Hidden Gauss–Markov Models for Signal Classification

Phillip L. Ainsleigh, *Member, IEEE,* Nasser Kehtarnavaz, *Senior Member, IEEE,* and Roy L. Streit, *Senior Member, IEEE*

***Abstract*—Continuous-state hidden Markov models (CS-HMMs) are developed as a tool for signal classification. Analogs of the Baum, Viterbi, and Baum–Welch algorithms are formulated for this class of models. The CS-HMM algorithms are then specialized to hidden Gauss–Markov models (HGMMs) with linear Gaussian state-transition and output densities. A new Gaussian refactorization lemma is used to show that the Baum and Viterbi algorithms for HGMMs are implemented by two different formulations of the fixed-interval Kalman smoother. The measurement likelihoods obtained from the forward pass of the HGMM Baum algorithm and from the Kalman-filter innovation sequence are shown to be equal. A direct link between the Baum–Welch training algorithm and an existing expectation-maximization (EM) algorithm for Gaussian models is demonstrated. A new expression for the cross covariance between time-adjacent states in HGMMs is derived from the off-diagonal block of the conditional joint covariance matrix. A parameter invariance structure is noted for the HGMM likelihood function. CS-HMMs and HGMMs are extended to incorporate mixture densities for the *a priori* density of the initial state. Application of HGMMs to signal classification is demonstrated with a three-class test simulation.**

***Index Terms*—Baum–Welch algorithm, continuous-state HMM, EM algorithm, fixed-interval smoother, forward–backward algorithm, hidden Markov model, Kalman filter, maximum likelihood classification, mixture density.**

I. Introduction

SIGNAL classification algorithms typically comprise a feature extractor and a decision function. The feature extractor compresses the observed signal into a smaller set of variables containing the essential information. The decision function makes class assignments based on the features. This paper presents a decision function that is based on a time-varying probability density function (PDF). Given the feature set and labeled training data from each class, a working classifier is obtained by using the training data to estimate the PDF of the features for each class hypothesis: a process known as *model training*. The PDF models used here are parametric; therefore, training is implemented using the expectation-maximization (EM) algorithm to optimize the model parameters. An unknown signal is assigned to a class by calculating its features, using the optimized PDF models to evaluate the conditional likelihood for each class, and then selecting the maximum.

When features exhibit a temporal dependence, there is a strong relationship between classification and *tracking* since much of the information regarding a signal can be inferred from the trajectory that its features form as a function of time. In spite of this, the classification and tracking fields have historically evolved independently, giving rise to two separate families of tools. From the tracking side comes the Kalman filter and a number of related smoothing algorithms. From the classification side comes the hidden Markov model (HMM) and its associated algorithms. This paper unifies these two families by developing a general theory of continuous-state HMMs (CS-HMMs) and then specializing the CS-HMM algorithms to models with linear Gaussian densities. These developments provide a PDF model for classes of signals with continuously varying features, in contrast to traditional HMMs that model features discontinuously in terms of a finite chain of stationary states.

The idea of an equivalence between Kalman filters and HMMs will come as no surprise to many readers since, for example, both models are represented using the same Bayesian inference network [1]. The specifics of the equivalence may be more surprising, however. The Kalman filter is much more than an analogy or sibling to the HMM. The Kalman-filter model *is* an HMM with linear Gaussian model densities. The Baum algorithm for this HMM *is* a Kalman smoother, as is the Viterbi algorithm. The likelihood defined by the HMM criteria is analytically the same as the likelihood defined for a Kalman filter. Finally, the EM algorithm for Kalman-filter models is obtained directly from the auxiliary function used to generate the Baum–Welch re-estimation algorithm for HMMs.

While the focus of this work is classification, the development of CS-HMMs and the equivalence of HGMMs with Kalman-filter models has ramifications for tracking as well. A common criticism of Kalman filters and smoothers is the disproportionate amount of credence that they give to the *a priori* state model when processing data. A symptom of this "narcissistic-model syndrome" is that the error covariances depend only on the *a priori* values of the model parameters and are independent of the observed data. This problem might be eliminated by using the EM algorithm to adapt the model covariance matrices during tracking.

A. Related Literature

Discrete-state HMMs (DS-HMMs) were developed by Baum and his colleagues in the late 1960s and early 1970s [2]–[5], in-

Manuscript received September 28, 2000; revised February 8, 2002. This work was supported by the Office of Naval Research. The associate editor coordinating the review of this paper and approving it for publication was Dr. Ali H. Sayed.

P. Ainsleigh and R. L. Streit are with the Naval Undersea Warfare Center, Newport, RI 02841 USA (e-mail: ainsleighpl@npt.nuwc.navy.mil).

N. Kehtarnavaz is with the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080 USA.

Publisher Item Identifier S 1053-587X(02)04379-9.

cluding a method for estimating the model parameters (referred to as the *Baum–Welch algorithm*) and a method for estimating the sequence of individually most likely hidden states (referred to as the *Baum* or *forward–backward algorithm*). An alternative approach to state estimation is to seek the single most likely sequence of states, which is obtained using the Viterbi algorithm [6], [7]. Dempster *et al.* [8] observed that the Baum–Welch algorithm for estimating the HMM parameters is an example of the EM algorithm. Heiser has since noted that the EM algorithm is a special case of iterative majorization [9].

The primary applications of HMMs as a classification tool have been in speech recognition. In 1980, Ferguson [10] helped to solidify HMM theory for speech recognition by succinctly identifying the three fundamental problems, namely, state estimation, likelihood evaluation, and parameter estimation. This "HMM paradigm" was developed further by Levinson *et al.* [11].

Most extensions of the basic HMM structure have focused on obtaining more general output densities. Liporace [12] treated HMMs with elliptically symmetric continuous output densities. Juang *et al.* [13] relaxed the elliptical symmetry requirement by treating models with Gaussian-mixture output densities. Other continuous-output HMMs include those whose measurements are governed by an autoregressive process [14]–[16], a polynomial regression function [17], and a linear Gaussian model [18]. When employed with models having variable-duration states [19], such continuous-output distributions lead to the versatile class of segmental models [20].

In contrast to these continuous-*output* models, the general class of CS-HMMs has received little attention in the signal processing literature. Kalman-filter models, on the other hand, have been extensively studied in the contexts of tracking [21], control [22], and optimal filtering [23], [24], although it is not generally known that these models are examples of CS-HMMs. Most early applications of Kalman filters considered the model parameters (i.e., the state-transition, output, and covariance matrices) to be known from physical considerations; therefore, parameter estimation was not considered. Two notable exceptions are Kashyap [25] and Gupta and Mehra [26], who used gradient-based nonlinear optimization techniques to maximize the likelihood as expressed in terms of the measurement innovations. Parameter estimation in Gaussian models is more prominent in the time series and econometrics literature, where the first applications of the EM algorithm for this purpose appeared in the early 1980s. In particular, Shumway and Stoffer [27] and Watson and Engle [28] independently developed EM algorithms for estimating parameters in time-invariant Gaussian models with linear constraints. Building on this work, signal processing researchers began using the EM algorithm with linear Gaussian models in the early 1990s. Ziskind and Hertz [29] used this approach to estimate directions of arrival for narrowband autoregressive processes on a multisensor array. Weinstein *et al.* [30] estimated the parameters in a linear Gaussian model while performing noise removal in signals received on a pair of sensors with known coupling. Digalakis *et al.* [18] extended the EM algorithm to treat time-varying models and used the linear Gaussian model to represent segments of speech. Deng and Shen [31] later provided a decomposition algorithm to speed processing when the state space has a large dimension. Finally, Elliot and Krishnamurthy [32] derived an efficient filter-based implementation of the correlation matrices required during the E-step of the EM algorithm.

The equivalence between the state estimates obtained using the Baum and Viterbi algorithms and those obtained using a Kalman smoother was documented by the third author a decade ago [33], but those results never appeared in the open literature. Relationships between the Viterbi algorithm and Kalman smoother were also examined by Delyon [34] using Legendre transforms of the appropriate quadratic forms. The dependency structure of HMMs and Kalman filters also make them amenable to representation using graphical models [1], [35], and the coding and artificial intelligence communities have developed general formalisms for computing *a posteriori* densities on such graphs. These include the generalized distributive law of Aji and McEliece [36] and the sum–product algorithm of Kschischang *et al.* [37]. When applied to the graph corresponding to the HMM and linear Gaussian model, these formalisms yield the forward–backward algorithm and Kalman filter. Previously, these results were derived by treating the HMM and linear Gaussian model as separate cases. In light of the present work, the two models need not be treated separately since, indeed, linear Kalman-filter models are HMMs.

B. Contributions

This paper presents a general theory of CS-HMMs, independent of the particular form of the model densities, addressing the state estimation, likelihood evaluation, and parameter estimation problems as outlined for DS-HMMs by Ferguson [10]. The first two problems are addressed by defining continuous-state versions of the Baum and Viterbi algorithms, which are obtained using methods outlined by Jazwinski [38]. Parameter estimation is addressed, to the extent possible, by giving a general formulation of the EM auxiliary function (the E-step of the EM algorithm). The M-step is then defined by maximizing the auxiliary function after the form is specified for the model densities.

The CS-HMM results are specialized to HGMMs using a novel "Gaussian refactorization lemma," which is a realization of Bayes rule for linear Gaussian densities [39]. This analysis demonstrates that the Baum and Viterbi algorithms are implemented by fixed-interval Kalman smoothers. In particular, the forward–backward algorithm is implemented by the two-filter smoother given by Mayne [40] and Fraser and Potter [41], and the Viterbi algorithm is implemented using the *RTS algorithm* of Rauch, Tung, and Striebel [42]. While similar connections between the state-estimation algorithms are observed in [34] and [43], the present work goes a step further by providing a comprehensive framework in which linear Kalman-filter models are subsumed by HMMs. This paper also gives more substance to the equivalence by showing that the HMM measurement likelihood function is identical to the likelihood expression from Kalman filter theory [44] and that the continuous-state formulation of the Baum–Welch auxiliary function results in the EM algorithm for Kalman-filter models given in [18], [27], and [28]. The conditional joint density of time-adjacent states in HGMMs

is also derived, leading to a new expression for the cross covariance between states that is simpler than the recursive definition given by Shumway and Stoffer [27], and the measurement likelihood for time-invariant HGMMs is shown to be invariant to a family of similarity transformations of the model parameters.

The CS-HMM and HGMM algorithms are extended to accommodate *a priori* state densities that are composed of mixtures. These new developments substantially extend previous work with mixture-based Kalman filters [45], [46], first by generalizing the mixture-based algorithms to the larger class of CS-HMMs and then by addressing the smoothing and parameter estimation problems. In the classification context, the inclusion of mixture-based prior densities allows the model to accommodate greater amounts of within-class variability than can be handled using "single-mode" models.

Detailed derivations of the results given here are provided in [47] and [48].

II. CS-HMMs

Discrete-time HMMs represent a sequence of observed M-dimensional measurements $\mathbf{Z}_N = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ made at times t_n, $n = 1, \ldots, N$ as probabilistic functions of the unobserved L-dimensional states $\mathbf{X}_N = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. The state vectors form a first-order Markov process, such that $p(\mathbf{x}_n|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1})$, and the measurements are conditionally independent, given the states. If the state vectors are constrained to take values on a finite discrete set, then the model is a DS-HMM. If the elements of the state vectors are allowed to assume values on a continuum, the model is a CS-HMM. Both DS-HMMs and CS-HMMs are characterized by the *a priori* distribution for the initial state, the state-transition distribution, and the output distribution. For DS-HMMs, these distributions are parameterized by discrete probabilities, which are typically denoted as π_i, a_{ij}, and $b_t(j)$, respectively, for states i and j. For CS-HMMs, the distributions are governed by the density functions $p(\mathbf{x}_1; \theta_1)$, $p(\mathbf{x}_n|\mathbf{x}_{n-1}; \theta_X)$, and $p(\mathbf{z}_n|\mathbf{x}_n; \theta_Z)$, whose parameters are θ_1, θ_X, and θ_Z.

For notation, the following indices are defined globally.

- n nth element of an N-point sequence;
- k kth member of a K-member set of training sequences;
- i ith iteration of the EM algorithm;
- j jth component of a J-mode mixture density.

These indices may appear as subscripts, superscripts, or function arguments, but their meaning is always the same. Symbol $\mathbf{Z}_n = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ denotes the partial measurement sequence through time t_n, and $\mathbf{Z}_n^{\mathrm{C}} = \{\mathbf{z}_{n+1}, \ldots, \mathbf{z}_N\}$ is the complement of $\mathbf{Z}_n$ in $\mathbf{Z}_N$.

The state evolution in CS-HMMs is characterized by the joint density of the measurement and state sequences, or simply *joint likelihood*, which is given by

$$p(\mathbf{Z}_N, \mathbf{X}_N) = p(\mathbf{x}_1)\, p(\mathbf{z}_1|\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1})\, p(\mathbf{z}_n|\mathbf{x}_n) \tag{1}$$

where explicit parametric dependence on the model parameters is dropped for convenience. Class assignments are made using the likelihood of the measurement sequence, or *measurement likelihood*, which is obtained by marginalizing (1) over all possible state sequences, giving

$$p(\mathbf{Z}_N) = \int d\mathbf{X}_N\, p(\mathbf{Z}_N, \mathbf{X}_N). \tag{2}$$

Here, the shorthand $\int d\mathbf{X}_N$ denotes the multiple integral $\int d\mathbf{x}_1 \cdots \int d\mathbf{x}_N$, where each single integral $\int d\mathbf{x}_n$ is an L-dimensional integration over state space.

Because of the intractable computational burden required to evaluate the discrete equivalent of (2), Baum *et al.* [4] developed recursive functions to characterize and marginalize the joint probability for DS-HMMs. The continuous-state counterparts to these functions are

- the forward density $\alpha(\mathbf{x}_n)$;
- the backward function $\beta(\mathbf{x}_n)$;
- the conditional state density $\gamma(\mathbf{x}_n)$;
- the conditional joint density for time-adjacent states $\gamma(\mathbf{x}_n, \mathbf{x}_{n-1})$.

These functions depend implicitly on the parameter set $\mathbf{\Theta} = \{\theta_1, \theta_X, \theta_Z\}$.

A. Baum Density Definitions

The forward densities are fundamental to all HMM algorithms and are defined as

$$\alpha(\mathbf{x}_n) = p(\mathbf{Z}_n, \mathbf{x}_n) \tag{3}$$

which is initialized at $n = 1$ as $\alpha(\mathbf{x}_1) = p(\mathbf{z}_1|\mathbf{x}_1)p(\mathbf{x}_1)$ and is calculated recursively for $n = 2, \ldots, N$ as

$$\alpha(\mathbf{x}_n) = p(\mathbf{z}_n|\mathbf{x}_n) \int d\mathbf{x}_{n-1}\; p(\mathbf{x}_n|\mathbf{x}_{n-1})\, \alpha(\mathbf{x}_{n-1}). \tag{4}$$

It is convenient for later discussions to define

$$\begin{aligned} \delta(\mathbf{x}_n, \mathbf{x}_{n-1}) &= p(\mathbf{Z}_{n-1}, \mathbf{x}_n, \mathbf{x}_{n-1}) \\ &= p(\mathbf{x}_n|\mathbf{x}_{n-1})\, \alpha(\mathbf{x}_{n-1}) \end{aligned} \tag{5}$$

such that the recursion for $\alpha(\mathbf{x}_n)$ proceeds by first computing and marginalizing $\delta(\mathbf{x}_n, \mathbf{x}_{n-1})$ and then multiplying by the output density.

The backward functions are needed primarily as an intermediate step in calculating the conditional densities $\gamma(\mathbf{x}_n)$ and $\gamma(\mathbf{x}_n, \mathbf{x}_{n-1})$. These are defined as

$$\beta(\mathbf{x}_n) = p(\mathbf{Z}_n^{\mathrm{C}}|\mathbf{x}_n). \tag{6}$$

While this expression cannot apply at t_N since $\mathbf{Z}_N^{\mathrm{C}}$ is empty, the terminal backward function is defined as $\beta(\mathbf{x}_N) = 1$ for any $\mathbf{x}_N$ (i.e., the "diffuse prior"). With this initialization, the recursion progresses in reverse time as

$$\beta(\mathbf{x}_{n-1}) = \int d\mathbf{x}_n\, p(\mathbf{x}_n|\mathbf{x}_{n-1})\, p(\mathbf{z}_n|\mathbf{x}_n)\, \beta(\mathbf{x}_n). \tag{7}$$

Introducing the function

$$\psi(\mathbf{x}_n) = p(\mathbf{Z}_{n-1}^{\mathrm{C}}|\mathbf{x}_n) = p(\mathbf{z}_n|\mathbf{x}_n)\, \beta(\mathbf{x}_n) \tag{8}$$

allows the backward recursion to proceed by first computing $\psi(\mathbf{x}_n)$ and then multiplying by the state-transition density and marginalizing.

The conditional state densities characterize individual states when conditioned on a given measurement sequence, which is important for state and model parameter estimation. These are defined as

$$\gamma(\mathbf{x}_n) = p(\mathbf{x}_n|\mathbf{Z}_N) = \frac{1}{p(\mathbf{Z}_N)}\,\alpha(\mathbf{x}_n)\,\beta(\mathbf{x}_n). \tag{9}$$

The conditional joint state densities, which are important for model parameter estimation, are defined as

$$\begin{aligned}\gamma(\mathbf{x}_n, \mathbf{x}_{n-1}) &= p(\mathbf{x}_n, \mathbf{x}_{n-1}|\mathbf{Z}_N)\\ &= \frac{1}{p(\mathbf{Z}_N)}\,\psi(\mathbf{x}_n)\,\delta(\mathbf{x}_n, \mathbf{x}_{n-1}).\end{aligned} \tag{10}$$

B. Likelihood Evaluation and State Estimation

Given the joint density $p(\mathbf{Z}_N, \mathbf{x}_n)$ of the measurement sequence and a single state vector, the measurement likelihood is $p(\mathbf{Z}_N) = \int d\mathbf{x}_n\, p(\mathbf{Z}_N, \mathbf{x}_n)$. This marginalization is alternatively expressed as

$$\begin{aligned}p(\mathbf{Z}_N) &= \int d\mathbf{x}_n\, p(\mathbf{Z}_n^{\mathrm{C}}|\mathbf{Z}_n, \mathbf{x}_n)\, p(\mathbf{Z}_n, \mathbf{x}_n)\\ &= \int d\mathbf{x}_n\, \beta(\mathbf{x}_n)\,\alpha(\mathbf{x}_n)\end{aligned} \tag{11}$$

where it has been noted that $p(\mathbf{Z}_n^{\mathrm{C}}|\mathbf{Z}_n, \mathbf{x}_n) = p(\mathbf{Z}_n^{\mathrm{C}}|\mathbf{x}_n)$. This result demonstrates that the definition given for the conditional state density is properly normalized, that is, $\int d\mathbf{x}_n \gamma(\mathbf{x}_n) = 1$ for all n. A similar argument holds for the conditional joint state densities. Since $\beta(\mathbf{x}_N) = 1$ by definition, the simplest likelihood formula is obtained by evaluating (11) at t_N, giving

$$p(\mathbf{Z}_N) = \int d\mathbf{x}_N\, \alpha(\mathbf{x}_N). \tag{12}$$

Maximum likelihood estimates for the hidden states are obtained by maximizing the state density $\gamma(\mathbf{x}_n)$ at each time step, giving the sequence of individually most likely states. Alternatively, the Viterbi algorithm [6], [7] can be used to find the most likely sequence of states. The Viterbi algorithm is discussed in the Appendix, where it is explicitly derived for HGMMs.

C. Parameter Estimation: Single-Mode Models

Hidden-state models are natural candidates for the EM algorithm, which distinguishes three types of data:

1) "incomplete" data, which are the observed measurements;
2) "hidden" data, which are the states;
3) "complete" data, which is the concatenation of the observed and hidden data.

The likelihood of the complete data, or *complete-data likelihood function* (CDLF), is obtained from the joint density in (1).

Since time averaging cannot be performed with time-varying models, these models must be trained using multiple-independent measurement sequences. Single-sequence training *can* be performed for time-invariant models by averaging across time, although classification models so obtained may have poor generalization performance. The multisequence training set is denoted

$$\mathcal{Z} = \{\mathbf{Z}_{N_1}^1, \mathbf{Z}_{N_2}^2, \ldots, \mathbf{Z}_{N_K}^K\} \tag{13}$$

where $\mathbf{Z}_{N_k}^k = \{\mathbf{z}_1^k, \mathbf{z}_2^k, \ldots, \mathbf{z}_{N_k}^k\}$ is the kth training sequence. The training sequences need not have equal length, although they are assumed to be ordered so that $N_{\max} = N_1 \geq N_2 \geq \cdots \geq N_K$.

The EM algorithm estimates parameters iteratively, where each iteration selecting estimates that maximize the conditional expectation of the log of the CDLF, given the observed data and the parameter estimates from the previous iteration. Letting $\mathcal{X} = \{\mathbf{X}_{N_1}^1, \mathbf{X}_{N_2}^2, \ldots, \mathbf{X}_{N_K}^K\}$ denote the state sequences corresponding to the measurement training sequences, the CDLF for the training set is

$$\begin{aligned}p(\mathcal{Z}, \mathcal{X}) &= \prod_{k=1}^{K} p\left(\mathbf{Z}_{N_k}^k, \mathbf{X}_{N_k}^k\right)\\ &= \prod_{k=1}^{K} p(\mathbf{x}_1^k)\, p(\mathbf{z}_1^k|\mathbf{x}_1^k)\\ &\quad \times \prod_{n=2}^{N_k} p(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k)\, p(\mathbf{z}_n^k|\mathbf{x}_n^k).\end{aligned} \tag{14}$$

The estimates resulting from the ith iteration are

$$\boldsymbol{\Theta}^{i+1} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^i) \tag{15}$$

where the *auxiliary function*, or Q-function, is the conditional expectation

$$\begin{aligned}Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^i) &= E_{\mathcal{X}|\mathcal{Z};\boldsymbol{\Theta}^i}\{\log p(\mathcal{Z}, \mathcal{X}; \boldsymbol{\Theta})\}\\ &\equiv \int d\mathcal{X} p(\mathcal{X}|\mathcal{Z}; \boldsymbol{\Theta}^i) \log p(\mathcal{Z}, \mathcal{X}; \boldsymbol{\Theta})\\ &= \prod_{\ell=1}^{K} \int d\mathbf{X}_{N_\ell}^\ell p\left(\mathbf{X}_{N_\ell}^\ell|\mathbf{Z}_{N_\ell}^\ell; \boldsymbol{\Theta}^i\right) \log p(\mathcal{Z}, \mathcal{X}; \boldsymbol{\Theta}).\end{aligned} \tag{16}$$

The E-step evaluates the Q-function at $\boldsymbol{\Theta}^i$ from the previous iteration, yielding a function that depends only on $\boldsymbol{\Theta}$. The M-step generates $\boldsymbol{\Theta}^{i+1}$ by optimizing the Q-function obtained from the E-step. Since the M-step optimizes the model parameters, it cannot be defined generically (i.e., the specific form of the model densities must be imposed). The E-step is generically specified in greater detail, however, by imposing the dependency structure of the HMM. Dropping the explicit dependence on the model parameters, the Q-function decomposes as

$$Q = Q_1 + Q_X + Q_Z \tag{17}$$

where each component corresponds to one of the three model densities. Parameter updates for the initial-state density are obtained by maximizing

$$\begin{aligned}Q_1 &= E_{\mathcal{X}|\mathcal{Z}}\left\{\log\left[\prod_{k=1}^{K} p(\mathbf{x}_1^k)\right]\right\}\\ &= \sum_{k=1}^{K} \int d\mathbf{x}_1^k\, \gamma(\mathbf{x}_1^k) \log p(\mathbf{x}_1^k).\end{aligned} \tag{18}$$

The parameters in the state-transition density are updated by maximizing

$$Q_X = E_{\mathcal{X}|\mathcal{Z}}\left\{\log\left[\prod_{k=1}^{K}\prod_{n=2}^{N_k} p(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k)\right]\right\} = \sum_{n=2}^{N_{\max}}\sum_{k=1}^{K_n} I_{nk} \tag{19}$$

where

$$I_{nk} = \int d\mathbf{x}_n^k \int d\mathbf{x}_{n-1}^k\, \gamma(\mathbf{x}_n^k,\, \mathbf{x}_{n-1}^k)\log p(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k). \tag{20}$$

The output-density parameters are updated using

$$Q_Z = E_{\mathcal{X}|\mathcal{Z}}\left\{\log\left[\prod_{k=1}^{K}\prod_{n=1}^{N_k} p(\mathbf{z}_n^k|\mathbf{x}_n^k)\right]\right\} = \sum_{n=1}^{N_{\max}}\sum_{k=1}^{K_n}\int d\mathbf{x}_n^k\, \gamma(\mathbf{x}_n^k)\log p(\mathbf{z}_n^k|\mathbf{x}_n^k). \tag{21}$$

The latter expressions in (18), (19), and (21) are obtained by considering the final form of (16), carrying out the integrations for all states (except those that appear as arguments in the current summand), and substituting the conditional state density $\gamma(\mathbf{x}_n^k) = p(\mathbf{x}_n^k|\mathbf{Z}_{N_k}^k;\, \mathbf{\Theta}^i)$. The order of summation over k and n is reversed by introducing the upper limit K_n. For each time t_n, K_n is the number of training sequences with length $N_k \geq n$ (i.e., the number of available training samples at time t_n). These limits satisfy $K_1 \geq K_2 \geq \cdots \geq K_{N_{\max}}$. With these so defined, any function $f(\mathbf{x}_n^k)$ satisfies

$$\sum_{k=1}^{K}\sum_{n=1}^{N_k} f(\mathbf{x}_n^k) = \sum_{n=1}^{N_{\max}}\sum_{k=1}^{K_n} f(\mathbf{x}_n^k). \tag{22}$$

This equality is also satisfied for the lower limit $n = 2$, as in (19).

If the continuous variables in (18)–(21) are replaced with their discrete counterparts and the Q-function components are maximized over those variables, the Baum–Welch re-estimation formulas [4] are obtained. Because the state-transition probabilities a_{ij} enter into the model linearly, subject to the constraint that the "exiting probabilities" for the ith state must sum to unity, the re-estimation formula for a_{ij} is obtained by solving the constrained optimization problem whose Lagrangian is

$$\tilde{Q}_X = \sum_{n=2}^{N_{\max}}\sum_{k=1}^{K_n}\sum_{i}\sum_{j}\gamma_n^k(i,\, j)\log a_{ij} + \lambda\left(1 - \sum_{j} a_{ij}\right). \tag{23}$$

D. Mixed-Mode CS-HMMs

This subsection extends CS-HMMs to incorporate the prior initial-state mixture density

$$p(\mathbf{x}_1) = \sum_{j=1}^{J}\rho_j\, p(\mathbf{x}_1|j) \tag{24}$$

where $p(\mathbf{x}_1|j)$ is the jth mode in the mixture, j is the mode assignment index, and $\rho_j = p(j)$ is the mode-assignment probability or mixing parameter. The mixing parameters satisfy $\rho_j \geq 0$ for all j and $\rho_1 + \cdots + \rho_J = 1$. Substituting (24) into (1) and (2), and interchanging summation and integration operations, gives the likelihood as

$$p(\mathbf{Z}_N) = \sum_{j=1}^{J}\rho_j\, p(\mathbf{Z}_N|j) \tag{25}$$

where $p(\mathbf{Z}_N|j) = \int d\mathbf{x}_N\, \alpha_j(\mathbf{x}_N)$ and where

$$\alpha_j(\mathbf{x}_n) = p(\mathbf{Z}_n,\, \mathbf{x}_n|j) \tag{26}$$

is the forward density obtained using the Baum recursion with the single-mode prior $p(\mathbf{x}_1|j)$. The model generates a state sequence corresponding to each mode in the mixture, which is weighted by the appropriate assignment probability at any given time. The forward densities are

$$\alpha(\mathbf{x}_n) = \sum_{j=1}^{J}\rho_j\,\alpha_j(\mathbf{x}_n). \tag{27}$$

The backward function $\beta(\mathbf{x}_n)$ is the same for all j and is identical to the single-mode case. The conditional mode-assignment probabilities are defined as

$$\rho_{j|N} = p(j|\mathbf{Z}_N) = \frac{1}{p(\mathbf{Z}_N)}\,\rho_j\, p(\mathbf{Z}_N|j). \tag{28}$$

The conditional state densities are then expressed as

$$\gamma(\mathbf{x}_n) = \sum_{j=1}^{J}\rho_{j|N}\,\gamma_j(\mathbf{x}_n) \tag{29}$$

where

$$\gamma_j(\mathbf{x}_n) = p(\mathbf{x}_n|\mathbf{Z}_N,\, j) = \frac{1}{p(\mathbf{Z}_N|j)}\,\alpha_j(\mathbf{x}_n)\,\beta(\mathbf{x}_n) \tag{30}$$

is the conditional density for a single-mode model with prior $p(\mathbf{x}_1|j)$. Finally, the joint state densities are

$$\gamma(\mathbf{x}_n,\, \mathbf{x}_{n-1}) = \sum_{j=1}^{J}\rho_{j|N}\,\gamma_j(\mathbf{x}_n,\, \mathbf{x}_{n-1}) \tag{31}$$

where

$$\gamma_j(\mathbf{x}_n,\, \mathbf{x}_{n-1}) = \frac{1}{p(\mathbf{Z}_N|j)}\,\psi(\mathbf{x}_n)\,\delta_j(\mathbf{x}_n,\, \mathbf{x}_{n-1}). \tag{32}$$

Here, $\delta_j(\mathbf{x}_n,\, \mathbf{x}_{n-1})$ is computed during the forward recursion that produces $\alpha_j(\mathbf{x}_n)$.

E. Parameter Estimation—Mixed-Mode Models

For any particular measurement sequence, knowledge of the mode assignment j would reduce the mixed-mode modeling problem to a single-mode problem. The natural choice of hidden data for mixed-mode models therefore includes the mode assignment in addition to the state sequence. In this context, the CDLF is

$$p(\mathcal{Z},\, \mathcal{X},\, \mathcal{J}) = \prod_{k=1}^{K} p\left(\mathbf{Z}_{N_k}^k,\, \mathbf{X}_{N_k}^k,\, j_k\right) = \prod_{k=1}^{K}\rho_{j_k}\, p(\mathbf{x}_1^k|j_k)p(\mathbf{z}_n^k|\mathbf{x}_n^k) \times \prod_{n=2}^{N_k} p(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k)p(\mathbf{z}_n^k|\mathbf{x}_n^k) \tag{33}$$

where j_k is the mode assignment for the kth measurement sequence, and $\mathcal{J}$ represents the collection of mode assignments for all sequences. In this case, the Q function for the EM algorithm is

$$\begin{aligned} Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^i) &= E_{\mathcal{X}, \mathcal{J}|\mathcal{Z}; \boldsymbol{\Theta}^i} \{\log p(\mathcal{Z}, \mathcal{X}, \mathcal{J}; \boldsymbol{\Theta})\} \\ &= \sum_{\mathcal{J}} \int d\mathcal{X}\, p(\mathcal{X}, \mathcal{J}|\mathcal{Z}; \boldsymbol{\Theta}^i) \log p(\mathcal{Z}, \mathcal{X}, \mathcal{J}; \boldsymbol{\Theta}) \\ &= \prod_{\ell=1}^{K} \sum_{j_\ell=1}^{J} \int d\mathbf{X}_{N_\ell}^{\ell}\, p\left(\mathbf{X}_{N_\ell}^{\ell}, j_\ell|\mathbf{Z}_{N_\ell}^{\ell}; \boldsymbol{\Theta}^i\right) \\ &\qquad \times \log p(\mathcal{Z}, \mathcal{X}, \mathcal{J}; \boldsymbol{\Theta}). \end{aligned} \tag{34}$$

As before, the Q-function is decomposed such that each component depends on a different subset of model parameters. In the present case, the decomposition is

$$Q = Q_J + Q_1 + Q_X + Q_Z \tag{35}$$

where

$$\begin{aligned} Q_J &= E_{\mathcal{X}, \mathcal{J}|\mathcal{Z}} \left\{ \log \left[\prod_{k=1}^{K} \rho_{j_k} \right] \right\} \\ &= \sum_{j=1}^{J} \sum_{k=1}^{K} \rho_{j|N_k} \log \rho_j \end{aligned} \tag{36}$$

depends only on the mode-assignment probabilities, and

$$\begin{aligned} Q_1 &= E_{\mathcal{X}, \mathcal{J}|\mathcal{Z}} \left\{ \log \left[\prod_{k=1}^{K} p(\mathbf{x}_1^k|j_k) \right] \right\} \\ &= \sum_{j=1}^{J} \sum_{k=1}^{K} \int d\mathbf{x}_1^k\, \gamma(\mathbf{x}_1^k) \log p(\mathbf{x}_1^k|j) \end{aligned} \tag{37}$$

depends only on the parameters for the individual components in the initial-state mixture density. Components Q_X and Q_Z are identical to those for the single-mode model given in (19) and (21). This occurs because the relevant components from the CDLF (i.e., the product of state-transition densities for Q_X and of output densities for Q_Z) are independent of the mode assignment, such that the summation over j_ℓ serves to marginalize the mode assignments from the conditional density $p(\mathbf{X}_{N_\ell}^{\ell}, j_\ell|\mathbf{Z}_{N_\ell}^{\ell}; \boldsymbol{\Theta}^i)$. Of course, the mixed-mode nature of the model shows up in the parameter estimates via the conditional state densities.

The update for the mixing parameters is independent of the model densities and is obtained using Lagrange multiplier techniques to maximize Q_J, subject to the constraint that the ρ_j sum to one. This gives the update

$$\rho_j^{i+1} = \frac{1}{K} \sum_{k=1}^{K} \rho_{j|N_k}. \tag{38}$$

The dimensions L, M, and J are treated here as known because M is dictated by the number of features, and L and J are dimensions that are not readily estimated using likelihood-based methods. Guidance on estimating L is obtained from the traditional model-order selection literature (e.g., [49]–[51]), and a heuristic approach for estimating J goes as follows. First, estimate the parameters for a single-mode CS-HMM whose initial-state density is fixed with a large spread parameter. Use these parameters to estimate the maximum *a posteriori* state sequences for each of the training sequences, extract the initial state from each estimated sequence, and apply a multivariate clustering algorithm to the resulting collection of initial state vectors. The number of clusters provides an estimate for J. The location and spread of each cluster might also be used to estimate the parameters in each mixture component.

III. HGMMs

This section specializes CS-HMMs to model densities of the form

$$p(\mathbf{x}_1; \theta_1) = \mathcal{N}(\mathbf{x}_1; \mu_1, \mathbf{P}_1) \tag{39}$$

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}; \theta_X) = \mathcal{N}(\mathbf{x}_n; \mathbf{A}_n\mathbf{x}_{n-1}, \mathbf{Q}_n) \tag{40}$$

$$p(\mathbf{z}_n|\mathbf{x}_n; \theta_Z) = \mathcal{N}(\mathbf{z}_n; \mathbf{B}_n\mathbf{x}_n, \mathbf{R}_n) \tag{41}$$

where $\mathcal{N}(\mathbf{y}; \mu, \mathbf{P})$ denotes the density function for a multivariate normal vector $\mathbf{y}$ with mean μ and covariance matrix $\mathbf{P}$. The time-varying model densities are parameterized by $\theta_1 = \{\mu_1, \mathbf{P}_1\}$, $\theta_X = \{\mathbf{A}_n, \mathbf{Q}_n n = 2, \ldots, N\}$, and $\theta_Z = \{\mathbf{B}_n, \mathbf{R}_n n = 1, \ldots, N\}$. Matrix $\mathbf{P}_1$ and all $\mathbf{A}_n$, $\mathbf{Q}_n$, and $\mathbf{R}_n$ are assumed nonsingular.

Development of the Baum algorithm for HGMMs involves recurring sets of operations, which are summarized by the following *Gaussian refactorization lemma* (GRL).

Lemma (GRL): Given the q-dimensional vectors $\mathbf{x}$ and μ, the nonsingular symmetric $q \times q$ matrix $\mathbf{P}$, the p-dimensional vector $\mathbf{y}$, the $p \times q$ matrix $\mathbf{F}$, and the nonsingular symmetric $p \times p$ matrix $\mathbf{S}$, then

$$\mathcal{N}(\mathbf{y}; \mathbf{F}\mathbf{x}, \mathbf{S})\ \mathcal{N}(\mathbf{x}; \mu, \mathbf{P}) = \mathcal{N}(\mathbf{y}; \omega, \boldsymbol{\Omega})\ \mathcal{N}(\mathbf{x}; \lambda, \boldsymbol{\Lambda}) \tag{42}$$

where the variables on the right-hand side are defined by

$$\boldsymbol{\Omega} = \mathbf{S} + \mathbf{F}\mathbf{P}\mathbf{F}^{\mathrm{T}} \tag{43}$$

$$\omega = \mathbf{F}\mu \tag{44}$$

$$\mathbf{H} = \mathbf{P}\mathbf{F}^{\mathrm{T}}\boldsymbol{\Omega}^{-1} \tag{45}$$

$$\boldsymbol{\Lambda} = (\mathbf{I} - \mathbf{H}\mathbf{F})\mathbf{P} \tag{46}$$

$$\lambda = (\mathbf{I} - \mathbf{H}\mathbf{F})\mu + \mathbf{H}\mathbf{y}. \tag{47}$$

The algebraic proof is given in [47, App. A]. Equations (43) and (44) are Kalman-filter time updates, (45) is the Kalman gain matrix, and (46) and (47) are Kalman filter measurement updates. As noted by Parzen [39], the GRL is a realization of Bayes rule $p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) = p(\mathbf{y})p(\mathbf{x}|\mathbf{y})$.

The GRL is now used to inductively derive the HGMM forward recursions. Standard Kalman-filter subscripting is adopted in the remainder of this section, for example, $\mu_{n|n} = E(\boldsymbol{x}_n|\mathbf{Z}_n)$.

A. Forward Densities

The forward densities are obtained using a two-stage recursion, starting with the assumed form

$$\alpha(\mathbf{x}_{n-1}) = c_{n-1}\mathcal{N}\left(\mathbf{x}_{n-1}; \mu_{n-1|n-1}, \mathbf{P}_{n-1|n-1}\right). \tag{48}$$

The first stage of the recursion evaluates $\delta(\mathbf{x}_n, \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{A}_n\mathbf{x}_{n-1}, \mathbf{Q}_n)\alpha(\mathbf{x}_{n-1})$. Applying the GRL gives

$$\begin{aligned}\delta(\mathbf{x}_n, \mathbf{x}_{n-1}) &= c_{n-1}\,\mathcal{N}\left(\mathbf{x}_n; \mu_{n|n-1}, \mathbf{P}_{n|n-1}\right)\\ &\quad\times\mathcal{N}\left(\mathbf{x}_{n-1}; \lambda_n, \mathbf{\Lambda}_n\right)\end{aligned} \tag{49}$$

whose variables are defined by

$$\mathbf{P}_{n|n-1} = \mathbf{Q}_n + \mathbf{A}_n\,\mathbf{P}_{n-1|n-1}\,\mathbf{A}_n^{\mathrm{T}} \tag{50}$$

$$\mu_{n|n-1} = \mathbf{A}_n\,\mu_{n-1|n-1} \tag{51}$$

$$\mathbf{H}_n = \mathbf{P}_{n-1|n-1}\,\mathbf{A}_n^{\mathrm{T}}\,\mathbf{P}_{n|n-1}^{-1} \tag{52}$$

$$\mathbf{\Lambda}_n = (\mathbf{I} - \mathbf{H}_n\mathbf{A}_n)\mathbf{P}_{n-1|n-1} \tag{53}$$

$$\lambda_n = (\mathbf{I} - \mathbf{H}_n\mathbf{A}_n)\mu_{n-1|n-1} + \mathbf{H}_n\,\mathbf{x}_n. \tag{54}$$

Whereas λ_n depends on $\mathbf{x}_n$, integrating $\mathcal{N}(\mathbf{x}_{n-1}; \lambda_n, \mathbf{\Lambda}_n)$ over all $\mathbf{x}_{n-1}$ produces unity, regardless of the mean. After thus marginalizing, multiplying by the output density $\mathcal{N}(\mathbf{z}_n; \mathbf{B}_n\mathbf{x}_n, \mathbf{R}_n)$, and again applying the GRL, the forward density becomes

$$\alpha(\mathbf{x}_n) = c_{n-1}\,\mathcal{N}\left(\mathbf{x}_n; \mu_{n|n}, \mathbf{P}_{n|n}\right)\mathcal{N}\left(\mathbf{z}_n; \hat{\mathbf{z}}_{n|n-1}, \mathbf{\Sigma}_n\right) \tag{55}$$

where

$$\mathbf{\Sigma}_n = \mathbf{R}_n + \mathbf{B}_n\,\mathbf{P}_{n|n-1}\,\mathbf{B}_n^{\mathrm{T}} \tag{56}$$

$$\hat{\mathbf{z}}_{n|n-1} = \mathbf{B}_n\,\mu_{n|n-1} \tag{57}$$

$$\mathbf{G}_n = \mathbf{P}_{n|n-1}\,\mathbf{B}_n^{\mathrm{T}}\,\mathbf{\Sigma}_n^{-1} \tag{58}$$

$$\mathbf{P}_{n|n} = (\mathbf{I} - \mathbf{G}_n\mathbf{B}_n)\,\mathbf{P}_{n|n-1} \tag{59}$$

$$\mu_{n|n} = (\mathbf{I} - \mathbf{G}_n\mathbf{B}_n)\,\mu_{n|n-1} + \mathbf{G}_n\,\mathbf{z}_n. \tag{60}$$

Since all variables in the factor $\mathcal{N}(\mathbf{z}_n; \hat{\mathbf{z}}_{n|n-1}, \mathbf{\Sigma}_n)$ are constant, it is absorbed into c_n to obtain the recursion

$$c_n = c_{n-1}\,\mathcal{N}\left(\mathbf{z}_n; \hat{\mathbf{z}}_{n|n-1}, \mathbf{\Sigma}_n\right). \tag{61}$$

The updated forward density then takes the desired form

$$\alpha(\mathbf{x}_n) = c_n\mathcal{N}\left(\mathbf{x}_n; \mu_{n|n}, \mathbf{P}_{n|n}\right). \tag{62}$$

The induction is completed by showing that the initialization matches the assumed form. The forward densities are initialized as

$$\begin{aligned}\alpha(\mathbf{x}_1) &= p(\mathbf{z}_1|\mathbf{x}_1)\,p(\mathbf{x}_1)\\ &= \mathcal{N}(\mathbf{z}_1; \mathbf{B}_1\mathbf{x}_1, \mathbf{R}_1)\,\mathcal{N}(\mathbf{x}_1; \mu_1, \mathbf{P}_1).\end{aligned} \tag{63}$$

Applying the GRL yields expressions similar to (55)–(62), which satisfy the assumed form.

B. Measurement Likelihood

The likelihood of a measurement sequence is obtained by marginalizing the state vector from the forward density, which is defined in (62), at time t_N. The term $\mathcal{N}(\mathbf{x}_N; \mu_{N|N}, \mathbf{P}_{N|N})$ integrates to unity, leaving

$$p(\mathbf{Z}_N) = c_N = \prod_{n=1}^{N}\mathcal{N}(\nu_n; \mathbf{0}, \mathbf{\Sigma}_n) \tag{64}$$

where $\nu_n = \mathbf{z}_n - \hat{\mathbf{z}}_{n|n-1}$ is the *measurement innovation*. This is identical to the traditional definition from Kalman-filter theory [44].

C. Backward Functions

When deriving the backward functions, the state representation must be altered to accommodate $\beta(\mathbf{x}_N) = 1$. This function does not fit the form of a Gaussian density, but it is representable using

$$\mathcal{G}(\mathbf{y}; \xi, \mathbf{\Gamma}) = \exp\left\{-(1/2)\,(\mathbf{\Gamma}\mathbf{y} - \xi)^{\mathrm{T}}\mathbf{\Gamma}^{\dagger}(\mathbf{\Gamma}\mathbf{y} - \xi)\right\} \tag{65}$$

where ξ is the *information vector*, $\mathbf{\Gamma}$ is the *information matrix*, and $\mathbf{\Gamma}^{\dagger}$ is the pseudoinverse of $\mathbf{\Gamma}$ (which satisfies $\mathbf{\Gamma}\mathbf{\Gamma}^{\dagger}\mathbf{\Gamma} = \mathbf{\Gamma}$ among other properties; see [52, Ch. 6]). When $\mathbf{\Gamma}$ is singular, $\mathcal{G}(\mathbf{y};\ \xi, \mathbf{\Gamma})$ is well defined, even though there is no valid density corresponding to $\mathcal{G}$. When $\mathbf{\Gamma}$ is nonsingular, a Gaussian density with covariance $\mathbf{P} = \mathbf{\Gamma}^{-1}$ and mean $\mu = \mathbf{\Gamma}^{-1}\xi$ is defined in terms of (65) as

$$\mathcal{N}(\mathbf{y};\,\mu,\,\mathbf{P}) = (2\pi)^{-q/2}|\mathbf{P}|^{-1/2}\mathcal{G}(\mathbf{y};\,\mathbf{P}^{-1}\mu,\,\mathbf{P}^{-1}). \tag{66}$$

When dealing with information representations of the form in (65), it is convenient to introduce a weaker statement of the GRL that handles both singular and nonsingular $\mathbf{\Gamma}$. Defining the q-dimensional vectors $\mathbf{x}$ and ξ, the p-dimensional vector $\mathbf{y}$, the symmetric (possibly singular) $q \times q$ matrix $\mathbf{\Gamma}$, the nonsingular symmetric $p \times p$ matrix $\mathbf{S}$, and the $p \times q$ matrix $\mathbf{F}$

$$\begin{aligned}&\mathcal{N}(\mathbf{y};\,\mathbf{F}\mathbf{x},\,\mathbf{S})\,\mathcal{G}(\mathbf{x};\,\xi,\,\mathbf{\Gamma})\\ &\qquad= (2\pi)^{-p/2}|\mathbf{S}|^{-1/2}\,\mathcal{G}(\mathbf{y};\,\zeta,\,\mathbf{\Upsilon})\,\mathcal{G}(\mathbf{x};\eta,\,\mathbf{\Pi})\end{aligned} \tag{67}$$

where the variables in the new factors are

$$\mathbf{\Pi} = \mathbf{\Gamma} + \mathbf{F}^{\mathrm{T}}\,\mathbf{S}^{-1}\,\mathbf{F} \tag{68}$$

$$\eta = \xi + \mathbf{F}^{\mathrm{T}}\,\mathbf{S}^{-1}\,\mathbf{y} \tag{69}$$

$$\mathbf{\Upsilon} = \mathbf{S}^{-1} - \mathbf{S}^{-1}\,\mathbf{F}\,\mathbf{\Pi}^{\dagger}\,\mathbf{F}^{\mathrm{T}}\,\mathbf{S}^{-1} \tag{70}$$

$$\zeta = \mathbf{S}^{-1}\,\mathbf{F}\,\mathbf{\Pi}^{\dagger}\,\xi. \tag{71}$$

Equations (68) and (69) are measurement updates, and (70) and (71) are time updates for the information formulation of a Kalman filter. These expressions are used here to develop the backward functions The information representation is also needed for the forward densities if some of the state variables naturally have a diffuse prior, in which case, $\mathbf{P}_1$ does not exist.

The backward recursions are now defined. The assumed form for the backward function is

$$\beta(\mathbf{x}_n) = d_n\,\mathcal{G}\left(\mathbf{x}_n; \xi_{n|n+1}, \mathbf{\Gamma}_{n|n+1}\right). \tag{72}$$

Setting $\mathbf{\Gamma}_{N|N+1} = \mathbf{0}$, $\xi_{N|N+1} = \mathbf{0}$, and $d_N = 1$ gives the desired initialization $\beta(\mathbf{x}_N) = 1$.

The first stage of the recursion evaluates the product $\psi(\mathbf{x}_n) = \mathcal{N}(\mathbf{z}_n; \mathbf{B}_n\mathbf{x}_n, \mathbf{R}_n)\beta(\mathbf{x}_n)$. Applying (67) to this product gives

$$\begin{aligned}\psi(\mathbf{x}_n) &= d_n\,(2\pi)^{-M/2}\,|\mathbf{R}_n|^{-1/2}\\ &\quad\times\mathcal{G}\left(\mathbf{x}_n;\,\xi_{n|n},\,\mathbf{\Gamma}_{n|n}\right)\mathcal{G}(\mathbf{z}_n;\,\zeta_n,\,\mathbf{\Upsilon}_n)\end{aligned} \tag{73}$$

where the measurement-updated variables are

$$\mathbf{\Gamma}_{n|n} = \mathbf{\Gamma}_{n|n+1} + \mathbf{B}_n^{\mathrm{T}}\,\mathbf{R}_n^{-1}\,\mathbf{B}_n \tag{74}$$

$$\xi_{n|n} = \xi_{n|n+1} + \mathbf{B}_n^{\mathrm{T}}\,\mathbf{R}_n^{-1}\,\mathbf{z}_n \tag{75}$$

$$\mathbf{\Upsilon}_n = \mathbf{R}_n^{-1} - \mathbf{R}_n^{-1}\,\mathbf{B}_n\,\mathbf{\Gamma}_{n|n}^{\dagger}\,\mathbf{B}_n^{\mathrm{T}}\,\mathbf{R}_n^{-1} \tag{76}$$

$$\zeta_n = \mathbf{R}_n^{-1}\,\mathbf{B}_n\,\mathbf{\Gamma}_{n|n}^{\dagger}\,\xi_{n|N+1}. \tag{77}$$

Defining the scale constant

$$e_n = d_n(2\pi)^{-M/2}|\mathbf{R}_n|^{-1/2}\,\mathcal{G}(\mathbf{z}_n;\,\zeta_n,\,\boldsymbol{\Upsilon}_n) \tag{78}$$

allows $\psi(\mathbf{x}_n)$ to be expressed as

$$\psi(\mathbf{x}_n) = e_n\,\mathcal{G}\left(\mathbf{x}_n;\,\xi_{n|n},\,\boldsymbol{\Gamma}_{n|n}\right). \tag{79}$$

The second stage of the recursion marginalizes $\mathbf{x}_n$ from the product $\mathcal{N}(\mathbf{x}_n;\,\mathbf{A}_n\mathbf{x}_{n-1},\,\mathbf{Q}_n)\psi(\mathbf{x}_n)$. This product does not immediately fit the form of (67), but for invertible $\mathbf{A}_n$, the variables in the state-transition density can be rearranged to yield

$$\begin{aligned}&\mathcal{N}(\mathbf{x}_n;\,\mathbf{A}_n\mathbf{x}_{n-1},\,\mathbf{Q}_n)\\ &\quad= |\mathbf{A}_n|^{-1}\mathcal{N}\left(\mathbf{x}_{n-1};\,\mathbf{A}_n^{-1}\mathbf{x}_n,\,\mathbf{A}_n^{-1}\mathbf{Q}_n\mathbf{A}_n^{-\mathrm{T}}\right).\end{aligned} \tag{80}$$

Applying (67) to the resulting product and marginalizing $\mathbf{x}_n$ gives the desired form

$$\beta(\mathbf{x}_{n-1}) = d_{n-1}\,\mathcal{G}\left(\mathbf{x}_{n-1};\,\xi_{n-1|n},\,\boldsymbol{\Gamma}_{n-1|n}\right) \tag{81}$$

where the reverse time updates are defined by

$$\boldsymbol{\Pi}_n = \boldsymbol{\Gamma}_{n|n} + \mathbf{Q}_n^{-1} \tag{82}$$

$$\boldsymbol{\Gamma}_{n-1|n} = \mathbf{A}_n^{\mathrm{T}}\left\{\mathbf{Q}_n^{-1} - \mathbf{Q}_n^{-1}\,\boldsymbol{\Pi}_n^{-1}\,\mathbf{Q}_n^{-1}\right\}\mathbf{A}_n \tag{83}$$

$$\xi_{n-1|n} = \mathbf{A}_n^{\mathrm{T}}\,\mathbf{Q}_n^{-1}\,\boldsymbol{\Pi}_n^{-1}\,\xi_{n|n} \tag{84}$$

and the scale constant is

$$d_{n-1} = e_n\,|\mathbf{A}_n|^{-1}\,|\mathbf{Q}_n|^{-1/2}\,|\boldsymbol{\Pi}_n|^{-1/2}. \tag{85}$$

Note that $\boldsymbol{\Pi}_n$ appears with an inverse instead of pseudoinverse because it is nonsingular. While recursions for the backward scale constants d_n and e_n are developed here to ensure that all $\beta(\mathbf{x}_n)$ are well defined, d_n and e_n are not required for the conditional state density and can be omitted from computational algorithms.

In classification applications, the state-space dimension is typically larger than the measurement dimension ($L > M$). With $\boldsymbol{\Gamma}_{N|N+1} = \mathbf{0}$, the rank of the information matrix therefore takes a few backward recursions to "grow" from 0 to L. Assuming that $\mathbf{B}_n$ has full row rank, then the measurement update in (74) is such that

$$\mathrm{rank}(\boldsymbol{\Gamma}_{n|n}) = \min\left\{L,\,\mathrm{rank}\left(\boldsymbol{\Gamma}_{n|n+1}\right) + M\right\}. \tag{86}$$

The time update in (83), on the other hand, does not alter the rank of the information matrix, which is shown as follows. Let $\boldsymbol{\Gamma}_{n|n}$ have rank $r \le L$, let the $L \times r$ matrix $\mathbf{U}_r$ contain its r principal eigenvectors, and let $r \times r$ diagonal matrix $\boldsymbol{\Lambda}_r$ contain its nonzero eigenvalues. Substituting $\boldsymbol{\Gamma}_{n|n} = \mathbf{U}_r\boldsymbol{\Lambda}_r\mathbf{U}_r^{\mathrm{T}}$ into (82), applying the matrix inversion lemma to invert $\boldsymbol{\Pi}_n$, substituting the resulting expression for $\boldsymbol{\Pi}_n^{-1}$ into (83), and simplifying gives

$$\boldsymbol{\Gamma}_{n-1|n} = \mathbf{U}_r\left(\boldsymbol{\Lambda}_r^{-1} + \mathbf{U}_r^{\mathrm{T}}\mathbf{Q}_n^{-1}\mathbf{U}_r\right)^{-1}\mathbf{U}_r^{\mathrm{T}}. \tag{87}$$

This structure has rank r. Given the effects of the time and measurement updates, the information matrices are singular at time steps $n = \{N - \lceil L/M\rceil + 1,\,\ldots,\,N\}$. At earlier times, $\boldsymbol{\Gamma}_{n-1|n}$ is nonsingular, and $\beta(\mathbf{x}_n)$ is a weighted Gaussian density.

D. Conditional State Densities

The conditional state density $\gamma(\mathbf{x}_n)$ is the normalized product of $\alpha(\mathbf{x}_n)$ and $\beta(\mathbf{x}_n)$ as defined in (9). Substituting (62) and (72) and expressing $\alpha(\mathbf{x}_n)$ in the form of (66) gives the conditional density as

$$\begin{aligned}\gamma(\mathbf{x}_n) &= \frac{1}{p(\mathbf{Z}_N)}\,c_n\,d_n(2\pi)^{-L/2}|\mathbf{P}_{n|n}|^{-1/2}\\ &\quad\times\mathcal{G}\left(\mathbf{x}_n;\,\mathbf{P}_{n|n}^{-1}\mu_{n|n},\mathbf{P}_{n|n}^{-1}\right)\mathcal{G}\left(\mathbf{x}_n;\,\xi_{n|n+1},\,\boldsymbol{\Gamma}_{n|n+1}\right).\end{aligned} \tag{88}$$

Since $\gamma(\mathbf{x}_n)$ is properly normalized by definition, the scale constant in (88) need not be computed. The product of exponential functions in (88) takes a well-known form, which is normalized to obtain the density function

$$\gamma(\mathbf{x}_n) = \mathcal{N}\left(\mathbf{x}_n;\,\mu_{n|N},\,\mathbf{P}_{n|N}\right) \tag{89}$$

where the mean and covariance are

$$\mathbf{P}_{n|N} = \left(\boldsymbol{\Gamma}_{n|n+1} + \mathbf{P}_{n|n}^{-1}\right)^{-1} \tag{90}$$

$$\mu_{n|N} = \mathbf{P}_{n|N}\left(\xi_{n|n+1} + \mathbf{P}_{n|n}^{-1}\,\mu_{n|n}\right). \tag{91}$$

The computations for $\alpha(\mathbf{x}_n)$, $\beta(\mathbf{x}_n)$, and $\gamma(\mathbf{x}_n)$ outlined above are identical to the two-filter implementation of the fixed-interval Kalman smoother [40], [41].

E. Conditional Joint State Densities

A full characterization of the states requires the joint density of time-adjacent states. Substituting (49) and (79) into (10) gives

$$\begin{aligned}\gamma(\mathbf{x}_n,\,\mathbf{x}_{n-1}) &= h_n\,\mathcal{G}\left(\mathbf{x}_n;\,\xi_{n|n},\,\boldsymbol{\Gamma}_{n|n}\right)\\ &\quad\times\mathcal{G}\left(\mathbf{x}_n;\,\mathbf{P}_{n|n-1}^{-1}\mu_{n|n-1},\,\mathbf{P}_{n|n-1}^{-1}\right)\\ &\quad\times\mathcal{N}(\mathbf{x}_{n-1};\,\lambda_n,\,\boldsymbol{\Lambda}_n)\end{aligned} \tag{92}$$

where $\mathbf{P}_{n|n-1}$, $\mu_{n|n-1}$, $\boldsymbol{\Lambda}_n$, λ_n, $\boldsymbol{\Gamma}_{n|n}$, and $\xi_{n|n}$ are defined in (50), (51), (53), (54), (74), and (75), respectively, and

$$h_n = \frac{1}{p(\mathbf{Z}_N)}\,c_{n-1}\,e_n\,(2\pi)^{-L/2}\left|\mathbf{P}_{n|n-1}\right|^{-1/2}. \tag{93}$$

Since $\gamma(\mathbf{x}_n) = \int d\mathbf{x}_{n-1}\,\gamma(\mathbf{x}_n,\,\mathbf{x}_{n-1})$ and only the last term in (92) depends on $\mathbf{x}_{n-1}$, an alternative expression for the conditional state density is

$$\begin{aligned}\gamma(\mathbf{x}_n) &= h_n\,\mathcal{G}\left(\mathbf{x}_n;\,\xi_{n|n},\,\boldsymbol{\Gamma}_{n|n}\right)\\ &\quad\times\mathcal{G}\left(\mathbf{x}_n;\,\mathbf{P}_{n|n-1}^{-1}\mu_{n|n-1},\,\mathbf{P}_{n|n-1}^{-1}\right).\end{aligned} \tag{94}$$

The joint density is therefore

$$\gamma(\mathbf{x}_n,\,\mathbf{x}_{n-1}) = \gamma(\mathbf{x}_n)\,\mathcal{N}(\mathbf{x}_{n-1};\,\lambda_n,\,\boldsymbol{\Lambda}_n). \tag{95}$$

This expression is used when deriving estimates for the parameters in the state-transition density. As is shown in [47, App. B], the joint density is alternatively written as

$$\gamma(\mathbf{x}_n,\,\mathbf{x}_{n-1}) = \mathcal{N}\left(\mathbf{x}_{[n,\,n-1]};\,\mu_{[n,\,n-1]|N},\,\mathbf{P}_{[n,\,n-1]|N}\right) \tag{96}$$

where $\mathbf{x}_{[n,\,n-1]} = [\mathbf{x}_n^{\mathrm{T}},\,\mathbf{x}_{n-1}^{\mathrm{T}}]^{\mathrm{T}}$ is the $2L \times 1$ joint random vector with mean $\mu_{[n,\,n-1]|N} = [\mu_{n|N}^{\mathrm{T}},\,\mu_{n-1|N}^{\mathrm{T}}]^{\mathrm{T}}$ and covariance matrix

$$\mathbf{P}_{[n,\,n-1]|N} = \begin{bmatrix}\mathbf{P}_{n|N} & \mathbf{P}_{n|N}\mathbf{H}_n^{\mathrm{T}}\\ \mathbf{H}_n\mathbf{P}_{n|N} & \mathbf{P}_{n-1|N}\end{bmatrix} \tag{97}$$

where $\mathbf{H}_n$ is defined in (52). The upper off-diagonal gives the adjacent-state cross-covariance matrix as

$$\mathbf{P}_{n,\,n-1|N} = \mathbf{P}_{n|N}\,\mathbf{H}_n^{\mathrm{T}} \tag{98}$$

which is considerably simpler than the recursive definition given in [27].

F. Parameter Estimation

For HGMMs, the E-step of the EM algorithm evaluates the Q-function components defined in (18), (19), and (21), which requires the identity [52, Ch. 10]

$$\int d\mathbf{x}\mathcal{N}(\mathbf{x};\,\mu,\,\mathbf{P})\left(\mathbf{x}^{\mathrm{T}}\mathbf{F}\mathbf{x}+\mathbf{x}^{\mathrm{T}}\mathbf{f}+f_0\right) = \mathrm{tr}\left\{\mathbf{F}\left(\mathbf{P}+\mu\mu^{\mathrm{T}}\right)\right\}+\mu^{\mathrm{T}}\mathbf{f}+f_0. \quad (99)$$

The M-step maximizes the Q-function components obtained from the E-step. In addition to standard matrix and trace derivatives, this requires the identity [53]

$$\frac{\partial}{\partial\mathbf{F}}\mathrm{tr}\left(\mathbf{F}^{\mathrm{T}}\mathbf{S}_1\mathbf{F}\mathbf{S}_2\right)=\mathbf{S}_1\mathbf{F}\mathbf{S}_2+\mathbf{S}_1^{\mathrm{T}}\mathbf{F}\mathbf{S}_2^{\mathrm{T}}. \quad (100)$$

Component Q_1 is treated first. Defining $\varepsilon_1^k=\mu_{1|N_k}^k-\mu_1$ gives

$$Q_1=\sum_{k=1}^{K}\int d\mathbf{x}_1^k\mathcal{N}\left(\mathbf{x}_1^k;\,\mu_{1|N_k}^k,\,\mathbf{P}_{1|N_k}^k\right)\log\mathcal{N}(\mathbf{x}_1^k;\,\mu_1,\,\mathbf{P}_1)$$
$$=-\frac{K}{2}\log|\mathbf{P}_1|^{-1}+\frac{1}{2}\sum_{k=1}^{K}\mathrm{tr}\left\{\mathbf{P}_1^{-1}\left(\mathbf{P}_{1|N_k}^k+\varepsilon_1^k\varepsilon_1^{k^{\mathrm{T}}}\right)\right\} \quad (101)$$

where the constant term $KL\log(2\pi)/2$ is neglected in the second expression. Differentiating with respect to μ_1 and equating the result to zero gives the update

$$\mu_1^{i+1}=\frac{1}{K}\sum_{k=1}^{K}\mu_{1|N_k}^k. \quad (102)$$

Equating to zero the derivative of Q_1 with respect to $\mathbf{P}_1^{-1}$, substituting (102), and defining $\varepsilon_1^{k,\,i+1}=\mu_{1|N_k}^k-\mu_1^{i+1}$ gives

$$\mathbf{P}_1^{i+1}=\frac{1}{K}\sum_{k=1}^{K}\left\{\mathbf{P}_{1|N_k}^k+\varepsilon_1^{k,\,i+1}\varepsilon_1^{k,i+1^{\mathrm{T}}}\right\}. \quad (103)$$

Component Q_X is evaluated by carrying out the double integration of I_{nk} in (20). Substituting the definitions of $p(\mathbf{x}_n^k|\mathbf{x}_{n-1}^k)$ and $\gamma(\mathbf{x}_n^k,\,\mathbf{x}_{n-1}^k)$, expanding the quadratic form in the log term, and neglecting the $L\log(2\pi)/2$ term yields

$$I_{nk}=-\tfrac{1}{2}\left\{\log|\mathbf{Q}_n|+I_{nk}^{(1)}+I_{nk}^{(2)}+I_{nk}^{(3)}\right\} \quad (104)$$

where

$$I_{nk}^{(1)}=\int d\mathbf{x}_n^k\,\gamma(\mathbf{x}_n^k)\,\mathbf{x}_n^{k^{\mathrm{T}}}\mathbf{Q}_n^{-1}\mathbf{x}_n^k \quad (105)$$

$$I_{nk}^{(2)}=\int d\mathbf{x}_n^k\int d\mathbf{x}_{n-1}^k\,\gamma(\mathbf{x}_n^k,\,\mathbf{x}_{n-1}^k)\times\left\{\mathbf{x}_n^{k^{\mathrm{T}}}\mathbf{Q}_n^{-1}\mathbf{A}_n\mathbf{x}_{n-1}^k+\mathbf{x}_{n-1}^{k^{\mathrm{T}}}\mathbf{A}_n^{\mathrm{T}}\mathbf{Q}_n^{-1}\mathbf{x}_n^k\right\} \quad (106)$$

$$I_{nk}^{(3)}=\int d\mathbf{x}_{n-1}^k\,\gamma(\mathbf{x}_{n-1}^k)\,\mathbf{x}_{n-1}^{k^{\mathrm{T}}}\mathbf{A}_n^{\mathrm{T}}\mathbf{Q}_n^{-1}\mathbf{A}_n\mathbf{x}_{n-1}^k. \quad (107)$$

$I_{nk}^{(1)}$ and $I_{nk}^{(3)}$ are evaluated using (99). $I_{nk}^{(2)}$ is evaluated by substituting (95) and integrating first with respect to $\mathbf{x}_{n-1}$ and then with respect to $\mathbf{x}_n$. The result is

$$Q_X=\sum_{n=2}^{N_{\max}}\Big\{\log|\mathbf{Q}_n^{-1}|-\mathrm{tr}\left(\mathbf{Q}_n^{-1}\mathbf{C}_{\mathbf{x}_n\mathbf{x}_n}\right)+\mathrm{tr}\left(\mathbf{Q}_n^{-1}\mathbf{A}_n\mathbf{C}_{\mathbf{x}_n\mathbf{x}_{n-1}}^{\mathrm{T}}\right)+\mathrm{tr}\left(\mathbf{Q}_n^{-1}\mathbf{C}_{\mathbf{x}_n\mathbf{x}_{n-1}}\mathbf{A}_n^{\mathrm{T}}\right)-\mathrm{tr}\left(\mathbf{A}_n^{\mathrm{T}}\mathbf{Q}_n^{-1}\mathbf{A}_n\mathbf{C}_{\mathbf{x}_{n-1}\mathbf{x}_{n-1}}\right)\Big\} \quad (108)$$

where the $\log(2\pi)$ term and the scale factor $1/2$ have been disregarded, and where

$$\mathbf{C}_{\mathbf{x}_n\mathbf{x}_n}=\sum_{k=1}^{K_n}\left\{\mathbf{P}_{n|N_k}^k+\mu_{n|N_k}^k\mu_{n|N_k}^{k^{\mathrm{T}}}\right\} \quad (109)$$

$$\mathbf{C}_{\mathbf{x}_n\mathbf{x}_{n-1}}=\sum_{k=1}^{K_n}\left\{\mathbf{P}_{n,\,n-1|N_k}^k+\mu_{n|N_k}^k\mu_{n-1|N_k}^{k^{\mathrm{T}}}\right\}. \quad (110)$$

The EM update for $\mathbf{A}_n$ is obtained by equating to zero the derivative of Q_X, which gives

$$\mathbf{A}_n^{i+1}=\mathbf{C}_{\mathbf{x}_n\mathbf{x}_{n-1}}\mathbf{C}_{\mathbf{x}_{n-1}\mathbf{x}_{n-1}}^{-1}. \quad (111)$$

The second and third trace terms in (108) are equal but are kept separate to facilitate estimation of the covariance matrix when $\mathbf{A}_n$ is fixed (e.g., in the tracking problem). In this case, the EM update for $\mathbf{Q}_n$ (obtained by differentiating Q_X with respect to $\mathbf{Q}_n^{-1}$) is

$$\mathbf{Q}_n^{i+1}=\frac{1}{K_n}\left\{\mathbf{C}_{\mathbf{x}_n\mathbf{x}_n}-\mathbf{C}_{\mathbf{x}_n\mathbf{x}_{n-1}}\mathbf{A}_n^{\mathrm{T}}-\mathbf{A}_n\mathbf{C}_{\mathbf{x}_n\mathbf{x}_{n-1}}^{\mathrm{T}}+\mathbf{A}_n\mathbf{C}_{\mathbf{x}_{n-1}\mathbf{x}_{n-1}}^{-1}\mathbf{A}_n\right\}. \quad (112)$$

When $\mathbf{A}_n$ is unknown, the covariance estimate

$$\mathbf{Q}_n^{i+1}=\frac{1}{K_n}\left\{\mathbf{C}_{\mathbf{x}_n\mathbf{x}_n}-\mathbf{C}_{\mathbf{x}_n\mathbf{x}_{n-1}}\mathbf{C}_{\mathbf{x}_{n-1}\mathbf{x}_{n-1}}^{-1}\mathbf{C}_{\mathbf{x}_n\mathbf{x}_{n-1}}^{\mathrm{T}}\right\} \quad (113)$$

is obtained by substituting (111) into (112).

Component Q_Z is evaluated in a manner similar to Q_X, giving

$$Q_Z=\sum_{n=1}^{N_{\max}}\left\{\log|\mathbf{R}_n^{-1}|-\mathrm{tr}\left(\mathbf{R}_n^{-1}\mathbf{C}_{\mathbf{z}_n\mathbf{z}_n}\right)+\mathrm{tr}\left(\mathbf{R}_n^{-1}\mathbf{B}_n\mathbf{C}_{\mathbf{z}_n\mathbf{x}_n}^{\mathrm{T}}\right)+\mathrm{tr}\left(\mathbf{R}_n^{-1}\mathbf{C}_{\mathbf{z}_n\mathbf{x}_n}\mathbf{B}_n^{\mathrm{T}}\right)-\mathrm{tr}\left(\mathbf{B}_n^{\mathrm{T}}\mathbf{R}_n^{-1}\mathbf{B}_n\mathbf{C}_{\mathbf{x}_n\mathbf{x}_n}\right)\right\} \quad (114)$$

where

$$\mathbf{C}_{\mathbf{z}_n\mathbf{z}_n}=\sum_{k=1}^{K_n}\mathbf{z}_n^k\,\mathbf{z}_n^{k^{\mathrm{T}}} \quad (115)$$

$$\mathbf{C}_{\mathbf{z}_n\mathbf{x}_n}=\sum_{k=1}^{K_n}\mathbf{z}_n^k\,\mu_{n|N_k}^{k^{\mathrm{T}}}. \quad (116)$$

Maximizing Q_Z using the same steps as for Q_X yields

$$\mathbf{B}_n^{i+1}=\mathbf{C}_{\mathbf{z}_n\mathbf{x}_n}\mathbf{C}_{\mathbf{x}_n\mathbf{x}_n}^{-1}. \quad (117)$$

If $\mathbf{B}_n$ is known, the covariance is

$$\mathbf{R}_n^{i+1}=\frac{1}{K_n}\left\{\mathbf{C}_{\mathbf{z}_n\mathbf{z}_n}-\mathbf{C}_{\mathbf{z}_n\mathbf{x}_n}\mathbf{B}_n^{\mathrm{T}}-\mathbf{B}_n\mathbf{C}_{\mathbf{z}_n\mathbf{x}_n}^{\mathrm{T}}+\mathbf{B}_n\mathbf{C}_{\mathbf{x}_n\mathbf{x}_n}\mathbf{B}_n^{\mathrm{T}}\right\}. \quad (118)$$

If $\mathbf{B}_n$ is unknown, then

$$\mathbf{R}_n^{i+1} = \frac{1}{K_n}\left\{\mathbf{C}_{\mathbf{z}_n\mathbf{z}_n} - \mathbf{C}_{\mathbf{z}_n\mathbf{x}_n}\mathbf{C}_{\mathbf{x}_n\mathbf{x}_n}^{-1}\mathbf{C}_{\mathbf{z}_n\mathbf{x}_n}^{\mathrm{T}}\right\}. \tag{119}$$

The Q-function components are individually concave in the parameter sets $\{\mu_1, \mathbf{P}_1^{-1}\}$, $\{\mathbf{A}_n, \mathbf{Q}_n^{-1}\}$, and $\{\mathbf{B}_n, \mathbf{R}_n^{-1}\}$ so that the parameter updates at each iteration are the unique maxima of the CDLF. The final EM parameter estimates are guaranteed only to be critical points of the measurement likelihood, however. Suboptimal local maxima are, therefore, a possibility, so multiple training runs from different starting points may be needed to find the global maximum.

The parameter update formulas are specialized to time-invariant HGMMs by performing a second averaging operation across time when calculating the sample correlation matrices in (109), (110), (115), and (116) and then using these correlation matrices in (111), (113), (117), and (119), which are each evaluated only once for all time. The measurement likelihood for these time-invariant models has a parameter invariance structure that is important for convergence considerations. Consider the time-invariant parameter set $\mathbf{\Theta} = \{\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}, \mu_1, \mathbf{P}_1\}$. Let $\mathbf{U}_A$ be any nonsingular $L \times L$ matrix, and let the $L \times L$ matrix $\mathbf{U}_1$ be any nonsingular matrix that commutes with $\mathbf{A}$. As an argument of the measurement likelihood function in (64), $\mathbf{\Theta}$ is equivalent to any $\tilde{\mathbf{\Theta}} = \{\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{Q}}, \mathbf{R}, \tilde{\mu}_1, \tilde{\mathbf{P}}_1\}$, where

$$\tilde{\mathbf{A}} = \mathbf{U}_A\mathbf{A}\mathbf{U}_A^{-1} \tag{120}$$

$$\tilde{\mathbf{B}} = \mathbf{B}\mathbf{U}_1^{-1}\mathbf{U}_A^{-1} \tag{121}$$

$$\tilde{\mathbf{Q}} = \mathbf{U}_A\mathbf{U}_1\mathbf{Q}\mathbf{U}_1^{\mathrm{T}}\mathbf{U}_A^{\mathrm{T}} \tag{122}$$

$$\tilde{\mu}_1 = \mathbf{U}_A\mathbf{U}_1\mu_1 \tag{123}$$

$$\tilde{\mathbf{P}}_1 = \mathbf{U}_A\mathbf{U}_1\mathbf{P}_1\mathbf{U}_1^{\mathrm{T}}\mathbf{U}_A^{\mathrm{T}}. \tag{124}$$

The invariance of the likelihood to this family of parameter estimates is demonstrated in [47, App. C]. While the EM algorithm may converge to any member of this invariant family, depending on the initial values for the parameter estimates, any member of this invariant family is theoretically as good as any other for classification. It may be desirable for numerical reasons, however, to constrain the EM algorithm to produce estimates of a given structure (e.g., forcing the state-transition and covariance matrices to be as close as possible to identity matrices).

G. Mixed-Mode HGMMs

Paralleling the development of CS-HMMs, HGMMs are extended to include a Gaussian mixture for the prior density of the initial state. In these mixed-mode HGMMs (MM-HGMM's), the single-mode prior in (39) is replaced by the J-component mixture

$$p(\mathbf{x}_1|\theta_1, \rho) = \sum_{j=1}^{J} \rho_j \mathcal{N}(\mathbf{x}_1; \mu_{1j}, \mathbf{P}_{1j}) \tag{125}$$

where $\theta_1 = \{\mu_{11}, \ldots, \mu_{1j}, \mathbf{P}_{11}, \ldots, \mathbf{P}_{1j}\}$ contains the parameters for each mode in the mixture. This more general model is introduced to represent classes of signals whose members are all well modeled by the same set of system matrices $\{\mathbf{A}_n, \mathbf{B}_n, \mathbf{Q}_n, \mathbf{R}_n\}$ but exhibit significant within-class variability due to different initial-state values.

The forward densities for MM-HGMMs are obtained by substituting an indexed version of (62) into (27), giving

$$\alpha(\mathbf{x}_n) = \sum_{j=1}^{J} \rho_j\, c_n^j\, \mathcal{N}\left(\mathbf{x}_n; \mu_{n|n}^j, \mathbf{P}_{n|n}^j\right). \tag{126}$$

The c_n^j, $\mu_{n|n}^j$, and $\mathbf{P}_{n|n}^j$ are calculated for each j by using the HGMM recursions with the single-mode prior $p(\mathbf{x}_1|j)$. The measurement likelihood is

$$p(\mathbf{Z}_N) = \sum_{j=1}^{J} \rho_j\, p(\mathbf{Z}_N|j) = \sum_{j=1}^{J} \rho_j\, c_N^j. \tag{127}$$

Drawing on (29) and (89), the conditional state densities are

$$\gamma(\mathbf{x}_n) = \sum_{j=1}^{J} \rho_{j|N}\, \mathcal{N}\left(\mathbf{x}_n; \mu_{n|N}^j, \mathbf{P}_{n|N}^j\right) \tag{128}$$

where $\mu_{n|N}^j$ and $\mathbf{P}_{n|N}^j$ are the conditional state means and covariances from the appropriate single-mode HGMM. The conditional joint densities have a similar form.

Parameter estimates in MM-HGMMs are obtained using the results for mixed-mode CS-HMMs and the analysis techniques developed previously for HGMMs. The EM updates for the parameters in the initial-state mixture modes are

$$\mu_{1j}^{i+1} = \frac{1}{\kappa_j}\sum_{k=1}^{K} \rho_{j|N_k}\, \mu_{1|N_k}^{jk} \tag{129}$$

and

$$\mathbf{P}_{1j}^{i+1} = \frac{1}{\kappa_j}\sum_{k=1}^{K} \rho_{j|N_k}\left\{\mathbf{P}_{1|N_k}^{jk} + \varepsilon_1^{jk,i+1}\varepsilon_1^{jk,i+1^{\mathrm{T}}}\right\} \tag{130}$$

where $\varepsilon_1^{jk,i+1} = \mu_{1|N_k}^{jk} - \mu_{1j}^{i+1}$ and $\kappa_j = \sum_{k=1}^{K} \rho_{j|N_k}$. An estimator for ρ_j is given in (38). The system-matrix estimators are given in terms of correlation matrices as in (111), (113), (117), and (119). The correlation matrices for MM-HGMMs are similar to those for single-mode HGMMs but with a weighted sum over the mode assignments. That is, the correlation matrix estimators take the form

$$\mathbf{C}_{\mathbf{uv}} = \sum_{j=1}^{J} \rho_{j|N_k}\, \mathbf{C}_{\mathbf{uv}}^j \tag{131}$$

where $\mathbf{C}_{\mathbf{x}_n\mathbf{x}_n}^j$, $\mathbf{C}_{\mathbf{x}_n\mathbf{x}_{n-1}}^j$, and $\mathbf{C}_{\mathbf{z}_n\mathbf{x}_n}^j$ are obtained using (109), (110), and (116) with $\mu_{n|N_k}^k$ and $\mathbf{P}_{n|N_k}^k$ indexed by j. The measurement correlation matrix is identical to (115) since the measurements do not depend on the mode index.

IV. Simulation Test

Classification using HGMMs is illustrated in this section for a simulated three-class problem. This simple test uses time-invariant two-mode MM-HGMMs as class generators. An ideal set of model parameters are chosen for each class, and the models generate measurement data in a "free running" synthesis mode. The ideal models for the generators all have the same dimensions ($L = 4$, $M = 2$, $J = 2$), system covariance matrices ($\mathbf{Q} = 0.1\mathbf{I}$, $\mathbf{R} = 0.1\mathbf{I}$), mixing parameters ($\rho_1 = 1/3$, $\rho_2 = 2/3$), and mixture-mode covariances ($\mathbf{P}_1^1 = \mathbf{P}_1^2 = 0.01\mathbf{I}$). The classes differ in the transition and output matrices $\mathbf{A}$ and $\mathbf{B}$ and the mixture-mode mean vectors

TABLE I
SIMULATION TEST MEASUREMENT LOG-LIKELIHOODS

	Ideal Model	Estimated Models		
		Class 1	Class 2	Class 3
Class 1 Measurements	−43.918	**−47.323**	−374.53	−346.23
	−37.090	**−41.599**	−251.77	−152.65
	−34.745	**−37.491**	−453.55	−319.65
	−62.330	**−62.454**	−156.88	−207.11
	−46.894	**−48.536**	−262.18	−360.14
Class 2 Measurements	−67.144	−835.21	**−70.302**	−765.57
	−58.085	−2476.1	**−60.384**	−376.37
	−68.835	−2203.5	**−70.146**	−656.61
	−62.910	−3508.3	**−65.668**	−397.94
	−45.963	−4360.4	**−46.500**	−782.65
Class 3 Measurements	−37.982	−698.63	−897.10	**−44.572**
	−34.104	−2763.8	−466.27	**−40.332**
	−71.692	−1764.8	−1029.2	**−77.412**
	−66.388	−3286.6	−244.29	**−71.005**
	−51.946	−3438.4	−563.91	**−62.491**

μ_1^1 and μ_1^2. The ideal values for these parameters are randomly selected for each class, subject to certain observability and reachability constraints.

Cross-validation testing is used to demonstrate classification. The ideal class generators are used to synthesize a training set with 20 measurement sequences per class and an independent test set with five sequences per class. Variable-length sequences, with lengths randomly chosen within the range 96 to 128 time points, are used in both the training and test sets.

All of the model parameters, including the ones whose values are equal across classes, are optimized using the EM algorithm. The initial estimates for this algorithm are randomly selected. No constraints are imposed on any of the parameters during model training. After training models for all classes, the likelihood of each test sequence is evaluated using every class model. The desire, of course, is for each test sequence to score highest under the model whose parameters were estimated using the corresponding training set. Table I reports the likelihoods of the test sequences for each class under each model. The block rows of the table correspond to the test sequences for each class. The first column contains the ideal log-likelihood value for each test sequence, which is obtained by analyzing the sequence using the class generator MM-HGMM with the ideal parameter values. The last three columns contain the log-likelihoods produced by the estimated models for each class. The likelihoods for the "winning" model are shown in bold. In all cases, the winning model corresponds to the correct class, giving perfect classification performance on the test set. In addition, the likelihood values produced by the winning model are only slightly below those produced by the ideal model. Tests run across different variance levels for the transition and output densities showed similar performance.

V. SUMMARY

This paper has two primary goals: one practical and one theoretical. The practical goal is the development of PDF models for feature classes that exhibit a temporal dependence (e.g., features computed from segments of time-series data) and whose time samples lie along a continuous trajectory through feature space (e.g., the instantaneous amplitude and frequency of a squeal or whistle). In support of this first goal, linear Gaussian models are examined in the context of continuous-state HMMs (CS-HMMs), providing a framework for using these models in classification. Methods are given for estimating the PDF model parameters and for evaluating measurement likelihoods, thus providing the tools needed to design practical classifiers. The models are also extended to include Gaussian mixtures for the *a priori* density of the initial state.

The theoretical goal is the unification of two widely used tool sets, namely HMMs, which developed predominantly in the statistics and speech recognition arenas, and Kalman filters, whose historical roots reside in the control and tracking literature. This unification is achieved by developing the general class of CS-HMMs and then showing that linear Gaussian models are a special case. Two versions of the fixed-interval Kalman smoother are shown to be implementations of the Baum and Viterbi algorithms for CS-HMMs with Gaussian model densities, and an existing EM algorithm for linear Gaussian models is shown to arise naturally within the CS-HMM framework. A theoretical byproduct of this work is a derivation of the Kalman filter and smoother recursions in terms of the Gaussian refactorization lemma. A practical byproduct is a new expression for the cross-covariance between time-adjacent states that is more compact and numerically simpler than previously available.

APPENDIX
VITERBI ALGORITHM FOR HGMMS

The maximum *a posteriori* (MAP) estimate of the state sequence for a CS-HMM is given by

$$\hat{\mathbf{X}}_N = \arg\max_{\mathbf{X}_N} p(\mathbf{X}_N|\mathbf{Z}_N) = \arg\max_{\mathbf{X}_N} p(\mathbf{X}_N, \mathbf{Z}_N). \quad (132)$$

The joint likelihood is factored as in (1). The Viterbi algorithm [6], [7] evaluates (132) using a two-pass dynamic programming algorithm. The forward pass propagates a function $\phi(\mathbf{x}_n)$, which, like the Baum forward pass, is initialized as $\phi(\mathbf{x}_1) = p(\mathbf{z}_1|\mathbf{x}_1)p(\mathbf{x}_1)$. The forward recursion is then defined for $n = 2, \ldots, N$ as

$$\phi(\mathbf{x}_n) = \max_{\mathbf{x}_{n-1}} \{p(\mathbf{z}_n|\mathbf{x}_n)\, p(\mathbf{x}_n|\mathbf{x}_{n-1})\, \phi(\mathbf{x}_{n-1})\}. \quad (133)$$

This expression is similar to the Baum forward recursion for $\alpha(\mathbf{x}_n)$, except that (133) maximizes over the previous state at each step, whereas the Baum recursion marginalizes the previous state. For HGMMs, the difference between marginalization and maximization is just a scale factor. Noting (62), the Viterbi forward density is therefore

$$\phi(\mathbf{x}_n) = \tilde{c}_n \mathcal{N}\left(\mathbf{x}_n;\, \mu_{n|n}, \mathbf{P}_{n|n}\right) \quad (134)$$

where $\mathbf{P}_{n|n}$ and $\mu_{n|n}$ are defined in (59) and (60), respectively, and are obtained using a Kalman filter.

The backward pass of the Viterbi algorithm is a backsubstitution operation defined by

$$\hat{\mathbf{x}}_{n-1} = \arg\max_{\mathbf{x}_{n-1}} \{p(\mathbf{z}_n|\hat{\mathbf{x}}_n)\, p(\hat{\mathbf{x}}_n|\mathbf{x}_{n-1})\, \phi(\mathbf{x}_{n-1})\} \quad (135)$$

which is initialized at t_N as $\hat{\mathbf{x}}_N = \arg\max \phi(\mathbf{x}_N)$. Since the constants $p(\mathbf{z}_n|\hat{\mathbf{x}}_n)$ and $\tilde{c}_n$ do not influence the $\arg\max$ operation, the backward recursion for HGMMs is

$$\hat{\mathbf{x}}_{n-1} = \arg\max_{\mathbf{x}_{n-1}} \left\{\mathcal{N}\left(\hat{\mathbf{x}}_n;\, \mathbf{A}_n\mathbf{x}_{n-1},\, \mathbf{Q}_n\right) \times \mathcal{N}\left(\mathbf{x}_{n-1};\, \mu_{n-1|n-1},\, \mathbf{P}_{n-1|n-1}\right)\right\}. \quad (136)$$

Applying the GRL gives

$$\hat{\mathbf{x}}_{n-1} = \arg\max_{\mathbf{x}_{n-1}} \left\{\mathcal{N}\left(\mathbf{x}_{n-1};\, \mu_{n-1|N},\, \mathbf{\Lambda}_n\right)\right\} \quad (137)$$

where

$$\mu_{n-1|N} = (\mathbf{I} - \mathbf{H}_n\mathbf{A}_n)\, \mu_{n-1|n-1} + \mathbf{H}_n\, \hat{\mathbf{x}}_n \quad (138)$$

and where $\mathbf{\Lambda}_n$ and $\mathbf{H}_n$ are defined in (49) and (52), respectively. The maximum of a Gaussian density occurs at its mean; therefore, the Viterbi estimate at time t_{n-1} is

$$\hat{\mathbf{x}}_{n-1} = \mu_{n-1|n-1} + \mathbf{H}_n\left(\hat{\mathbf{x}}_n - \mu_{n|n-1}\right) \quad (139)$$

where $\mu_{n|n-1} = \mathbf{A}_n\, \mu_{n-1|n-1}$. This recursion is initialized with $\hat{\mathbf{x}}_N = \mu_{N|N}$, which is obtained from $\phi(\mathbf{x}_N)$ by inspection.

The Viterbi estimate is identical to the smoothed state estimate from the RTS formulation of the fixed-interval Kalman smoother [42]. For HGMMs, the means of the Baum densities are therefore the same as the Viterbi track estimates. Matrix $\mathbf{\Lambda}_n$ is *not* the error covariance for the state estimate, however. The Viterbi algorithm does not, in general, provide covariance estimates. An alternate derivation of the RTS algorithm, including the covariance matrices, is obtained in [47] by marginalizing the joint state density $\gamma(\mathbf{x}_n,\, \mathbf{x}_{n-1})$.

ACKNOWLEDGMENT

P. L. Ainsleigh is grateful to T. Luginbuhl and M. Graham of NUWC for enlightening technical discussions and comments on the manuscript and to S. Greineder and P. Baggenstoss, also of NUWC, for providing the perspective that motivated this research.

REFERENCES

[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufman, 1988.

[2] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Stat.*, vol. 37, pp. 1554–1563, 1966.

[3] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology," *Bull. Amer. Meteorol. Soc.*, vol. 73, pp. 360–363, 1967.

[4] 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.

[5] L. E. Baum, "An inequality and associate maximization technique in statistical estimation for probabilistic functions of a Markov process," *Inequal.*, vol. 3, pp. 1–8, 1972.

[6] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotic optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

[7] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood estimation from incomplete data," *J. R. Stat. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[9] W. J. Heiser, "Convergent computation by iterative majorization: Theory and applications in multidimensional data analysis," in *Recent Advances in Descriptive Multivariate Analysis*, W. J. Krzanowski, Ed. Oxford, U.K.: Clarendon, 1995.

[10] J. D. Ferguson, "Hidden Markov analysis: An introduction," in *Proc. IDA-CDR Symp. Appl. Hidden Markov Models Text and Speech*, J. D. Ferguson, Ed. Princeton, NJ, Oct. 1980.

[11] 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. Tech. J.*, vol. 62, no. 4, pp. 1035–1074, Apr. 1983.

[12] L. A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 729–734, Sept. 1982.

[13] B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of Markov chains," *IEEE Trans. Inform. Theory*, vol. 32, pp. 307–309, Mar. 1986.

[14] A. B. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, May 1982, pp. 1291–1294.

[15] B. H. Juang and L. R. Rabiner, "Mixture autoregressive hidden Markov models for speech signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1404–1413, Dec. 1985.

[16] P. Kenny, M. Lennig, and P. Mermelstein, "A linear predictive HMM for vector-valued observations with applications to speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 220–225, Feb. 1990.

[17] L. Deng, M. Aksmanovic, X. Sun, and C. F. J. Wu, "Speech recognition using hidden Markov models with polynomial regression functions as nonstationary states," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 507–520, Oct. 1994.

[18] V. Digalakis, J. R. Rohlicek, and M. Ostendorf, "ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 431–442, Oct. 1993.

[19] J. D. Ferguson, "Variable duration models for speech," in *Proc. IDA-CDR Symp. Appl. Hidden Markov Models Text Speech*, J. D. Ferguson, Ed. Princeton, NJ, Oct. 1980.

[20] M. Ostendorf, V. V. Digalakis, and O. A. Kimball, "From HMMs to segment models: A unified view of stochastic modeling for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 360–378, Sept. 1996.

[21] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York: Academic, 1988.

[22] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*. New York: Hemisphere, 1975.

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

[24] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Englewood Cliffs, NJ: Prentice-Hall, 2000.

[25] R. L. Kashyap, "Maximum likelihood identification of stochastic linear systems," *IEEE Trans. Automat. Contr.*, vol. AC-15, pp. 25–34, Feb. 1970.

[26] N. K. Gupta and R. K. Mehra, "Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations," *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 774–783, Dec. 1974.

[27] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *J. Time Series Anal.*, vol. 3, no. 4, pp. 253–264, 1982.

[28] M. W. Watson and R. F. Engle, "Alternative algorithms for the estimation of dynamic factor, mimic and varying coefficient regression models," *J. Econometr.*, vol. 23, pp. 385–400, 1983.

[29] I. Ziskind and D. Hertz, "Maximum likelihood localization of narrow-band autoregressive sources via the EM algorithm," *IEEE Trans. Signal Processing*, vol. 41, pp. 2719–2724, Aug. 1993.

[30] E. Weinstein, A. V. Oppenheim, M. Feder, and J. R. Buck, "Iterative and sequential algorithms for multisensor signal enhancement," *IEEE Trans. Signal Processing*, vol. 42, pp. 846–859, Apr. 1994.

[31] L. Deng and X. Shen, "Maximum likelihood in statistical estimation of dynamic systems: Decomposition algorithm and simulation results," *Signal Process.*, vol. 57, pp. 65–79, 1997.

[32] R. J. Elliot and V. Krishnamurthy, "New finite-dimensional filters for parameter estimation of discrete-time linear Gaussian models," *IEEE Trans. Automat. Contr.*, vol. 44, pp. 938–951, May 1999.
[33] R. L. Streit, "The relationship between Kalman filters and infinite-state hidden Markov models," Naval Undersea Warfare Cent., Newport, RI, NUWC Tech. Memo. 921 088, 1992.
[34] B. Delyon, "Remarks on linear and nonlinear filtering," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 317–322, Jan. 1995.
[35] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*. Chichester, U.K.: Wiley, 1990.
[36] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inform. Theory*, vol. 46, pp. 325–343, Mar. 2000.
[37] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum–product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.
[38] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic, 1970.
[39] E. Parzen, Texas A&M University, College Station, TX, personal communication, Nov. 2000.
[40] D. Q. Mayne, "A solution to the smoothing problem for linear dynamical systems," *Automatica*, vol. 4, pp. 73–92, 1966.
[41] D. C. Fraser and J. E. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Trans. Automat. Contr.*, vol. AC-7, pp. 387–390, Aug. 1969.
[42] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AAIA J.*, vol. 3, pp. 1445–1450, 1965.
[43] B. C. Levy, A. Benveniste, and R. Nikoukhah, "High-level primitives for recursive maximum likelihood estimation," *IEEE Trans. Automat. Contr.*, vol. 41, pp. 1125–1145, Aug. 1996.
[44] F. C. Schweppe, "Evaluation of likelihood functions for Gaussian signals," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 61–70, 1965.
[45] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, pp. 465–479, 1971.
[46] J. T.-H. Lo, "Finite-dimensional sensor orbits and optimal nonlinear filtering," *IEEE Trans. Inform. Theory*, vol. 18, pp. 583–588, Sept. 1972.
[47] P. L. Ainsleigh, "Theory of continuous-state hidden Markov models and hidden Gauss–Markov models," Naval Undersea Warfare Cent., Newport, RI, NUWC Tech. Rep. 11 274, Mar. 2001.
[48] ——, "Segmented chirp features and hidden Gauss–Markov nodels for signal classification," Ph.D. dissertation, Texas A&M Univ., College Station, TX, May 2001.
[49] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 716–723, 1974.
[50] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.
[51] G. Schwartz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, pp. 461–464, 1978.
[52] F. A. Graybill, *Matrices With Applications in Statistics*. Pacific Grove, CA: Wadsworth, 1983.
[53] A. Graham, *Kronecker Products and Matrix Calculus With Applications*. New York: Wiley, 1981.

Phillip L. Ainsleigh (M'92) received the B.S. and M.S. degrees from the University of South Florida, Tampa, in 1988 and the Ph.D. degree from Texas A&M University, College Station, in 2001, all in electrical engineering.

From 1988 to 1996, he worked with the Navy's Underwater Sound Reference Detachment, Orlando, FL, where he devised signal-modeling algorithms for acoustic measurements in water-filled tanks. Since 1997, he has been with the Naval Undersea Warfare Center, Newport, RI, where he has investigated methods for automatic signal classification. His interests include all facets of statistical signal processing, signal modeling, and time-frequency analysis.

Nasser Kehtarnavaz (S'82–M'86–SM'92) received the Ph.D. degree from Rice University, Houston, TX, in 1987.

He is currently a Professor in the Department of Electrical Engineering at the University of Texas at Dallas, Richardson. He was previously a Professor at Texas A&M University, College Station. His research interests include digital signal processing, image processing, pattern recognition, and medical image analysis. He is currently serving as Editor-of-Chief of the *Journal of Real-Time Imaging*

Dr. Kehtarnavaz is currently an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.

Roy L. Streit (M'83–SM'84) received the Ph.D. degree in mathematics from the University of Rhode Island, Kingston, in 1978.

He has been with the Naval Undersea Warfare Center, Newport, RI, since 1970, when he joined one of its predecessor organizations, the Navy Underwater Sound Laboratory, New London, CT. Past appointments include Visiting Scholar with the Department of Operations Research, Stanford University, Stanford, CA, from 1981 to 1982, Visiting Scientist with the Computer Science Department, Yale University, New Haven, CT, from 1982 to 1984, and Exchange Scientist with the Defense Science and Technology Organization, Adelaide, Australia from 1987 to 1989. His current research interests include passive detection and localization of distributed targets in hyperspectral images, tracking in random media using low-fidelity wave propagation models, numerical acoustic hull array design, and acoustic aberration.

Dr. Streit received the 1999 Solberg Award from the American Society of Naval Engineers for achievement in research and development related to naval engineering.