

Improved Target Tracking with Road Network Information

Umut Orguner

Division of Automatic Control
Department of Electrical Engineering
Linköping University
581 83 Linköping, Sweden
umut@isy.liu.se

Thomas B. Schön

Division of Automatic Control
Department of Electrical Engineering
Linköping University
581 83 Linköping, Sweden
schon@isy.liu.se

Fredrik Gustafsson

Division of Automatic Control
Department of Electrical Engineering
Linköping University
581 83 Linköping, Sweden
fredrik@isy.liu.se

Abstract—In this paper we consider the problem of tracking targets, which can move both on-road and off-road, with particle filters utilizing the road-network information. It is argued that the constraints like speed-limits and/or one-way roads generally incorporated into on-road motion models make it necessary to consider additional high-bandwidth off-road motion models. This is true even if the targets under consideration are only allowed to move on-road due to the possibility of imperfect road-map information and drivers violating the traffic rules. The particle filters currently used struggles during sharp mode transitions, with poor estimation quality as a result. This is due to the fact the number of particles allocated to each motion mode is varying according to the mode probabilities. A recently proposed interacting multiple model (IMM) particle filtering algorithm, which keeps the number of particles in each mode constant irrespective of the mode probabilities, is applied to this problem and its performance is compared to a previously existing algorithm. The results of the simulations on a challenging bearing-only tracking scenario show that the proposed algorithm, unlike the previously existing algorithm, can achieve good performance even under the sharpest mode transitions.

TABLE OF CONTENTS

1 INTRODUCTION	1
2 PROBLEM DEFINITION	2
3 EXISTING PARTICLE FILTERS	3
4 IMM-PF APPROACH	4
5 SIMULATION RESULTS	5
6 CONCLUSIONS	9
ACKNOWLEDGEMENTS	9
REFERENCES	10
BIOGRAPHY	11

1. INTRODUCTION

Target tracking with road network information requires methodologies which can keep the inherent multi-modality of the underlying probability densities. The classical framework for handling such problems is the so-called jump-Markov

(non)linear systems. Early studies dealing with the problem used the interacting multiple model (IMM) algorithm [1, 2] with extended Kalman filters (EKFs) as sub-blocks [3–5]. Since the different road segments corresponded to different modes in these IMM algorithms, there were too many of them to be considered at a single step of the multiple model filter. Hence, these algorithms applied the so-called variable structure interacting multiple model (VS-IMM) algorithm [6] which adds/removes modes into/from the filter when necessary. The interest on the problem has steadily increased which lead to a considerable amount of research, see for example [7–17], a considerable amount of which involves particle filters.

Particle filters provide stochastic sampling based approximate solutions for Bayesian filtering (See the book [18] or tutorials [19–21] and the references therein). Multiple model filtering using particle filters is an already explored area in the literature. In fact, particle filters have been applied to filtering of jump Markov (non)linear systems in several papers [22–28] in a short time after their introduction [29]. They are one of the most suitable alternatives in target tracking using road-network information due to their capabilities in keeping multi-modularity and non-Gaussianity of the involved densities and in incorporating constraints. To the best of the authors' knowledge, the first particle filter used in the context of target tracking with road network information was given in [30, 31] which concluded that it significantly improves the performance of the classical VS-IMM algorithms. Other, more recent examples of the particle filters in this respect are given in [32–37].

When the tracking algorithms are required to track targets which can move both on the road network whose information is supplied by the user, and off-road, the particle filters use multiple models to cover these different types of motions. Notice here that on-road target motion might have constraints like upper or lower speed limits on different road segments and/or constraints like one-way roads. The on-road target motion model used in the algorithm is then selected to include as many of these constraints as possible. However, most of the times when the algorithm runs on real data, there could be map-errors which would result in wrong on-road constraints or drivers that would violate the rules coded in the map infor-

mation. At those times, the availability of a high-bandwidth off-road target motion model would make the algorithm capable of taking care of these unpredicted target motions. A typical example of such a situation would be the case in which the used map information misses some (possibly short) road segments the target is traveling. When the target swerves into such road segments, the on-road model which is significantly constrained cannot track the target. On the other hand, if a high-bandwidth model which does not have the road constraints can enable the algorithm to keep its track until the target gets into another road segment whose information is provided in the road database. Therefore, use of multiple models might be necessary to track unexpected motions (i.e., motions that do not suit to assumed on-road target motion model) even if we plan to track only on-road targets.

Multiple model particle filters employed for target tracking with road networks [31–33] (except for some recent unconventional ones like [35, 36]) use

1. Bootstrap multiple model particle filter (BS-MMPF) [22] which uses state transition density to generate predicted particles [31].
2. Sampling Importance Resampling (SIR) multiple model particle filter (SIR-MMPF) which uses the optimal proposal density to generate predicted particles [33, 34].

Both of the algorithms above use variable number of particles for each model which changes according to the current posterior probability of the modes. As an example, suppose one of the algorithms uses N_p particles in total for two modes $r = 1, 2$ where the modes $r = 1$ and $r = 2$ correspond to on-road particles and off-road particles, respectively. Then, at an arbitrary time k in the process of estimation, the number of on road particles in the algorithm will be approximately $N_p^1 = N_p P(r_k = 1 | y_{0:k})$ where $P(r_k = 1 | y_{0:k})$ is the posterior probability that the target is on the road at time k . The remaining particles of the algorithm are then assigned to the off-road target motion model. This varying number of particles in the operation of the algorithm has the advantage that most of the computational resources are allocated automatically to the most probable model the algorithm uses. On the other hand, it has quite important drawbacks. When the target remains on-road or off-road for a long time, the corresponding posterior mode probability gets very close to unity. Therefore almost all particles in the algorithm concentrate in the dominant mode. If then the target makes a swift mode switch, the particles representing the new mode might be initially too few to obtain a satisfactory estimation performance. This problem is especially crucial when the overall number of particles used in the algorithm is constrained by the computational resources available. In many cases, degraded estimation performance might be unavoidable which can even lead to track loss. In applications like fault detection, the exact time of fault onset might be blurred by this low performance estimation.

A second disadvantage of using the above filters in road net-

work estimation is the waste of computation resources. In estimation with road network information available, the on-road model requires much less particles than required by the off-road model. This is evident in most applications where a lower order model is used for on-road motion. The constrained motion characteristics of the on-road motion makes large number of particles used in general unnecessary. The total number of particles used in the filters mentioned above is selected to accommodate good estimation performance in the off-road estimation. However, these algorithms would use almost all the particles available for the on-road model when the target posterior probability of the on-road mode is close to unity.

In order to overcome the difficulties described above, in this paper, we apply the interacting multiple model particle filter (IMM-PF) proposed by Driessen and Boers [26] to this estimation problem. This filter uses interacting multiple model type [1, 2] approximations to handle multiple models and it has the flexibility to choose the number of particles allocated to each mode independent of the posterior mode probabilities.

2. PROBLEM DEFINITION

Suppose we would like to track a target which can move both on the road network whose information is given and off-road. We will use \mathcal{I}_{RN} to denote the road network information. Then we consider two different state space representations corresponding to on-road and off-road target modes,

$$x_{k+1}^r = f^r(x_k^r, \mathcal{I}_{RN}, \eta_{k+1}^r, \nu_{k+1}^r), \quad (1)$$

$$x_{k+1}^g = f^g(x_k^g, \eta_{k+1}^g), \quad (2)$$

where the vectors $x_k^r \in \mathbb{R}^{n_x^r}$ and $x_k^g \in \mathbb{R}^{n_x^g}$ represent the state vector of the target in on-road and off-road (global) coordinates, respectively. The functions $f^r(\cdot)$ and $f^g(\cdot)$ are in general nonlinear functions. The continuous process noise terms $\eta_k^r \in \mathbb{R}^{n_x^r}$ and $\eta_k^g \in \mathbb{R}^{n_x^g}$ are assumed to be white. The discrete process noise $\nu_{k+1}^r \in \{1, 2, \dots, N_r(x_k^r)\}$ determines which road segment the target will follow in the next sampling interval in case more than one alternative exists. We assume the availability of prior probability density functions (or probability mass functions in the discrete case) $p_{\eta_k^r}(\cdot)$, $p_{\eta_k^g}(\cdot)$ and $p_{\nu_k^r}(\cdot)$ for the random variables η_k^r , η_k^g and ν_k^r , respectively. In order to be able to use both models at the same time, one always needs the appropriate functions to convert the state vectors given in one of the representations into the other representation. For this purpose we assume the availability of two transformation functions named $T^{gr}(\cdot)$ (transformation from road coordinates to global coordinates) and $T^{rg}(\cdot)$ (transformation from global coordinates to road coordinates). The function $T^{gr}(\cdot)$ converts a state vector given in on-road coordinates to off-road (global) coordinates. This is generally an easy task, assuming that the road network information is available. The function $T^{rg}(\cdot)$, on the other hand, has to find the closest on-road coordinate state corresponding to a state vector in global coordinates. This is more involved in that one generally has to search in a large road database

for the closest point on the road network to the position component of the global state vector and has to project the velocity and other quantities onto their equivalents in the road network.

The measurements associated with the target are modelled according to relations

$$y_k = h^g(x_k^g) + v_k^g, \quad (3)$$

where $h^g(\cdot)$ is in general a nonlinear function of the global state of the target and v_k^g is white measurement noise. Note that with this notation, the measurements related to on-road coordinates of the target can be written to satisfy

$$y_k = h^g(T^{gr}(x_k^r, \mathcal{I}_{RN})) + v_k^g. \quad (4)$$

In the following, we are going to denote the hypothesis (event) that the target is moving on-road or off-road by a discrete variable $r_k \in \{1, 2\}$ where events $\{r_k = 1\}$ and $\{r_k = 2\}$ correspond to the hypotheses that the target is on-road (represented by (1)) and off-road (represented by (2)), respectively. According to the value of the variable r_k the corresponding dynamics of the target given in (1) and (2) must be used. It is assumed that r_k is a homogeneous¹ Markov chain with transition probability matrix denoted as $\Pi = [\pi_{ij}]$ where

$$\pi_{ij} \triangleq P(r_k = j | r_{k-1} = i). \quad (5)$$

3. EXISTING PARTICLE FILTERS

In estimation theory, it is well known that all of the information about a random variable can be propagated by calculating its posterior density given all the measurements. Particle filters do this by keeping a number of particles and their corresponding weights for the posterior density. In estimation applications as described above, one keeps the particles $\{x_k^{(i)}, r_k^{(i)}\}_{i=1}^{N_p}$ and their weights $\{w_k^{(i)}\}_{i=1}^{N_p}$ where $x_k^{(i)}$ is the state of the particle with respect to either road coordinates $(x_k^{r,(i)})$ or global coordinates $(x_k^{g,(i)})$ according to the value of the on-road/off-road hypothesis variable $r_k^{(i)}$ i.e.,

$$x_k^{(i)} = \begin{cases} x_k^{r,(i)}, & r_k^{(i)} = 1 \\ x_k^{g,(i)}, & r_k^{(i)} = 2 \end{cases} \quad (6)$$

Having these particles one can always calculate the density of the state of the target in global coordinates as

$$p(x_k | y_{0:k}) = \sum_{i=1}^N w_k^{(i)} \delta_{T^g(x_k^{(i)}, r_k^{(i)})}(z_k) \quad (7)$$

where $T^g(\cdot, \cdot)$ is the globalization function defined as

$$T^g(x_k^{(i)}, r_k^{(i)}) \triangleq \begin{cases} T^{gr}(x_k^{(i)}, r_k^{(i)}) & r_k^{(i)} = 1 \\ x_k^{(i)} & r_k^{(i)} = 2 \end{cases} \quad (8)$$

¹Although the Markov chain is assumed to be time-invariant, the corresponding transition probabilities might always be selected in a time-varying way based on state or environment conditions in the algorithms presented in this document.

Using the density function (7), the minimum mean square error estimate of the target state in global coordinates is given by

$$\hat{x}_{k|k} = \sum_{i=1}^{N_p} w_k^{(i)} T^g(x_k^{(i)}, r_k^{(i)}) \quad (9)$$

with a covariance

$$P_{k|k} = \sum_{i=1}^{N_p} w_k^{(i)} \left(T^g(x_k^{(i)}, r_k^{(i)}) - \hat{z}_k \right) \times \left(T^g(x_k^{(i)}, r_k^{(i)}) - \hat{z}_k \right)^T \quad (10)$$

A particle filter then needs to calculate with each measurement the updated particles $\{x_k^{(i)}, r_k^{(i)}\}_{i=1}^{N_p}$ and their weights $\{w_k^{(i)}\}_{i=1}^{N_p}$ from the corresponding previous particles $\{x_{k-1}^{(i)}, r_{k-1}^{(i)}\}_{i=1}^{N_p}$ and weights $\{w_{k-1}^{(i)}\}_{i=1}^{N_p}$.

The most basic form of a particle filter is the so-called bootstrap filter [29]. Its application to estimation with road networks was presented in [22]. We give a brief description of its single step below.

Algorithm 1 (BS-MMPF) Suppose we have the previous particles $\{x_{k-1}^{(i)}, r_{k-1}^{(i)}\}_{i=1}^{N_p}$ and weights $\{w_{k-1}^{(i)}\}_{i=1}^{N_p}$ available and we have received a new measurement y_k .

1. Resampling: Sample $\{\tilde{x}_{k-1}^{(i)}, \tilde{r}_{k-1}^{(i)}\}_{i=1}^{N_p}$ from $\{x_{k-1}^{(i)}, r_{k-1}^{(i)}\}_{i=1}^{N_p}$ according to weights $\{w_{k-1}^{(i)}\}_{i=1}^{N_p}$ such that

$$P(\tilde{x}_{k-1}^{(i)} = x_{k-1}^{(j)}, \tilde{r}_{k-1}^{(i)} = r_{k-1}^{(j)}) = w_{k-1}^{(j)} \quad (11)$$

for each $i = 1, \dots, N_p$.

2. Prediction Step:

(a) Sample $r_k^{(i)}$ from $\tilde{r}_{k-1}^{(i)}$ such that

$$P(r_k^{(i)} | \tilde{r}_{k-1}^{(i)}) = \pi_{\tilde{r}_{k-1}^{(i)} r_k^{(i)}} \quad (12)$$

for each $i = 1, \dots, N_p$.

(b) Generate $x_k^{(i)}$ from $\tilde{x}_{k-1}^{(i)}, r_k^{(i)}$ and $\tilde{r}_{k-1}^{(i)}$ by using samples from the process noise sequences $\eta_k^{r,(i)} \sim p_{\eta_k^r}(\cdot)$, $\eta_k^{g,(i)} \sim p_{\eta_k^g}(\cdot)$ and $\nu_k^{r,(i)} \sim p_{\nu_k^r}(\cdot)$ as shown in Equation (13) at the top of next page. for each $i = 1, \dots, N_p$.

3. Update Step: Set $w_k^{(i)}$ as

$$w_k^{(i)} \propto p_{vg} \left(y_k - h \left(T^g(x_k^{(i)}, r_k^{(i)}) \right) \right) \quad (14)$$

such that $\sum_{i=1}^{N_p} w_k^{(i)} = 1$.

As seen in the algorithm, the user cannot directly control the number of particles which have $r_k^{(i)} = 1$. The same is true

$$x_k^{(i)} = \begin{cases} f^r \left(\tilde{x}_{k-1}^{(i)}, \mathcal{I}_{RN}, \eta_k^{r,(i)}, \nu_k^{r,(i)} \right), & \tilde{r}_{k-1}^{(i)} = 1, r_k^{(i)} = 1 \\ f^g \left(T^{gr} \left(\tilde{x}_{k-1}^{(i)}, \mathcal{I}_{RN} \right), \eta_k^{g,(i)} \right), & \tilde{r}_{k-1}^{(i)} = 1, r_k^{(i)} = 2 \\ f^r \left(T^{rg} \left(\tilde{x}_{k-1}^{(i)}, \mathcal{I}_{RN} \right), \mathcal{I}_{RN}, \eta_k^{r,(i)}, \nu_k^{r,(i)} \right), & \tilde{r}_{k-1}^{(i)} = 2, r_k^{(i)} = 1 \\ f^g \left(\tilde{x}_{k-1}^{(i)}, \eta_k^{g,(i)} \right), & \tilde{r}_{k-1}^{(i)} = 2, r_k^{(i)} = 2 \end{cases} \quad (13)$$

for the particles which have $r_k^{(i)} = 2$. The values of the mode state $r_k^{(i)}$ are sampled from the previous mode state $\tilde{r}_{k-1}^{(i)}$ using transition probabilities and they are resampled using the likelihoods. One can make an approximate guess that the number particles for each mode would be approximately

$$N_p^j \triangleq \sum_{i=1}^{N_p} \mathcal{I}\{r_k^{(i)} = j\} \approx N_p P(r_k = j | y_{0:k-1}) \quad (15)$$

where $\mathcal{I}\{\mathcal{A}\}$ is the indicator function of the event \mathcal{A} .

Notice that the BS-MMPF algorithm samples the particles $\{x_k^{(i)}, r_k^{(i)}\}_{i=1}^{N_p}$ from the previous particles $\{\tilde{x}_{k-1}^{(i)}, \tilde{r}_{k-1}^{(i)}\}_{i=1}^{N_p}$ independent of the measurement y_k . This is a manifestation of the fact that the Bootstrap filter samples $\{x_k^{(i)}, r_k^{(i)}\}$ according to

$$\{x_k^{(i)}, r_k^{(i)}\} \sim p(x_k, r_k | \tilde{x}_{k-1}^{(i)}, \tilde{r}_{k-1}^{(i)}) \quad (16)$$

One can make this sampling scheme better guided by using the following sampling method

$$\{x_k^{(i)}, r_k^{(i)}\} \sim p(x_k, r_k | \tilde{x}_{k-1}^{(i)}, \tilde{r}_{k-1}^{(i)}, y_k). \quad (17)$$

This is the method selected in [33] to apply the SIR-MMPF filter to the problem. In this case, one has to obtain the samples from the sequences as

$$r_k^{(i)}, \eta_k^{r,(i)}, \eta_k^{g,(i)}, \nu_k^{r,(i)} \sim p_{r_k, \eta_k^r, \eta_k^g, \nu_k^r}(\cdot | y_k, \tilde{x}_{k-1}^{(i)}, \tilde{r}_{k-1}^{(i)}). \quad (18)$$

Calculation of this density function makes prediction and weight calculation steps more complicated than the bootstrap filter, but since the inclusion of the measurements in sampling enhances the quality of the selected particles, the number of particles can typically be significantly reduced. This type of algorithm can be useful, especially in high signal to noise ratio applications. It can, on the other hand, result in low performance with low quality measurements because the density function used to select the new particles might force the particles into wrong regions in the road database due to high measurement noise. The algorithm might also have a disadvantage in the case where the measurements need to be associated to targets in multi-target applications. In that case, since the predictions are required before the measurements are processed, the algorithm might require multiple prediction steps.

4. IMM-PF APPROACH

In this section, we are going to apply the interacting multiple model particle filter (IMM-PF) proposed by Driessen and Boers in [26] to our problem. In this approach, one fixes the number of the particles in each mode to predetermined values². In this way, even if the posterior probability of one or more of the modes gets too small, the number of particles in that mode does not shrink. Compared to the algorithms given in the previous section, IMM-PF also keeps and updates the posterior mode probabilities

$$\mu_k^j \triangleq P(r_k = j | y_{0:k}), \quad (19)$$

for $j = 1, 2$ along with particles and their weights. Below, we give a brief description of a single step of the algorithm.

Algorithm 2 (IMM-PF) Suppose we have the previous on-road particles $\{x_{k-1}^{r,(i)}\}_{i=1}^{N_p^r}$ and their weights $\{w_{k-1}^{r,(i)}\}_{i=1}^{N_p^r}$, off-road particles $\{x_{k-1}^{g,(i)}\}_{i=1}^{N_p^g}$ and their weights $\{w_{k-1}^{g,(i)}\}_{i=1}^{N_p^g}$ and the mode probabilities $\{\mu_{k-1}^j\}_{j=1}^2$ available and we have received a new measurement y_k .

1. Resampling Step:

(a) Sample $\{\tilde{x}_{k-1}^{r,(i)}\}_{i=1}^{N_p^r}$ from $\{x_{k-1}^{r,(i)}\}_{i=1}^{N_p^r}$ according to weights $\{w_{k-1}^{r,(i)}\}_{i=1}^{N_p^r}$ such that

$$P(\tilde{x}_{k-1}^{r,(i)} = x_{k-1}^{r,(j)}) = w_{k-1}^{r,(j)} \quad (20)$$

for $i = 1, \dots, N_p^r$.

(b) Sample $\{\tilde{x}_{k-1}^{g,(i)}\}_{i=1}^{N_p^g}$ from $\{x_{k-1}^{g,(i)}\}_{i=1}^{N_p^g}$ according to weights $\{w_{k-1}^{g,(i)}\}_{i=1}^{N_p^g}$ such that

$$P(\tilde{x}_{k-1}^{g,(i)} = x_{k-1}^{g,(j)}) = w_{k-1}^{g,(j)} \quad (21)$$

for $i = 1, \dots, N_p^g$.

2. Mixing Step:

(a) Calculate mixing probabilities $\mu_{k-1}^{\ell j} \triangleq P(r_{k-1} = \ell | r_k = j, y_{0:k-1})$ as

$$\mu_{k-1}^{\ell j} = \frac{\pi_{\ell j} \mu_{k-1}^{\ell}}{\sum_{s=1}^2 \pi_{s j} \mu_{k-1}^s} \quad (22)$$

²Actually, in its most general form, one can select the number of particles in each mode arbitrarily.

for $1 \leq \ell, j \leq 2$.

(b) Sample $\{r_{k-1}^{r,(i)}\}_{i=1}^{N_p^r}$ and $\{r_{k-1}^{g,(i)}\}_{i=1}^{N_p^g}$ such that

$$P(r_{k-1}^{r,(i)} = \ell) = \mu_{k-1}^{\ell 1} \quad (23)$$

$$P(r_{k-1}^{g,(i)} = \ell) = \mu_{k-1}^{\ell 2} \quad (24)$$

for $\ell = 1, 2$.

(c) Sample $\{\bar{x}_{k-1}^{r,(i)}\}_{i=1}^{N_p^r}$ and $\{\bar{x}_{k-1}^{g,(i)}\}_{i=1}^{N_p^g}$ such that

$$\bar{x}_{k-1}^{r,(i)} \sim \begin{cases} \frac{1}{N_p^r} \sum_{j=1}^{N_p^r} \delta_{\bar{x}_{k-1}^{r,(j)}}(x_{k-1}), & r_{k-1}^{r,(i)}=1 \\ \frac{1}{N_p^g} \sum_{j=1}^{N_p^g} \delta_{T^{rg}(\bar{x}_{k-1}^{g,(j)}, \mathcal{I}_{RN})}(x_{k-1}), & r_{k-1}^{r,(i)}=2 \end{cases} \quad (25)$$

$$\bar{x}_{k-1}^{g,(i)} \sim \begin{cases} \frac{1}{N_p^r} \sum_{j=1}^{N_p^r} \delta_{T^{gr}(\bar{x}_{k-1}^{r,(j)}, \mathcal{I}_{RN})}(x_{k-1}), & r_{k-1}^{g,(i)}=1 \\ \frac{1}{N_p^g} \sum_{j=1}^{N_p^g} \delta_{\bar{x}_{k-1}^{g,(j)}}(x_{k-1}), & r_{k-1}^{g,(i)}=2 \end{cases} \quad (26)$$

3. Prediction Step:

(a) Generate $\{x_k^{r,(i)}\}_{i=1}^{N_p^r}$ and $\{x_k^{g,(i)}\}_{i=1}^{N_p^g}$ from $\{\bar{x}_{k-1}^{r,(i)}\}_{i=1}^{N_p^r}$ and $\{\bar{x}_{k-1}^{g,(i)}\}_{i=1}^{N_p^g}$ using samples from the process noise sequences $\eta_k^{r,(i)} \sim p_{\eta_k^r}(\cdot)$, $\eta_k^{g,(i)} \sim p_{\eta_k^g}(\cdot)$ and $\nu_k^{r,(i)} \sim p_{\nu_k^r}(\cdot)$.

$$x_k^{r,(i)} = f^r(\bar{x}_{k-1}^{r,(i)}, \mathcal{I}_{RN}, \eta_k^{r,(i)}, \nu_k^{r,(i)}) \quad (27)$$

$$x_k^{g,(i)} = f^g(\bar{x}_{k-1}^{g,(i)}, \eta_k^{g,(i)}) \quad (28)$$

4. Update Step:

(a) Set $w_k^{r,(i)}$ and $w_k^{g,(i)}$ as

$$w_k^{r,(i)} \propto p_{vg} \left(y_k - h^g \left(T^{gr} \left(x_k^{r,(i)}, \mathcal{I}_{RN} \right) \right) \right) \quad (29)$$

$$w_k^{g,(i)} \propto p_{vg} \left(y_k - h^g \left(x_k^{g,(i)} \right) \right) \quad (30)$$

such that $\sum_{i=1}^{N_p^r} w_k^{r,(i)} = 1$ and $\sum_{i=1}^{N_p^g} w_k^{g,(i)} = 1$.

(b) Set μ_k^j as

$$\mu_k^1 \propto \left[\sum_{s=1}^{N_p^r} p_{vg} \left(y_k - h \left(T^{gr} \left(x_k^{g,(s)}, \mathcal{I}_{RN} \right) \right) \right) \right] \times \left[\sum_{s=1}^2 \pi_{s1} \mu_{k-1}^s \right] \quad (31)$$

$$\mu_k^2 \propto \left[\sum_{s=1}^{N_p^g} p_{vg} \left(y_k - h \left(x_k^{g,(s)} \right) \right) \right] \times \left[\sum_{s=1}^2 \pi_{s2} \mu_{k-1}^s \right] \quad (32)$$

such that $\mu_k^1 + \mu_k^2 = 1$.

Remark 1: The use of optimal proposal densities is much easier for the IMM-PF algorithm. For an efficient distribution

of particles among the modes, BS-MMPF filter must make the selection of the mode samples $r_k^{(i)}$ according to the optimal density given in (18). On the other hand, by its construction, IMM-PF filter has already enough number of particles in each of its modes and therefore, it is not necessary for it to consider mode sequences in sampling from the optimal importance density. The IMM-PF can be modified to employ optimal proposal density by selecting the process noise terms used in (27) and (28) as

$$\eta_k^{r,(i)}, \nu_k^{r,(i)} \sim p_{\eta_k^r, \nu_k^r}(\cdot | y_k, \bar{x}_{k-1}^{r,(i)}), \quad (33)$$

$$\eta_k^{g,(i)} \sim p_{\eta_k^g}(\cdot | y_k, \bar{x}_{k-1}^{g,(i)}). \quad (34)$$

The densities in the above equations are easier to obtain and involve less combinations than the one in (18). ■

Remark 2: Since the IMM-PF algorithm's filtering units for each mode are decoupled, one can replace the filter of each mode with any other possible nonlinear filter, e.g., Gaussian sum filter, in the literature as long as transformations from empirical (particle) representation to analytical (parametric) representation of the densities and vice versa are available. In this way, one can obtain hybrid filters which can run particle filters for some of the modes and analytical approximation based filters (like EKF, UKF, Gaussian sum filter) for the others. ■

5. SIMULATION RESULTS

In this section, we compare the performance of the BS-MMPF algorithm to that of the IMM-PF filter using two examples. In the examples, we consider same map and target information given in Figures 1 and 2, respectively. The map is composed of linear road segments separated by dots in Figure 1. The true target positions are shown with crosses along with their time stamps at their upper right corners in Figure 2. Target moves with approximately constant speed on the road segments for 75 seconds and then it swerves into an off-road region and stays off-road about 20 seconds. At the 95th second it again enters the road with a sharp swerve and continues on-road until the end of the scenario. Notice that this type of sharp on-road/off-road switchings could be commonly encountered in the case that the map information is missing some road segments.

For the on-road motion, we use the state variable $x_k^r = [p_k^r, v_k^r, i_k^r]^T$ where the scalar variables p_k^r, v_k^r denote the position and speed values of the target on the road segment which is identified by the integer index i_k^r . The following model is used for the dynamics of x_k^r .

$$\begin{bmatrix} p_{k+1}^r \\ v_{k+1}^r \\ i_{k+1}^r \end{bmatrix} = f^r \left(\begin{bmatrix} p_k^r \\ v_k^r \\ i_k^r \end{bmatrix}, \mathcal{I}_{RN}, \nu_{k+1}^r \right) \quad (35)$$

where

$$\begin{bmatrix} p_{k+1}^r \\ v_{k+1}^r \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_k^r \\ v_k^r \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \eta_{k+1}^r \quad (36)$$

The continuous process noise η_k^r is a scalar white Gaussian acceleration noise with zero mean and $0.2m/sec^2$ standard deviation and $T = 5secs$. The predicted position and speed values p_{k+1}^r, v_{k+1}^r might not be on the road segment indicated by i_k^r . The function $f^r(\cdot)$ therefore projects the values p_{k+1}^r, v_{k+1}^r into the road segment denoted by i_{k+1}^r . If there are more than one candidate for the next road segment index i_{k+1}^r , the function also selects a random one according to the value of the discrete on-road process noise term $\nu_{k+1}^r \in \{1, 2, \dots, N_r(x_k^r)\}$ where $N_r(x_k^r)$ is the number of possible road segments that the target with on-road state x_k^r might go in the following T seconds.

The off-road target model is a two dimensional constant velocity model given as

$$\begin{bmatrix} p_{k+1}^{g,x} \\ p_{k+1}^{g,y} \\ v_{k+1}^{g,x} \\ v_{k+1}^{g,y} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_2 & T\mathbf{I}_2 \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} p_k^{g,x} \\ p_k^{g,y} \\ v_k^{g,x} \\ v_k^{g,y} \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2}\mathbf{I}_2 \\ T\mathbf{I}_2 \end{bmatrix} \eta_{k+1}^g \quad (37)$$

where η_k^g is a two dimensional white Gaussian noise with zero mean and covariance $5^2\mathbf{I}_2$ and \mathbf{I}_n is the identity matrix of dimension n .

Example 1: Stationary Sensor with Range and Bearing Measurements

In this first example, we assume that the range and bearing measurements given as

$$y_k = \begin{bmatrix} r_k \\ \phi_k \end{bmatrix} = \begin{pmatrix} \sqrt{(p_k^{g,x} - p_S^x)^2 + (p_k^{g,y} - p_S^y)^2} \\ \arctan 2 \frac{p_k^{g,y} - p_S^y}{p_k^{g,x} - p_S^x} \end{pmatrix} + v_k \quad (38)$$

are collected by a stationary sensor at $[p_S^x, p_S^y] = [-200m, 0m]$ illustrated in Figure 1. Here, v_k is a white Gaussian measurement noise with zero mean and covariance R given as

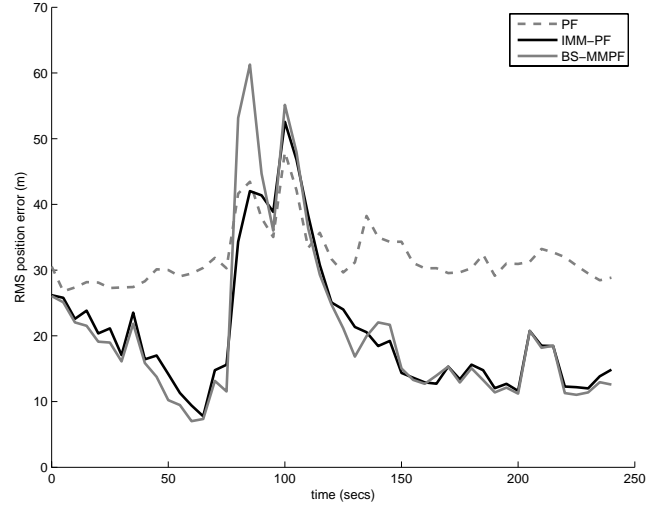
$$R = \begin{bmatrix} 25^2 & 0 \\ 0 & 0.01^2 \end{bmatrix}. \quad (39)$$

The transition probability matrix of the underlying Markov chain is selected to be

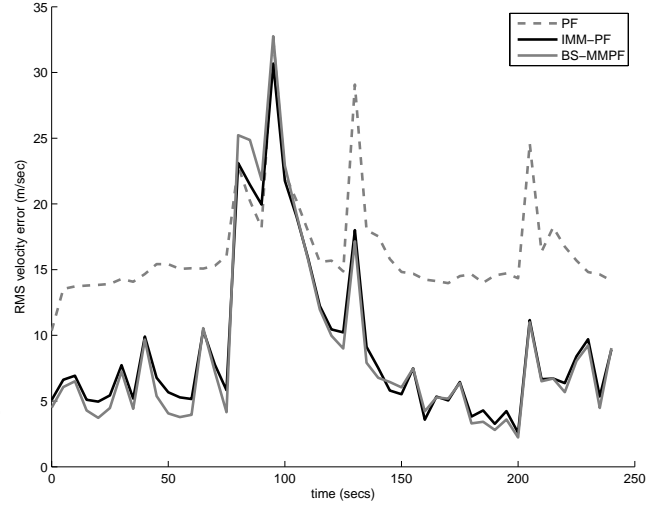
$$\Pi_1 = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}. \quad (40)$$

With the given model information, we make 1000 Monte-Carlo runs with a standard particle filter (PF) which does not use any map information³ and BS-MMPF, IMM-PF algorithms which use the available map. PF and BS-MMPF use $N_p = 200$ particles each. IMM-PF uses $N_p^r = 75$ and $N_p^g = 125$ particles. Note that when both range and bearing measurements are given, the number of particles used in the algorithms can be reduced to until even $N_p = 75$ with negligible or small performance degradation as also observed in the literature [33].

³This particle filter uses only the off-road model to track the target.



(a) Position errors



(b) Velocity errors

Figure 3. Example 1: RMS position and velocity errors of the particle filters.

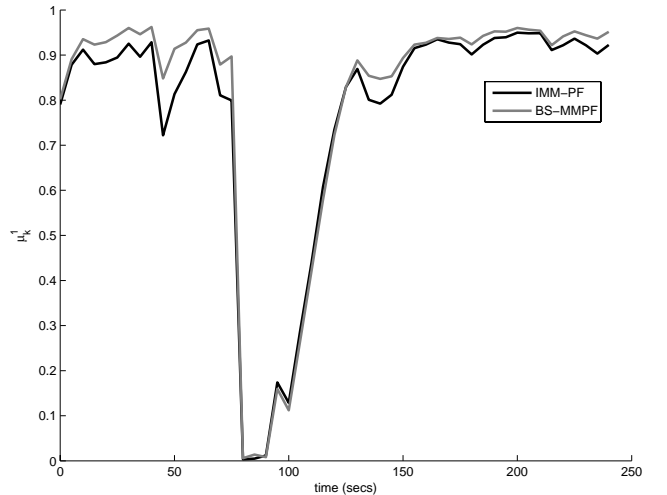


Figure 4. Example 1: Average posterior mode probabilities μ_k^1 of the multiple model particle filters.

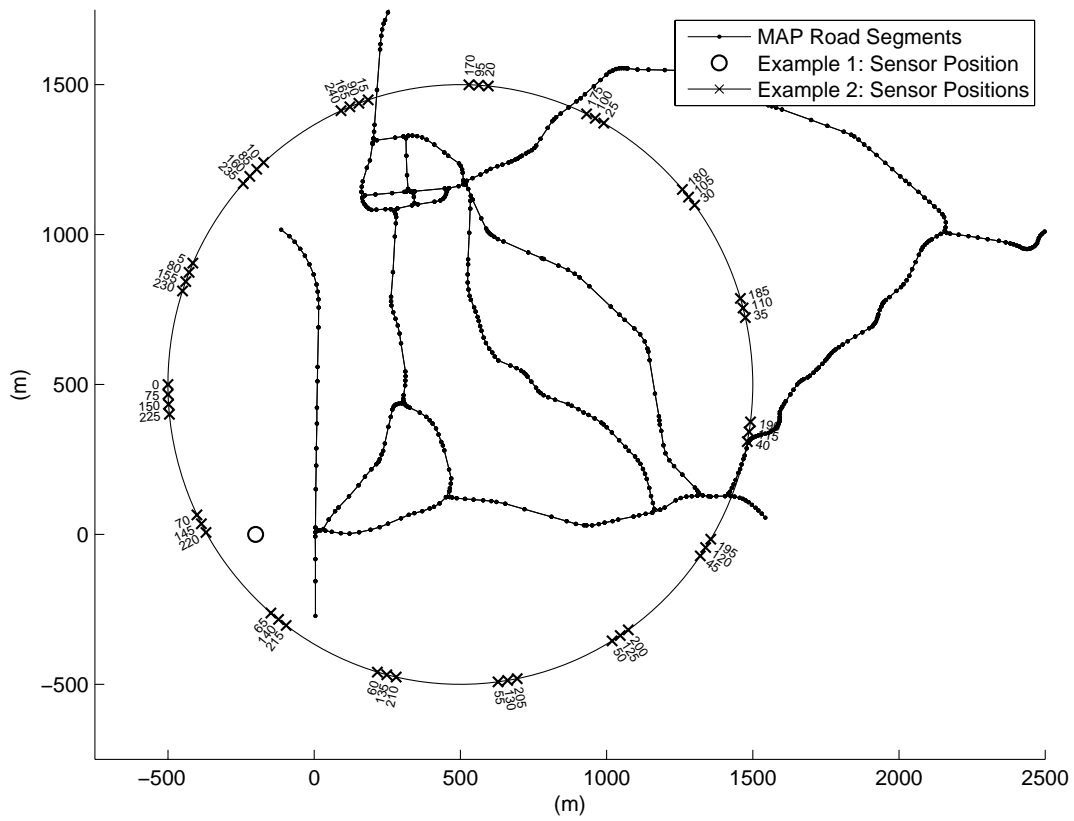


Figure 1. Map and sensor positions used in the examples.

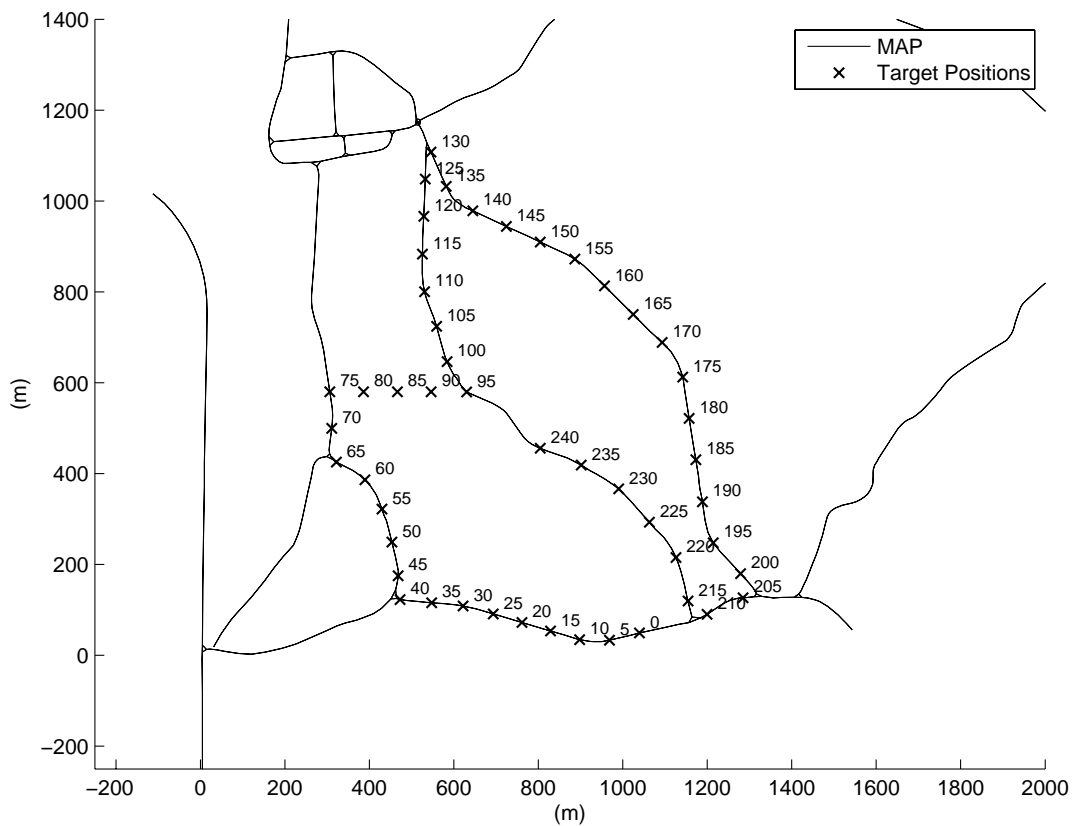


Figure 2. Map and true target positions used in the examples.

Figures 3(a) and 3(b) show the RMS position and velocity errors of the particle filters respectively and Figure 4 illustrates the average mode probabilities obtained by the multiple model particle filters.

As seen in the figures the performances of IMM-PF algorithm obtains a performance much below that of standard PF at most of the times and slightly worse performance around the switchings. The BS-MMPF filter can get slightly better performance than IMM-PF until and after the mode-switches. This is the effect of the IMM-type approximations made in the IMM-PF. On the other hand, between mode switchings, BS-MMPF estimates can excessively deteriorate even to values much worse than the standard filter due to the particle deficiency phenomenon mentioned in the previous parts of the document.

Example 2: Moving Sensor with Bearing Only Measurements

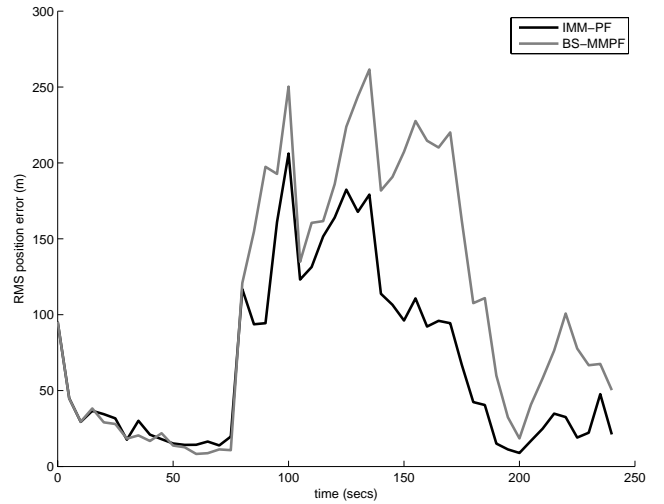
In this second example, we assume that an UAV flying with constant speed 300km/h around the region collects bearing only measurements of the target given as

$$y_k = \phi_k = \arctan 2 \frac{p_k^{g,y} - p_S^y(k)}{p_k^{g,x} - p_S^x(k)} + v_k \quad (41)$$

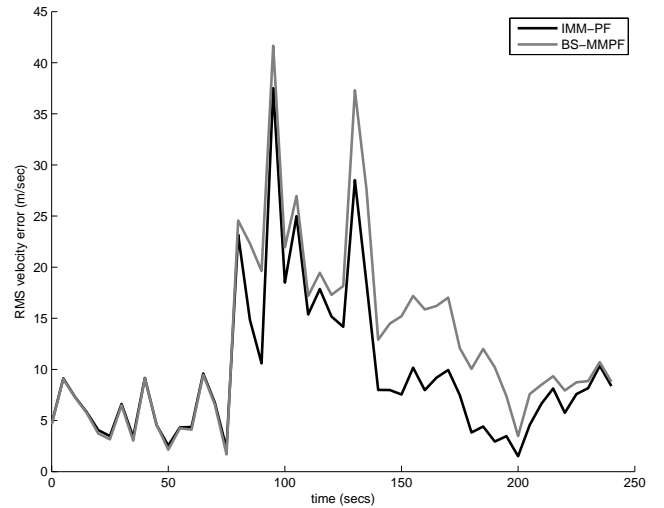
v_k is a white Gaussian measurement noise with zero mean and standard deviation 0.01rads . The UAV trajectory is assumed to be perfectly circular in clockwise direction with center $[500\text{m}, 500\text{m}]$ and radius 1km which is illustrated in Figure 1. The initial position of the UAV is $[-500\text{m}, 500\text{m}]$.

This example poses real challenges for the algorithms. Compared to the previous example, the particles used in the filters spread out to a large region because the single sensor observations are not enough to localize the target. Therefore, the particle filters had to use large number of particles compared to Example 1. The standard particle filter used 2000 particles and it has been seen that it totally diverges in most of the Monte Carlo runs and therefore its results has been excluded from the figures. BS-MMPF and IMM-PF algorithms use $N_p^r = 2000$ and $N_p^g = 500$, $N_p^s = 1500$ particles respectively.

Since both on-road and off-road particles in the filters can spread to a large region, different modes of the algorithm can degrade the performance of the overall filter when posterior mode probabilities can assume values similar to 0.9–0.1 in their steady state as observed in Figure 4. A simple example of this phenomenon was observed especially when the target is moving off-road. The on-road particles in the filters then could have quite wrong position values which can survive the resampling processes due to the localization incapability of single measurements. These very wrong position values, then, can affect the overall estimates of the algorithm if the mode probabilities can take values other than (close to) zero or unity. One method to force the mode probabilities towards those values is to use a highly diagonally dominant transition probability matrix. The one selected for this exam-



(a) Position errors



(b) Velocity errors

Figure 5. Example 2: RMS position and velocity errors of the particle filters.

ple is given as

$$\Pi_2 = \begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix} \quad (42)$$

A total of 1000 Monte-Carlo runs has been made with the algorithms. Divergence has been observed on some of the runs for both filters. For this reason, the Monte Carlo runs in which the filtered position estimates differ from the true position of the target (at any estimation step) more than 1km are discarded as divergent runs. Out of 1000 Monte Carlo runs the number of divergent runs are 64 and 6 for BS-MMPF and IMM-PF algorithms respectively. The RMS position and velocity estimation results for the remaining runs are shown in Figures 5(a) and 5(b) respectively. Figure 6 shows the posterior average mode probabilities μ_k^1 obtained in the multiple model particle filter which are closer to values 0 and 1 in the steady-state as opposed to the ones shown in Figure 4.

Table 1. Summary of the results obtained in Example 1 and Example 2.

		PF	IMM-PF	BS-MMPF
Ex-1	Total # of Particles	200	200	200
	Avg. Pos. RMSE (m)	31.95	20.67	20.66
	Avg. Vel. RMSE (m/sec)	16.41	9.06	8.82
Ex-2	Total # of Particles	2000	2000	2000
	Avg. Pos. RMSE (m)	388.77	68.14	105.67
	Avg. Vel. RMSE (m/sec)	35.24	9.62	12.52
	Total # of MC Runs	1000	1000	1000
	# of Divergent Runs	928	6	64

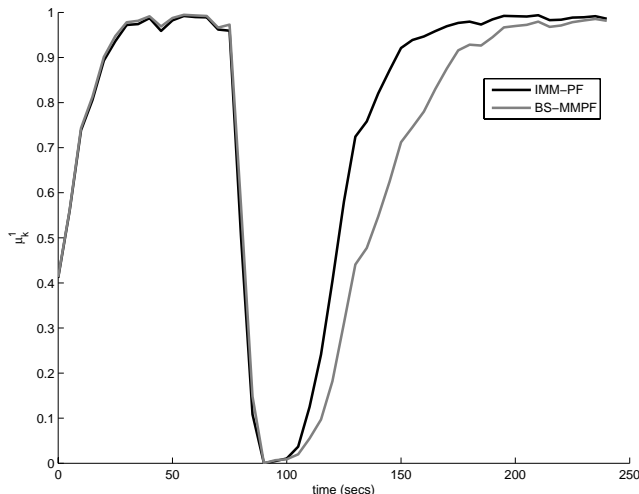


Figure 6. Example 2: Average posterior mode probabilities μ_k^1 of the multiple model particle filters.

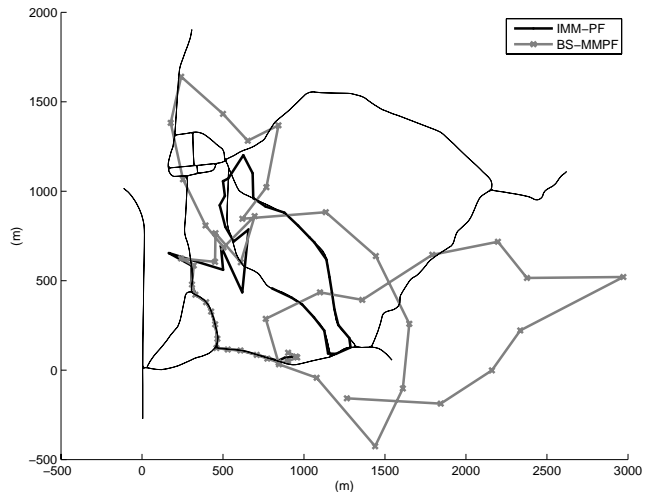


Figure 7. Example 2: One of the runs where BS-MMPF filter diverges and IMM-PF filter does not lose the track.

The performances of the algorithms are quite similar until the first switching instant. When the first switching occurs the BS-MMPF filter loses 100m's more accuracy than IMM-PF. After the switchings both filters have difficulty reducing the errors, however, it takes about 50 seconds longer for BS-MMPF to reduce the errors to the level that is achieved before the switchings. Slower transient characteristics of the BS-MMPF posterior mode probabilities after the switching also confirms the observation that the filter has difficulty adjusting mode probabilities due to particle depletion in the modes. These and the larger number of divergent cases of BS-MMPF show its susceptibility to especially sharp switchings which can, in turn, even lead to divergence. An example of this divergent behavior is illustrated in Figure 7 where estimated target trajectories for the particle filters in one of the discarded divergent runs of the BS-MMPF filter are shown. During the second transition (from off-road to on-road) BS-MMPF algorithm cannot catch the target due to particle deficiency in on-road mode. For the same scenario, the IMM-PF filter does not lose track and continue successfully until the end of the data. Finally, the results observed in Examples 1 and 2 are summarized in Table 1 where the average RMS position and velocity errors (along time) are also given.

6. CONCLUSIONS

The current particle filters applied to the problem of target tracking with road network information uses varying number of particles for each mode in the filter which changes according to the posterior mode probabilities. The drawbacks of such approaches are discussed and an available alternative in the literature, called IMM-PF, is instead proposed to avoid such drawbacks. The algorithm keeps a constant number of particles for all modes in the filter irrespective of the posterior mode probabilities. The algorithm details are given along with simulation results comparing it with the standard BS-MMPF algorithm. Results suggest the preference of IMM-PF over BS-MMPF in especially sharp and often switching scenarios.

ACKNOWLEDGMENTS

The authors gratefully acknowledge fundings from SSF (Swedish Foundation for Strategic Research) Strategic Research Center MOVIII and the Vinnova/FMV TAIS project ARCUS.

REFERENCES

- [1] H. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markov switching coefficients," *IEEE Trans. Automat. Contr.*, vol. 33, no. 8, pp. 780–783, Aug. 1988.
- [2] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Norwell, MA: Artech House, 1993.
- [3] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, and I. Kadar, "Ground target tracking with variable structure IMM estimator," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 1, pp. 26–46, Jan. 2000.
- [4] P. J. Shea, T. Zadra, D. Klamer, E. Frangione, and R. Brouillard, "Improved state estimation through use of roads in ground tracking," in *Proceedings of Signal and Data Processing of Small Targets*, vol. 4048. SPIE, 2000, pp. 312–332.
- [5] —, "Precision tracking of ground targets," in *Proceedings of Aerospace Conference*, vol. 3. IEEE, 2000, pp. 473–482.
- [6] X. R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Trans. Automat. Contr.*, vol. 41, no. 4, pp. 478–493, Apr. 1996.
- [7] U. Forssell, P. Hall, S. Ahlqvist, and F. Gustafsson, "Novel map-aided positioning system," in *Proceedings of FISITA*, 2002.
- [8] N. Svenzen, "Real time implementation of map aided positioning using a Bayesian approach," Master's thesis, Linköping University, Department of Electrical Engineering, Linköping, Sweden, 2002.
- [9] C. Yang, M. Bakich, and E. Blasch, "Nonlinear constrained tracking of targets on roads," in *Proceedings of International Conference on Information Fusion*, vol. 1, July 2005, pp. 235–242.
- [10] D. Salmond, M. Clark, R. Vinter, and S. Godsill, "Ground target modelling, tracking and prediction with road networks," in *Proceedings of International Conference on Information Fusion*, July 2007.
- [11] N. Castañeda, M. Charbit, and E. Moulines, "A new bearings-only tracking algorithm for ground moving targets constrained to roads," in *Proceedings of IEEE Workshop on Signal Processing Advances in Wireless Communications*, June 2007.
- [12] J. Koller and M. Ulmke, "Road-map assisted ground target tracking," *Aerospace Science and Technology*, vol. 11, no. 4, pp. 261–270, May 2007.
- [13] M. Zhang, S. Knedlik, and O. Loffeld, "Road-constrained target tracking in GSM networks," in *Proceedings of International Symposium on Wireless Communication Systems*, Oct. 2007, pp. 138–142.
- [14] M. Ekman, K. Davstad, and L. Sjöberg, "Ground target tracking using acoustic sensors," in *Proceedings of Information, Decision and Control*, Feb. 2007, pp. 182–187.
- [15] B. Pannetier, V. Nimier, and M. Rombaut, "Multiple ground target tracking," *Aerospace Science and Technology*, vol. 11, no. 4, pp. 271–278, May 2007.
- [16] H. Johansson, "Road-constrained target tracking using particle filter," Master's thesis, Linköping University, Department of Electrical Engineering, Linköping, Sweden, 2008.
- [17] M. Zhang, S. Knedlik, and O. Loffeld, "On nonlinear road-constrained target tracking in GSM networks," in *Proceedings of IEEE Vehicular Technology Conference*, May 2008, pp. 2026–2030.
- [18] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.
- [19] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [20] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forsell, J. Jansson, R. Karlsson, and P. Nordlund, "Particle filters for positioning, navigation and tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 425–437, Feb. 2002.
- [21] A. Doucet, S. Godsill, and C. Andrieu, "On sequential simulation-based methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [22] S. McGinnity and G. W. Irwin, "Multiple model bootstrap filter for maneuvering target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 3, pp. 1006–1012, July 2000.
- [23] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 613–624, Mar. 2001.
- [24] A. Doucet, A. Logothetis, and V. Krishnamurthy, "Stochastic sampling algorithms for state estimation of jump Markov linear systems," *IEEE Trans. Signal Processing*, vol. 45, no. 1, pp. 188–202, Jan. 2000.
- [25] Y. Boers and J. Driessen, "Interacting multiple model particle filter," *IEE P-Radar Son. Nav.*, vol. 150, no. 5, pp. 344–349, Oct. 2003.
- [26] H. Driessen and Y. Boers, "Efficient particle filter for jump Markov nonlinear systems," *IEE P-Radar Son. Nav.*, vol. 152, no. 5, pp. 323–326, Oct. 2005.
- [27] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions," *IEEE Trans. Signal Processing*, vol. 51, no. 7, pp. 1762–1770, July 2003.
- [28] F. Caron, M. Davy, E. Duflos, and P. Vanheeghe, "Particle filtering for multisensor data fusion with switching

observation models: Application to land vehicle positioning,” *IEEE Trans. Signal Processing*, vol. 55, no. 6, pp. 2703–2719, June 2007.

- [29] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “A novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings on Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
- [30] M. S. Arulampalam, N. Gordon, M. Orton, and B. Ristic, “A variable structure multiple model particle filter for GMTI tracking,” in *Proceedings of International Conference on Information Fusion*, vol. 2, July 2002, pp. 927–934.
- [31] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. London: Artech House, 2004, ch. 10.
- [32] M. Ulmke and W. Koch, “Road-Map assisted ground target tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 3, pp. 1264–1274, Oct. 2006.
- [33] Y. Cheng and T. Singh, “Efficient particle filtering for road-constrained target tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1454–1469, Oct. 2007.
- [34] O. Payne and A. Marrs, “An unscented particle filter for GMTI tracking,” in *Proceedings of Aerospace Conference*, vol. 3, Mar. 2004, pp. 1869–1875.
- [35] M. Ekman and E. Sviestins, “Multiple model algorithm based on particle filters for ground target tracking,” in *Proceedings of International Conference on Information Fusion*, July 2007.
- [36] G. Kravaritis and B. Mulgrew, “Variable-mass particle filter for road-constrained vehicle tracking,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 2008.
- [37] L. Hong, N. Cui, M. Bakich, and J. R. Layne, “Multi-rate interacting multiple model particle filter for terrain-based ground target tracking,” *IEE Proc.-Control Theory Appl.*, vol. 153, no. 6, pp. 721–731, Nov. 2006.

BIOGRAPHY



Umut Orguner received B.S., M.S. and Ph.D. degrees all in electrical engineering from Middle East Technical University, Ankara, Turkey in 1999, 2002 and 2006 respectively. Between 1999 and 2007, he was with the Department of Electrical and Electronics Engineering of the same university as a teaching and research assistant. Since January 2007 he has been working as a postdoctoral associate in Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden. His research interests include estimation theory, multiple-model estimation, target tracking and information fusion.



Thomas B. Schön was born in Sweden in 1977. In Feb. 2006 he received the PhD degree in Automatic Control from the Division of Automatic Control, Linköping University, Linköping, Sweden. In Sep. 2001 he received the MSc degree in Applied Physics and Electrical Engineering and in Feb. 2001 he received the BSc degree in Business Administration and Economics, both from Linköping University. He has held visiting positions at the University of Cambridge (UK) and the University of Newcastle (Australia). His research interests are mainly within the areas of sensor fusion, signal processing, and system identification, with applications to the automotive and the aerospace industry. He is currently an Associate Professor at Linköping University.



Fredrik Gustafsson is professor in Sensor Informatics at Department of Electrical Engineering, Linköping University, since 2005. He received the M.Sc. degree in electrical engineering 1988 and the Ph.D. degree in Automatic Control, 1992, both from Linköping University. During 1992-1999 he held various positions in automatic control, and in 1999 he got a professorship in Communication Systems. In 2004, he was awarded the Arnberg prize by the Royal Swedish Academy of Science (KVA) and in 2007 he was elected member of the Royal Academy of Engineering Sciences (IVA). His research interests are in stochastic signal processing, adaptive filtering and change detection, with applications to communication, vehicular, airborne and audio systems, where the current focus is on sensor fusion algorithms for navigation and tracking problems. He was an associate editor for *IEEE Transactions of Signal Processing* 2000-2006 and is currently associate editor for *EURASIP Journal on Applied Signal Processing* and *International Journal of Navigation and Observation*.