Recursive Expectation-Maximization (EM) Algorithms for Time-Varying Parameters with Applications to Multiple Target Tracking

Liron Frenkel and Meir Feder, *Fellow, IEEE*

Abstract— **We investigate the application of expectation-maximization (EM) algorithms to the classical problem of multiple target tracking (MTT) for a known number of targets. Conventional algorithms, which deal with this problem, have a computational complexity that depends exponentially on the number of targets, and usually divide the problem into a localization stage and a tracking stage. The new algorithms achieve a linear dependency and integrate these two stages. Three optimization criteria are proposed, using deterministic and stochastic dynamic models for the targets.**

I. Introduction

The PROBLEM of tracking superimposed signals embedded in noise is important in sonar, radar, spectral estimation, and other fields. The observed data $\boldsymbol{y}_1(t)$, $\boldsymbol{y}_2(t)$, $\cdots$, $\boldsymbol{y}_k(t)$, $\cdots$ can be a nonlinear noisy function of the trajectories' parameters $\theta_1^{(n)}$, $\theta_2^{(n)}$, $\cdots$, $\theta_k^{(n)}$, $\cdots$

$$\boldsymbol{y}_k(t) = \sum_{n=1}^{N} \boldsymbol{s}_k^{(n)}\left(t, \theta_k^{(n)}\right) + \boldsymbol{n}_k(t) \qquad 0 < t < T \quad (1)$$

where

k time interval index;

$\boldsymbol{y}_k(t)$ vector of M elements composed of N superimposed signals $\boldsymbol{s}_k^{(n)}(t, \theta_k^{(n)})$;

$\boldsymbol{n}_k(t)$ observation noise.

In an active sonar problem, for example, a signal is sent each time interval k, and its reflections from the N targets are received by an array of M sensors. $\boldsymbol{y}_k(t)$ holds the received data, and each one of its elements corresponds to one sensor. The received data is a function of the parameters, which can be the locations and velocities of the N targets, for each time interval (Fig. 1).

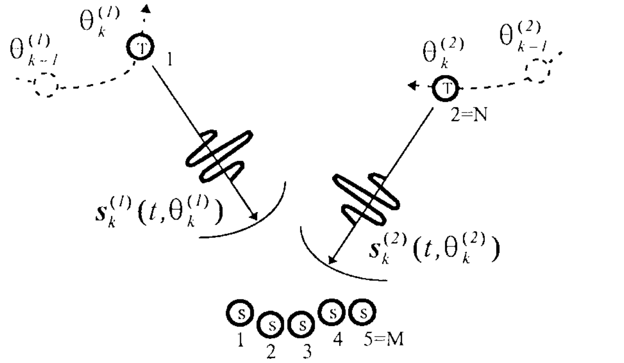


Fig. 1. Active sonar problem with five sensors ($M = 5$) and two targets ($N = 2$). At each time interval (k), the sonar sends a signal that is reflected from the moving targets and received by the sensors. The algorithm must estimate the targets' locations and velocities ($\theta_k^{(n)}$) from this data.

The parameters $\theta_k^{(n)}$ themselves can be modeled as a stochastic process or as a deterministic vector. The problem is to estimate the parameters $\theta_k^{(n)}$ given a model for parameters' dynamics. A direct attempt to solve the problem with an exact MAP or maximum likelihood estimation of these parameters will produce an algorithm with a computational complexity that is exponential with respect to the number of the snapshots and the targets. Consequently, the algorithms that are traditionally used for multiple target tracking (MTT) are not optimal in the sense that the problem is usually considered in two separate stages: *localization,* in which the new parameters $\theta_k^{(n)}$ are estimated from the recent snapshot $\boldsymbol{y}_k(t)$, and *tracking,* which makes use of some estimations from the localization part to produce the final track. Even the estimation in the localization part is usually not optimal (in a minimum estimation error sense) because optimality requires a computational complexity that depends exponentially on the number of the targets.

In this paper, we investigate the application of the EM algorithm to this classical problem of MTT with a known number of targets. The algorithms that are used integrate the localization stage with the tracking stage and achieve a linear computational complexity with respect to the targets number. The algorithms must be sequential in order to avoid accumulating data and calculations and to produce a continuous parameters' estimate.

The organization of the paper is as follows. In Section III, we define a *deterministic* dynamic model for the parameters with a "forgetting" mechanism that allows changes over time. As an example, we use straight lines to model the targets' trajectories. The parameters' estimation can be accomplished by a multiparameter maximum likelihood search. Using a

Manuscript received June 16, 1995; revised March 10, 1998. This work was supported by the U.S.–Israel Binational Science Foundation (BSF). The associate editor coordinating the review of this paper and approving it for publication was Dr. Petar M. Djuric.

L. Frenkel is with Orckit Communications, Tel-Aviv, Israel.

M. Feder is with the Department of Electrical Engineering—Systems, Tel-Aviv University, Tel-Aviv, Israel.

Publisher Item Identifier S 1053-587X(99)00763-1.

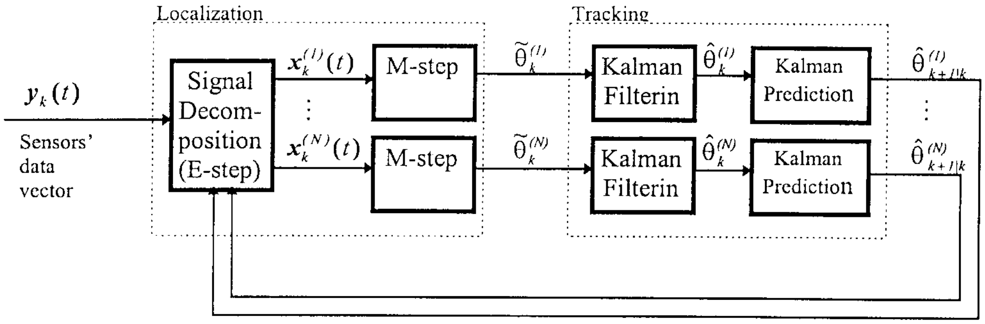


Fig. 2. EM–Kalman algorithm.

computationally efficient recursive EM algorithm, the search over targets and time is avoided. Applying, further, a Newton approximation, we obtain a new, fast stochastic algorithm that benefits from the given model.

In Section IV, we describe a second approach that uses a Bayesian model with a *MSE estimation* criterion. The parameters are modeled as a stochastic process, and the observations are a nonlinear noisy function of this process. In this case, complete integration of the localization and the tracking is not possible. We use the EM algorithm for localization and Kalman filtering for tracking with feedback to the localization stage.

In Section V, we take a third approach, which uses a Bayesian model with *MAP estimation* criterion. The parameters' values $\theta_k^{(n)}$ are taken from a finite set of states (for example, a finite grid of locations). The process $\theta_1^{(n)}$, $\theta_2^{(n)}, \cdots, \theta_k^{(n)}, \cdots$ is modeled as a discrete Markov chain, and the probabilistic description of the process is assumed to be known. This process is not observed directly (it is hidden). The algorithm that estimates the parameters from the observed data is described in [5]. In [6] and [7], this algorithm is applied to tracking time-varying tones embedded in noise. We shall refer to this algorithm as the *hidden Markov model* (HMM) algorithm. The HMM algorithm does not reduce the exponential dependence of the computational complexity on the targets' number. However, integrating the EM algorithm with the HMM algorithm eliminates this problem. In the next section, we briefly review the EM algorithm (see Figs. 2 and 3). For the derivation of the EM algorithm, see [10].

II. The EM Algorithm

Given a conditional probability density $f_{\boldsymbol{Y}}(\boldsymbol{y};\, \theta)$, the maximum likelihood (ML) estimation of the parameters vector θ from the observed data $\boldsymbol{Y}$ is

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \log f_{\boldsymbol{Y}}(\boldsymbol{y};\, \theta).$$

If the vector θ contains several unknowns [such as locations and velocities of multiple targets at different snapshots in our model (1)], this maximization tends to be very complex.

One of the reasons why it is difficult to maximize the likelihood function is that the observed data $\boldsymbol{Y}$ is "incomplete." Let $\boldsymbol{X}$ denote a "complete data." For our purposes, the complete data is any vector $\boldsymbol{X}$ such that

$$\boldsymbol{Y} = \boldsymbol{H}(\boldsymbol{X})$$

where $\boldsymbol{H}(\cdot)$ is a noninvertable (many-to-one) transformation. This relation means that we observe the complete data only through a noninvertable transformation, where this transformation generally causes a reduction in the available data. In the superimposed signals problem, the incomplete data $\boldsymbol{Y}$ is sum of the signals in (1), and the complete data $\boldsymbol{X}$ can be the observation of each signal $\boldsymbol{s}_k^{(n)}(t,\, \theta_k^{(n)})$ separately.

The EM algorithm starts with an arbitrary initial guess $\hat{\theta}^{(0)}$ (hereafter denoted $\hat{\theta}^{(p)}$, which is the current estimate of θ after p iterations of the algorithm). The subsequent iteration cycle can then be described in two steps as

E step:

$$\text{compute } Q\Big(\theta,\, \hat{\theta}^{(p)}\Big) = E\Big\{\log f_{\boldsymbol{X}}(\boldsymbol{x};\, \theta) | \boldsymbol{Y} = \boldsymbol{y};\, \hat{\theta}^{(p)}\Big\} \quad (2)$$

M step:

$$\underset{\theta}{\textbf{ArgMax}}\, Q\Big(\theta,\, \hat{\theta}^{(p)}\Big) \longrightarrow \hat{\theta}^{(p+1)} \quad (3)$$

where $f_{\boldsymbol{X}}(\boldsymbol{x};\, \theta)$ is the probability density of the "complete data," which is a function of θ, and $E\{\cdot;\, \hat{\theta}^{(p)}\}$ is a conditional expectation given $\hat{\theta}^{(p)}$.

If $Q(\theta,\, \theta')$ is continuous in both θ and θ', the algorithm converges to a stationary point of the log-likelihood function where the maximization in (3) ensures that each iteration cycle increases the likelihood until a local maximum is reached.

We utilize those previous results to develop the algorithms in the sequel. Three major approaches are proposed.

III. Maximum Likelihood Estimation and the EM–Newton Algorithm

The first approach uses a deterministic model for the parameters. A "forgetting" mechanism is used to allow changes over time. First, we derive a general algorithm under the assumption that the data is characterized by a probability density that is a function of the parameters. We then apply this algorithm to the special case of superimposed signals. We end the section with a detailed example of time delayed signals.

A. A General Algorithm

Suppose $\boldsymbol{y}_1(t),\, \boldsymbol{y}_2(t), \cdots, \boldsymbol{y}_k(t), \cdots, \boldsymbol{y}_K(t)$ are independent snapshots (the incomplete data), each with a probability density $f_{\boldsymbol{Y}}(\boldsymbol{y}_k,\, \theta_k)$. The parameters θ_k are unknown vectors that vary according to

$$\theta_k = \boldsymbol{F}\theta_{k-1} \tag{4}$$

where $\boldsymbol{F}$ is a constant invertible transition matrix. According to this model, estimating the last parameter θ_K will automatically derive the estimation of all the preceding parameters. Consequently, the parameters estimation can be accomplished by a multiparameter *maximum likelihood* search on θ_K. Using a computationally efficient recursive EM algorithm, the search over the targets is avoided. Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_k, \cdots$ represent an independent complete data that is related to the observations by

$$\boldsymbol{H}\boldsymbol{x}_k = \boldsymbol{y}_k. \tag{5}$$

The EM algorithm starts with an arbitrary initial guess $\theta_K^{(0)}$ and, after l iterations, estimates θ_K by $\hat{\theta}_K^{(l)}$. Each iteration cycle can be described as having the following two steps:

E: Evaluate

$$\begin{aligned} Q\left(\theta_K, \hat{\theta}_K^{(l)}\right) &= E\left\{\log f(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K;\, \theta_1, \cdots, \theta_K)\right| \\ &\qquad \left.\boldsymbol{y}_1, \cdots, \boldsymbol{y}_K;\, \hat{\theta}_K^{(l)}\right\} \\ &= \sum_{i=1}^{K} E\left\{\log f(\boldsymbol{x}_i, \mathbf{F}^{i-K}\theta_K)\right| \\ &\qquad \left.\boldsymbol{y}_1, \cdots, \boldsymbol{y}_K;\, \hat{\theta}_K^{(l)}\right\} \\ &= \sum_{i=1}^{K} E\left\{\log f(\boldsymbol{x}_i, \mathbf{F}^{i-K}\theta_K)\middle|\, \boldsymbol{y}_i;\, \hat{\theta}_K^{(l)}\right\} \end{aligned}$$

M:

$$\operatorname*{Max}_{\theta_K} Q\left(\theta_K, \hat{\theta}_K^{(l)}\right) \rightarrow \hat{\theta}_K^{(l+1)}$$

where $E\{\cdot;\, \hat{\theta}_k^{(l)}\}$ denotes a statistical expectation with respect to the last parameter estimation. In each iteration, all the data has to be processed. Our goal is, however, to obtain a recursive (sequential) method. We follow the sequential algorithm proposed by Titterington for constant parameters [2], and the algorithm becomes the following.

The recursive EM algorithm for deterministicaly varying parameters:

E:

$$\begin{aligned} &Q(\theta_{k+1}, \hat{\theta}_k) \\ &\quad = L_{k+1}(\theta_{k+1}) \\ &\quad = \sum_{l=1}^{k+1} \gamma^{k+1-l} \\ &\qquad \cdot E\{\log\, f(\boldsymbol{x}_l, \boldsymbol{F}^{l-k-1}\theta_{k+1}) |\, \boldsymbol{y}_l;\, F\hat{\theta}_{l-1}\} \\ &\quad = \gamma L_k(\boldsymbol{F}^{-1}\theta_{k+1}) \\ &\qquad + E\{\log f(\boldsymbol{x}_{k+1}, \theta_{k+1}) |\, \boldsymbol{y}_{k+1};\, F\hat{\theta}_{l-1}\} \end{aligned} \tag{6}$$

M:

$$\operatorname*{Max}_{\theta_{k+1}} Q(\theta_{k+1}, \hat{\theta}_k) \rightarrow \hat{\theta}_{k+1} \tag{7}$$

where each of the statistical expectations of the log-likelihood is done once and used for the next expectation. The constant γ was suggested by Weinstein and Feder [1]. For varying parameters, γ is expected to be a tradeoff between good tracking ability (γ small) and noise insensitivity ($\gamma = 1$). This algorithm is still suffers from complexity drawbacks. In Appendix A, we will show that the Newton second-order approximation of the recursive EM algorithm (6) and (7) is as follows.

The recursive EM-Newton algorithm:

$$\hat{\theta}_{k+1} = \boldsymbol{F}\hat{\theta}_k + \boldsymbol{Ic}_{k+1}^{-1}\boldsymbol{S}(\boldsymbol{y}_{k+1}, F\hat{\theta}_k) \tag{8}$$

$$\hat{\boldsymbol{I}}\boldsymbol{c}_{k+1} = \gamma\boldsymbol{F}^{-1T}\hat{\boldsymbol{I}}\boldsymbol{c}_k\boldsymbol{F}^{-1} + \boldsymbol{Ic}(\boldsymbol{F}\hat{\theta}_k) \tag{9}$$

where $S(\boldsymbol{y}_k, \theta_k)$ denotes the score vector, and $\boldsymbol{Ic}(\theta_k)$ is a Fisher information matrix corresponding to a single complete snapshot, that is

$$\boldsymbol{S}(\boldsymbol{y}_k, \theta_k) = \nabla_{\theta_k} \log f_{\boldsymbol{Y}}(\boldsymbol{y}_k;\, \theta_k) \tag{10}$$

$$\boldsymbol{Ic}(\theta_k) = -E\{\nabla_{\theta_k}^2 \log f_{\boldsymbol{X}}(\boldsymbol{x}_k;\, \theta_k) |\, \theta_k\}. \tag{11}$$

Notice that for $\boldsymbol{F} = \boldsymbol{\mathit{Identity}}$ (constant parameters) and for $\gamma = 1$ (no forgetting factor), this algorithm becomes

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \left[\sum_{i=1}^{k} \boldsymbol{Ic}(\hat{\theta}_i)\right]^{-1} \boldsymbol{S}(\boldsymbol{y}_{k+1}, \hat{\theta}_k)$$

and for a small change of $\boldsymbol{Ic}(\hat{\theta}_i)$ near the ML estimation θ, we obtain approximately

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \left[k\boldsymbol{Ic}(\hat{\theta}_k)\right]^{-1} \boldsymbol{S}(\boldsymbol{y}_{k+1}, \hat{\theta}_k)$$

which is a stochastic approximation algorithm suggested by Titterington [2] for constant parameters. The conventional approach to handle time-varying parameters (see [9]) is to substitute the converging series $(1/k)$ with a small positive constant γ_0 in the algorithm obtained for constant parameters, that is

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \gamma_0\left[\boldsymbol{Ic}(\hat{\theta}_k)\right]^{-1} \boldsymbol{S}(\boldsymbol{y}_{k+1}, \hat{\theta}_k). \tag{12}$$

This *ad hoc* procedure can now be replaced by the new EM–Newton algorithm in order to utilize a dynamic model for the parameters.

B. Superimposed Signals Tracking

Consider the superimposed signals model (1), and let $\boldsymbol{n}(t)$ be a zero-mean white Gaussian process whose covariance matrix is $E\{\boldsymbol{n}(t)\boldsymbol{n}(\tau)^*\} = \boldsymbol{R}\delta(t - \tau)$.

The log-likelihood function of the incomplete data $\boldsymbol{y}_k(t)$ is

$$\begin{aligned}\log f_{\boldsymbol{Y}}(\boldsymbol{y}_k;\, \theta_k) = c - \int_T &\left[\boldsymbol{y}_k(t) - \sum_{n=1}^{N} \boldsymbol{s}_k^{(n)}\left(t,\, \theta_k^{(n)}\right)\right]^* \\ &\cdot \boldsymbol{R}^{-1}\left[\boldsymbol{y}_k(t) - \sum_{n=1}^{N} \boldsymbol{s}_k^{(n)}\left(t,\, \theta_k^{(n)}\right)\right] dt\end{aligned} \tag{13}$$

where c is independent of θ. The complete data can be obtained by decomposing the observed data $\boldsymbol{y}_k(t)$ into its signal components (see [3]), that is

$$\mathbf{x}_k(t) = \begin{bmatrix}\mathbf{x}_k^{(1)}(t) \\ \mathbf{x}_k^{(2)}(t) \\ \vdots \\ \mathbf{x}_k^{(N)}(t)\end{bmatrix} \tag{14}$$

where

$$\boldsymbol{x}_k^{(n)}(t) = \boldsymbol{s}_k^{(n)}\left(t,\, \theta_k^{(n)}\right) + \boldsymbol{n}^{(n)}(t).$$

The total noise was decomposed to N components

$$\sum_{n=1}^{N} \boldsymbol{n}^{(n)}(t) = \boldsymbol{n}(t)$$

with covariance matrices $E\{\boldsymbol{n}_k^{(n)}(t)^*\boldsymbol{n}_k^{(n)}(\tau)\} = \boldsymbol{R}^{(n)}\delta(t-\tau)$, where $\boldsymbol{R}^{(n)} = \beta^{(n)}\boldsymbol{R}$. The $\beta^{(n)}$'s are arbitrary positive scalars satisfying

$$\sum_{n=1}^{N} \beta^{(n)} = 1$$

and, consequently

$$\boldsymbol{y}_k(t) = \sum_{n=1}^{N} \boldsymbol{x}_k^{(n)}(t) = \boldsymbol{H}\boldsymbol{x}_k(t)$$

where

$$\boldsymbol{H} = [\boldsymbol{I}\ \ \boldsymbol{I}\ \ \cdots\ \ \boldsymbol{I}].$$

The log-likelihood function of the complete data $\boldsymbol{x}(t)$ is

$$\begin{aligned}\log f_{\mathbf{X}}(\mathbf{x}_k;\, \theta_k) &= c - \int_T [\mathbf{x}_k(t) - \mathbf{s}_k(t, \theta_k)]^* \\ &\qquad \cdot \Lambda^{-1}[\mathbf{x}_k(t) - \mathbf{s}_k(t,\, \theta_k)]\, dt \\ &= c - \sum_{n=1}^{N} \int_T \left[\boldsymbol{x}_k^{(n)}(t) - \boldsymbol{s}_k^{(n)}\left(t,\, \theta_k^{(n)}\right)\right]^* \\ &\qquad \cdot \boldsymbol{R}^{(n)^{-1}}\left[\boldsymbol{x}_k^{(n)}(t) - \boldsymbol{s}_k^{(n)}\left(t,\, \theta_k^{(n)}\right)\right] dt\end{aligned} \tag{15}$$

where

$$\theta_k = \begin{bmatrix}\theta_k^{(1)} \\ \theta_k^{(2)} \\ \vdots \\ \theta_k^{(N)}\end{bmatrix}, \qquad \boldsymbol{s}_k(t,\, \theta_k) = \begin{bmatrix}\boldsymbol{s}_k^{(1)}\left(t,\, \theta_k^{(1)}\right) \\ \boldsymbol{s}_k^{(2)}\left(t,\, \theta_k^{(2)}\right) \\ \vdots \\ \boldsymbol{s}_k^{(n)}\left(t,\, \theta_k^{(N)}\right)\end{bmatrix}$$

$$\Lambda = \begin{bmatrix}\boldsymbol{R}^{(1)} & & & 0 \\ & \boldsymbol{R}^{(2)} & & \\ & & \ddots & \\ 0 & & & \boldsymbol{R}^{(N)}\end{bmatrix}. \tag{16}$$

The score is

$$\begin{aligned}\boldsymbol{S}(\boldsymbol{y}_k,\, \theta_k) &= \nabla_{\theta_k} \log f_y(\boldsymbol{y}_k;\, \theta_k) \\ &= 2\int_T \text{Real}\left\{\left[\boldsymbol{y}_k(t) - \sum_{n=1}^{N} \boldsymbol{s}_k^{(n)}\left(t,\, \theta_k^{(n)}\right)\right]^* \right. \\ &\qquad \left. \cdot \boldsymbol{R}^{-1}\nabla_{\theta_k}\left[\sum_{n=1}^{N} \boldsymbol{s}_k^{(n)}\left(t,\, \theta_k^{(n)}\right)\right]\right\} dt \\ &= 2\int_T \text{Real}\left\{\left[\boldsymbol{y}_k(t) - \sum_{n=1}^{N} \boldsymbol{s}_k^{(n)}\left(t,\, \theta_k^{(n)}\right)\right]^* \right. \\ &\qquad \cdot \boldsymbol{R}^{-1}\left[\nabla_{\theta_k^{(1)}}\boldsymbol{s}_k^{(1)}\left(t,\, \theta_k^{(1)}\right),\, \cdots \right. \\ &\qquad\qquad \left.\left. \nabla_{\theta_k^{(N)}}\boldsymbol{s}_k^{(N)}\left(t,\, \theta_k^{(N)}\right)\right]\right\} dt.\end{aligned}$$

The Fisher information matrix of the complete data is

$$\begin{aligned}\boldsymbol{Ic}(\theta_k) &= -E\{\nabla_{\theta_k}^2 \log f_x(\boldsymbol{x}_k;\, \theta_k) |\, \theta_k\} \\ &= E\left\{\nabla_{\theta_k} \text{Real}\left[\int_T [\mathbf{x}_k(t) - \mathbf{s}(t, \theta_k)]^* \right.\right. \\ &\qquad \left.\left. \cdot\ \Lambda^{-1}[-\nabla_{\theta_k}\mathbf{s}(t,\, \theta_k)]\right] \middle|\, \theta_k\right\} \\ &= \text{Real}\left\{\int_T \nabla_{\theta_k}\boldsymbol{s}(t,\, \theta_k)^*\Lambda^{-1}\nabla_{\theta_k}\boldsymbol{s}(t,\, \theta_k)\right\}\end{aligned}$$

and using (16), we obtain

$$\begin{aligned}\boldsymbol{Ic}(\theta_k) &= \text{Real}\left\{\int_T \sum_{n=1}^{N} \left[\nabla_{\theta_k}\boldsymbol{s}\left(t,\, \theta_k^{(n)}\right)\right]^* \right. \\ &\qquad \left. \cdot\ \beta^{(n)^{-1}}\boldsymbol{R}^{-1}\left[\nabla_{\theta_k}\boldsymbol{s}\left(t,\, \theta_k^{(n)}\right)\right] dt\right\} \\ &= \sum_{n=1}^{N} \boldsymbol{Ic}^{(n)}(\theta_k)\end{aligned}$$

where $\boldsymbol{Ic}^{(n)}(\theta_k)$ is the Fisher information matrix for one target:

$$\begin{aligned}\boldsymbol{Ic}^{(n)}(\theta_k) &= \text{Real}\left\{\int_T \left[\nabla_{\theta_k^{(n)}}\boldsymbol{s}\left(t,\, \theta_k^{(n)}\right)\right]^* \right. \\ &\qquad \left. \cdot\ \beta^{(n)^{-1}}\boldsymbol{Q}^{-1}\left[\nabla_{\theta_k^{(n)}}\boldsymbol{s}\left(t,\, \theta_k^{(n)}\right)\right] dt\right\}.\end{aligned}$$

Remark: One way of choosing $\beta^{(n)}$ is $\beta^{(n)} = 1/N$. However, different distributions can be used, defining different "complete" data.

C. Example: Line Array, Far-Field Targets with Constant Velocities

Suppose that the targets are moving approximately with constant radial and angular velocities. If the parameters are

$$\theta_k^{(n)} = \begin{bmatrix} \tau_k^{(n)} \\ \Delta\tau_k^{(n)} \\ \phi_k^{(n)} \\ \Delta\phi_k^{(n)} \end{bmatrix} \tag{17}$$

where $\tau_k^{(n)}$, $\Delta\tau_k^{(n)}$, $\phi_k^{(n)}$, and $\Delta\phi_k^{(n)}$ are the signal time delay, time-delay rate, angular position, and angular rate, then the dynamic model is

$$\theta_{k+1}^{(n)} = \boldsymbol{F}^{(n)}\theta_k^{(n)}, \quad \text{where} \quad \boldsymbol{F}^{(n)} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

and

$$\theta_{k+1} = \boldsymbol{F}\theta_k, \quad \text{where} \quad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}^{(1)} & & & 0 \\ & \boldsymbol{F}^{(2)} & & \\ & & \ddots & \\ 0 & & & \boldsymbol{F}^{(N)} \end{bmatrix}.$$

The baseband representation of a signal $\boldsymbol{s}(t, \theta_k^{(n)})$ that is received by a line array with targets at far-field and negligible Doppler effect is

$$\mathbf{s}_k^{(n)}\left(t, \theta_k^{(n)}\right) = \alpha_k^{(n)} s\left(t - \tau_k^{(n)}\right) \begin{bmatrix} e^{j2\pi\, d^{(1)}\, \cos(\phi)} \\ \vdots \\ e^{j2\pi\, d^{(M)}\, \cos(\phi)} \end{bmatrix}$$

where

$s(t)$	envelope of the radiated signal;
$d^{(m)}$	position of the m's sensor in carrier wave lengths;
$\alpha_k^{(n)}$	known complex amplitudes.

$s(t)$ is the baseband representation of the modulated signal

$$\tilde{\boldsymbol{s}}_k^{(n)}\left(t, \theta_k^{(n)}\right) = \mathbf{Real}\left\{ s_k^{(n)}\left(t - \tau_k^{(n)}\right) e^{-j2\pi \mathbf{F_c} t} \right\}$$

where F_c is the carrier frequency. Taking the derivatives of $\tau_k^{(n)}$, $\Delta\tau_k^{(n)}$, $\phi_k^{(n)}$, and $\Delta\phi_k^{(n)}$, we obtain

$$\begin{aligned} &\nabla_{\theta_k^{(n)}} \boldsymbol{s}_k^{(n)}\left(t, \theta_k^{(n)}\right) \\ &= \Bigg[-\alpha_k^{(n)} \dot{s}\left(t - \tau_k^{(n)}\right) \begin{bmatrix} e^{j2\pi\, d^{(1)}\, \cos(\phi)} \\ \vdots \\ e^{j2\pi\, d^{(M)}\, \cos(\phi)} \end{bmatrix} \\ &\qquad 0, \; -\alpha_k^{(n)} s\left(t - \tau_k^{(n)}\right) \\ &\qquad \cdot \begin{bmatrix} j2\pi\, d^{(1)}\, \sin(\phi) e^{j2\pi\, d^{(1)}\, \cos(\phi)} \\ \vdots \\ j2\pi\, d^{(M)}\, \sin(\phi) e^{j2\pi\, d^{(M)}\, \cos(\phi)} \end{bmatrix}, 0 \Bigg]. \end{aligned}$$

Using this expression, the score and the information matrix can be evaluated, and the EM–Newton algorithm can be used. Note that the information matrix is initially singular due to the fact that in this example, the radial and angular rates are not observed directly, and two observations are needed to gain information about the velocities.

A more careful look at this example reveals a weakness of the algorithm in tracking signals with rapidly changing phase. This problem is typical to stochastic approximation. Observe that

$$\alpha_k^{(n)} = \left|\alpha_k^{(n)}\right| e^{j \cdot \arg(\alpha_k^{(n)})} = \left|\alpha_k^{(n)}\right| e^{-j \cdot 2\pi(\tau_k^{(n)} F_c)}.$$

The phases $\arg(\alpha_k^{(n)})$ and energies $|\alpha_k^{(n)}|$ in the example above were assumed to be known constants. This is a good model for tracking over periods in which the energy and phases of the targets do not change considerably. However, in the typical sonar tracking case, the energies and phases do change. Regarding the energy tracking, $|\alpha_k^{(n)}|$ can be added to the parameters vectors $\theta_k^{(n)}$ with a dynamic model that will be useful to describe slow changes. The situation is usually different for the phase, which is a function of the time delay $\tau_k^{(n)}$. Taking the dependence of $\alpha_k^{(n)}$ on $\tau_k^{(n)}$ into account, the gradient becomes

$$\begin{aligned} &\nabla_{\theta_k^{(n)}} \boldsymbol{s}_k^{(n)}\left(t, \theta_k^{(n)}\right) \\ &= \Bigg[-\alpha_k^{(n)} \left(\dot{s}\left(t - \tau_k^{(n)}\right) + j2\pi F_c s\left(t - \tau_k^{(n)}\right) \right) \\ &\qquad \cdot \begin{bmatrix} e^{j2\pi\, d^{(1)}\, \cos(\phi)} \\ \vdots \\ e^{j2\pi\, d^{(M)}\, \cos(\phi)} \end{bmatrix}, \; 0, \; -\alpha_k^{(n)} s\left(t - \tau_k^{(n)}\right) \\ &\qquad \cdot \begin{bmatrix} j2\pi\, d^{(1)}\, \sin(\phi) e^{j2\pi\, d^{(1)}\, \cos(\phi)} \\ \vdots \\ j2\pi\, d^{(M)}\, \sin(\phi) e^{j2\pi\, d^{(M)}\, \cos(\phi)} \end{bmatrix}, 0 \Bigg]. \end{aligned}$$

For this model of $\alpha_k^{(n)}$, a small error in the estimation of the range can cause a large error in the phase and, consequently, an incorrect score. This is a result of an inherent property of the likelihood function (13) that, for modulated signals, has many modes. The large phase errors degrade the tracking performance of the stochastic approximation algorithm because the estimation of the range must be located at a smaller neighborhood of the true range in order to justify the approximation. The algorithm is, therefore, sensitive to estimation errors in $\tau_\kappa^{(\nu)}$ that are in the order of $1/F_c$ instead of errors that are in the order of the duration of the signal envelope (assuming a smooth envelope). Nevertheless, the algorithm can still be used when the deterministic model for the dynamics of the targets is good with respect to the SNR. The tracking performance is better than the one of a stochastic approximation algorithm without a dynamic model.

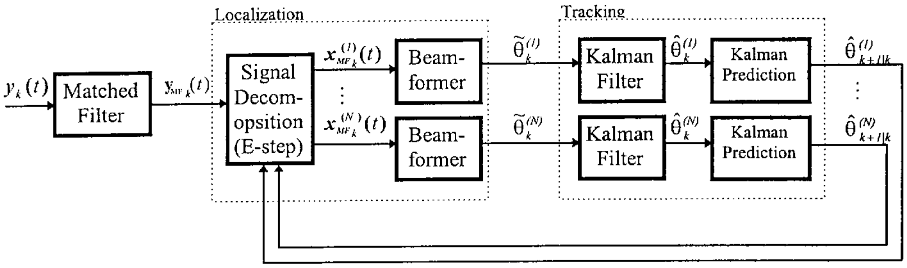


Fig. 3. EM–Kalman algorithm for time-delayed signals.

The algorithms that are described in the sequel are of a more global nature, and the range estimation does not suffer fatal errors due to phase errors.

IV. MSE Estimation and the EM-Kalman Algorithm

A natural approach to the problem of tracking superimposed signals is to model the dynamic parameters as a stochastic process. A reasonable choice would be

$$\theta_k^{(n)} = \boldsymbol{F}\theta_{k-1}^{(n)} + \boldsymbol{q}_k \tag{19}$$

where $\boldsymbol{q}_k$ is the driving noise of the parameters process, and $\theta_k^{(n)}$ is the parameters describing the nth signal the kth snapshot.

The data observed at snapshot k are

$$\boldsymbol{y}_k(t) = \sum_{n=1}^{N} \boldsymbol{s}_k^{(n)}\left(t,\, \tilde{\theta}_k^{(n)}\right) + \boldsymbol{n}(t). \tag{20}$$

Note that the parameter vector $\tilde{\theta}_k^{(n)}$, which affects the observed data $\boldsymbol{y}_k(t)$, can be shorter than the full parameter vector $\theta_k^{(n)}$. Define the relation $\tilde{\theta}_k^{(n)} = \boldsymbol{G}\theta_k^{(n)}$. For example, in the line array case (17), the velocity is not measured directly, and we can define

$$\tilde{\theta}_k^{(n)} = \begin{bmatrix} \tau_k^{(n)} \\ \phi_k^{(n)} \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \theta_k^{(n)} = \begin{bmatrix} \tau_k^{(n)} \\ \Delta\tau_k^{(n)} \\ \phi_k^{(n)} \\ \Delta\phi_k^{(n)} \end{bmatrix}.$$

Two new algorithms that estimate the parameters $\theta_k^{(n)}$ (the track) from the snapshots data $\boldsymbol{y}_k(t)$ using the Bayesian approach are proposed: one using the MAP estimation criterion (formulated in the next section) and the other using the MSE criterion.

The parameter vector that minimizes the MSE criterion is

$$\hat{\theta} = E\{\theta | \boldsymbol{y}_1 \cdots \boldsymbol{y}_k\}$$

where

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_K \end{bmatrix}, \qquad \theta_k = \begin{bmatrix} \theta_k^{(1)} \\ \theta_k^{(2)} \\ \vdots \\ \theta_k^{(N)} \end{bmatrix}. \tag{21}$$

This estimation can be accomplished by the Kalman filter if $\boldsymbol{s}_k^{(n)}(t,\, \theta_k^{(n)})$ is linear. Unfortunately, this is usually not the case, and a complete integration of localization and tracking is not possible. [The extended Kalman filter can be used to approximate $\hat{\theta}_k^{(n)}$ by linearization of the system around the current state estimate and application of the Kalman filter to the resulting time-varying, linear system. However, this is algorithm is suboptimal and fairly complex.]

In the following algorithm, localization, and tracking are two separate blocks (in contrast to the previous algorithm and to the algorithm in the next section, which integrates them). Processing is done at each time interval using only data received in that interval to update the results of the previous processing.

A. Localization

In order to achieve a maximum likelihood localization, $\tilde{\theta}_k^{(n)}$ are estimated from the recent observation using the EM algorithm. The initial parameter's value for the EM iterations was predicted by the tracking stage. This prediction is based on previous estimations and, therefore, is close to the actual parameter. Thus, only a few EM iterations may be sufficient.

The complete data is defined as the decomposition of the observed data $\boldsymbol{y}_k(t)$ into its signal components (recall Section III-B). Since the complete data $\boldsymbol{x}_k(t)$ and the incomplete data $\boldsymbol{y}_k(t)$ are jointly Gaussian and related by a linear transformation, the complete data can be calculated in a straightforward manner (see [3]) as

(E-step:)

$$\hat{\boldsymbol{x}}_k^{(n)} = \boldsymbol{s}_k^{(n)}\left(t,\, \tilde{\theta}_k^{(n)}\right) + \beta_n \left[\boldsymbol{y}_k(t) - \sum_{i=1}^{N} \boldsymbol{s}_k^{(n)}\left(t,\, \tilde{\theta}_k^{(n)}\right)\right]. \tag{22}$$

The log-likelihood function is given in (15). Taking a conditional expectation, we obtain

$$\begin{aligned} &E\{\log f_{\boldsymbol{X}}(\boldsymbol{x}_k;\, \tilde{\theta}_k)\,|\,\boldsymbol{y}_k\} \\ &\quad = c' - \sum_{n=1}^{N} \int_T \left[\hat{\boldsymbol{x}}_k^{(n)}(t) - \boldsymbol{s}_k^{(n)}\left(t,\, \tilde{\theta}_k^{(n)}\right)\right]^* \\ &\qquad \cdot \boldsymbol{R}^{-1}\left[\hat{\boldsymbol{x}}_k^{(n)}(t) - \boldsymbol{s}_k^{(n)}\left(t,\, \tilde{\theta}_k^{(n)}\right)\right] dt. \end{aligned} \tag{23}$$

The parameters $\tilde{\theta}_k^{(n)}$, which maximize this expression, are the recent EM estimation for $\boldsymbol{G}\theta_k$. This maximization is equivalent to

(M-step:)

$$\begin{aligned} \tilde{\theta}_k^{(n)} = \underset{\tilde{\theta}_k^{(n)}}{\textbf{argmin}} &\int_T \left[\hat{\boldsymbol{x}}_k^{(n)}(t) - \boldsymbol{s}_k^{(n)}\left(t,\, \tilde{\theta}_k^{(n)}\right)\right]^* \\ &\cdot \boldsymbol{R}^{\mathbf{-1}}\left[\hat{\boldsymbol{x}}_k^{(n)}(t) - \boldsymbol{s}_k^{(n)}\left(t,\, \tilde{\theta}_k^{(n)}\right)\right] dt. \end{aligned}$$

Note that although the initial parameters for each target are close to final EM estimates for that target, target switching can still occur during the EM iterations for low SNR. Therefore, it is advantageous to associate the EM estimates with the tracks (e.g., by a minimum distance search or gating). In addition, note that in the two other algorithms described in this paper, localization and tracking are unified, and consequently, a separate association of tracks to target locations is not required.

B. Tracking

In the tracking stage, the estimated parameters $\tilde{\theta}_k^{(n)}$ produced by the localization stage are used as the observed data. The model for the actual parameters $\theta_k^{(n)}$, which is stated in (19), is repeated here for convenience, and the EM estimates are modeled as a noisy measurement as in

$$\theta_k^{(n)} = \boldsymbol{F}\theta_{k-1}^{(n)} + \boldsymbol{q}_k \tag{24}$$

$$\tilde{\theta}_k^{(n)} = \boldsymbol{G}\theta_k^{(n)} + \boldsymbol{n}_{\mathrm{EM}_k}. \tag{25}$$

The driving noise and the EM-estimation noise are modeled as Gaussian, with covariance matrixes $\boldsymbol{Q}_k = E\{\boldsymbol{q}_k\boldsymbol{q}_k^*\}$ and $\boldsymbol{R}_{\mathrm{EM}_k} = E\{\boldsymbol{n}_{\mathrm{EM}_k}\boldsymbol{n}_{\mathrm{EM}_k}^*\}$. The choice of $\boldsymbol{R}_{\mathrm{EM}_k}$ depends on the observation noise $\boldsymbol{n}_{\mathrm{EM}_k}(t)$ and on the shape of the likelihood function in the neighborhood of $\theta_k^{(n)}$. Using the Kalman filter, an approximated MSE estimation can be evaluated, and a prediction can be made, which can be used as initial the parameter for the next EM localization stage. See the EM-Kalman algorithm in the next column.

C. Example: Time-Delayed Signals

In many applications, $\boldsymbol{s}_k^{(n)}(t,\, \tilde{\theta}_k^{(n)})$ is a time-delayed signal of the form

$$\boldsymbol{s}_k^{(n)}\left(t,\, \tilde{\theta}_k^{(n)}\right) = \alpha_k^{(n)} s\left(t - \tau_k^{(n)}\right)\boldsymbol{a}\left(\psi_k^{(n)}\right) \tag{28}$$

```
The EM-Kalman algorithm
1. Initial states:
  for $n = 1, 2, \cdots, N$ (signals):
      Guess state $\#0,\ \hat{\theta}_{1|0}^{(n)}$
      Set initial error variance, $\boldsymbol{P}_{1|0}$
For $k = 1, 2, \cdots$ (time intervals)
2. Localization:
  for $p = 1, 2, \cdots, Np$ (iterations number):
  for $n = 1, 2, \cdots, N$ (signals):
E-step:
  $\hat{\boldsymbol{x}}_k^{(n)} = \boldsymbol{s}_k^{(n)}\left(t,\, \boldsymbol{G}\hat{\theta}_{k|k-1}^{(n)}\right) + \beta_n\left[\boldsymbol{y}_k(t) - \sum_{i=1}^{N}\boldsymbol{s}_k^{(n)}\left(t,\, \boldsymbol{G}\hat{\theta}_{k|k-1}^{(n)}\right)\right]$    (26)
      where $\sum \beta_n = 1$
M-step:
  $\tilde{\theta}_k^{(n)} = \underset{\tilde{\theta}_k^{(n)}}{\textbf{argmin}} \int_T \left[\hat{\boldsymbol{x}}_k^{(n)}(t) - \boldsymbol{s}_k^{(n)}\left(t,\, \tilde{\theta}_k^{(n)}\right)\right]^* \cdot \boldsymbol{R}^{\mathbf{-1}}\left[\hat{\boldsymbol{x}}_k^{(n)}(t) - \boldsymbol{s}_k^{(n)}\left(t,\, \tilde{\theta}_k^{(n)}\right)\right]$    (27)
3. Kalman tracking:
  for $n = 1, 2, \cdots, N$ (signals):
  $\boldsymbol{K}_k^{(n)} = \boldsymbol{P}_{k/k-1}^{(n)}\boldsymbol{G}^T\left[\boldsymbol{G}\boldsymbol{P}_{k/k-1}^{(n)}\boldsymbol{G}^T + \boldsymbol{R}_{\mathrm{EM}_k}\right]$
  $\hat{\theta}_k^{(n)} = \hat{\theta}_{k/k-1}^{(n)} - \boldsymbol{K}_k^{(n)}\left[\tilde{\theta}_k^{(n)} - \boldsymbol{G}\hat{\theta}_{k/k-1}^{(n)}\right]$
  $\boldsymbol{P}_k^{(n)} = [\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{G}]\boldsymbol{P}_{k/k-1}^{(n)}$
  $\hat{\theta}_{k+1/k}^{(n)} = \boldsymbol{F}\hat{\theta}_k^{(n)}$
  $\boldsymbol{P}_{k+1|k}^{(n)} = \boldsymbol{F}\boldsymbol{P}_k^{(n)}\boldsymbol{F}^T + \boldsymbol{Q}$
```

where the parameters vector $\tilde{\theta}_k^{(n)}$ is composed of the time-delay parameter $\tau_k^{(n)}$ and the vector $\psi_k^{(n)}$, which holds other parameters (such as azimuth and elevation)

$$\tilde{\theta}_k^{(n)} = \begin{bmatrix} \tau_k^{(n)} \\ \psi_k^{(n)} \end{bmatrix}. \tag{29}$$

$\alpha_k^{(n)}$ is an unknown additional deterministic parameter.

The matched filter can now be placed in the sensors level as in

$$y_{\mathrm{MF}_k}^{(n)}\left(\tau_k^{(n)}\right) = \int_T y_k^{(n)}(t)s^*\left(t - \tau_k^{(n)}\right)dt \tag{30}$$

$$s_{\mathrm{MF}}^{(n)}(\tau) = \int_T s(t)s^*(t-\tau)\,dt \tag{31}$$

and $Es = \int_T |s(t)|^2$ is the signal's energy.

Substituting (30) and (31) in (26) and (28) in (27) (see Appendix B), the following algorithm is produced.

```
The EM-Kalman algorithm for time delayed signals:
1. Initial states:
for $n = 1, 2, \cdots, N$ (signals):
        guess state $\#0$, $\hat{\theta}_{1|0}^{(n)}$
    set initial error variance, $\boldsymbol{P}_{1|0}$
For $k = 1, 2, \cdots$
2. Localization:
for $p = 1, 2, \cdots, Np$ (iterations number):
for $n = 1, 2, \cdots, N$ (signals):
E-step:
    $\hat{\boldsymbol{x}}_{\mathrm{MF}_k}^{(n)}(t) = \hat{\alpha}_{k|k-1}^{(n)} s_{\mathrm{MF}}\left(\hat{\tau}_{k|k-1}^{(n)}\right)\boldsymbol{a}\left(\hat{\psi}_{k|k-1}^{(n)}\right) + \beta_n \left[\boldsymbol{y}_{\mathrm{MF}_k}(t) - \sum_{i=1}^{N} \hat{\alpha}_k^{(n)} s_{\mathrm{MF}} \cdot \left(\hat{\tau}_{k|k-1}^{(n)}\right)\boldsymbol{a}\left(\hat{\psi}_{k|k-1}^{(n)}\right)\right]$    (32)
M-step:
$g\left(\tau_k^{(n)}, \psi_k^{(n)}\right) = \sum_{m=1}^{M} a_m^*\left(\psi_k^{(n)}\right)\hat{x}_{\mathrm{MF}_{k,m}}^{(n)} \cdot \left(\tau_k^{(n)}\right) r_{k,m}^{-1}$
    $\tilde{\theta}_k^{(n)} = \mathop{\mathbf{argmax}}_{\tau_k^{(n)}, \psi_k^{(n)}} \left|g\left(\tau_k^{(n)}, \psi_k^{(n)}\right)\right|$
    $\arg\left(\alpha_k^{(n)}\right) = -\arg\left(g\left(\tilde{\theta}_k^{(n)}\right)\right)$
    $\left|\alpha_k^{(n)}\right| = \frac{\left|g\left(\tilde{\theta}_k^{(n)}\right)\right| / Es}{\sum_{m=1}^{M} r_{k,m}^{-1}\left|a_m\left(\psi_k^{(n)}\right)\right|^2}$
3. Kalman tracking:
for $n = 1, 2, \cdots, N$ (signals):
    $\boldsymbol{K}_k^{(n)} = \boldsymbol{P}_{k/k-1}^{(n)}\boldsymbol{G}^T\left[\boldsymbol{G}\boldsymbol{P}_{k/k-1}^{(n)}\boldsymbol{G}^T + \boldsymbol{R}_{\mathrm{EM}_k}\right]$
    $\hat{\theta}_k^{(n)} = \hat{\theta}_{k/k-1}^{(n)} - \boldsymbol{K}_k^{(n)}\left[\tilde{\theta}_k^{(n)} - \boldsymbol{G}\hat{\theta}_{k/k-1}^{(n)}\right]$
    $\boldsymbol{P}_k^{(n)} = [\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{G}]\boldsymbol{P}_{k/k-1}^{(n)}$
$\hat{\theta}_{k+1/k}^{(n)} = \boldsymbol{F}\hat{\theta}_k^{(n)}$
$\boldsymbol{P}_{k+1|k}^{(n)} = \boldsymbol{F}\boldsymbol{P}_k^{(n)}\boldsymbol{F}^T + \boldsymbol{Q}$
```

V. MAP Estimation and the EM-HMM Algorithm

An alternative Bayesian criterion is the MAP estimation

$$\hat{\theta} = \max_{\theta}\{f(\theta|\boldsymbol{y}_1 \cdots \boldsymbol{y}_k)\} \tag{33}$$

where the parameter vector θ is composed of the vectors $\theta_k^{(n)}$ (21) and modeled as a discrete stochastic process (19). The observed data $\boldsymbol{y}$ is composed of superimposed signals that are a function of this process (1). Here, an exact solution can be achieved by maximizing the conditioned distribution (13) over all the parameters. However, each parameter $\theta_k^{(n)}$ has a value from a set of N_s discrete states (for example, N_s possible combinations of locations and velocities) and direct search over the parameters requires calculation of the likelihood function (13) for each state, target, and time interval, namely, N_s^{NK} calculations. The number of calculations is, therefore, exponential with respect to the number of observations K and to the number of signals N. In this section, we first formulate an HMM problem and briefly refer to an algorithm that reduces the computational complexity with respect to the observations' number K. Then, we integrate this algorithm with the EM algorithm and reduce the dependence with respect to the number of signals N in the same way.

A. Formulating the HMM Problem

For discrete parameters, the stochastic model (19) is a discrete, first-order, Markov chain model. Let $\{S_1, S_2, \cdots, S_{N_S}\}$ denote all the possible states of the vector $\hat{\theta}_k^{(n)}$ (N_S is the number of all the possible combinations of the parameters' values for a single observation k, and a single signal n). The state transition probability $\boldsymbol{A}_{\boldsymbol{y}} = \{a_{i_1 i_2 \cdots i_N\, j_1 j_2 \cdots j_N}\}$, which is the probability that the N targets reach state $\boldsymbol{S}_{j_1}, \boldsymbol{S}_{j_2}, \cdots, \boldsymbol{S}_{j_N}$ from state $\boldsymbol{S}_{i_1}, \boldsymbol{S}_{i_2}, \cdots, \boldsymbol{S}_{i_N}$ in a single step, is determined by the probability distribution of the driving noise $\boldsymbol{q}$. One possibility of choosing this distribution is a discrete approximation of the normal distribution

$$\begin{aligned} &a_{i_1 i_2 \cdots i_N\, j_1 j_2 \cdots j_N} \\ &\quad = P\Big\{\theta_{k+1}^{(1)} = \boldsymbol{S}_{j_1}, \theta_{k+1}^{(2)} = \boldsymbol{S}_{j_2}, \cdots, \theta_{k+1}^{(N)} = \boldsymbol{S}_{j_N} | \\ &\qquad\quad \theta_k^{(1)} = \boldsymbol{S}_{i_1}, \theta_k^{(2)} = \boldsymbol{S}_{i_2}, \cdots, \theta_k^{(N)} = \boldsymbol{S}_{i_N}\Big\} \\ &\quad = c \cdot \prod_{l=1}^{N} \int_{\boldsymbol{s}_j \in N_{S_j}} \exp\{-\tfrac{1}{2}[\boldsymbol{s}_j - \boldsymbol{F}\boldsymbol{S}_{i_l}]^* \boldsymbol{Q}^{-1}[\boldsymbol{s}_j - \boldsymbol{F}\boldsymbol{S}_{i_l}]\} \\ &\qquad \cdot d\boldsymbol{s}_j \end{aligned} \tag{34}$$

where N_{S_j} is some neighborhood of S_j, which is defined for the specific chosen parameters. The initial state distributions are

$$\begin{aligned} \pi_{i_1 i_2 \cdots i_N} &= P\Big\{\theta_0^{(1)} = \boldsymbol{S}_{i_1}, \theta_0^{(2)} = \boldsymbol{S}_{i_2}, \cdots, \theta_0^{(N)} = \boldsymbol{S}_{i_N}\Big\} \\ \Pi_y &= \{\pi_{i_1 i_2 \cdots i_N}\}. \end{aligned} \tag{35}$$

The process $\theta_1^{(n)}, \theta_2^{(n)}, \cdots, \theta_k^{(n)}, \cdots$ is hidden; in other words, it is not observed directly. The observations $\boldsymbol{y}_k(t)$ in (1) are probabilistic functions of the process. $b_j(\boldsymbol{y}(t))$ is defined as the continuous probability distribution of the snapshot data $\boldsymbol{y}_k(t)$ that is conditioned by state S_j. Let $\boldsymbol{B}_y = \{b_j(\boldsymbol{y}(t))\}$. For

Gaussian observation noise $n(t)$, with covariance matrix $\boldsymbol{R}$

$$\begin{aligned} &b_{j_1,\cdots,j_N}(\boldsymbol{y}_k(t)) \\ &\quad = f_{y_k(t)}\Big(\boldsymbol{y}_k(t)|\theta_k^{(1)} = S_{j_1},\cdots,\theta_k^{(N)} = S_{j_N}\Big) \\ &\quad = c\cdot\exp\Bigg\{-\int_T\Bigg[\boldsymbol{y}_k(t)-\sum_{n=1}^{N}\boldsymbol{s}_k^{(n)}(t,S_{j_n})\Bigg]^* \\ &\qquad\qquad \cdot\boldsymbol{R}^{-1}\Bigg[\boldsymbol{y}_k(t)-\sum_{n=1}^{N}\boldsymbol{s}_k^{(n)}(t,S_{j_n})\Bigg]\Bigg\}. \end{aligned}$$

The probability measures $\boldsymbol{A}_y$, $\boldsymbol{B}_y$, Π_y, and the states number N_S^N constitutes a complete specification of an HMM. The general HMM algorithm that uncovers the hidden part of the model, i.e., finds $\hat{\theta}_k^{(n)}$ that satisfies (33) given the model parameters, is described in a tutorial fashion in [5]. The heart of this HMM algorithm is the Viterbi algorithm, in which the new probability of each state combination is calculated from the one saved for time $k-1$ and from the new data $\boldsymbol{y}_k(t)$. This probability distribution is denoted by $\delta_k(j_1,j_2,\cdots,j_N)$, $j_n\in\{1\cdots N_S\}$. The number of calculations is, therefore, proportional to $(N_S^N)^2$ and is exponential with respect to the signal's number N. In a typical tracking sonar problem, for example, even two superimposed signals tracking may be unfeasible.

B. An Integrated EM–HMM Algorithm

The EM algorithm can be used to reduce this exponential dependence in the targets' number N to a linear dependence. The complete data is defined, again, as the decomposition of the observed data $\boldsymbol{y}_k(t)$ into its signal components (14).

In this case, the state transition matrix for *one* signal from state $\theta_k^{(n)}=\boldsymbol{S}_i$ to state $\theta_{k+1}^{(n)}=\boldsymbol{S}_j$ is $\boldsymbol{A}=\{a_{ij}\}$, where

$$\begin{aligned} a_{ij} &= P\Big\{\theta_{k+1}^{(n)}=\boldsymbol{S}_j|\theta_k^{(n)}=\boldsymbol{S}_i\Big\} \\ &= c\cdot\int_{s_j\in N_{S_j}}\exp\{-\tfrac{1}{2}[\boldsymbol{s}_j-\boldsymbol{F}\boldsymbol{S}_i]^*\boldsymbol{Q}^{-1}[\boldsymbol{s}_j-\boldsymbol{F}\boldsymbol{S}_i]\}d\boldsymbol{s}_j. \end{aligned} \tag{36}$$

The initial state distribution is $\pi_i = P\{\theta_0^{(n)}=\boldsymbol{S}_i\}$, $\Pi=\{\pi_i\}$, and the continuous probability distribution of $\boldsymbol{x}_k^{(n)}(t)$ given state $\theta_k^{(n)}=\boldsymbol{S}_j$ can be defined as $\boldsymbol{B}=\{b_j(\boldsymbol{x}_k^{(n)}(t))\}$, where

$$\begin{aligned} b_j\Big(\boldsymbol{x}_k^{(n)}(t)\Big) &= f_{X_k(t)}\Big(\boldsymbol{x}_k^{(n)}(t)|\theta_k^{(n)}=S_j\Big) \\ &= c\cdot\exp\Bigg\{-\int_T\Big[\boldsymbol{x}_k^{(n)}(t)-\boldsymbol{s}_k^{(n)}(t,S_j)\Big]^* \\ &\qquad\cdot\boldsymbol{R}^{-1}\Big[\boldsymbol{x}_k^{(n)}(t)-\boldsymbol{s}_k^{(n)}(t,S_j)\Big]\Bigg\}. \end{aligned}$$

The probability measures $\boldsymbol{A}$, $\boldsymbol{B}$, Π and the states number Ns define an HMM that uses the given data for observations. Therefore, given the complete data, an HMM algorithm can be utilized to obtain

$$\hat{\theta}^{(n)} = \max_{\theta^{(n)}}\Big\{f\Big(\theta^{(n)}|\boldsymbol{x}_1^{(n)}\cdots\boldsymbol{x}_K^{(n)}\Big)\Big\} \tag{37}$$

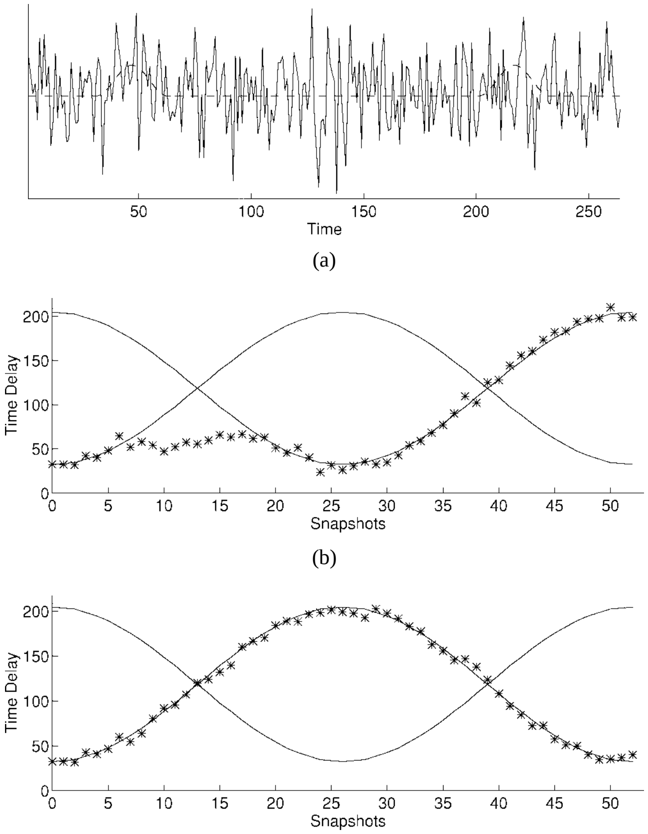


Fig. 4. Comparison of stochastic approximation algorithms. (a) First snapshot, the original signal (–), and the noisy signal (-). (b) Actual track (-) and a conventional stochastic approximation algorithm estimations (o, $*$). (c) Actual track (-) and the EM–Newton algorithm estimates (o, $*$).

where $\theta^{(n)}$ track parameters of the Nth target

$$\theta^{(n)} = \begin{bmatrix}\theta_1^{(n)}\\ \theta_2^{(n)}\\ \vdots\\ \theta_K^{(n)}\end{bmatrix}. \tag{38}$$

We will use this algorithm as the M-step in a EM-Algorithm. The E-step, which produces the estimations of the complete data, is similar to the one used for the EM–Kalman algorithm (22).

To calculate the M-step (37) using the Viterbi algorithm, we need to define the quantity $\delta_k^{(n)}(j)$, which is the best score (highest probability) along a single track, after k observations, which accounts for the first k observations and ends in state S_j

$$\begin{aligned} \delta_k^{(n)}(j) = &\max_{\theta_1^{(n)},\theta_2^{(n)},\cdots,\theta_{k-1}^{(n)}} P\Big\{\theta_1^{(n)},\theta_2^{(n)},\cdots,\theta_{k-1}^{(n)} \\ &\qquad \theta_k^{(n)}=S_j,\boldsymbol{x}_1^{(n)},\boldsymbol{x}_2^{(n)},\cdots,\boldsymbol{x}_k^{(n)}(t)|\boldsymbol{A},\boldsymbol{B},\Pi,\theta\Big\} \\ = c\cdot &\max_{\theta_1^{(n)},\theta_2^{(n)},\cdots,\theta_{k-1}^{(n)}} P\Big\{\theta_1^{(n)},\theta_2^{(n)},\cdots,\theta_{k-1}^{(n)} \\ &\qquad \theta_k^{(n)}=S_j|\boldsymbol{x}_1^{(n)},\boldsymbol{x}_2^{(n)},\cdots,\boldsymbol{x}_k^{(n)}(t),\boldsymbol{A},\boldsymbol{B},\Pi,\theta\Big\} \end{aligned}$$

where c is independent of θ. By induction

$$\delta_k^{(n)}(j) = \max_{S_i,\, 1 \le i \le N_S} \left[\delta_{k-1}^{(n)}(i) a_{ij}\right] b_j\left(\boldsymbol{x}_k^{(n)}(t)\right). \qquad (39)$$

In order to keep the track that maximizes $\delta_k^{(n)}(j)$, the index i of the previous most likely state S_i is saved in the array $\psi_k^{(n)}(j)$. This index is the argument that maximizes (39).

The EM parameters, in this case, are all the track parameters. The initial guess $\hat{\theta}_{k,0}^{(n)}$ is, therefore, a guess of the initial track of the nth target.

The integrated algorithm can now be stated as follows.

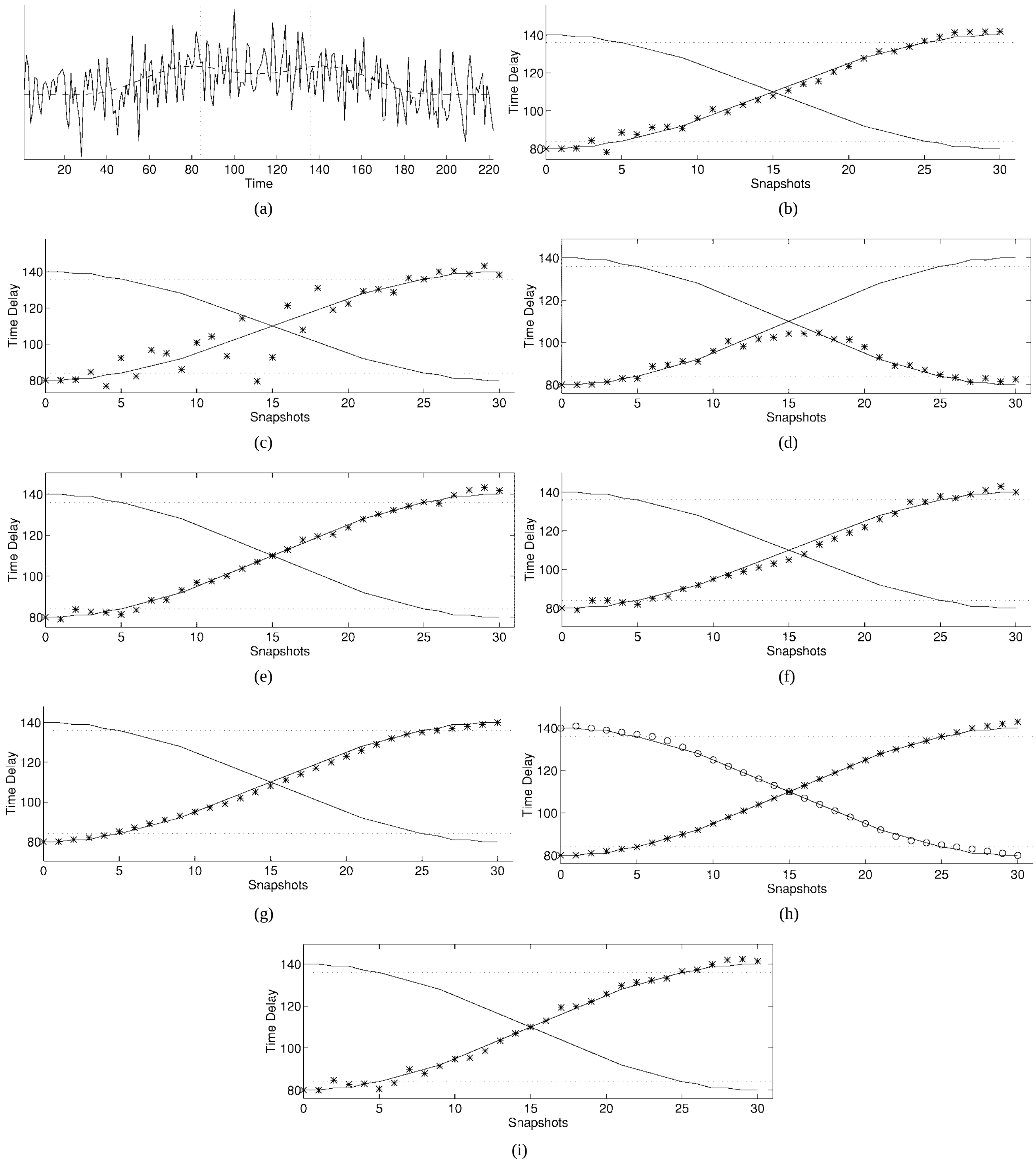
```
The batch EM-HMM algorithm:
1. Initial track guess
        for $n = 1, 2, \cdots, N$ (signals):
                guess track #0, $\hat{\theta}_{k,0}^{(n)} (1 \le k \le K)$
for $p = 1, 2, \cdots, Np$ (EM iterations):
2. Recursion:
        for $k = 1, 2, \cdots, K$ (snapshots):
        for $n = 1, 2, \cdots, N$ (signals):
E-step:
    $\hat{\boldsymbol{x}}_{k,p}^{(n)} = \boldsymbol{s}_k^{(n)}\left(t, \hat{\theta}_{k,p-1}^{(n)}\right) + \beta_n \left[\boldsymbol{y}_k(t) - \sum_{i=1}^{N} \boldsymbol{s}_k^{(n)}\left(t, \hat{\theta}_{k,p-1}^{(n)}\right)\right]$
            where $\sum \beta_n = 1 \qquad (1 \le k \le K)$
M-step:
        for $j = 1, 2, \cdots, Ns$ (new state)
    $\delta_k^{(n)}(j) = \begin{cases} \pi_j b_j\left(\boldsymbol{x}_{1,p}^{(n)}(t)\right) & (k = 1) \\ \max_{S_i,\, 1 \le i \le N_S} \left[\delta_{k-1}^{(n)}(S_i) a_{ij}\right] \cdot b_j\left(\boldsymbol{x}_{k,p}^{(n)}(t)\right) & (2 \le k \le K) \end{cases}$
    $\psi_k^{(n)}(S_j) = \operatorname{argmax}_{S_i,\, 1 \le i \le N_S} \left[\delta_{k-1}^{(n)}(S_i) a_{ij}\right] \qquad (2 \le k \le K)$
3. Termination:
        for $n = 1, 2, \cdots, N$ (signals):
            $i_{K,p}^{(n)} = \operatorname{argmax}_{1 \le i \le N_S} \left[\delta_K^{(n)}(i)\right],$
            $\hat{\theta}_{K,p}^{(n)} = S_{i_{K,p}^{(n)}} \qquad (k = K)$
4. Backtracking
        for $k = K-1, K-2, \cdots, 1$ (snapshots):
        for $n = 1, 2, \cdots, N$ (signals):
            $i_{k,p}^{(n)} = \psi_{k+1}^{(n)}\left(i_{k+1,p}^{(n)}\right), \quad \hat{\theta}_{k,p}^{(n)} = S_{i_{k,p}^{(n)}}$
```

For sufficient resolution of the parameters' quantization, the algorithm above is an exact EM algorithm and will produce parameters that locally maximize the likelihood function (33).

A recursive algorithm, which uses the last observation and does not accumulate data, can be now formulated. In each iteration, a new observation (k) is used, and a new complete data for time k is estimated (E-step). The complete data for the old observations is not changed since the old observations data is not saved, and their complete data can not be estimated. The M-step produces the parameters at time k that maximize the track probability given the complete data. Only one probability matrix is required to be saved in order to achieve this maximization.

The parameters at time k are associated with the track covered until that time, and each iteration a new parameters vector is added to the track. The initial parameters $\hat{\theta}_0^{(n)}$ are, therefore, the initial guesses for signal n, for the state before the first observation, and not a full track, as in the batch algorithm.

```
The recursive EM-HMM algorithm:
1. Initial state guess
        for $n = 1, 2, \cdots, N$ (signals):
            guess state #0, $\hat{\theta}_0^{(n)}$                                  (40)
for $k = 1, 2, \cdots$ (observations):
2. Recursion:
        for $n = 1, 2, \cdots, N$ (signals):
E-step:
    $\hat{\boldsymbol{x}}_k^{(n)} = \boldsymbol{s}_k^{(n)}\left(t, \boldsymbol{F}\hat{\theta}_{k-1}^{(n)}\right) + \beta_n \left[\boldsymbol{y}_k(t) - \sum_{i=1}^{N} \boldsymbol{s}_k^{(n)}\left(t, \boldsymbol{F}\hat{\theta}_{k-1}^{(n)}\right)\right]$
M-step:
        for $j = 1, 2, \cdots, Ns$ (new states indices)
    $\delta_k^{(n)}(j) = \begin{cases} \pi_j b_j\left(\boldsymbol{x}_1^{(n)}(t)\right) & (k = 1) \\ \max_{S_i,\, 1 \le i \le N_S} \left[\delta_{k-1}^{(n)}(i) a_{ij}\right] \cdot b_j\left(\boldsymbol{x}_k^{(n)}(t)\right) & (k \ge 2) \end{cases}$    (41)
    $\psi_k^{(n)}(S_j) = \operatorname{argmax}_{S_i,\, 1 \le i \le N_S} \left[\delta_{k-1}^{(n)}(S_i) a_{ij}\right],$    (42)
            (optional)    $(k \ge 2)$
3. On line estimate:
        for $n = 1, 2, \cdots, N$ (signals):
            $i_k^{(n)} = \operatorname{argmax}_{1 \le i \le N_S} \left[\delta_k^{(n)}(i)\right],$
            $\hat{\theta}_k^{(n)} = S_{i_k^{(n)}} \qquad (k \ge 2)$                   (43)
4. Backtracking (optional)
        for $l = k-1, k-2, \cdots, 1$
            (observations):
        for $n = 1, 2, \cdots, N$ (signals):
            $i_l^{(n)} = \psi_{l+1}^{(n)}\left(i_{l+1}^{(n)}\right), \quad \hat{\theta}_l^{(n)} = S_{i_l^{(n)}}$
```



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 5. Resolution performance. (a) First snapshot, the original signal (–), the noisy signal (-), a match filter resolution cell (..). (b) EM–Newton algorithm with $\lambda = 0.9$. (c) EM–Newton algorithm with $\lambda = 0.6$. (d) Conventional stochastic approximation algorithm with $\lambda 0 = 0.4$. (e) EM–Kalman algorithm. (f) Recursive EM–HMM algorithm. (g) Backtracking. (h) Batch EM–HMM algorithm. (i) EM–Kalman algorithm for signals with unknown phase and energy.

VI. Simulation

Simulations were performed for the case of a sonar with linear array of omnidirectional sensors. The algorithms achieved a high resolution in range and azimuth, in low SNR, and random phase. The EM–Kalman algorithm was implemented for the FTV three-dimensional (3-D) sonar [4], and tested with real data. Hanning signals $(s(t) = (0.5 - 0.5\cos(2\pi t/T))\cos(2\pi F_C t)$ for $0 < t < T)$ with duration of $T = 31$ or $T = 101$ time units were used for all simulations. The carrier period was normalized to a single time unit $(F_C = 1)$. The delayed complex baseband representations of the signals were sampled at a rate of one sample per time unit and a white Gaussian noise $n(k)$, which was 5 dB larger than the amplitude of the signal $(E(n(k)n(k)^*) = 10^{5/10})$, was added. Snapshots of the Hanning baseband signals before and after adding the noise are shown in Figs. 4(a) and 5(a).

Fig. 4 demonstrates the advantage of using the EM–Newton algorithm over a conventional stochastic approximation algorithm when there is a good dynamic model for the targets. Fig. 4(a) shows the first snapshot of two Hanning signals of duration $T = 31$ time units that were received by a single sensor. The time delay of the signal represents the targets' ranges. In the following snapshots, the ranges change, and targets maneuver in sinus-like tracks. Fig. 4(b) shows the time-delay estimations of the conventional stochastic approximation algorithm (12) with $\lambda_0 = 0.4$. Tracking was not stable for any λ_0. Fig. 4(c) shows the EM–Newton algorithm (8) and (9) estimates, using a linear model for the track $(\lambda = 0.6)$. Clearly, for this case, there is an advantage of using the model-based EM–Newton algorithm.

Fig. 5 compares the algorithms' resolution performance for known energy and phase. Two Hanning signals of duration $T = 101$ were received by five omnidirectional sensors spaced half a wave length apart on a straight line. These signals had a longer time span than the signals of Fig. 1 and, therefore, were more difficult to separate. The dotted lines in the figures bounds a resolution cell. Inside this cell, a conventional matched filter fails to separate the two targets. The signals are then getting closer, moving in curved tracks that are modeled by straight lines. All the algorithms manage to track for this SNR and resolution.

The *EM–Newton* algorithm was used with forgetting factor of $\lambda = 0.9$ and initial time delay within the signal duration [Fig. 5(b)]. λ is the only parameter that has to be predetermined. The tracking was found to be very robust with respect to this parameter, and tracks were not lost for low λ values, in spite of the noisy tracking [see Fig. 5(c) for $\lambda = 0.6$]. The conventional stochastic approximation algorithm fails to follow the crossed track since it does not estimate the velocity [Fig. 5(d)]. Notice that the reason for this failure is different from the example of Fig 4. For the example of Fig. 5, the EM–Newton algorithm uses the better model to resolve the targets interference, whereas in the example of Fig. 4, it was used to keep the location estimation in the neighborhood of the signal.

The *EM–Kalman* algorithm for time-delayed signals has a more global nature and, therefore, was less sensitive to starting conditions and temporary losses of track. On the other hand, more parameters must be established in order to set a satisfactory Bayesian dynamic model for the system. Results are plotted in Fig. 5(e) for four iterations per snapshot.

The *recursive EM–HMM* algorithm results for this example are shown in Fig. 5(f). The track is not smooth since every new location may be the ending point of a completely different track. Backtracking can be done at each stage to reveal the track that was last chosen. The track of Fig. 5(g), for example, is the result of a such back tracking taken at the last snapshot of Fig. 5(f). This is the most probable track under the model assumptions, given the complete data estimation. Using the track of the recursive EM–HMM algorithm in Fig. 5(g) as an initial parameter (iteration #0), the batch EM–HMM algorithm produces the track of Fig. 5(h).

The algorithms were tested with modulated signals, of unknown energy and phase. The energy uncertainty introduced only a small degradation in tracking. Phase estimation required more EM iterations for each snapshot. The track in Fig. 5(i) is the result of the EM–Kalman algorithm for the same example, with unknown energy and phases of the signals, and 16 iterations per snapshot.

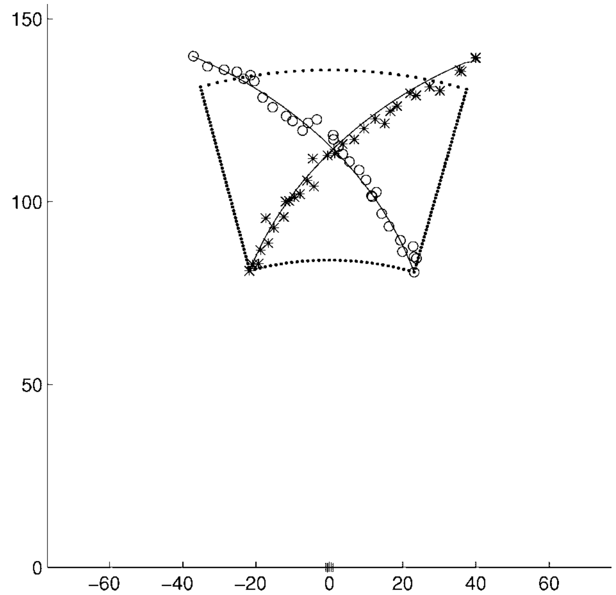
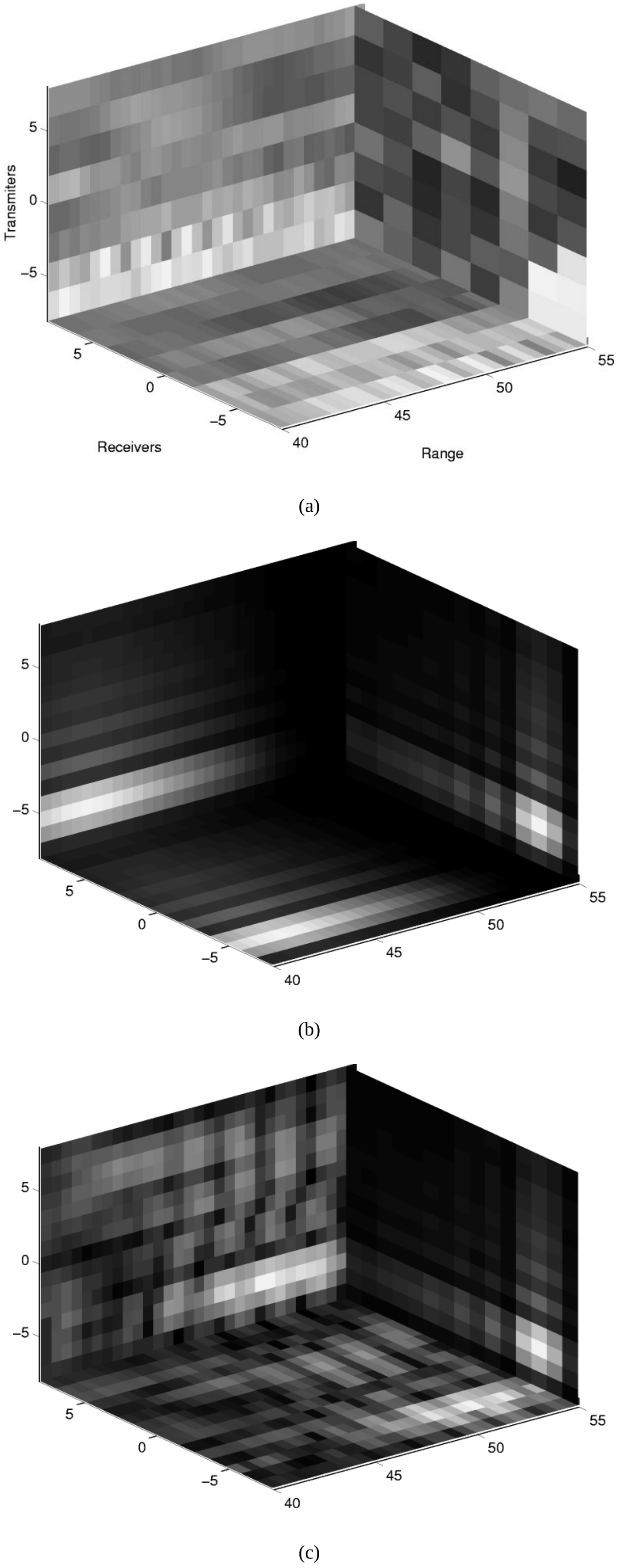


Fig. 6. Two-dimensional tracking with five omnidirectional sensors.

Figs. 6 and 7 demonstrate an application of the algorithms to tracking two and three dimensions. The five dots at the bottom of Fig. 6 represent five omnidirectional sensors spaced a half wave length apart on a straight line. Hanning signals of duration $T = 101$ were used. The dotted line encloses an area in which a beamformer and a matched filter cannot separate the two targets. The EM–Kalman algorithm was used with a spatial gradient to track the targets in the plane (angle and time).

Fig. 7 illustrates a simulated FTV sonar snapshot [4]. The FTV is a special sonar developed at Scripps Oceanography Institute to track zoo plankton in three dimensions. This sonar can be modeled as an 8 × 8 sensor array, each with a narrow (2°) beam pattern that is directed to a different point, together covering a rectangle of 16 × 16° and a range of a few meters. The three planes in Fig. 7(a) display the projections of the 3-D data after match filtering. Lighter areas represent more received energy. Complete data can be defined as the data received from each target separately. Using a model for the beampatterns of the sensors and knowing the shape of the transmitted signal, the superimposed signal model can be formulated, and the complete data can be estimated from the received data in Fig. 7(a) and from a previous estimate of the target location. This is an E-step of the EM–Kalman algorithm, and its result is displayed in Fig. 7(b) and (c). Notice that the limited spatial resolution, which was imposed by the hardware, is increased by the algorithm (the resolution is only restricted by the SNR and the accuracy of the model for the sensors). A 3-D track can be produced by applying the algorithm to the next snapshots.



(a)

(b)

(c)

Fig. 7. Three-dimensional tracking with the "FTV" sonar. (a) Projections of the 3-D data of one snapshot after match filtering ("incomplete" data). (b) Projections of the estimated "complete" data for the first target. (c) Projections of the estimated "complete" data for the second target.

VII. Conclusions

In this paper, the EM algorithm was applied to multiple target tracking for a known number of targets. Three major algorithms were proposed based on different optimization criteria.

The "EM–Newton" was a second approximation of the recursive EM algorithm (maximum likelihood approach). We showed, through simulations, that by using a model for the dynamics, the EM–Newton can provide velocity information and keep track even when targets cross each other or have high velocity compared with the data rate. However, like other gradient-based algorithms, the EM–Newton should not be used with modulated signals if the phases change rapidly between iterations as this can produce incorrect scores. Investigating conditions for convergence when the model is exact would be worthwhile.

The "EM–Kalman" algorithm used EM localization with Kalman tracking (MSE approach). The algorithm for time-delayed signals was capable of estimating unknown phase and energy. This, and its low complexity, make the algorithm suitable for real-time sonar applications. The EM–Kalman is composed of matched filtering and signal decomposition followed by beamforming and Kalman filtering for each of the N targets (Fig. 3). Its computational complexity is, therefore, roughly N times larger than a standard beamformer sonar that uses a matched filter, shared beamforming, and a Kalman filter for each target. The signal decomposition complexity is negligible compared with the search over the parameters.

The "EM-HMM" algorithm that used a discrete model for the parameters and a Viterbi search for the maximum (MAP approach) achieved the best tracking performance. On the other hand, it is computationally complex, especially when the parameters are defined with high resolution. Gating and targets to tracks association that are commonly used for the real-time sonar tracking [8] are also recommended when the EM algorithms are used for this application. This can prevent target switching between iterations and suppress the effect of additional targets (over the N that are being tracked). Gating also decreases the computation complexity considerably and reduces the probability of false detection of local maxima.

Appendix A
Derivation of the EM–Newton Algorithm (8) and (9) from the Recursive EM Algorithm (6) and (7)

The Taylor expansion, for the "complete" log-likelihood function is, approximately

$$\begin{aligned}\log f_X(\boldsymbol{x}_{k+1};\theta_{k+1}) &= \log f_X(\boldsymbol{x}_{k+1};\boldsymbol{F}\hat{\theta}_k)+(\theta_{k+1}-\boldsymbol{F}\hat{\theta}_k)^T\nabla_{\theta_{k+1}}\\ &\quad\cdot \log f_X(\boldsymbol{x}_{k+1};\theta_{k+1})|_{\theta_{k+1}=\boldsymbol{F}\hat{\theta}_k}+\tfrac{1}{2}(\theta_{k+1}-\boldsymbol{F}\hat{\theta}_k)^T\nabla^2_{\theta_{k+1}}\\ &\quad\cdot \log f_X(\boldsymbol{x}_{k+1};\theta_{k+1})|_{\theta_{k+1}=\boldsymbol{F}\hat{\theta}_k}(\theta_{k+1}-\boldsymbol{F}\hat{\theta}_k). \qquad (44)\end{aligned}$$

Using the relations

$$\begin{aligned}\nabla_{\theta_{k+1}} \log f_X(\boldsymbol{x}_{k+1};\,\theta_{k+1}) &= \nabla_{\theta_{k+1}} \log f_Y(\boldsymbol{y}_{k+1};\,\theta_{k+1})\\ &\quad+\nabla_{\theta_{k+1}} \log f_{X/Y}(\boldsymbol{x}_{k+1}|\boldsymbol{y}_{k+1};\,\theta_{k+1}) \qquad (45)\end{aligned}$$

and

$$E\{\nabla_{\theta_{k+1}} \log f_{X/Y}(\boldsymbol{x}_{k+1}/\boldsymbol{y}_{k+1};\, \theta_{k+1})|\boldsymbol{y}_{k+1}\} = 0$$

and taking a conditional expectation, (10) becomes

$$\boldsymbol{S}(\boldsymbol{y}_{k+1},\, \theta_{k+1}) = E\{\nabla_{\theta_{k+1}} \log f_X(\boldsymbol{x}_{k+1};\, \theta_{k+1})|\boldsymbol{y}_{k+1}\}.$$

Defining

$$\begin{aligned}&\tilde{\boldsymbol{Ic}}(\boldsymbol{y}_{k+1},\, \boldsymbol{F}\hat{\theta}_k)\\ &\quad = -E\{\nabla^2_{\theta_{k+1}} \log f_X(\boldsymbol{x}_{k+1};\, \theta_{k+1})|_{\theta_{k+1}=\boldsymbol{F}\hat{\theta}_k}|\boldsymbol{y}_{k+1}\}\end{aligned}$$

and taking a conditional expectation on (44), we have

$$\begin{aligned}&E\{\log f_X(\boldsymbol{x}_{k+1};\, \theta_{k+1})|\boldsymbol{y}_{k+1};\, \boldsymbol{F}\hat{\theta}_k\}\\ &\quad = c + (\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k)^T \boldsymbol{S}(\boldsymbol{y}_{k+1};\, \boldsymbol{F}\,\hat{\theta}_k)\\ &\qquad - \tfrac{1}{2}(\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k)^T \tilde{\boldsymbol{Ic}}(\boldsymbol{y}_{k+1},\, \boldsymbol{F}\hat{\theta}_k)(\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k)\\ &\quad \approx c + (\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k)^T \boldsymbol{S}(\boldsymbol{y}_{k+1};\, \boldsymbol{F}\hat{\theta}_k)\\ &\qquad - \tfrac{1}{2}(\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k)^T \boldsymbol{Ic}(\boldsymbol{F}\hat{\theta}_k)(\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k) \qquad (46)\end{aligned}$$

where c is a constant independent of θ_{k+1}, and $\tilde{\boldsymbol{Ic}}(\boldsymbol{F}\hat{\theta}_k;\, \boldsymbol{y}_{k+1})$ is being approximated by its expectation

$$\boldsymbol{Ic}(\boldsymbol{F}\hat{\theta}_k).$$

Assuming the parameters $\hat{\theta}_k$ were evaluated during the last M-step [maximizing $L_k(\theta_k)$] and

$$L_k(\theta_k) = c - \tfrac{1}{2}(\theta_k - \hat{\theta}_k)^T \hat{\boldsymbol{Ic}}_k(\theta_k - \hat{\theta}_k) \qquad (47)$$

then, substituting (46) and (47) into the (6) (E-step), we obtain

$$\begin{aligned}L_{k+1}(\theta_{k+1}) =\,& c' - \gamma\tfrac{1}{2}(\boldsymbol{F}^{-1}\theta_{k+1} - \hat{\theta}_k)^T \hat{\boldsymbol{Ic}}_k(\boldsymbol{F}^{-1}\theta_{k+1} - \hat{\theta}_k)\\ &+ (\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k)^T \boldsymbol{S}(\boldsymbol{y}_{k+1};\, \boldsymbol{F}\hat{\theta}_k)\\ &- \tfrac{1}{2}(\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k)^T \boldsymbol{Ic}(\boldsymbol{F}\hat{\theta}_k)(\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k)\end{aligned} \qquad (48)$$

or

$$\begin{aligned}L_{k+1}(\theta_{k+1}) =\,& c' - \tfrac{1}{2}(\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k)^T\\ &\cdot \{\gamma \boldsymbol{F}^{-1T} \hat{\boldsymbol{Ic}}_k \boldsymbol{F}^{-1} + \boldsymbol{Ic}(\boldsymbol{F}\hat{\theta}_k)\}(\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k)\\ &+ (\theta_{k+1} - \boldsymbol{F}\hat{\theta}_k)^T \boldsymbol{S}(\boldsymbol{y}_{k+1};\, \boldsymbol{F}\hat{\theta}_k). \qquad (49)\end{aligned}$$

The maximizing parameters of (49) are given by (8). Using the notation (9), we obtain

$$\begin{aligned}&L_{k+1}(\theta_{k+1})\\ &\quad = c' - \tfrac{1}{2}\{\theta_{k+1} - \hat{\theta}_{k+1} + \hat{\boldsymbol{Ic}}^{-1}_{k+1} S(\boldsymbol{y}_{k+1},\, \boldsymbol{F}\hat{\theta}_k)\}^T\\ &\qquad \cdot \hat{\boldsymbol{Ic}}_{k+1}\{\theta_{k+1} - \hat{\theta}_{k+1} + \hat{\boldsymbol{Ic}}^{-1}_{k+1} \boldsymbol{S}(\boldsymbol{y}_{k+1},\, F\hat{\theta}_k)\}\\ &\qquad \cdot (\theta_{k+1} - \hat{\theta}_{k+1} + \hat{\boldsymbol{Ic}}^{-1}_{k+1} \boldsymbol{S}(\boldsymbol{y}_{k+1},\, \boldsymbol{F}\hat{\theta}_k))^T\\ &\qquad \cdot \boldsymbol{S}(\boldsymbol{y}_{k+1};\, \boldsymbol{F}\hat{\theta}_k)\end{aligned}$$

which can be written

$$L_{k+1}(\theta_{k+1}) = c'' - \tfrac{1}{2}(\theta_{k+1} - \hat{\theta}_{k+1})^T \hat{\boldsymbol{Ic}}_{k+1}(\theta_{k+1} - \hat{\theta}_{k+1})$$

where c' is independent of θ_{k+1}. Thus, the parameters in (8) maximize $L_{k+1}(\theta_{k+1})$ (M-step), and the assumption (47) is validated. This completes the recursion.

Appendix B
M-Step of the EM–Kalman Algorithm with Time-Delayed Signals

Using the time-delayed signal definition (28), the M-step of the EM–Kalman (23) becomes

$$\begin{aligned}\tilde{\theta}_k^{(n)} = \underset{\tilde{\theta}_k^{(n)}}{\text{argmin}} \int_T &\left[\hat{\boldsymbol{x}}_k^{(n)}(t) - \alpha_k^{(n)} s\left(t - \tau_k^{(n)}\right)\boldsymbol{a}\left(\psi_k^{(n)}\right)\right]^*\\ &\cdot \boldsymbol{R}^{-1}\left[\hat{\boldsymbol{x}}_k^{(n)}(t) - \alpha_k^{(n)} s\left(t - \tau_k^{(n)}\right)\boldsymbol{a}\left(\psi_k^{(n)}\right)\right].\end{aligned}$$

Assume that the observation noise vector is not correlated (this is usually the case for a sensor's array, where each element in the vector is related to a different sensor). Denoting

$$\boldsymbol{R}_k = \begin{bmatrix} r_{k,\,0} & & 0 \\ & O & \\ 0 & & r_{k,\,M} \end{bmatrix}$$

$$\hat{\boldsymbol{x}}_k^{(n)}(t) = \begin{bmatrix} \hat{x}_{k,\,1}^{(n)}(t) \\ \hat{x}_{k,\,2}^{(n)}(t) \\ \vdots \\ \hat{x}_{k,\,M}^{(n)}(t) \end{bmatrix}, \quad \boldsymbol{a}\left(\psi_k^{(n)}\right) = \begin{bmatrix} a_1\left(\psi_k^{(n)}\right) \\ a_2\left(\psi_k^{(n)}\right) \\ M \\ a_M\left(\psi_k^{(n)}\right) \end{bmatrix}$$

we obtain

$$\begin{aligned}\tilde{\theta}_k^{(n)} &= \underset{\tilde{\theta}_k^{(n)}}{\mathbf{argmin}} \sum_{m=1}^{M} \int_T r_{k,\,m}^{-1}\\ &\quad \cdot \left[\hat{x}_{k,\,m}^{(n)}(t) - \alpha_k^{(n)} s\left(t - \tau_k^{(n)}\right) a_m\left(\psi_k^{(n)}\right)\right]^*\\ &\quad \cdot \left[\hat{x}_{k,\,m}^{(n)}(t) - \alpha_n^{(n)} s\left(t - \tau_k^{(n)}\right) a_m\left(\psi_k^{(n)}\right)\right]\\ \tilde{\theta}_k^{(n)} &= \underset{\tilde{\theta}_k^{(n)}}{\mathbf{argmax}} \sum_{m=1}^{M} \int_T r_{k,\,m}^{-1}\\ &\quad \cdot \left\{2\,\text{Real}\left[\hat{x}_{k,\,m}^{(n)}(t)\alpha_k^{(n)^*} s^*\left(t - \tau_k^{(n)}\right) a_m^*\left(\psi_k^{(n)}\right)\right]\right.\\ &\qquad \left. - \left|\alpha_k^{(n)}\right|^2 \left|s^*\left(t - \tau_k^{(n)}\right)\right|^2 \left|a_m\left(\psi_k^{(n)}\right)\right|^2\right\}.\end{aligned}$$

For signals $s(t - \tau_k^{(n)})$ that are fully inside the integration interval T, we obtain

$$\begin{aligned}\tilde{\theta}_k^{(n)} = \underset{\tau_k^{(n)},\, \psi_k^{(n)}}{\text{argmax}} &\left\{\mathbf{2}\left|\alpha_k^{(n)}\right| \mathbf{Real}\left[e^{j\,\mathbf{arg}(\alpha_n^{(n)})} \sum_{m=1}^{M}\right.\right.\\ &\quad \left. \cdot\, a_m^*\left(\psi_k^{(n)}\right) r_{k,\,m}^{-1} \int_T \hat{x}_{k,\,m}^{(n)}(t) s^*\left(t - \tau_k^{(n)}\right)\right]\\ &\quad \left. - \left|\alpha_k^{(n)}\right|^2 Es \sum_{m=1}^{M} r_{k,\,m}^{-1} \left|a_m\left(\psi_k^{(n)}\right)\right|^2\right\}. \qquad (50)\end{aligned}$$

Let $\hat{x}_{\text{MF}_{k,\,m}}^{(n)}(\tau_k^{(n)})$ be the m's element of the complete data vector after a filter matched to the signal $s(t)$

$$\hat{x}_{\text{MF}_{k,\,m}}^{(n)}\left(\tau_k^{(n)}\right) = \int_T \hat{x}_{k,\,m}^{(n)}(t) s^*\left(t - \tau_k^{(n)}\right)$$

and define

$$g\left(\tau_k^{(n)},\, \psi_n^{(n)}\right) = \sum_{m=1}^{M} a_m^*\left(\psi_n^{(n)}\right)\hat{x}^{(n)}_{\mathrm{MF}_{k,\,m}}\left(\tau_k^{(n)}\right) r_{k,\,m}^{-1}.$$

Then, maximization in (50) is equivalent to

$$\begin{aligned}\tilde{\theta}_k^{(n)} &= \mathop{\mathbf{argmax}}_{t_k^{(n)},\, \psi_k^{(n)}} \left| g\left(\tau_k^{(n)},\, \psi_k^{(n)}\right)\right| \\ \arg\left(\alpha_n^{(n)}\right) &= -\arg\left(g\left(\tilde{\theta}_k^{(n)}\right)\right) \\ \left|\alpha_n^{(n)}\right| &= \frac{\left|g\left(\tilde{\theta}_k^{(n)}\right)\right| \Big/ Es}{\displaystyle\sum_{m=1}^{M} r_{k,\,m}^{-1}\left|a_m\left(\psi_k^{(n)}\right)\right|^2}.\end{aligned}$$

Acknowledgment

The authors wish to thank J. Jaffe, Scripps Institute, for providing the FTV data and for interesting discussions. They also wish to thank A. Genin, Hebrew University, for introducing to them the problem of biological underwater tracking.

References

[1] E. Weinstein, M. Feder, and A. V. Oppenheim, "Sequential algorithms for parameter estimation based on the Kullbak–Leibler information matrix," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 38, pp. 1652–1654, Sept. 1990.

[2] D. M. Titterington, "Recursive parameter estimation using incomplete data," *J. Roy. Stat. Soc. B,* vol. 46, pp. 256–267, 1984.

[3] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Trans. Signal Processing,* vol. 36, pp. 477–489, Apr. 1988.

[4] J. S. Jaffe, E. Reuss, D. McGehee, and G. Chandran, "FTV, a sonar for tracking zooplankton in three-dimensions," Marine Phys. Lab, Scripps Inst. Oceanogr., La Jolla, CA, 1993.

[5] L. R. Rabiner, "A tutorial on hidden Markov model and selected applications in speech recognition," *Proc. IEEE,* vol. 77, pp. 257–286, Feb. 1989.

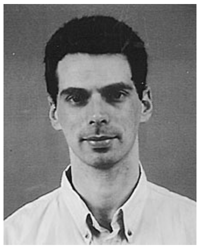
[6] R. L. Streit and R. F. Barrett, "Frequency line tracking using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 38, pp. 586–598, Apr. 1990.

[7] X. Xie and R. J. Evans, "Multiple target tracking and multiple frequency line tracking using hidden Markov models," *IEEE Trans. Signal Processing,* vol. 39, pp. 2659–2676, Dec. 1991.

[8] S. S. Blackman, *Multiple-Target Tracking with Radar Applications.* Norwood, MA: Artech House, 1986.

[9] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification.* Cambridge, MA: MIT Press, 1987.

[10] A. P. Demster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete via the EM algorithm," *J. Roy. Stat. Soc.,* ser. 39, pp. 1–38, 1977.



Liron Frenkel received the B.Sc. and M.Sc. degrees in electrical engineering from Tel-Aviv University, Tel-Aviv, Israel, in 1991 and 1995, respectively.

He joined Orckit Communications, Tel-Aviv, in 1994. His current interests are development of advanced line codes for cable modems and communication over digital subscriber lines.

Meir Feder (SM'93–F'99) received the B.Sc. and M.Sc. degrees from Tel-Aviv University, Tel-Aviv, Israel, and the Sc.D. degree from the Massachusetts Institute of Technology (MIT), Cambridge, and the Woods Hole Oceanographic Institution, Woods Hole, MA, all in electrical engineering, in 1980, 1984, and 1987, respectively.

After being a Research Associate and Lecturer at MIT, he joined the Department of Electrical Engineering—Systems, Tel-Aviv University, in 1989. He also had visiting appointments at Woods Hole Oceanographic Institution, Scripps Institute, and Bell Laboratories. From 1995 to 1996, he spent a sabbatical year as a Visiting Professor at the Electrical Engineering and Computer Science Department, MIT.

Between June 1993 and June 1996, he served as an Associate Editor for Source Coding of the IEEE Transactions on Information Theory. He received the 1993 Information Theory Best Paper Award, the 1994 Tel-Aviv University Prize for Excellent Young Scientists, the 1995 Research Prize of the Israeli Electronic Industry, and in October 1995, he received the Research Prize in Applied Electronics from the Ex-Serviceman Association, London, which was awarded by Ben-Gurion University.