X. Rong Li*
University of New Orleans
New Orleans, LA 70148
Phone: 504-280-7416, Fax: 504-280-3950
xrlee@uno.edu

Jean Dezert
ONERA (DTIM/MCT)
29 Avenue de la Division Leclerc
92320 Châtillon, France
E-mail: dezert@onera.fr

ABSTRACT

The model set used in a multiple-model (MM) algorithm in many practical situations has a layered structure. This paper presents a novel and highly efficient MM algorithm, called Layered MM (LMM) algorithm, which takes full advantage of the layered structure. Compared with the standard MM algorithm, it is expected that the LMM algorithm will have a substantial saving in computation and similar performance. The need for such a layered structure and layered algorithm is motivated by the problem of tracking maneuvering and bending extended targets in a cluttered environment. The basic idea of the LMM algorithm is to incorporate the coupling between the layers using time-varying transition probabilities of the models. In conjunction with a fast point-pattern matching algorithm and a modified conditional matching pairs support finding algorithm proposed recently, the LMM algorithm is applied to solve the problem of tracking maneuvering and bending extended targets in a cluttered environment.

1. INTRODUCTION

Most modern tracking algorithms described in the literature (see e.g.,[1–3]) deal mainly with *point targets* where a target is represented by its center of mass (COM) and the tracking systems track the COMs of targets. Such point-target tracking is sufficient for many applications. It is also the only viable solution in many situations because of the limited resolution of sensors. The dynamic models involved in such algorithms are therefore only relative to the evolution of the COM of targets expressed in a relative (platform) or an absolute frame. Since the attitude of the target is not observable using point-target modeling, it cannot be taken into account in classical trackers even though it could provide very useful information for other functions (e.g., threat assessment) of a defense system.

With the advancement of sensor technology and signal processing (for example, the development of advanced super-resolution algorithms for radar signals), the above point-target assumption is losing its validity in many situations. For the next generation of tracking systems, targets should be considered as extended targets having high maneuvering and intelligent hiding abilities rather than as point targets. We will refer to such non-point targets as bending extended targets. To the best knowledge of the authors, few algorithms for bending extended target tracking are available in the literature. These algorithms are based on digital image processing techniques, which assume the availability of the target images and thus require (high resolution) imaging sensors and have a heavy computational load. Point targets and imaging targets are two extreme cases. Many practical situations lie in between. Many sensors provide extended target observations that are not rich enough to form images. In such cases, we cannot track extended targets directly by existing image processing approaches.

The first algorithm for maneuvering and bending extended target tracking (MBETT) in a cluttered environment was proposed in.[9] It is, however, computationally intense even for the simple design of the model set. We propose in this paper an improved algorithm for MBETT problem in a cluttered environment based on a novel multiple-model (MM) estimation algorithm combined with an improved point pattern matching algorithm developed recently to tackle the data association problem of tracking in clutter. The proposed MM algorithm is highly efficient. Its novelty lies in that it takes full advantage of the layered structure of the problem: The total set of models is decomposed into layers (subsets) of models and the standard MM algorithm of running a large bank of filters is decomposed into the running of several layers of much smaller banks of filters with the coupling among layers taken into account in the transition probabilities. As a result, a substantial reduction in computation is achieved, which is crucial for many practical problems, in particular the MBETT problem. The proposed LMM algorithm does not need the fundamental assumption of[8] that the layers are independent, which is seldom valid in reality.

2. PROBLEM OF TRACKING MANEUVERING AND BENDING EXTENDED TARGETS

To simplify analysis, we consider only the mono-sensor case with only one maneuvering and bending extended target moving in a 2D cluttered space with a detection probability that is less than one.

We model the evolution and the observation of an extended target by the following stochastic hybrid sys-

*Supported in part by ONR via grant N00014-97-1-0570, NSF via Grants ECS-9496319 and ECS-9734285, LEQSF via Grant (1996–99)-RD-A-32.

$$x_{k+1} = \begin{bmatrix} f_k^g(x_k^g, x_k^p, s_{k+1}) \\ f_k^p(x_k^g, x_k^p, s_{k+1}) \end{bmatrix} + \begin{bmatrix} w_k^g(s_{k+1}) \\ w_k^p(s_{k+1}) \end{bmatrix} \quad (1)$$

$$z_k = h_k(x_k^g, x_k^p, s_k) + v_k(s_k) \quad (2)$$

where $x_k = [(x_k^g)', (x_k^p)']'$ is the **global** base state vector of the extended target; x_k^g represents the **principal** state of the target relative to its center of mass (COM) (x_k^g is the conventional state used in a classical description of the dynamic system); x_k^p will be referred to as the **proper** state of the target and its evolution describes the proper (angular) motion of the target around its COM; $s_k = [s_k^g, s_k^p, s_k^s]'$ is the discrete-valued **modal state vector** of the system at time k; s_k^g and s_k^p denote respectively the *principal* and *proper* modes in effect during the sampling period k; s_k^s denotes the **shape-mode** in effect during the sampling period k, which describes the evolution of the point patterns of the extended target. The sequence of system modes is assumed a first-order homogeneous Markov chain with *known* transition probabilities

$$P\Big\{s_{k+1} = \{m_j^g, m_n^p, m_s^s\} | s_k = \{m_i^g, m_m^p, m_r^s\}\Big\} = \pi_{i,j,m,n,r,s} \quad (3)$$

where m_j^g, m_n^p, m_s^s and m_i^g, m_m^p, m_r^s are the particular values of the the components of the modal state vector s at time $k+1$ and k, respectively. The system mode sequence s_k is then an indirectly observed (or hidden) Markov chain. The vector-valued functions f and h are assumed known; $w_k(s_{k+1}) = [w_k^g(s_{k+1})', w_k^p(s_{k+1})']'$ is the mode-dependent process noise with mean $\bar{w}_k(s_{k+1})$ and covariance $Q_k(s_{k+1})$; the measurement vector z_k is the stacked vector of n_k reflection points $z_i(k)$, $(i = 1, \ldots, n_k)$ of the extended target. Since the probability of detection of the target can be less than one and the shape of the target can change during tracking, the number n_k of reflection points coming from the the extended target is a discrete random variable. Therefore, measurement vector z_k described by (2) is actually of a **random dimension**. Eq. (2) implies that the global base state observations are mode-dependent and the mode information is embedded in the measurement sequence.[10] v_k is the stacked vector $[(v_k^1)', \ldots, (v_k^{n_k})']'$ of measurement noise $v_k^i(s_k)$ with mean $\bar{v}_k^i(s_k)$ and covariance $R_k^i(s_k)$. It is assumed that v, w, uncorrelated with x_0, are mutually uncorrelated; x_0 is assumed to have mean $x_{0|0}$ and covariance $P_{0|0}$; $h_k(s_k)$ is the known — when data association is solved (see later) — stacked matrix $[h_k^1(s_k)', \ldots, h_k^{n_k}(s_k)']'$. All vectors and matrices are assumed to have appropriate dimensions.

The problem we are facing is to estimate the global base state and the hidden modal state of the hybrid system with all available information (including prior information and past and present observations $Z^k = \{z_1, \ldots, z_k\}$).

Note that the measurement of the principal and proper state, as described in (2), may not be separated in some measurement of the principal state is simply the COM measurement, which can be easily obtained from the n_k reflection points (after point pattern matching). In such a case, we have

$$z_k = \begin{bmatrix} z_k^g \\ z_k^p \end{bmatrix} = \begin{bmatrix} h_k^g(x_k^g, x_k^p, s_k) \\ h_k^p(x_k^g, x_k^p, s_k) \end{bmatrix} + \begin{bmatrix} v_k^g(s_k) \\ v_k^p(s_k) \end{bmatrix} \quad (4)$$

3. LAYERED MULTIPLE-MODEL ALGORITHM

It is clear from the above formulation of tracking a maneuvering and bending extended target that there exist three natural layers in the system modes: (a) the principal mode s_g, (b) the proper mode s_p, and (c) the shape mode s_s. Suppose that the mode spaces of s_g, s_p, and s_s are covered by r_g, r_p and r_s models, respectively. Then a standard MM algorithm would have to run a bank of filters based on $r_g r_p r_s$ models in parallel at each time, which could have an infeasible computational burden. We propose in this work a structured MM algorithm, called **Layered MM (LMM) algorithm**, that needs to use only $r_g + r_p + r_s$ models by taking advantage of the layered structure of the system modes using time-varying transition probabilities of the models.

For simplicity of presentation, assume as in[9] that the the space of the principal mode s_g can be covered effectively by two models: a nearly constant velocity (CV) model and a nearly coordinated turn (CT) model; the space of the proper mode s_p can be covered effectively by two models: a nearly null velocity (NV) model and a nearly constant velocity (CV) model; the space of the shape mode s_s can be covered effectively by two models: a 4-point pattern model and a 5-point pattern model. This is represented concisely as $s_g \in \{CV, CT\}, s_p \in \{NV, CV\}, s_s \in \{4, 5\}$.

A standard MM algorithm for the above global mode set is to use 8 models: $\{CV, NV, 4\}$, $\{CV, NV, 5\}$, $\{CV, CV, 4\}$, $\{CV, CV, 5\}$, $\{CT, NV, 4\}$, $\{CT, NV, 5\}$, $\{CT, CV, 4\}$, $\{CT, CV, 5\}$. The layered MM uses three 2-model MM estimators separately: CV and CT for s_g, NV and CV for s_p, and 4- and 5-point patterns for s_s. Note that in practice two models for each layer is usually not sufficient. If three models were used for each layer then the standard MM algorithm would require the use of 27 models and the layered MM algorithm would only require the use of 9 models. Clearly, the computational savings of the layered MM algorithm over the standard MM algorithm will be tremendous if more models are used in each layer.

Note that when model set are decomposed into layers,

$$
\begin{aligned}
P\Big\{s^g_{k+1} &= m^g_j | s^g_k = m^g_i, s^p_k, s^s_k, s^p_{k+1}, s^s_{k+1}\Big\} \\
&= \pi_{i,j}(s^p_k, s^s_k, s^p_{k+1}, s^s_{k+1}) \\
P\Big\{s^p_{k+1} &= m^g_n | s^p_k = m^p_m, s^g_k, s^s_k, s^g_{k+1}, s^s_{k+1}\Big\} \\
&= \pi_{m,n}(s^g_k, s^s_k, s^g_{k+1}, s^s_{k+1}) \\
P\Big\{s^s_{k+1} &= m^s_s | s^s_k = m^s_r, s^p_k, s^g_k, s^p_{k+1}, s^g_{k+1}\Big\} \\
&= \pi_{r,s}(s^p_k, s^g_k, s^p_{k+1}, s^g_{k+1})
\end{aligned} \tag{5}
$$

which is more general and precise than the following

$$
\begin{aligned}
P\Big\{s^g_{k+1} = m^g_j | s^g_k = m^g_i, s^p_k, s^s_k\Big\} &= \pi_{i,j}(s^p_k, s^s_k) \\
P\Big\{s^p_{k+1} = m^g_n | s^p_k = m^p_m, s^g_k, s^s_k\Big\} &= \pi_{m,n}(s^g_k, s^s_k) \\
P\Big\{s^s_{k+1} = m^s_s | s^s_k = m^s_r, s^p_k, s^g_k\Big\} &= \pi_{r,s}(s^p_k, s^g_k)
\end{aligned} \tag{6}
$$

(5)–(6) indicate that the probability of a mode transition within one layer depends in general on the modes in effect in other layers before (and after) the transition. The basic idea of LMM is to account for the coupling between different layers in the mode transition probabilities.

For the MBETT problem, it can be observed that the principal mode s_g and the proper mode s_p are coupled because e.g., nearly coordinated turn motion of the principal mode is more often accompanied by a nearly constant angular motion in the proper mode and thus mode $\{s_g, s_p\} = \{CV, NV\}$ is much more likely than mode $\{CV, CV\}$ or $\{CT, NV\}$ and similarly, mode $\{CT, CV\}$ is more likely than mode $\{CT, NV\}$ or $\{CV, CV\}$. In other words, the fundamental assumption of that the layers are independent is seldom valid in reality. Also, the computational advantage of the algorithm proposed in[8] is questionable since the computational load of the standard MM algorithm can also be easily reduced by say a simple logic under the independence assumption.

3.1. The LMM Algorithm for MBETT Problem

We first illustrate our LMM algorithm for the simplified MBETT problem of two layers with $s_g \in \{CV, CT\}$ and $s_p \in \{NV, CV\}$. The standard MM algorithm would use four models $\{CV, NV\}$, $\{CV, CV\}$, $\{CT, NV\}$, $\{CT, CV\}$. The LMM operates two 2-model layers based on the model sets $\{CV, CT\}$ for s_g and $\{NV, CV\}$ for s_p. If s_g and s_p are independent, then we can run the two 2-model layers independently. Otherwise, we have to run them in a fashion that accounts for the coupling between the modes in different layers. This coupling can be accounted for using coupled transition probabilities. This is the basic idea of the LMM algorithm.

Since the evolution of the COM is affected to a lesser degree by the proper motion than the dependence of the proper motion on the principal mode, an implementation of the LMM for the simplified MBETT problem consists of two step at each time :

1. Run a 2-model MM algorithm for the principal mode s_g (s_g and s_p were independent but the transition probabilities of the principal mode s_g are time varying and are given by, by total probability theorem, for $m_i, m_j \in \{CV, CT\}$,

$$
\begin{aligned}
&P\{s^g_k = m_j | s^g_{k-1} = m_i, Z^{k-1}\} \\
&= \sum_{m_n \in \{NV,CV\}} P\{s^g_k = m_j | s^g_{k-1} = m_i, s^p_{k-1} = m_n, \\
&\quad Z^{k-1}\} P\{s^p_{k-1} = m_n | Z^{k-1}\} \\
&= \sum_{m_n \in \{NV,CV\}} P\{s^g_k = m_j | s^g_{k-1} = m_i, s^p_{k-1} = m_n\} \\
&\quad \times P\{s^p_{k-1} = m_n | Z^{k-1}\}
\end{aligned} \tag{7}
$$

where the proper mode probabilities $P\{s^p_{k-1} = m_n | Z^{k-1}\}$ were obtained from the previous recursive time cycle $k-1$.

Note that this equation reveals that the transition probability $P\{s^g_k = m_j | s^g_{k-1} = m_i, Z^{k-1}\}$ is a time-varying (more precisely, data-dependent) probabilistic weighted sum (average) of the time-invariant (more precisely, data-independent) but proper-mode-dependent transition probabilities $P\{s^g_k = m_j | s^g_{k-1} = m_i, s^p_{k-1} = m_n\}$, which are design parameters. If the evolution of the principal mode does not depend on the proper mode, then, for all $m_n \in \{NV, CV\}$,

$$
\begin{aligned}
P\{s^g_k = m_j | s^g_{k-1} = m_i, s^p_{k-1} = m_n\} &\quad (8) \\
= P\{s^g_k = m_j | s^g_{k-1} = m_i\} &\quad (9)
\end{aligned}
$$

In this case, step 1 is identical to a standard MM algorithm using model set $\{CV, CT\}$ that is independent of the proper mode.

2. Run a 2-model MM algorithm for the proper mode s_p (s_g and s_p were independent but the transition probabilities of the proper mode s_p are time varying and are given by, by total probability theorem, for $m_m, m_n \in \{NV, CV\}$,

$$
\begin{aligned}
&P\{s^p_k = m_m | s^p_{k-1} = m_n, Z^{k-1}\} \\
&= \sum_{i,j} P\{s^p_k = m_n | s^p_{k-1} = m_m, s^g_k = m_j, \\
&\quad s^g_{k-1} = m_i, Z^{k-1}\} P\{s^g_k = m_j, s^g_{k-1} = m_i | Z^{k-1}\} \\
&= \sum_{i,j} P\{s^p_k = m_n | s^p_{k-1} = m_m, s^g_k = m_j, \\
&\quad s^g_{k-1} = m_i\} P\{s^g_k = m_j | s^g_{k-1} = m_i, Z^{k-1}\} \\
&\quad \times P\{s^g_{k-1} = m_i | Z^{k-1}\}
\end{aligned} \tag{10}
$$

where the summation is over all possible combinations of m_i and m_j such that they are both in the model set $\{CV, CT\}$. The mode-dependent (but

$m_j|s^p_{k-1} = m_i, s^g_k = m_m, s^g_{k-1} = m_n\}$ are design parameters. The probabilities $P\{s^g_k = m_m|s^g_{k-1} = m_n, Z^{k-1}\}$, given by (7), and $P\{s^g_{k-1} = m_n|Z^{k-1}\}$ were obtained from step 1 before Z_k is received. In some situations, the transition probabilities $P\{s^p_k = m_j|s^p_{k-1} = m_i, s^g_k = m_m, s^g_{k-1} = m_n\}$ does not depend on the current principal mode and thus, for $m_m, m_n \in \{NV, CV\}$,

$$\begin{aligned} & P\{s^p_k = m_m|s^p_{k-1} = m_n, Z^{k-1}\} \\ = & \sum_{m_i \in \{CV,CT\}} P\{s^p_k = m_n|s^p_{k-1} = m_m, \\ & s^g_{k-1} = m_i, Z^{k-1}\} P\{s^g_{k-1} = m_i|Z^{k-1}\} \end{aligned} \quad (11)$$

In such a case, step 1 and step 2 can be interchanged without difference.

To improve accuracy, the two steps can be iterated with (7) replaced by the following equation except for the first time step 1 is executed, for $m_i, m_j \in \{CV, CT\}$,

$$\begin{aligned} & P\{s^g_k = m_j|s^g_{k-1} = m_i, Z^{k-1}\} \\ = & \sum_{m,n} P\{s^g_k = m_j|s^g_{k-1} = m_i, s^p_k = m_n, s^p_{k-1} = m_m, \\ & Z^{k-1}\} P\{s^p_k = m_n, s^p_{k-1} = m_m|Z^{k-1}\} \\ = & \sum_{m,n} P\{s^g_k = m_j|s^g_{k-1} = m_i, s^p_k = m_n, s^p_{k-1} = m_m\} \\ & P\{s^p_k = m_n|s^p_{k-1} = m_m, Z^{k-1}\} P\{s^p_{k-1} = m_m|Z^{k-1}\} \end{aligned} \quad (12)$$

Note that the above LMM algorithm is valid provided that the mode-dependent transition probabilities are really data-independent. This assumption is usually valid and convenient for practical application. Otherwise, there is no easy and general way to determine these transition probabilities.

There are two possible couplings between the layers: (a) between the principal and proper modes and (b) between the principal and proper states. The above deals with the coupling between modes. measurements are The coupling in states poses no problem for the LMM algorithm. In fact, the LMM algorithm can take care of this coupling better than the standard MM algorithm for a nonlinear system. If the measurements of the principal and proper states can be separated as in (4), use the following for step 1:

$$\begin{aligned} \hat{x}^g_{k|k-1} = & f^g_{k-1}(\hat{x}^g_{k-1|k-1}, \hat{x}^p_{k-1|k-1}, \hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \\ & + \bar{w}^g_{k-1}(\hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \end{aligned} \quad (13)$$

$$\begin{aligned} \hat{z}^g_{k|k-1} = & h^g_k(\hat{x}^g_{k|k-1}, \hat{x}^p_{k|k-1}, \hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \\ & + \bar{v}^g_k(\hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \end{aligned} \quad (14)$$

$$\begin{aligned} \hat{x}^p_{k|k-1} = & f^p_{k-1}(\hat{x}^g_{k-1|k}, \hat{x}^p_{k-1|k-1}, \hat{s}^g_{k|k}, \hat{s}^p_{k|k-1}) \\ & + \bar{w}^p_{k-1}(\hat{s}^g_{k|k}, \hat{s}^p_{k|k-1}) \end{aligned} \quad (15)$$

$$\begin{aligned} \hat{z}^p_{k|k-1} = & h^p_k(\hat{x}^g_{k|k}, \hat{x}^p_{k|k-1}, \hat{s}^g_{k|k}, \hat{s}^p_{k|k-1}) + \bar{v}^p_k(\hat{s}^g_{k|k}, \hat{s}^p_{k|k-1}) \end{aligned} \quad (16)$$

If the measurements of the principal and proper states cannot be separated, the measurement prediction should be replaced in step 1 by

$$\begin{aligned} \hat{z}_{k|k-1} = & h_k(\hat{x}^g_{k|k-1}, \hat{x}^p_{k|k-1}, \hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \\ & + \bar{v}_k(\hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \end{aligned}$$

and in step 2 by

$$\hat{z}_{k|k-1} = h_k(\hat{x}^g_{k|k}, \hat{x}^p_{k|k-1}, \hat{s}^g_{k|k}, \hat{s}^p_{k|k-1}) + \bar{v}_k(\hat{s}^g_{k|k}, \hat{s}^p_{k|k-1})$$

In the above, $\hat{x}^g_{k-1|k}$ is one-step fixed-lag smoothed estimate of x^g_{k-1}. If an extended Kalman filter (EKF) is used for nonlinear dynamics and/or measurements, the Jacobian (and/or Hessian) of f^p and h^p in the k-th time cycle should be evaluated at $(\hat{x}^g_{k-1|k}, \hat{x}^p_{k-1|k-1})$ and $(\hat{x}^g_{k|k}, \hat{x}^p_{k|k-1})$, respectively (although the Jacobian (and/or Hessian) of f^g and h^g are the same as in the standard MM algorithm based on EKF); for example,

$$F^p_k = \frac{\partial}{\partial x^p_k} f^p_k(\hat{x}^g_{k|k+1}, x^p_k, \hat{s}^g_{k+1|k+1}, \hat{s}^p_{k+1|k})|_{x^p_k = \hat{x}^p_{k|k}} \quad (17)$$

$$H^p_k = \frac{\partial}{\partial x^p_k} h^p_k(\hat{x}^g_{k|k}, x^p_k, \hat{s}^g_{k|k}, \hat{s}^p_{k|k-1})|_{x^p_k = \hat{x}^p_{k|k-1}} \quad (18)$$

Note that the standard MM algorithm based on EKF would use, if the measurements can be separated (similarly for the case where measurements cannot be separated)

$$\begin{aligned} \begin{bmatrix} \hat{x}^g_{k|k-1} \\ \hat{x}^p_{k|k-1} \end{bmatrix} = & \begin{bmatrix} f^g_{k-1}(\hat{x}^g_{k-1|k-1}, \hat{x}^p_{k-1|k-1}, \hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \\ f^p_{k-1}(\hat{x}^g_{k-1|k-1}, \hat{x}^p_{k-1|k-1}, \hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \end{bmatrix} \\ & + \begin{bmatrix} \bar{w}^g_{k-1}(\hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \\ \bar{w}^p_{k-1}(\hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \end{bmatrix} \end{aligned} \quad (19)$$

$$\begin{aligned} \begin{bmatrix} \hat{z}^g_{k|k-1} \\ \hat{z}^p_{k|k-1} \end{bmatrix} = & \begin{bmatrix} h^g_k(\hat{x}^g_{k|k-1}, \hat{x}^p_{k|k-1}, \hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \\ h^p_k(\hat{x}^g_{k|k-1}, \hat{x}^p_{k|k-1}, \hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \end{bmatrix} \\ & + \begin{bmatrix} \bar{v}^g_k(\hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \\ \bar{v}^p_k(\hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1}) \end{bmatrix} \end{aligned} \quad (20)$$

and the Jacobian (and/or Hessian) of f and h in the k-th time cycle should be evaluated at $\hat{x}_{k-1|k-1}$ and $\hat{x}_{k|k-1}$, respectively; for example,

$$F^p_k = \frac{\partial}{\partial x^p_k} f^p_k(\hat{x}^g_{k|k}, x^p_k, \hat{s}^g_{k+1|k}, \hat{s}^p_{k+1|k})|_{x^p_k = \hat{x}^p_{k|k}} \quad (21)$$

$$H^p_k = \frac{\partial}{\partial x^p_k} h^p_k(\hat{x}^g_{k|k-1}, x^p_{k-1}, \hat{s}^g_{k|k-1}, \hat{s}^p_{k|k-1})|_{x^p_k = \hat{x}^p_{k|k-1}} \quad (22)$$

smoothing estimate $\hat{x}^g_{k-1|k}$ is used in the LMM algorithm, then one cycle of the LMM algorithm is "equivalent" to the iterated EKF for the principal state part of the standard MM algorithm.

3.2. The LMM Algorithm in General Case

The above algorithm for the simplified MBETT problem can be extended to the general case without difficulty. As a result, the application of the LMM algorithm consists, in general, of the following conceptual steps:

S1. Identify the layers in the mode space.

S2. If there are layers that are (almost) independent of the other layers, then the original mode space decomposes naturally into several independent subspaces of the mode, where the modes in each independent layer form a subspace. As a result, the standard MM algorithm with a model set that covers the original mode space is equivalent to the group of (layered) MM algorithms, each with the model set that covers an independent layer.

S3. Order the coupled layers such that the layer that is least dependent on the other layers is at the top and the layer that is most dependent on the others is at the bottom.

S4. Run an MM algorithm for the top layer first (whose mode is denoted as s^1) as if the other layers were non-existent. This MM algorithm is obtained by replacing the transition probabilities in the standard MM algorithm that uses the models for the top layer as follows (where every quantity is conditioned on Z^{k-1}, which is dropped for simplicity)

$$\begin{aligned} &P\{s^1_k = m_j | s^1_{k-1} = m_i\} \\ &= \sum_{\text{all } M_n^{N-1}} P\{s^1_k = m_j | s^1_{k-1} = m_i, \\ &\quad S^{N-1}_{k-1} = M^{N-1}_n, Z^{k-1}\} P\{S^{N-1}_{k-1} = M^{N-1}_n\} \quad (23) \end{aligned}$$

where $S^{N-1}_{k-1} = M^{N-1}_n$ is a shorthand notation for $\{s^2_{k-1} = m^2_{n_2}, \ldots, s^N_{k-1} = m^N_{n_N}\}$ and the summation is over all possible $M^{N-1}_n = \{m^2_{n_2}, \ldots, m^N_{n_N}\}$; that is, over all possible combinations of the models used to cover layers 2 through N. Note that the mode probabilities

$$\begin{aligned} &P\{S^{N-1}_{k-1} = M^{N-1}_n\} \\ &\qquad = P\{s^2_{k-1} = m^2_{n_2}, \ldots, s^N_{k-1} = m^N_{n_N}\} \end{aligned}$$

were obtained from the previous recursive time cycle and the mode-dependent transition probabilities $P\{s^1_k = m_j | s^1_{k-1} = m_i, S^{N-1}_{k-1} = M^{N-1}_n\}$ are design parameters.

S5. Run an MM algorithm for the r-th layer as if the other layers were non-existent. This MM algorithm in the standard MM algorithm that uses the models for the r-th layer as follows

$$\begin{aligned} &P\{s^r_k = m_j | s^r_{k-1} = m_i\} \\ &= \sum P\{s^r_k = m_j | s^r_{k-1} = m_i, S^{N-r}_{k-1} = M^{N-r}_n, S^{r-1}_k \\ &\quad = M^{r-1}_m, \} P\{S^{r-1}_k = M^{r-1}_m, S^{N-r}_{k-1} = M^{N-r}_n\} (24) \end{aligned}$$

where the following shorthand notations were used

$$\begin{aligned} S^{r-1} &= \{s^1, s^2, \ldots, s^{r-1}\} \\ S^{N-r} &= \{s^1, \ldots, s^{r-1}, s^{r+1}, \ldots, s^N\} \\ M^{r-1}_m &= \{m^1_{m_1}, m^2_{m_2}, \ldots, m^{r-1}_{m_{r-1}}\} \\ M^{N-r}_n &= \{m^1_{n_1}, \ldots, m^{r-1}_{n_{r-1}}, m^{r+1}_{n_{r+1}}, \ldots, m^N_{n_N}\} \end{aligned}$$

and the summation is over all possible combinations of M^{r-1}_m and M^{N-r}_n; that is, over all possible combinations of the models used to cover the other layers. Note that the mode probabilities $P\{S^{r-1}_k = M^{r-1}_m, S^{N-r}_{k-1} = M^{N-r}_n | Z^{k-1}\}$ were obtained from the previous recursive time cycle and the higher layers in the same time cycle without using Z_k. The mode-dependent transition probabilities $P\{s^r_k = m_j | s^r_{k-1} = m_i, S^{r-1}_k = M^{r-1}_m, S^{N-r}_{k-1} = M^{N-r}_n\}$ are design parameters.

The handling of the coupling in the state components is similar to the case for the simplified MBETT problem.

4. 2D POINT PATTERN MATCHING

Point pattern matching (PPM) plays an important role in pattern recognition,[7,6] computer/stereo vision,[19,20,16] autonomous navigation,[11] and astronautics[13,14,18] and has recently been proposed for solving data association problem involving in tracking maneuvering/bending extended target (MBET).[9] The general PPM problem consists of finding an optimal (in minimum mean square error sense) matching $\mathcal{M}^*$ between an unknown subset of given point pattern (i.e., reference target) $\mathcal{P} = \{x_i, i = 1, \ldots, m\}$ and an unknown subset of point measurements set $\mathcal{Q} = \{y_j, j = 1, \ldots, n\}$.

In 2D space, every matching pairs of points $(x_i \leftrightarrow y_j)$ is related by the affine registration mapping $y_j = (T + sRx_i) + b_i$ where $T = [t_x \; t_y]'$ is an unknown translation vector, s an unknown scaling factor, R an unknown rotation matrix with angle θ and b_i is a zero-mean white Gaussian noise with known covariance P_{bb}. (R, T, s) is the set of unknown affine registration parameters of the PPM problem.

Many algorithms have been developed over the past two decades for solving such PPM problems, but very few of them are able to take simultaneously into account the five defective conditions: adding or suppressing points, location distortion, rotation, scaling and translation. Recent surveys on PPM techniques can be found in.[15,5,21,17]

fast 2D-PPM algorithm recently proposed in,[9] which was initiated by Chang et al. in.[6] A fast PPM algorithm capable of taking into account all the five above-mentioned conditions is specially required for MEBTT applications because:

- condition 1 (adding or suppressing points) reflects the ability to have a target detection probability less than unity within a cluttered environment.
- condition 2 (location distortion) reflects the ability to deal with a noisy observation sensor.
- conditions 3, 4 and 5 (rotation,scaling and translation) reflect the ability of the algorithm to carry on full affine registration uncertainty on the reference point pattern with respect to the observation point set.

When two pairs of different points (vectors) (x_i, x_h) in $\mathcal{P}$ and (y_j, y_k) in $\mathcal{Q}$ matches, (i.e., $(x_i \leftrightarrow y_j)$ and $(x_h \leftrightarrow y_k)$) under (R, T, s), then vector $x = \overrightarrow{x_i x_h}$ matches with $y = \overrightarrow{y_j y_k}$ under the reduced affine registration (R, s) since

$$\begin{aligned}\overrightarrow{y_j y_k} &= (T + sRx_h) - (T + sRx_i) + (b_h - b_i) \\ &= sR\overrightarrow{x_i x_h} + \overrightarrow{b_i b_h} = sRx + b\end{aligned}$$

It can be shown[9] that maximum likelihood estimation (MLE) of scaling factor and rotation angle for such a given pair of matching vectors is given by $\hat{\theta} = \theta_y - \theta_x$ and $\hat{s} = \|y\|/\|x\|$.

Moreover, the pair of independent variables $z = x'y$ and $z^{\perp} = (x^{\perp})'y$ contains exactly the same information as $(\hat{s}, \hat{\theta})$ ($x^{\perp}$ being the orthogonal vector of x). Assuming observation noise vector b to be Gaussian zero-mean with covariance $2\sigma^2 I$ then the random vector $z = [z\ z^{\perp}]'$ is Gaussian with mean $\bar{z} = \|x\|^2[\hat{s}\cos(\hat{\theta})\ \hat{s}\sin(\hat{\theta})]'$ and covariance $P_{zz} = 2\sigma^2\|x\|^2 I$ with I being the 2×2 identity matrix.

Let $\mathcal{X} = \{x_i, i = 1, \ldots, p\} \subset \mathcal{P}$ and $\mathcal{Y} = \{y_i, i = 1, \ldots, p\} \subset \mathcal{Q}$ be two sets of p matching points under an affine registration (R, T, s). Then for each matching pair $(x_i \leftrightarrow y_i)$, the sets of vectors $\mathcal{X}_i = \{\overrightarrow{x_i x_1}, \ldots, \vec{0}, \ldots, \overrightarrow{x_i x_p}\}$ and $\mathcal{Y}_i = \{\overrightarrow{y_i y_1}, \ldots, \vec{0}, \ldots, \overrightarrow{y_i y_p}\}$ match under the reduced affine registration (R, s). Since for $j = 1, \ldots, p$ $(j \neq i)$, vector $\overrightarrow{x_i x_j}$ matches with $\overrightarrow{y_i y_j}$, there are necessary $p - 1$ matching vectors under the reduced registration (R, s) for each matching pair of points $(x_i \leftrightarrow y_i)$. Conversely if $p \leq \min(m, n)$ is the (unknown) maximum number of matched pairs between $\mathcal{P}$ and $\mathcal{Q}$ under an unknown affine registration and if $(x_i \leftrightarrow y_i)$, then there exists $p-1$ other pairs of matching points. Hence there are exactly $p - 1$ matching vectors under the reduced registration (s, θ).

An accumulator array $M_{i,j}(\hat{s}, \hat{\theta})$ can be used to accumulate $(\hat{s}, \hat{\theta})$ determined by matching $\overrightarrow{x_i x_h}$ with $\overrightarrow{y_j y_k}$ for all $h \neq i$ and $k \neq j$ and $(\hat{s}, \hat{\theta})$ being the MLE of (s, θ) for the given pair of matching vectors. If $(x_i \leftrightarrow y_j)$ and $k = 1, \ldots, n$ with $h \neq i$ and $k \neq j$), it was reported in[6] that (s, θ) corresponds to the peak value of the accumulator array $M_{i,j}(\hat{s}, \hat{\theta})$. Moreover, this value w_{ij}^*, called the conditional **support** of point matching $(x_i \leftrightarrow y_j)$, is exactly equal to $p - 1$. When x_i does not match with y_j, one has $w_{ij}^* < p - 1$. This fundamental property is the basis of the fast 2D PPM algorithm.

The practical construction of accumulator arrays $M_{i,j}(\hat{s}, \hat{\theta})$ and peak values w_{ij}^* determination constitutes the Conditional Matching Pairs Support Finding (CMPSF) algorithm.[6] The original CMPSF algorithm was based on Hough transform, a discretization of the (s, θ)-space with an *arbitrary* vote procedure to increase accumulator arrays. An improvement of the CMPSF algorithm based on statistical test has been proposed recently in.[9] The major difficulty associated with the CMPSF algorithm is the correct increasing of the accumulators $M_{i,j}(\hat{s}, \hat{\theta})$. This difficulty is related to decide if two pairs of assumed matching vectors, say $(\overrightarrow{x_i x_h}, \overrightarrow{y_j y_k})$ and $(\overrightarrow{x_i x_{h'}}, \overrightarrow{y_j y_{k'}})$, support the same underlying reduced affine registration. In other words, to increase $M_{i,j}(\hat{s}, \hat{\theta})$ we first have to decide if $(\hat{s}_N, \hat{\theta}_N)$ computed from matching $(\overrightarrow{x_i x_h}, \overrightarrow{y_j y_k})$ is statistically equivalent to $(\hat{s}, \hat{\theta})$ computed from matching $(\overrightarrow{x_i x_h}, \overrightarrow{y_j y_k})$. Because of the equivalence between $z = [z\ z^{\perp}]'$ with $(\hat{s}, \hat{\theta})$ and $z_N = [z_N\ z_N^{\perp}]'$ with $(\hat{s}_N, \hat{\theta}_N)$ and the Gaussian property of the pdfs, we can perform (after some decorrelation preprocessing) a chi-square test to check the similarity of z_N with z. The result of the test allow us to increase and/or initialize $M_{i,j}(\hat{s}, \hat{\theta})$ accumulators. A more sophisticated test has been developed for the general case where we have to check the similarity of z_N with an existing cluster of, say $N-1$, similar vectors $\mathcal{C}_{N-1} = \{z_1, \ldots, z_{N-1}\}$. The reader is referred to[9] for details of the CMPSF algorithm.

For matching two point patterns $\mathcal{P}$ and $\mathcal{Q}$ under translation, rotation, scaling, local distortion, and missing/extra points conditions, the 2D-PPM algorithm consists of the following main steps:

S1. Initialization
- Set $w^* = 0$ and set $\hat{p} = \min(m, n)$

S2. Scanning of $\mathcal{P}$ and $\mathcal{Q}$ sets
- For $i = 1, \ldots, m$
  - For $j = 1, \ldots, n$
    - ⋆ Determine w_{ij}^* and $\mathcal{M}_{ij}^*$ using CMPSF algorithm below
    - ⋆ if $w_{ij}^* > w^*$ then set $w^* = w_{ij}^*$, $\mathcal{M}^* = \mathcal{M}_{ij}^*$, $i^* = i$ and $j^* = j$
  - End loop on j
  - The maximum support is not found, decrease $\hat{p}$ by one $\Rightarrow \hat{p} = \hat{p} - 1$
- End loop on i

tections (matching pairs) will be given by $\hat{p}$.

The fast version of this 2D-PPM algorithm which requires an additional matching flag array and minimal and maximal support thresholding steps can be found in.[9] In short, the 2D-PPM algorithm detects using CMPSF algorithm, for each pair of tried matching points $(x_i \leftrightarrow y_j)$, the point pairs that own the maximum conditional matching pairs support and the registration parameters (s, θ) which can match the most points between $\mathcal{P}$ and $\mathcal{Q}$; then 2D-PPM looks for the overall (unconditional) maximum matching support to get the optimal PPM solution. If required, the estimation of the affine registration parameters underlying the PPM solution can be obtained using the generalized least square estimator (GLSE) as described briefly at the end of this section.

The CMPSF algorithm for an assumed point matching $(x_i \leftrightarrow y_j)$ can be stated as follows:

S1. Initialization
- Choose an arbitrary but different pair of points $(x_{h'}, y_{k'}) \in \mathcal{P} \times \mathcal{Q}$ with $h' \neq i$ and $k' \neq j$
- Set $Cluster(1) = \{(x_{h'}, y_{k'})\}$ and $nc_{\max} = 1$

S2. Data clustering
- For $h = 1, \ldots, m$ $(h \neq h'$ and $h \neq i)$
  - For $k = 1, \ldots, n$ $(k \neq k'$ and $k \neq j)$
    - ⋆ Perform chi-square similarity test between $(\overrightarrow{x_i x_h}, \overrightarrow{y_j y_k})$ and $Cluster(nc)$ for $nc = 1, \ldots, nc_{\max}$
    - ⋆ if $(\overrightarrow{x_i x_h}, \overrightarrow{y_j y_k})$ belongs to an existing cluster (say $nc^\star$), update it by $Cluster(nc^\star) = Cluster(nc^\star) \cup \{(x_h, y_k)\}$; otherwise create a new one ($nc_{\max} = nc_{\max} + 1$ and $Cluster(nc_{\max}) = \{(x_h, y_k)\}$)
  - End loop on k
- End loop on h

S3. Search for the index nc^* of the cluster having the maximal cardinality
- $w_{ij}^* = Card[Cluster(nc^*)]$ is the desired support for pair (x_i, y_j)
- $\mathcal{M}_{ij}^* = \{(x_i, y_j)\} \cup Cluster(nc^*)$ is the *conditional* PPM solution with respect to (x_i, y_j).

The peak value $w_{ij}^* = c$ means that there are other c pairs of corresponding points supporting x_i matched with y_j; and therefore there are $c + 1$ pairs of corresponding points matching under the same underlying affine registration $(s_{ij}^*, \theta_{ij}^*)$. The cardinality of $\mathcal{M}_{ij}^*$ is exactly equal to $c+1$. A fast version of this CMPSF algorithm requires an additional match flag array to represent the matching condition between each point in $\mathcal{P}$ and each point in $\mathcal{Q}$.

previous 2D-PPM algorithm, and assuming that the number p of matching pairs $(x_i \leftrightarrow y_j)$ is greater than 2 (which is the number of independent parameters involved in the affine registration divided by the space dimension), the estimation of the affine registration parameters can easily be obtained using the generalized least square (GLSE) estimator as follows[1,9,6]:

$$\left[\hat{t}_x, \hat{t}_y, \hat{s}\cos(\hat{\theta}), \hat{s}\sin(\hat{\theta})\right]' = (C'P^{-1}C)^{-1}C'P^{-1}Y \quad (25)$$

where $P^{-1} = \mathrm{diag}[P_{bb}^{-1}, \ldots, P_{bb}^{-1}]$, $Y = [y_1', \ldots, y_p']'$ and $C = [C_1', \ldots, C_p']'$ with $C_j = \begin{bmatrix} 1 & 0 & x_i^1 & -x_i^2 \\ 0 & 1 & x_i^2 & x_i^1 \end{bmatrix}$, $j \in \{1, \ldots, p\}$.

5. LAYERED IMM WITH FAST PPM ALGORITHM FOR MBETT PROBLEM

The Interacting Multiple-Model (IMM) has been shown to be one of the most cost-effective and simple schemes for the estimation in hybrid systems,[2,3,10,12,4] in particular for tracking maneuvering targets. Therefore, a layered IMM algorithm is used for tracking maneuvering and bending extended targets. Specifically, we use the Layered MM structure combined with the IMM configuration in the sense that the MM estimator for each layer is an IMM algorithm. On the other hand, the data association problem is solved by the PPM algorithm.

Conceptually, the solution to the simplified MBETT problem consists of the following steps:

S1. Run the interaction and mode-conditioned prediction steps of the IMM estimator for the principal layer with the transition probabilities given by (7).

S2. Measurement validation and data association via PPM:
- Validate measurements using a statistical gate which takes into account the (mode-dependent) target size.
- Solve the PPM problem using Fast PPM algorithm to get the (mode-dependent) matched measurement vector z_k.

S3. Run the model-conditioned filtering and combination steps of the IMM estimator for the principal layer.

S4. Run the full cycle of the IMM estimator for the proper layer with the transition probabilities given by (10).

6. CONCLUSIONS

A novel multiple-model (MM) algorithm, called Layered MM (LMM) algorithm, has been developed. It takes full advantage of the layered structure of the models required for solving many practical problems. The need for

from the problem of tracking maneuvering and bending extended targets in a cluttered environment and other practical problems. The coupling in the models between the layers is accounted for by using time-varying transition probabilities of the models, which are probabilistic weighed sum (average) of the data-independent but other-model-dependent transition probabilities and the other-model probabilities. The coupling in the state between the layers is readily incorporated into the LMM algorithm. It is expected that the LMM will have a substantial reduction in computation and similar performance as compared with the standard MM algorithm. The uncertainties in the received measurements of the reflection points of an extended target is handled by the use of a fast point-pattern matching algorithm, in conjunction with a modified Conditional Matching Pairs Support Finding (CMPSF) algorithm proposed recently in.[9]

REFERENCES

1. Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York: Academic Press, 1988.
2. Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Boston, MA: Artech House, 1993.
3. Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. Storrs, CT: YBS Publishing, 1995.
4. H. A. P. Blom and Y. Bar-Shalom, "The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients," *IEEE Trans. Automatic Control*, vol. AC-33, pp. 780–783, Aug. 1988.
5. L. Brown, "A Survey of Image Registration Techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325–376, 1992.
6. S. Chang, F. Cheng, W. Hsu, and G. Wu, "Fast Algorithm for Point pattern Matching : Invariant to Translations, Rotations and Scales Changes," *Pattern Recognition*, vol. 30, no. 2, pp. 311–320, 1997.
7. F. Cheng, W. Hsu, and M. Kuo, "Recognition of handprinted Chinese characters via stroke relaxation," *Pattern Recognition*, vol. 26, no. 4, pp. 579–593, 1993.
8. E. Daeipour and Y. Bar-Shalom, "IMM for Tracking Maneuvering Targets in the Presence of Glint," *IEEE Trans. Aerospace and Electronic Systems* (to appear), vol. AES-34, July 1998.
9. J. Dezert, "Tracking Maneuvering and Bending Extended Target in Cluttered Environment," in *Proc. 1998 SPIE Conf. Signal and Data Processing of Small Targets,* vol. 3373, (Orlando, Florida, USA), April 1998.
10. X. R. Li, "Hybrid Estimation Techniques," in *Control and Dynamic Systems: Advances in Theory and* 287, New York: Academic Press, 1996.
11. L. Lucido, "Toward terrain based underwater navigation by bathymetric profiles matching," Tech. Rep. RR 3211, INRIA, Sophia Antipolis, France, Jul 1997.
12. E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting Multiple Model Methods in Target Tracking," *IEEE Trans. Aerospace and Electronic Systems*, vol. 34, pp. 103–123, jan 1998.
13. S. McClintock, T. Lunney, and A. Hashim, "A Genetic Algorithm Environment for Star Pattern Recognition," *Preprint INFM-97-007, University of Ulster, Londonderry,*, pp. 1–13, 1997.
14. F. Murtagh, "A New Approach to Point Pattern Matching," *Astronomical Society of The Pacific*, vol. 104, 1992.
15. J. Sprinkaz and M. Werman, "Affine Point Matching," *Pattern Recognition Letter*, vol. 15, pp. 337–339, apr 1994.
16. P. Tsang, P. Yuen, and F. Lam, "Classification of partially occluded objects using 3-point matching and distance transformation," *Pattern Recognition*, vol. 27, no. 1, pp. 27–40, 1994.
17. G. Weber, "Point Pattern Matching," Tech. Rep. B 95-19, Freie Universit t Berlin, Dec 1995.
18. G. Weber, L. Knipping, and H. Alt, "An Application of Point Pattern Matching in Astronautics," *J. Symbolic Computation*, vol. 17, pp. 321–340, 1994.
19. Z. Zhang, "Iterative Point Matching for Registration of free-form curves," Tech. Rep. RR 1658, INRIA, 1992.
20. Z. Zhang, "Registration of Free-Form Surfaces Representated by Point Sets," *Traitement du signal*, vol. 10, no. 4, 1993.
21. Z. Zhang, "The Matching Problem: The State of Art," Tech. Rep. RR 2146, INRIA, Dec 1993.