

On-Line Estimation of Hidden Markov Model Parameters Based on the Kullback–Leibler Information Measure

Vikram Krishnamurthy, *Member, IEEE*, and John B. Moore, *Fellow, IEEE*

Abstract—In this paper, sequential or “on-line” hidden Markov model (HMM) signal processing schemes are derived and their performance illustrated in simulation studies. The on-line algorithms are sequential expectation maximization (EM) schemes and are derived by using stochastic approximations to maximize the Kullback–Leibler information measure. The schemes can be implemented either as filters or fixed-lag or sawtooth-lag smoothers. They yield estimates of the HMM parameters including transition probabilities, Markov state levels, and noise variance. In contrast to the off-line EM algorithm (Baum–Welch scheme) which uses the fixed-interval “forward-backward” scheme, the on-line schemes have significantly reduced memory requirements, improved convergence (as shown in simulations), and can estimate HMM parameters that vary slowly with time or undergo infrequent jump changes.

Using similar techniques we also derive on-line schemes to extract finite-state Markov chains imbedded in a mixture of white Gaussian noise (WGN) and deterministic signals of known functional form with unknown parameters. In particular, deterministic periodic signals with unknown and time-varying amplitudes and phases are considered. Simulations presented show that these schemes satisfactorily estimate the HMM parameters and also the time-varying amplitudes and phases.

I. INTRODUCTION

IN this paper, sequential or “on-line” hidden Markov model (HMM) signal processing schemes are derived and used to extract finite-state homogeneous Markov chains imbedded in white Gaussian noise. The on-line algorithms are sequential EM schemes and are derived by using stochastic approximations to maximize the Kullback–Leibler (KL) information measure. For sufficiently long data sequences, these schemes yield accurate estimates of the HMM parameters including transition probabilities, Markov state levels, and noise statistics.

Our work is motivated by two main applications:

1) Neurobiological signal processing applications: The transmission of channel currents in cell membranes is commonly modeled as a finite-state homogeneous Markov chain [1]. Such signals are invariably contaminated

by background noise from electrodes, amplifiers, etc. [2]. This background noise is very often modeled as white Gaussian noise (WGN). The Markov state levels in particular provide useful information to the neurobiologists about the dynamics of the propagation of nerve impulses. However, the transition probabilities and Markov levels slowly drift with time due to fluctuations in the experimental environment. Hence the necessity of on-line techniques.

2) Communication systems: In the modeling of communication systems, often discrete-level signals that are assumed to be homogeneous Markov are imbedded in noise, which for simplicity is assumed to be WGN. Often such chains are only locally stationary, i.e., the transition probabilities slowly vary with time. We have recently used similar on-line techniques to that developed in this paper in the “blind equalization” of time-varying FIR channels with Markov inputs [3].

In this paper we also derive on-line schemes to extract finite-state Markov chains imbedded in a mixture of WGN and deterministic signals of known functional form with unknown time-varying parameters. Two such examples of deterministic disturbances are considered:

1) Periodic or almost periodic disturbances with unknown phases and amplitudes of the components of the periodic signal. An example of such periodic disturbances with approximately known frequency components (fundamental and harmonics) is the periodic “hum” from the electricity mains which can be too costly to eliminate from some experimental environments.

2) Polynomial drift in the states of the Markov process. Such polynomial drift often occurs in cell-membrane channel measurements due to slow development of liquid junction potential arising from two different solutions of ionic compositions [4].

The problem of extracting finite-state Markov chains in WGN has been studied extensively in the off-line case using the expectation maximization (EM) algorithm [5]–[9]. The E step is noncausal, involving the forward-backward algorithm which is a “fixed-interval smoothing” scheme since the estimates are based on the entire data batch of observations. Repeated applications of the E step along with the M step (Baum–Welch [6]–[8] reestimation formulas) yield maximum likelihood estimates of the signal

Manuscript received May 15, 1991; revised September 16, 1992. The associate editor coordinating the review of this paper and approving it for publication was Prof. V. John Mathews. This work was supported in part by D.S.T.O. Australia and Boeing (BCAC), U.S.A.

The authors are with the Department of Systems Engineering, Research School of Physical Sciences and Engineering, Australian National University, Canberra ACT 2601, Australia.

IEEE Log Number 9209387.

model parameters. We have also recently developed similar off-line schemes based on the EM algorithm that effectively extract the Markov signals in a mixture of WGN and deterministic interferences [10].

A limitation of the off-line EM methods is the "curse of dimensionality" which arises because the required computational effort, speed, and memory requirements are in proportion to the square of the number of states of the finite Markov model. Memory requirements are also proportional to the length of data being processed. Hence the incentive to explore on-line (sequential) algorithms to i) seek improvements in terms of memory and computational speed and facilitate implementation on a small personal computer; ii) cope with time-varying HMM parameters such as transition probabilities and state levels. For example, the off-line schemes for extracting Markov chains from a mixture of WGN and deterministic interferences assume constant parameters and this is unrealistic. Often the amplitude phase of the ac mains and so the ac hum can change slowly with time. The on-line schemes proposed here deal with amplitudes and phases that vary slowly with time or undergo infrequent jump changes.

Sequential algorithms have been proposed in [11] for the case where the imbedded chain is independent (rather than Markov). Specifically, in [11] a "sequential" EM algorithm is presented which is based on maximizing the KL information measure. In this paper we develop on-line schemes for the case when the imbedded signal is a Markov chain. In such a case, it is necessary to replace the fixed-interval forward-backward variables of the EM algorithm by filtered or fixed-lag variables to obtain an effective reestimation scheme.

The concept of using fixed-lag smoothed estimates in on-line signal processing is well understood for linear stochastic signal models [12]. Also, its application to finite-state Markov models has been studied in [13]. The results in [13] evolved from discretizing Wonham's stochastic differential equation [14] yielding fixed-point smoothing equations leading to fixed-lag smoothing equations. The algorithms proposed in [13], [14] even for filtering (causal) case, appear subject to numerical instability problems. Here we develop smoothing schemes based on hidden Markov models that are numerically stable in that none of the variables can grow unstable. Also the algorithms in [13] assume complete knowledge of the signal model whereas here we adaptively estimate the signal model while achieving adaptive signal estimates.

Whereas Baum-Welch reestimation is done on a block of data (typically a few thousand observations) and uses fixed-interval estimates, our reestimation scheme is designed to be implemented recursively, thereby achieving on-line estimation. Thus in our on-line recursive estimation formulas the signal model estimates and signal statistics are updated at each time instant so that the KL measure of the updated model is greater than that of the model in the previous time instant. To improve convergence appropriate forgetting of past estimates is also introduced. This results in truly on-line reestimation in that for suffi-

ciently long data sequences only one pass is required through the data to learn the signal model. Simulations show that the on-line scheme has improved convergence compared to the off-line EM algorithm which has linear convergence.

For efficient implementation of the on-line schemes, we develop fixed- and "sawtooth"-lag schemes. The "sawtooth"-lag scheme is a variable-lag scheme with the lag varying in a sawtooth fashion. It is computationally more efficient than the fixed-lag scheme with the computational cost independent of the lag. Typically with a lag of 20 or so, our sawtooth-lag and fixed-lag smoothed estimates approach the optimal fixed interval smoothed estimate of the forward-backward algorithm. Simulations show that larger lags do not yield significantly improved smoothed estimates.

The paper is organized as follows: In Section II we formulate our signal model as a hidden Markov model, define our estimation objectives, and review standard HMM techniques. We derive our on-line HMM estimation scheme in Section III. In Section IV, fixed and sawtooth-lag smoothing schemes are presented. In Section V, we consider the case when in addition to WGN, deterministic interferences are also present. Simulation studies are presented in Section VI and some conclusions are drawn in Section VII.

II. HIDDEN MARKOV MODEL SIGNAL PROCESSING

In this section we describe the signal model, define our on-line estimation objectives, and briefly review the off-line EM algorithm.

A. Signal Model

Let s_k be a finite-state, discrete-time, homogeneous, first-order Markov process. Consequently, the state s_k at time k is one of a finite number N of states $q = q_1, q_2, \dots, q_N$. The transition probability matrix is $A = (a_{ij})$ where $a_{ij} = P(s_{k+1} = q_j | s_k = q_i)$. Of course $a_{ij} \geq 0$, $\sum_{j=1}^N a_{ij} = 1$, for each i . Let π denote the initial state probability vector: $\pi = (\pi_i)$, $\pi_i = P(s_1 = q_i)$.

Now assume that the Markov process s_k is hidden, that is, indirectly observed by measurements y_k where

$$y_k = s_k + w_k, \quad w_k \sim N[0, \sigma_w^2]. \quad (2.1)$$

Here w_k is zero-mean white Gaussian noise (WGN). Denote the sequence $\{s_1, s_2, \dots, s_k\}$ as S_k and the sequence $\{y_1, y_2, \dots, y_k\}$ by Y_k . Also let $Y_k^* = \{y_1, y_2, \dots, y_k\}$. Define the vector of parameterized probability densities which we shall loosely call symbol probabilities¹ as $b(\cdot) = (b_i(\cdot))$ where

$$\begin{aligned} b_i(y_k) &= f(y_k | s_k = q_i) \\ &= (\sqrt{2\pi}\sigma_w)^{-1} \exp(-(y_k - q_i)^2 / (2\sigma_w^2)) \end{aligned}$$

¹In some HMM applications, the observations belong to a finite alphabet set. Then $b_i(y_k)$ are probability mass functions. Hence the term symbol probabilities in the standard HMM literature.

are assumed invariant of k , with an independence property

$$f(y_k | s_k = q_i, s_{k-1} = q_j, Y_{k-1}) = f(y_k | s_k = q_i).$$

The HMM is denoted $\lambda = (A, q, \sigma_w^2, \pi)$.

For the rest of this paper, we shall use $f(\cdot)$ to denote probability densities (discrete as well as continuous).

In this paper we are concerned with recursively reestimating the HMM at each time instant. Denote the model estimate at time $k \geq 1$ as $\lambda_k = (A(k), q(k), \sigma_w^2(k))$. Our initial guess of the model is denoted as λ_0 . Let λ^0 denote the true model. Also given λ_k , let

$$\begin{aligned} b_i(k, y_k) &= f(y_k | s_k = q_i(k)) \\ &= (\sqrt{2\pi} \sigma_w(k))^{-1} \exp(-(y_k - q_i(k))^2 / (2\sigma_w^2(k))). \end{aligned}$$

Remark: We shall not consider the estimation of π in this paper. A reasonable estimate is: $\pi_i = \gamma_{1/1+\Delta, \lambda_0}(i)$ where $\gamma_{1/1+\Delta, \lambda_0}(i)$ is defined in (2.2). ■

Our Markov model is assumed to be first order for simplicity. Extending the theory to m th order Markov processes and semi-Markov processes is straightforward [20], although the associated computations are more formidable. Aggregation (reduced state) estimation techniques can be used as in [18] to simplify the computations.

B. Estimation Objectives

Given an ergodic sequence of observations, in this paper we provide on-line solutions to the following two problems. For comparison we also give the analogous objectives associated with standard off-line HMM processing which uses the EM algorithm.

1) MAP State Estimation:

On-line: Let $\Lambda_{k-1} = (\lambda_1, \lambda_2, \dots, \lambda_{k-1})$ denote the sequence of estimated models till time $k-1$. At time k , for a fixed-lag $\Delta \geq 0$, estimate the *a posteriori* probability densities

$$\begin{aligned} \gamma_{k|k+\Delta, \Lambda_{k-1}} &= (\gamma_{k|k+\Delta, \Lambda_{k-1}}(i)); \\ \gamma_{k|k+\Delta, \Lambda_{k-1}}(i) &\triangleq f(s_k = q_i | Y_{k+\Delta}, \Lambda_{k-1}). \end{aligned} \quad (2.2)$$

So for $\Delta = 0$, $\gamma_{k|k+\Delta, \Lambda_{k-1}}$ is the filtered Markov state estimate and for $\Delta > 0$ it is the fixed-lag smoothed estimate. From $\gamma_{k|k+\Delta, \Lambda_{k-1}}(i)$, maximum *a posteriori* (MAP) state estimates can be generated as

$$\hat{s}_k^{\text{MAP}} = q_j \quad \text{where } j = \arg \max_{1 \leq i \leq N} \gamma_{k|k+\Delta, \Lambda_{k-1}}(i). \quad (2.3)$$

We show how to compute $\gamma_{k|k+\Delta, \Lambda_{k-1}}$ in Section IV.

Off-line: Given the sequence Y_T and signal model λ , estimate $\gamma_{k|T}(i) \triangleq f(s_k = q_i | Y_T, \lambda)$. Again MAP state estimates can be generated similarly to (2.3).

$\gamma_{k|T}$ is computed from the forward-backward algorithm [5] which has computational complexity $O(N^2 T)$ per pass and requires a memory of NT because all the “forward” variables need to be stored.

2) Model Estimation:

On-line: Derive a sequential estimator yielding estimates λ_k of the true model λ^0 so as to maximize the KL information measure. That is we derive an on-line scheme by using the stochastic approximation algorithm on an off-line scheme that improves the KL measure with each iteration. For a model λ , the KL measure is defined as

$$J(\lambda) = E\{\log f(Y_k | \lambda) | \lambda^0\}. \quad (2.4)$$

In (2.4), $f(Y_k | \lambda)$ denotes the marginal probability density of Y_k and $E\{\cdot | \lambda^0\}$ denotes the statistical expectation with respect to the true model λ^0 . It is easily shown [19] using Jensen's inequality that $J(\lambda)$ has a unique maximum at $\lambda = \lambda^0$, as long as the following identifiability assumption holds: $f(Y_k | \lambda) = f(Y_k | \lambda^0)$ a.e. Y_k implies $\lambda = \lambda^0$. Therefore by estimating λ_k such that $J(\lambda_k)$ increases with k , for sufficiently large k our scheme yields $\lambda_k \rightarrow \lambda^0$. The on-line scheme derived here is analogous to the sequential EM algorithm developed in [15].

Off-line: Given Y_T and model estimate λ , reestimate the model as $\bar{\lambda}$ such that $f(Y_T | \bar{\lambda}) \geq f(Y_T | \lambda)$ with equality holding when λ is the ML estimate.

Off-line schemes for solving the two problems above using the EM algorithm are presented in [5]. In Sections III and IV, on-line solutions are presented.

3) Review of EM Algorithm (Baum-Welch Formulae): The estimation of the HMM in the off-line case involves the EM algorithm. It is an iterative algorithm; each iteration consists of an expectation step (E step) and a maximization step (M step).

E step: This involves computing $\mathcal{Q}(\lambda, \bar{\lambda})$ which is the expectation of the log of the likelihood function of the fully categorized data (see [11, pp. 84–91] for details), i.e.,

$$\mathcal{Q}(\lambda, \bar{\lambda}) \triangleq E\{\log f(Y_T, S_T | \bar{\lambda}) | Y_T, \lambda\}. \quad (2.5)$$

M step: This involves finding $\bar{\lambda}$ to maximize $\mathcal{Q}(\lambda, \bar{\lambda})$. By doing so it follows that $f(Y_T | \bar{\lambda}) \geq f(Y_T | \lambda)$ with equality holding when λ is the ML model. The M step yields the standard Baum-Welch equations for reestimating $\bar{\lambda} = (\bar{A}, \bar{q}, \bar{\sigma}_w^2, \bar{\pi})$, see [5], [6] for details.

III. SEQUENTIAL HMM ESTIMATION SCHEME

We first present an off-line algorithm for maximizing $J(\lambda)$. From this we straightforwardly derive a sequential algorithm. Some implementation aspects of the equations are then discussed.

Motivation: A direct recursive algorithm for estimating a HMM is obtained by using stochastic approximations to maximize the likelihood function. This yields:

$$\lambda_{k+1} = \lambda_k + \mathcal{J}^{-1}(\lambda_k) \partial / \partial \lambda_k \log f(Y_{k+1} | \lambda_k) \quad (3.1)$$

where $\mathcal{J}(\lambda_k) = \partial^2 \log f(Y_{k+1} | \lambda_k) / \partial \lambda_k^2$ is the Fisher information matrix (FIM) of the “incomplete” data. However, this algorithm cannot be easily implemented since it is not possible to express $\log f(Y_k | \lambda_k)$ or its derivatives analytically; even in the simpler hidden independent chain

case computing \mathcal{J} is very difficult (see [11, p. 207]). So we shall derive a sequential scheme based on computing FIM's and score vectors of the "complete" data which can be easily expressed analytically. Also, due to the block diagonal structure of the complete data FIM, our scheme requires only $O(N^2)$ computations at each time instant. Our algorithms are obtained by using stochastic approximations to maximize the KL information measure and are motivated by considerations leading to the EM algorithm.

A. Off-line Algorithm for Maximizing KL Measure

Let X_k denote the "complete" data, i.e., $X_k = (S_k, Y_k)$. Then (using a reasoning similar to that in [19]) we have

$$\begin{aligned} f(X_k | \lambda) &= f(X_k | Y_k, \lambda) f(Y_k | \lambda) \\ \log f(Y_k | \lambda) &= f(X_k | \lambda) - f(X_k | Y_k, \lambda). \end{aligned} \quad (3.2)$$

Taking conditional expectations with respect to the observations at a parameter value λ' yields

$$\begin{aligned} \log f(Y_k | \lambda) &= E\{\log f(X_k | \lambda) | Y_k, \lambda'\} \\ &\quad - E\{\log f(X_k | Y_k, \lambda) | Y_k, \lambda'\}. \end{aligned} \quad (3.3)$$

Denote

$$\begin{aligned} Q_k(\lambda', \lambda) &= E\{\log f(X_k | \lambda) | Y_k, \lambda'\}; \\ P_k(\lambda', \lambda) &= E\{\log f(X_k | Y_k, \lambda) | Y_k, \lambda'\}. \end{aligned} \quad (3.4)$$

Remark: $Q_k(\lambda', \lambda)$ is the conditional expectation of the log likelihood of the complete data until time k . Recall that in the EM algorithm described in Section II-B above, $Q(\lambda, \bar{\lambda})$ defined in (2.5) is simply $Q_T(\lambda_T, \bar{\lambda})$ above. ■

So by its definition in (2.4),

$$\begin{aligned} J(\lambda) &= E\{Q_k(\lambda', \lambda) | \lambda^0\} - E\{P_k(\lambda', \lambda) | \lambda^0\} \\ &= \bar{Q}(\lambda', \lambda) - \bar{P}(\lambda', \lambda) \quad (\text{say}). \end{aligned} \quad (3.5)$$

Then from Jensen's inequality, it follows (see [19, p. 1652])

$$\bar{Q}(\lambda', \lambda) > \bar{Q}(\lambda', \lambda') \quad \text{implies that} \quad J(\lambda) > J(\lambda'). \quad (3.6)$$

Based on (3.6), we have the following off-line algorithm for maximizing the KL measure:

$$\text{Estimate } \lambda^{l+1} \text{ as } \max_{\lambda} \bar{Q}(\lambda^l, \lambda) \quad (3.7)$$

where λ^l, λ^{l+1} are the models at the l th and $l+1$ th pass, respectively.

B. Sequential Algorithms

Because $\bar{Q}(\lambda', \lambda) = E\{Q_k(\lambda', \lambda) | \lambda^0\}$, the algorithm (3.7) gives the following sequential algorithm:

Let λ_0 be the initial model estimate.

Let $\Lambda_k = (\lambda_1, \dots, \lambda_k)$ denote the sequence of estimated HMMs till time k based on the observations Y_k .

Then given y_{k+1} at time $k+1$

$$\text{Estimate } \lambda_{k+1} \text{ as } = \max_{\lambda} Q_{k+1}(\Lambda_k, \lambda) \quad (3.8)$$

Finally, for a T -point observation sequence, estimate λ^0 as λ_T .

In (3.8)

$$Q_{k+1}(\Lambda_k, \lambda) \triangleq E\{\log f(Y_{k+1}, S_{k+1} | \lambda) | Y_{k+1}, \Lambda_k\}. \quad (3.9)$$

Remark 1: Given appropriate regularity, it can be shown (using results similar to those in [15]) that the algorithm (3.8) converges a.s. and in m.s. to the true parameter value. Also the limit distribution can be evaluated.

Remark 2: In the hidden independent chain case, algorithm (3.8) specializes to the "recursive" EM algorithm proposed in [11, p. 211]. ■

Sequential Variables: In order to implement algorithm (3.8), we define the following variables from which we evaluate $Q_{k+1}(\Lambda_k, \lambda)$:

$$\begin{aligned} \alpha_{k|\Lambda_{k-1}}(i) &= f(Y_k, s_k = q_i | \Lambda_{k-1}); \\ \beta_{t|k, \Lambda_{k-1}}(i) &= f(Y_{t+1}^k | s_t = q_i, \Lambda_{k-1}) \end{aligned} \quad (3.10)$$

$$\gamma_{t|k, \Lambda_{k-1}}(i) = f(s_t = q_i | Y_k, \Lambda_{k-1});$$

$$\xi_{t|k, \Lambda_{k-1}}(i, j) = f(s_t = q_i, s_{t+1} = q_j | Y_k, \Lambda_{k-1}). \quad (3.11)$$

The following two lemmas are proved in the Appendix.

Lemma 3.1: The sequential variables defined above are computed as follows:

$$\alpha_{k|\Lambda_{k-1}}(j) = \sum_{i=1}^N \alpha_{k-1|\Lambda_{k-2}}(i) a_{ij}(k-1) b_j(k-1, y_k), \quad \alpha_{1|\Lambda_0}(j) = \pi_j b_j(0, y_1) \quad (3.12)$$

$$\beta_{t|k, \Lambda_{k-1}}(i) = \sum_{j=1}^N a_{ij}(t-1) b_j(t-1, y_{t+1}) \beta_{t+1|k, \Lambda_{k-1}}(j), \quad \beta_{k|k, \Lambda_{k-1}}(i) = 1 \quad (3.13)$$

$$\gamma_{t|k, \Lambda_{k-1}}(i) = \frac{\alpha_{t, \Lambda_{k-1}}(i) \beta_{t|k, \Lambda_{k-1}}(i)}{\sum_{j=1}^N \alpha_{t, \Lambda_{k-1}}(j) \beta_{t|k, \Lambda_{k-1}}(j)} \quad (3.14)$$

$$\xi_{t|k, \Lambda_{k-1}}(i, j) = \frac{\alpha_{t, \Lambda_{k-1}}(i) a_{ij}(t-1) \beta_{t+1|k, \Lambda_{k-1}}(j) b_j(t-1, y_{t+1})}{\sum_i \sum_j \alpha_{t, \Lambda_{k-1}}(i) a_{ij}(t-1) \beta_{t+1|k, \Lambda_{k-1}}(j) b_j(t-1, y_{t+1})}. \quad (3.15)$$

Lemma 3.2: $\mathcal{Q}_{k+1}(\Lambda_k, \lambda)$ defined in (3.9) is evaluated as

$$\begin{aligned} \mathcal{Q}_{k+1}(\Lambda_k, \lambda) &= \sum_{t=1}^{k+1} \mathcal{L}_{t|k+1}(\Lambda_{t-1}, \lambda) \\ &+ \sum_{i=1}^N \gamma_{t|k+1, \Lambda_{t-1}}(i) \log \pi_i \end{aligned} \quad (3.16)$$

where

$$\begin{aligned} \mathcal{L}_{t|k+1}(\Lambda_{t-1}, \lambda) &= \sum_{i=1}^N \sum_{j=1}^N \zeta_{t|k+1, \Lambda_{t-1}}(i, j) \log a_{ij} \\ &+ \sum_{i=1}^N \gamma_{t|k+1, \Lambda_{t-1}}(i) \log \frac{1}{\sqrt{2\pi}\sigma_w} \\ &\cdot \exp\left(-\frac{(y_t - q_i)^2}{2\sigma_w^2}\right). \end{aligned} \quad (3.17)$$

Stochastic Approximation Algorithm: We now present a stochastic approximation algorithm to implement the sequential algorithm (3.8).

Let $I_{k+1}(\lambda_k)$ be the Fisher information matrix of the complete (fully categorized) data, i.e., $I_{k+1}(\lambda_k) = -\partial^2 \mathcal{Q}_{k+1}(\Lambda_k, \lambda) / \partial \lambda^2 \big|_{\lambda=\lambda_k}$.

Also let $S(\lambda_k, y_{k+1})$ be the score vector of the incomplete data defined as

$$S(\lambda_k, y_{k+1}) = \partial \mathcal{L}_{k+1|k+1}(\Lambda_k, \lambda) / \partial \lambda \big|_{\lambda=\lambda_k}.$$

Now consider the following theorem.

Theorem 3.3: Approximately given appropriate regularity, recursion (3.8) can be written as

$$\lambda_{k+1} = \lambda_k + (I_{k+1}(\lambda_k))^{-1} S(\lambda_k, y_{k+1}). \quad (3.18)$$

Remark 1: In [11, p. 207], where the hidden independent chain version of the theorem is considered, (3.18) is expressed in terms of the Fisher information matrix per k sample points. So the right-hand side of [11, p. 207, eq. 6.4.7] is multiplied by k .

Remark 2: Instead of I_{k+1}^{-1} , in general any sequence of positive numbers K_k can be used in the stochastic approximation as long as [19]

$$\lim_{k \rightarrow \infty} K_k = 0, \quad \sum_{k=1}^{\infty} K_k = \infty, \quad \sum_{k=1}^{\infty} K_k^2 < M < \infty. \quad (3.19)$$

Recently, other novel ways of choosing K_k have been proposed in [21].

Remark 3: The regularity conditions for convergence of (3.18) are presented in [11]. Verification of these conditions is not straightforward, see [11], [16] for details.

Remark 4: Effect of Initial Estimates: A theoretical analysis of the effect of initial estimates is beyond the scope of this paper. We have carried out extensive simulations with known variance σ_w up to 3.0 (with unit spacing of the true states) and $N = 4$ with initial estimates of q_i as far away as $3\sigma_w$ away from the true values and have not found initial conditions for which convergence to the

true model does not occur. We discuss the effect of initial estimates in Section IV-A. \blacksquare

Proof: By Taylor's series expansion

$$\begin{aligned} \mathcal{Q}_{k+1}(\Lambda_k, \lambda) &\approx \mathcal{Q}_{k+1}(\Lambda_k, \lambda_k) + (\lambda - \lambda_k)^T \\ &\cdot \frac{\partial \mathcal{Q}_{k+1}(\Lambda_k, \lambda)}{\partial \lambda} \bigg|_{\lambda=\lambda_k} \\ &+ \frac{1}{2} (\lambda - \lambda_k)^T \frac{\partial^2 \mathcal{Q}_{k+1}(\Lambda_k, \lambda)}{\partial \lambda^2} \bigg|_{\lambda=\lambda_k} \\ &\cdot (\lambda - \lambda_k). \end{aligned}$$

Maximizing this with respect to λ and calling the maximum point λ_{k+1} , we have

$$\lambda_{k+1} = \lambda_k + I_{k+1}^{-1}(\lambda_k) \frac{\partial \mathcal{Q}_{k+1}(\Lambda_k, \lambda)}{\partial \lambda} \bigg|_{\lambda=\lambda_k} \quad (3.20)$$

which corresponds to one step of the Newton Raphson algorithm, initialized at λ_k .

In order to proceed with the proof, recursive relationships for \mathcal{Q}_{k+1} and its derivatives are required.

Recall from (3.16) in Lemma (3.2) that

$$\mathcal{Q}_{k+1}(\Lambda_k, \lambda) = \mathcal{Q}_k(\Lambda_{k-1}, \lambda) + \mathcal{L}_{k+1|k+1}(\Lambda_k, \lambda). \quad (3.21)$$

So,

$$\begin{aligned} \frac{\partial \mathcal{Q}_{k+1}(\Lambda_k, \lambda)}{\partial \lambda} \bigg|_{\lambda=\lambda_k} &= \frac{\partial \mathcal{Q}_k(\Lambda_{k-1}, \lambda)}{\partial \lambda} \bigg|_{\lambda=\lambda_k} \\ &+ \frac{\partial \mathcal{L}_{k+1|k+1}(\Lambda_k, \lambda)}{\partial \lambda} \bigg|_{\lambda=\lambda_k}. \end{aligned} \quad (3.22)$$

Now we assume that λ_k minimizes $\mathcal{Q}_k(\Lambda_{k-1}, \lambda)$, so that $\partial \mathcal{Q}_k(\Lambda_{k-1}, \lambda) / \partial \lambda \big|_{\lambda=\lambda_k} = 0$. So (3.22) yields

$$\begin{aligned} \frac{\partial \mathcal{Q}_{k+1}(\Lambda_k, \lambda)}{\partial \lambda} \bigg|_{\lambda=\lambda_k} &= \frac{\partial \mathcal{L}_{k+1|k+1}(\Lambda_k, \lambda)}{\partial \lambda} \bigg|_{\lambda=\lambda_k} \\ &= S(\lambda_k, y_{k+1}) \text{ by definition.} \end{aligned} \quad (3.23)$$

Substituting this into (3.20) yields (3.18). \square

Remark: The above derivation is also analogous to that of the well-known recursive prediction error (RPE) equations for model estimation [22, pp. 328–334]. Of course in RPE the cost function is the prediction error while here it is the KL measure. This insight arises from our current work in developing RPE schemes for adaptive estimation of HMM's [23]. \blacksquare

Theorem 3.4: For known A , recursion (3.8) can exactly be implemented as (3.18).

Proof: The proof is the same as [15, p. 265, theorem 3]. \square

Remark: As proved in [15], Theorem 3.4 holds as long as the distribution of w_k belongs to the exponential family of distributions (which includes Gaussian distributions).

case computing \mathcal{J} is very difficult (see [11, p. 207]). So we shall derive a sequential scheme based on computing FIM's and score vectors of the "complete" data which can be easily expressed analytically. Also, due to the block diagonal structure of the complete data FIM, our scheme requires only $O(N^2)$ computations at each time instant. Our algorithms are obtained by using stochastic approximations to maximize the KL information measure and are motivated by considerations leading to the EM algorithm.

A. Off-line Algorithm for Maximizing KL Measure

Let X_k denote the "complete" data, i.e., $X_k = (S_k, Y_k)$. Then (using a reasoning similar to that in [19]) we have

$$\begin{aligned} f(X_k | \lambda) &= f(X_k | Y_k, \lambda) f(Y_k | \lambda) \\ \log f(Y_k | \lambda) &= f(X_k | \lambda) - f(X_k | Y_k, \lambda). \end{aligned} \quad (3.2)$$

Taking conditional expectations with respect to the observations at a parameter value λ' yields

$$\begin{aligned} \log f(Y_k | \lambda) &= E\{\log f(X_k | \lambda) | Y_k, \lambda'\} \\ &\quad - E\{\log f(X_k | Y_k, \lambda) | Y_k, \lambda'\}. \end{aligned} \quad (3.3)$$

Denote

$$\begin{aligned} Q_k(\lambda', \lambda) &= E\{\log f(X_k | \lambda) | Y_k, \lambda'\}; \\ P_k(\lambda', \lambda) &= E\{\log f(X_k | Y_k, \lambda) | Y_k, \lambda'\}. \end{aligned} \quad (3.4)$$

Remark: $Q_k(\lambda', \lambda)$ is the conditional expectation of the log likelihood of the complete data until time k . Recall that in the EM algorithm described in Section II-B above, $Q(\lambda, \bar{\lambda})$ defined in (2.5) is simply $Q_T(\lambda_T, \bar{\lambda})$ above. ■

So by its definition in (2.4),

$$\begin{aligned} J(\lambda) &= E\{Q_k(\lambda', \lambda) | \lambda^0\} - E\{P_k(\lambda', \lambda) | \lambda^0\} \\ &= \bar{Q}(\lambda', \lambda) - \bar{P}(\lambda', \lambda) \quad (\text{say}). \end{aligned} \quad (3.5)$$

Then from Jensen's inequality, it follows (see [19, p. 1652])

$$\bar{Q}(\lambda', \lambda) > \bar{Q}(\lambda', \lambda') \quad \text{implies that} \quad J(\lambda) > J(\lambda'). \quad (3.6)$$

Based on (3.6), we have the following off-line algorithm for maximizing the KL measure:

$$\text{Estimate } \lambda^{l+1} \text{ as } \max_{\lambda} \bar{Q}(\lambda^l, \lambda) \quad (3.7)$$

where λ^l, λ^{l+1} are the models at the l th and $l+1$ th pass, respectively.

B. Sequential Algorithms

Because $\bar{Q}(\lambda', \lambda) = E\{Q_k(\lambda', \lambda) | \lambda^0\}$, the algorithm (3.7) gives the following sequential algorithm:

Let λ_0 be the initial model estimate.

Let $\Lambda_k = (\lambda_1, \dots, \lambda_k)$ denote the sequence of estimated HMMs till time k based on the observations Y_k .

Then given y_{k+1} at time $k+1$

$$\text{Estimate } \lambda_{k+1} \text{ as } \max_{\lambda} Q_{k+1}(\Lambda_k, \lambda) \quad (3.8)$$

Finally, for a T -point observation sequence, estimate λ^0 as λ_T .

In (3.8)

$$Q_{k+1}(\Lambda_k, \lambda) \triangleq E\{\log f(Y_{k+1}, S_{k+1} | \lambda) | Y_{k+1}, \Lambda_k\}. \quad (3.9)$$

Remark 1: Given appropriate regularity, it can be shown (using results similar to those in [15]) that the algorithm (3.8) converges a.s. and in m.s. to the true parameter value. Also the limit distribution can be evaluated.

Remark 2: In the hidden independent chain case, algorithm (3.8) specializes to the "recursive" EM algorithm proposed in [11, p. 211]. ■

Sequential Variables: In order to implement algorithm (3.8), we define the following variables from which we evaluate $Q_{k+1}(\Lambda_k, \lambda)$:

$$\begin{aligned} \alpha_{k|\Lambda_{k-1}}(i) &= f(Y_k, s_k = q_i | \Lambda_{k-1}); \\ \beta_{t|k, \Lambda_{k-1}}(i) &= f(Y_{t+1}^k | s_t = q_i, \Lambda_{k-1}) \end{aligned} \quad (3.10)$$

$$\begin{aligned} \gamma_{t|k, \Lambda_{k-1}}(i) &= f(s_t = q_i | Y_k, \Lambda_{k-1}); \\ \xi_{t|k, \Lambda_{k-1}}(i, j) &= f(s_t = q_i, s_{t-1} = q_j | Y_k, \Lambda_{k-1}). \end{aligned} \quad (3.11)$$

The following two lemmas are proved in the Appendix.

Lemma 3.1: The sequential variables defined above are computed as follows:

$$\alpha_{k|\Lambda_{k-1}}(j) = \sum_{i=1}^N \alpha_{k-1|\Lambda_{k-2}}(i) a_{ij}(k-1) b_j(k-1, y_k), \quad \alpha_{1|\Lambda_0}(j) = \pi_j b_j(0, y_1) \quad (3.12)$$

$$\beta_{t|k, \Lambda_{k-1}}(i) = \sum_{j=1}^N a_{ij}(t-1) b_j(t-1, y_{t+1}) \beta_{t+1|k, \Lambda_{k-1}}(j), \quad \beta_{k|k, \Lambda_{k-1}}(i) = 1 \quad (3.13)$$

$$\gamma_{t|k, \Lambda_{k-1}}(i) = \frac{\alpha_{t, \Lambda_{k-1}}(i) \beta_{t|k, \Lambda_{k-1}}(i)}{\sum_{j=1}^N \alpha_{t, \Lambda_{k-1}}(j) \beta_{t|k, \Lambda_{k-1}}(j)} \quad (3.14)$$

$$\xi_{t|k, \Lambda_{k-1}}(i, j) = \frac{\alpha_{t, \Lambda_{k-1}}(i) a_{ij}(t-1) \beta_{t+1|k, \Lambda_{k-1}}(j) b_j(t-1, y_{t+1})}{\sum_i \sum_j \alpha_{t, \Lambda_{k-1}}(i) a_{ij}(t-1) \beta_{t+1|k, \Lambda_{k-1}}(j) b_j(t-1, y_{t+1})}. \quad (3.15)$$

C. On-Line Reestimation Equations

We now explicitly obtain the HMM sequential reestimation equations by obtaining expressions for $I_{k+1}(\lambda_k)$ and $S(\lambda_k, y_{k+1})$ in (3.18), Theorem 3.3.

Evaluation of Fisher Information Matrix I_{k+1} : Because each term in (3.17) depends on only one of the model parameters,

$$I_{k+1}(\lambda_k) = \text{blockdiag}(I_{k+1}^A(\lambda_k), I_{k+1}^Q(\lambda_k), I_{k+1}^\Sigma(\lambda_k))$$

where $I_{k+1}^A(\lambda_k)$, $I_{k+1}^Q(\lambda_k)$, and $I_{k+1}^\Sigma(\lambda_k)$ are matrices obtained by taking derivatives with respect to the transition probabilities, state levels, and noise variance, respectively.

Evaluation of $I_{k+1}^A(\lambda_k)$: Notice that because $\sum_j a_{ij} = 1$, only $N - 1$ elements of each row of A need to be estimated. For each row let l_i be such that a_{il_i} is the element not estimated, i.e., it is not obtained as

$$a_{il_i} = 1 - \sum_{j=1}^N a_{ij}, \quad j \neq l_i.$$

So $I_{k+1}^A(\lambda)$ is of the form

$$I_{k+1}^A(\lambda) = \text{blockdiag}(P_1, \dots, P_N),$$

$$P_i = M^{(i)} + C^{(i)} C^{(i)T}$$

$$M^{(i)} = \text{diag}(\mu_j^{(i)}), \quad 1 \leq j \leq N; \quad j \neq l_i$$

with

$$\mu_j^{(i)} = \left(\frac{\sum_{t=1}^{k+1} \zeta_{t|k+1, \Lambda_{t-1}}(i, j)}{a_{ij}^2} \right) \quad (3.24)$$

and $C = (c_i)$ is a constant $N - 1$ element column vector with elements

$$c_i = \left(\frac{\sum_{t=1}^{k+1} \zeta_{t|k+1, \Lambda_{t-1}}(i, l_i)}{a_{il_i}^2} \right)^{1/2}. \quad (3.25)$$

To find I_{k+1}^{A-1} we use the matrix inversion lemma and get

$$P_i^{-1} = M^{(i)-1} - \left(\frac{1}{c^{(i)2}} + \sum_{j \neq l_i} \frac{1}{\mu_j^{(i)}} \right)^{-1} F^{(i)} F^{(i)T} \quad (3.26)$$

where $F^{(i)}$ is a $N - 1$ element column vector with elements $(\mu_j^{(i)})^{-1}$, $j = 1, \dots, N$, $j \neq l_i$.

Evaluation of $I_{k+1}^Q(\lambda)$: $I_{k+1}^Q(\lambda)$ is a diagonal matrix with elements $-\partial^2 \mathcal{Q}_{k+1}(\Lambda_k, \lambda) / \partial q_i^2$. Evaluating this yields

$$I_{k+1}^Q(\lambda) = \text{diag} \left(\frac{\sum_{t=1}^{k+1} \gamma_{t|k+1, \Lambda_{t-1}}(1)}{\sigma_w^2}, \dots, \frac{\sum_{t=1}^{k+1} \gamma_{t|k+1, \Lambda_{t-1}}(N)}{\sigma_w^2} \right). \quad (3.27)$$

Evaluation of $I_{k+1}^\Sigma(\lambda)$: $I_{k+1}^\Sigma(\lambda)$ is a scalar and is evaluated as

$$I_{k+1}^\Sigma(\lambda) = (k+1)/(2\sigma_w^4). \quad (3.28)$$

Evaluation of Score Vector S : We evaluate the score as

$$\partial \mathcal{L}_{k+1|k+1}(\Lambda_k, \lambda) / \partial \lambda|_{\lambda=\lambda_k}.$$

Evaluating $\partial \mathcal{L}_{k+1|k+1}(\Lambda_k, \lambda) / \partial \lambda$ yields:

$$S(\lambda, y_{k+1}) = [S_A(\lambda, y_{k+1}), S_Q(\lambda, y_{k+1}), S_\Sigma(\lambda, y_{k+1})]'$$

where $S_A(\cdot)$ is a vector with elements $\partial \mathcal{L}_{k+1|k+1} / \partial a_{ij}$, $j \neq l_i$, $S_Q(\cdot)$ is a vector with elements $\partial \mathcal{L}_{k+1|k+1} / \partial q_i$, and $S_\Sigma(\cdot)$ is the scalar of $\partial \mathcal{L}_{k+1|k+1} / \partial \sigma_w^2$

$$S_A(\lambda, y_{k+1}) = (S_A(1), S_A(2), \dots, S_A(N))'$$

$$S_A(i) = (s_A(i, 1), \dots, s_A(i, l_i - 1),$$

$$s_A(i, l_i + 1), \dots, s_A(i, N)) \quad (3.29)$$

where

$$s_A(i, j) = g_j^{(i)} - g_{l_i}^{(i)}, \quad j \neq l_i$$

$$g_j^{(i)} = \frac{\zeta_{k+1|k+1, \Lambda_k}(i, j)}{a_{ij}} \quad \forall j \quad (3.30)$$

$$S_Q(\lambda, y_{k+1}) = (\gamma_{k+1|k+1, \Lambda_k}(1)(y_{k+1} - q_1)/\sigma_w^2, \dots, \gamma_{k+1|k+1, \Lambda_k}(N)(y_{k+1} - q_N)/\sigma_w^2) \quad (3.31)$$

$$S_\Sigma(\lambda, y_{k+1}) = \frac{1}{2\sigma_w^4} \sum_i \gamma_{k+1|k+1, \Lambda_k}(i)(y_{k+1} - q_i)^2 - \frac{1}{2\sigma_w^2}. \quad (3.32)$$

Summary of On-line Reestimation Equations: Substituting for $I_{k+1}(\cdot)^{-1}$ from (3.26), (3.27), and (3.28) and for $S(\cdot)$ from (3.29), (3.31), and (3.32) into the algorithm (3.18) yields the on-line reestimation formulas which we now summarize:

Let $\lambda_k = (A(k), q(k), \sigma_w^2(k))$ be the model estimate at time k . Then λ_{k+1} is calculated as

$$a_{ij}(k+1) = a_{ij}(k) + \frac{1}{\mu_j^{(i)}} \left(g_j^{(i)} - \frac{\sum_{h=1}^N g_h^{(i)} / \mu_h^{(i)}}{\sum_{h=1}^N (\mu_h^{(i)})^{-1}} \right), \quad (3.33)$$

$$i = 1, \dots, N; \quad j = 1, \dots, N;$$

$$q_i(k+1) = q_i(k) + \frac{\gamma_{k+1|k+1, \Lambda_k}(i)(y_{k+1} - q_i(k))}{\sum_{t=1}^{k+1} \gamma_{t|k+1, \Lambda_{t-1}}(i)} \quad (3.34)$$

$$\sigma_w^2(k+1) = \sigma_w^2(k) + \frac{\sum_{i=1}^N \gamma_{k+1|k+1, \Lambda_k}(i) ((y_{k+1} - q_i(k))^2 - \sigma_w^2(k))}{k+1} \quad (3.35)$$

where $g_h^{(i)}$ is defined in (3.30) and $\mu_h^{(i)}$ is defined in (3.24).

Remark 1: Notice that the above reestimation formula for the transition probabilities (3.33) ensures that $\sum_j a_{ij}(k) = 1$ for all k . This is so since the incremental term in the recursion (3.33) when summed over j , equals zero. So if the initial transition probability matrix at time $k = 1$ has its rows adding up to one, the estimate for any k will also satisfy this constraint.

Remark 2: On-line reestimation formulas for arbitrary (not necessarily Gaussian) symbol probabilities can be derived similarly: The second term in (3.17) is then

$$\sum_i \gamma_{t|k+1, \Lambda_{t-1}}(i) \log b_i(k, y_{k+1}).$$

Evaluating the Fisher information matrix and score vector and substituting in (3.18) yields the reestimation formula for the symbol probabilities. However, we shall not deal with arbitrary symbol probabilities in this paper.

Remark 3: As expected, (3.34) and (3.35) specialize to the equations in [11, p. 210] when the imbedded chain is independent rather than Markov. ■

Of course, because the noise is Gaussian, once the state levels $q_i^{(k)}$ are known then the symbol probability densities are

$$b_i(k, y_{k+1}) = \frac{1}{\sqrt{2\pi} \sigma_w(k)} \exp \left(-\frac{(y_{k+1} - q_i(k))^2}{2\sigma_w^2(k)} \right),$$

$$i = 1, \dots, N. \quad (3.36)$$

D. Implementation Aspects

We discuss some implementation aspects which enhance performance on finite length data.

1) **Fixed-Lag Implementation:** The implementation of the on-line algorithm requires calculation of the fixed-interval variables $\zeta_{t|k+1, \Lambda_{t-1}}$ and $\gamma_{t|k+1, \Lambda_{t-1}}$, $t \leq k+1$. This is not computationally feasible. Instead we approximate these fixed interval variables by fixed-lag variables $\zeta_{t|k+\Delta, \Lambda_{t-1}}$ and $\gamma_{t|k+\Delta, \Lambda_{t-1}}$, $\Delta \geq 0$. We show how to evaluate these fixed-lag variables in the next section.

2) **Exponential Forgetting:** As we shall show in simulations, to track time-varying parameters, it is advantageous to do exponential forgetting which reduces the effect of past observations relative to the new input data. We propose the following scheme which is analogous to that proposed in [19]:

Let $0 < \rho \leq 1$ be the forgetting factor. ($\rho = 1$ means that there is no forgetting.) In the evaluation of $\mu_h^{(i)}$ in (3.24) use $\sum_{t=1}^{k+1} \rho^{k+1-t} \zeta_{t|k+1, \Lambda_{t-1}}(i, j)$ (or its fixed-lag approximation) instead of $\zeta_{t|k+1, \Lambda_{t-1}}(i, j)$. Similarly, use $\sum_{t=1}^{k+1} \rho^{k+1-t} \gamma_{t|k+1, \Lambda_{t-1}}(i)$ (or its fixed-lag approximation) instead of $\sum_{t=1}^{k+1} \gamma_{t|k+1, \Lambda_{t-1}}(i)$ in (3.34). Also use $\sum_{t=1}^{k+1} \rho^{k+1-t}$ in the denominator of (3.35).

IV. FIXED AND SAWTOOTH-LAG SMOOTHING

As described above, we approximate the fixed interval variables in the reestimation equations by $\gamma_{t|t+\Delta, \Lambda_{t-1}}$ and $\zeta_{t|t+\Delta, \Lambda_{t-1}}$. Here we show how to evaluate these vectors using fixed-lag and sawtooth-lag smoothing schemes. The sawtooth-lag smoothing turns out to be computationally more attractive.

Both fixed and sawtooth-lag schemes are based on the standard backward recursions reviewed in [5]. Besides being required in the on-line reestimation equations, these smoothing schemes are of independent interest since they significantly decrease the memory requirement of storing T vectors to compute γ in the standard forward-backward scheme. For sufficiently large lags they can be used instead of the fixed interval forward-backward algorithm in standard HMM processing.

A. Fixed-Lag Estimation

The fixed-lag variable $\beta_{k|k+\Delta, \Lambda_{k-1}}$ is calculated as follows:

$$\beta_{k|k+\Delta, \Lambda_{k-1}} = \mathbf{A}(k-1) \mathbf{B}(k-1, k+1) \cdots \mathbf{A}(k-1) \mathbf{B}(k-1, k+\Delta) \mathbf{1} \quad (4.1)$$

where $\mathbf{B}(k-1, k+1)$ is the diagonal matrix $\text{diag}(b_1(k-1, y_{k+1}), \dots, b_N(k-1, y_{k+1}))$ and $\mathbf{1}$ is the N vector with elements 1.

The fixed-lag *a posteriori* probabilities associated with Problem 2 can be computed similarly to (3.14) as

$$\gamma_{k|k+\Delta, \Lambda_{k-1}}(i) = \frac{\alpha_{k, \Lambda_{k-1}}(i) \beta_{k|k+\Delta, \Lambda_{k-1}}(i)}{\sum_{i=1}^N \alpha_{k, \Lambda_{k-1}}(i) \beta_{k|k+\Delta, \Lambda_{k-1}}(i)},$$

$$i = 1, 2, \dots, N. \quad (4.2)$$

Then using (2.3) the fixed-lag MAP state sequence estimates are obtained.

Notice from (4.1) that $\beta_{k|k+\Delta, \Lambda_{k-1}}$ can be computed forwards in time. This has the advantage that only one vector $\beta_{k|k+\Delta, \Lambda_{k-1}}$ needs to be stored to compute $\gamma_{k|k+\Delta, \Lambda_{k-1}}$. Thus the computational cost of the fixed-lag backward scheme is ΔC_F and the memory requirement is N ; in comparison the computational cost of the forward-backward scheme is C_F and memory required is NT .

B. Sawtooth-Lag Estimation

In the sawtooth-lag scheme we work with a lag varying from a fixed lag Δ_{\min} to another of $\Delta_{\max} \geq 2\Delta_{\min}$ maintaining an average lag of $(\Delta_{\min} + \Delta_{\max})/2$. Let us consider k belonging to a subinterval $[k_1, k_1 + \Delta_{\max} - \Delta_{\min}]$. Now instead of using the fixed-lag variable $\beta_{k|k+\Delta, \Lambda_{k-1}}$,

we use the sawtooth-lag variable $\beta_{k|k_1+\Delta, \lambda_{k-1}}$. That is, we compute β for a fixed model λ_{k_1-1} starting at $k_1 + \Delta_{\max}$.

$\beta_{k|k_1+\Delta_{\max}, \lambda_{k-1}}$, $k \in [k_1, k_1 + \Delta_{\max} - \Delta_{\min}]$, is computed using a similar backward recursion to (3.13):

$$\beta_{k|k_1+\Delta_{\max}, \lambda_{k-1}}(i) = \sum_{j=1}^N a_{ij}(k_1-1) b_j(k_1-1, y_{k+1}) \cdot \beta_{k+1|k_1+\Delta_{\max}, \lambda_{k-1}}(j) \quad (4.3)$$

initialized by $\beta_{k_1+\Delta_{\max}, \lambda_{k_1-1}}(i) = 1$. Notice that out of Δ_{\max} values of β computed above, only $\Delta_{\max} - \Delta_{\min}$ values (namely, from time k_1 , $k_1 + \Delta_{\max} - \Delta_{\min}$) are used and Δ_{\min} values (from time $k_1 + \Delta_{\max} - \Delta_{\min}$, $k_1 + \Delta_{\max}$) are discarded.

There is a tradeoff in selecting Δ_{\max} and Δ_{\min} . We must choose Δ_{\min} sufficiently large so that the initialization of β at time $k_1 + \Delta_{\max}$ has negligible effect at time $k_1 + \Delta_{\max} - \Delta_{\min}$. On the other hand, Δ_{\max} should be sufficiently small so that the estimated model at time $k_1 + \Delta_{\max}$ is not too different from that at time k_1 .

The sawtooth-lag scheme is efficient because the computational cost for T observations is $\Delta_{\max} C_F / (\Delta_{\max} - \Delta_{\min})$ and the memory required is $(\Delta_{\max} - \Delta_{\min})N$. So there is the possibility of trading computation time with memory, e.g., Choosing $\Delta_{\max} = 2\Delta_{\min}$, the memory requirements are minimal ($\Delta_{\min}N$) and the computation effort required is twice that for the forward-backward procedure. By choosing $\Delta_{\max} \gg \Delta_{\min}$, the computation effort required approaches the forward-backward procedure and the memory requirements are increased in proportion to Δ_{\max} .

V. ELIMINATION OF DETERMINISTIC INTERFERENCES

We now illustrate an application of the proposed on-line techniques. We consider the case where in addition to white Gaussian noise (WGN), the Markov process s_k is also corrupted by a deterministic signal of known form with unknown parameters.

Let $p_k(\Theta)$, parametrized by $\Theta \in \mathbb{R}^R$ denote the deterministic disturbance. We assume that the functional form of p_k is known but the parameter vector $\Theta = (\theta_1, \dots, \theta_R)$ is unknown. In addition, it is assumed that Θ is slowly time varying or undergoes infrequent jump changes. We consider two such examples of deterministic disturbances:

1) Periodic or almost periodic disturbances (with known frequency components ω_n)

$$p_k(\Theta) = \sum_{n=1}^p a_n \sin(\omega_n k + \phi_n) \quad \text{and}$$

$$\Theta = (a_1, \dots, a_p, \phi_1, \dots, \phi_p)^2$$

2) Drift in the states of the Markov process, $p_k(\Theta) = \Sigma_{n=1}^p a_n k^n$ and $\Theta = (a_1, \dots, a_p)$.

²Expressing $p_k(\Theta)$ as $\Sigma_n c_n \sin(\omega_n k) + d_n \cos(\omega_n k)$ does not simplify the resulting reestimation equations in any way.

Signal Model: We assume that s_k is hidden, that is indirectly observed by measurements y_k :

$$y_k = s_k + p_k + w_k \quad (5.1)$$

where s_k , w_k are defined in Section II-A. The resulting HMM is denoted $\lambda = (A, \Theta, q, \pi)$. For notational convenience we assume that σ_w^2 is known.

On-line Reestimation: Again given the above signal model, two interrelated problems can be posed and solved. The first problem is identical to that in Section II-B and its solution is identical to that in Section IV, with $b_i(k-1, y_k)$ defined as follows:

$$\begin{aligned} b_i(k-1, y_k) &= f(y_k - p_k(\Theta(k-1)) | s_k = q_i(k-1)) \\ &= \frac{1}{\sqrt{2\pi}\sigma_w} \exp \\ &\quad \cdot \left(-\frac{(y_k - q_i(k-1) - p_k(\Theta(k-1)))^2}{2\sigma_w^2} \right). \end{aligned} \quad (5.2)$$

Problem 2 now involves on-line reestimation based on maximizing the KL measure of the estimated model $\lambda_k = (A(k), \Theta(k), q(k))$.

Derivation of On-line Reestimation Equations: We now obtain expressions for $I_{k+1}(\lambda_k)$, $S(\lambda_k, y_{k+1})$ for the periodic disturbance case then derive the reestimation equations using (3.18) of Theorem 4.1. The reestimation equations in the case of polynomial drift can be derived similarly; for brevity we omit the proofs.

Evaluation of Fisher Information Matrix: In this case Lemma 3.2 holds with

$$\begin{aligned} \mathcal{L}_{t|k+1}(\Lambda_{t-1}, \lambda) &= \sum_{i=1}^N \sum_{j=1}^N \xi_{t|k+1, \Lambda_{t-1}}(i, j) \log a_{ij} \\ &\quad + \sum_{i=1}^N \gamma_{t|k+1, \Lambda_{t-1}}(i) \log \frac{1}{\sqrt{2\pi}\sigma_w} \\ &\quad \cdot \exp \left(-\frac{(y_t - q_i - p_t(\Theta))^2}{2\sigma_w^2} \right). \end{aligned} \quad (5.3)$$

Then evaluating $-\partial^2 \mathcal{Q}_{k+1}(\Lambda_k, \lambda) / \partial \lambda^2$ yields

$$I_{k+1}(\lambda) = \text{diag}(I_{k+1}^A(\lambda), I_{k+1}^\Theta(\lambda), I_{k+1}^q(\lambda))$$

where $I_{k+1}^A(\lambda)$ is computed in (3.26), $I_{k+1}^\Theta(\lambda)$ is computed in (3.27) and $I_{k+1}^q(\lambda) = \text{diag}(I_{k+1}^a, I_{k+1}^\phi)$,

$$\begin{aligned} I_{k+1}^a &= \text{diag} \left(-\left(\sum_{t=1}^{k+1} \sin^2(\omega_1 t + \phi_1) / \sigma_w^2 \right), \right. \\ &\quad \left. \dots, -\left(\sum_{t=1}^{k+1} \sin^2(\omega_p t + \phi_p) / \sigma_w^2 \right) \right) \end{aligned} \quad (5.4)$$

$I_k^\phi = \text{diag}(I_k^\phi(1), \dots, I_k^\phi(p))$ where for $m = 1, \dots, p$

$$I_{k+1}^\phi(m) = - \left(a_m^2 \sum_{t=1}^{k+1} \cos^2(\omega_m t + \phi_m) + \left(y_t - g_t - \sum_{n=1}^p a_n \sin(\omega_n t + \phi_n) \right) a_m \cdot \sin(\omega_m t + \phi_m) / \sigma_w^2 \right), \quad (5.5)$$

Evaluation of Score Vector: The data score is simply $S(\lambda_k, y_{k+1}) = \partial \mathcal{L}_{k+1|k+1}(\Lambda_k, \lambda) / \partial \lambda|_{\lambda=\lambda_k}$ where $\mathcal{L}_{k+1|k+1}(\Lambda_k, \lambda)$ is defined in (5.3).

Evaluating $S(\lambda, y_{k+1}) = \partial \mathcal{L}_{k+1|k+1}(\Lambda_k, \lambda) / \partial \lambda$ gives $S(\lambda, y_{k+1}) = (S_A(\lambda, y_{k+1}), S_Q(\lambda, y_{k+1}), S_\Theta(\lambda, y_{k+1}))$

for notational convenience we shall drop the time dependency of λ_k on the right-hand side of the reestimation equations, i.e., we shall write $\lambda_k = (A, q, \Theta)$. Then $\lambda_{k+1} = (A(k+1), q(k+1), \Theta(k+1))$ is calculated as A_{k+1} is calculated from (3.33):

$$q_i(k+1) = q_i + \frac{\gamma_{k+1|k+1, \Lambda_k}(i) (y_{k+1} - q_i - p_{k+1}(\Theta))}{\sum_{i=1}^{k+1} \gamma_{i|k+1, \Lambda_k}(i)}. \quad (5.9)$$

For periodic disturbances, Θ_{k+1} is reestimated as

$$a_n(k+1) = a_n + \frac{\left(y_{k+1} - \sum_{n=1}^p a_n \sin(\omega_n(k+1) + \phi_n) - g_{k+1} \right) \sin(\omega_n(k+1) + \phi_n)}{\sum_{t=1}^{k+1} \sin^2(\omega_n t + \phi_n)} \quad (5.10)$$

$$\phi_n(k+1) = \phi_n + \frac{\left(y_{k+1} - \sum_{n=1}^p a_n \sin(\omega_n(k+1) + \phi_n) - g_{k+1} \right) a_n \cos(\omega_n(k+1) + \phi_n)}{\sum_{t=1}^{k+1} a_n^2 \cos^2(\omega_n t + \phi_n) + \left(y_t - \sum_{n=1}^p a_n \sin(\omega_n t + \phi_n) + g_t \right) a_n \sin(\omega_n t + \phi_n)}. \quad (5.11)$$

For polynomial drift, Θ_{k+1} is reestimated as

$$a_n(k+1) = a_n + \frac{\left(y_{k+1} - \sum_{n=1}^p a_n (k+1)^n - g_{k+1} \right) (k+1)}{\sum_{t=1}^{k+1} a_n t^{2n}}. \quad (5.12)$$

where $S_A(\cdot)$ is computed in (3.29)

$$S_Q(\lambda, y_{k+1}) = (\gamma_{k+1|k+1, \Lambda_k}(1) (y_{k+1} - q_1 - p_{k+1}(\Theta)) / \sigma_w^2, \dots, \gamma_{k+1|k+1, \Lambda_k}(N) \cdot (y_{k+1} - q_N - p_{k+1}(\Theta)) / \sigma_w^2) \quad (5.6)$$

$S_\Theta(\lambda, y_{k+1}) = (S_a, S_\phi)$ where $S_a(\lambda, y_{k+1}) = (S_a(1), \dots, S_a(p))$,

$$S_a(m) = \left(y_{k+1} - g_{k+1} - \sum_{n=1}^p a_n \sin(\omega_n(k+1) + \phi_n) \right) \cdot \sin(\omega_n(k+1) + \phi_m) / \sigma_w^2, \quad m = 1, \dots, p \quad (5.7)$$

$S_\phi(\lambda, y_{k+1}) = (S_\phi(1), \dots, S_\phi(p))$ where for $m = 1, \dots, p$

$$S_\phi(m) = \left(y_{k+1} - g_{k+1} - \sum_{n=1}^p a_n \sin(\omega_n k + 1 + \phi_n) \right) a_m \cdot \cos(\omega_m k + 1 + \phi_m) / \sigma_w^2 \quad (5.8)$$

where $g_{k+1} = \sum_{i=1}^N \gamma_{k+1|k+1, \Lambda_k}(i) q_i$.

Summary of On-line Reestimation Equations: Let $\lambda_k = (A(k), q(k), \Theta(k))$ be the model estimate at time k . For

In the above equations, $g_t = \sum_{i=1}^N \gamma_{t|k+1, \Lambda_k}(i) q_i$.

(Recall that in the above equations, the right-hand side parameters are estimates at time k ; for notational convenience we have dropped this time dependency.)

Implementation Details:

1) Again in the actual implementation, $\gamma_{t|k+1, \Lambda_{t-1}}$ and $\xi_{t|k+1, \Lambda_{t-1}}$ are replaced by fixed or sawtooth-lag variables.

2) Applying forgetting leads to quicker convergence. Apart from the forgetting scheme outlined in Section IV for the Markov chain, we propose setting the denominators of (5.10) and (5.11) to a fraction of their value when these terms exceed a particular threshold. We illustrate the advantages of applying forgetting in the simulations.

3) Use $a_m(t-1)$, $a_n(t-1)$, $\phi_m(t-1)$, and $\phi_n(t-1)$ to approximate the computation of I_{k+1}^a , I_{k+1}^ϕ in (5.4) and (5.5).

Discussion:

Frequency Reestimation: On-line reestimation equations for updating the frequency components ω_n can also

be derived similarly. The equations are

$$\omega_n(k+1) = \omega_n + \frac{(k+1) \left(y_{k+1} - \sum_{n=1}^p a_n \sin(\omega_n(k+1) + \phi_n) - g_{k+1} \right) \cos(\omega_n(k+1) + \phi_n)}{\sum_{t=1}^{k+1} t^2 \left(a_n \cos(\omega_n t + \phi_n) + \left(y_t - \sum_{n=1}^p a_n \sin(\omega_n t + \phi_n) - g_t \right) \sin(\omega_n t + \phi_n) \right)}. \quad (5.13)$$

However, there is a large number of local maxima in $J(\lambda_k)$ [10], so there is no guarantee that (5.13) will converge to a global maxima.

We have developed off-line frequency reestimation techniques in [10]. We believe that it may be possible to combine the off-line scheme with (5.13) to track time-varying frequencies. Alternatively, it may be possible to run a bank of on-line processors and select at each time k the estimate which maximizes $J(\lambda_k)$. Further discussion of frequency reestimation is beyond the scope of this paper.

Independent Distributed (i.d.) Chain: We briefly consider the case when the imbedded chain is i.d. instead of Markov. Then the observations are characterized by the model $\lambda = (\pi, q, \Theta)$ where $\pi_i = P(s_k = q_i)$. The model estimate at time k is $\lambda_k = (\pi(k), q(k), \Theta(k))$ where $\pi_i(k)$ is the estimate of π_i at time k . The online reestimation formulas are

$$\pi_i(k+1) = \pi_i(k) + k^{-1}(\gamma_{k+1|k+1, \Lambda_k}(i) - \pi_i(k)) \quad (5.14)$$

$$q_i(k+1) = q_i(k) + \gamma_{k+1|k+1, \Lambda_k}(i) \cdot \frac{(y_{k+1} - q_i(k) - p_{k+1}(\Theta(k)))}{k\pi_i(k)}. \quad (5.15)$$

Here $\gamma_{k+1|k+1, \Lambda_k}$ is computed as

$$\begin{aligned} \gamma_{k+1|k+1, \Lambda_k}(i) &\triangleq f(s_{k+1} = q_i | y_{k+1}) \\ &= \frac{\pi_i(k) b_i(k, y_{k+1})}{\sum_{j=1}^N \pi_j(k) b_j(k, y_{k+1})}. \end{aligned} \quad (5.16)$$

The above equations are derived in [11], [15] using a similar approach to Section IV. The amplitude and phase reestimation equations are given by (5.10) and (5.11) with γ_k defined in (5.16). These can be derived using a similar proof to that outlined above for the Markov case. ■

VI. SIMULATION STUDIES

We illustrate the proposed schemes for the case where only white Gaussian noise is present and the case when also a periodic interference is present.

A. Markov Signals Imbedded in White Gaussian Noise

1) *Sawtooth-Lag Signal Extraction:* Here we examine the effect of sawtooth lag on the signal estimates. We assume that the transition probabilities and Markov levels are known.

A 200000 point two state Markov chain with $\pi_1 = \pi_2 = 0.5$, $a_{11} = a_{22} = 0.97$, and $q_1 = 0$, $q_2 = 1$ was generated.

To this was added zero-mean white Gaussian noise (WGN) sequences with standard deviations $\sigma_w = 0.5, 1.0, 1.5, 2.0$, respectively, to yield 4 sets of 200000 point observations.

With the initial transition probability estimates set at 0.97 and initial state level estimates set at 0 and 1, our on-line scheme was run on the 4 sets of observations. MAP signal estimates were obtained as in (2.3). Fig. 1 shows the number of errors in the MAP signal estimates per 10000 points plotted versus sawtooth-lag Δ_{\min} . Δ_{\max} was taken as $2\Delta_{\min}$.

Notice that for small noise variance σ_w , choosing $\Delta_{\min} = 4$ yields satisfactory results. For larger σ_w , it is necessary to increase the lag to obtain acceptable signal estimates. In all cases increasing σ_w increases the number of errors.

2) *Effect of Sawtooth-Lag on Convergence:* Using the same 4 observation sequences as above we illustrate the effect of lag on convergence of the on-line scheme.

Initial transition probability estimates $a_{ii}(0) = 0.8$ were chosen. In Fig. 2 we plot $a_{11}(k)$ versus k at 20000 point intervals for sawtooth lags with $\Delta_{\min} = 2$ and 20 ($\Delta_{\max} = 2\Delta_{\min}$).

Notice that for the low noise case ($\sigma_w = 0.5$) the convergence is independent of lag. With increasing noise larger lags result in faster convergence. As in the state estimation case above, simulations show that there is no appreciable improvement in convergence beyond a certain lag. For example, for $\sigma_w = 1$, choosing $\Delta_{\min} > 20$ does not result significant improvements. Also convergence slows with increasing σ_w .

Simulations not presented here show that the estimates of π_i improve with lag in a similar fashion to the improvement of the signal estimates with lag as described above. This is not surprising since both the MAP estimates as well as the estimates of π_i are obtained from $\gamma_{k|k+\Delta}$ and $\gamma_{1|1+\Delta}$, respectively.

3) *Comparison with Baum-Welch Reestimation:* We compare the performance of the proposed on-line scheme with the Baum-Welch (off-line) scheme.

A 100000 point two state Markov chain with $a_{11} = a_{22} = 0.97$, $\pi_1 = \pi_2 = 0.5$, and $q_1 = 0$, $q_2 = 1$ was generated. To this was added zero-mean WGN with $\sigma_w = 2.0$ yielding the observations.

Initial estimates were taken as $q_1(0) = 0.1$, $q_2(0) = 0.6$, $a_{ii}(0) = 0.9$. Also the sawtooth lag was chosen as $\Delta_{\min} = 20$, $\Delta_{\max} = 40$. Fig. 3 shows the performance of the on-line algorithm on the observations. Notice that $q_i(k)$ are close to the true values q_i from $k = 30000$ onwards.

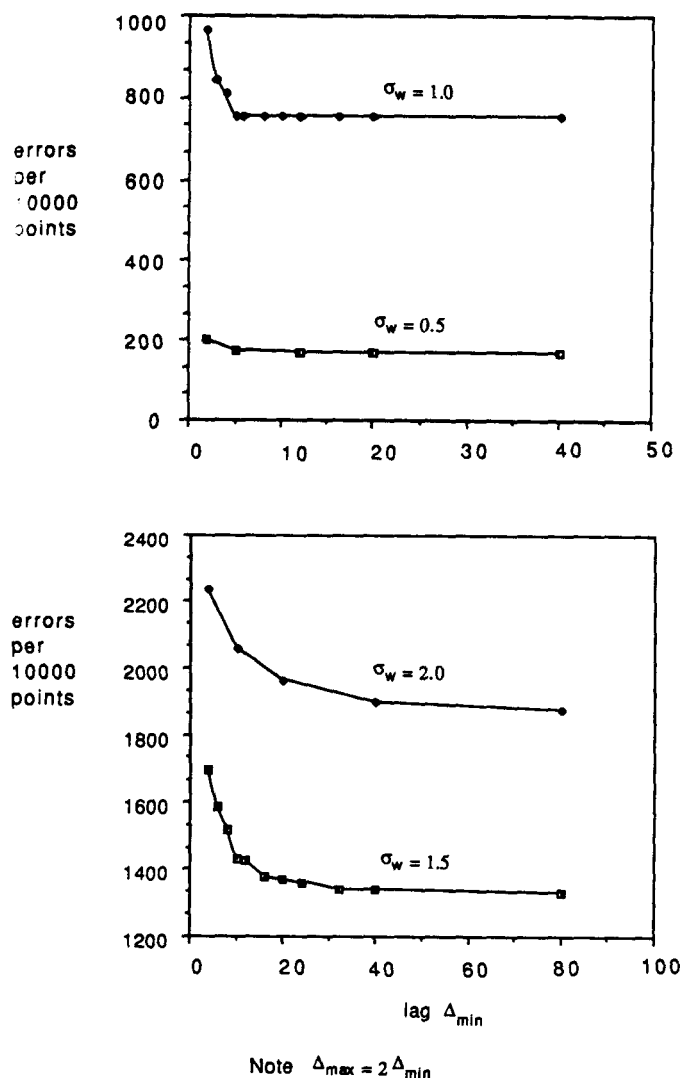


Fig. 1. Sawtooth-lag smoother performance. Note that $\Delta_{max} = 2\Delta_{min}$.

The estimates $a_{11}(k)$, $a_{22}(k)$ also go towards the true values.

Fig. 4 shows the performance of the Baum-Welch (off-line) reestimation scheme on the same observation sequence. The same initial estimates were used as in the on-line case. Notice that the on-line estimates $q_i(k)$ for large k are close to the off-line estimates \bar{q}_i in the 10th pass. Also the on-line estimates $a_{ii}(k)$ are close to that of the off-line estimates after 7 passes. These illustrate the significant improvement in convergence of the on-line scheme.

4) Illustration of Forgetting: We show how using exponential forgetting can result in much faster convergence. Also only filtered variables ($\Delta = 0$) are used in this example.

A three state Markov chain with $a_{ii} = 0.9$, $a_{ij} = 0.05$, $\pi_i = 0.33$, $[q(1), q(2), q(3)] = [0, 1, 2]$ was generated. Then WGN with $\sigma_w = 1$ was added yielding the observations. Initial parameter estimates were $a_{ii}(0) = 0.1$, $q_i(0) = 0.5, 0.6, 0.7$.

Fig. 5(a) shows the evolution of $q_i(k)$ versus time without forgetting ($\rho = 1$) and with exponential forgetting (ρ

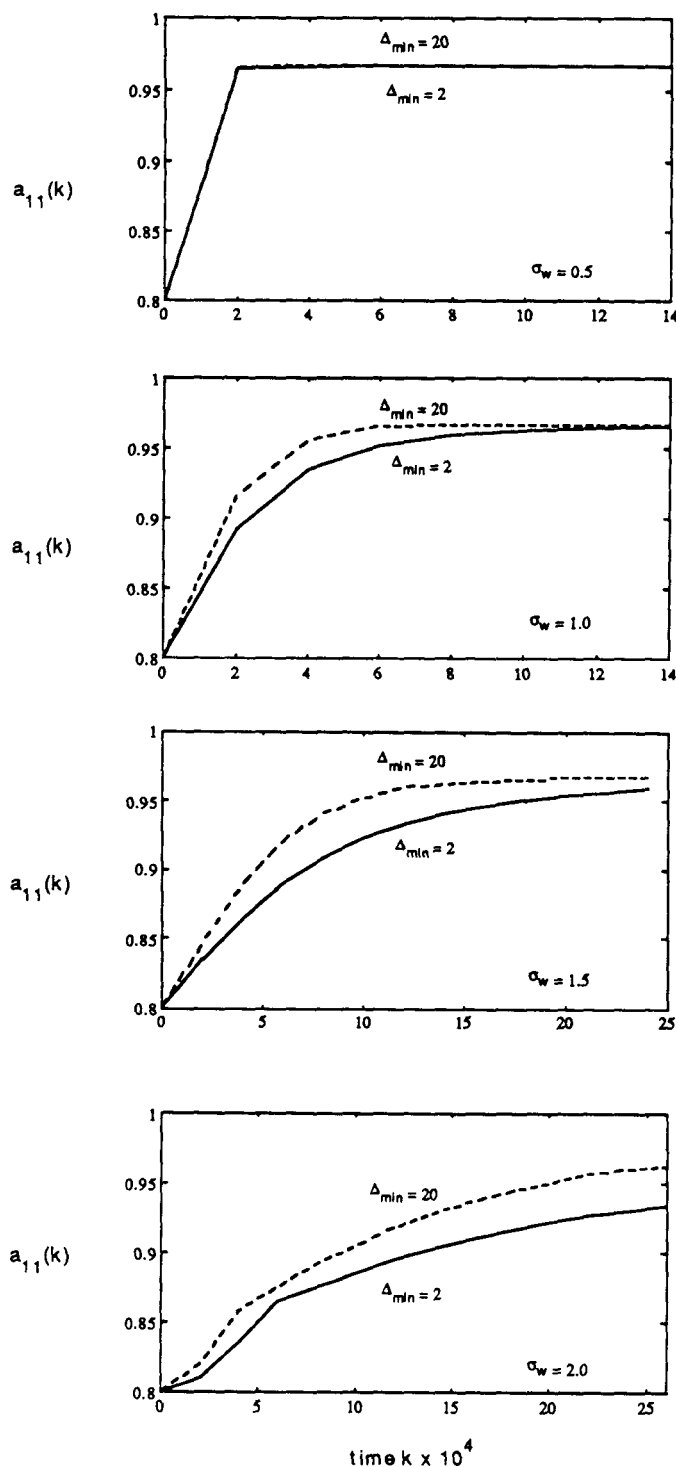


Fig. 2. Effect of sawtooth lag and noise variance on convergence.

$= 0.9979$). Fig. 5(b) shows the evolution of $a_{ii}(k)$. The estimates were plotted at 1000 point intervals.

Notice that the level estimates with forgetting are very close to the true values in less than 10000 points and are comparable with the estimates at $k = 50000$ without forgetting. Similarly, the transition probabilities are very close to the true values after 250000 points with forgetting. The estimates without forgetting after 400000 points are still not close to the true values. (Actually they ap-

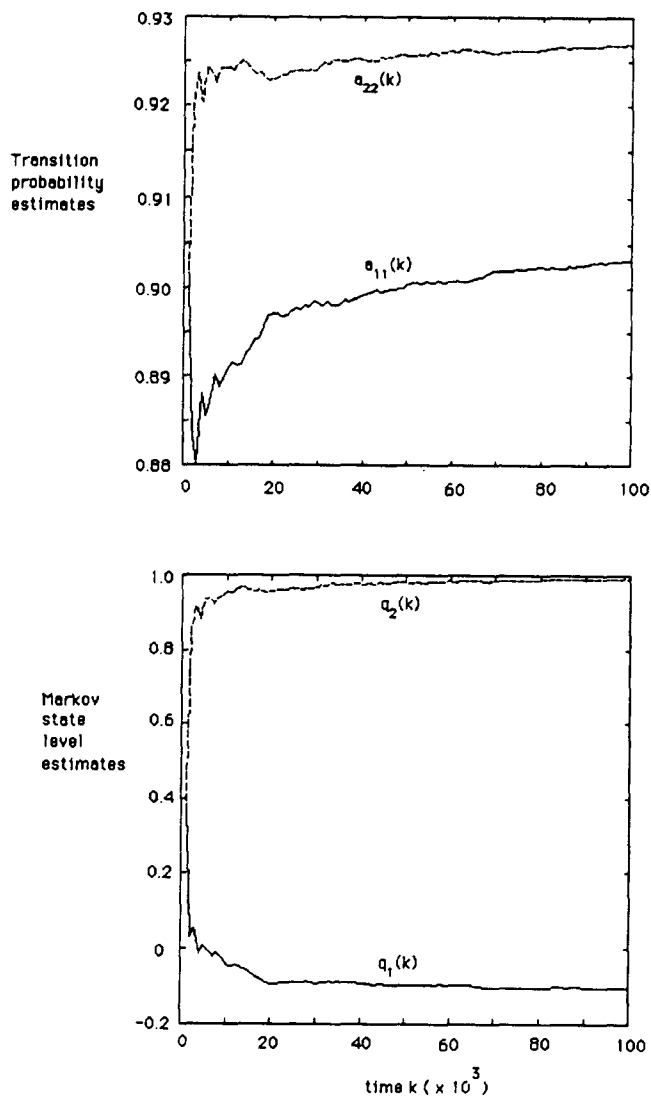


Fig. 3. On-line reestimation scheme performance. True model parameters: $a_{ii} = 0.97$, $\pi_i = 0.5$, $q_1 = 1$, $q_2 = 1$, $\sigma_w = 2.0$.

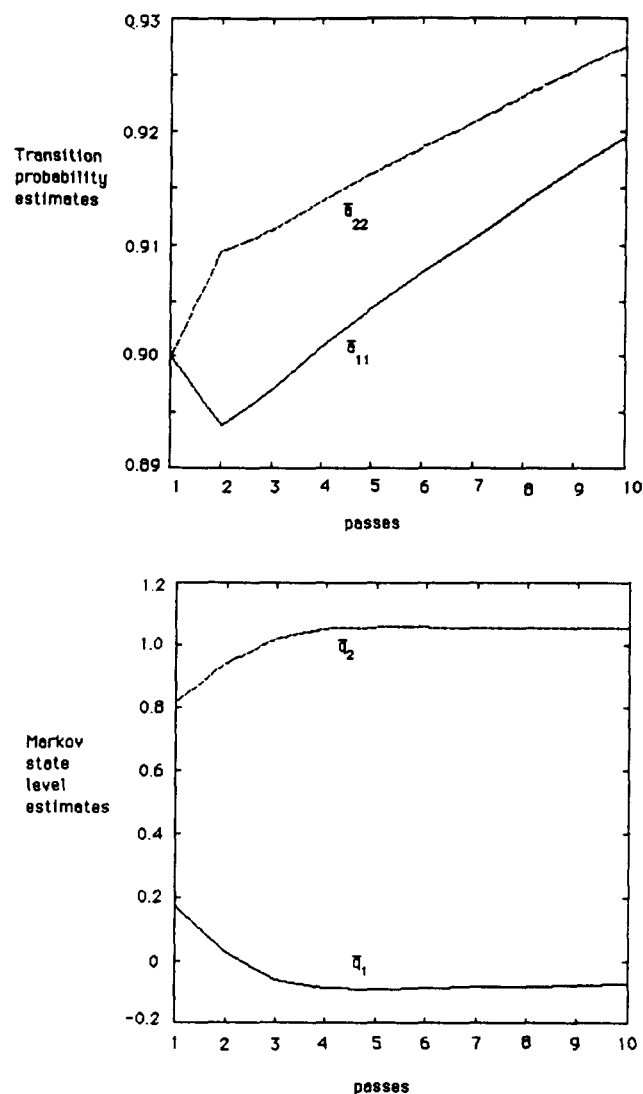


Fig. 4. Performance of Baum-Welch reestimation (same data as used in Fig. 3).

proach the true value after 3 million points.) This shows that forgetting can significantly speed up convergence.

5) *Effect of Initial Estimates:* Extensive simulations have been carried out to ascertain the effect of initial estimates on the performance of the proposed on-line scheme. We have tested the on-line scheme over the following range of models, assuming $q_i = i$:

$N = 2, 3, 4$, $a_{ii} = 0.5$ to 0.99 , σ_w up to 2 and over the following range of initial conditions:

$a_{ii}(0) = 0.1$ to 0.99

$q_i(0)$ up to $3\sigma_w$ away from true values q_i .

In all cases, over this range of models and initial conditions, the on-line scheme yielded estimates that converged to the true model. Of course, when σ_w is large convergence is slow and so longer data sequences are required to check for convergence. The initial conditions (particularly level estimates) do, however, appear to have some effect on convergence rates as described below.

Fig. 6 shows the effect of 5 different initial estimates on the on-line scheme for the same data as in the previous simulation example. To speed up convergence, the same

forgetting scheme as in the previous simulation example was used. The initial estimates are:

Fig 6(a): $a_{ii}(0) = 0.9$, $q(0) = (0.5, 1, 1.5, 2)$.

Fig 6(b): $a_{ii}(0) = 0.9$, $q(0) = (0, 0.5, 1, 1.5)$.

Fig 6(c): $a_{ii}(0) = 0.9$, $q(0) = (-2, -1.5, -1, -0.5)$.

Fig 6(d): $a_{ii}(0) = 0.9$, $q(0) = (0, 0.05, 0.1, 0.15)$.

Figs 6(e) and (f): $a_{ii}(0) = 0.1$, $q(0) = (-2, -1.5, -1, -0.5)$.

Figs. 6(a) to (d) show the state level estimates $q(k)$ for the above initial conditions. The estimates are plotted every 5×10^4 points and the time scale is in units of 5×10^4 . Notice that when the initial estimates are chosen such that more than one initial value is close to a particular state then the convergence can slow down since both the estimates may track the same state level for some time (see Fig. 6(a) and especially Fig. 6(b)). Nevertheless, the level estimates converge to the true values. Similarly, the transition probability estimates converge to the true values (not shown in the figures).

Fig. 6(e) shows the transition probability estimates

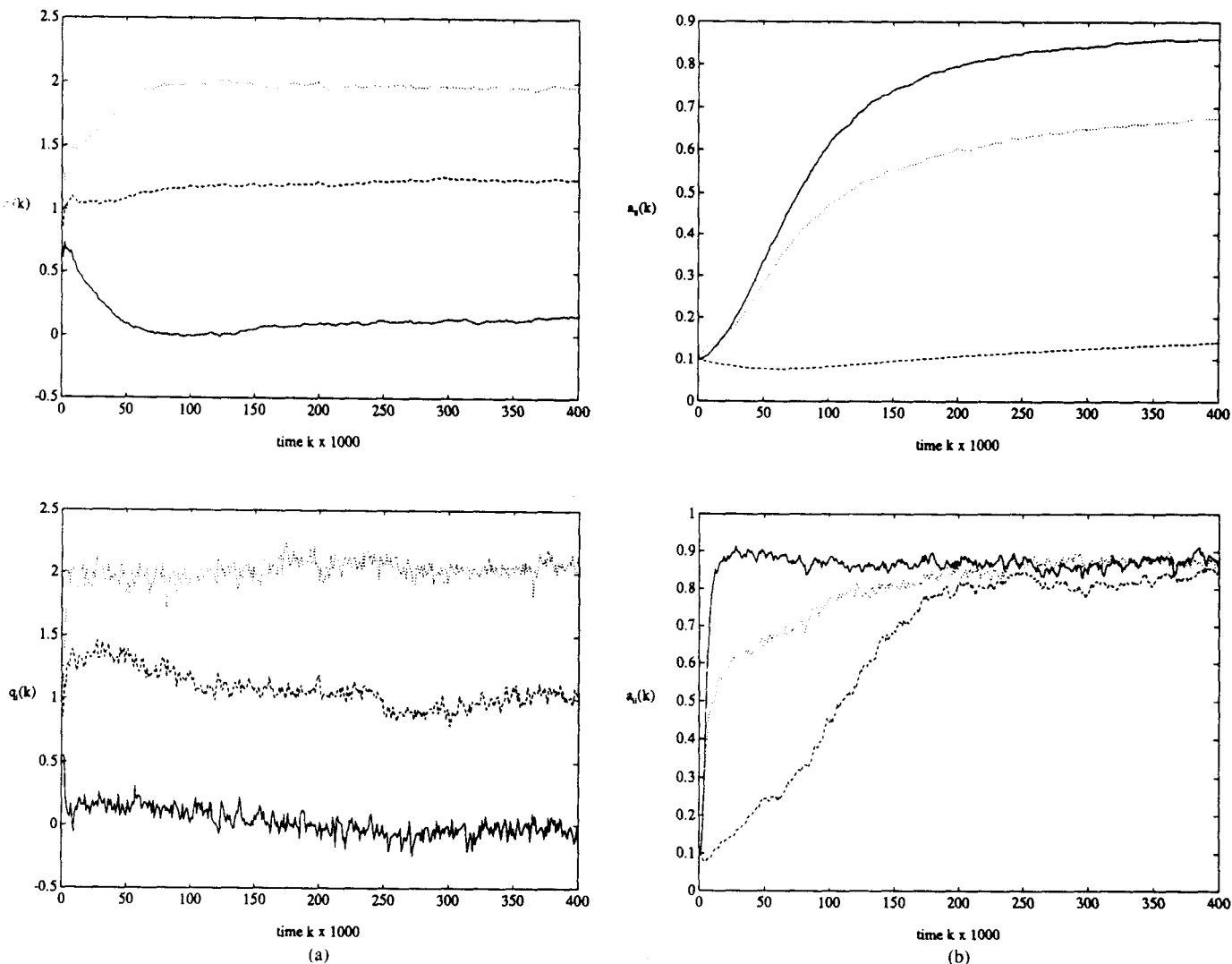


Fig. 5. (a) On-line estimation of Markov state levels. Top: Without exponential forgetting ($\rho = 1$). Bottom: With exponential forgetting ($\rho = 0.9979$), initial estimates: $q_1(0) = 0.5$, $q_2(0) = 0.6$, $q_3(0) = 0.7$. (b) On-line estimation of transition probabilities. Top: Without exponential forgetting ($\rho = 1$). Bottom: With exponential forgetting ($\rho = 0.9979$), initial estimates: $a_{ii}(0) = 0.1$, $a_{ij}(0) = 0.45$.

when the initial estimates are chosen at some distance from the true value $a_{ii} = 0.1$. Again the probabilities converge to the true values. Also the level estimates converge (Fig. 6(f)).

B. Markov Signal Imbedded in White Noise and Periodic Interference

We here illustrate the performance of the proposed schemes when the Markov chain is imbedded in periodic interference in addition to WGN.

1) *Illustration of Proposed Scheme:* The purpose of this example is to show that the proposed algorithm satisfactorily learns the Markov state levels, transition probabilities, and also amplitude and phase components of the periodic signal (with known frequency components) from the sum of the Markov signal, periodic signal, and WGN of known variance.

A 40000-point 2 state Markov chain was generated with

$a_{ii} = 0.97$ and $q_i = 0$, $q_2 = 1$. To this chain was added zero-mean WGN with $\sigma_w = 0.5$. Also a periodic interference $a_1 \sin(\omega_1 k + \phi_1) + a_2 \sin(\omega_2 k + \phi_2)$ was added where $\Theta = (a_1, a_2, \phi_1, \phi_2) = (0.8, 0.5, \pi/3, 0)$. Initial Markov estimates were taken as $q_1(0) = -0.1$, $q_2(0) = 0.6$, and $a_{ii}(0) = 0.90$. Also the initial estimate $\Theta(0) = (0.5, 0.5, 0.3, 0.3)$.

Fig. 7 shows the amplitude and phase estimates plotted versus time. Notice that after 50 000 points the amplitudes and phases are very close to the true values. At $k = 50000$, the state level estimates were $q_1(k) = 0.021$, $q_2(k) = 1.003$ and the transition probability estimates were $a_{11}(k) = 0.963$, $a_{22}(k) = 0.967$ which are also close to the true values. Thus the proposed schemes satisfactorily cope with the presence of periodic interferences.

2) *Tracking Sinusoid with Jump Change in Phase and Amplitude:* Here we consider a Markov chain imbedded in sinusoidal interference with jump changes in amplitude and phase in addition to white noise. We illustrate the

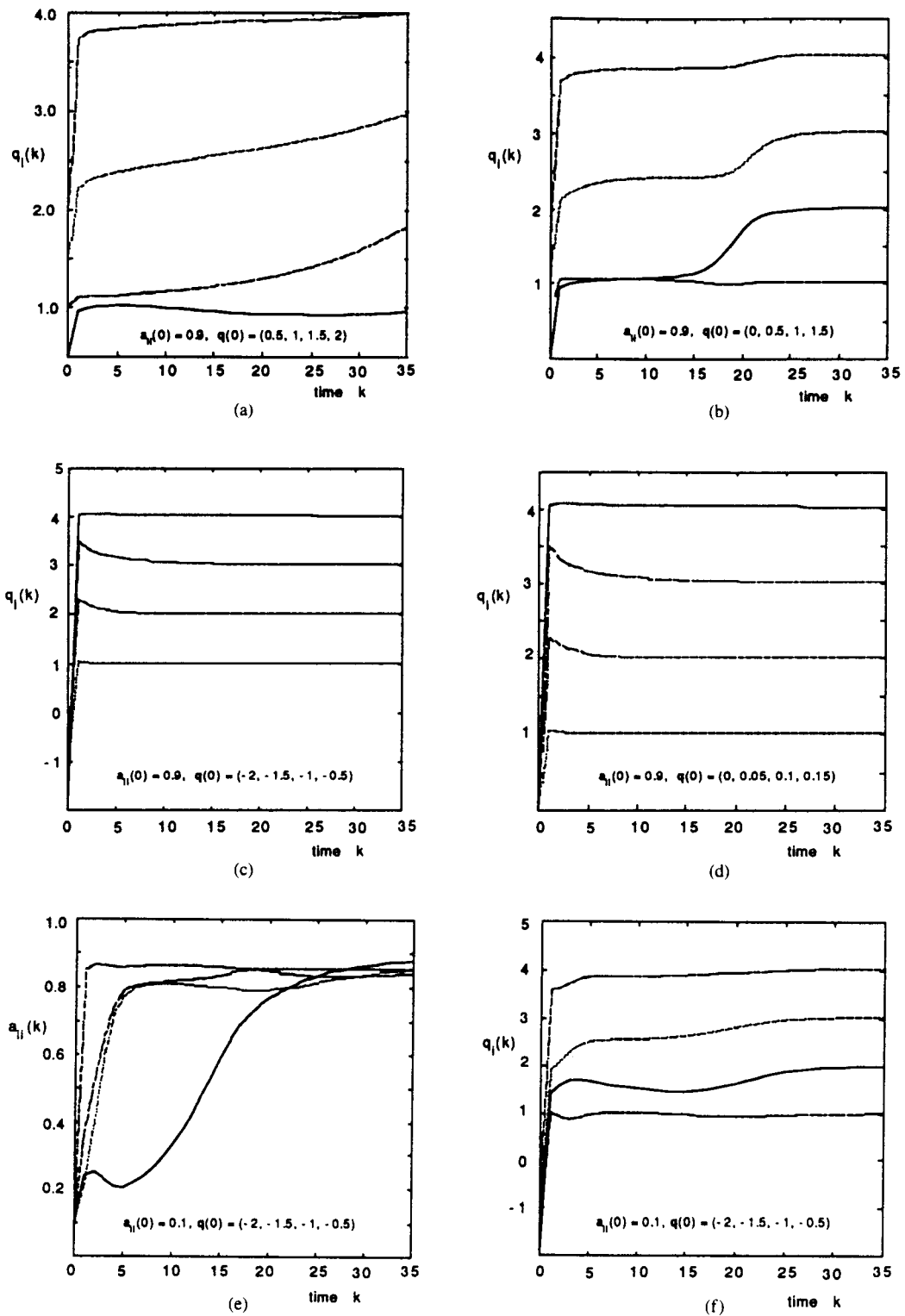


Fig. 6. Effect of initial estimates.

performance of the proposed on-line schemes tracking the amplitudes and phases.

To the Markov chain of the previous example was added sinusoidal interference $a \sin(\omega k + \phi)$ where $a = 0.8$ and $\phi = \pi/6$ for $1 \leq k \leq 20000$ and $a = 0.5$ and $\phi = \pi/3$ for $20001 \leq k \leq 40000$.

Then WGN with $\sigma_w = 0.5$ was added to yield the observations.

We ran our standard on-line scheme (without forgetting) and the on-line scheme with forgetting on the data. In the scheme with forgetting, when the denominator of (5.10) exceeded 3000, the denominators of (5.10) and (5.11) were reset to a tenth of their value. Initial estimates were taken as $a_{ii}(0) = 0.8$, $q_1(0) = 0$, $q_2(0) = 0.6$. The initial amplitude and phase were chosen as 0.5 and 0.5 rad, respectively.

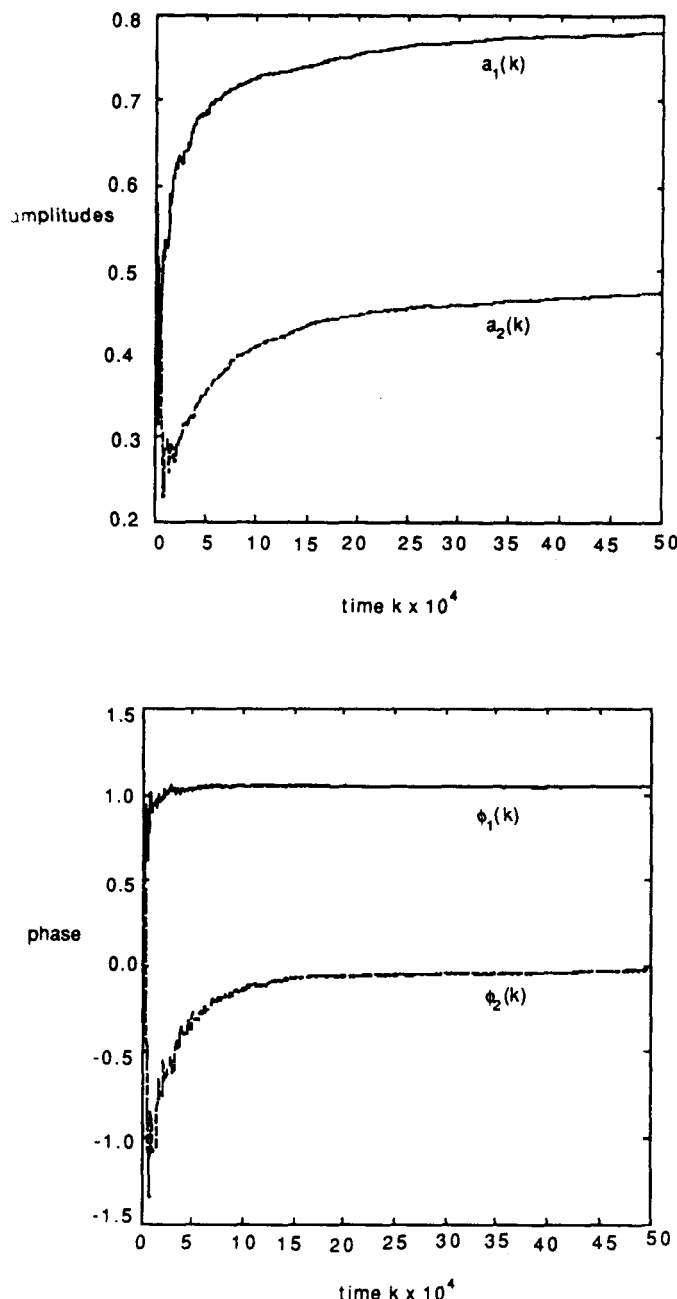


Fig. 7. On-line estimation of phases and amplitudes of periodic interference. True parameters $\Theta = (a_1, a_2, \Phi_1, \Phi_2) = (0.8, 0.5, \pi/3, 0)$, $a_{ii} = 0.97$, $q = (0, 1)$, $\sigma_u = 0.5$.

Fig. 8 shows how the amplitude and phase are tracked versus time. Notice that our standard scheme learns the amplitude and phase for $k < 20000$. Then it slowly starts moving towards the new amplitude and phase. In contrast with forgetting the convergence is fast to the new amplitude and phase. Of course with forgetting the estimates drift a little due to the effect of local statistics. For the standard on-line scheme, the state level estimates at $k = 50000$ were $q_1(k) = -0.005$, $q_2(k) = 1.004$ and the transition probabilities were $a_{11}(k) = 0.961$, $a_{22}(k) = 0.964$. For the on-line scheme with forgetting at $k = 50000$, $q_1(k) = -0.007$, $q_2(k) = 1.007$, $a_{11}(k) = 0.965$, $a_{22}(k) = 0.968$.

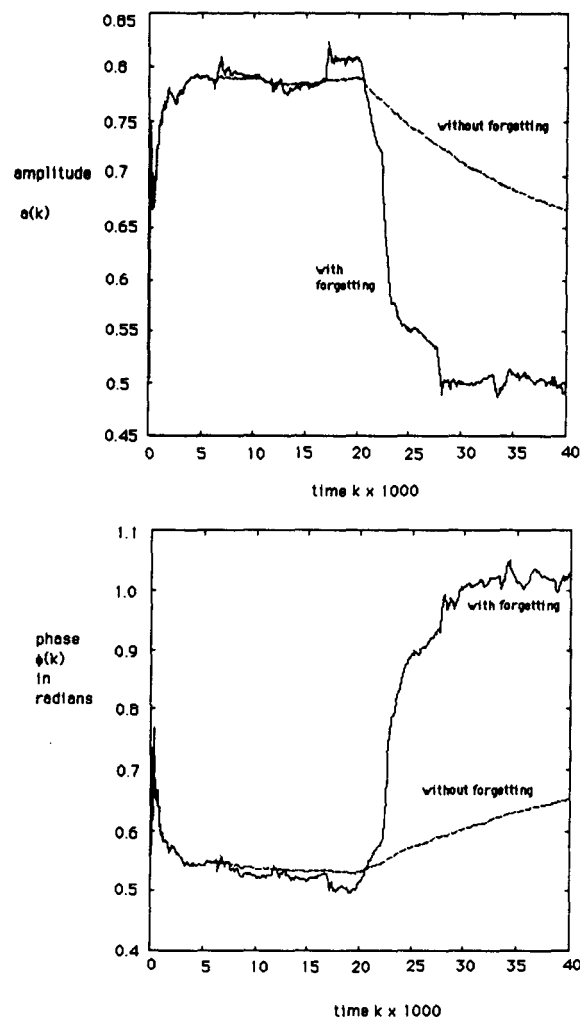


Fig. 8. On-line estimation of periodic interference with jump changes in phase and amplitude.

VII. CONCLUSIONS

In this paper we have derived sequential algorithms for HMM estimation based on maximizing the Kullback-Leibler information measure. The algorithms proposed use robust, memory efficient, fixed-lag, and sawtooth-lag smoothing schemes. Simulations confirm that significant improvements in convergence occur when these on-line schemes are used.

As an application of the proposed schemes we have obtained on-line estimates of the Markov signal model and periodic signal from a mixture of the Markov process, periodic signal, and additive white Gaussian noise. With the on-line reestimation it is possible to deal with sinusoidal disturbances with slowly varying amplitudes, and phases.

APPENDIX

A. Proof of Lemma 3.1

Proof: Consider the RHS of (3.12), namely,

$$\sum_{i=1}^N \alpha_{k-1|\Lambda_{k-2}}(i) a_{ij}(k-1) b_j(k-1, y_k).$$

Now

$$\begin{aligned}\alpha_{k-1|\Lambda_{k-2}}(i) &\triangleq f(Y_{k-1}, s_{k-1} = q_i | \Lambda_{k-2}) \\ &= f(Y_{k-1}, s_{k-1} = q_i | \Lambda_{k-2}, \lambda_{k-1})\end{aligned}$$

because $\alpha_{k-1|\Lambda_{k-2}}(i)$ is independent of the value of λ_{k-1} .

Similarly

$$\begin{aligned}a_{ij}(k-1) &\triangleq P(s_k = q_j | s_{k-1} = q_i, \lambda_{k-1}) \\ &= P(s_k = q_j | s_{k-1} = q_i, \lambda_{k-1})\end{aligned}$$

and

$$\begin{aligned}b_j(k-1, y_k) &\triangleq f(y_k | s_k = q_j(k-1), \lambda_{k-1}) \\ &= f(y_k | s_{k-1} = q_j(k-1), \Lambda_{k-1}).\end{aligned}$$

We can now follow the same proof for the recursion of the standard forward variable in [5]:

$$\begin{aligned}P(s_k = q_j | s_{k-1} = q_i, \Lambda_{k-1}) \\ &= P(s_k = q_j | s_{k-1} = q_i, Y_{k-1}, \Lambda_{k-1}) \\ f(y_k | s_k = q_j(k-1), \Lambda_{k-1}) \\ &= f(y_k | s_k = q_j(k-1), s_{k-1} = q_i, Y_{k-1}, \Lambda_{k-1})\end{aligned}\quad (7.2)$$

because w_k is white. Substituting these into the RHS immediately yields $\alpha_{k|\Lambda_{k-1}}(j)$.

The proof for backward recursion (3.13), and for (3.14) and (3.15) are identical to that in [5]. \square

B. Proof of Lemma 3.2

Proof: Notice that

$$\begin{aligned}\log f(Y_{k+1}, S_{k+1} | \lambda) \\ &= \log f(Y_{k+1} | S_{k+1}, \lambda) + \log f(S_{k+1} | \lambda) \\ &= \sum_{i=1}^N \sum_{t=1}^{k+1} \delta(s_t - q_i) \log b_i(y_t) \\ &\quad + \sum_{i=1}^N \sum_{j=1}^N n_{ij}(k+1) \log a_{ij}\end{aligned}$$

where $n_{ij}(k+1)$ is the number of jumps from state i to state j till time $k+1$ and $\delta(s_t - q_i) = 1$ if $s_t = q_i$ and zero otherwise. Also

$$\begin{aligned}E\{\delta(s_t - q_i) | Y_{k+1}, \Lambda_k\} &= f(s_t = q_i | Y_{k+1}, \Lambda_{t-1}) \\ &= \gamma_{t|k+1, \Lambda_{t-1}}(i)\end{aligned}\quad (7.3)$$

$$\begin{aligned}E\{n_{ij}(k+1) | Y_{k+1}, \Lambda_k\} \\ &= \sum_{t=1}^{k+1} f(s_t = q_i, s_{t+1} = q_j | Y_{k+1}, \Lambda_{t-1}) \\ &= \sum_{t=1}^{k+1} \zeta_{t|k+1, \Lambda_{t-1}}(i, j)\end{aligned}\quad (7.4)$$

where the last equalities in (7.3) and (7.4) follow from the definitions (3.11). Substituting these into the definition of $\mathcal{Q}_{k+1}(\Lambda_k, \lambda)$ in (3.9) proves the lemma. \square

ACKNOWLEDGMENT

The authors thank P. L. Slingsby for correcting an error in the evaluation of $I_{k+1}^A(\lambda)$ of Section III-C, in a preliminary draft of the paper.

REFERENCES

- [1] D. Colquhoun and A. G. Hawkes, "On stochastic properties of single ion channels," *Proc. Roy. Soc. Ser. B*, no. 211, pp. 205-235, 1981.
- [2] S. H. Chung, V. Krishnamurthy, and J. B. Moore, "Adaptive processing techniques based on hidden Markov models for characterizing very small channel currents buried in noise and deterministic interferences," *Phil. Trans. Roy. Soc.*, London, vol. 334, pp. 357-384, Dec. 1991.
- [3] V. Krishnamurthy and L. B. White, "Blind equalization of FIR channels with Markov inputs," in *Proc. IFAC ACASP'92*, Grenoble, France, July 1992, pp. 633-638.
- [4] P. H. Barry and J. M. Diamond, "Junction potentials, electrode standard potentials, and other problems in interpreting electrical properties in membranes," *J. Membrane Biology*, vol. 3, pp. 93-122, 1970.
- [5] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in such speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257-285, 1989.
- [6] L. E. Baum and T. Petrie, "Statistical interference for probabilistic functions of finite state Markov chains," *Ann. Math. Stat.*, vol. 37, pp. 1554-1563, 1966.
- [7] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Stat.*, vol. 41, no. 1, pp. 164-171, 1970.
- [8] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, vol. 3, pp. 1-8, 1972.
- [9] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, ser. 39, vol. 6, pp. 1-38, 1977.
- [10] V. Krishnamurthy, J. B. Moore, and S. H. Chung, "Hidden Markov model signal processing for eliminating unknown deterministic interferences," *IEEE Trans. Automat. Contr.*, vol. 38, no. 1, pp. 146-152, Jan. 1993.
- [11] D. M. Titterton, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*. New York: Wiley Series in Probability and Mathematical Statistics, 1985.
- [12] J. B. Moore, "Discrete-time fixed-lag smoothing algorithms," *Automatica*, vol. 9, no. 2, pp. 163-174, 1973.
- [13] D. Clements and B. D. O. Anderson, "A nonlinear fixed-lag smoother for finite-state Markov processes," *IEEE Trans. Inform. Theory*, vol. IT-21, no. 4, pp. 446-452, July 1975.
- [14] W. M. Wonham, "Some applications of stochastic differential equations to optimal nonlinear filtering," *SIAM J. Contr.*, vol. 2, pp. 347-369, 1965.
- [15] D. M. Titterton, "Recursive parameter estimation using incomplete data," *J. Roy. Stat. Soc. B*, vol. 46, no. 2, pp. 257-267, 1984.
- [16] J. Behboodian, "Information matrix for a mixture of two normal distributions," *J. Stat. Comput. Simul.*, vol. 1, pp. 295-314, 1972.
- [17] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Syst. Theory J.*, no. 6, pp. 1035-1074, 1983.
- [18] M. V. Eyuboglu and S. V. H. Qureshi, "Reduced state sequence estimations with set partitioning and decision feedback," *IEEE Trans. Commun.*, no. 36, pp. 12-20, 1988.
- [19] E. Weinstein, M. Feder, and A. V. Oppenheim, "Sequential algorithms for parameter estimation based on the Kullback-Leibler information measure," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 9, Sept. 1990.

- [20] V. Krishnamurthy, J. B. Moore, and S. H. Chung, "Hidden fractal model signal processing," *Signal Processing*, vol. 24, pp. 177-192, 1991.
- [21] B. T. Polyak, "New method of stochastic approximation type," *Automat. Remote Contr.*, vol. 51, pp. 937-946, 1990.
- [22] T. Soderstrom and P. Stoica, *System Identification*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [23] I. B. Collings, V. Krishnamurthy, and J. B. Moore, "Recursive prediction error techniques for adaptive estimation of hidden Markov models," in preparation.



Vikram Krishnamurthy (S'90-M'91) was born in India in 1966. He received the B.E. degree in electrical engineering with first class honours from the University of Auckland, New Zealand, in 1987, and the Ph.D. degree from the Department of Systems Engineering, Australian National University, Canberra, in 1991.

From September to December 1991 he was a postdoctoral fellow at the Department of System Theory, Technical University of Vienna, Austria. Since January 1992, he has been a Research En-

gineer at the Center for Robust and Adaptive Systems, Department of Systems Engineering, Australian National University. His current interests are in nonlinear estimation, including stochastic filtering theory, and blind-equalization algorithms for communication channels.



John B. Moore (S'66-M'67-SM'77-F'79) was born in China in 1941. He received the bachelor's and master's degrees in electrical engineering in 1963 and 1964, respectively, and the doctoral degree in electrical engineering from the University of Santa Clara, California, in 1967.

He was appointed Senior Lecturer at the University of Newcastle in 1967, promoted to Associate Professor in 1968, and full Professor (personal chair) in 1973. In 1982, he was appointed as a Professorial Fellow in the Department of Sys-

tems Engineering, Research School of Physical Sciences, Australian National University, and promoted to Professor in 1990. He has been head of the department since 1992 and a team leader in the Cooperative Research Center for Robust and Adaptive Systems in the department. He has held visiting academic appointments at the University of Santa Clara (1968); the University of Maryland (1970); Colorado State University and Imperial College (1974); the University of California, Davis (1977); the University of Washington, Seattle (1981); Cambridge University and the National University of Singapore (1985); University of California, Berkeley (1987, 1989, and 1991); University of Alberta, Edmonton, Canada (1992). He has spent periods in industry as a design engineer and as a consultant, and currently has research grants from industry and government laboratories. His current research is in control and communication systems. He is coauthor with Brian Anderson of three books: *Linear Optimal Control* (Prentice-Hall, 1971), *Optimal Filtering* (Prentice-Hall, 1979), and *Optimal Control—Linear Quadratic Methods* (Prentice-Hall, 1989).

Dr. Moore is a Fellow of the Australian Academy of Technological Sciences.