Maximum-Likelihood Estimation of a Class of Chaotic Signals

Haralabos C. Papadopoulos, *Student Member, IEEE,*
and Gregory W. Wornell, *Member, IEEE*

Abstract—**The chaotic sequences corresponding to tent map dynamics are potentially attractive in a range of engineering applications. Optimal estimation algorithms for signal filtering, prediction, and smoothing in the presence of white Gaussian noise are derived for this class of sequences based on the method of Maximum Likelihood. The resulting algorithms are highly nonlinear but have convenient recursive implementations that are efficient both in terms of computation and storage. Performance evaluations are also included and compared with the associated Cramér–Rao bounds.**

Index Terms—**Chaos, nonlinear dynamics, recursive estimation, maximum likelihood, Kalman filtering.**

I. Introduction

Chaotic signals, i.e., signals which can be described as outputs of nonlinear dynamical systems exhibiting chaotic behavior are appealing candidates for use in a variety of engineering contexts. In terms of signal analysis, these signals constitute potentially useful models for a range of natural phenomena. In terms of signal synthesis, the special characteristics of chaotic signals are potentially attractive in a number of broadband communication and radar applications. In order to exploit chaotic signals in both types of applications, there is a need for robust and efficient algorithms for the detection and estimation of these signals in the presence of various forms and amounts of distortion.

A variety of heuristically reasonable algorithms have been proposed for estimating chaotic signals in backgrounds of additive, stationary white Gaussian noise given varying degrees of *a priori* information; see e.g., [1]–[3]. However, the development of optimal estimators for these scenarios has generally proved to be rather difficult.

In this correspondence, we focus our attention on the particular class of first-order, discrete-time chaotic signals whose dynamics are governed by the so-called tent map. For these chaotic signals, we develop estimators that are optimal in a Maximum-Likelihood (ML) sense, possess highly convenient recursive implementations, and are closely related to traditional Kalman filters.

II. Chaotic Sequences from Tent Maps

The chaotic sequences $x[n]$ of interest in this work are generated according to the following one-dimensional dynamics:

$$x[n] = F(x[n-1]) \tag{1}$$

where $F(\cdot)$ is a symmetric tent map, i.e.,

$$F(x) = \beta - 1 - \beta|x| \tag{2}$$

with parameter $1 < \beta \leq 2$.

For almost all initial conditions $x[0]$ in $(-1, \beta - 1)$, it is well known [4] that these mappings produce ergodic sequences whose values remain in the range $(-1, \beta - 1)$. In the sequel, we restrict our attention to the ergodic case, which allows us to use time- and ensemble-averages interchangeably. The associated invariant distribution (first-order) is then obtained as a solution to the corresponding Perron–Frobenius equation [5], although, in general, it cannot be expressed in closed form. Likewise, these processes have broadband spectra, although closed-form expressions are not available in general.

The Lyapunov exponent λ is a measure of the numerical sensitivity of the map, describing, in particular, the average rate at which successive iterates generated from nearby initial conditions $x[0]$ diverge. For the tent maps given by (2)

$$\lambda = E\left[\log|dF/dx|\right] = \log\beta \tag{3}$$

where we use $E[\cdot]$ to denote expectation with respect to a random initial condition selected according to the invariant density $p(x)$ of the map [4].

For the map corresponding to $\beta = 2$, one can derive more detailed results. In particular, the invariant density is uniform [4], i.e.,

$$p(x) = \begin{cases} 1/2, & |x| < 1 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

which implies, among other properties, that the sequences are zero-mean. Furthermore, since both $F(\cdot)$ and the invariant density (4) are even functions, it follows that the autocorrelation of such sequences is given by

$$R[k] = E[x[n+k]x[n]] = E[x[n]F^{(k)}(x[n])] = \begin{cases} 1/3, & k = 0 \\ 0, & \text{otherwise} \end{cases}$$

where $F^{(k)}(\cdot)$ denotes the k-fold composition of $F(\cdot)$ with itself. Hence $x[n]$ has a time-averaged spectrum that is white, i.e.,

$$S(\omega) = 1/3, \quad \text{all } \omega. \tag{5}$$

For this reason, we refer to those sequences corresponding to $\beta = 2$ as "chaotic white noise," and we will frequently specialize our results to this case.

Because $F(\cdot)$ in (2) is unimodal and even, it has two inverse images that differ only in sign—i.e., given $v = F(x)$, x can be determined from v to within its sign. For convenience, we denote the two inverses of $F(\cdot)$ by

$$F_s^{-1}(v) = \frac{\beta - 1 - v}{\beta} \cdot s \tag{6}$$

where $s = \pm 1$. Thus we have the relation

$$v = F(x) \Rightarrow x = F_{\operatorname{sgn} x}^{-1}(v).$$

For future reference we also define

$$F_s(x) = \beta - 1 - \beta s x \tag{7}$$

and note that

$$F(x) = F_{\operatorname{sgn} x}(x).$$

Since, given s, $F_s(\cdot)$ and $F_s^{-1}(\cdot)$ are inverses, and since both are linear in their arguments two useful identities can be readily verified. First, for any a and b, and any $s = \pm 1$, we have

$$|a - F_s^{-1}(b)| = |F_s(a) - b|/\beta. \tag{8}$$

Manuscript received March 28, 1993; revised February 4, 1994. This work was supported in part by the Advanced Research Projects Agency monitored by ONR under Contract N00014-93-1-0686, and the Air Force Office of Scientific Research under Grant AFOSR-91-0034.

The authors are with the Department of Electrical Engineering and Computer Science, and the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

IEEE Log Number 9407110.

Second, for any $u_1, u_2, \cdots, u_m$, we have

$$\sum_{k=1}^{m} a_k = 1 \Rightarrow F_s\left(\sum_{k=1}^{m} a_k u_k\right) = \sum_{k=1}^{m} a_k F_s(u_k). \tag{9}$$

For convenience, we also introduce the following notation for k-fold compositions of $F_s(\cdot)$ and $F_s^{-1}(\cdot)$; specifically,

$$F_{s_k} \circ F_{s_{k-1}} \circ \cdots \circ F_{s_1}(x) \triangleq F^{(k)}_{s_k, s_{k-1}, \ldots, s_1}(x) \tag{10}$$

$$F^{-1}_{s_1} \circ F^{-1}_{s_2} \circ \cdots \circ F^{-1}_{s_k}(x) \triangleq F^{(-k)}_{s_1, s_2, \ldots, s_k}(x) \tag{11}$$

where $s_1, s_2, \cdots, s_k$ and x are arbitrary.

Via inverse mappings, we obtain a useful alternative representation for a sequence

$$x[0], x[1], \cdots, x[N] \tag{12}$$

generated according to (1) with (2). In particular, for each n we have, using the notation (11)

$$x[n] = F^{(n-N)}_{s[n], s[n+1], \ldots, s[N-1]}(x[N]) \tag{13}$$

with

$$s[n] = \operatorname{sgn} x[n].$$

Hence

$$s[0], s[1], \cdots, s[N-1], x[N] \tag{14}$$

is an equivalent representation for (12), and (13) defines the coordinate transformation. It is this representation we exploit in the sequel.

III. ML Estimation of Tent Map Sequences

Let us consider the estimation of a chaotic tent map sequence from a set of N noisy observations

$$y[0], y[1], \cdots, y[N]. \tag{15}$$

Specifically, suppose

$$y[n] = x[n] + w[n] \tag{16}$$

where $w[n]$ is a stationary, zero-mean white Gaussian noise sequence with variance σ_w^2, and $x[n]$ is a tent map sequence generated by iterating some unknown $x[0] \in (-1, \beta - 1)$ according to (1) using (2) for some parameter $1 < \beta \leq 2$. The objective is to obtain ML estimates of

$$x[0], x[1], \cdots, x[N]$$

from the noisy data.

For future convenience, let $\hat{x}[n|m]$ denote the ML estimate of $x[n]$ given $y[k]$ for $0 \leq k \leq m$. In addition, note that since ML estimation is invariant to nonlinear transformations,

$$\hat{x}[n \mid m] = F^{(k)}(\hat{x}[n-k \mid m]) \tag{17}$$

for any m and $k \leq n$. Hence, given $\hat{x}[0|N]$, the remainder of the sequence can, in principle, be obtained by iterating this estimate according to (17). Although estimating $x[0]$ directly is therefore appealing, this approach leads to a difficult optimization problem. Indeed, as reported in [6], for chaotic maps of this type the associated likelihood function is typically a highly irregular function with fractal characteristics. Consequently, gradient descent algorithms cannot practically be applied to this problem to obtain ML estimates of $x[0]$.

A more effective approach, and the one we employ in this correspondence, involves recasting the problem into one of finding ML estimates for the coordinates (14) as an intermediate step. The resulting estimation is then naturally partitioned into a filtering stage and a smoothing stage. Before proceeding with the derivation of the algorithm, we first develop some preliminary results.

We begin by denoting the parameters to be estimated by $\hat{\theta}_N$, where

$$\hat{\theta}_n = \{\hat{s}[0 \mid n], \hat{s}[1 \mid n], \cdots, \hat{s}[n-1 \mid n], \hat{x}[n \mid n]\} \tag{18}$$

and where $\hat{s}[n|m]$ denotes the ML estimate of $s[n]$ given $y[k]$ for $0 \leq k \leq m$. Then, since for any $n \geq 0$, $y[0], y[1], \cdots, y[n]$ is a collection of independent Gaussian random variables with equal variance

$$\hat{\theta}_n = \operatorname*{argmin}_{\theta} \varepsilon[n] \tag{19}$$

where

$$\varepsilon[n] = \sum_{k=0}^{n} (y[k] - x[k])^2. \tag{20}$$

We may rewrite (20) in the form

$$\begin{aligned} \varepsilon[n] &= \varepsilon[n; s[0], s[1], \cdots, s[n-1], x[n]] \\ &= \sum_{k=0}^{n} \left(y[k] - F^{(k-n)}_{s[k], s[k+1], \cdots, s[n-1]}(x[n])\right)^2 & (21) \\ &= \sum_{k=0}^{n} \beta^{2(k-n)} \left(F^{(n-k)}_{s[n-1], s[n-2], \cdots, s[k]}(y[k]) - x[n]\right)^2 & (22) \end{aligned}$$

where (21) follows from (13), and where (22) follows from applications of the identity (8).

The following lemma will be especially useful, a proof of which is provided in the Appendix.

Lemma 1: Let N and n be arbitrary integers such that $0 \leq n \leq N-1$, let $s[0], s[1], \cdots, s[n-1]$, and $s[n+1], s[n+2], \cdots, s[N]$ be arbitrary binary (± 1) sequences, and let x be an arbitrary real number such that $x \in (-1, \beta - 1)$. Then

$$\operatorname*{argmin}_{s[n]} \varepsilon[N; s[0], s[1], \cdots, s[N-1], x] = \operatorname{sgn} \sum_{k=0}^{n} \beta^{2(k-n)} F^{(n-k)}_{s[n-1], \cdots, s[k]}(y[k]). \tag{23}$$

Note that the right-hand side of (23) is independent of both x and $s[n+1], s[n+2], \cdots, s[N]$. As an immediate consequence we therefore have that

$$\hat{s}[n \mid n] = \hat{s}[n \mid N] = \operatorname{sgn} \sum_{k=0}^{n} \beta^{2(k-n)} F^{(n-k)}_{\hat{s}[n-1], \hat{s}[n-2], \cdots, \hat{s}[k]}(y[k]). \tag{24}$$

The right equality in (24) follows immediately from (23) and the definition of $\hat{s}[n|n]$. The left equality in (23) is a consequence of the fact that the right-hand side of (23) is independent of the data samples $y[k]$ for $k > n$. As a result of (24), we will, without risk of ambiguity, use $\hat{s}[n]$ to denote both $\hat{s}[n|n]$ and $\hat{s}[n|N]$.

Using the above results, the filtering, smoothing, and prediction algorithms can all be derived in a straightforward manner, as we now show.

A. Filtering

Filtering provides the ML estimates $\hat{x}[n \mid n]$ for $n = 0, 1, \cdots, N$. These estimates are obtained by sequential, causal processing of the data using the following efficient recursive algorithm.

Due to (19), $\hat{x}[n|n]$ is readily obtained by differentiating (22) with respect to $x[n]$ given $s[k] = \hat{s}[k]$ for $k = 0, 1, \cdots, n-1$, and solving for the unique stationary point. In particular, we get

$$\hat{x}[n \mid n] = \frac{\sum_{k=0}^{n} \beta^{2(k-n)} F^{(n-k)}_{\hat{s}[n-1], \hat{s}[n-2], \cdots, \hat{s}[k]}(y[k])}{\sum_{k=0}^{n} \beta^{2(k-n)}}. \tag{25}$$

The recursion for $\hat{x}[n \mid n]$ then takes the form

$$\hat{x}[n \mid n] = \frac{(\beta^2 - 1)\beta^{2n} y[n] + (\beta^{2n} - 1)\hat{x}[n \mid n-1]}{\beta^{2(n+1)} - 1} \tag{26}$$

where, via (17), $\hat{x}[n|n-1]$ is given by

$$\hat{x}[n|n-1] = F(\hat{x}[n-1 \mid n-1]) \tag{27}$$

and where the initialization is

$$\hat{x}[0 \mid 0] = y[0]. \tag{28}$$

To verify (26), it suffices to apply the identity (9) to (27) with $\hat{x}[n-1 \mid n-1]$ expanded according to (25), and substitute the result into the right side of (26). For sufficiently large n, the weights in (26) settle to their steady-state values and the recursion simplifies to

$$\hat{x}[n \mid n] = \frac{\beta^2 - 1}{\beta^2} y[n] + \frac{1}{\beta^2} \hat{x}[n \mid n-1].$$

It is important to recognize, however, that the quantities $\hat{x}[n \mid n]$ computed according to the recursion (26) above are not quite the ML estimates. This is because $\hat{x}[n \mid n]$ is only given by (25) when the right-hand side is in the range $(-1, \beta - 1)$. However, it is straightforward to verify that the true ML estimates $\hat{x}_{\mathrm{ML}}[n \mid n]$ are closely related. In particular, to generate $\hat{x}_{\mathrm{ML}}[n \mid n]$ an intermediate sequence is first generated via the recursion (26) initialized with (28), then this intermediate sequence is amplitude-limited according to

$$\hat{x}_{\mathrm{ML}}[n \mid n] = \mathcal{I}_\beta(\hat{x}[n \mid n]) \tag{29}$$

where

$$\mathcal{I}_\beta(x) = \begin{cases} x, & x \in (-1, \beta - 1) \\ -1, & x \leq -1 \\ \beta - 1, & x \geq \beta - 1. \end{cases} \tag{30}$$

Interestingly, the ML filtering algorithm just developed is closely related to the extended Kalman filter [7] for this problem. To see this, we note that (26) may be rewritten in the form

$$\hat{x}[n \mid n] = \hat{x}[n \mid n-1] + K[n](y[n] - \hat{x}[n \mid n-1]) \tag{31a}$$

where

$$K[n] = P[n \mid n]/\sigma_w^2 \tag{31b}$$

and where

$$1/P[n \mid n] = 1/P[n \mid n-1] + 1/\sigma_w^2 \tag{31c}$$

$$1/P[0 \mid 0] = 1/\sigma_w^2 \tag{31d}$$

with

$$P[n \mid n-1] = \beta^2 P[n-1 \mid n-1]. \tag{31e}$$

Noting, in addition, that the coefficient β^2 in (31e) corresponds to

$$\beta^2 = \left(\frac{dF(x)}{dx}\right)^2 \Bigg|_{x = \hat{x}[n-1|n-1]}$$

we see that (31a)–(31e) are precisely the extended Kalman filter equations for the problem, with $K[n]$ denoting the Kalman gain and $P[n \mid m]$ denoting a generalized error variance associated with the estimate $\hat{x}[n|m]$.

As is well known, for nonlinear estimation problems, in general, estimates produced by extended Kalman filters are usually not optimal with respect to any meaningful criterion, and the associated generalized error variances do not correspond to the actual error variances. However, in this particular estimation problem, we have, somewhat remarkably, that the extended Kalman filter produces estimates $\hat{x}[n \mid n]$, which when amplitude-limited according to (29), are true ML estimates $\hat{x}_{\mathrm{ML}}[n|n]$. In addition, as will become apparent in Section IV, the pseudo error variance $P[n|n]$ in this case turns out to be precisely the associated Cramér–Rao bound.

B. Smoothing

The ML filtered estimates $\hat{x}_{\mathrm{ML}}[n|n]$ are obtained using a forward pass through the data; to obtain the smoothed estimates $\hat{x}_{\mathrm{ML}}[n|N]$ requires backward propagation of the filtered estimates. Specifically, as an immediate consequence of the invertible coordinate transformation (13), the backward recursion is

$$\hat{x}_{\mathrm{ML}}[n \mid N] = F^{-1}_{\hat{s}[n|N]}(\hat{x}_{\mathrm{ML}}[n+1 \mid N]) \tag{32}$$

and is initialized with $\hat{x}_{\mathrm{ML}}[N \mid N]$.

Furthermore, comparing (24) with (25), we see immediately that

$$\hat{s}[n \mid N] = \hat{s}[n] = \mathrm{sgn}\ \hat{x}[n \mid n]. \tag{33}$$

Consequently, the $\hat{s}[n]$ are readily computed during the filtering pass. In fact, since smoothing requires no further access to the data, the estimation may be implemented so as to be efficient not only in terms of computation, but also in terms of storage. In particular, each $y[n]$ for $n < N$ may be replaced in memory with $\hat{s}[n]$ as it is computed, and $y[N]$ may be replaced with $\hat{x}[N|N]$.

Note, too, that (32) and (33) imply that for $n \geq 1$

$$\hat{x}_{\mathrm{ML}}[n \mid N] = F(\hat{x}_{\mathrm{ML}}[n-1 \mid N]) \tag{34}$$

consistent with (17).

C. Prediction

ML one-step predictors arise rather naturally in the solution to the filtering problem. As a final remark, we note that, more generally, ML K-step predictors can be also be readily derived. In particular, as an immediate consequence of (17), we see that the K-step ML predictions are constructed via the recursion

$$\hat{x}_{\mathrm{ML}}[N + K \mid N] = F(\hat{x}_{\mathrm{ML}}[N + K - 1 \mid N]) \tag{35}$$

initialized with the ML filtered estimate $\hat{x}_{\mathrm{ML}}[N \mid N]$. This prediction result (35) is also consistent with our smoothing result (34).

IV. Performance Characteristics

In this section we focus on two aspects of the performance of the ML estimators developed in this work: bias and error variance.

A. Bias

In general, the ML estimates corresponding to filtering, smoothing, and prediction are all biased. As an illustration, in Fig. 1, we plot the magnitude of the bias in the estimates $\hat{x}[N + K \mid N]$ for $\beta = 2$, $N = 49$, and $-50 < K < 20$ at three different SNR levels, corresponding to 20, 30, and 40 dB.

Additional features are apparent in Fig. 2, where the dashed curve indicates the steady-state bias in the filtered signal estimates as a function of signal-to-noise ratio (SNR) for the case $\beta = 2$, as determined from Monte Carlo simulations. The solid curve in this

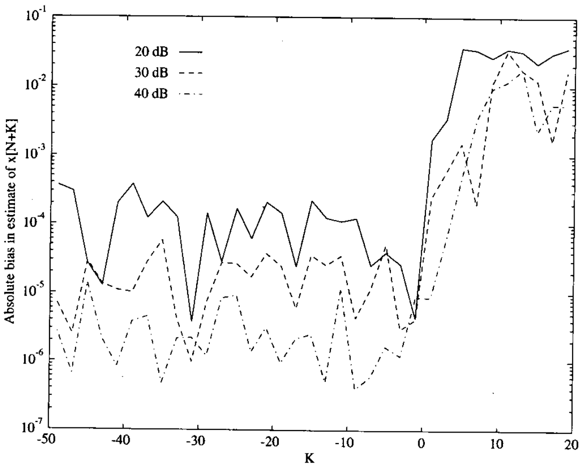

Fig. 1. Magnitude of the bias in the ML estimates $\hat{x}_{\mathrm{ML}}[N+K \mid K]$ as a function of K ($\beta = 2$).

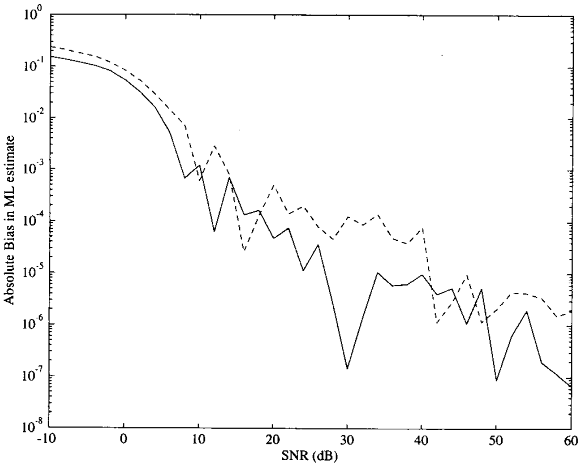

Fig. 2. Bias in the ML estimates ($\beta = 2$). The dashed curve corresponds to the bias in filtering, while the solid curve corresponds to the bias in smoothing.

same figure plots the bias in the smoothed ML estimates for the case $\beta = 2$ based on Monte Carlo simulations. Note that the fact that the ML estimates are asymptotically unbiased at high SNR is reflected in these results.

B. Variance

It is straightforward to obtain the Cramér–Rao bound on the variance of any unbiased estimate. In particular, we have, in general

$$\mathrm{var}\ \hat{x}[n \mid m] \geq 2\sigma_w^2 / E[\partial^2 \varepsilon[m] / \partial x^2[n]] \tag{36}$$

where $E[\cdot]$ denotes expectation with respect to the observation noise for a fixed initial condition.[1] For the case of filtering, it is convenient to express $\varepsilon[n]$ in the form (22), which when combined with (36) yields

$$\mathrm{var}\ \hat{x}[n \mid n] \geq \sigma_w^2 \cdot (1 - \beta^{-2}) \cdot [1 - \beta^{-2(n+1)}]^{-1}. \tag{37}$$

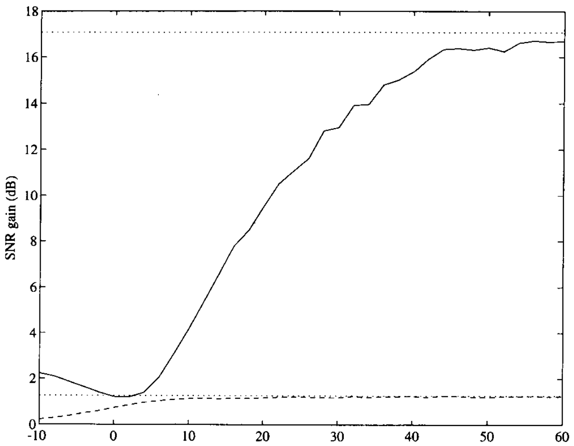

Fig. 3. SNR gain in the ML estimates ($\beta = 2$). The dashed curve corresponds to filtering, while the solid curve corresponds to smoothing. The lower and upper dotted lines correspond to the Cramér–Rao bounds for filtering and smoothing, respectively.

As $n \to \infty$, (37) decays to $(1 - \beta^{-2})\sigma_w^2$, which, for the case $\beta = 2$ corresponds to $(3/4)\sigma_w^2$. Consequently, the asymptotic filtering gain of the ML estimate is

$$10 \log_{10} \frac{\beta^2}{\beta^2 - 1} \text{ dB} \tag{38}$$

or only approximately 1.25 dB in the case $\beta = 2$. In Fig. 3, the dashed curve depicts the steady-state SNR gain (over simply amplitude-limiting $y[n]$) in the filtered signal estimates as a function of SNR as determined from Monte Carlo simulations. The lower dotted line in Fig. 3 indicates the bound (38). Again, both correspond to the case $\beta = 2$. Note that, as expected, at high SNR the Cramér–Rao bound is attained asymptotically, i.e., the estimates are asymptotically efficient.

We may similarly compute Cramér–Rao bounds on the variance of unbiased smoothed estimates. In particular, using (36) with $\varepsilon[N]$ expressed in the form

$$\varepsilon[N] = \sum_{k=0}^{n-1} \left(y[k] - F^{(k-n)}_{s[k],s[k+1],\ldots,s[n-1]}(x[n])\right)^2 + \sum_{k=n}^{N} (y[k] - F^{(k-n)}(x[n]))^2$$

we get

$$\mathrm{var}\ \hat{x}[n \mid N] \geq \sigma_w^2 \cdot (1 - \beta^{-2}) \cdot \beta^{2(n-N)} [1 - \beta^{-2(N+1)}]^{-1}. \tag{39}$$

Furthermore, it is straightforward to show using (39) that the *average* estimation error in these estimates is bounded according to

$$\frac{1}{N+1} \sum_{n=0}^{N} \mathrm{var}\ \hat{x}[n \mid N] \geq \frac{\sigma_w^2}{N+1}. \tag{40}$$

The solid curve in Fig. 3 indicates the average SNR gain in the smoothed estimates over a range of SNR when $N + 1 = 50$ and $\beta = 2$, as determined from Monte Carlo simulations. The upper dotted line indicates the asymptotic average smoothing gain of

$$10 \log_{10}(N+1) \approx 17 \text{ dB}$$

computed from (40). Note that smoothing yields dramatically better signal estimates than filtering alone, particularly in the high SNR regime. Clearly, the backward-filtering stage of the smoothing algorithm is critical to achieving good signal estimation performance.

[1] In fact, since, as we will see, this bound turns out to be independent of the initial condition, this expectation can be interpreted as being with respect to both the observation noise and a randomly chosen initial condition.

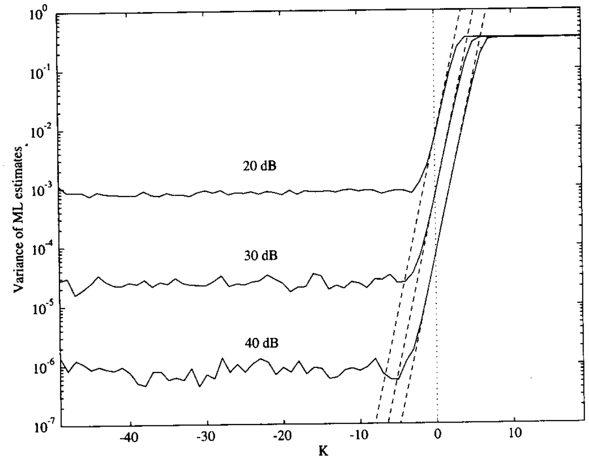

Fig. 4. Variance of ML estimates $\hat{x}[N+K \mid N]$ ($\beta = 2$, $N = 49$) as a function of K. The successively lower solid curves correspond to the actual estimator variance at SNR levels of 20, 30, and 40 dB, while the successively lower dashed curves depict the associated Cramér–Rao bounds.

Fig. 3 also illustrates the asymptotic efficiency of the ML smoothed estimates at high SNR.

Finally, the Cramér–Rao bound on the variance of unbiased K-step predictors is given by

$$\text{var } \hat{x}[N+K \mid N] \geq \sigma_w^2 \cdot (1-\beta^{-2}) \cdot \beta^{2K}[1-\beta^{-2(N+1)}]^{-1} \quad (41)$$

where we have used (36) with $\varepsilon[N]$ expressed in the form

$$\varepsilon[N] = \sum_{k=0}^{N} \left(y[k] - F^{(k-N-K)}_{s[k],s[k+1],\ldots,s[N+K-1]}(x[N+K])\right)^2.$$

Note that (41) implies that the error variance of unbiased predictors necessarily grows exponentially with K at a rate given by the Lyapunov exponent defined in (3). This is, of course, entirely consistent with "the sensitivity to initial condition" characteristic of chaotic maps.

It is useful to view the relationship between the variance characteristics associated with smoothing, filtering, and prediction. Indeed, we note that (37), (39), and (41) all share a common form. In Fig. 4, the solid curves depict the variance of the estimates $\hat{x}[N+K \mid N]$ as a function of K as determined from Monte Carlo simulations, while the dashed curves depict the associated Cramér–Rao bounds (37), (39), and (41). In all these experiments, we used $\beta = 2$, $N = 49$, and three representative SNR levels: 20, 30, and 40 dB. Note that smoothing applies for $K < 0$, filtering for $K = 0$, and prediction for $K > 0$.

From Fig. 4, we see that the ML estimate variance agrees with the associated Cramér–Rao bound only for a range of values of K near $K = 0$, although the extent of this range increases with SNR. This is consistent with the notion that such estimates are asymptotically efficient at high SNR.

The disagreement in the smoothing region ($K < 0$) can be attributed to the sign errors that are made in the filtering stage. Specifically, due to the small SNR gain during filtering (which does not depend on $N+1$, the sequence length), sign errors occur inevitably. The higher the original SNR, the lower the sign error rate, and the higher the attainable smoothing gain. Note that the initial SNR determines the sign error rate and thus the maximum achievable smoothing gain.

It is also important to note from Fig. 4 that despite what is suggested by the Cramér–Rao smoothing bound, the ML estimator does not yield a consistent estimator for the initial condition—in particular, for long data sets, the error variance in the estimate of the initial condition settles out at a threshold determined by the SNR of the observations. This interesting behavior, which would appear to be a consequence of the sensitive dependence characteristics of chaotic nonlinear dynamics, clearly warrants further analysis in future work.

In the prediction region ($K > 0$) the disagreement between the estimate variance and the respective bound is due to the inadequacy of the model used to compute the bound (41). Specifically, the constraint $|x[n]| < 1$ is not taken into account. This state-space constraint implies that exponential error growth at rate β^2 occurs only for small error values. For large values of K, the error variance reaches macroscopic proportions and saturates.

V. Concluding Remarks

We have derived ML estimation algorithms for a class of chaotic signals, including a form of chaotic white noise, derived from one-dimensional nonlinear maps. Their systematic description and broadband characteristics make such sequences potentially appealing in a number of applications involving signal synthesis. In addition, while this class of signals may be overly restrictive for many signal modeling and analysis applications, the results suggest a general estimator structure with close connections to traditional Kalman algorithms that may prove useful for much broader classes of chaotic signals.

It is also apparent that the ML estimation algorithms when implemented recursively constitute a dynamic programming algorithm. As such there are potentially interesting connections between the ML algorithms developed here and traditional Viterbi algorithms. While exploring such connections is beyond the scope of this work, it represents an interesting direction for future work and may provide additional insights.

Finally, this preliminary work raises several interesting open questions regarding optimal estimation of chaotic sequences. For example, while the ML estimators in this problem have some attractive characteristics—including that the estimates satisfy the tent map dynamics—it remains to be determined whether lower variance estimators for such sequences can be constructed, and, in fact, whether minimum-variance unbiased estimators exist. In terms of performance, a number of aspects of both the bias and variance characteristics of the ML estimates remain to be explored. For instance, as discussed in Section IV, the Cramér–Rao bounds do not predict the smoothing performance of the ML estimator that is obtained in practice for large data sets. Clearly, there is a need for some insightful analysis that would describe the asymptotic relationship between SNR and the error variance in the estimate of the initial condition.

Appendix
Proof of Lemma 1

Using induction, suppose

$$s_* = \underset{s[n]}{\text{argmin}}\ \varepsilon[N-1; s[0], s[1], \ldots, s[N-2], x]$$

is independent of x. Then because

$$\begin{aligned}\varepsilon[N; s[0], s[1], \cdots, s[N-1], x] \\ = \varepsilon[N-1; s[0], s[1], \cdots, s[N-2], v] + (y[N]-x)^2\end{aligned}$$

where

$$v = F^{-1}_{s[N-1]}(x)$$

we must also have

$$s_* = \underset{s[n]}{\text{argmin}}\ \varepsilon[N; s[0], s[1], \cdots, s[N-1], x].$$

Hence, it remains only to show

$$\begin{aligned} s_* &= \underset{s[n]}{\text{argmin}}\ \varepsilon[n+1; s[0], s[1], \cdots, s[n], x] \\ &= \text{sgn} \sum_{k=0}^{n} \beta^{2(k-n)} F^{(n-k)}_{s[n-1],\cdots,s[k]}(y[k]) \end{aligned} \tag{42}$$

to initiate the induction.

But since

$$\begin{aligned} &\varepsilon[n+1; s[0], s[1], \cdots, s[n], x] \\ &\quad = \varepsilon[n; s[0], s[1], \cdots, s[n-1], F^{-1}_{s[n]}(x)] + (y[n+1] - x)^2 \end{aligned}$$

we have

$$\begin{aligned} s_* &= \underset{s[n]}{\text{argmin}}\ \varepsilon[n+1; s[0], s[1], \ldots, s[n], x] \\ &= \underset{s[n]}{\text{argmin}} \sum_{k=0}^{n} \beta^{2(k-n)} (F^{(n-k)}_{s[n-1],\cdots,s[k]}(y[k]) - F^{-1}_{s[n]}(x))^2 \end{aligned} \tag{43}$$

Expanding the quadratic terms in (43) and noting from (6) that, for $x \in (-1, \beta - 1)$

$$|F^{-1}_{s[n]}(x)| = \frac{\beta - 1 - x}{\beta} \tag{44}$$

is independent of $s[n]$, we get

$$s_* = \underset{s[n]}{\text{argmax}} \left\{ F^{-1}_{s[n]}(x) \sum_{k=0}^{n} \beta^{2(k-n)} F^{(n-k)}_{s[n-1],\cdots,s[k]}(y[k]) \right\}. \tag{45}$$

In turn, since

$$s[n] = \text{sgn}\, F^{-1}_{s[n]}(x)$$

for $x \in (-1, \beta - 1)$ in accordance with (6), (45) implies s_* must be given by the right-hand side of (42).

Acknowledgment

The authors wish to thank the anonymous reviewer for several helpful comments and suggestions, including those on connections to the extended Kalman filter.

References

[1] M. D. Richard, "Probabilistic state estimation with discrete-time chaotic systems," RLE Tech. Rep. 571, MIT, Cambridge, MA, Mar. 1992.

[2] C. Myers, A. Singer, B. Shin, and E. Church, "Modeling chaotic systems with hidden Markov models," in *Proc. Int. Conf. on Acoustics Speech, and Signal Processing*, 1992.

[3] M. Casdagli, S. Eubank, J. D. Farmer, and J. Gibson, "State space reconstruction in the presence of noise," *Physica D*, vol. 51, pp. 52–98, 1991.

[4] A. Lasota and M. C. Mackey, *Probabilistic Properties of Deterministic Systems*. Cambridge, UK: Cambridge Univ. Press, 1985.

[5] A. V. Holden, Ed., *Chaos*. Princeton, NJ: Princeton Univ. Press, 1986.

[6] C. Myers, S. Kay, and M. Richard, "Signal separation for nonlinear dynamical systems," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1992.

[7] B. D. O. Anderson and J. B. Moore, *Optimal Filtering* Englewood Cliffs, NJ: Prentice-Hall, 1979.

Remarks on Linear and Nonlinear Filtering

Bernard Delyon

Abstract— **This communication tries to give some insight into relationships existing between Viterbi and the Forward–Backward algorithm (used in the context of Hidden Markov Models) on one hand and Kalman filtering and Rauch–Tung–Striebel smoothing on the other. We give a unifying view which shows how those algorithms are related and give an example of a nonlinear hybrid system that can be filtered through a mixed algorithm.**

Index Terms—**Nonlinear filtering, Viterbi algorithm.**

I. Introduction

In this communication, we consider estimation of the state of semi-Markov processes. These processes arise in two quite different fields: Hidden Markov Models (widely used for speech recognition, [1]) and Kalman–Bucy filtering; in the first case, the state-space is discrete (generally finite) while in the second one it is the Euclidean space. However, algorithms which are used have considerable similarities. Inspection of these similarities will lead us first to a generalization of Kalman–Bucy filtering in a particular extension to nonlinear systems and secondly to extend this model to a state-space which is mixed continuous-discrete.

A. Model

Semi-Markov processes (Y_n) are defined through their state-space representation (X_n, Y_n) (in some measurable space $\mathcal{X} \times \mathcal{Y}$) where Y is the observation and X is the hidden state; (X_n, Y_n) is a Markov chain and the assumption is that the transition from (X_n, Y_n) to (X_{n+1}, Y_{n+1}) does not depend on Y_n, which implies that (X_n) is itself a Markov chain. This process is thus characterized by its transition function $\Pi(x, x', y')$.

- In the case of discrete spaces

$$\Pi(x, x', y') = P(X_{n+1} = x', Y_{n+1} = y' | X_n = x)$$

- for continuous spaces

$$\Pi(x, x', y')dx'\, dy' = P(X_{n+1} \in dx', Y_{n+1} \in dy' | X_n = x)$$

and analogous formulas for mixed discrete/continuous spaces. An initial distribution is also given for X_0. The assumption imply that (Y_n) "depends" only on (X_{n-1}, X_n) in the sense that

$$\begin{aligned} P(Y_n | X_0, \cdots, X_N) &= P(Y_n | X_{n-1}, X_n) \\ &= \Pi(X_{n-1}, X_n, Y_n) / p(X_{n-1}, X_n) \end{aligned}$$

where $p(x, x')$ is the transition function of the chain (X_n)

$$p(x, x') = \int \Pi(x, x', y')\, dy'.$$

Thus the distribution of the process may also be realized by first running the Markov chain (X_n) with its own law and then drawing the random variables (Y_n) with their distribution conditioned on (X_{n-1}, X_n).

Manuscript received May 14, 1993; revised February 9, 1994.

The author is with IRISA-INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France.

IEEE Log Number 9407107.

0018-9448/95$04.00 © 1995 IEEE