Radar Target Tracking - Viterbi vs. IMM

appeared in IEEE Trans. on Aerospace and Electronic Systems, Vol. 27, No. 3, pp. 550-563, May 1991

A. Averbuch
*S. Itzikowitz**
T. Kapon
The Moise and Frida Eskenasy Institute of Computer Science
The Raymond and Beverly Sackler Faculty of Exact Sciences
Tel Aviv University
Tel Aviv 69978 , Israel
E-MAIL : amir@math.tau.ac.il

Abstract

The performances of the Viterbi and the Interacting Multiple Model algorithms, applied to radar tracking and detection, are investigated and compared. Two different cases are considered. In the first case, targets acceleration is identical to one of the system models while in the second case it is not. The algorithms performance depends monotonically on the maximal magnitude of the difference between models acceleration, on the time interval between measurements and on the reciprocal of the measurements error standard deviation. In general, when these parameters are relatively large, both algorithms perform well. When they are relatively small, the Viterbi algorithm is better. However, during delay periods, namely right after the start of a maneuver, the Interacting Multiple Model algorithm provides better estimations.

Key Words - Interacting mutliple model, maneuvering target, multifilters, radar tracking, Viterbi algorithm.

I. INTRODUCTION

Maneuvering targets are modeled by the following dynamical equation

$$\vec{x}(t+1) = A(t)\vec{x}(t) + \vec{U}(t) + \vec{v}_0(t) \quad t = 0, 1, 2, \ldots \tag{1}$$

and the corresponding measurements equation:

$$\vec{z}(t) = H(\vec{x}(t)) + \vec{v}_1(t) \quad t = 0, 1, 2, \ldots \tag{2}$$

where $\vec{x}(t)$ is a vector which describes the system state at time t, $A(t)$ is the transition matrix, $\vec{U}(t)$ is a deterministic input, $\vec{v}_0(t)$ is a white noise called the process noise, $\vec{z}(t)$ is a measurement taken at time t, $H(\cdot)$ is a nonlinear operator, and $\vec{v}_1(t)$ is a white noise called the measurement noise.

The techniques for detection and estimation of abrupt changes in dynamical systems such as radar tracking, can be divided into two main classes: the Residual Based approach [15], and the Multiple Model (or Multifilter) approach [15]. In the first case there is a single filter driven by the measurements. In the Multiple Model approach several models $\{M_i\}_1^N$ of the form of Eqs. (1) and (2) are postulated along with their probabilities of being correct. These models differ by one (or more) parameters, and the transition from one model to another is governed by a finite state Markov [6] (or semi-Markov [10]) process. An additional

*The work of this author was supported by research grant #0337 of the Israeli National Council for Research and Development

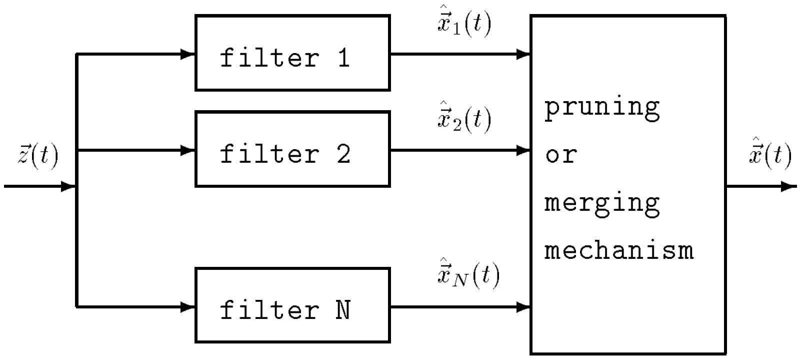

Figure 1: Multi-Filters Algorithm: $\hat{x}_i(t)$ denotes the estimation based on the i-th model.

mechanism uses merging techniques in order to keep the number of hypothese contant (see Fig. 1). A Kalman Filter is set up for each model and all filters are driven by the same measurements. The multifilter approach provides a better accuracy on one hand, but has higher computational load on the other hand, as compared to the Residual Based approach[4,12]. However, the multifilter approach is very suitable to parallel processing [1]. Within the Multiple Model class of algorithms there are two primary approaches: the Bayesian approach and the Maximum A-posteriori Probability (MAP) approach.

The Bayesian Approach: In this approach the system state vector $\vec{x}(t)$ is estimated by an approximation to its conditional mean $E(\vec{x}(t)|Z^t)$, where Z^t is the measurements sequence $\{\vec{z}(0), \vec{z}(1), \vec{z}(2), \ldots, \vec{z}(t)\}$.

Let us denote by $M_i(t)$ the event that the model M_i is valid at time t, and let $Hist_{\vec{k}}(t) \stackrel{\text{def}}{=} \{M_{k_0}(0), M_{k_1}(1), M_{k_2}(2), \ldots, M_{k_t}(t)\}$ where $\vec{k} = (k_0, k_1, k_2, \ldots, k_t)$, be a specific sequence of models from the space of all possible sequences. The conditional mean $E(\vec{x}(t)|Z^t)$ can be computed by the weighted sum of all the conditional means given specific histories:

$$E(\vec{x}(t)|Z^t) = \sum_{\vec{k}} E(\vec{x}(t)|Hist_{\vec{k}}(t), Z^t) P(Hist_{\vec{k}}(t)|Z^t) \tag{3}$$

Each conditional mean on the right hand side of Eq. (3) can be approximated by using the corresponding Extended Kalman Filter. However, as the number of such 'histories' is growing exponentially with time, the optimal solution is intractable and approximating techniques based on 'merging histories' are being used [6,13]. For example, the i-th order General Pseudo Bayesian (GPBi) multiple model algorithm considers only the current and the last $(i-1)$ sampling periods. In this way, at all times the number of terms in Eq. (3) remains constant (N^i for GPBi). However, the Performance to Computational Load Ratio (PCLR) in these models is unsatisfactory. The Interacting Multiple Model (IMM) algorithm is another merging algorithm which conceptually is similar to GPB2, but reduces the computational load by suitable merging of previous estimates. Specifically, in IMM the number of estimations passed from one time step to the next is N. Each estimation corresponds to another model:

$$\hat{\vec{x}}_i(t-1) \stackrel{\text{def}}{=} E(\vec{x}(t-1)|M_i(t-1), Z^{t-1}) \quad 1 \le i \le N \tag{4}$$

These estimations are the input to the mixing step. The output of the mixing step are again N estimations:

$$\vec{x}_j^0(t-1) \stackrel{\text{def}}{=} E(\vec{x}(t-1)|M_j(t), Z^{t-1}) \quad 1 \le j \le N \tag{5}$$

The next step is prediction and measurement update by Kalman filters under the hypotheses $\{M_j(t), 1 \le j \le N\}$ which result in the updated estimates:

$$\hat{\vec{x}}_j(t) = E(\vec{x}(t)|M_j(t), Z^t) \quad 1 \le j \le N \tag{6}$$

The final estimation is given by:

$$\hat{\vec{x}}(t) = \sum_{j=1}^{N} \hat{\vec{x}}_j(t) P(M_j(t)|Z^t) \tag{7}$$

The IMM is considered the most cost-efficient in this class [3].

The MAP (Maximum A-posteriori Probability) Approach: In this approach we try to approximate the conditional mean of the system state vector based on a single 'most probable history'. This conditional mean is approximated by using the corresponding Extended Kalman Filters. Again, the optimal solution is impractical and we use suboptimal techniques [11,14].

An example of one way of reducing the number of histories is to throw away 'most unprobable' histories. Let us assume that there are N possible models, and until time t we kept track of the N histories with the maximal likelihood. At the next time step, we have N^2 possible extensions of the previous histories. Each such extension is treated as a new hypothesis and its likelihood is computed. At this stage, we choose only the N hypotheses with the maximal likelihood and delete the other $N^2 - N$ histories. Now, we can proceed to the next time step and repeat this procedure.

Another possible algorithm is the Viterbi Algorithm (VA) which is a dynamic programming algorithm, extensively used in speech processing and communication applications [7,11]. In VA, as in the previous example, we keep track of N histories. At the beginning of each time step we have to choose the N surviving histories out of the N^2 possible extensions of the previous histories. The strategy in VA is to choose one history with maximal likelihood for each possible current model $M_j(t)$ $1 \leq j \leq N$. The assumption underlying this algorithm is that the Probability Density Function of the measurements depends only on the current model which the system obeys. When this assumption is satisfied the Viterbi algorithm gives the optimal solution. In the tracking problem, however, this is not the case because the measurements depend on the target dynamics history. Thus, the VA does not give the optimal MAP history of models but some 'approximated' history.

The objective of this paper is to examine the estimation quality of MAP algorithms as compared to Bayesian algorithms for radar tracking of a single maneuvering target. We note here that MAP approach is extensively used in the problem of multitarget tracking, but to the best of our knowledge, has not been applied to a single target radar tracking. As representative algorithms we have chosen from the MAP approach the Viterbi Algorithm (VA) [7]. For the Bayesian approach we have chosen the Interacting Multiple Model (IMM) algorithm.

In Section II we present the VA algorithm; the IMM algorithm is presented in Section III. Section IV describes our kinematic model and Section V compares the performance of our two algorithms as demonstrated by numerical simulations. Finally, our conclusions are summarized in Section VI.

II. THE VITERBI ALGORITHM

A. The Maximum A-posteriori Probability (MAP) Approach

In this approach we first find:

$$Hist^{\star}(t) = arg\{\max_{\vec{k}}(P(Hist_{\vec{k}}(t)|Z^t)\} \tag{8}$$

and then apply an Extended Kalman Filter on $Hist^{\star}(t)$ to get an approximation to the conditional mean:

$$\hat{\vec{x}}(t) = E(\vec{x}(t)|Hist^{\star}(t), Z^t) \tag{9}$$

The term $P(Hist_{\vec{k}}(t)|Z^t)$ can be computed recursively in the following way:

$$P(Hist_{\vec{k}}(t)|Z^t) = P(Hist_{\vec{k}}(t)|\vec{z}(t), Z^{t-1}) =$$

$$\stackrel{\text{Bayes rule}}{=} \frac{P(\vec{z}(t)|Hist_{\vec{k}}(t), Z^{t-1})P(Hist_{\vec{k}}(t)|Z^{t-1})}{P(\vec{z}(t)|Z^{t-1})} =$$

$$= \frac{P(\vec{z}(t)|Hist_{\vec{k}}(t), Z^{t-1})P(M_{k_1}(1)\dots M_{k_t}(t)|Z^{t-1})}{P(\vec{z}(t)|Z^{t-1})} =$$

$$= \frac{P(\vec{z}(t)|Hist_{\vec{k}}(t), Z^{t-1})P(M_{k_t}(t)|M_{k_1}(1)\dots M_{k_{t-1}}(t-1), Z^{t-1})}{P(\vec{z}(t)|Z^{t-1})} \cdot$$

$$P(M_{k_1}(1)\dots M_{k_{t-1}}(t-1)|Z^{t-1}) =$$

$$\overset{\text{Markov proc.}}{=} \frac{P(\vec{z}(t)|Hist_{\vec{k}}(t), Z^{t-1})p_{k_{t-1}k_t}P(M_{k_1}(1)\dots M_{k_{t-1}}(t-1)|Z^{t-1})}{P(\vec{z}(t)|Z^{t-1})} \tag{10}$$

where $p_{k_{t-1}k_t}$ is the transition probability from model $M_{k_{t-1}}$ to model M_{k_t} and $P(\vec{z}(t)|Z^{t-1})$ is a normalization constant. The term $P(\vec{z}(t)|Hist_{\vec{k}}(t), Z^{t-1})$ is called the likelihood function and is given by:

$$P(\vec{z}(t)|Hist_{\vec{k}}(t), Z^{t-1}) = N(\nu_{\vec{k}}(t); 0, S_{\vec{k}}(t)) \tag{11}$$

where $\nu_{\vec{k}}(t)$ is the innovation of the Kalman filter matched to $Hist_{\vec{k}}(t)$ and $S_{\vec{k}}(t)$ is the corresponding covariance matrix. Note that the term $P(M_{k_1}(1)\dots M_{k_{t-1}}(t-1)|Z^{t-1})$ is already known at this stage because it was computed in the same way at time $t-1$. To implement Eq. (8) we need a growing number of Kalman filters (N^t) in order to compute $P(Hist_{\vec{k}}(t)|Z^t)$ for every possible $\vec{k}$. As this optimal solution is impractical some hypotheses reduction technique is needed as in the Bayesian approach. In order to keep constant the number of histories we keep track of, we have to decide at every stage which of the histories that 'survived' until this time should be ignored and which should be passed on to the next step.

The Viterbi Algorithm (VA) is a dynamic programming algorithm, designed for a system which obeys one of N models with Markovian transition process, where the measurement depends probabilistically only on the current model:

$$P(\vec{z}(t)|Hist_{\vec{k}}(t), Z^{t-1}) = P(\vec{z}(t)|M_j(t), Z^{t-1}) \tag{12}$$

in which $M_j(t)$ is the **last** model in the sequence $Hist_{\vec{k}}(t)$. In other words, the Probability Density Function (PDF) of the measurement depends only on the current model regardless of when and how this model was entered. In that case the Viterbi Algorithm finds the optimal MAP history of models. In our case however, as

$$\vec{z}(t) = H(\vec{x}(t)) + \vec{v}_1(t)$$

and

$$P(\vec{x}(t)|Hist_{\vec{k}}(t), Z^{t-1}) \neq P(\vec{x}(t)|M_j(t), Z^{t-1})$$

we get

$$P(\vec{z}(t)|Hist_{\vec{k}}(t), Z^{t-1}) \neq P(\vec{z}(t)|M_j(t), Z^{t-1}).$$

Thus, the PDF of the measurement depends on the entire history and not only on the current model. As the assumption in Eq. (12) is not satisfied, the VA does not give the optimal MAP history of models but some 'approximated' history.

B. The Equivalent Problem in a Trellis Graph Form

As mentioned before, the term $P(Hist_{\vec{k}}(t)|Z^t)$ can be computed recursively using Bayes rule:

$$P(Hist_{\vec{k}}(t)|Z^t) = P(Hist_{\vec{k}}(t)|\vec{z}(t), Z^{t-1}) =$$

$$= \frac{P(\vec{z}(t)|Hist_{\vec{k}}(t), Z^{t-1})P(Hist_{\vec{k}}(t)|Z^{t-1})}{P(\vec{z}(t)|Z^{t-1})}$$

Substituting according to the assumption in Eq. (12) yields:

$$P(Hist_{\vec{k}}(t)|Z^t) = \frac{P(\vec{z}(t)|M_{k_t}(t), Z^{t-1})P(M_{k_t}(t), Hist_{\vec{k}}(t-1)|Z^{t-1})}{P(\vec{z}(t)|Z^{t-1})} =$$

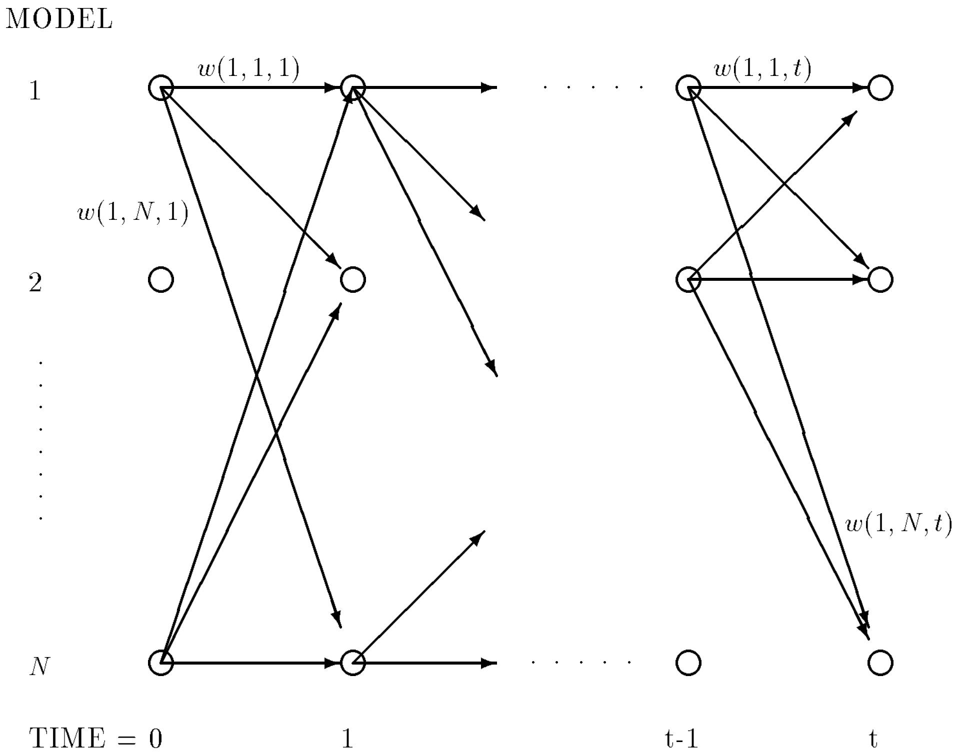

Figure 2: The optimal path problem

$$= \frac{P(\vec{z}(t)|M_{k_t}(t), Z^{t-1}) p_{k_{t-1}k_t} P(Hist_{\vec{k}}(t-1)|Z^{t-1})}{P(\vec{z}(t)|Z^{t-1})} \tag{13}$$

$$\text{where} \quad Hist_{\vec{k}}(t) = \{Hist_{\vec{k}}(t-1), M_{k_t}(t)\}$$

For a given measurements record Z^t we can write:

$$P(Hist_{\vec{k}}(t)|Z^t) = const \cdot L(t, k_t) p_{k_{t-1}k_t} P(Hist_{\vec{k}}(t-1)|Z^{t-1}) \tag{14a}$$

$$\text{where} \quad L(t, k_t) = P(\vec{z}(t)|M_{k_t}(t), Z^{t-1}) \tag{14b}$$

If we continue the recursion step t times we get:

$$\begin{aligned} P(Hist_{\vec{k}}(t)|Z^t) = const \cdot & \left[L(t, k_t) p_{k_{t-1}k_t}\right] \cdot \\ & \left[L(t-1, k_{t-1}) p_{k_{t-2}k_{t-1}}\right] \cdot \\ & \cdots \left[L(1, k_1) p_{k_1}\right] \end{aligned} \tag{15}$$

where p_{k_1} is the initial probability of model M_{k_1}. Let us denote:

$$w(k_{t-1}, k_t, t) = -\ln(L(t, k_t) p_{k_{t-1}k_t}) \tag{16}$$

Now we construct a Trellis graph (see Fig. 2) which consists of $N \cdot (t+1)$ nodes organized in N rows and $(t+1)$ columns. Each node in column i is connected by an edge to each node in column $i+1$. Each edge is assigned a weight(cost) $w(k_{t-1}, k_t, t)$ where k_{t-1} is the origin row, k_t is the destination row and t is the column of the destination node. It is evident that the problem of finding MAP history in Multiple-Model system as stated in Eq. (8) is equivalent to the problem of finding the optimal path (i.e. the path with minimal total weight) in the trellis graph. This problem has the following simple solution [7].

We will use 3 new definitions:

$$
\begin{aligned}
path_j(t) &\stackrel{\text{def}}{=} \{\text{the optimal path to node } j \text{ at time(column) } t\} \\
W_j(t) &\stackrel{\text{def}}{=} \{\text{the total weight of } path_j(t)\} \\
path(t) &\stackrel{\text{def}}{=} \{\text{the optimal path to time(column) } t\}
\end{aligned}
$$

The Algorithm:

For $t = 1, 2, 3, \ldots$ do:

1. For each node in row j at time t find the optimal path to it and its weight by:

$$
\begin{aligned}
i_{min}(j) &= arg\{\min_{1 \le i \le N}(path_i(t-1) + w(i,j,t))\} \\
path_j(t) &= \{path_{i_{min}(j)}(t-1), j\} \\
W_j(t) &= W_{i_{min}(j)}(t-1) + w(i_{min}(j), j, t)
\end{aligned}
$$

2. The optimal path at time t is:

$$path(t) = path_{j_{min}}(t)$$

$$\text{where} \quad j_{min} = arg\{\min_{1 \le j \le N}(W_j(t)\}$$

C. The Viterbi Algorithm Applied to the Tracking Problem

Let us now return to the original form of the problem. The equivalent term to $path_j(t)$ is $Hist_j(t)$ which denotes the MAP history of models which ends with model $M_j(t)$. The equivalent term to the path total weight $W_j(t)$ is the probability $\mu_j(t) = P(Hist_j(t)|M_j(t), Z^t)$. Thus, when we start the operations of the time slice t we have as input the histories $\{Hist_j(t),\ 1 \le j \le N\}$, the corresponding weights $\{\mu_j(t),\ 1 \le j \le N\}$ and in addition, the estimations and covariances $\{(\hat{\vec{x}}_j(t), C_j(t)),\ 1 \le j \le N\}$ produced by Kalman filters matched to the corresponding histories. Each time slice consists of the following stages:

1. *Probability calculations* - For each possible model $M_j(t)$ do:

   (a) For each possible previous model $M_i(t-1)$ compute the probability weight of the history $\{Hist_i(t-1), M_j(t)\}$, where $Hist_i(t-1)$ is the MAP history until time $t-1$ which ends in model M_i. This is given by:

$$
\begin{aligned}
\mu_{ij}(t) &= \mu_i(t-1) p_{ij} P(\vec{z}(t)|M_j(t), M_i(t-1), Z^{t-1}) = \\
&= \mu_i(t-1) p_{ij} P(\vec{z}(t)|\hat{\vec{x}}_i(t-1), C_i(t-1), M_j(t))
\end{aligned}
\tag{17a}
$$

   The term $P(\vec{z}(t)|\hat{\vec{x}}_i(t-1), C_i(t-1), M_j(t))$ is evaluated from the innovation of the Extended Kalman Filter matched to $M_j(t)$ with $(\hat{\vec{x}}_i(t-1), C_i(t-1))$ as input:

$$P(\vec{z}(t)|\hat{\vec{x}}_i(t-1), C_i(t-1), M_j(t)) = N(\nu_{ij}(t); 0, S_{ij}(t)) \tag{17b}$$

   where $\nu_{ij}(t)$ and $S_{ij}(t)$ are the innovation and its covariance accordingly.

   (b) Choose the surviving history by:

$$
\begin{aligned}
i_{max}(j) &= arg\{\max_{1 \le i \le N}(\mu_{ij}(t))\} \\
Hist_j(t) &= \{Hist_{i_{max}(j)}(t-1), M_j(t)\} \\
\mu_j(t) &= \mu_{i_{max}(j)j}(t)
\end{aligned}
$$

2. *Output generation* - The output may be generated in at least two ways. If we follow the original algorithm then:

$$j_{max} = arg\{\max_{1 \le j \le N} \mu_j(t)\} \quad \text{and} \quad \hat{\vec{x}}(t) = \hat{\vec{x}}_{j_{max}}(t) \tag{18}$$

Another possibility is to merge histories which gives:

$$\hat{\vec{x}}(t) = \sum_{j=1}^{N} \hat{\vec{x}}_j(t)\mu_j(t) \tag{19}$$

III. THE INTERACTING MULTIPLE MODEL (IMM) ALGORITHM

It is evident [5] that GPB2 is superior to GPB1. However the number of hypotheses that have to be checked in every stage in GPB2 is N^2 compared to only N hypotheses in GPB1. The IMM algorithm is conceptually similar to GPB2 and produces similar estimates, but its computational load is similar to that of GPB1.

The innovation in IMM is that the hypotheses reduction step, which in the GPB family is performed after the measurement update, is replaced by a mixing step which is performed before the prediction. The input to the mixing step are the estimates made in the previous time step:

$$\hat{\vec{x}}_i(t-1) \stackrel{\text{def}}{=} E(\vec{x}(t-1)|M_i(t-1), Z^{t-1}) \quad 1 \le i \le N \tag{20}$$

The approximation introduced here is

$$P(\vec{x}(t-1)|M_i(t-1), Z^{t-1}) \sim N(\hat{\vec{x}}_i(t-1), C_i(t-1)) \tag{21}$$

where $C_i(t-1)$ is the error covariance associated with $\hat{\vec{x}}_i(t-1)$. The output is again N estimates

$$\vec{x}_j^0(t-1) \stackrel{\text{def}}{=} E(\vec{x}(t-1)|M_j(t), Z^{t-1}) \quad 1 \le j \le N \tag{22}$$

with the corresponding covariances. These estimates are the basis of prediction and measurement update by Kalman filters under the hypotheses $\{M_j(t) \quad 1 \le j \le N\}$ which result in the updated estimates:

$$\hat{\vec{x}}_j(t) = E(\vec{x}(t)|M_j(t), Z^t) \tag{23}$$

Thus, in the IMM we have only N hypotheses to be checked at every stage. The final output is given by combining the model conditioned estimates as follows:

$$\hat{\vec{x}}(t) = E(\vec{x}(t)|Z^t) = \sum_{j=1}^{N} E(\vec{x}(t)|M_j(t), Z^t)P(M_j(t)|Z^t)$$

$$= \sum_{j=1}^{N} \hat{\vec{x}}_j(t)P(M_j(t)|Z^t) \tag{24}$$

A. IMM - Detailed Description

Given the forms of the dynamics equation (Eq. (1)) and the measurement equation (Eq. (2)), we wish to estimate $\vec{x}(t)$. The IMM algorithm is based on N Extended Kalman filters set up for N different models of the form of Eqs. (1)-(2). At the beginning of each time slice an initial condition is composed for each filter separately. At the end of each time slice the estimations produced by the filters are combined to provide the final estimation. The input to each cycle is N model conditioned estimates $\hat{\vec{x}}_i(t-1)$ with their corresponding covariances $C_i(t-1)$ and N weights

$$\mu_i(t-1) \stackrel{\text{def}}{=} P(M_i(t-1)|Z^{t-1}) \tag{25}$$

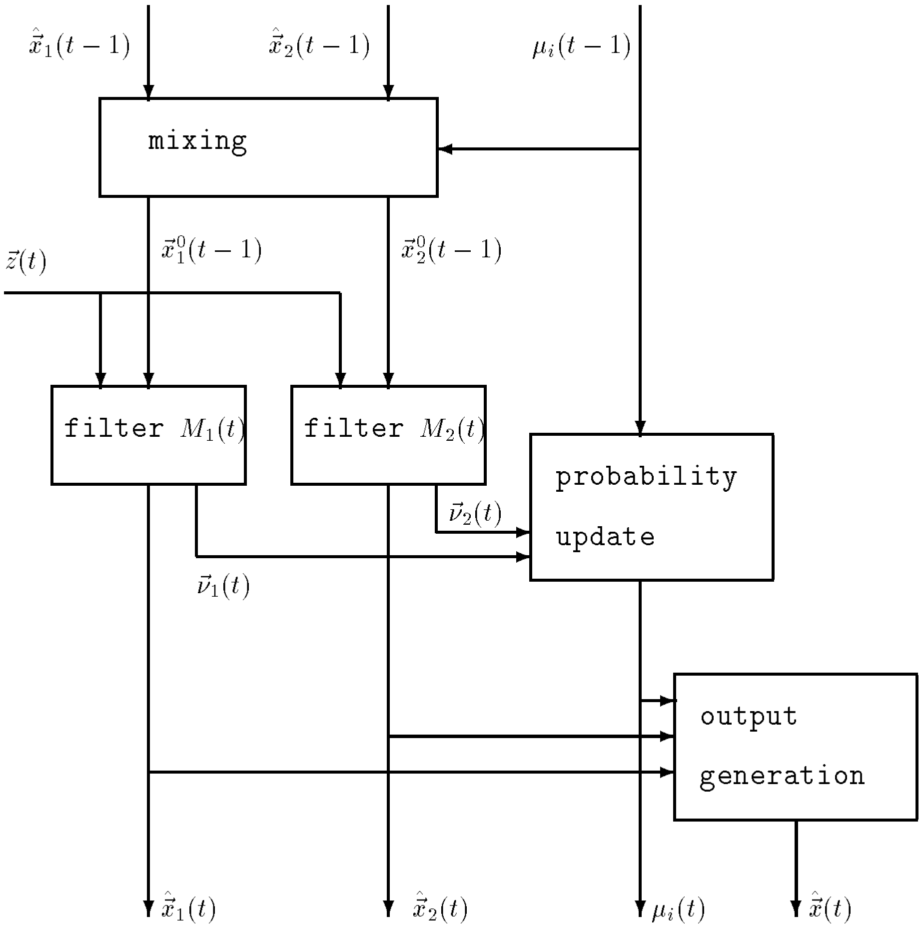

Figure 3: The Interacting Multiple Model algorithm for $N = 2$

The corresponding probability density function is assumed to be normal, i.e.

$$P(\vec{x}(t-1)|M_i(t-1), Z^{t-1}) \sim N(\hat{\vec{x}}_i(t-1), C_i(t-1)) \tag{26}$$

Each cycle consists of 4 steps which are performed for each possible model $M_j(t) \quad 1 \leq j \leq N$ (see Fig. 3):

1. *Input generation(mixing)* - The initial condition to the filter matched to $M_j(t)$ is generated by mixing estimates of all filters in the previous cycle as follows:

$$\vec{x}_j^0(t-1) \stackrel{\text{def}}{=} E(\vec{x}(t-1)|M_j(t), Z^{t-1}) =$$

$$= \sum_{i=1}^{N} E(\vec{x}(t-1)|M_i(t-1), M_j(t), Z^{t-1}) P(M_i(t-1)|M_j(t), Z^{t-1})$$

   If $M_i(t-1)$ is known then $\vec{x}(t-1)$ is independent of $M_j(t)$ and we get:

$$\vec{x}_j^0(t-1) = \sum_{i=1}^{N} E(\vec{x}(t-1)|M_i(t-1), Z^{t-1}) P(M_i(t-1)|M_j(t), Z^{t-1}).$$

   It then follows from Eq. 20 that

$$\vec{x}_j^0(t-1) = \sum_{i=1}^{N} \hat{\vec{x}}_i(t-1) P(M_i(t-1)|M_j(t), Z^{t-1}) \tag{27}$$

   where:

$$P(M_i(t-1)|M_j(t), Z^{t-1}) = \frac{P(M_j(t)|M_i(t-1), Z^{t-1}) P(M_i(t-1)|Z^{t-1})}{P(M_j(t)|Z^{t-1})} =$$

$$= \frac{p_{ij} P(M_i(t-1)|Z^{t-1})}{\sum_{i=1}^{N} p_{ij} P(M_i(t-1)|Z^{t-1})} = \frac{p_{ij}\mu_i(t-1)}{\sum_{i=1}^{N} p_{ij}\mu_i(t-1)} =$$

$$= \frac{p_{ij}\mu_i(t-1)}{c} \tag{28a}$$

   in which p_{ij} is given by

$$p_{ij} = P(M_j(t)|M_i(t-1)) \tag{28b}$$

   $\mu_i(t-1)$ is given by Eq. 25 and

$$c = \sum_{i=1}^{N} p_{ij}\mu_i(t-1) \tag{28c}$$

   The corresponding covariances are computed in a similar way and thus we get:

$$C_j^0(t-1) = \sum_{i=1}^{N} P(M_i(t-1)|M_j(t), Z^{t-1})[C_i(t-1) + (\hat{\vec{x}}_i(t-1) - \vec{x}_i^0(t-1))(\hat{\vec{x}}_i(t-1) - \vec{x}_i^0(t-1))^T] \tag{29}$$

2. *Kalman step* - Each of the pairs $(x_j^0(t-1), C_j^0(t-1))$ is used as an input to the Extended Kalman filter matched to the j-th model. The output is denoted by $(\hat{\vec{x}}_j(t), C_j(t))$.

3. *Probability calculations* - The weights $\{\mu_j(t)\} \quad 1 \leq j \leq N$, are computed from the innovations of the filters:

$$\mu_j(t) = P(M_j(t)|Z^t) = P(M_j(t)|\vec{z}(t), Z^{t-1}) =$$

$$= \frac{P(\vec{z}(t)|M_j(t), Z^{t-1}) P(M_j(t)|Z^{t-1})}{P(\vec{z}(t)|Z^{t-1})} =$$

$$= \frac{P(\vec{z}(t)|M_j(t), Z^{t-1}) \sum_{i=1}^{N} p_{ij} P(M_i(t-1)|Z^{t-1})}{P(\vec{z}(t)|Z^{t-1})} =$$

$$= \frac{P(\vec{z}(t)|M_j(t), Z^{t-1}) \cdot c}{c_1} \tag{30a}$$

where c is given in Eq. (28c) and

$$c_1 = P(\vec{z}(t)|Z^{t-1}) \tag{30b}$$

is a normalization constant which ensures that $\sum_{j=1}^{N} P(M_j(t)|Z^t) = 1$. In Eq. (30a) $P(\vec{z}(t)|M_j(t), Z^{t-1})$ is the likelihood function obtained from the innovation of the j-th Kalman filter - $\nu_j(t)$:

$$P(\vec{z}(t)|M_j(t), Z^{t-1}) = \frac{1}{\sqrt{(2\pi)^n |S_j(t)|}} \exp(-\frac{1}{2}\nu_j^T(t) S_j(t) \nu_j(t)) \tag{31}$$

where $\vec{\nu}_j(t) = \vec{z}(t) - H(\hat{\vec{x}}(t))$, $S_j(t)$ is the corresponding covariance and n is the state vector dimension.

4. *Output generation* - The final estimation for output purposes is:

$$\hat{\vec{x}}(t) = \sum_{j=1}^{N} \hat{\vec{x}}_j(t) \mu_j(t) \tag{32}$$

with the corresponding covariance:

$$C(t) = \sum_{j=1}^{N} \mu_j(t) [C_j(t) + (\hat{\vec{x}}_j(t) - \hat{\vec{x}}(t))(\hat{\vec{x}}_j(t) - \hat{\vec{x}}(t))^T] \tag{33}$$

IV. THE KINEMATIC MODEL

In solving the maneuvering target problem, we first have to choose a mathematical model of the target dynamics. If we want to apply the Multifilter approach, we then have to select one of the unknown parameters in the model, describe it by a finite set of values, and assign each such value to each of the filter models. We have chosen to use a kinematic model in which the acceleration during the maneuver is assumed to be almost constant, and its mean is taken from a finite set of accelerations. The perturbation upon this constant acceleration is modeled as white Gaussian noise. Thus the continuous time dynamics equation in a generic coordinate system is:

$$\ddot{\vec{r}}(\tau) = \vec{a}_i + \vec{V}(\tau) \quad 1 \le i \le N \tag{34}$$

where $\vec{r}(\tau)$ is the position vector, " $\cdot$ " denotes differentiation with respect to time, $\vec{a}_i$ is a discrete acceleration level and $\vec{V}(\tau)$ is a continuous time zero-mean Gaussian white noise. The corresponding state-space equation is:

$$\dot{\vec{x}}_0 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \vec{x}_0 + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \vec{a}_i + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \vec{V}(\tau) \tag{35}$$

where $\vec{x}_0 = \begin{pmatrix} \vec{r} \\ \dot{\vec{r}} \end{pmatrix}$ is the 4×1 state vector and $\vec{V}(\tau)$ is the 2×1 noise vector.

Preliminary tests showed that both IMM and Viterbi algorithms, suffer from a severe bias problem when the true acceleration of the target does not match one of the discrete levels of the model. Our solution was to augment an additional acceleration component to the state vector. This component includes the bias from the given discrete acceleration. After this augmentation, Eq. (35) is replaced by:

$$\dot{\vec{x}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \vec{x} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \vec{a}_i + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \vec{V}(\tau) \tag{35a}$$

in which

$$\vec{x} = \begin{pmatrix} \vec{r} \\ \dot{\vec{r}} \\ \ddot{\vec{r}} \end{pmatrix}$$

and $\vec{V}(\tau)$ is as before. The discretized equation becomes:

$$\vec{x}(\tau + \Delta\tau) = A\vec{x}(\tau) + \vec{U}(\tau) + \vec{v}(\tau) \tag{36}$$

where

$$\vec{U}(\tau) = B\vec{a}_i, \qquad \vec{v}(\tau) = C\vec{V}(\tau) \tag{37}$$

$$A = \begin{pmatrix} 1 & 0 & \Delta\tau & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta\tau & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} \frac{(\Delta\tau)^2}{2} & 0 \\ 0 & \frac{(\Delta\tau)^2}{2} \\ \Delta\tau & 0 \\ 0 & \Delta\tau \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and } C = \begin{pmatrix} \frac{(\Delta\tau)^2}{2} & 0 \\ 0 & \frac{(\Delta\tau)^2}{2} \\ \Delta\tau & 0 \\ 0 & \Delta\tau \\ 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{38}$$

Equation (36) is of the form of Eq. (1) when $\tau = t\Delta\tau, \quad t = 0, 1, \ldots$ and $\Delta\tau$ is the time increment.

The transitions between models (accelerations) are governed by a Markov process i.e.:

$$P(M_i(t)|Hist(t-1)) = P(M_i(t)|M_j(t-1)) \tag{39}$$

where $M_k(t)$ denotes the event that model M_k is valid at time t, and $Hist(t-1)$ is the complete history of models until time $t-1$. We adopted the approach of Gholson and Moose in [9] that suggested choosing the probabilities as follows:

$$P(M_i(t)|M_j(t-1)) = \begin{cases} p & i = j \\ (1-p)/(N-1) & i \neq j \end{cases} \tag{40}$$

where $p \sim 1$ and N is the number of models. The covariance of the process noise was recomputed according to the procedure described in [2] for the constant acceleration model, and is given by:

$$Q = E(\vec{v}(t)\vec{v}^T(t+k)) = CC^T q\delta(k) \tag{41}$$

V. SIMULATION AND PERFORMANCE ANALYSIS

A. Choice of Parameters

We have selected the use of 12 models which cover an accelerations range of $0-4$g ($0-40$ meter/sec^2) . The assignment of discrete acceleration levels to the models is as follows:

1. model #0 : $a_x = 0$ m/sec^2, $a_y = 0$ m/sec^2
2. model #1 : $a_x = 0$ m/sec^2, $a_y = 20$ m/sec^2
3. model #2 : $a_x = 20$ m/sec^2, $a_y = 0$ m/sec^2
4. model #3 : $a_x = -20$ m/sec^2, $a_y = 0$ m/sec^2
5. model #4 : $a_x = 0$ m/sec^2, $a_y = -20$ m/sec^2
6. model #5 : $a_x = 20$ m/sec^2, $a_y = 20$ m/sec^2
7. model #6 : $a_x = -20$ m/sec^2, $a_y = 20$ m/sec^2
8. model #7 : $a_x = 20$ m/sec^2, $a_y = -20$ m/sec^2
9. model #8 : $a_x = -20$ m/sec^2, $a_y = -20$ m/sec^2

10. model #9 : $a_x = 40$ m/sec^2, $a_y = 0$ m/sec^2
11. model #10: $a_x = 0$ m/sec^2, $a_y = 40$ m/sec^2
12. model #11: $a_x = 0$ m/sec^2, $a_y = -40$ m/sec^2

The allowed rate of change of acceleration within a specific model ($\sqrt{q\Delta t}$) was chosen to be of the order of 1 m/sec^2. The probability for staying in the same model, $P(M_i(t)|M_i(t-1))$, was chosen to be 97%.

B. The Measurement Model

The measurements are taken in polar coordinates by range and azimuth. We assume that both range and azimuth errors have normal distribution and are independent of each other. The standard deviation of the range error is given by:

$$\sigma_r = \begin{cases} 50 \text{ m} & \text{if } r > 5000 \text{ m} \\ r/100 & \text{if } r \leq 5000 \text{ m} \end{cases} \tag{42}$$

where r is the range. The standard deviation of the azimuth error σ_{θ_A} is 2mrad(miliradians). Thus, the matrix equation for the measurement can be written:

$$\vec{z}(t) = \begin{pmatrix} r \\ \theta_A \end{pmatrix} = H(\vec{x}(t)) + v_1(t) \tag{43}$$

in which $H(\vec{x}(t))$ is the non-linear function:

$$H(\vec{x}(t)) = \begin{pmatrix} \sqrt{x^2(t) + y^2(t)} \\ \arctan(y(t)/x(t)) \end{pmatrix}$$

and $v_1(t)$ is the measurement noise with the covariance:

$$R(t) = E(v_1(t)v_1^T(t)) = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_{\theta_A}^2 \end{pmatrix}$$

The performance of our algorithms were examined and compared for:

1. Different types of tracks of the maneuvering target. For this purpose we simulated two tracks of a maneuvering target. During the first period of both tracks the target moves at constant velocity. During the second period the target accelerates constantly. In the first track the target's acceleration level almost matches the acceleration level of one of the system models. In the second track, there is a significant difference between the target's true acceleration and the acceleration level of the closest model. Table 1 summarizes the parameters of the two tracks.

2. Different values of the nondimensional parameter $\eta = \frac{|\Delta\vec{a}|\Delta\tau^2}{\sigma_r}$ where:

   $|\Delta\vec{a}|$ is the maximal magnitude of the difference between the acceleration levels of any two models.
   $\Delta\tau$ is the time interval between measurements, and
   σ_r is the standard deviation of the measurement error during the tracking process.

   To demonstrate the significance of this parameter let us assume that in our system the target can move only along the x axis in one of the two discrete accelerations a_1 and a_2, and that at time t both the position x_0 and the velocity v_0 are known. At time $\tau + \Delta\tau$ a new measurement z is taken and we wish to decide according to it which is the true acceleration. If the target's true acceleration was a_1, than the new position would be:

   $$x(\tau + \Delta\tau) = x_0 + v_0\Delta\tau + \frac{1}{2}a_1\Delta\tau^2$$

   If, on the other hand the true acceleration was a_2 than the new position would be:

   $$x(\tau + \Delta\tau) = x_0 + v_0\Delta\tau + \frac{1}{2}a_2\Delta\tau^2$$

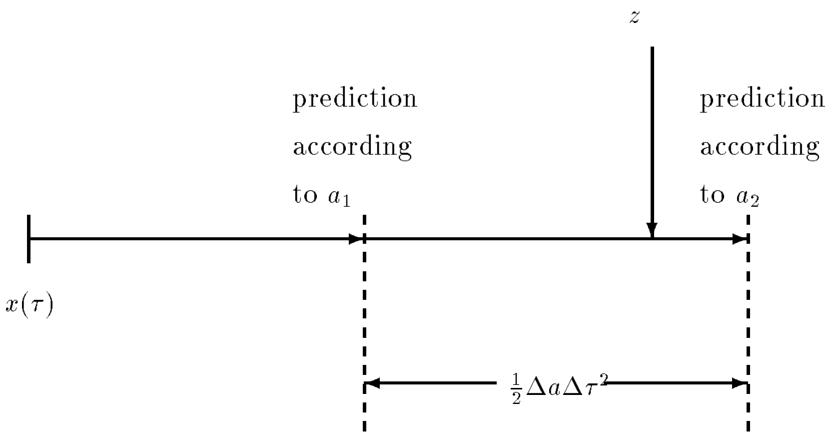

Figure 4: Distinction between models according to the measurement

The difference between the two positions would therefore be:

$$\frac{1}{2}(a_1 - a_2)\Delta\tau^2 = \frac{1}{2}\Delta a \Delta\tau^2$$

The ratio between the magnitude of the last term and the standard deviation of the measurement error determines the ability of the algorithms to make the right decision. The bigger this ratio, the easier is the distinction between models (see Fig. 4).

C. Representation of the Results

The simulations shown here correspond to fixed $|\Delta\vec{a}|$ (which is determined by the choice of our models) and fixed σ_r, while we let $\Delta\tau$ be 1s, 0.2s, an 0.1s. As explained previously, we have selected two different tracks which are presented in Table 1. The results of VA correspond to Eq. (19) but we note that their difference from those corresponding to Eq. (18) is negligible.

Table 1: The simulated tracks parameters

	Initial Position	Velocity until iteration #50	Acceleration after iteration #50
Track #1	$x = 8000$ m $y = 8000$ m	$v_x = 200$ m/sec $v_y = -25$ m/sec	$a_x = 37\,\mathrm{m/sec}^2$ $a_y = 0\,\mathrm{m/sec}^2$
Track #2	$x = 8000$ m $y = 8000$ m	$v_x = 200$ m/sec $v_y = -25$ m/sec	$a_x = 25\,\mathrm{m/sec}^2$ $a_y = 25\,\mathrm{m/sec}^2$

As we have 2 different tracks, and 3 different values of $\Delta\tau$, the total number of runs for each algorithm is 6. The results of each such run are represented by 3 graphs:

1. A graph which shows the probability weights given by VA to the closest models during a typical run (Figs. 5,8,11,14,17 and 20).
2. A graph which shows the probability weights given by IMM to the closest models during a typical run (Figs. 6,9,12,15,18 and 21).
3. A graph which compares the RMS error in the position estimation in IMM algroithm and VA over 50 Monte-Carlo runs (Figs. 7,10,13,16,19 and 22).

In addition we examined the correlation coefficient between the position estimation errors in IMM algorithm and VA and used an hypothesis testing technique suggested in [2] to compare the two algorithms. Figure 21 and Figs. 21-24 show the corresponding results for the first track. .

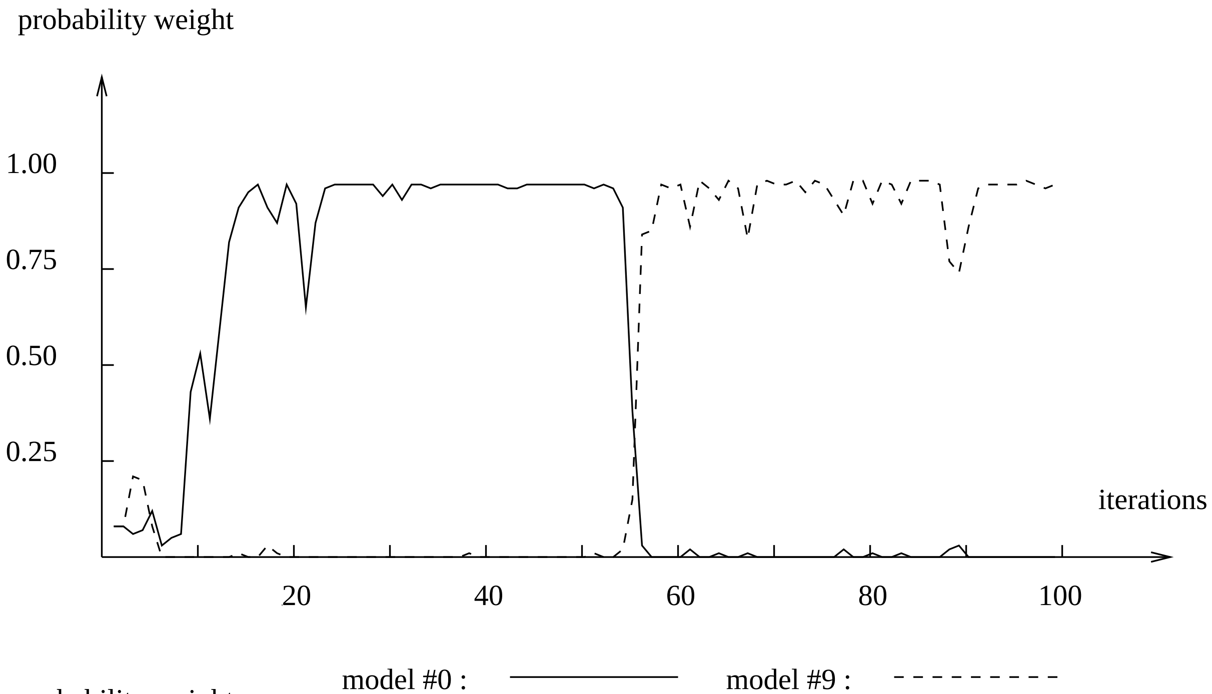

Figure 5: Probability weights in VA (track #1,$\Delta\tau = 1$sec.)

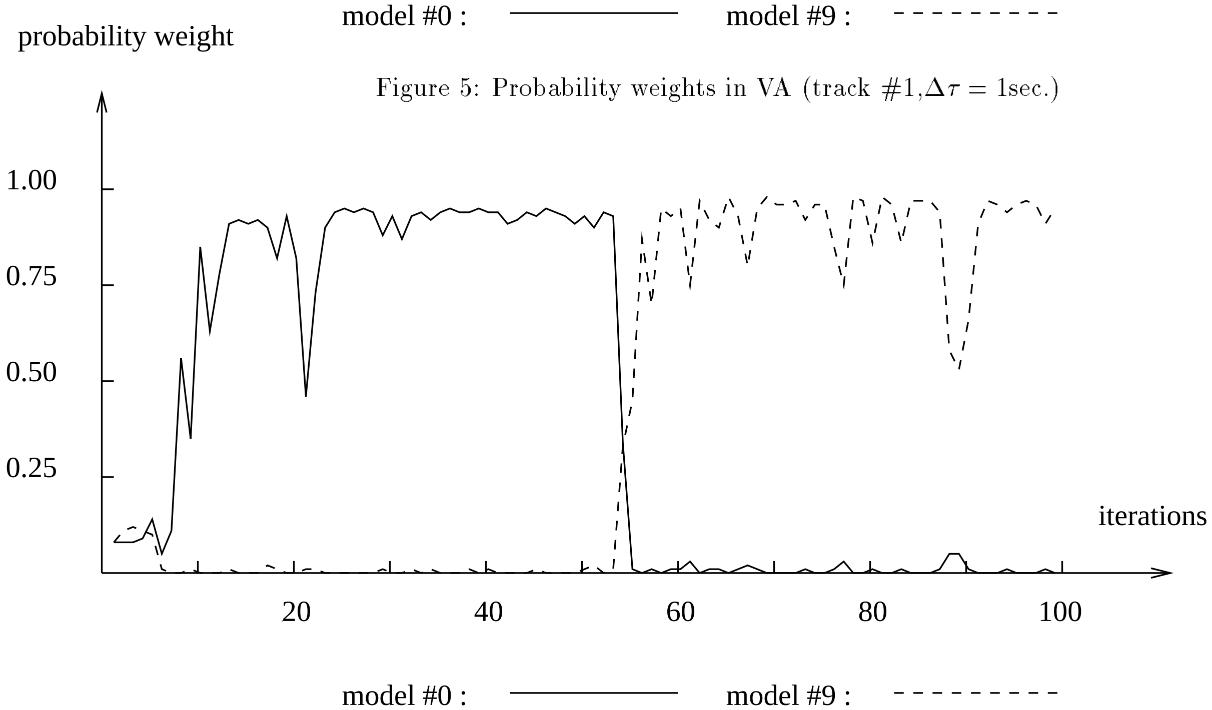

Figure 6: Probability weights in IMM (track #1,$\Delta\tau = 1$sec.)

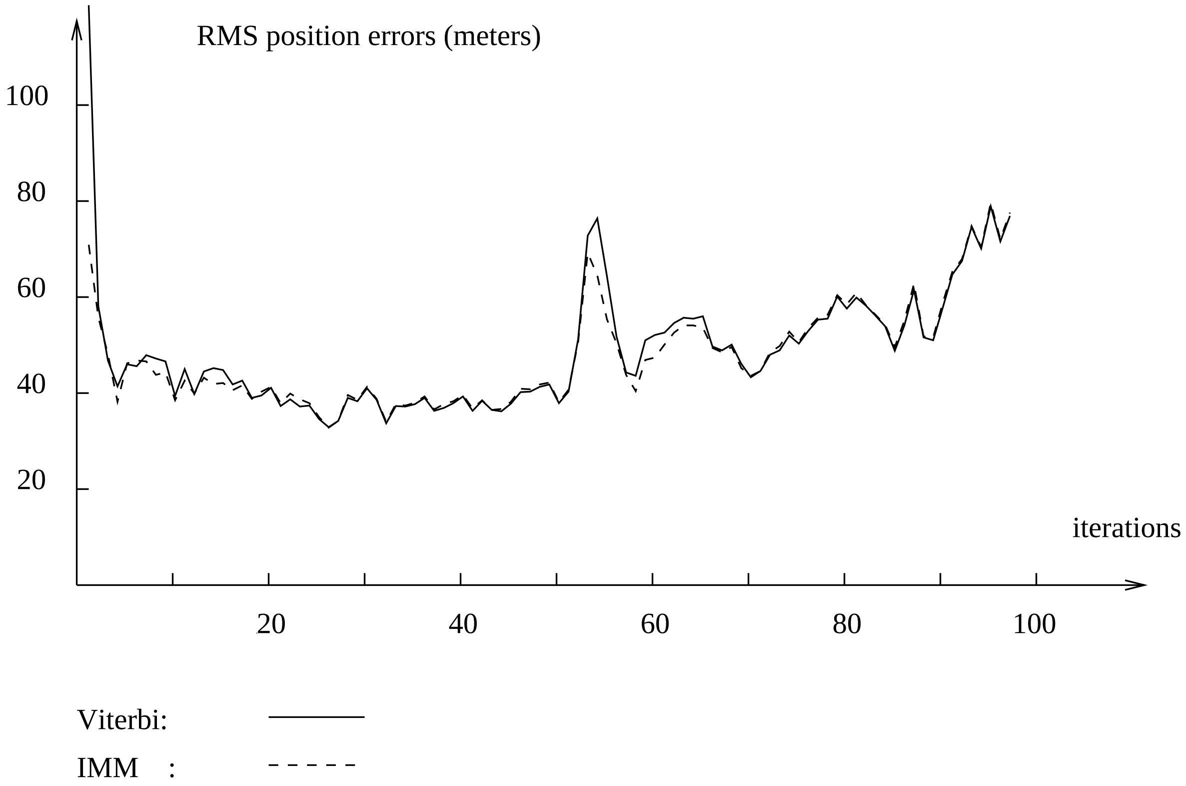

Figure 7: RMS position errors (track #1,$\Delta\tau = 1$sec.)

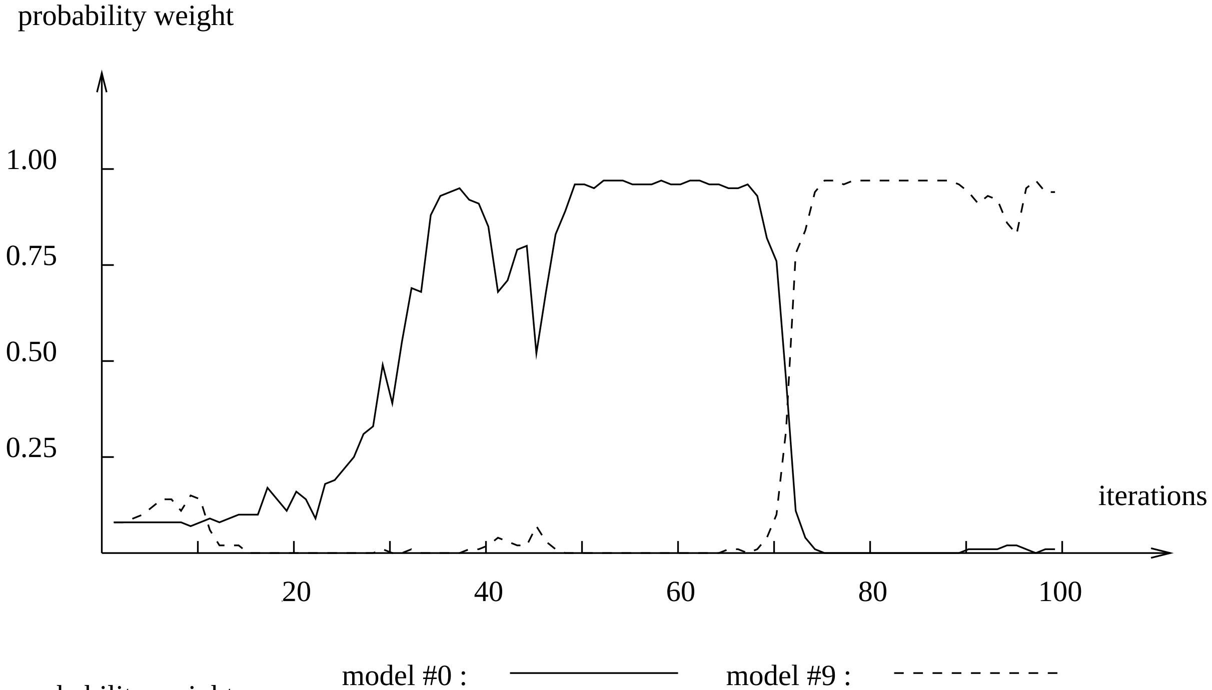

Figure 8: Probability weights in VA (track #1,$\Delta\tau = 0.2$sec.)

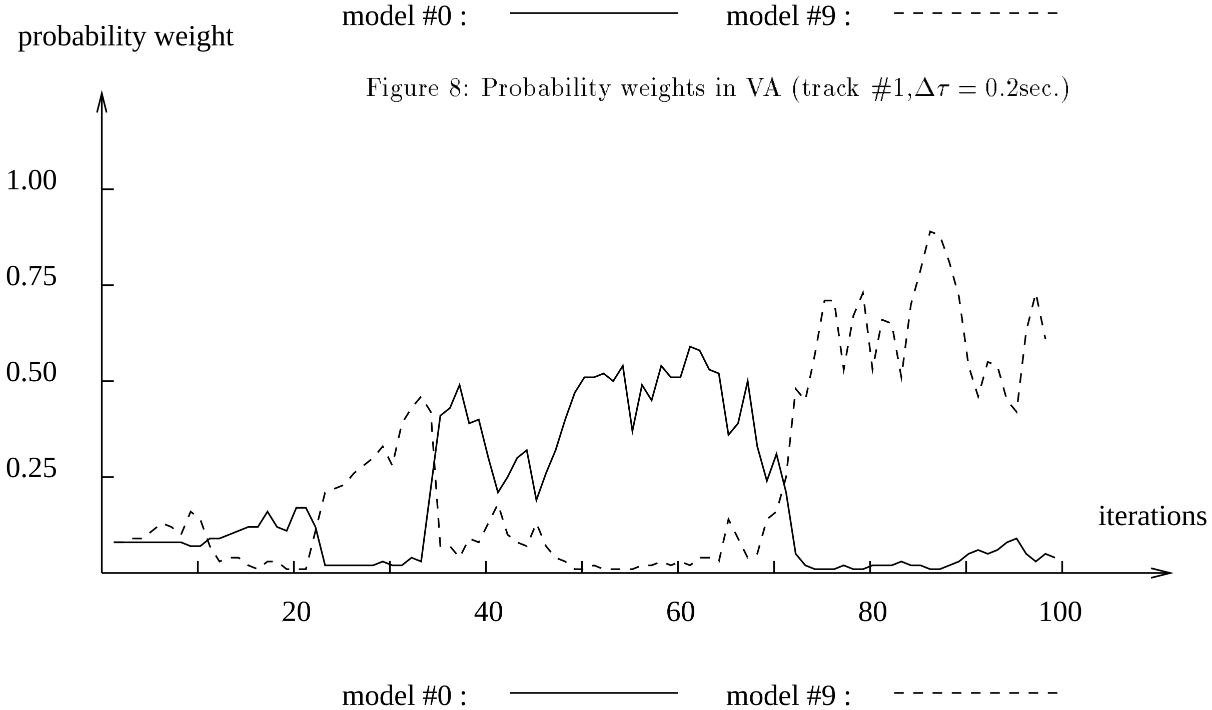

Figure 9: Probability weights in IMM (track #1,$\Delta\tau = 0.2$sec.)

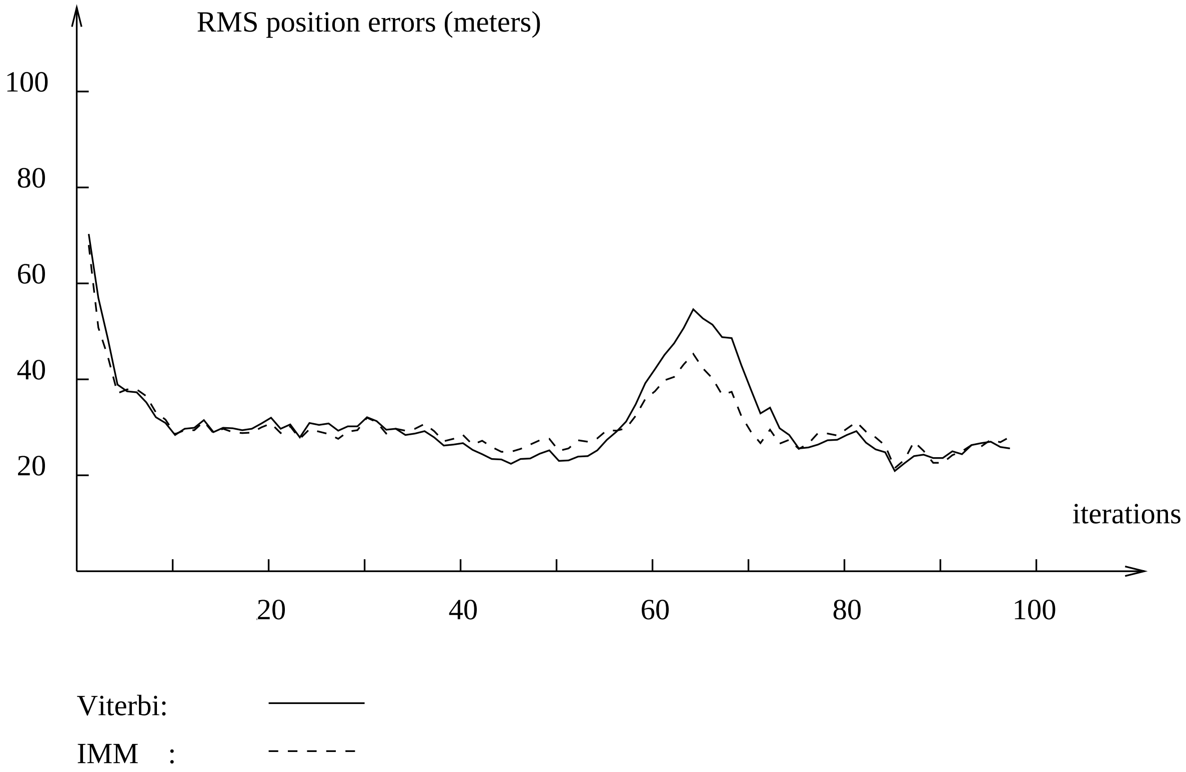

Figure 10: RMS position errors (track #1,$\Delta\tau = 0.2$sec.)

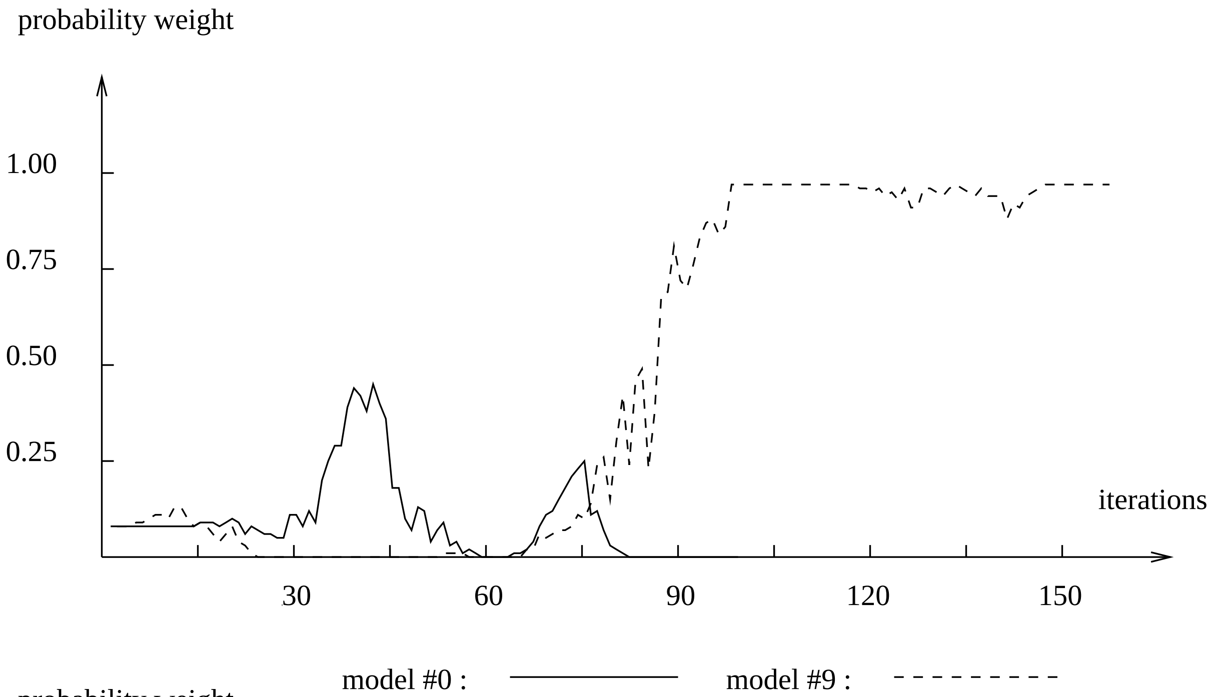

Figure 11: Probability weights in VA (track #1,$\Delta\tau = 0.1$sec.)

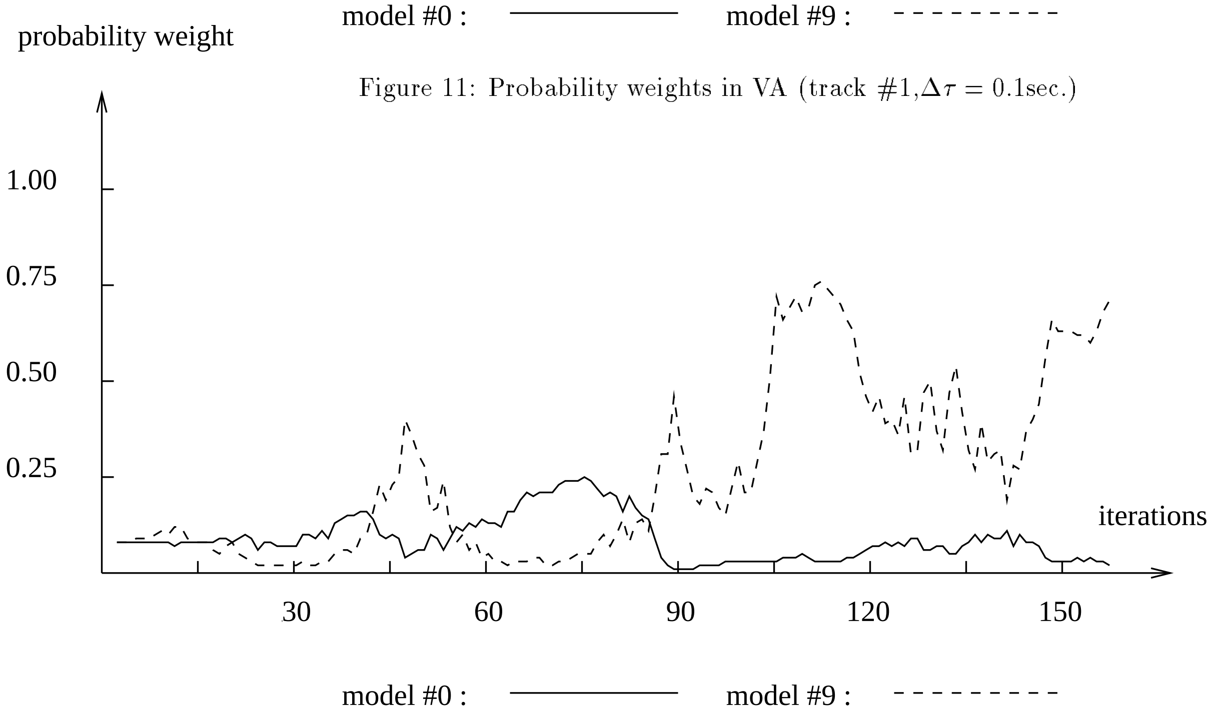

Figure 12: Probability weights in IMM (track #1,$\Delta\tau = 0.1$sec.)

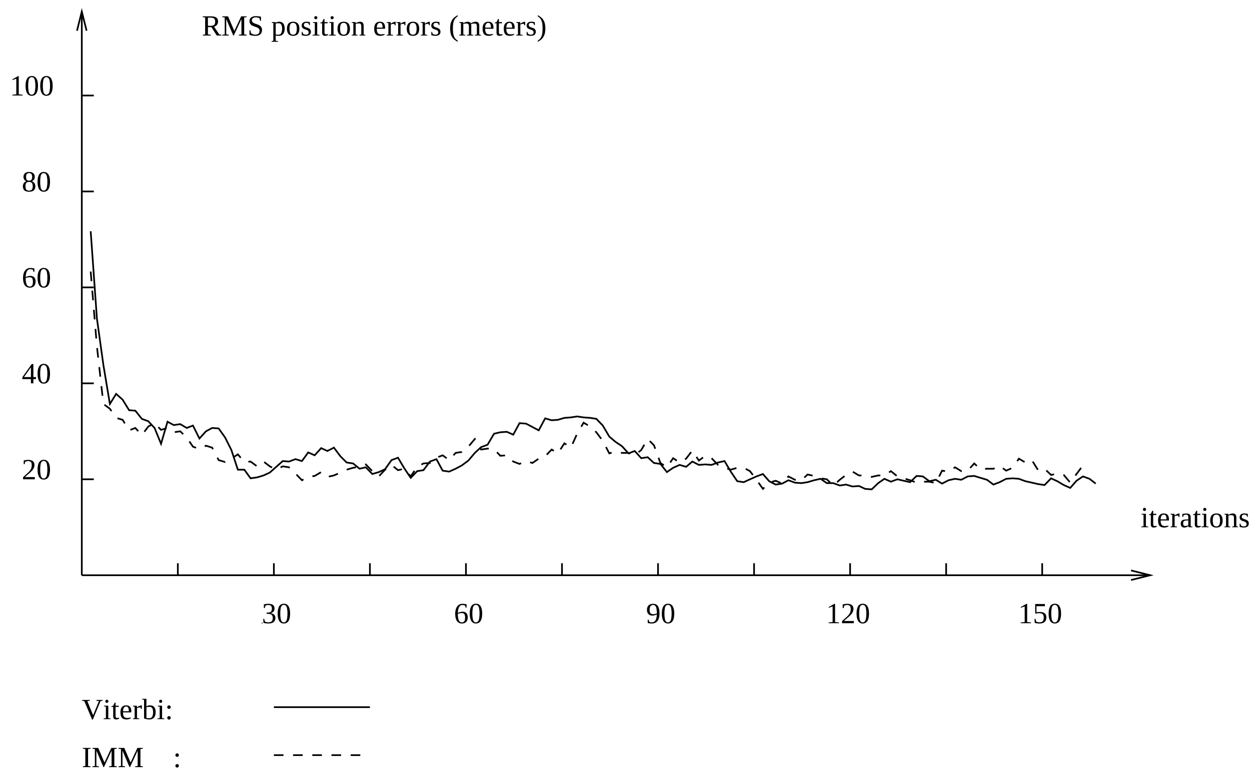

Figure 13: RMS position errors (track #1,$\Delta\tau = 0.1$sec.)

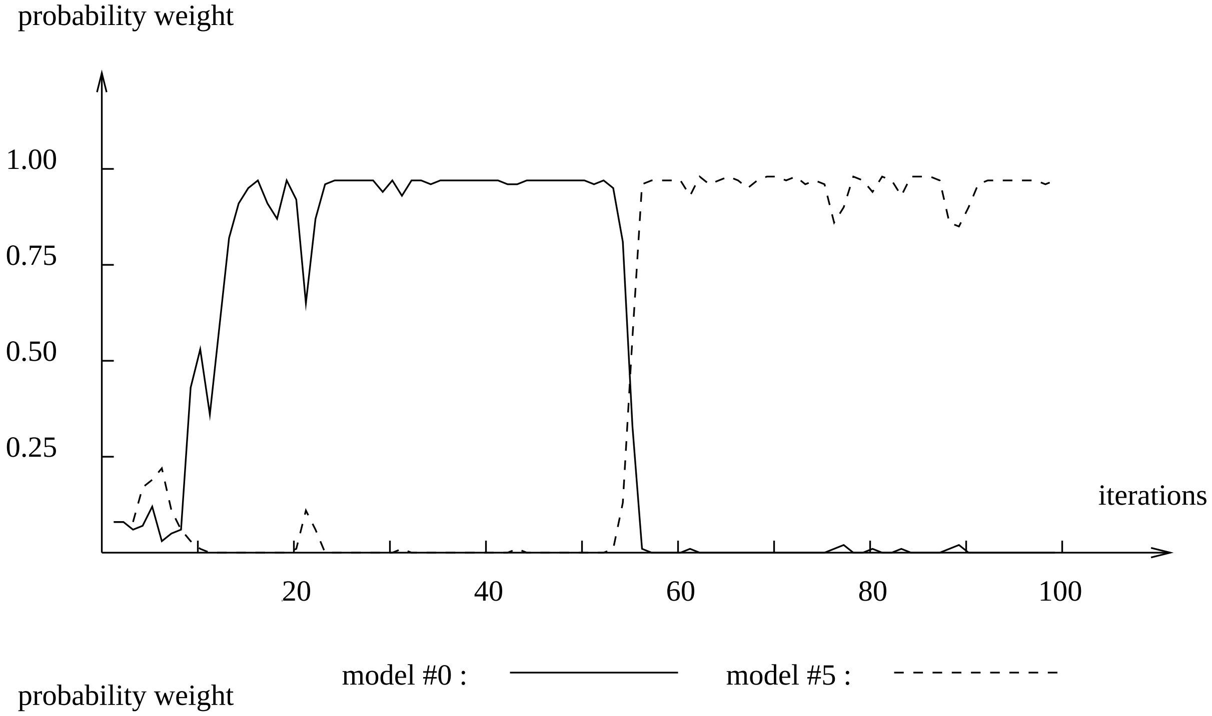

Figure 14: Probability weights in VA (track #2,$\Delta\tau = 1$sec.)

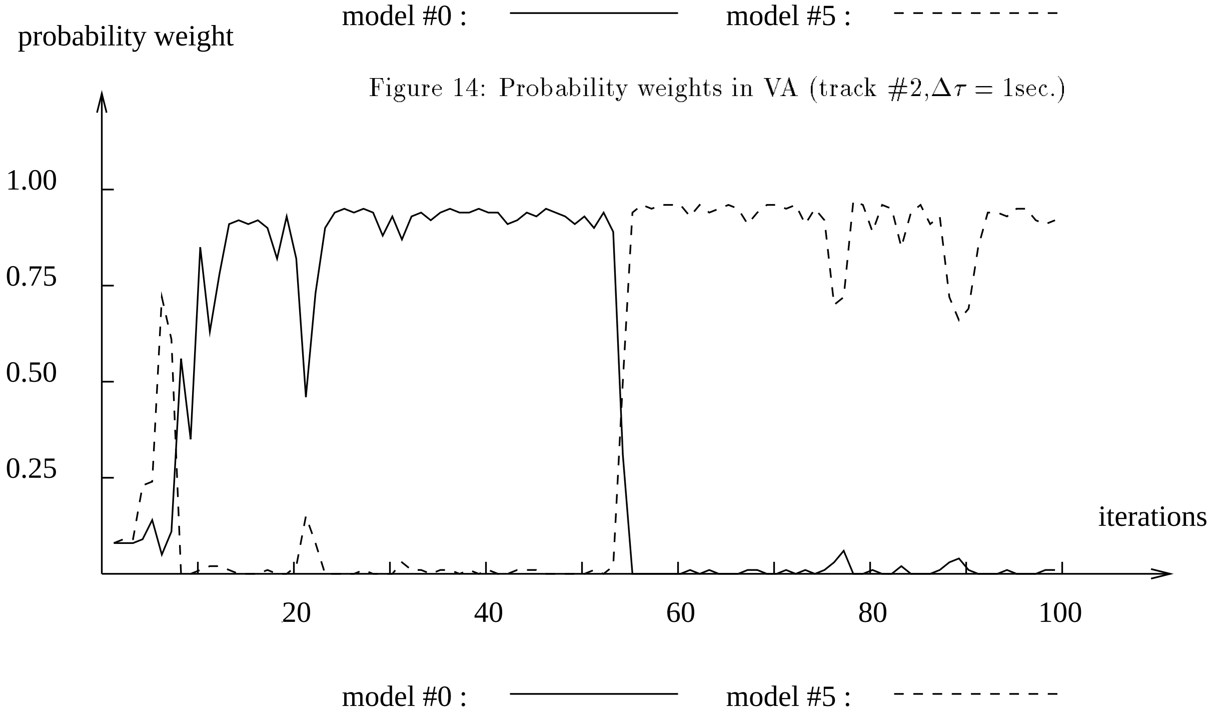

Figure 15: Probability weights in IMM (track #2,$\Delta\tau = 1$sec.)

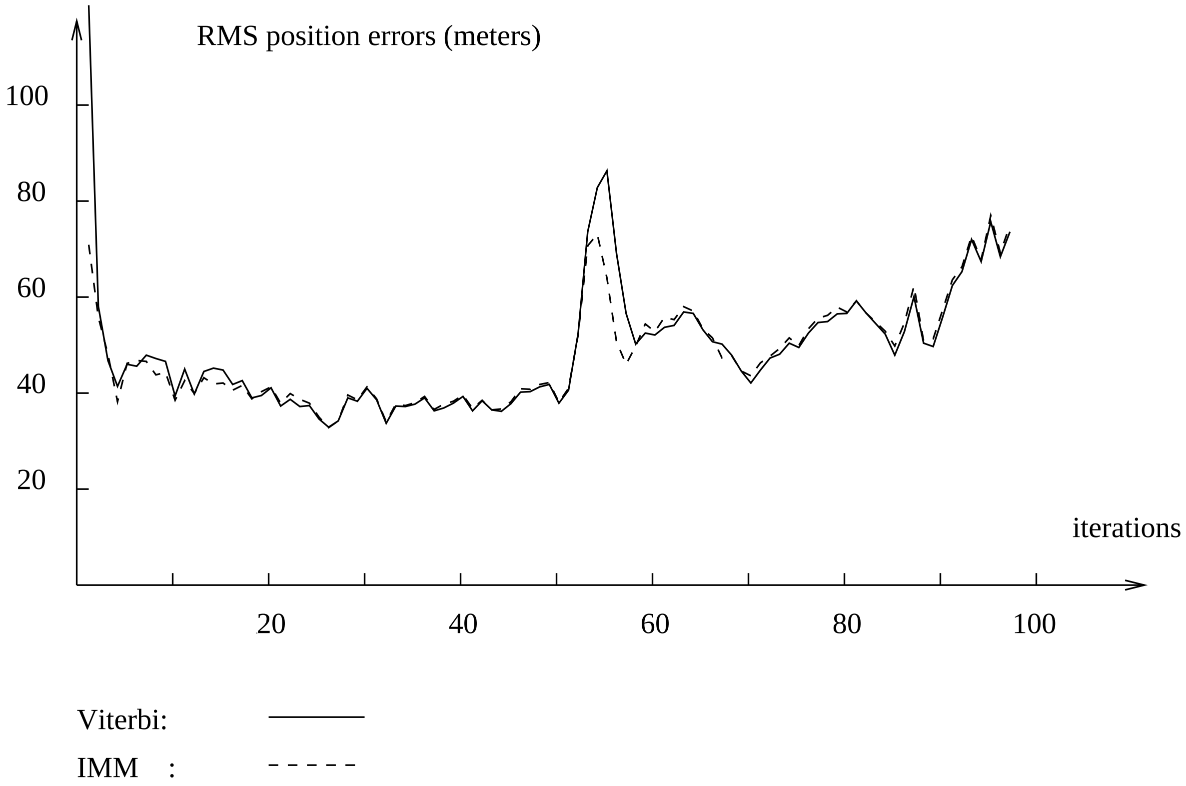

Figure 16: RMS position errors (track #2,$\Delta\tau = 1$sec.)

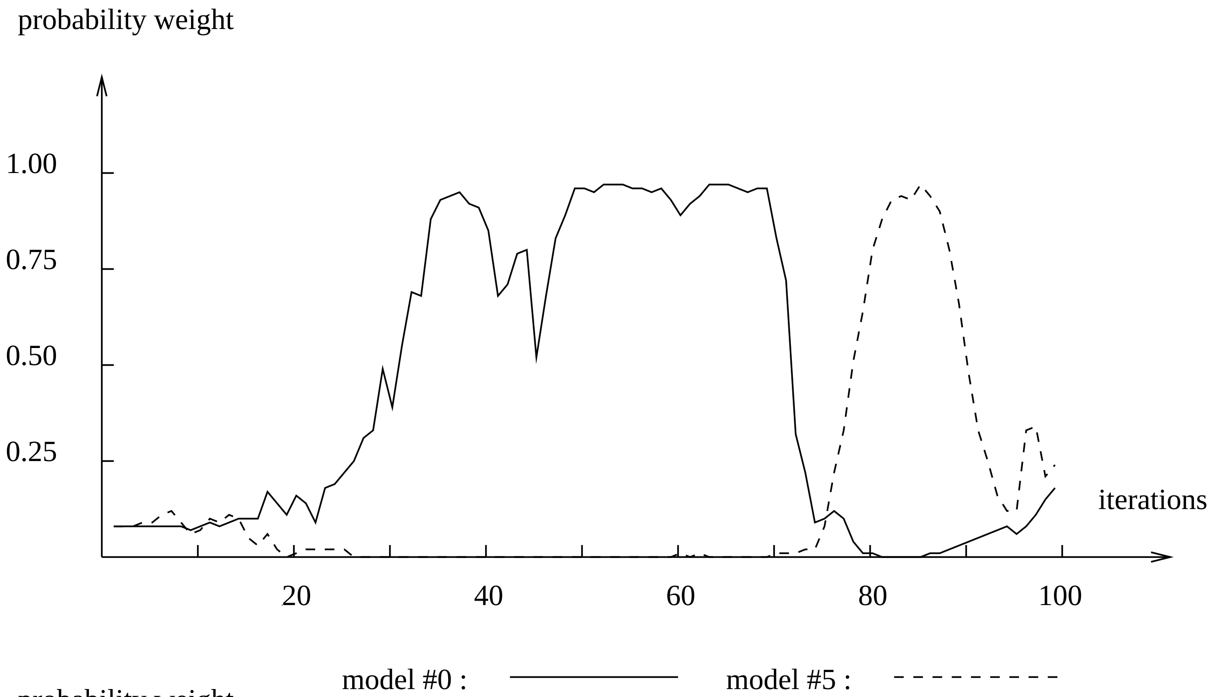

Figure 17: Probability weights in VA (track #2,$\Delta\tau = 0.2$sec.)

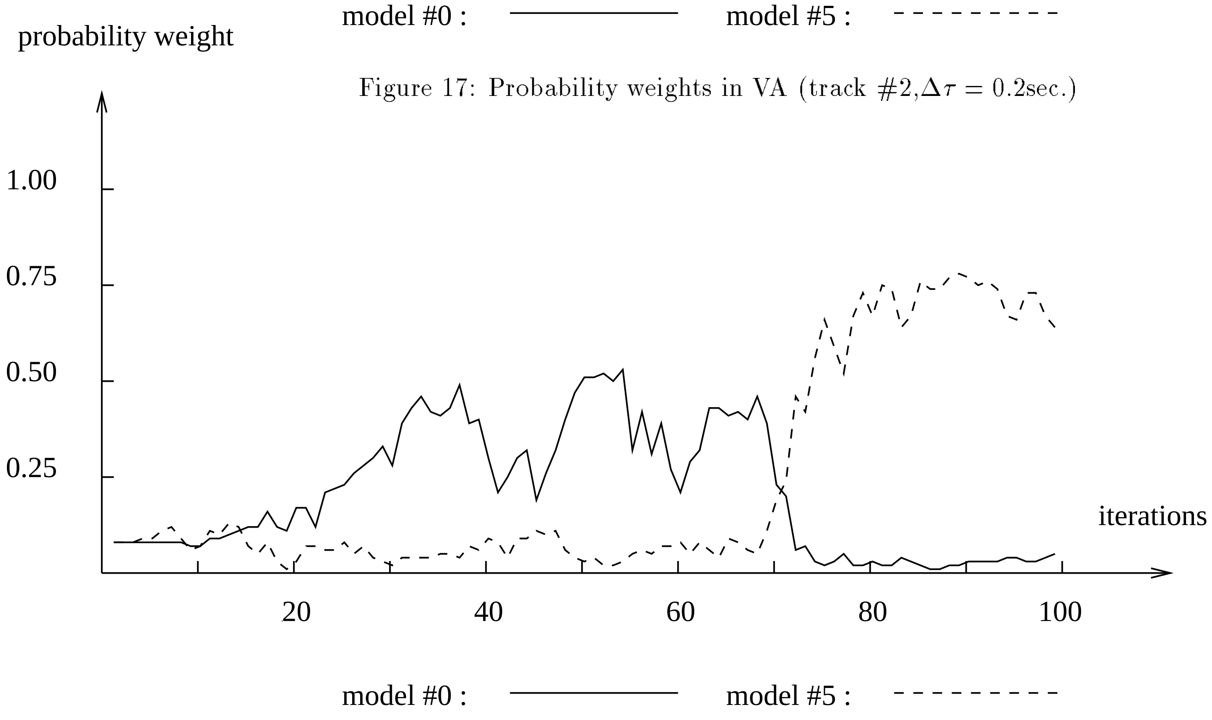

Figure 18: Probability weights in IMM (track #2,$\Delta\tau = 0.2$sec.)

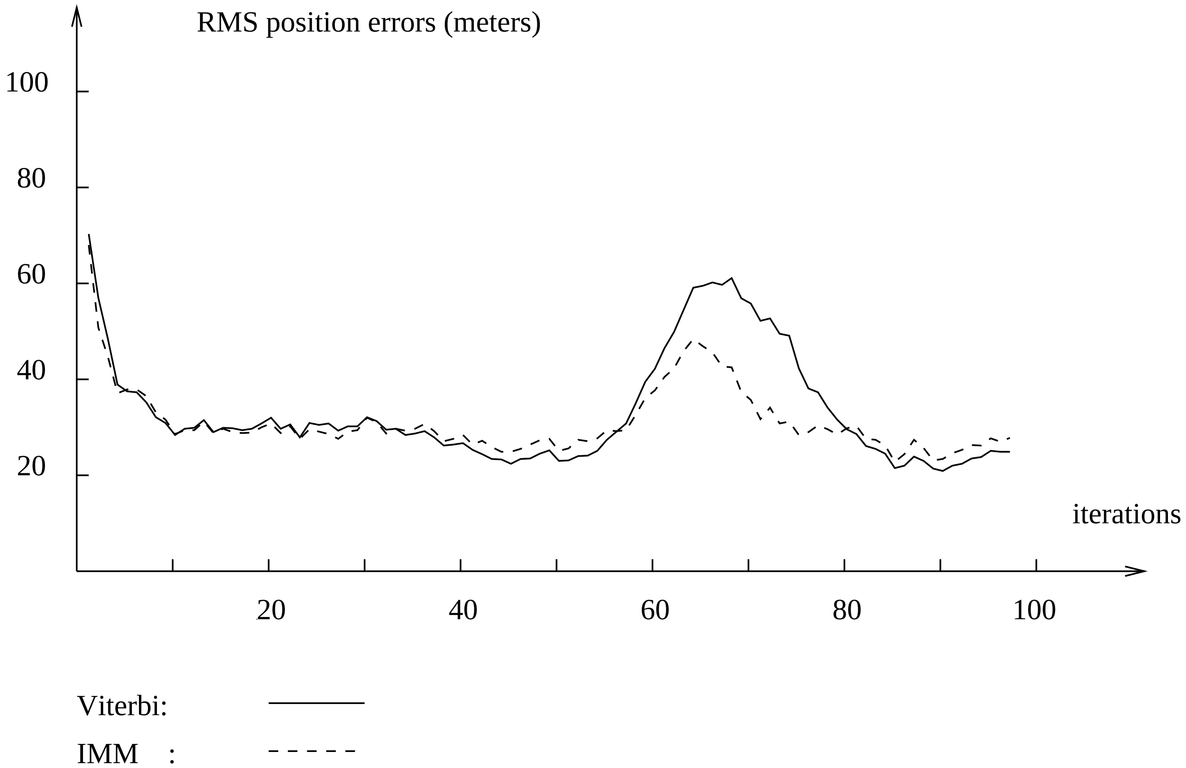

Figure 19: RMS position errors (track #2,$\Delta\tau = 0.2$sec.)

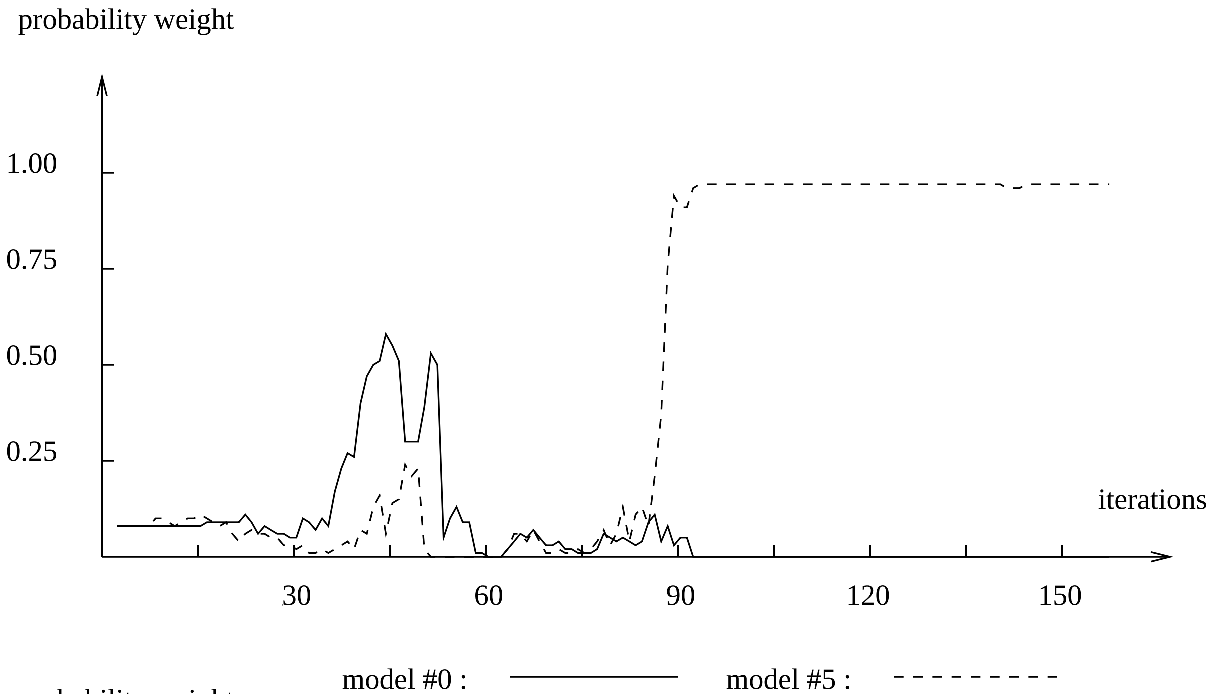

Figure 20: Probability weights in VA (track #2,$\Delta\tau = 0.1$sec.)

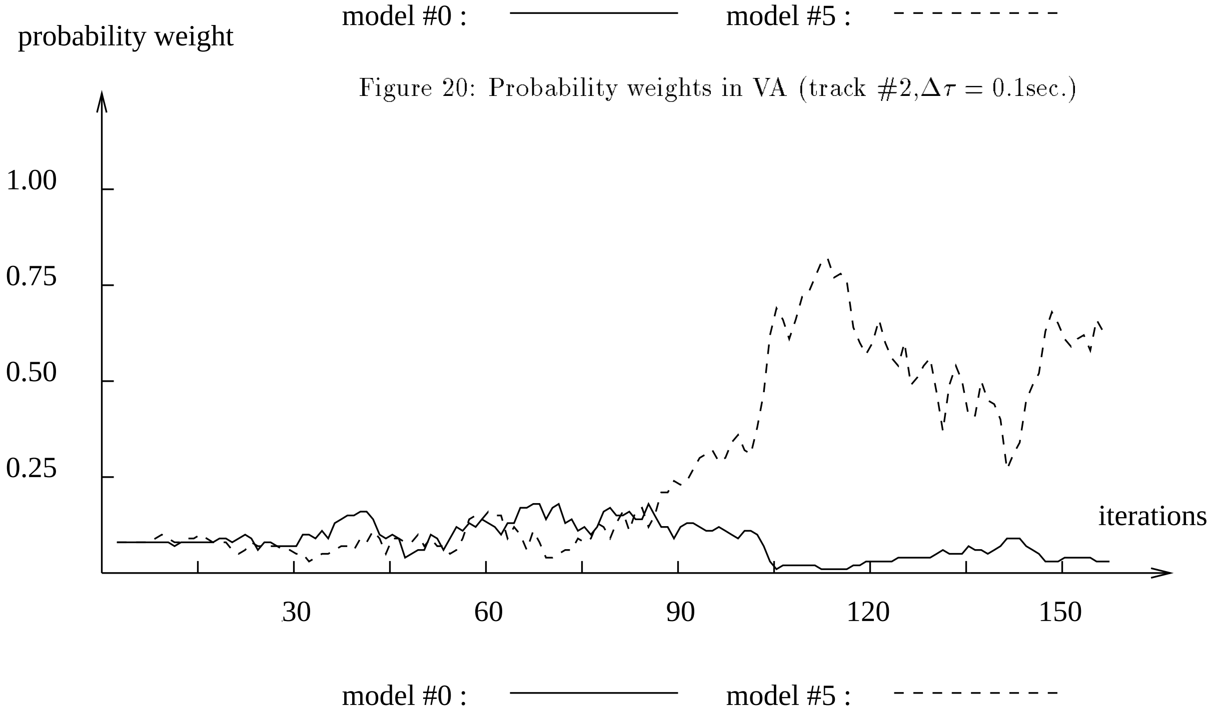

Figure 21: Probability weights in IMM (track #2,$\Delta\tau = 0.1$sec.)

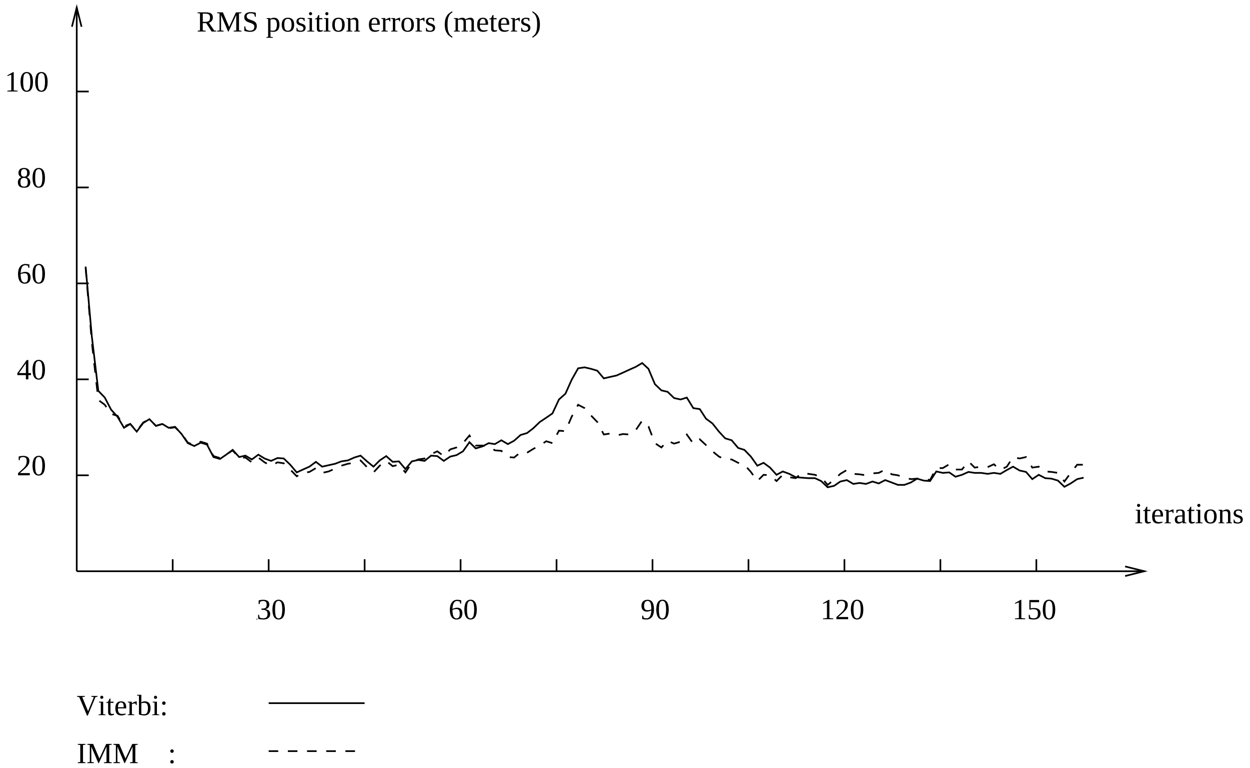

Figure 22: RMS position errors (track #2,$\Delta\tau = 0.1$sec.)

Figure 23: Correlation of position errors(track #1)

Figure 24: Comparison of IMM and VA (track #1 $\Delta\tau = 1.0$sec.)

Figure 25: Comparison of IMM and VA (track #1 $\Delta\tau = 0.2$sec.)

Figure 26: Comparison of IMM and VA (track #1 $\Delta\tau = 0.1$sec.)

D. Analysis of the Results

1) Identifying the Correct Model

In order to analyze the results let us introduce a new definition - that the algorithm identified a model when its probability weight is more than 0.50. We notice from the graphs a clear connection between the parameter η and the ability of the algorithms to identify the correct model. As mentioned before, we kept $|\Delta\vec{a}|$ and σ_r constant while $\Delta\tau$ varies with 3 possible values: 1 sec, 0.2 sec, and 0.1 sec. In Figs. 5,6,14 and 15 we see that when $\Delta\tau = 1$ sec both algorithms identify the correct model decisively. When $\Delta\tau = 0.2$ sec we see (Fig. 8) that VA identifies correctly the models except for an identification problem in track #2 during the time interval 90 - 100 (Fig. 17). IMM is less decisive and the probability weights given to the correct models are significantly smaller than in VA (Figs. 9,18). When $\Delta\tau = 0.1$ sec, it seems that (see Figs. 11,12,20 and 21) the first 50 iterations, which are 5 seconds, are not a sufficient period for neither one of the algorithms to identify the constant velocity model (model #0). After switching to the accelerating model, VA identifies the correct model in both tracks. IMM is again significantly less decisive.

Another important issue is the delay between the time the target starts to accelerate and the time the algorithms identify the new model . Table 2 presents the approximated delay periods in the tested cases.

Table 2: Delay periods in IMM and VA

$\Delta\tau =$	1 sec	0.2 sec	0.1 sec
Delay in Iterations	6	25	35
Delay Time in Seconds	6	5	3.5

We notice that when the time interval becomes smaller, the needed number of iterations increases, but the delay absolute time decreases.

Finally let us note that we see a difference in the identification ability of the algorithms between track #1 and track #2. It can be seen especially in VA that the delay period after switching to a new model is longer in track #2 (compare for example Figs. 8 and 17).

2) Estimation Errors

We observe from the RMS error graphs and the probability weights graphs the following conclusions:

1. When both algorithms decisively identify the correct model, as in the case when $\Delta\tau = 1$ sec, the estimation errors are very similar (see Figs. 7 and 16).

2. When VA has positive identification of the right model, and IMM has less decisive identification there is an advantage of about 10% of the RMS error to VA. That is the case when $\Delta\tau = 0.2$ sec before the maneuver and after the new model is identified, as illustrated by Figs. 10 and 19.

3. When there is a relatively great mismatch between the model identified by the algorithms and the true model, and IMM is again less decisive, there is an advantage of up to 40% of the RMS error to IMM (see Figs. 13 and 22). This is the case right after the start of the maneuver during the delay period, when both algorithms still stick to the old model but the probability weight given to it by IMM is smaller.

Since the position errors in IMM and VA are correlated (see Fig. 23) we can use another comparison scheme suggested by Bar-Shalom in [2] in which we test the hypothesis that:

$$E(\mathrm{ERR_{VA}}) > E(\mathrm{ERR_{IMM}}) \tag{44}$$

where $\mathrm{ERR_{VA}}$ and $\mathrm{ERR_{IMM}}$ are the position estimation errors. The test is based on 50 Monte-Carlo runs. We define:

$$\begin{aligned}
\Delta_i &= \mathrm{ERR_{VA}}^i - \mathrm{ERR_{IMM}}^i \quad i = 1 \ldots 50 \\
\bar{\Delta} &= \frac{1}{50}\sum_{i=1}^{50} \Delta_i \\
\sigma_{\bar{\Delta}}^2 &= \frac{1}{50^2}\sum_{i=1}^{50} (\Delta_i - \bar{\Delta})^2
\end{aligned}$$

Assuming $\bar{\Delta}$ is normal we accept the hypothesis when $\bar{\Delta}/\sigma_{\bar{\Delta}} > \mu$,where μ is a threshold based on a one-sided tail of the normal density. Hence, for example, $\mu = 1.65$ and $\mu = 0$ corresponds tp 95% and 50% confidence, respectively. The results are presented in Figs. 24 - 26 which plot $\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}}$ and confidence versus the number of iterations. The upper part of the graph represents the hypothesis $E(\mathrm{ERR_{VA}}) > E(\mathrm{ERR_{IMM}})$ and the lower side represents the opposite hypothesis. The results here confirm once again our previous conclusions for the first track, namely that there is an advantage to IMM when there isn't a decisive identification of the correct model (i.e. at the beginning of the track and after the start of the maneuver). On the other hand VA is better when the correct model is identified. In a similiar manner, the confidence graphs for the second track (not shown here) would confirm our corresponding previous results.

VI. SUMMARY

This paper investigates and compares the performance of the Interlacing Multiple Model (IMM) algorithm and Viterbi algorithm (VA) when they are applied to the problem of radar tracking of a maneuvering target. We assume a kinematic model in which the the acceleration during a maneuver is almost constant and its mean value is one of a finite set of accelerations. The perturbation upon this constant acceleration is modeled as white zero-mean Gaussian noise. We consider two different tracks and twelve models, each being assigned a specific acceleration value ranging from 0 to $4g$ $(= 40m/s^2)$. In the first track the target's acceleration level almost matches the acceleration level of one of the system models. In the second track, there is a significant difference between the target's acceleration and the acceleration level of the closest model.

It is shown that when the performance depends strongly on the parameter $\eta = \frac{|\Delta\vec{a}|(\Delta\tau^2)}{\sigma_r}$ in which $|\Delta\vec{a}|$ is the maximal difference between the acceleration levels of any two models, $\Delta\tau$ is the time interval bewteen measurements and σ_r is the standard deviation of the measurement error. When the parameter is relatively large, both algorithms identify decisively the model which is closest to the real acceleration level, and provide good position estimates. When η is small, neither one is decisive but VA performs better. During a delay period of the begining of a track and after the start of a maneuver both algorithms are still attached to the old model but the probability weight given to it by IMM algorithm is smaller. Therefore, in such cases, IMM provide better position estimates.

ACKNOWLEDGMENT

The authors wish to thank Dr S. Rogers and Dr I. Hammermann from Elta Corp., Israel, for introducing the problem and continuing support through all the stages of the research.

References

1. Averbuch A., Itzikowitz S. and Kapon T., "Parallel Implementation of Multiple Model Tracking Algorithms", IEEE Trans. on Parallel and Distributed Systems, Vol. 2, No. 2, pp. 242-252, April 1991.

2. Bar-Shalom Y. and Fortman T.E., "*Tracking and data association*", New York : Academic Press, 1986.
3. Blom H.A.P. "An efficient filter for abruptly changing systems", Proc. of the 23rd Conf. on Decision and Control, Las Vegas NV, Dec 1984.

4. Blom H.A.P. "Overlooked potential of systems with Markovian coefficients", Proc.of the 25th Conf. on Decision and Control, Athens Greece, Dec 1986.
5. Blom H.A.P and Bar-Shalom Y., "The interacting multiple model algorithm with Markovian switching coefficients", IEEE Tran. on Automatic Control, Vol AC-33, No 8,pp 780-783, Aug 1988.
6. Chang C.B. and Athans M., "State estimation for discrete systems with switching parameters", IEEE Tran. on Aerospace and Electronic Systems, Vol AES-14, No 3, pp.418-424, May 1978.
7. Forney G.D.Jr., "The Viterbi algorithm", Proc. of the IEEE, Vol 61, No. 3 March 73.
8. Gabber E., "Parallel programming using the MMX operating system and its processor", Technical Report 107/88,Tel Aviv University, May 1988.
9. Gholson N.H. and Moose R.L., "Maneuvering target tracking using adaptive state estimation" , IEEE Tran. on Aerospace and Electronic Systems, Vol AES-13, No 3 , pp.310-317 , May 1977.
10. Mookerjee P., Campo L. ,and Bar-Shalom Y., "Estimation in systems with semi-markov switching model" , Proc. of the 26th Conf. on Decision and Control, Los Angeles CA, 1987.
11. Rabiner L.R and Juang B.H, "An introduction to hidden Markov models", IEEE ASSP magazine, pp. 4-16, Jan 1986.
12. Schnepper K., "A comparison of GLR and multiple model filters for target tracking problem", Proc. of the 25th Conf. on Decision and Control, Athens, Dec 1986.
13. Tugnait J.K., "Detection and estimation for abruptly changing systems", Automatica, Vol 18, No 5, pp.607-615, 1982.
14. Tugnait J.K. and Haddad A.H., "A detection estimation scheme for state estimation in switching environments", Automatica , Vol 15, pp.477-481, 1979.
15. Willsky A.S., "Detection of abrupt changes in dynamic systems" in *Detection of abrupt changes in signals and dynamical systems (Edt.)*, Bassenville and Beneviste, Heidelberg : Springer-Verlag, 1985.