and hence must be estimated. Furthermore, any method we choose for estimating these statistics must be capable of adapting to changes in the structure of the signal with time, in view of our initial remark that speech can be considered stationary only over brief intervals. We can compute an estimate of the signal correlation matrix based on measurements received up to time t by using the error covariance matrix $\mathbf{P}(t|t)$ from Equation (2.17). In particular, if we define the state estimation error at time t as

$$\mathbf{e}(t) = \hat{\mathbf{x}}(t|t) - \mathbf{x}(t), \tag{2.32}$$

then we can express $\mathbf{P}(t|t)$ as (dropping the conditioning notation for convenience)

$$\begin{aligned} \mathbf{P}(t|t) &= E\left\{(\hat{\mathbf{x}}(t|t) - \mathbf{x}(t))\,\mathbf{e}^T(t)\right\} && (2.33)\\ &= E\left\{\hat{\mathbf{x}}(t|t)\mathbf{e}^T(t)\right\} - E\left\{\mathbf{x}(t)\mathbf{e}^T(t)\right\}. && (2.34) \end{aligned}$$

Because $\hat{\mathbf{x}}(t|t)$ is a minimum mean square linear estimate of the state vector, it is orthogonal to the error term $\mathbf{e}(t)$. Hence, the first term in Equation (2.34) vanishes, and we are left with

$$\begin{aligned} \mathbf{P}(t|t) &= -E\left\{\mathbf{x}(t)\left(\hat{\mathbf{x}}(t|t) - \mathbf{x}(t)\right)^T\right\} && (2.35)\\ &= E\left\{\mathbf{x}(t)\mathbf{x}^T(t)\right\} - E\left\{\mathbf{x}(t)\hat{\mathbf{x}}^T(t|t)\right\} && (2.36)\\ &= E\left\{\mathbf{x}(t)\mathbf{x}^T(t)\right\} - E\left\{\left(\hat{\mathbf{x}}(t|t) - \mathbf{e}(t)\right)\hat{\mathbf{x}}^T(t|t)\right\} && (2.37)\\ &= E\left\{\mathbf{x}(t)\mathbf{x}^T(t)\right\} - E\left\{\hat{\mathbf{x}}(t|t)\hat{\mathbf{x}}^T(t|t)\right\}. && (2.38) \end{aligned}$$

By dropping the expectations from this last equation, we see that a reasonable instantaneous estimate of the outer product $\mathbf{x}(t)\mathbf{x}^T(t)$ is given by

$$\widehat{\mathbf{x}(t)\mathbf{x}^T(t)} = \hat{\mathbf{x}}(t|t)\hat{\mathbf{x}}^T(t|t) + \mathbf{P}(t|t). \tag{2.39}$$

Moreover, these instantaneous estimates generated at each time can be incorporated into a weighted average to be used as an estimate of the current signal correlation matrix. A computationally convenient way of averaging these instantaneous estimates is to use an exponential weighting, whereby the most recently generated terms have the most significant contribution to the average, and terms generated in the remote past have a negligible contribution. For the case of exponential weighting, the estimate $\widehat{\mathbf{R}}(t)$ of the correlation matrix is given by

$$\widehat{\mathbf{R}}(t) = \frac{\sum_{\tau=0}^{t} \lambda^{t-\tau} \widehat{\mathbf{x}(\tau)\mathbf{x}^T(\tau)}}{\sum_{\tau=0}^{t} \lambda^{t-\tau}}, \tag{2.40}$$

where λ is the so-called "forgetting factor" that determines the memory length of the estimator, chosen such that $0 < \lambda < 1$. If we assume that the above estimator is operating in the steady state (i.e., that t is large), then for the overall scaling term in the estimate we can write

$$\frac{1}{\sum_{\tau=0}^{t} \lambda^{t-\tau}} = \frac{1-\lambda}{1-\lambda^t} \approx 1-\lambda \tag{2.41}$$

With this simplification, the estimate $\widehat{\mathbf{R}}(t)$ is easily computed recursively, as shown by

$$\widehat{\mathbf{R}}(t) = (1-\lambda) \sum_{\tau=0}^{t} \lambda^{t-\tau} \widehat{\mathbf{x}(\tau)\mathbf{x}^T(\tau)} \tag{2.42}$$

$$= (1-\lambda) \left[\widehat{\mathbf{x}(t)\mathbf{x}^T(t)} + \lambda \sum_{\tau=0}^{t-1} \lambda^{t-1-\tau} \widehat{\mathbf{x}(\tau)\mathbf{x}^T(\tau)} \right] \tag{2.43}$$

$$= (1-\lambda) \widehat{\mathbf{x}(t)\mathbf{x}^T(t)} + \lambda \widehat{\mathbf{R}}(t-1) \tag{2.44}$$

Since we now have a method of generating an estimate of the signal correlation matrix at each time instant, we can consider formulating a parameter estimation scheme based on the Yule-Walker equations derived earlier. In particular, we replace the matrix $\mathbf{R}$ in Equation (2.30) by its current estimate $\widehat{\mathbf{R}}(t)$ to obtain a set of equations in the unknown parameters $\boldsymbol{\alpha}$ and σ_u^2 that can be uniquely solved at time t. If we

first partition the matrix $\widehat{\mathbf{R}}(t)$ as

$$\widehat{\mathbf{R}}(t) = \begin{bmatrix} \hat{r}_{11}(t) & \hat{\mathbf{r}}_{21}^T(t) \\ \hat{\mathbf{r}}_{21}(t) & \widehat{\mathbf{R}}_{22}(t) \end{bmatrix}, \tag{2.45}$$

where $\hat{r}_{11}(t)$ is 1×1, $\hat{\mathbf{r}}_{21}(t)$ is $p \times 1$, and $\widehat{\mathbf{R}}_{22}(t)$ is $p \times p$, then by using the above strategy Equation (2.30) becomes

$$\begin{bmatrix} \hat{r}_{11}(t) & \hat{\mathbf{r}}_{21}^T(t) \\ \hat{\mathbf{r}}_{21}(t) & \widehat{\mathbf{R}}_{22}(t) \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \sigma_u^2 \\ \mathbf{0} \end{bmatrix}. \tag{2.46}$$

After simple rearrangement, this expression yields the two equations

$$\hat{\mathbf{r}}_{21}(t) + \widehat{\mathbf{R}}_{22}(t)\boldsymbol{\alpha} = \mathbf{0} \tag{2.47}$$

$$\sigma_u^2 - \hat{r}_{11}(t) - \hat{\mathbf{r}}_{21}^T(t)\boldsymbol{\alpha} = 0. \tag{2.48}$$

Although these equations can readily be solved for the unknown parameters, a direct solution at each time t requires inversion of the matrix $\widehat{\mathbf{R}}_{22}(t)$, and is therefore computationally unattractive. A somewhat less direct but much more computationally efficient solution can be obtained by applying the theory of stochastic approximation [10, 23, 33, 34]. In a typical stochastic approximation problem, we observe the values of a sequence of identically distributed random vectors $\{\mathbf{y}(t)\}$, and we seek the value of an unknown parameter vector $\boldsymbol{\theta}$ that satisfies an equation of the form

$$E\{\mathbf{f}(\mathbf{y}(t), \boldsymbol{\theta})\} = \mathbf{0}, \tag{2.49}$$

where $\mathbf{f}(\cdot, \cdot)$ is a deterministic vector function that is known in advance. With the stochastic approximation approach, a solution to this equation evolves sequentially in time as each new observation becomes available. A standard search algorithm for converging on a solution to Equation (2.49), which was proposed in its original form

by Robbins and Munro [34], is the recursion given by

$$\widehat{\boldsymbol{\theta}}(t+1) = \widehat{\boldsymbol{\theta}}(t) - \gamma(t)\mathbf{Q}(t)\mathbf{f}(\mathbf{y}(t), \widehat{\boldsymbol{\theta}}(t)), \tag{2.50}$$

where $\widehat{\boldsymbol{\theta}}(t)$ is the estimate of $\boldsymbol{\theta}$ at time t, $\gamma(t)$ is a user-specified scalar determining the size of the adjustment made to $\widehat{\boldsymbol{\theta}}(t)$, and $\mathbf{Q}(t)$ is a user-specified matrix of appropriate dimension determining the direction of the adjustment made to $\widehat{\boldsymbol{\theta}}(t)$.

To gain an understanding of how the above algorithm can be used for parameter estimation, let us briefly consider a simple application. Suppose that we wish to estimate the common mean of the random vectors $\{\mathbf{y}(t)\}$, and that the actual value of this mean is $\boldsymbol{\theta}$. We define $\mathbf{f}(\cdot,\cdot)$ in this case as

$$\mathbf{f}(\mathbf{y}(t), \boldsymbol{\theta}) = \boldsymbol{\theta} - \mathbf{y}(t), \tag{2.51}$$

so that the expectation in Equation (2.49) is satisfied, and we choose the updating scale and direction parameters as

$$\gamma(t) = \frac{1}{t+1} \tag{2.52}$$

$$\mathbf{Q}(t) = \mathbf{I}. \tag{2.53}$$

With these definitions, a stochastic approximation algorithm for estimating the mean is given by

$$\widehat{\boldsymbol{\theta}}(t+1) = \widehat{\boldsymbol{\theta}}(t) - \frac{1}{t+1}(\widehat{\boldsymbol{\theta}}(t) - \mathbf{y}(t)). \tag{2.54}$$

If this algorithm is initialized with the value $\widehat{\boldsymbol{\theta}}(0) = \mathbf{0}$, then after N observations have been processed the estimate $\widehat{\boldsymbol{\theta}}(N)$ is identical to the sample mean estimate. Thus, we have

$$\widehat{\boldsymbol{\theta}}(N) = \frac{1}{N}\sum_{\tau=0}^{N-1}\mathbf{y}(\tau), \tag{2.55}$$

which is easily verified by iterating Equation (2.54) a total of N times.

The foregoing example illustrates the power and generality of the stochastic approximation method; in particular, it shows that a simple application of the stochastic approximation method can lead to a classical and widely used estimator as a special case. In any given estimation problem, an infinite variety of solutions can be realized through different choices of the algorithm parameters $\gamma(t)$, $\mathbf{Q}(t)$, and $\mathbf{f}(\cdot,\cdot)$. For example, if Equation (2.49) represents a condition for minimizing a prespecified cost function, and if derivatives of the cost function can be readily computed, then directional information can be incorporated through the matrix parameter $\mathbf{Q}(t)$ in order to accelerate convergence [20]. Moreover, if it is known that the observations $\{\mathbf{y}(t)\}$ are not identically distributed, but instead are characterized by a distribution that varies slowly over time, then exponential data weighting can be applied through the step-size parameter $\gamma(t)$ [11].

To apply the stochastic approximation method to our present parameter estimation problem, we define $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta} = \begin{bmatrix} \sigma_u^2 \\ \boldsymbol{\alpha} \end{bmatrix}, \tag{2.56}$$

and we consider the random vector $\mathbf{y}(t)$ to be our system state vector $\mathbf{x}(t)$. With these definitions, it is clear that the previously derived Yule-Walker equations can, after some manipulation, be cast in the general form of Equation (2.49). Based on Equations (2.47) and (2.48), a suitable stochastic approximation algorithm for our problem is specified by

$$\hat{\boldsymbol{\alpha}}(t+1) = \hat{\boldsymbol{\alpha}}(t) - \gamma(t)\left[\hat{\mathbf{r}}_{21}(t) + \widehat{\mathbf{R}}_{22}(t)\hat{\boldsymbol{\alpha}}(t)\right] \tag{2.57}$$

$$\widehat{\sigma_u^2}(t+1) = \widehat{\sigma_u^2}(t) - \gamma(t)\left[\widehat{\sigma_u^2}(t) - \hat{r}_{11}(t) - \hat{\mathbf{r}}_{21}^T(t)\hat{\boldsymbol{\alpha}}(t)\right]. \tag{2.58}$$

In these equations, we have set $\mathbf{Q}(t) = \mathbf{I}$, and have defined $\hat{\boldsymbol{\alpha}}(t)$ and $\widehat{\sigma_u^2}(t)$ to be the estimates at time t of the signal parameters $\boldsymbol{\alpha}$ and σ_u^2, respectively.

We can now include Equations (2.57) and (2.58), as well as Equation (2.44),

as part of our speech enhancement algorithm. With these additions, the complete sequential/adaptive algorithm consists of two main stages: (1) the *signal estimation algorithm*, in which the Kalman filter equations (based on the original measurement model) are implemented using the current estimates of the signal parameters, and (2) the *parameter estimation algorithm*, in which the stochastic approximation formulas (based on the Yule-Walker equations) are implemented using the current estimate of the signal correlation matrix. The two corresponding sets of computations are specified as follows:

Signal Estimation Algorithm:

TIME UPDATE EQUATIONS

$$\hat{\mathbf{x}}(t|t-1) = \hat{\mathbf{F}}(t)\hat{\mathbf{x}}(t-1|t-1) \tag{2.59}$$

$$\hat{\mathbf{P}}(t|t-1) = \hat{\mathbf{F}}(t)\hat{\mathbf{P}}(t-1|t-1)\hat{\mathbf{F}}^T(t) + \widehat{\sigma_u^2}(t)\mathbf{g}\mathbf{g}^T \tag{2.60}$$

MEASUREMENT UPDATE EQUATIONS

$$\hat{\mathbf{x}}(t|t) = \hat{\mathbf{x}}(t|t-1) + \hat{\mathbf{k}}(t)\left(z(t) - \mathbf{g}^T\hat{\mathbf{x}}(t|t-1)\right) \tag{2.61}$$

$$\hat{\mathbf{P}}(t|t) = \hat{\mathbf{P}}(t|t-1) - \hat{\mathbf{k}}(t)\mathbf{g}^T\hat{\mathbf{P}}(t|t-1) \tag{2.62}$$

Parameter Estimation Algorithm:

$$\widehat{\mathbf{R}}(t) = (1-\lambda)\left[\hat{\mathbf{x}}(t|t)\hat{\mathbf{x}}^T(t|t) + \hat{\mathbf{P}}(t|t)\right] + \lambda\widehat{\mathbf{R}}(t-1) \tag{2.63}$$

$$\hat{\boldsymbol{\alpha}}(t+1) = \hat{\boldsymbol{\alpha}}(t) - \gamma(t)\left[\hat{\mathbf{r}}_{21}(t) + \widehat{\mathbf{R}}_{22}(t)\hat{\boldsymbol{\alpha}}(t)\right] \tag{2.64}$$

$$\widehat{\sigma_u^2}(t+1) = \widehat{\sigma_u^2}(t) - \gamma(t)\left[\widehat{\sigma_u^2}(t) - \hat{r}_{11}(t) - \hat{\mathbf{r}}_{21}^T(t)\hat{\boldsymbol{\alpha}}(t)\right] \tag{2.65}$$

In Figure 2-2, we show once again the front end of the augmented ASR system, but now with detailed views of both the speech signal generation model and the sequential/adaptive speech enhancement algorithm. In particular, in the lower portion of this figure we show the interaction between the two main components (i.e., the signal estimation and parameter estimation components) of the enhancement algorithm.

Note that the estimates $\hat{\mathbf{x}}(t|t)$ and $\hat{\mathbf{P}}(t|t)$ from the current iteration of the Kalman equations are used to generate an updated parameter vector estimate $\hat{\boldsymbol{\theta}}(t+1)$. This parameter vector estimate is then incorporated in the next iteration of the Kalman equations, and so on until the entire input waveform has been processed.

Though not explicitly stated above, it is understood that the estimate of the current signal sample $s(t)$ is taken to be the first element of the vector $\hat{\mathbf{x}}(t|t)$ at each iteration. It is important to note, however, that the quantity $\hat{\mathbf{x}}(t|t)$ generated during the measurement update equations can no longer be interpreted as the conditional mean estimate suggested by Equation (2.16), because the underlying signal parameters are not known precisely at any time. By the same reasoning, the quantity $\mathbf{P}(t|t)$ is not, strictly speaking, the error covariance associated with $\hat{\mathbf{x}}(t|t)$; nonetheless, it can be considered an *estimate* of the error covariance, and we have employed the modified notation $\hat{\mathbf{P}}(t|t)$ to indicate this. Similarly, we have replaced $\mathbf{k}(t)$ by $\hat{\mathbf{k}}(t)$ and $\mathbf{F}$ by $\hat{\mathbf{F}}(t)$ to indicate that these quantities are not known precisely, but rather are evaluated at time t using relevant estimates that are available.

An example in which the above enhancement algorithm was applied to a noisy speech waveform is shown in Figure 2-3. The upper plot in this figure shows the original clean speech waveform representing the utterance "Aluminum silverware can often be flimsy." The middle plot shows a corrupted version of this waveform obtained by adding a stationary white-noise time series whose average power over the utterance is equal to that of the speech. (This level of corruption corresponds to a signal-to-noise ratio of 0 dB.) The lower plot shows the result of applying the sequential/adaptive algorithm to the corrupted speech waveform. Although the enhanced signal somewhat resembles the original signal for this processing example, much of the resemblance is due solely to the adaptive scaling applied through the Kalman gain. The mere scaling of the corrupted waveform is the dominant part of the enhancement in this example because, in such an extremely noisy environment, the temporal correlation structure of the underlying speech cannot be estimated with a significant degree of confidence.

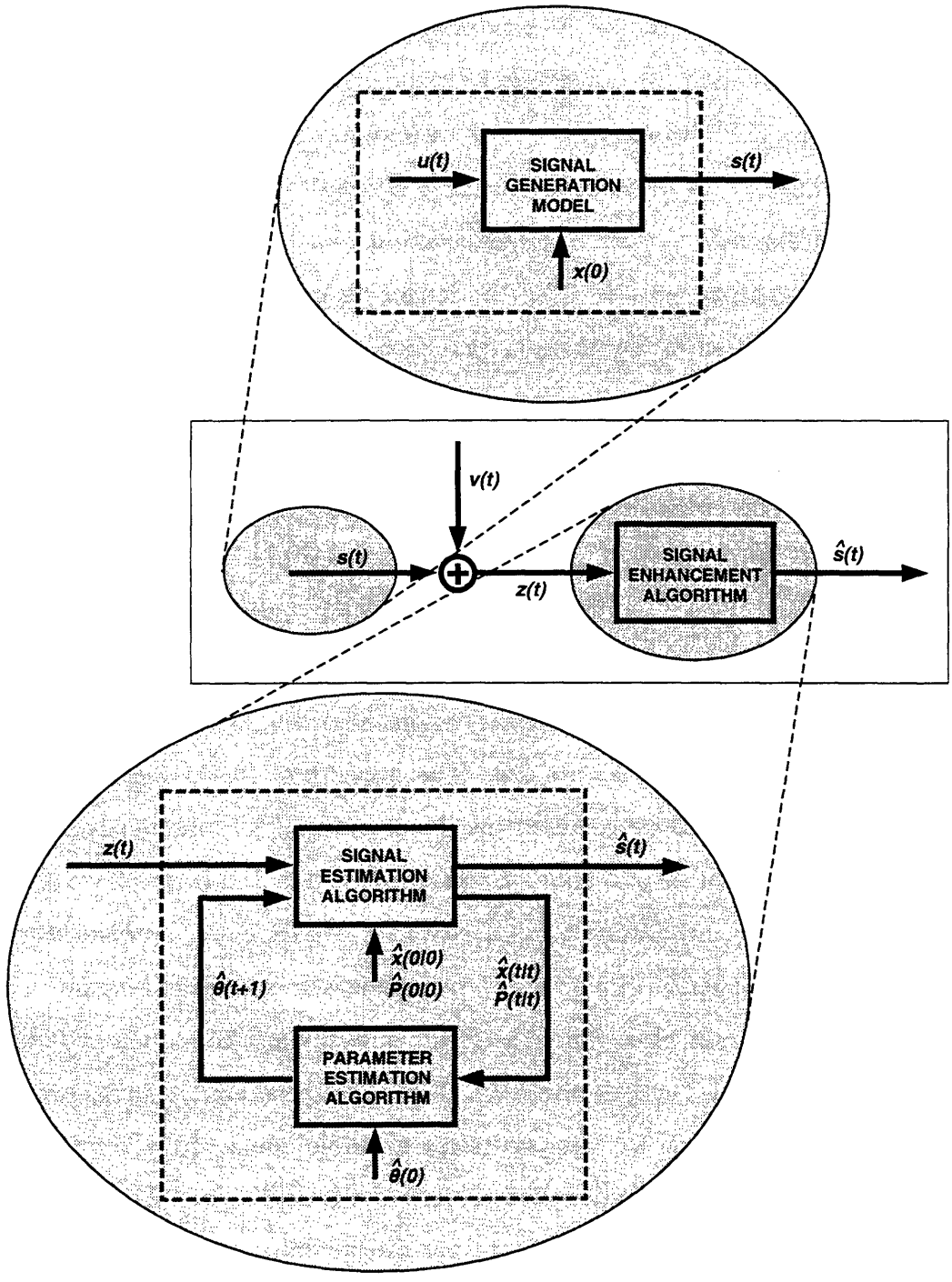


Figure 2-2: Block diagrams of the speech signal generation model and the sequential/adaptive speech enhancement algorithm.

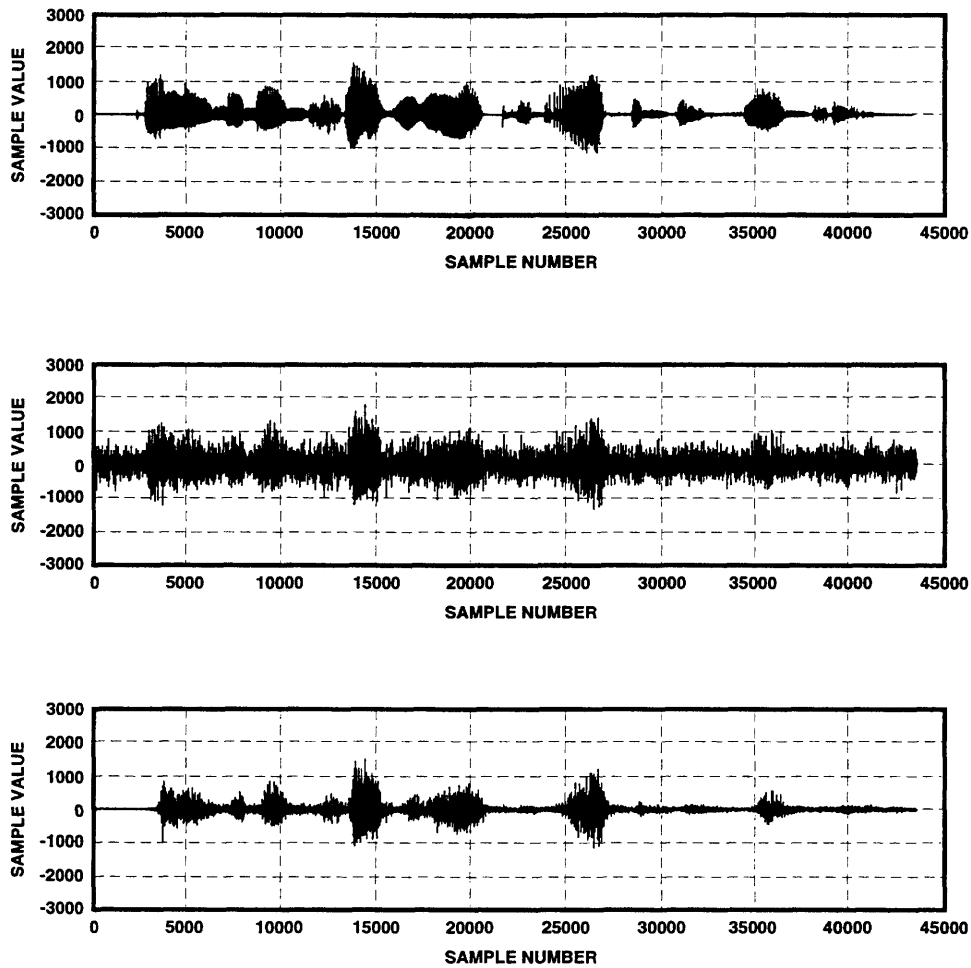


Figure 2-3: *Top to bottom:* Clean speech waveform representing the utterance "Aluminum silverware can often be flimsy"; speech waveform corrupted by white noise at an SNR of 0 dB; enhanced waveform produced by the sequential/adaptive algorithm.

2.3 Enhancing Speech in Additive Colored Noise

We now turn our attention to the case in which the speech signal of interest has been corrupted by temporally correlated or *colored* noise, rather than by white noise. In this case, since the corrupting noise is no longer memoryless, we change our measurement model of Equations (2.10) and (2.11) to account for the dynamics of the noise process as it evolves in time. We can make the necessary modification to our model by augmenting the state vector $\mathbf{x}(t)$ with a sufficient number of new state variables to accurately represent the autoregressive noise process specified by Equation (2.4). Thus, we now define $\mathbf{x}(t)$ as

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix}, \tag{2.66}$$

where $\mathbf{x}_1(t)$ represents the $(p+1) \times 1$ vector of signal states, given by

$$\mathbf{x}_1(t) = \begin{bmatrix} s(t) & s(t-1) & \cdots & s(t-p) \end{bmatrix}^T, \tag{2.67}$$

and $\mathbf{x}_2(t)$ represents the $q \times 1$ vector of noise states, given by

$$\mathbf{x}_2(t) = \begin{bmatrix} v(t) & v(t-1) & \cdots & v(t-q+1) \end{bmatrix}^T. \tag{2.68}$$

In the new dynamical equations, the signal portion of the state vector is, as before, driven by the white noise process $u(t)$; however, the noise portion is driven by the independent white noise process $w(t)$. Thus, in terms of the composite state vector definition given above, our new measurement model can be represented as

$$\begin{bmatrix} \mathbf{x}_1(t+1) \\ \mathbf{x}_2(t+1) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} + \begin{bmatrix} \mathbf{g}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{g}_2 \end{bmatrix} \begin{bmatrix} u(t) \\ w(t) \end{bmatrix} \tag{2.69}$$

$$z(t) = \begin{bmatrix} \mathbf{g}_1^T & \mathbf{g}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix}, \tag{2.70}$$

where $\mathbf{F}_1$ is the $(p+1)\times(p+1)$ matrix containing the autoregressive signal coefficients, given by

$$\mathbf{F}_1 = \begin{bmatrix} -\alpha_1 & -\alpha_2 & \cdots & \cdots & -\alpha_p & 0 \\ 1 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 1 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & 1 & 0 \end{bmatrix}, \tag{2.71}$$

$\mathbf{F}_2$ is the $q \times q$ matrix containing the autoregressive noise coefficients, given by

$$\mathbf{F}_2 = \begin{bmatrix} -\beta_1 & -\beta_2 & \cdots & \cdots & -\beta_{q-1} & -\beta_q \\ 1 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 1 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & 1 & 0 \end{bmatrix}, \tag{2.72}$$

$\mathbf{g}_1$ is the $(p+1) \times 1$ elementary vector given by

$$\mathbf{g}_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T, \tag{2.73}$$

and $\mathbf{g}_2$ is the $q \times 1$ elementary vector given by

$$\mathbf{g}_2 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T. \tag{2.74}$$

Note that because the signal and noise components of the current measurement are now both included in the state vector, there is no additional noise term appearing in the equation for $z(t)$.

Using the above colored-noise measurement model, we can, as before, develop an appropriate set of Kalman filtering equations to serve as the basis for a new signal

enhancement algorithm. To this end, we now introduce notation (analogous to that used in the previous section) for the Kalman filter that is based on the composite system model of Equations (2.69) and (2.70). In particular, we define the current estimates of the state vector components as

$$\hat{\mathbf{x}}_1(t|t) = E\{\mathbf{x}_1(t)|\, z(0), z(1), \ldots, z(t)\} \tag{2.75}$$

$$\hat{\mathbf{x}}_2(t|t) = E\{\mathbf{x}_2(t)|\, z(0), z(1), \ldots, z(t)\}, \tag{2.76}$$

and the error covariances associated with these estimates as

$$\mathbf{P}_{11}(t|t) = E\left\{(\hat{\mathbf{x}}_1(t|t) - \mathbf{x}_1(t))(\hat{\mathbf{x}}_1(t|t) - \mathbf{x}_1(t))^T \middle| z(0), z(1), \ldots, z(t)\right\} \tag{2.77}$$

$$\mathbf{P}_{22}(t|t) = E\left\{(\hat{\mathbf{x}}_2(t|t) - \mathbf{x}_2(t))(\hat{\mathbf{x}}_2(t|t) - \mathbf{x}_2(t))^T \middle| z(0), z(1), \ldots, z(t)\right\} \tag{2.78}$$

$$\mathbf{P}_{12}(t|t) = E\left\{(\hat{\mathbf{x}}_1(t|t) - \mathbf{x}_1(t))(\hat{\mathbf{x}}_2(t|t) - \mathbf{x}_2(t))^T \middle| z(0), z(1), \ldots, z(t)\right\} \tag{2.79}$$

$$\mathbf{P}_{21}(t|t) = E\left\{(\hat{\mathbf{x}}_2(t|t) - \mathbf{x}_2(t))(\hat{\mathbf{x}}_1(t|t) - \mathbf{x}_1(t))^T \middle| z(0), z(1), \ldots, z(t)\right\}. \tag{2.80}$$

With these definitions, it can be readily verified that the Kalman gain vector for the new measurement model is given (in partitioned form) by

$$\begin{bmatrix} \mathbf{k}_1(t) \\ \mathbf{k}_2(t) \end{bmatrix} = \frac{\begin{bmatrix} \mathbf{P}_{11}(t|t-1) & \mathbf{P}_{12}(t|t-1) \\ \mathbf{P}_{21}(t|t-1) & \mathbf{P}_{22}(t|t-1) \end{bmatrix} \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix}}{\begin{bmatrix} \mathbf{g}_1^T & \mathbf{g}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{P}_{11}(t|t-1) & \mathbf{P}_{12}(t|t-1) \\ \mathbf{P}_{21}(t|t-1) & \mathbf{P}_{22}(t|t-1) \end{bmatrix} \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix}}. \tag{2.81}$$

As we remarked in the previous section, the computation of each of the Kalman filtering variables defined above requires exact knowledge of all parameters in our measurement model. However, we shall assume once again that only the noise parameters are known, and that the speech parameters must be estimated concurrently with the speech signal values themselves. As a consequence of our incomplete knowledge of the measurement model, the above Kalman filtering variables cannot be computed exactly at any time, and hence must be approximated using parameter estimates made available at each iteration. Fortunately, by exploiting the partitioning of the Kalman

filtering variables into signal and noise components, we can compute the parameter estimates exactly as before. In particular, we simply replace the estimates $\hat{\mathbf{x}}(t|t)$ and $\hat{\mathbf{P}}(t|t)$ in Equation (2.63) by the newly defined estimates $\hat{\mathbf{x}}_1(t|t)$ and $\hat{\mathbf{P}}_{11}(t|t)$, which are associated strictly with the signal component of the state vector. Following the development in the previous section, we can then combine the signal estimation algorithm (based on the Kalman filtering equations) with the parameter estimation algorithm (based on the Yule-Walker equations) to produce a complete algorithm for enhancing speech in additive colored noise. The resulting sequential/adaptive algorithm is once again specified in two main stages as follows:

Signal Estimation Algorithm:

TIME UPDATE EQUATIONS

$$\begin{bmatrix} \hat{\mathbf{x}}_1(t|t-1) \\ \hat{\mathbf{x}}_2(t|t-1) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{F}}_1(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_1(t-1|t-1) \\ \hat{\mathbf{x}}_2(t-1|t-1) \end{bmatrix} \tag{2.82}$$

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{P}}_{11}(t|t-1) & \hat{\mathbf{P}}_{12}(t|t-1) \\ \hat{\mathbf{P}}_{21}(t|t-1) & \hat{\mathbf{P}}_{22}(t|t-1) \end{bmatrix} &= \begin{bmatrix} \hat{\mathbf{F}}_1(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{P}}_{11}(t-1|t-1) & \hat{\mathbf{P}}_{12}(t-1|t-1) \\ \hat{\mathbf{P}}_{21}(t-1|t-1) & \hat{\mathbf{P}}_{22}(t-1|t-1) \end{bmatrix} \\ &\quad \times \begin{bmatrix} \hat{\mathbf{F}}_1^T(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_2^T \end{bmatrix} + \begin{bmatrix} \widehat{\sigma_u^2}(t)\mathbf{g}_1\mathbf{g}_1^T & \mathbf{0} \\ \mathbf{0} & \sigma_w^2\mathbf{g}_2\mathbf{g}_2^T \end{bmatrix} \end{aligned} \tag{2.83}$$

MEASUREMENT UPDATE EQUATIONS

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{x}}_1(t|t) \\ \hat{\mathbf{x}}_2(t|t) \end{bmatrix} &= \begin{bmatrix} \hat{\mathbf{x}}_1(t|t-1) \\ \hat{\mathbf{x}}_2(t|t-1) \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{k}}_1(t) \\ \hat{\mathbf{k}}_2(t) \end{bmatrix} \\ &\quad \times \left(z(t) - \begin{bmatrix} \mathbf{g}_1^T & \mathbf{g}_2^T \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_1(t|t-1) \\ \hat{\mathbf{x}}_2(t|t-1) \end{bmatrix} \right) \end{aligned} \tag{2.84}$$

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{P}}_{11}(t|t) & \hat{\mathbf{P}}_{12}(t|t) \\ \hat{\mathbf{P}}_{21}(t|t) & \hat{\mathbf{P}}_{22}(t|t) \end{bmatrix} &= \begin{bmatrix} \hat{\mathbf{P}}_{11}(t|t-1) & \hat{\mathbf{P}}_{12}(t|t-1) \\ \hat{\mathbf{P}}_{21}(t|t-1) & \hat{\mathbf{P}}_{22}(t|t-1) \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{k}}_1(t) \\ \hat{\mathbf{k}}_2(t) \end{bmatrix} \begin{bmatrix} \mathbf{g}_1^T & \mathbf{g}_2^T \end{bmatrix} \\ &\quad \times \begin{bmatrix} \hat{\mathbf{P}}_{11}(t|t-1) & \hat{\mathbf{P}}_{12}(t|t-1) \\ \hat{\mathbf{P}}_{21}(t|t-1) & \hat{\mathbf{P}}_{22}(t|t-1) \end{bmatrix} \end{aligned} \tag{2.85}$$

Parameter Estimation Algorithm:

$$\widehat{\mathbf{R}}(t) = (1-\lambda)\left[\hat{\mathbf{x}}_1(t|t)\hat{\mathbf{x}}_1^T(t|t) + \hat{\mathbf{P}}_{11}(t|t)\right] + \lambda\widehat{\mathbf{R}}(t-1) \quad (2.86)$$

$$\hat{\boldsymbol{\alpha}}(t+1) = \hat{\boldsymbol{\alpha}}(t) - \gamma(t)\left[\hat{\mathbf{r}}_{21}(t) + \widehat{\mathbf{R}}_{22}(t)\hat{\boldsymbol{\alpha}}(t)\right] \quad (2.87)$$

$$\widehat{\sigma_u^2}(t+1) = \widehat{\sigma_u^2}(t) - \gamma(t)\left[\widehat{\sigma_u^2}(t) - \hat{r}_{11}(t) - \hat{\mathbf{r}}_{21}^T(t)\hat{\boldsymbol{\alpha}}(t)\right] \quad (2.88)$$

Once again, though not explicitly stated in the algorithm above, it is understood that the estimate of the current signal sample $s(t)$ is taken to be the first element of the vector $\hat{\mathbf{x}}_1(t|t)$ at each iteration.

An example in which the above algorithm was applied to a noisy speech waveform is presented in Figure 2-4. The three plots appearing in this figure are analogous to those shown in the earlier white-noise example of Figure 2-3. As before, the upper plot shows the original clean speech signal representing the utterance "Aluminum silverware can often be flimsy." The middle plot shows a corrupted version of this waveform obtained by adding colored noise at a signal-to-noise ratio of 0 dB. (In this case, the corrupting noise was chosen to be a fifth-order autoregressive process with parameter values $\beta_1 = -2.542$, $\beta_2 = 2.281$, $\beta_3 = -1.058$, $\beta_4 = 0.518$, and $\beta_5 = -0.195$.) The lower plot in the figure shows the result of applying the above sequential/adaptive enhancement algorithm to the corrupted waveform. In this colored-noise example, the enhanced waveform is of slightly better quality than the corresponding waveform shown in the earlier white-noise example, even though the signal-to-noise ratio was the same in both cases. This difference exists because, whereas white noise is completely unpredictable from sample to sample, at least some portion of temporally correlated noise added at a future sample *is* predictable, and hence is removable. The general notion of comparing the quality of output signals produced by the enhancement algorithms derived in this chapter will be made more precise in Chapter 4 as part of an analysis of experimental speech recognition results.

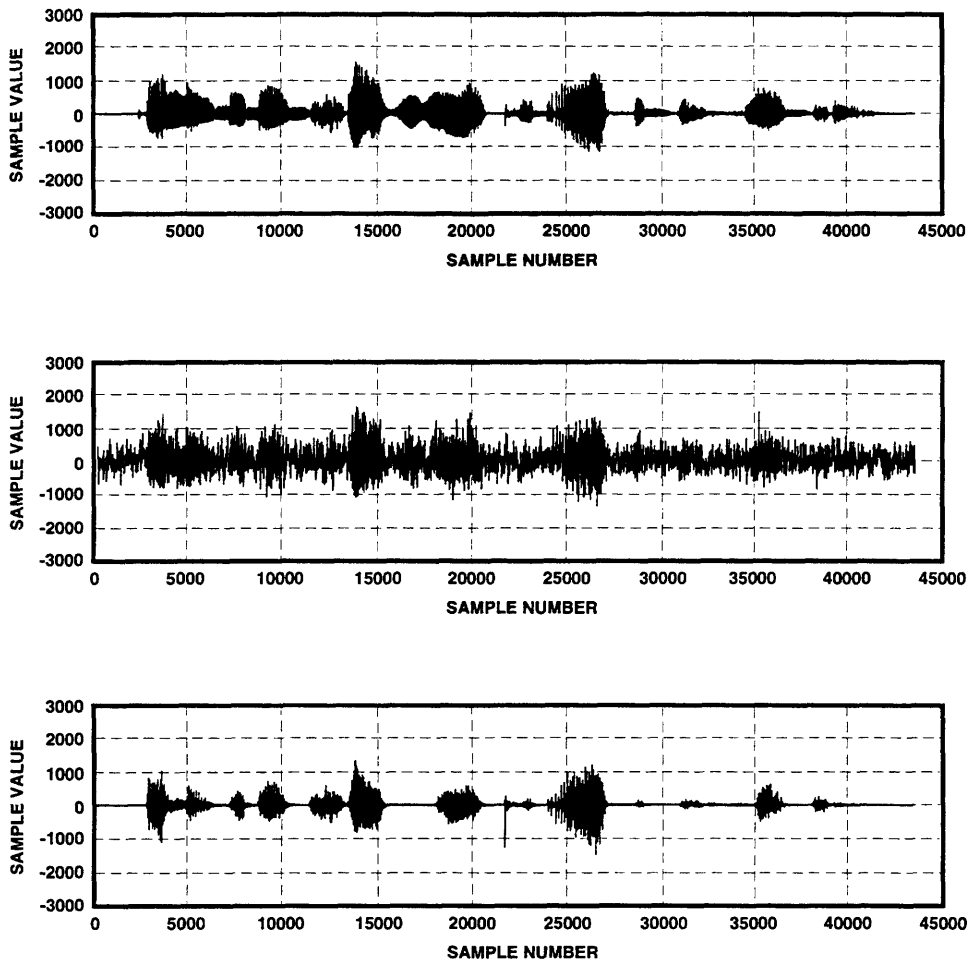


Figure 2-4: *Top to bottom:* Clean speech waveform representing the utterance "Aluminum silverware can often be flimsy"; speech waveform corrupted by colored noise at an SNR of 0 dB; enhanced waveform produced by the sequential/adaptive algorithm.

Chapter 3

Experiment Methodology and Design

Thus far, we have proposed an approach for increasing the robustness of an automatic speech recognition system whereby a front-end signal enhancement component serves to prefilter the noisy input speech before it reaches the recognizer. In addition, we have developed two sequential/adaptive algorithms for the specific task of enhancing speech in two different kinds of additive noise. In this chapter, we describe how the enhancement algorithms were incorporated in a set of baseline speech recognition experiments. These experiments were designed to evaluate the performance of an ASR system, with and without the use of a front-end speech enhancement component, in a variety of environment conditions. Recognition accuracy was tested at different linguistic levels (specifically, the *phone* and *word* levels), and at different input SNR values for both the white-noise and colored-noise cases. We provide a general overview of each experiment that was conducted, and then describe in detail the various components that comprised each experiment.

3.1 Overview of the Speech Recognition Experiments

To determine how the prefiltering of noisy speech affects overall recognition performance, we must first measure the baseline performance of the recognizer operating alone in the presence of additive noise. A standard way of gauging performance under such conditions is to have the recognizer process a test collection of speech signals repeatedly in a sequence of experimental trials, while increasing the level of corruption applied to the signals on each successive trial.

The structure of the baseline performance evaluation experiment used for this thesis is presented in the form of a block diagram in Figure 3-1. Note from the figure that two data bases are used in the baseline experiment: (1) a speech data base, which contains a large number of high-quality, prerecorded spoken sentences (together with a corresponding set of sentence transcriptions), and (2) a noise data base, which contains computer-generated white-noise and colored-noise waveforms (together with parameter values that characterize the noise). At the start of an experimental trial, waveforms $s(t)$ and $v(t)$ are extracted from the speech and noise data bases, respectively. The noise waveform is then scaled such that the average energy in $s(t)$ and the average energy in $v(t)$ are in the appropriate proportions for the signal-to-noise ratio (SNR) currently being tested (i.e., if the required SNR is equal to ρ (measured in dB), then $v(t)$ is scaled such that

$$\rho = 10 \log_{10} \left(\frac{\sum_{t=0}^{N-1} s^2(t)}{\sum_{t=0}^{N-1} v^2(t)} \right), \tag{3.1}$$

where N is the length in samples of both $s(t)$ and $v(t)$). Once $v(t)$ is brought to the required energy level, it is added to $s(t)$ to produce the corrupted speech waveform $z(t) = s(t) + v(t)$. The speech recognition system accepts this corrupted waveform as input and subsequently generates an estimate, in the form of a text string $\hat{T}$, of the utterance represented by $s(t)$. Finally, this hypothesized utterance transcription $\hat{T}$ is systematically compared, by means of a specially designed string alignment algorithm,

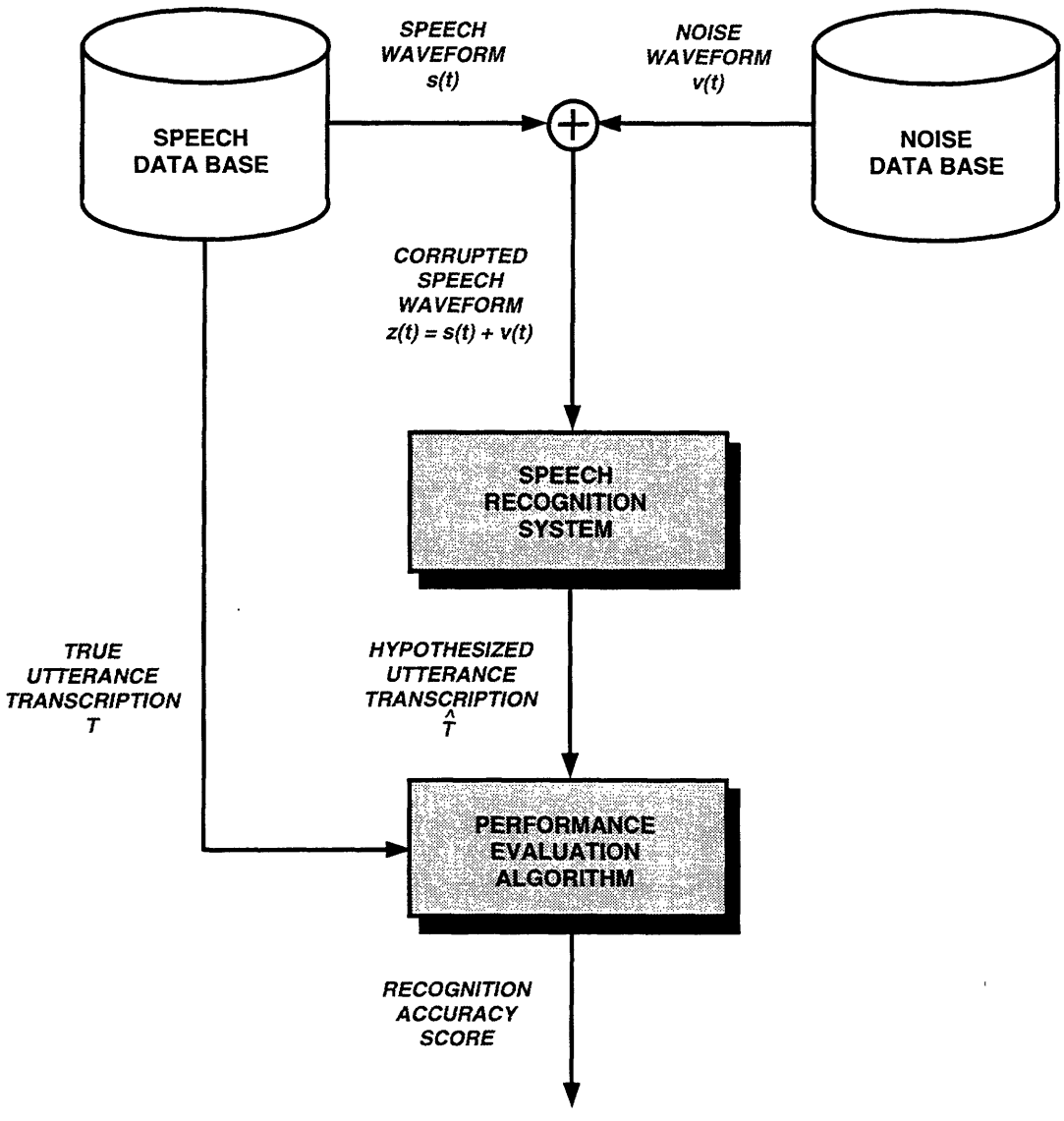


Figure 3-1: Block diagram of the experiment for evaluating baseline speech recognition performance in the presence of additive noise.

to the true utterance transcription T, which accompanies $s(t)$ in the speech data base. The comparison of the text strings T and $\hat{T}$ results in a numerical score between 0 and 1 that indicates the accuracy of the recognizer output.

Of course, to obtain a reliable measure of speech recognition performance at a prespecified noise level, we must perform a suitably large number of experimental trials such as the one described above, using a variety of speech waveforms from the data base. Once baseline performance has been established for a given noise level, we can then measure the corresponding change in performance that results from prefiltering the noisy speech before recognition is attempted. The structure of the modified experiment used for this purpose is shown in Figure 3-2. Note that the new experiment is virtually identical to the baseline experiment, except that it includes as its first stage of processing a signal enhancement algorithm, which is furnished in advance with the values of all relevant parameters that characterize the corrupting noise waveform $v(t)$.

Using the basic experiment configurations depicted in Figures 3-1 and 3-2, it is possible to generate many different kinds of performance results. For this thesis, a total of three experimental factors were varied within each configuration. These factors are:

(1) *The type of noise added to the speech.* As suggested in Chapter 2, we added either white Gaussian noise or colored Gaussian noise (generated with an autoregressive model) to each speech waveform. For the configuration shown in Figure 3-2, the appropriate signal enhancement algorithm from Chapter 2 was used.

(2) *The scaling of the noise relative to the speech.* To generate a complete performance curve, we conducted each experiment at many different noise levels. Recognition performance was evaluated at SNR values ranging from -10 dB to 30 dB in increments of 10 dB; performance was also evaluated at an SNR of ∞ dB by using only the clean speech signals with no added noise.

(3) *The function performed by the speech recognition system.* The recognizer was

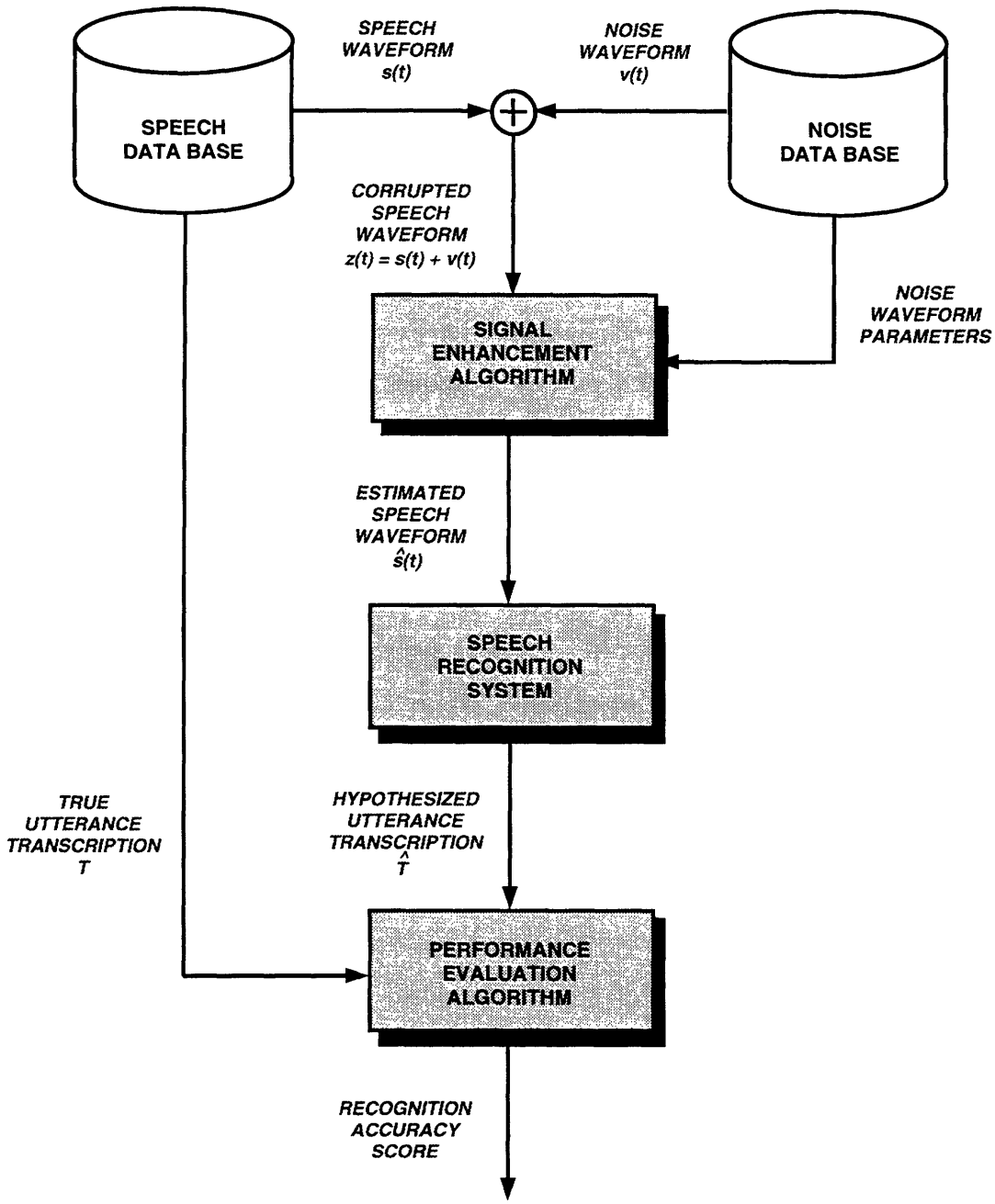


Figure 3-2: Block diagram of the experiment for evaluating performance of the two-stage system (speech enhancement followed by speech recognition) in the presence of additive noise.

set up to perform either *phone classification* or *word recognition.* In phone classification mode, the recognizer is supplied with precise time boundaries marking the beginning and end of all phonetic units (or phones) comprising the underlying utterance; using this information, which is extracted from the speech data base, the recognizer attempts to correctly identify each isolated phone represented in the noisy input waveform. In word recognition mode, the recognizer is given no auxiliary information about the underlying utterance; nonetheless, it attempts to determine the entire sequence of words represented in the input waveform. (Of course, different performance evaluation algorithms are required for the phone classification and word recognition functions.)

The performance results obtained by varying each of the above factors are presented and analyzed in Chapter 4. In the next section of this chapter, we give details of each of the experimental components shown in Figures 3-1 and 3-2, namely: (1) the speech data base, (2) the noise data base, (3) the signal enhancement algorithm, (4) the speech recognition system, and (5) the performance evaluation algorithm.

3.2 Elements of the Speech Recognition Experiments

3.2.1 The Speech Data Base

The speech waveforms used in each experiment described above were drawn from a standard data base known as TIMIT, which was produced jointly by MIT, SRI International, and Texas Instruments. The sentences comprising the TIMIT speech corpus were designed specifically to aid in the development and evaluation of phonetically-based automatic speech recognition systems [39].

The TIMIT utterances were recorded under very favorable acoustic conditions, and are therefore virtually free of distortion. Each speaker who participated in the recordings was placed in an anechoic room, equipped with a close-talking, noise-cancelling, headset-boom microphone, and instructed to read aloud, in a natural

conversational voice, a sequence of preselected sentences. The spoken sentences were stored in digital form at a sampling rate of 16kHz, with each speech sample quantized to 16 bits [16].

A total of 630 speakers participated in the recordings, each contributing 10 sentences to the 6300-sentence TIMIT data base. Each speaker was associated with one of eight major dialect categories, labeled as: (1) New England, (2) Northern, (3) North Midland, (4) South Midland, (5) Southern, (6) New York City, (7) Western, and (8) Army Brat. In general, the dialect category indicates the geographical region within the United States where the speaker lived during his childhood years. The Army Brat label was associated with speakers who moved frequently throughout the United States during their childhood years.

The text material in the TIMIT corpus consists of three kinds of specially designed speaker prompts, identified in the data base as SA, SX, and SI sentences. The SA sentences were designed to reveal the variations in phones and phone pairs that exist because of dialectal differences among speakers. Only 2 SA sentences were constructed, and both of these were read by all 630 speakers. The phonetically compact SX sentences were designed to provide efficient and thorough coverage of phone pairs considered to be of particular interest for the recognition problem. A total of 450 SX sentences were constructed; each speaker read 5 of these sentences, and each sentence was read by 7 different speakers. The phonetically diverse SI sentences were selected from existing text sources to provide a greater variety of phonetic contexts for study. A total of 1890 SI sentences were constructed; each speaker read 3 of these sentences, and each sentence was read by only one speaker.

Each utterance in the TIMIT data base is associated with four descriptive files: (1) a *waveform file*, which contains the digitized samples of recorded speech, (2) an *orthographic transcription file*, which contains the precise text of the spoken sentence, (3) a *word transcription file*, which contains a segmentation of the utterance into its component words, with beginning and ending waveform sample numbers provided for each word, and (4) a *phonetic transcription file*, which contains a segmentation of the utterance into its component phones, with beginning and ending waveform sample

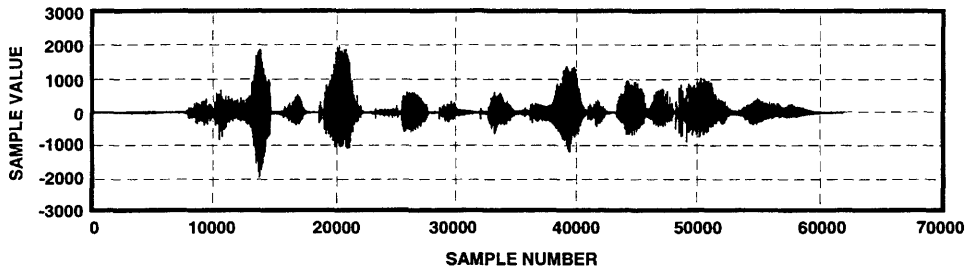
numbers provided for each phone. In Figure 3-3, we show the four descriptive files associated with the TIMIT utterance "She had your dark suit in greasy wash water all year."

An exhaustive list of phones used in the TIMIT phonetic transcriptions is presented in Table 3.1. For each phone appearing in this table, we give the corresponding symbol as well as an example of an occurrence of the phone (highlighted in bold letters) in a common English word. For convenience, the phones have been sorted into eight major categories, including stops, affricates, fricatives, nasals, semivowels, diphthongs, and miscellaneous others.

The TIMIT phones are defined very specifically, and thus the inventory of phonetic symbols associated with the TIMIT data base is slightly larger than a typical phonetic dictionary. For example, in the TIMIT dictionary the stops are decomposed into an initial closure interval (e.g., `/bcl/`) and a subsequent release interval (e.g., `/b/`). In addition, there exist a number of variations of traditional phones (commonly known as *allophones*), such as the flap `/dx/` (an allophone of `/t/`), the nasal flap `/nx/` (an allophone of `/n/`), the voiced-h `/hv/` (an allophone of `/h/`, typically occurring between two voiced phones), the fronted-u `/ux/` (an allophone of `/uw/`), and the devoiced-schwa `/ax-h/` (an allophone of `/ax/`, typically occurring for reduced vowels between two unvoiced consonants). Finally, there are miscellaneous symbols such as `/h#/` (used to represent the non-speech events at the beginning and end of a waveform), `/pau/` (used to represent a pause within the utterance), and `/epi/` (used to represent an epenthetic closure, which often occurs between a fricative and a semivowel or nasal, as in the first part of the word "slow").

3.2.2 The Noise Data Base

As mentioned previously, two basic types of noise were added to the speech in separate experiments, namely: (1) temporally uncorrelated (or *white*) Gaussian noise, and (2) temporally correlated (or *colored*) Gaussian noise. All of the noise waveforms used in the speech recognition experiments were created by computer with standard random number generation algorithms [29]. The model for the computer-generated colored



ORTHOGRAPHIC TRANSCRIPTION FILE
0 61748 She had your dark suit in greasy wash water all year.

WORD TRANSCRIPTION FILE	
7470 11362 she	30960 36971 greasy
11362 16000 had	36971 42290 wash
15420 17503 your	43120 47480 water
17503 23360 dark	49021 52184 all
23360 28360 suit	52184 58840 year
28360 30960 in	

PHONETIC TRANSCRIPTION FILE			
0 7470 h#	21053 22200 r	32550 33253 r	46040 47480 axr
7470 9840 sh	22200 22740 kcl	33253 34660 iy	47480 49021 q
9840 11362 iy	22740 23360 k	34660 35890 z	49021 51348 ao
11362 12908 hv	23360 25315 s	35890 36971 iy	51348 52184 l
12908 14760 ae	25315 27643 ux	36971 38391 w	52184 54147 y
14760 15420 dcl	27643 28360 tcl	38391 40690 ao	54147 56654 ih
15420 16000 jh	28360 29272 q	40690 42290 sh	56654 58840 axr
16000 17503 axr	29272 29932 ih	42290 43120 epi	58840 61680 h#
17503 18540 dcl	29932 30960 n	43120 43906 w	
18540 18950 d	30960 31870 gcl	43906 45480 ao	
18950 21053 aa	31870 32550 g	45480 46040 dx	

Figure 3-3: *Top to bottom:* Sample time-domain waveform of an utterance from the TIMIT data base; orthographic transcription file associated with the utterance; word transcription file; phonetic transcription file.

STOPS		AFFRICATES		FRICATIVES		NASALS	
PHONE	EXAMPLE	PHONE	EXAMPLE	PHONE	EXAMPLE	PHONE	EXAMPLE
/b/	bee	/jh/	joke	/s/	sea	/m/	mom
/d/	day	/ch/	choke	/sh/	she	/n/	none
/g/	go			/z/	zoo	/ng/	sing
/p/	pay			/zh/	measure	/em/	bottom
/t/	tea			/f/	fan	/en/	button
/k/	key			/th/	thin	/eng/	washington
/dx/	butter			/v/	van	/nx/	runner
				/dh/	then		
SEMIVOWELS		VOWELS		DIPHTHONGS		OTHERS	
PHONE	EXAMPLE	PHONE	EXAMPLE	PHONE	EXAMPLE	PHONE	DESCRIPTION
/l/	low	/iy/	beet	/ey/	bait	/q/	*glottal stop*
/r/	row	/ih/	bit	/aw/	bout	/bcl/	*closure before* /b/
/w/	wet	/eh/	bet	/ay/	bite	/dcl/	*closure before* /d/
/y/	yet	/ah/	but	/oy/	boy	/gcl/	*closure before* /g/
/hh/	hay	/ae/	bat	/ow/	boat	/pcl/	*closure before* /p/
/hv/	ahead	/aa/	cot	/ux/	beauty	/tcl/	*closure before* /t/
/el/	bottle	/ao/	bought			/kcl/	*closure before* /k/
		/uh/	book			/epi/	*epenthetic closure*
		/uw/	boot			/pau/	*pause*
		/er/	bird			/h#/	*begin/end marker*
		/ax/	again				
		/ix/	debit				
		/axr/	diner				
		/ax-h/	sustain				

Table 3.1: Categorized list of phonetic symbols associated with the TIMIT data base.

noise was based on a segment of actual car noise recorded from the interior of a moving automobile. The original car noise was not used in the speech recognition experiments because it was deemed to have too little spectral overlap with speech data to pose an interesting enhancement problem. Thus, the recorded car noise was first downsampled to produce sufficient spectral overlap with the speech, and was then modeled as a fifth-order autoregressive process. The autoregressive parameters estimated from the modified car noise were $\beta_1 = -2.542$, $\beta_2 = 2.281$, $\beta_3 = -1.058$, $\beta_4 = 0.518$, and $\beta_5 = -0.195$. These parameter values were then used to generate synthetic car noise on a computer. In Figure 3-4, we show the true power spectral densities for both the white and colored noise processes used in the speech recognition experiments.

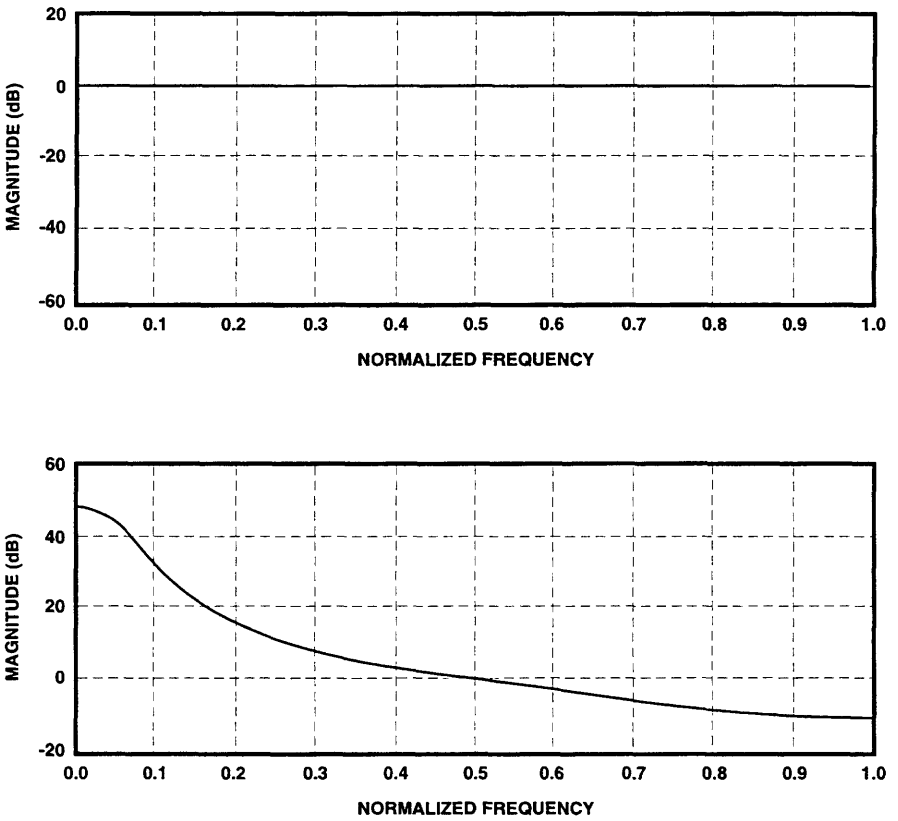


Figure 3-4: *Top to bottom:* Power spectral density of the white noise process used in the speech recognition experiments; power spectral density of the colored noise process used. (The horizontal axis represents frequencies ranging from 0 to 8 kHz.)

3.2.3 The Signal Enhancement Algorithm

To conduct the experiments that required filtering of the corrupted speech prior to the recognition stage, we implemented the adaptive signal processing algorithms derived in Chapter 2. For each of these experiments, in both the white-noise and colored-noise cases, the noise parameters were assumed known and constant over the entire length of the speech signal. The true values of these parameters, taken directly from the noise data base, were supplied as inputs to the signal enhancement algorithms together with the corrupted speech waveform.

In each experiment, the underlying speech was modeled locally as a fifth-order autoregressive process. To obtain suitable estimates of the time-varying autoregressive parameters, it was necessary to specify appropriate values for the forgetting factor λ and the step-size parameter $\gamma(t)$ appearing in the latter set of computations of the adaptive algorithm (refer to Equations (2.63) through (2.65)). The parameter λ is the exponential weighting factor in the recursive updating algorithm for the signal correlation matrix $\mathbf{R}$, and therefore determines the time constant (or, more suggestively, the *memory length*) associated with the algorithm. In this context, the memory length is the number of samples into the past at which measurements are weighted at $1/e$ or less of the weight given to the most recent measurement; this length, which we denote by L, is defined implicitly through the equation

$$\lambda^L = 1/e. \tag{3.2}$$

Since the forgetting factor λ is typically chosen to be very close to 1, we can use the approximate expression for L given by

$$L = \frac{-1}{\ln \lambda} \approx \frac{1}{1-\lambda}. \tag{3.3}$$

The speech enhancement algorithm should have enough memory to allow for the computation of an accurate estimate, but not so much that the estimate includes measurements that do not reflect the current properties of the speech. It was deter-

mined that a memory length of roughly 100 samples was suitable for the recognition experiments; this length corresponds to a forgetting factor of $\lambda = 0.99$.

The selection of the step-size parameter $\gamma(t)$ involves a similar trade-off between the ability of the algorithm to track changes in the structure of the underlying speech and the sensitivity of the algorithm to disturbances in the measurement data. After some experimentation, the step size was fixed at the value $\gamma(t) = 0.01$ for all t.

3.2.4 The Speech Recognition System

All of the experiments described earlier were performed using a spoken language understanding system called SUMMIT, which was developed by Zue and other researchers from the Spoken Language Systems Group at the MIT Laboratory for Computer Science [45, 46, 47]. The SUMMIT recognizer is a phonetically-based, speaker-independent, continuous-speech system that can be configured to perform a variety of recognition tasks and trained automatically on large sets of prerecorded speech data.

A block diagram of the SUMMIT system is shown in Figure 3-5. From this figure, we see that the input to the system is a waveform representing a spoken sentence, and the output is a hypothesized transcription of the sentence. Note that the three main components of SUMMIT incorporate three distinct levels of linguistic knowledge into the utterance decoding procedure: phonetic, lexical, and grammatical.

The phonetic recognition subsystem of SUMMIT is itself composed of three stages, namely: (1) signal representation, (2) acoustic segmentation, and (3) phonetic classification. In the first stage of this subsystem, the raw input speech waveform is nonlinearly transformed into a more convenient signal representation based on a model of the human peripheral auditory system [36]; this transformation simultaneously enhances important acoustic landmarks in the time-domain signal and sharpens spectral peaks that are useful in phonetic classification. In the second stage, the output of the auditory model is used to divide the utterance into time segments that share common acoustic characteristics; this segmentation procedure is carried out at many different temporal resolution levels in order to capture both the slow and sudden transitions in the acoustic structure of the speech signal. In the final stage, a feature vector

is computed for each acoustic segment, and the segment then becomes associated, by means of conventional statistical classification procedures, with the label of the phoneme having the greatest likelihood. The output of the phonetic recognition subsystem is a forward-directed network that connects the labeled acoustic segments at various resolution levels from the beginning of the utterance to the end. Each arc in this acoustic-phonetic network points to a single acoustic segment, and is weighted by the likelihood of the associated phonetic label.

The lexical component of SUMMIT maintains the extensive inventory of words to be recognized by the system, and accounts for the many possible pronunciations of each word. Using the basic phonetic representation of each word in the vocabulary, the lexical expansion subsystem produces a pronunciation network for the word. The rules employed for this lexical expansion are defined in advance, and they attempt to account for both intra-word and inter-word phonological effects that occur in natural speech. Each node in the pronunciation network corresponds to a phoneme within the word. Pairs of these nodes are connected by arcs, and each arc is assigned a weight indicating the probability that the two associated phonemes occur in succession when the word is pronounced. A path through the network represents a particular pronunciation of the word.

The pronunciation networks for all words in the lexicon are connected together by introducing new arcs from the terminal node of each word to the initial node of every other word. The resulting lexical network allows for all sequences of words that might occur in a sentence. When this extensive network is constructed, the high-level linguistic component of SUMMIT imposes local grammatical constraints on sentence structure by permitting connections to be made only between certain words, and by weighting these connections with the probabilities of the corresponding word pairs [37]. Once the lexical network has been constructed, the utterance can be decoded with the aid of a dynamic programming algorithm. The decoding operation is essentially a search for the best match between a directed path in the lexical network (representing a particular sequence of phonemes that comprise a string of words from the lexicon) and a directed path in the acoustic-phonetic network. Because

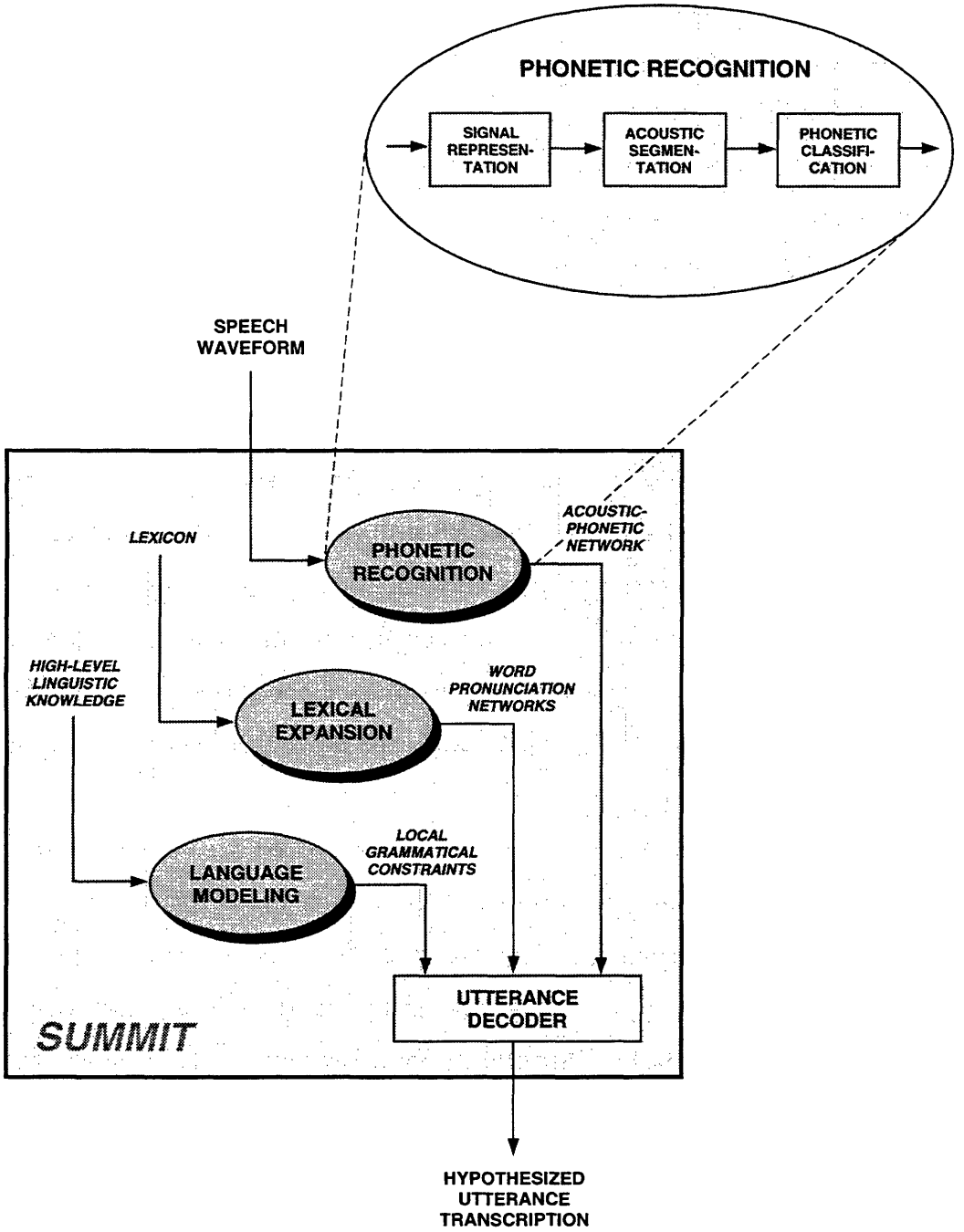


Figure 3-5: Block diagram showing the primary components of the SUMMIT spoken language understanding system.

the acoustic segmentation of the speech waveform may be inaccurate, and because the assignment of phonetic labels for the segments may be incorrect, allowances are made during the search for the insertion and deletion of segments in the acoustic-phonetic network.

For each of the phone classification and word recognition experiments conducted for this thesis, the SUMMIT system was trained using a set of 800 utterances extracted from the TIMIT data base. This training set consisted of 8 utterances (representing the SX and SI sentences) for each of 100 speakers selected from various dialect categories. In addition, an independent testing set of 200 utterances was used in the experiments. The testing set consisted of 4 randomly chosen SX and SI sentences for each of 50 speakers. For the word recognition experiments, the SUMMIT system operated with a 6000-word vocabulary.

3.2.5 The Performance Evaluation Algorithm

Once a hypothesized transcription of an utterance is generated by the speech recognition system, it must be scored in a meaningful way through a comparison with the actual utterance transcription so that the performance of the recognizer can be evaluated. The scoring procedure that is applied in a given instance depends on the nature of the recognition task being tested. Within the scope of this thesis, two basic kinds of experiments were performed: (1) *phone classification*, in which the recognizer attempts to determine the identity of a phone located between two prespecified time boundaries in a waveform, and (2) *word recognition*, in which the recognizer attempts to determine the entire sequence of words in an utterance in the absence of any boundary information. In this section, we discuss standard methods of evaluating the output of the speech recognition system for each kind of experiment that was conducted.

For the phone classification experiments, the scoring procedure is relatively straightforward. First, before any evaluation is done, certain "equivalence" classes of phones are defined according to the type of recognition task being tested. Typically, multiple phones are assigned to the same class if they are similar enough that distinguishing

between them would have a negligible effect on ultimate recognition performance. For all of the phone classification experiments conducted in the present work, each phone associated with the TIMIT corpus was assigned to one of the 39 classes displayed in Table 3.2.

Once these classes have been specified, each phone in both the hypothesized transcription and the reference transcription can be given a class label. (In our case, this label is simply an integer from 1 to 39.) After all phones in the time-aligned utterance transcriptions have been labeled, a count is made of those phones in the hypothesized transcription whose class labels match the labels of corresponding phones in the reference transcription. The accuracy score is then simply the number of matches divided by the total number of phones in the utterance. To redefine this score in terms of the error rate, let us suppose that after processing a particular utterance we have N_{ERR} mismatches out of a total of N_{TOT} phones. Then the *phone classification accuracy* (PCA) is defined as

$$\mathrm{PCA} = \frac{N_{\mathrm{TOT}} - N_{\mathrm{ERR}}}{N_{\mathrm{TOT}}} \tag{3.4}$$

If multiple utterances are processed in a batch, the individual counts N_{ERR} and N_{TOT} are simply incremented with each new utterance, and phone classification accuracy is then computed using the above formula with the final count totals.

The scoring procedure applied to the word recognition experiments is much more complicated, since the recognizer does not have access *a priori* to any phone or word boundaries in the waveform, and hence must estimate where each word begins and ends in addition to estimating the identity of the word itself. One result of having the recognizer perform this more complex task is that the hypothesized transcription may contain more or fewer words than the reference transcription; moreover, although many of the words in these two transcriptions may match, the entire transcriptions themselves may not be well-aligned (i.e., when a match occurs across transcriptions, the matching elements may not be positioned the same number of words from the beginning in their respective strings).

CLASS	PHONES	CLASS	PHONES	CLASS	PHONES
1	/uw/ /ux/	15	/r/	29	/f/
2	/uh/	16	/l/ /el/	30	/jh/
3	/ah/ /ax/ /ax-h/	17	/w/	31	/ch/
4	/aa/ /ao/	18	/m/ /em/	32	/b/
5	/ae/	19	/n/ /en/ /nx/	33	/d/
6	/eh/	20	/ng/ /eng/	34	/g/
7	/ih/ /ix/	21	/dx/	35	/p/
8	/ey/	22	/v/	36	/t/
9	/iy/ /y/	23	/th/	37	/k/
10	/ay/	24	/dh/	38	/h#/ /pau/
11	/ow/	25	/hh/ /hv/	39	/bcl/ /dcl/
12	/aw/	26	/z/		/gcl/ /kcl/
13	/oy/	27	/s/		/pcl/ /tcl/
14	/er/ /axr/	28	/sh/ /zh/		/epi/ /q/

Table 3.2: List of phonetic equivalence classes used in the phone classification experiments.

This general problem of string alignment is easily demonstrated with an example. Let us suppose that the waveform processed by the speech recognizer actually represents the utterance "This old key does not work," but that the hypothesized utterance generated by the recognizer is "His key does the top work." With no preprocessing of these two strings, we might align them word for word in the following way:

REFERENCE STRING: THIS OLD KEY DOES NOT WORK

HYPOTHESIZED STRING: HIS KEY DOES THE TOP WORK

Upon examining these transcriptions, we can see that the recognizer has successfully identified the words "key," "does," and "work," which appear in the original utterance. On the other hand, it has probably misidentified the word "this" as "his" and the word "not" as "top," and has also inadvertently missed the word "old" in the original utterance and erroneously added the word "the" in its own hypothesized utterance.

If, for the above example, we wish to compute an error rate for the output of the recognizer by directly counting mismatches between the ith word in the reference

string and the ith word in the hypothesized string, the above alignment is clearly inappropriate, since the only valid match using this method occurs at the final word "work." We can arrive at a better method for computing the error rate by observing that there are three basic types of errors introduced by the recognizer into the hypothesized transcription, namely: (1) *deletion errors*, which occur when words in the original utterance are not detected, (2) *insertion errors*, which occur when words not in the original utterance are erroneously added, and (3) *substitution errors*, which occur when words in the original utterance are incorrectly identified. Using this observation, we can modify the reference and hypothesis strings slightly to produce the following more appropriate alignment:

```
REFERENCE STRING:   THIS  OLD  KEY  DOES  ***  NOT  WORK
HYPOTHESIZED STRING: HIS  ***  KEY  DOES  THE  TOP  WORK
```

Now, the alignment of the words "this" and "his" as well as the words "not" and "top" correctly indicates the occurrence of two substitution errors. In addition, the alignment of the word "old" and the placeholder token "***" indicates the occurrence of a deletion error; similarly, the alignment of placeholder token "***" and the word "the" indicates the occurrence of an insertion error. Thus, for this example, since there are two substitution errors, one deletion error, and one insertion error out of a total of seven words in the modified hypothesized string, a reasonable estimate of the error rate is 4/7.

The foregoing example of error rate estimation illustrates the main idea underlying the current standard procedure for evaluating the performance of speech recognition systems. In this example, however, we arrived at the "optimal" string alignment simply by comparing the reference and hypothesized strings and subsequently inferring the kinds of errors produced at key word locations. Thus, although the example affords valuable intuition for addressing the problem of how two strings should be aligned, it does not provide us with a concrete algorithm for aligning strings. In the remaining portion of this section, we examine a procedure for alignment-based error rate estimation that has been widely distributed to speech recognition researchers by

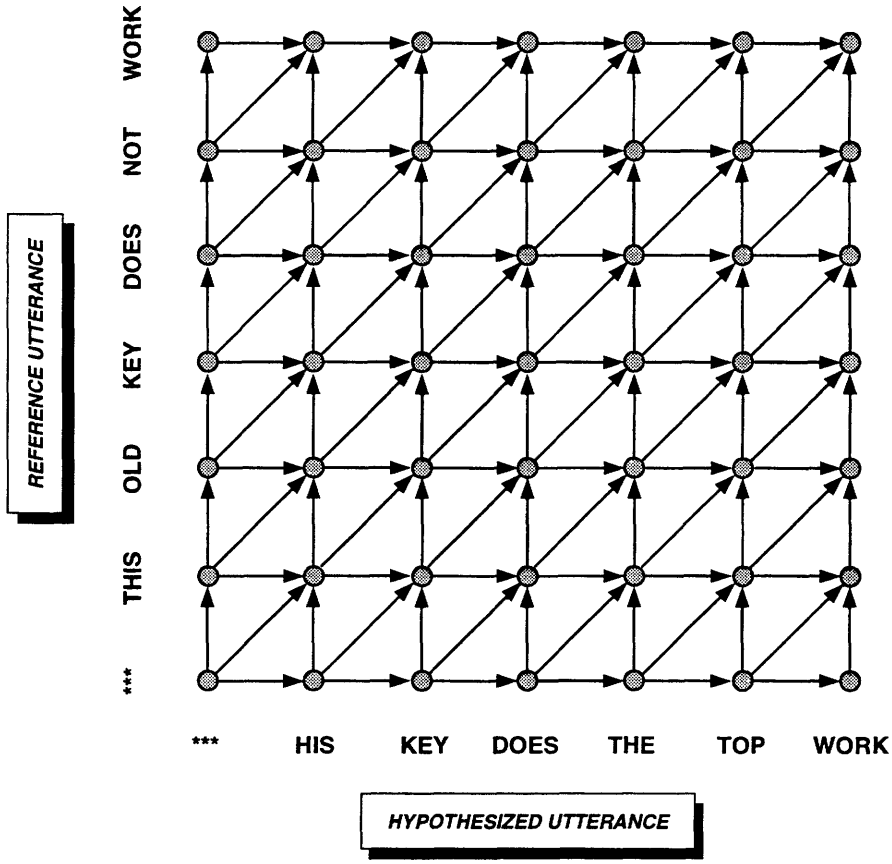


Figure 3-6: Directed graph used by the dynamic programming algorithm to align the hypothesized and reference utterances.

the National Institute of Standards and Technology (NIST), and has been employed in the DARPA Speech Recognition Program [26]. It is important to note, however, that although this procedure has become a *de facto* standard in the field of automatic speech recognition, more advanced methods for performing text alignment are currently available [27, 28].

In the NIST procedure, the optimal alignment of two transcription strings is accomplished with the aid of a dynamic programming algorithm [35]. This algorithm operates on a graph such as the one shown in Figure 3-6, which has been constructed using the utterances from our string alignment example. Each node in the graph represents a pair of words, one of which is an element of the reference utterance, and the other an element of the hypothesized utterance. Observe that a dummy word has been added at the beginning of each string, producing an extra column of nodes

along the left side of the graph and an extra row along the bottom. As we will see, this has been done to facilitate the execution of the algorithm.

Let us now introduce some convenient notation to refer to the augmented utterance transcriptions displayed in Figure 3-6, as well as the node coordinates in the graph. Specifically, let $T(i)$ represent the ith word in the reference transcription and $\widehat{T}(j)$ represent the jth word in the hypothesized transcription, and suppose that the transcriptions T and $\widehat{T}$ contain N_{REF} and N_{HYP} words, respectively. To fix the ordering of the graph coordinate system, we define the node (i, j) to be that which corresponds to the words $T(i)$ and $\widehat{T}(j)$. Finally, for future use we define a function $D(i, j)$ over all nodes in the graph to indicate whether corresponding words in T and $\widehat{T}$ are equal. This function is given by

$$D(i,j) = \begin{cases} 0 \text{ if } T(i) = \widehat{T}(j) \\ 1 \text{ if } T(i) \neq \widehat{T}(j) \end{cases} \tag{3.5}$$

for $0 \leq i \leq N_{\text{REF}}$ and $0 \leq j \leq N_{\text{HYP}}$.

Observe that each node in the graph has at most three directed arcs and leading from it to immediately neighboring nodes. A path through this graph (i.e., a contiguous sequence of directed arcs) beginning at node $(0, 0)$ and terminating at node $(N_{\text{REF}}, N_{\text{HYP}})$ will correspond to some alignment of the reference and hypothesized utterance transcriptions. Note, for example, that a vertically directed arc on such a path represents a deletion error made by the recognizer. This is true because traversing a vertically directed arc corresponds to advancing from the current word to the next word in the reference string while remaining at the current word in the hypothesized string, ultimately resulting in the alignment of a deleted word from the original reference string with a placeholder token in the modified hypothesized string. By an analogous argument, a horizontally directed arc represents an insertion error, and a diagonally directed arc represents either a substitution error or no error at all.

Before the dynamic programming algorithm can operate on the graph, a weight is assigned to each arc to indicate the penalty associated with the kind of error the arc represents. The algorithm then searches for the path leading from node $(0, 0)$ to node

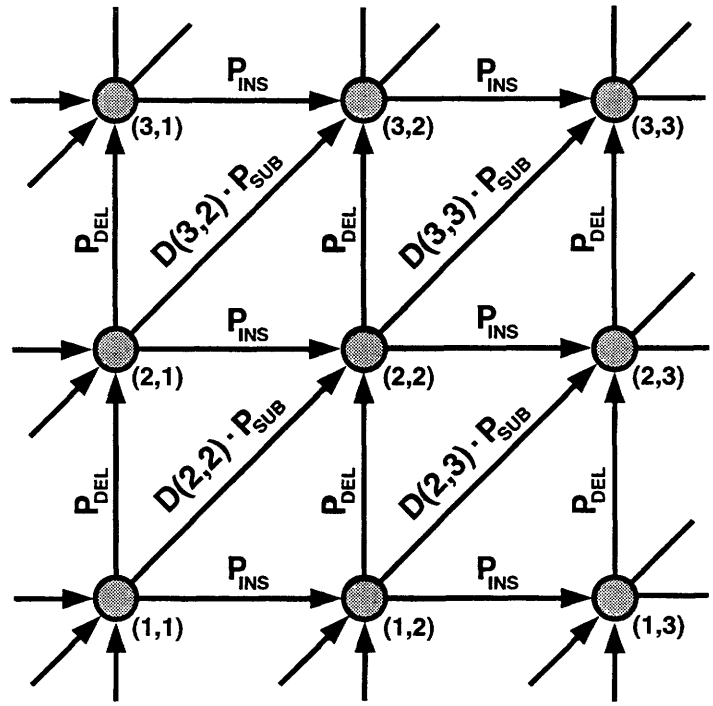


Figure 3-7: Detailed view of a portion of the graph used in the string alignment problem, labeled with node coordinates and arc weights.

$(N_{\text{REF}}, N_{\text{HYP}})$ whose arc weights sum to the smallest possible value. We define the quantities P_{SUB}, P_{INS}, and P_{DEL} to be the penalties incurred for a substitution error, an insertion error, and a deletion error, respectively, and we assume that a penalty of zero is incurred in the absence of an error. In addition, to obtain meaningful solutions from the dynamic programming algorithm, we typically require the penalty values P_{SUB}, P_{INS}, and P_{DEL} to satisfy the following three constraints:

(1) $P_{\text{SUB}} > 0,\ P_{\text{INS}} > 0,\ P_{\text{DEL}} > 0$

(2) $P_{\text{INS}} = P_{\text{DEL}}$

(3) $P_{\text{SUB}} < P_{\text{INS}} + P_{\text{DEL}}$

With the penalty values fixed, we define the arc weight assignment function $A(\cdot,\cdot)$ as

$$A((i-1,j),(i,j)) = P_{\text{DEL}} \quad \text{for } 1 \le i \le N_{\text{REF}},\ 0 \le j \le N_{\text{HYP}} \tag{3.6}$$

$$A((i,j-1),(i,j)) = P_{\text{INS}} \quad \text{for } 0 \le i \le N_{\text{REF}},\ 1 \le j \le N_{\text{HYP}} \tag{3.7}$$

$$A((i-1,j-1),(i,j)) \;=\; D(i,j)P_{\text{SUB}} \quad \text{for } 1 \le i \le N_{\text{REF}},\ 1 \le j \le N_{\text{HYP}} \tag{3.8}$$

Clearly, this function is defined only for certain neighboring pairs of nodes in the graph. Note that for diagonally directed arcs, the complementary indicator function $D(i,j)$ serves to activate the substitution penalty if $T(i) \neq \widehat{T}(j)$, and to deactivate this penalty if $T(i) = \widehat{T}(j)$. In Figure 3-7, we show a small portion of the original graph with the node coordinates and the above defined arc weights appropriately labeled.

As one might surmise from the basic structure of the graph and the fact that the cost function is additive, the path corresponding to optimal string alignment can be found using a recursive procedure. We define the function $C(i,j)$ to be the cost incurred along the optimal path from node $(0,0)$ to node (i,j), so that our ultimate objective is to solve for $C(N_{\text{REF}}, N_{\text{HYP}})$. We initialize the search algorithm by setting

$$C(0,0) \;=\; 0 \tag{3.9}$$

$$C(i,0) \;=\; iP_{\text{DEL}} \quad \text{for } 1 \le i \le N_{\text{REF}} \tag{3.10}$$

$$C(0,j) \;=\; jP_{\text{INS}} \quad \text{for } 1 \le j \le N_{\text{HYP}}. \tag{3.11}$$

Observe from the graph in Figure 3-6 that each node (i,j) with $1 \le i \le N_{\text{REF}}$ and $1 \le j \le N_{\text{HYP}}$ has exactly three arcs leading to it. In particular, these three arcs emanate from the three nodes $(i-1,j)$, $(i,j-1)$, and $(i-1,j-1)$. Thus, if we know the values $C(i-1,j)$, $C(i,j-1)$, and $C(i-1,j-1)$, we can compute $C(i,j)$ recursively (either by rows or by columns) using the formula

$$\begin{aligned} C(i,j) \;=\; \min\{&C(i-1,j) + P_{\text{INS}}, \\ &C(i,j-1) + P_{\text{DEL}}, \\ &C(i-1,j-1) + D(i,j)P_{\text{SUB}}\}. \end{aligned} \tag{3.12}$$

When carrying out this recursion, our goal is not only to compute the array of optimal cost values $C(i,j)$, but also to retain the specific paths that yield these optimal values.

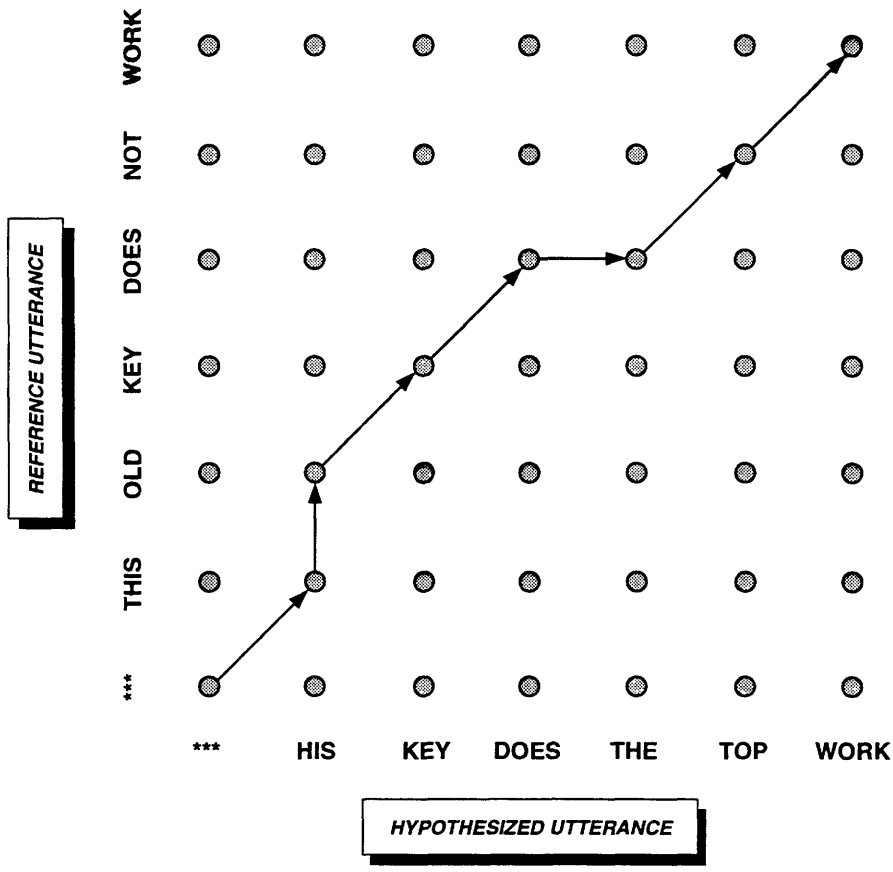


Figure 3-8: A path through the directed graph that represents optimal alignment of the hypothesized and reference utterances.

We can easily access this information during the recursion by recording the coordinate of the node at which the minimum occurs in Equation (3.12). In effect, this provides a backward pointer at each node that can be used when the recursion is done to trace the optimal path from the terminal node $(N_{\text{REF}}, N_{\text{HYP}})$ back to the initial node $(0, 0)$. For the particular example of Figure 3-6, an optimal (nonunique) path is shown in Figure 3-8. This graphical result corresponds to the following alignment of the reference and hypothesized utterances, which is consistent with our earlier string alignment found by inspection:

```
REFERENCE STRING:   THIS  OLD  KEY  DOES  ***  NOT  WORK
HYPOTHESIZED STRING: HIS  ***  KEY  DOES  THE  TOP  WORK
```

Let us suppose in the general case that, after alignment of the transcriptions

has been performed, we count N_{SUB} substitution errors, N_{DEL} deletion errors, and N_{INS} insertion errors. Then, assuming there are N_{TOT} words in each of the aligned transcriptions, the *word recognition accuracy* (WRA) is defined as

$$\text{WRA} = \frac{N_{\text{TOT}} - N_{\text{SUB}} - N_{\text{DEL}} - N_{\text{INS}}}{N_{\text{TOT}}} \tag{3.13}$$

As in the case of scoring phone classification performance, if multiple utterances are processed in a batch, the individual counts N_{SUB}, N_{DEL}, N_{INS}, and N_{TOT} are simply incremented with each new utterance, and word accuracy is computed using the above formula with the final count totals.

Chapter 4

Analysis of Experimental Results

In this chapter, we present and discuss the numerical results generated in the recognition performance tests described in Chapter 3. First, we examine the phone and word accuracy rates achieved by the recognizer as a function of the input SNR for the case in which white noise was added to the speech; we then analyze an analogous set of results for the colored-noise case. To aid in the interpretation of the results, we introduce a further series of experiments designed to evaluate the performance of each speech enhancement algorithm operating in isolation. In particular, these auxiliary tests are intended to measure the gain in speech quality afforded by each algorithm as a function of the input SNR.

4.1 Speech Recognition Performance in Additive White Noise

4.1.1 Phone Classification Results

In Figure 4-1, we show experimental results for the case in which the recognizer was configured to perform phone classification in the presence of additive white noise. Each curve in this figure represents the different levels of performance achieved by the recognizer (measured in terms of phone classification accuracy, as defined in Section 3.2.5) as the input signal-to-noise ratio was varied. The upper curve indicates

the performance achieved when the enhancement algorithm was applied to the noisy input speech, and the lower curve indicates the performance achieved in the absence of enhancement.

Although gains in performance can be measured either along the horizontal axis or the vertical axis of such plots, we shall concentrate mainly on the horizontal separation of the performance curves, since the horizontal-axis variable of input SNR is common to all of the experiments conducted. For the plots shown in Figure 4-1, we note that performance is uniformly improved by applying the speech enhancement algorithm, but only by a modest amount. For example, to achieve the moderately high phone classification accuracy of 0.6 in the experiment that includes a speech enhancement stage, the system requires input with an SNR of approximately 20 dB; however, to achieve the same level of accuracy in the experiment that does not include a speech enhancement stage, the system requires input of slightly higher quality, with an SNR of about 26 dB.

4.1.2 Word Recognition Results

In Figure 4-2, we show experimental results for the case in which the recognizer was configured to perform word recognition in the presence of additive white noise. The curves in this figure indicate the variation in word recognition accuracy (as defined in Section 3.2.5), with and without speech enhancement, as a function of the input SNR. Note that these performance curves exhibit much sharper slopes than do those from the phone classification experiments. In fact, as the SNR is varied from ∞ dB to 30 dB, there is a significant and immediate drop in performance; at an SNR of 20 dB — a moderate level of corruption by most standards — the performance degrades even more dramatically; at 10 dB, virtually no words are recognized.

To achieve the moderately high word recognition accuracy of 0.5 in the experiment that includes a speech enhancement stage, the system requires input with an SNR of approximately 27 dB; to achieve the same level of accuracy in the experiment that does not include a speech enhancement stage, the system requires input of slightly higher quality, with an SNR of about 29 dB. Thus, in this case the enhancement

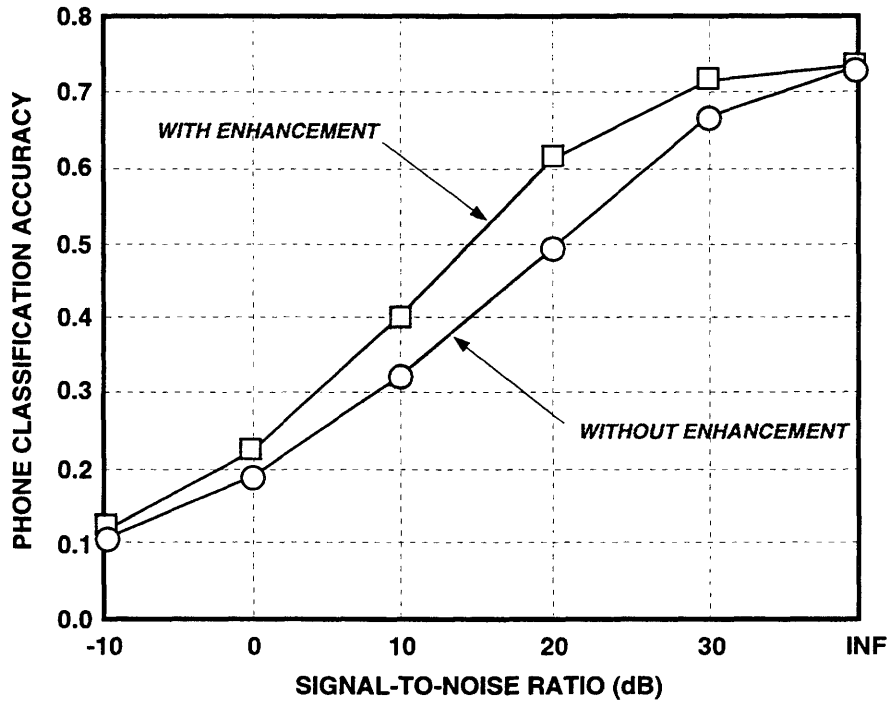


Figure 4-1: Phone classification accuracy of the SUMMIT system in the presence of white noise, with and without the use of a front-end speech enhancement stage.

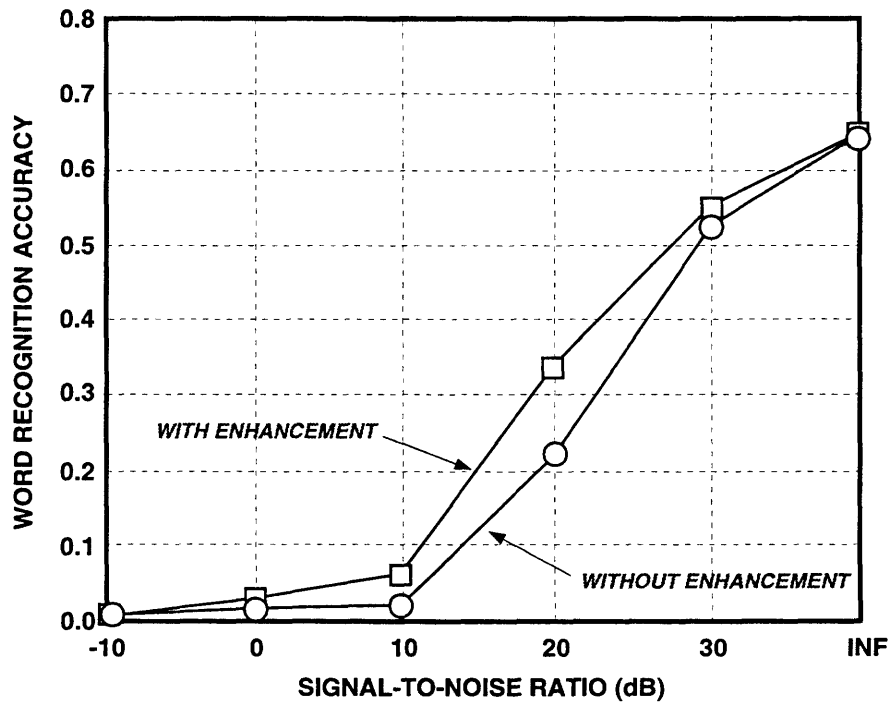


Figure 4-2: Word recognition accuracy of the SUMMIT system in the presence of white noise, with and without the use of a front-end speech enhancement stage.

operation yields only a slight improvement in noise immunity of about 2 dB.

4.2 Speech Recognition Performance in Additive Colored Noise

4.2.1 Phone Classification Results

In Figure 4-3, we show experimental results for the case in which the recognizer was configured to perform phone classification in the presence of additive colored noise. Upon comparing this figure to Figure 4-1, it is clear that the recognizer performs better in each of the colored-noise experiments than in the corresponding white-noise experiments. This improvement in performance results from the fact that there is much less spectral overlap between speech and the specific kind of colored noise that was added than between speech and white noise. Apart from the general boost in performance exhibited by both curves in Figure 4-3, there is also greater separation between the curves, indicating that the speech enhancement algorithm is generally more effective in this kind of noise. As mentioned earlier, the enhancement algorithm yields a larger performance gain in the colored-noise because at least some portion of the noise is predictable from sample to sample, and is therefore removable.

From Figure 4-3, we see that in order to achieve an accuracy score of 0.6 in the experiment that includes a speech enhancement stage, the system requires input with an SNR of approximately 6 dB; by contrast, to achieve the same level of accuracy in the experiment that does not include a speech enhancement stage, the system requires input with an SNR of about 17 dB. Thus, in this instance the speech enhancement operation yields an improvement in noise immunity of approximately 11 dB, which is significantly greater than the 6 dB gain observed in the white-noise experiments at the same accuracy rate of 0.6.

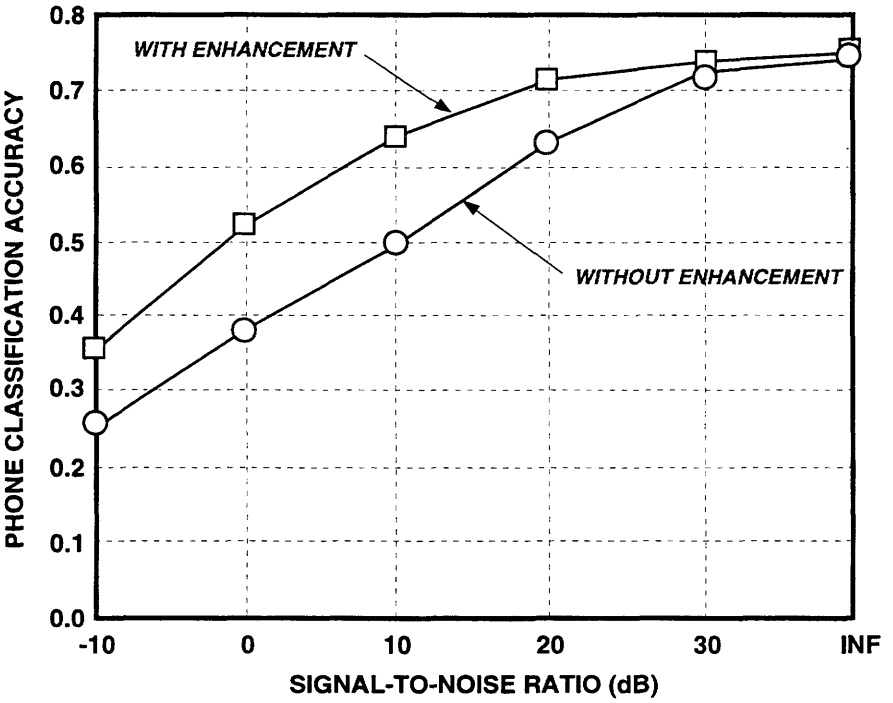


Figure 4-3: Phone classification accuracy of the SUMMIT system in the presence of colored noise, with and without the use of a front-end speech enhancement stage.

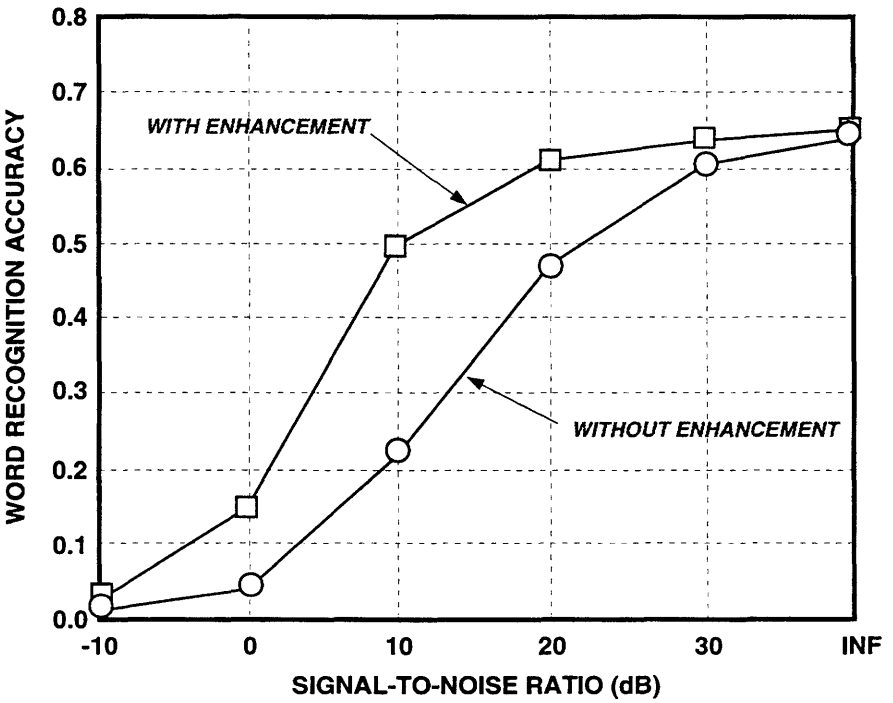


Figure 4-4: Word recognition accuracy of the SUMMIT system in the presence of colored noise, with and without the use of a front-end speech enhancement stage.

4.2.2 Word Recognition Results

In Figure 4-4, we show experimental results for the case in which the recognizer was configured to perform word recognition in the presence of additive colored noise. A comparison of this figure to Figure 4-2 indicates that the recognizer performs better at the task of word recognition in each of the colored-noise experiments than in the corresponding white-noise experiments. In addition, just as we observed in the phone classification experiments, the inclusion of a speech enhancement stage prior to recognition leads to a greater performance gain in the colored-noise case than in the white-noise case. For example, to achieve a word recognition accuracy of 0.5 in the experiment that includes a speech enhancement stage, the system requires an input SNR of approximately 10 dB; however, to achieve the same level of accuracy in the experiment that does not include a speech enhancement stage, the system requires the much higher input SNR of about 22dB. Thus, in this case the enhancement operation yields an improvement in noise immunity of about 12dB, which is significantly greater than the 2dB gain observed in the white-noise experiments at the same accuracy rate of 0.5.

It is instructive to examine the differences between the phone classification performance curves shown in Figure 4-3 and the word recognition performance curves shown in Figure 4-4. In particular, note that the curves in Figure 4-3 are flat at the upper end of the SNR scale, and then fall off steadily in the lower SNR range; the curves in Figure 4-4 have approximately the same shape as those in Figure 4-3 for large SNR values, but then drop off dramatically in the middle SNR range. This suggests that there is a minimum percentage of phones that must be classified correctly in order to guarantee a suitably high level of word recognition accuracy. By inspection, it appears that this critical threshold for phone classification accuracy is slightly higher than 0.6; the corresponding word recognition accuracy at this critical threshold is approximately 0.5. This result must be interpreted with care, however, because the classification and recognition tasks are performed and evaluated very differently.

4.3 Measuring Improvement in Speech Quality

From the material presented in Sections 4.1 and 4.2, we observed that gains in recognition performance after enhancement were greater in each of the colored-noise experiments than in the corresponding white-noise experiments. This suggests the possibility that the speech enhancement algorithm designed for the colored-noise case yields a higher-quality output signal for a given input SNR level than does the algorithm designed for the white-noise case. In this section, we investigate a method for measuring the isolated performance of each speech enhancement algorithm, and compare the gains in speech quality afforded by each algorithm as a function of the input SNR.

Recall that the time-adaptive signal enhancement algorithms derived in Chapter 2 were based on the assumption that speech can be accurately modeled as an autoregressive process over relatively brief intervals. Each algorithm makes use of this autoregressive model to sequentially estimate the present underlying speech sample based on past noisy observations, while incurring a minimum average squared error. Given this general model-based approach, together with the minimum-error design criterion, one might expect (at least for moderate levels of signal corruption) that the output waveform produced by such an algorithm will in some sense be a better representation of the original speech signal than will the input waveform. One might conjecture, for example, that in a subjective speech quality test a human listener would judge the output waveform to be more speech-like and to "sound" more like the original than the input waveform.

Because good speech quality is crucial for good speech recognition results, we would like to have a method of measuring the improvement in speech quality afforded by our signal enhancement algorithms. Moreover, we would like such a measure to be *objective*, so that it is automatically computable, yields consistent and predictable results, and is independent of any human judgment or interpretation. This objective measure should, however, give results that correlate fairly well with those obtained through subjective tests, so that it remains a reliable indicator of speech quality as

determined by human listeners.

Many objective measures of signal purity involve some definition of signal-to-noise ratio. These signal-to-noise measures are generally used to quantify amounts of signal distortion introduced by waveform coders, signal estimation algorithms, and a variety of other signal processing systems, and they are often applied to the basic problem of determining changes in speech quality that result from such signal processing operations. Of course, a measure of signal-to-noise improvement can only be applied to a system whose input and output waveforms are reasonable facsimiles of the original undistorted waveform, so that all of these waveforms can be aligned in time and the signal and noise components in each can be unambiguously identified.

The most popular of these signal-to-noise measures is the *classical* signal-to-noise ratio (SNR). To compute the SNR for a particular waveform that has undergone distortion, we require the distorted waveform itself along with the original version of this waveform, which serves as a reference pattern. For the specific case of measuring speech quality, let us suppose that we have available a clean speech signal $s(t)$ and a time-aligned but distorted version of this signal $s_d(t)$. (Such a distorted speech signal could arise in many ways, but in the present case we shall assume that the distortion results either from adding noise to the clean speech or from subsequently processing this speech-plus-noise signal with a signal enhancement algorithm.) The classical SNR for the waveform $s_d(t)$ is then given by

$$\text{SNR} = 10 \log_{10} \left(\frac{\sum_{t=0}^{N-1} s^2(t)}{\sum_{t=0}^{N-1} \left(s(t) - s_d(t) \right)^2} \right), \tag{4.1}$$

where N is the length in samples of each signal. This is precisely the measure we used to quantify the amount of corruption present in the input signal for the speech recognition experiments described in Chapter 3. However, while it may be true that this measure is suitable for studying system performance as a function of input signal purity, many researchers have consistently demonstrated that classical SNR is not a good indicator of subjective speech quality as measured through listening tests [30]. A much more accurate estimator of subjective speech quality is the *segmental* signal-

to-noise ratio (SEGSNR), which is obtained by computing the classical SNR over short nonoverlapping segments of the waveform and then averaging these SNR measurements over all such segments. Thus, assuming that our signals $s(t)$ and $s_d(t)$ consist of M adjacent segments with K samples each, the segmental SNR is given by

$$\text{SEGSNR} = \frac{1}{M} \sum_{m=0}^{M-1} 10 \log_{10} \left(\frac{\sum_{t=mK}^{mK+K-1} s^2(t)}{\sum_{t=mK}^{mK+K-1} (s(t) - s_d(t))^2} \right). \tag{4.2}$$

For this measure, the segment length K is usually chosen such that it represents a time interval of approximately 15 to 30 milliseconds. Clearly, for a prespecified segment length K, we will rarely have that $N = MK$ for some integer M; thus, in practice, when the overall waveform length is not an exact multiple of the segment length, the remaining samples at the end of the waveform are typically discarded.

The segmental SNR is a successful estimator of subjective speech quality because it accounts for the fact that speech signals are inherently nonstationary and are often processed by systems that are time-adaptive. Thus, it captures important and subtle aspects of speech quality by giving equal weight to high-energy and low-energy sections of an utterance. This is in contrast to the classical SNR measure, which generally does not provide a good indication of signal quality unless the signals of interest are stationary and are processed by time-invariant systems.

Since we now have an appropriate objective measure for speech quality, we can meaningfully analyze the effect of our signal enhancement algorithms on corrupted speech signals. To this end, we conducted a series of experiments to determine the gain in speech quality afforded by each of the enhancement algorithms as a function of classical SNR. These experiments were performed using a collection of 100 speech signals from the TIMIT data base, which comprised a subset of the signals used for the speech recognition experiments described earlier.

Each experiment was conducted in the following way. First, a classical SNR level was fixed and a set of distorted input signals was created by adding the appropriate amount of noise to each of the original clean speech signals. The average value of the segmental SNR was computed for this set of input signals, and then, after

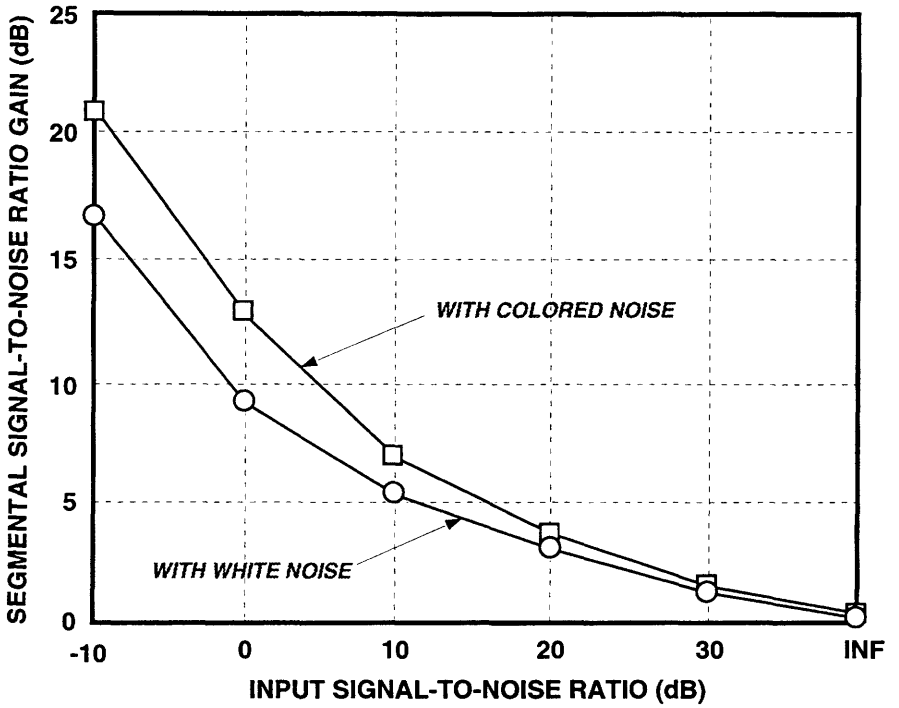


Figure 4-5: Improvement in speech quality (measured in terms of segmental SNR gain) yielded by the adaptive enhancement algorithms for the white-noise and colored-noise cases.

application of the signal enhancement algorithm, the average segmental SNR was computed for the resulting set of output signals. The average *gain* in segmental SNR, defined as the mean output SEGSNR value minus the mean input SEGSNR value, was then computed. This basic experiment was performed over a range of classical SNR levels from -10 dB to $+30$ dB in increments of 10 dB, and was also performed at a classical SNR of $+\infty$ dB by using the clean speech signals with no added noise. By conducting this suite of experiments for the separate cases in which white noise or colored noise is added to the speech, performance curves were generated for each of the signal enhancement algorithms derived in Chapter 2. These plots of segmental SNR gain versus classical SNR are shown in Figure 4-5. Note from this figure that the gain in speech quality provided by the colored-noise algorithm is greater than the gain provided by the white-noise algorithm at each classical SNR level; hence, these results are consistent with the recognition performance results observed in Sections 4.1 and 4.2.

Chapter 5

Conclusions and Future Directions

In this thesis, we have addressed the fundamental issue of how to increase the robustness of an automatic speech recognition system with respect to stationary additive environmental noise. The general approach adopted for improving recognition performance was to use a two-stage system, in which: (1) the noisy input waveform is filtered in order to enhance the underlying speech, and (2) the enhanced waveform is processed by the recognizer in an attempt to decode the utterance. Two adaptive speech enhancement algorithms were derived for use in the initial stage of the two-stage system; one of these algorithms was designed to operate in a white-noise environment, and the other in a colored-noise environment. A number of experiments were conducted to evaluate the performance of the recognizer in the presence of noise, with and without the use of a front-end speech enhancement stage. Recognition performance was evaluated at the phone and word levels, and at wide range of SNR values for both white and colored additive noise.

We found, for each kind of noise that was added, that the inclusion of a front-end speech enhancement stage uniformly improved recognition performance at each linguistic level. For a fixed word recognition accuracy of 0.5, the noise immunity of the system improved only slightly in the white-noise case (approximately 2 dB), but improved quite substantially in the colored-noise case (approximately 12 dB). The larger gain achieved in the colored-noise case was attributed to the fact that colored noise, unlike white noise, is in part predictable from sample to sample, and

hence is more readily suppressed. Moreover, it was demonstrated empirically that the enhancement algorithm designed to operate in colored noise produced better output speech quality for each input SNR level than did the algorithm designed to operate in white noise. However, even with the relatively large performance gain achieved in the colored-noise case, the two-stage system still required an input SNR of at least 10 dB (not a severe noise level when the noise is highly correlated in time) in order to keep word recognition accuracy above the critical level of 0.5. Thus, for the particular recognition task considered in this thesis, the adaptive speech enhancement algorithms are suitable for use only in environments with low noise levels.

The basic approach proposed herein for improving recognition performance is ultimately limited by the division of functionality (i.e., speech enhancement followed by speech recognition) inherent in its two-stage structure. For example, if the speech recognition component remains fixed, then overall performance of the two-stage system can be improved only by incorporating a better noise-suppression component at the front end. One straightforward approach for boosting noise immunity, aside from the development of new single-sensor enhancement algorithms, is to combine measurements from multiple microphones placed in different locations throughout the environment. Indeed, this multi-sensor approach has already received some attention in the literature [7, 38, 44].

Regardless of whether single or multiple microphone measurements are available for processing, however, the basic two-stage approach for improving speech recognition performance need not be used. Clearly, better performance could be achieved by estimating the content of the entire utterance *directly* from the noisy measurements, rather than by first estimating the underlying speech signal values and then estimating the content of the utterance based on the assumption that the enhanced speech signal is in fact completely free of noise. A direct utterance estimation problem of this kind poses a formidable challenge, since it requires a strong interaction between the technologies of signal processing and speech recognition; nonetheless, consideration of this complex problem will be an important element of future research in robust speech recognition.

Bibliography

[1] A. Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*, Kluwer Academic Publishers, Boston, 1993.

[2] B. Anderson and J. Moore, *Optimal Filtering*, Prentice-Hall Inc., Englewood Cliffs, 1979.

[3] B. Carlson and M. Clements, "Speech Recognition in Noise Using a Projection-Based Likelihood Measure for Mixture Density HMMs," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 237-240, San Francisco, CA, 1992.

[4] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, vol. B39, pp. 1-38, 1977.

[5] Y. Ephraim, D. Malah, and B. Huang, "Speech Enhancement Based upon Hidden Markov Modeling," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 353-356, Glasgow, Scotland, 1989.

[6] A. Erell and M. Weintraub, "Estimation Using a Log-Spectral-Distance Criterion for Noise-Robust Speech Recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 853-856, Albuquerque, NM, 1990.

[7] M. Feder, A. Oppenheim, and E. Weinstein, "Maximum Likelihood Noise Cancellation Using the EM Algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 2, pp. 204-216, February 1989.

[8] A. Gelb, editor, *Applied Optimal Estimation*, The MIT Press, Cambridge, MA, 1974.

[9] O. Ghitza, "Auditory Nerve Representation as a Front-End for Speech Recognition in a Noisy Environment," *Computer Speech and Language*, vol. 1, no. 2, pp. 109-130, December 1986.

[10] D. Gilliland, "Sequential Compound Estimation," *Annals of Mathematical Statistics*, vol. 39, no. 6, pp. 1890-1904, 1968.

[11] G. Goodwin and K. Sin, *Adaptive Filtering Prediction and Control*, Prentice-Hall Inc., Englewood Cliffs, 1984.

[12] J. Hansen, "Adaptive Source Generator Compensator and Enhancement for Speech Recognition in Noisy Stressful Environments," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 95-98, Minneapolis, MN, 1993.

[13] J. Hansen and M. Clements, "Constrained Iterative Speech Enhancement with Application to Automatic Speech Recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 561-564, New York, NY, 1988.

[14] S. Haykin, *Adaptive Filter Theory*, 2nd edition, Prentice-Hall Inc., Englewood Cliffs, 1991.

[15] M. Hunt, "Evaluating the Performance of Connected-Word Speech Recognition Systems," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 457-460, New York, NY, 1988.

[16] K. Lee, *Automatic Speech Recognition: The Development of the SPHINX System*, Kluwer Academic Publishers, Boston, 1989.

[17] K. Lee, B. Lee, I. Song, and S. Ann, "Robust Estimation of AR Parameters and its Application for Speech Enhancement," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 309-312, San Francisco, CA, 1992.

[18] J. Lim, *Speech Enhancement*, Prentice-Hall Inc., Englewood Cliffs, 1983.

[19] J. Lim and A. Oppenheim, "All-Pole Modeling of Degraded Speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 192-210, June 1978.

[20] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, The MIT Press, Cambridge, MA, 1983.

[21] L. Ljung, *System Identification: Theory for the User*, Prentice-Hall Inc., Englewood Cliffs, 1987.

[22] B. Mellor and A. Varga, "Noise Masking in a Transform Domain," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 87-90, Minneapolis, MN, 1993.

[23] Y. Nogami, "The k-Extended Set Compound Estimation Problem in a Nonregular Family of Distributions over $[\theta, \theta + 1]$," *Annals of the Institute of Statistical Mathematics*, vol. 31a, pp. 169-176, 1979.

[24] A. Oppenheim, E. Weinstein, K. Zangi, M. Feder, and D. Gauger, "Single Sensor Active Noise Cancellation Based on the EM Algorithm," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 277-280, San Francisco, CA, 1992.

[25] A. Oppenheim, E. Weinstein, K. Zangi, M. Feder, and D. Gauger, "Single-Sensor Active Noise Cancellation," *IEEE Transactions on Speech and Audio Processing*, April 1994.

[26] D. Pallett, "Test Procedures for the March 1987 DARPA Benchmark Tests," *Proceedings of the DARPA Speech Recognition Workshop*, San Diego, CA, pp. 75-78, March 1987.

[27] J. Picone, G. Doddington, and D. Pallett, "Phone-Mediated Word Alignment for Speech Recognition Evaluation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 3, pp. 559-562, March 1990.

[28] J. Picone, K. Goudie-Marshall, G. Doddington, and W. Fisher, "Automatic Text Alignment for Speech System Evaluation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 780-784, August 1986.

[29] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd edition, Cambridge University Press, Cambridge, 1992.

[30] S. Quackenbush, T. Barnwell, and M. Clements, *Objective Measures of Speech Quality*, Prentice-Hall Inc., Englewood Cliffs, 1988.

[31] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall Inc., Englewood Cliffs, 1978.

[32] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall Inc., Englewood Cliffs, 1993.

[33] H. Robbins, "Asymptotically Subminimax Solutions of Compound Statistical Decision Problems," *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistical Problems*, pp. 131-148, 1951.

[34] H. Robbins and S. Monro, "A Stochastic Approximation Method," *Annals of Mathematical Statistics*, vol. 22, pp. 400-407, 1951.

[35] D. Sankoff and J. Kruskal, eds., *Time Warps, String Edits, and Macromolecules: Theory and Practice of Sequence Comparison*, Addison-Wesley Publishing Company Inc., Reading, MA, 1983.

[36] S. Seneff, "A Joint Synchrony/Mean-Rate Model of Auditory Speech Processing," *Journal of Phonetics*, vol. 16, no. 1, pp. 55-76, 1988.

[37] S. Seneff, "TINA: A Probabilistic Syntactic Parser for Speech Understanding Systems," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 711-714, Glasgow, Scotland, 1989.

[38] T. Sullivan and R. Stern, "Multi-Microphone Correlation-Based Processing for Robust Speech Recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 91-94, Minneapolis, MN, 1993.

[39] "The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus," CD-ROM *NIST Speech Disc 1-1.1*, National Institute of Standards and Technology, October 1990.

[40] D. Tsoukalas, M. Paraskevas, and J. Mourjopoulos, "Speech Enhancement Using Psychoacoustic Criteria," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 359-362, Minneapolis, MN, 1993.

[41] J. Van Ryzin, "The Sequential Compound Decision Problem with $m \times n$ Finite Loss Matrix," *Annals of Mathematical Statistics*, vol. 37, pp. 954-975, 1966.

[42] S. Vardeman, "Admissible Solutions of k-Extended Finite State Set and Sequence Compound Decision Problems," *Journal of Multivariate Analysis*, vol. 10, pp. 426-441, 1980.

[43] S. Vaseghi and B. Milner, "Noisy Speech Recognition Based on HMMs, Wiener Filters, and Re-evaluation of Most-Likely Candidates," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 103-106, Minneapolis, MN, 1993.

[44] E. Weinstein, A. Oppenheim, and M. Feder, "Signal Enhancement Using Single and Multi-Sensor Measurements," RLE Technical Report No. 560, November 1990.

[45] V. Zue, J. Glass, M. Phillips, and S. Seneff, "Acoustic Segmentation and Phonetic Classification in the SUMMIT Speech Recognition System," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 389-392, Glasgow, Scotland, 1989.

[46] V. Zue, J. Glass, D. Goodine, M. Phillips, and S. Seneff, "The SUMMIT Speech Recognition System: Phonological Modelling and Lexical Access," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 49-52, Albuquerque, NM, 1990.

[47] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff, "Recent Progress on the SUMMIT System," *Proceedings of the Speech and Natural Language Workshop*, pp. 380-384, Hidden Valley, PA, 1990.