MULTIPLE MODEL PARTICLE FILTERING FOR MULTI-TARGET TRACKING

Chris Kreucher[a,b], *Alfred O. Hero III*[a], *Keith Kastella*[b]

[a]The University of Michigan, Dept. of EECS
1301 Beal Avenue, Ann Arbor, MI 48109-2122
ckreuche@umich.edu, hero@eecs.umich.edu

[b]General Dynamics Advanced Information Systems
P.O. Box 134008, Ann Arbor, MI 48113-4008
Keith.Kastella@gd-ais.com

ABSTRACT

This paper addresses the problem of tracking multiple moving targets by recursively estimating the joint multitarget probability density (JMPD). Estimation of the JMPD is done in a Bayesian framework, providing a method of tracking multiple targets which allows nonlinear target motion, nonlinear measurement to state coupling, and non-Gaussian target state densities. We utilize a particle filter implementation which has been detailed elsewhere [1].

Real targets are poorly described by a single kinematic model. Target behavior may change dramatically – e.g. targets stop moving or begin rapid acceleration. In the literature, the Interacting Multiple Model (IMM) algorithm [4] is used to address this. The IMM uses multiple models for target behavior and adaptively determines which model(s) are the most appropriate at each time step.

We demonstrate the IMM in the context of our PF based multitarget tracker in two settings. First, we consider application to targets that switch between kinematic modes. The target motion used is field data recorded during a military battle simulation and includes multiple modes of target behavior. Second, we present a nontraditional application of IMM as multiple models on the state of the filter. In the context of PF based target tracking, this technique may be viewed as a (biased) sampling scheme for particle proposal. This strategy adds robustness to the tracker as it is able to automatically detect model violations and compensate by altering the filter model.

This work was supported under the USAF contract F33615-02-C-1199, AFRL contract SPO900-96-D-0080 and by ARO-DARPA MURI Grant DAAD19-02-1-0262. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

1. INTRODUCTION

The goal of target tracking is to estimate the state of a target using a model of target kinematics, a probabilistic model of a sensor, and a set of noisy measurements. Since real targets are poorly described by a single kinematic model, researchers have developed the Interacting Multiple Model (IMM) target tracker [4] and variants such as VS-IMM [6].

The IMM characterizes a target as behaving according to one of M modes (e.g. stopped, moving with constant velocity, or accelerating). Each mode has an associated probability. Transition rates between modes (e.g the probability that a moving target stops) are defined *a priori*. As new data comes in, mode probabilities adjust based on agreement with measurements. The goal is to correctly estimate mode probabilities to minimize tracking error.

This paper contains two contributions. First, we investigate the IMM in a multi-target tracking environment where target motion is taken from real recorded data. Using a multitarget particle filter with IMM, we investigate the tradeoff between adaptation time and steady state error. Second, we investigate a new application of the IMM, where the *state of the filter* is modeled rather than the state of the target. In the context of particle filter based target tracking, this can be interpreted as having multiple (biased) proposal schemes as the models. We show via simulation that this strategy adds robustness to the filter, keeping targets in track more often than otherwise.

The paper proceeds as follows. In Section 2, we briefly review Bayesian multitarget tracking and the standard particle filter based implementation. In Section 3, we outline the IMM strategy. In section 4, we give an example of the IMM-particle filter (IMMPF) applied to the problem of tracking two targets that can each be modeled as behaving according to one of 2 modes – stopped and moving. This is the regime in which the IMM is typically applied, although most of

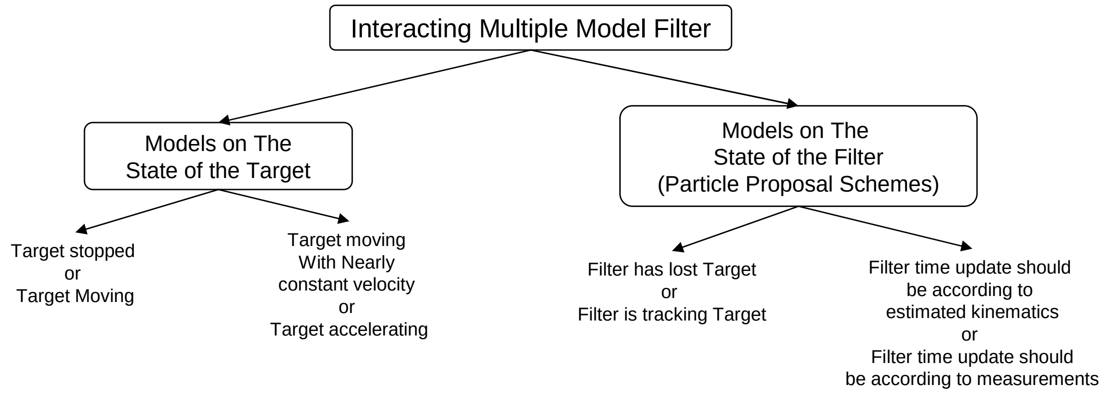

Figure 1: A top-level view of the two interpretations of the Interacting Multiple Model strategy considered here.

the literature uses the IMM algorithm in conjunction with a Kalman filter tracker. In Section 5, we give a new application where the modes are associated with the filter rather than the target. In this application, the IMM estimates the state of the filter rather than the target.

2. BAYESIAN MULTITARGET TRACKING AND PARTICLE FILTERING

We track a collection of moving targets by recursively estimating the Joint Multitarget Probability Density (JMPD). We restrict ourselves to the case where the number of targets is known and fixed although the framework is general.

The statistical model uses the joint multitarget conditional probability density $p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k|\mathbf{Z}^k)$ as the probability for T targets with states $\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k$ at time k based on a set of observations $\mathbf{Z}^k$. $\mathbf{Z}^k$ refers to the collection of measurements up to and including time k, i.e. $\mathbf{Z}^k = \{\mathbf{z}^1, \mathbf{z}^2, ...\mathbf{z}^k\}$, where each of the $\mathbf{z}^i$ may be a single measurement or a vector of measurements made at time i. Each of the state vectors $\mathbf{x}_i$ is a vector quantity and may (for example) be of the form $[x, \dot{x}, y, \dot{y}]'$. For convenience, the density will be written more compactly as $p(\mathbf{X}^k|\mathbf{Z}^k)$, where $\mathbf{X} = [\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k]$.

The temporal update of the posterior likelihood on this density proceeds according to the usual rules of Bayesian filtering. Given a model of how the JMPD evolves over time $p(\mathbf{X}^{k+1}|\mathbf{X}^k)$, we compute the time-updated or prediction density via marginalization of a conditional density:

$$p(\mathbf{X}^{k+1}|\mathbf{Z}^k) = \int d\mathbf{X}^k p(\mathbf{X}^{k+1}|\mathbf{X}^k) p(\mathbf{X}^k|\mathbf{Z}^k) \quad (1)$$

$p(\mathbf{X}^{k+1}|\mathbf{Z}^k)$ is referred to as the prior or prediction density at time $k+1$, as it is the density at time $k+1$ conditioned on measurements up to and including time k.

Given a model of the sensor, $p(\mathbf{z}|\mathbf{X}^k)$, Bayes' rule is used to update the posterior density as a new measurement vector $\mathbf{z}$ arrives at time $k+1$ via

$$p(\mathbf{X}^{k+1}|\mathbf{Z}^{k+1}) = \frac{p(\mathbf{z}|\mathbf{X}^{k+1}) p(\mathbf{X}^{k+1}|\mathbf{Z}^k)}{p(\mathbf{z}|\mathbf{Z}^k)} \quad (2)$$

$p(\mathbf{X}^{k+1}|\mathbf{Z}^{k+1})$ is referred to as the posterior or the updated density at time $k+1$ as it is the density at time $k+1$ conditioned on all measurements up to and including time $k+1$.

The sample space of $\mathbf{X}$ is very large. It contains all possible configurations of state vectors $\mathbf{x}_i$. We find that a particle filter based representation of the JMPD allows tractable implementation [1]. The particle filter approximation represents the JMPD by a collection of weighted samples, i.e.

$$p(\mathbf{X}|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} w_p \delta(\mathbf{X} - \mathbf{X}_p) \quad (3)$$

Particle filtering is a method of approximately solving the prediction and update equations (1) and (2) by simulation [5]. Samples are used to represent the density and to propagate it through time. The prediction equation (1) is implemented by proposing new particles from the existing particles using a model of state dynamics and the measurements. The update equation (2) is implemented by assigning a weight to each of the particles that have been proposed using the measurements and the model of state dynamics.

Of particular interest in this work is the best way to propagate samples through time (simulate (1)). As real targets have time varying kinematic modes, the traditional IMM seeks to estimate which of the modes the target is following and use this to time evolve (predict) the density. Here we extend this to allow additional methods of time evolution which are related to the state of the filter rather than the state of the target. This allows the filter to detect model violations when measurements are inconsistent with the current method of time evolution and compensate.

3. MULTIPLE MODEL TARGET TRACKING

In this section, we outline the IMM algorithm [6]. For simplicity, we give details for a single target. Extension to multiple targets is straightforward.

Real targets rarely obey a single kinematic model. The IMM algorithm estimates on-line the target mode, and uses it for filtering. The designer selects a set of M models or modes $m = 1 \cdots M$ that represent all possible priors on motion of the target (e.g. stopped, accelerating, performing a coordinated turn). Associated with each model m is the mode probability (probability the target is following this mode at the current time). At initialization, mode probabilities are given based on prior knowledge. While the filter tracks the target, mode probabilities are continuously re-estimated online.

The target mode is assumed to evolve in a Markov fashion, specified *a priori* by transition probabilities π_{ij} between target mode i and j. Sensor measurements allow the filter to update the estimate of the mode probabilities at each time step. A sub-filter is associated with each of the M modes. The sub-filters estimate the state $\mathbf{x}$ conditioned on both the measurements $\mathbf{Z}$ and the mode i, i.e. the i^{th} sub-filter estimates $p_i(\mathbf{x}|\mathbf{Z})$.

When a particle filter is used as the target tracker, the IMM algorithm is especially simple. Each particle is expanded to contain a mode estimate for each target. The particle is propagated forward in time according to the dynamics implied by the modes of the targets. Transitions between modes happen for each target according to π. The weighting and resampling process work to reinforce modes that are in agreement with measurements at the expense of those that are not. Specifically, for each particle at time k (which contains an estimate of the mode m^k and state $\mathbf{x}^k$) we propose a particle at time $k+1$ according to Table 1.

Table 1: Generic IMM Particle Filter Propogation

<u>Time Update</u>

- Select the mode : $m^{k+1} \sim \pi_{m^k, m^{k+1}}$
- Propose target state : $\mathbf{x}^{k+1} \sim q_{m^{k+1}}(\mathbf{x}^{k+1}|\mathbf{x}^k, \mathbf{z})$

<u>Measurement Update</u>

- Update weight : $w^{k+1} = w^k \frac{p(\mathbf{z}|\mathbf{x}^{k+1})p(\mathbf{x}^{k+1}|\mathbf{x}^k)}{q(\mathbf{x}^{k+1}|\mathbf{x}^k,\mathbf{z})}$

The important issue for efficient particle filtering is the choice of importance density q. It is known that the optimal importance density is typically intractable to use for particle proposal [5]. We study here two methods of particle proposal, both of which use the IMM as control logic. In the first method (Section 4) proposals are always made using target kinematics (as is commonly done in the literature) and the IMM is used to estimate which of the kinematic models the target is following at each time step. In the second method (Section 5) proposals are made in a more generic way, allowing arbitrary forms of q, again controlled by the IMM.

4. MULTIPLE MODELS ON THE TARGET STATE

Here we consider the traditional application of the IMM, tracking a target that switches between kinematic modes. We specialize to the case where the filter has $M = 2$ models: target stopped and target moving. The filter estimates the probability the target is stopped and the probability the target is moving for each target.

Particles are always proposed using the target kinematics. Different particles may have different estimates of target mode and hence different kinematic priors. This gives $q(\mathbf{x}^{k+1}|\mathbf{x}^k, \mathbf{z}) = p(\mathbf{x}^{k+1}|\mathbf{x}^k)$ in Table 1, leading to a simple form for the weight update, $w^{k+1} = w^k p(\mathbf{z}|\mathbf{x}^{k+1})$.

4.1. Description of Simulation

Two targets move in a surveillance area. At each time step, measurements of the entire region are made from two sensors. Sensor A measures the area with a moving target indicator, characterized by detection probability $P_d^{MTI}(O)$ and false alarm probability $P_f^{MTI}(O)$. O indicates occupation of a cell, i.e. the number of targets in the cell. Sensor B measures the area with a fixed (stopped) target indicator and is characterized by $P_d^{FTI}(O)$ and $P_f^{FTI}(O)$. Both sensors make thresholded measurements on a fixed grid. Target motions in the simulation are taken from real recorded data. The filter in the simulation is the IMMPF with two modes: target stopped and target moving with constant velocity.

The modes are distinguished by their kinematic (model) updates. The target moving mode has a model given by $p(\mathbf{x}^{k+1}|\mathbf{x}^k) \sim N(\mathbf{F}x^k, \mathbf{Q})$, i.e. normally distributed with vector mean $\mathbf{F}x^k$ and covariance $\mathbf{Q}$. $\mathbf{F}$ performs the deterministic update and $\mathbf{Q}$ models uncertainty that accumulates during the discrete time interval. $\mathbf{F}$ and $\mathbf{Q}$ were fit to the target motion using a training set of targets. The target stopped mode uses $p(\mathbf{x}^{k+1}|\mathbf{x}^k) \sim \delta(\mathbf{x}^{k+1} - \mathbf{x}^k)$. These modes constitute q_1 and q_2 in Table 1.

We study the trade between adaption time and steady state error. The parameters that control this trade are in π, where $\pi = \begin{pmatrix} p_{\text{moving to moving}} & p_{\text{moving to stopped}} \\ p_{\text{stopped to moving}} & p_{\text{stopped to stopped}} \end{pmatrix}$. If π allows probability to flow from one mode to another rapidly (i.e.

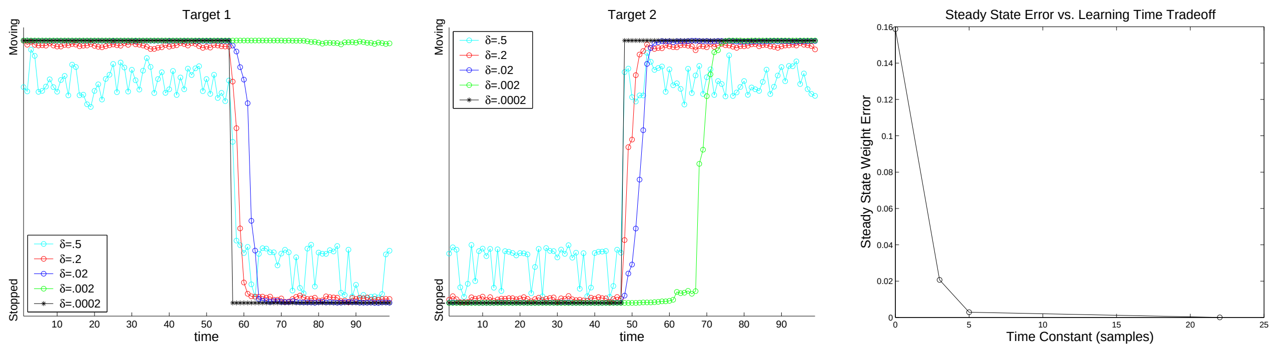

Figure 2: Left two plots: Performance of the IMM particle filter for different values of the adaption parameter, δ. True target mode is indicated by the dark line. Both targets change modes (i.e. stop or start moving) during the simulation. The filter estimates on-line mode probabilities, and the estimate is plotted. For large values of adaption parameter (e.g. $\delta = .2$) reaction time is fast, but steady state error is large. Conversely, for small values (e.g. $\delta = .002$) reaction time is slow, but steady state error is small. Rightmost curve: A plot of steady state error versus time constant parameterized by δ for Target 2.

off-diagonal elements are large), the filter adapts quickly but has poor steady state behavior. Conversely, slow adaptation corresponds to good steady state behavior. In this experiment we consider π of the form $\pi = \begin{pmatrix} 1-\delta & \delta \\ \delta & 1-\delta \end{pmatrix}$. δ plays the role of an adaption speed parameter.

4.2. Simulation Results

We show the performance of the IMM at estimating target modes for targets 1 and 2 as a function of adaptation parameter δ in Figure 2. Target 1 starts out moving and then stops at time step 58. Target 2 starts out stopped and then starts moving at time step 48.

The results show a tradeoff between steady state error and adaption time. For example, consider the curves (corresponding to different values of δ) for Target 2. We see that for small δ (e.g. $\delta = .0002$), the filter is very slow to adapt but has very low steady state error. Conversely, for larger δ (e.g. $\delta = .2$), the filter is very quick to adapt but has large steady state error. We summarize this trade in Figure 2, which shows a plot of steady state error versus time constant parameterized by the adaption parameter δ. The actual transition rate from studying the true target trajectory (which is not available to the filter) is $\delta \approx .02$.

5. MULTIPLE MODELS ON THE FILTER STATE

An alternate application of the IMM strategy is multiple models on the state of the filter. Here we model transitions in tracking error rather than in kinematic behavior of the target. It is straightforward to combine models relating to the filter and models relating to the target but we do not pursue that here. We find that this approach adds robustness to the filter as it allows the filter to automatically detect a model violation and compensate by adjusting the filter.

In the context of a particle filter tracker, using the IMM to model the filter state may be interpreted as a biased sampling scheme for particle proposal. Specifically, targets are proposed from a mixture importance density, q:

$$q(\mathbf{x}^{k+1}|\mathbf{x}^k, \mathbf{z}^k) = \begin{cases} q_1(\mathbf{x}^{k+1}|\mathbf{x}^k) & \text{with prob. } \beta \\ q_2(\mathbf{x}^{k+1}|\mathbf{x}^k) & \text{with prob 1-}\beta \end{cases} \quad (4)$$

i.e. samples are drawn from q_1 with probability β and q_2 with probability $1 - \beta$. This biassed sampling is accounted for in the weight update of the particles [5]

$$w_p^{k+1} \propto \frac{p(\mathbf{z}|\mathbf{x}_p^{k+1})p(\mathbf{x}_p^{k+1}|\mathbf{x}_p^k)}{\beta q_1(\mathbf{x}_p^{k+1}|\mathbf{x}_p^k, \mathbf{z}) + (1-\beta) q_2(\mathbf{x}_p^{k+1}|\mathbf{x}_p^k, \mathbf{z})} \quad (5)$$

One should not get the impression that this interpretation is wedded to a particle filter implementation. It is as implementation independent as the traditional IMM (recall most of the research on the traditional IMM has been done in the context of Kalman Filter tracking). One can envision an IMM Kalman Filter tracker where one models the filter rather than the target in exactly the manner discussed here.

5.1. Description of Simulation I

We use two models of filter mode: “target in track” and “target lost”. The filter estimates on-line the probability that the target is being successfully tracked (model obeyed) and the probability that the target has been lost (model violation).

The first model (target in track) the kinematics of the target is used to update the filter, $p(\mathbf{x}^{k+1}|\mathbf{x}^k) \sim N(\mathbf{F}x^k, \mathbf{Q})$.

Targets get lost by the filter in the following manner. A series of missed detections or unlikely maneuvers cause

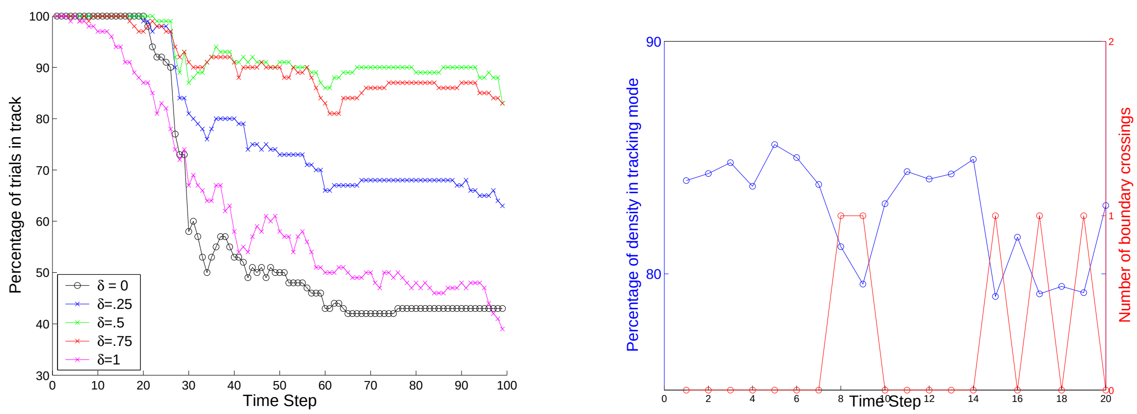

Figure 3: Performance of the IMM preventing lost targets. Left: Percent of trials the target is successfully tracked for different choices of δ. $\delta = 0$ corresponds to always using tracking mode and $\delta = 1$ corresponds to always using searching mode. We find that $\delta = .5$ performs best. Right: Percent of density in tracking mode and the number of boundary crossings that occur at each time step. The sensor makes detection on a fixed grid, so when a target crosses a boundary, the chance of getting lost is greatest. It is these occasions that the mode probabilities move to favor search mode over tracking mode.

particles (mass of the target state density) to become concentrated in an area where the target is not present. At that point, the kinematic model is insufficient to allow the density to flow back to the proper area of state space. The target is then lost forever. At the point where the target has just been lost, it is critical to recapture or risk being out-of-track for good. To address this situation, we include a second model given by $p(\mathbf{x}^{k+1}|\mathbf{x}^k) \sim N(\mathbf{F}x^k, \mathbf{Q}_{searching})$ which has a large diffusive component $\mathbf{Q}_{searching}$. In times of model violation, this second model should be more heavily used and the mass of the state density spread throughout state space quickly. Practically speaking, particles should be diffused more quickly from their nominal location. These modes constitute q_1 and q_2 in Table 1.

We study a difficult scenario, consisting of low SNR measurements, a small number of particles, and a target that moves erratically (i.e. has large $\mathbf{Q}$). Again, there is an adaptivity parameter δ which controls how readily the filter switches modes. A critical distinction in this setting is that the parameter no longer has a direct physical interpretation with respect to the targets. The transition matrix is $\pi = \begin{pmatrix} p_{\text{tracking to tracking}} & p_{\text{tracking to searching}} \\ p_{\text{searching to tracking}} & p_{\text{searching to searching}} \end{pmatrix}$. We choose to use a π of the form $\pi = \begin{pmatrix} 1-\delta & \delta \\ 1-\delta & \delta \end{pmatrix}$. δ will control how readily switches out of tracking mode and into searching mode.

5.2. Results of Simulation I

We show in Figure 3 algorithm performance (percentage of trials the target was in track) versus times for several different choices of the adaptivity parameter δ. $\delta = 0$ corresponds to using the tracking model all of the time while $\delta = 1$ corresponds to using the searching model all of the time. We see that $\delta = .5$ outperforms both $\delta = 0$ and $\delta = 1$.

Unlike the earlier situation wherein the mode probabilities eventually reached steady state of $[1, 0]$ or $[0, 1]$ corresponding to moving or stopped mode, we find a different steady state behavior here. There is always some mass in each of the searching and tracking modes. In Figure 3 we also show the mode probabilities versus time and we plot the number of boundary crossings at each time step. The sensor is pixelated and makes detections on a grid. The most likely place to lose targets is when the target moves from one sensor cell to another. We see that the probabilities are adjusted to give more mass to the searching mode at precisely these occasions, stabilizing the filter.

5.3. Description of Simulation II

In this simulation we consider two filter modes: a mode where the filter biasses proposals towards target kinematics and a mode where the filter biasses proposals towards the measurements. The first mode should be used if the filter estimates that its model of target kinematics is good as compared to the measurements it is receiving (e.g. the SNR is low). The second model should be used if the filter estimates it is in the opposite situation.

We make use of the mixing parameter β (in (5), which controls how readily the filter uses each of the two modes. This parameter is analogous to the adaption parameter in earlier simulations as it controls switching between modes. We wish to determine how filter performance is effected by choice of β. As $\beta \to 1$, the filter uses the kinematics exclusively when evolving the target state density through time (i.e. ignores the measurements). As $\beta \to 0$, the filter

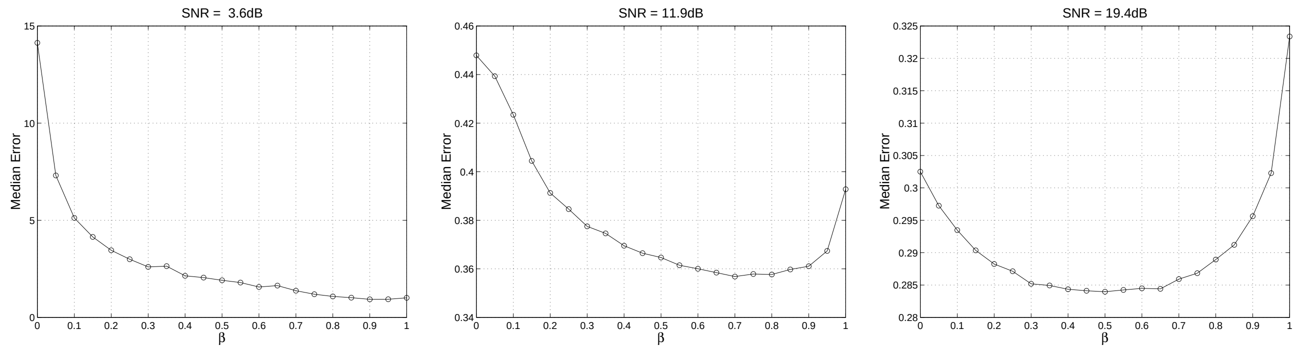

Figure 4: Tracking error as a function of β for low, medium and high SNR (SNR known to the filter). As $\beta \to 1$, the filter uses vehicle kinematics exclusively to time update the density. Conversely, as $\beta \to 0$, the filter uses measurements exclusively to time update the density. As is seen in the figures, for low SNR measurements $\beta \approx 1$ yields the best tracking performance (lowest tracking error). On the other hand, for high SNR measurements, $\beta \approx 0.5$ yields the best tracking performance.

uses the measurements exclusively when evolving the target state density through time (i.e. ignores the target kinematics). Of course, both the measurements and the kinematics are always used when weighting the particles, in the manner given by eq. (5).

5.4. Results of Simulation II

Figure 4 shows results versus β of tracking simulations in three situations: low, medium and high SNR. For the low SNR case, the best performance occurs with $\beta \approx 1$, which implies that the measurements are ignored when evolving the density through time. This is consistent with the fact that the measurements are poor and ought not be allowed to unduly influence the propagation of the density. Conversely, in the high SNR case, $\beta \approx 0.5$ yields the best tracking performance. This implies that roughly half of the particles should be proposed using the kinematics and half from the measurements. Since the measurements are very reliable, using them to bias particle proposals leads to improved performance.

6. CONCLUSIONS

We have investigated the use of the IMM algorithm in the setting of particle filter based multitarget tracking. First, we considered application to targets that switch between kinematic modes. Second, we presented experiments where the IMM uses multiple models on the state of the filter rather than on the state of the target. In the context of particle filter target tracking, this technique may be viewed as a biased sampling scheme for particle proposal. Through simulation, we showed that this strategy adds robustness by helping to prevent the filter from losing targets.

The approach we take has the merit of a unifying framework of Bayesian posterior propagation of a multiple target state vector given noisy measurements. We have previously demonstrated that JMPD provides a reliable tracking capability in a fully Bayesian setting [1]. This paper goes further in illustrating the benefits in using multiple models for targets whose kinematics may be very different and therefore do not obey the same linear diffusion model.

7. REFERENCES

[1] C. Kreucher, K. Kastella, and A. Hero, "Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density," SPIE International Symposium on Optical Science and Technology, San Diego, California, August 2003.

[2] K. Kastella and C. Kreucher, "Multiple model nonlinear filtering for low signal ground target applications", under review at IEEE Transactions on AES.

[3] Y. Bar-Shalom, *Multitarget Multisensor Tracking: Advanced Applications*, Artech House, 1990.

[4] S. S. Blackman, *Mulitple-Target Tracking with Radar Applications*, Massachusetts: Artech House, 1986.

[5] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, New York: Springer Publishing, 2001.

[6] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking-part I: dynamic models," IEEE Transactions on AES, 39(4): 1333-1364, October 2003