Abstract – In this paper a novel multiple model particle filter algorithm for tracking ground targets on constrained paths is developed. The algorithm is designed to let the different modes be represented by constrained likelihood models, whereas the state dynamics are the same for all models. The mixing procedure is performed over the likelihood models and the mixing parameters are calculated in a standard interacting multiple model (IMM) manner. The performance of the developed estimator is compared with several other multiple model particle filters in a Monte Carlo simulation study. A ground target scenario consisting of road networks is used to evaluate the behaviour of the tracking filters and to illustrate the selection of design parameters.

Keywords: Particle Filter, Ground Target Tracking, IMM algorithms.

1 Introduction

The interacting multiple model (IMM) filter [1] has been suggested by several authors to solve various target tracking problems. Most of the work on IMM estimators has considered filtering using switching dynamic models to describe for instance jump Markov systems. Typically, different models representing distinct manoeuvres of the target are utilized.

Traditionally within the IMM framework, Kalman filters (KF) have been used for linear dynamic models and extended KF for nonlinear models. However, the last decade it has been an increasingly interest in using particle filters (PF) as estimators in multiple model (MM) applications, see for instance [2], [3], and [4]. An important reason for this is that PF provides a general method to the Bayesian nonlinear filtering problem. Particle filters have been proven to work well in many applications where the standard IMM based on KF’s are inappropriate, see e.g. [3].

Ground target tracking is an application where multiple models PF have been used with promising results. The ground target tracking scenarios often include road networks. Constrained paths imposed on the targets and hard constraints on e.g. speed or acceleration, lead to highly non-Gaussian densities. Some targets may also move off the road and, thus, the constraints on the target are dependent on the terrain condition. One approach to handle these difficulties using standard Kalman filtering techniques is the so-called variable structure multiple model algorithm [5]. Nevertheless, constraints on the state vector are naturally incorporated in the estimation problem within the PF framework, and PF approaches have been successfully implemented in ground targets applications, see e.g. [4], [6] and [7].

In this paper we propose a new algorithm based on the sampling importance resampling algorithm [8] and IMM techniques. The novelty lies in that the different modes are represented by constrained likelihood density models, whereas the state dynamics is allowed to be the same for all models. The basic principle is to take into account the probability of each likelihood model in the correction step, where the mixing probabilities are computed in a standard IMM manner. In the time update step the predicted particles, representing different target motion modes, are obtained from passing the particles through the same dynamics. Since the likelihood functions have different structure whereas the dynamics is fixed, we term the algorithm as a multiple likelihood model particle filter (MLM-PF). The performance of the MLM-PF algorithm is investigated under hard constraint conditions using a ground target scenario consisting of road networks. The target may also move in e.g. an open field and different models representing the on/off-road motion are implemented. The performance of the developed MLM-PF method is compared with other multiple model particle filters in a Monte Carlo simulation study. Simulations are also performed to illustrate the effects from different design parameters on the tracking behaviour.

This paper is organized as follows: In Section 2 the tracking system is presented together with the problem formulation, where focus is on different methods of implementing hard constraints within the PF framework. Section 3 develops the MLM-PF algorithm and gives a brief presentation of other multiple model PF approaches. A description of the ground target simulation scenario together with discussions on the results from the study is given in Section 4. Finally, concluding remarks are presented in Section 5.
2 System description and problem formulation

2.1 Dynamics and measurement models

The state vector at the time $k$ is $x_k = [X_k \ Y_k \ \dot{X}_k \ \dot{Y}_k]^T$, where $X$ and $Y$ represent the position in Cartesian coordinates. A discrete time kinematic model for the problem can be written as

$$x_k = F x_{k-1} + G v_{k-1}, \quad k \in \mathbb{N}$$

where $x_k \in \mathbb{R}^4$ and $v_{k-1,j}$ are a white Gaussian process noise sequence with the variances $\sigma_{v}^2$ and $\sigma_{w}^2$ along the $X$ and $Y$ directions. Thus, the covariance is given by $Q = \mbox{diag}([\sigma_{v}^2 \ \sigma_{w}^2])$. The system matrices are

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix},$$

where $T$ is the sampling time.

The applied measurement model is given by

$$z_k = h(x_k) + e_k,$$

where the azimuth and range measurements can be expressed as

$$h(x_k) = \begin{bmatrix} \theta_k \\ \rho_k \end{bmatrix} = \begin{bmatrix} \tan^{-1}(X_k / Y_k) \\ \sqrt{X_k^2 + Y_k^2} \end{bmatrix}.$$ (4)

The measurement noise is Gaussian, $e_k \sim \mathcal{N}(0,\mathbf{R})$, with the Covariance matrix $\mathbf{R} = \mbox{diag}([\sigma_\theta^2 \ \sigma_\rho^2])$.

2.2 Problem description

The solution to the state estimation problem is given within the Bayesian filtering framework where the algorithms are based on particle filters. In the sequel we let $Z_k = [z_{k1}, \ldots, z_{kn}]^T$ denote the sequence of measurements up to time $k$. The Bayesian recursion of the posterior filter density is given by

$$p(x_k | Z_k) \propto p(z_k | x_k) p(x_k | Z_{k-1}),$$ (5)

$$p(x_{k1+1} | Z_k) = \int_{x_k} p(x_{k1+1} | x_k) p(x_k | Z_k) dx_k.$$ (6)

The solution to the state estimation problem can be written as

$$p(z_k | x_k) = p(z_k | x_{k+1} - F x_k),$$

where $w_i = Gv_i$. 

2.3 Hard constraints

For particle filter methods hard constraints on the state vector or the measurement process are naturally incorporated in the estimation problem. The reason is that no restrictions on the distribution of the noise or the initial density are imposed using the particle filter approach, and the posterior constrained density is therefore easily constructed. Let the set of constrained state vectors be denoted by $\Psi$. Then, the state estimate is given by the mean of the posterior constrained density $p(x_k | Z_k, \psi)$. Using particle filters, $p(x_k | Z_k, \psi)$ is readily assembled by modifying the initial distribution and the process noise density. An easy technique to sample from this kind of distributions is to use rejection sampling from the unconstrained distributions until the condition $x_k \in \psi$ is satisfied.

In this section two different approaches of implementing constraints on state variables, for example due to the terrain in ground or sea tracking, will be emphasized. Implementation features of these two approaches are given in [11]. In [12], different aspects of the two approaches are discussed. Assuming that the particle filter is represented by the SIR method, the following two implementation methods for handling constraints are considered:

Constraints via time update:
In this case the constraints are implemented via the process noise. In the update equation, after the addition of process noise, a predicted particle, $x_{k+1}^{(n)}$, is rejected if it does not belong to the set, i.e.:

$$p_{\psi_k}(v_k, x_k) \propto \begin{cases} p_{\psi_k}(\Gamma_k), & x_{k+1}^{(n)} \in \Psi \\ 0, & otherwise \end{cases},$$

where $p_{\psi_k}(\Gamma_k)$ is an appropriate, possibly truncated, density. Thus, we need to draw samples from a state dependent truncated process noise density. If samples are easily drawn from $p_{\psi_k}(\Gamma_k)$, then samples from $p_{\psi_k}(v_k, x_k)$ can be obtained by rejection sampling from $p_{\psi_k}(\Gamma_k)$ until $x_{k+1}^{(n)} \in \psi$ is satisfied.

Constraints via measurement update:
In the second method the constraints are introduced in the update of the likelihood, i.e. the new constrained likelihood becomes:

$$p_\psi(\bar{z}_k) = \begin{cases} p_\psi(z_k - h(x^{(n)}_k)), & x^{(n)}_k \in \Psi \\ 0, & otherwise \end{cases}.$$ (10)

Hence, the importance weights are set equal to zero if the particles don’t belong to the set after the measurements update.

The method suggested in (9) is commonly used to incorporate hard constraints in particle filters. However,
one important drawback with this method is that it might be difficult to draw samples from the distribution $p_{\gamma_i}(v_i)$. especially if exact draws from $p_{\gamma_i}(v_i)$ are to be considered. This also means that the samples have to be generated within a reasonable time. Acceptance-rejection sampling that is time effective utilizes for instance techniques where, instead of simulating a new particle, a current particle inside the constrained set acts as the new sample from the density if the sample previously has been rejected. However, those kinds of algorithms only generate samples that are approximately distributed as the target process noise density. There are several alternative approaches where the pdf in (9) is re-constructed to simplify the acceptance-rejection sampling. For some applications it is possible to assemble the process noise density such that particles predicted outside the region $\Psi$ are properly mapped to the boundary of $\Psi$. This is convenient especially when considering magnitude constraints of vectors, for instance the velocity. Then the magnitude can be manipulated by proper scalar multiplication knowing that the direction of the vector will be the same. Nevertheless, when considering scalar quantities it might be ambiguous where at the boundary of $\Psi$ the particles should be located. Moreover, it is a considerable risk that a majority of the particles end up just at the boundary of $\Psi$, which will give a poor representation of the “true” posterior density. A simple approach is to map particles outside the region on to particles that are inside $\Psi$. However, if few particles are inside the region, again, it is a risk that the particle cloud becomes degenerated. One way of avoiding this is, if possible, to increase the probability that the particles will be propagated inside the set $\Psi$ by constructing $p_{\gamma_i}(v_i)$ in such a way that samples are drawn only in certain directions. This will be further discussed in Section 4.

The approach in (10) is generally implemented within a particle filter framework. It is not necessary to sample from a perhaps complicated noise pdf. Hence, the additional information needed is the shape of the set $\Psi$. Particles which are not meeting the constraints will just have their associated weight set equal to zero or equal to some small value just before the resampling step. One disadvantage compared to (9) is that the effective sample size may be reduced in (10). However, as it will be shown in Section 4, this will not necessarily have a negative effect on the ground target tracking performance for the MLM-PF method.

### 3 Tracking algorithms

#### 3.1 Multiple likelihood model particle filter

In this section we present a novel approach, the MLM-PF algorithm for multiple model filtering under hard constraint conditions, which is based on a combination of standard MM algorithms and particle filters. The basic idea is to let the different modes within the MM estimator framework be represented by constrained likelihood models, whereas the state dynamics is the same for all models. Hence, if the regimes considered are e.g. hard constraints acting on the system and unconstrained system. Then the models used for calculating the mixing parameters will be (8) and (10), and the dynamics used for the time update is (1).

Let $M^j_k$ denote $j=1,...,m$ different models at time $k$. Furthermore, the probability for each model is expressed by $\gamma^j_k = P(M^j_k | Z_k)$. Now, applying the total probability theorem with $m$ different models the posterior density can be written as

$$p(x_k | Z_k) = \sum_{j=1}^m \gamma^j_k p(x_k | M^j_k, Z_k).$$

(11)

Using Bayes’ theorem on the second factor in (11) gives

$$p(x_k | M^j_k, Z_k) \propto p(z_k | M^j_k, x_k) p(x_k | M^j_k, Z_{k-1}).$$

(12)

Since the models only differ in the measurement update, i.e. constraints are imposed on the likelihood function, we will in what follows set the different models as $M^j_k = \Psi^j_k$, where $\Psi^j_k$ is a constrained set acting on the states at time $k$. Thus, the pdf in (11) can be expressed as

$$p(x_k | Z_k) = \frac{1}{\alpha} \sum_{j=1}^m \gamma^j_k p(z_k | \Psi^j_k, x_k) p(x_k | \Psi^j_k, Z_{k-1}),$$

(13)

where $\alpha$ is a normalization factor. The probability of each model is also given by Bayes’ theorem

$$\gamma^j_k = P(\Psi^j_k | Z_k) = \frac{1}{c} p(z_k | \Psi^j_k, Z_{k-1}) P(\Psi^j_k | Z_{k-1})$$

(14)

$$= \frac{1}{c} p(z_k | \Psi^j_k, Z_{k-1}) \sum_{j=1}^m p_{\gamma_j} \gamma^j_k,$$

where $c$ is a normalization factor and $p_{\gamma_j}$ is the probability $P(\Psi^j_k | \Psi^i_{k-1}, Z_{k-1})$ for $i, j \in M = \{1,2,...,m\}$. In the sequel this will be modelled as a first-order Markov chain with transitional probabilities

$$p_{\gamma_{ij}} = P(\Psi^j_{k+1} | \Psi^i_k),$$

(15)

where $p_{\gamma_{ij}} \geq 0$, $\sum_{j=1}^m p_{\gamma_{ij}} = 1$, for all $i, j \in M$. The model conditioned likelihood in (14) is assumed Gaussian as

$$p(z_k | \Psi^j_k, Z_{k-1}) \approx l^j_k = p(z_k | \hat{\xi}_{ij}^k, \hat{S}_{ij}^k)$$

(16)

$$= \mathcal{N}(z_k - h(\hat{\xi}_{ij}^k);0,S^k_{ij}),$$

where $\hat{\xi}_{ij}^k$ is the minimum mean square (MMS) estimate and $\hat{S}_{ij}^k$ is the innovation covariance matrix from a filter estimate with model $j$.

It is straightforward to use (13) and (14) within the particle filtering framework. The basic idea in the MLM-PF algorithm is to let a set of particles, generated by resampling with replacement among previously predicted particles, approximate the total posterior in (11). Interaction between the models is then performed before the resampling step by mixing the acceptance probabilities. Hence, the density $p(x_k | \Psi^j_k, Z_{k-1})$ is approximated with samples. Those samples are, using the SIR scheme, generated from the underlying model (1) and the filtering particles from the previous time step. Since
(1) constitutes only one mode, we make the model assumption that the mode conditioning in the posterior density in (12) is caused only by the likelihood model conditioned on mode $\Psi_k^j$, i.e., (12) can be written as

$$p(x_k \mid M^j_k, Z_k) \approx p(z_k \mid M^j_k, x_k)p(x_k \mid Z_{k-1}) \quad (17).$$

Then, the following assumption about the transitional weights of the samples as

$$p(x_k \mid \Psi_k^j, x_{k-1}^j) = \mathcal{N}(x_k; F x_{k-1}^j, Q) = p(x_k \mid x_{k-1}^j) \quad (18)$$

and, thus, the predicted particles are generated from a common prediction density as

$$x_{k+1}^{(n)} - p(x_{k+1} \mid Z_k) = \frac{1}{N} \sum_{n=1}^{N} p(x_{k+1} \mid x_{k}^{(n)}) \quad (19),$$

where $N$ is the number of particles. The particles $x_{k}^{(n)}$ are obtained from the resampling (with replacement) step. Note that the predicted particles are not conditioned on the model $j$ since they are obtained from passing the particles through the same dynamics in the time update.

From (13) it is obvious that we need to consider the probability of each likelihood model in the correction step. This probability is obtained by approximating the MMS estimates used in (16) as

$$\hat{x}_k^j = \sum_{n=1}^{N} w_{k}^{(n),j} x_{k}^{(n)}, \quad (20)$$

where $w_{k}^{(n),j}$ is the weight associated to particle $x_{k}^{(n)}$. Now, according to (13) and (16) the probability of each mode or model is given by

$$\gamma_k^j = \frac{1}{C} l_k^{j} \sum_{i=1}^{c} p_{i}^k \gamma_k^{j-1} \quad (21).$$

Given the mode probability it is possible to compute the weights of the samples as

$$w_{k}^{(n),j} = \frac{1}{\alpha} \gamma_{k}^{j}, p_{x} (z_k \mid \Psi_k^j, x_{k}^{(n)}) \quad (22).$$

Worth noticing is that even if the model conditioned likelihood is zero for some particles, the total weight for this particle is not necessarily zero. The sampling importance resampling (SIR) MLM-PF algorithm is summarized in Table 1.

Table 1. SIR MLM-PF algorithm

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>For $k=0$, generate $N$ samples ${x_{k,0}^{(n)}}<em>{n=1}^{N}$ from the initial distribution $p</em>{x_0}(x_0)$</td>
</tr>
<tr>
<td>2.</td>
<td>Calculate the importance weights conditioned on the constrained set $\bar{w}<em>{k}^{(n),j} = p</em>{x} (z_k \mid \Psi_k^j, x_{k}^{(n)})$ and normalize $w_{k}^{(n),j} = \bar{w}<em>{k}^{(n),j} / \sum</em>{n=1}^{N} \bar{w}_{k}^{(n),j}$ for $n=1, \ldots, N$ and $j=1, \ldots, m$</td>
</tr>
<tr>
<td>3.</td>
<td>Compute the estimates $\hat{x}_k^j$ and $\hat{S}_k^j$ (the mixing conditions) from (20) using the weights from step 2</td>
</tr>
<tr>
<td>4.</td>
<td>For $j=1, \ldots, m$, calculate the conditioned likelihood functions $l_k^j = p(z_k \mid \hat{x}_k^j, \hat{S}_k^j)$</td>
</tr>
<tr>
<td>5.</td>
<td>For $j=1, \ldots, m$, update the mode probability $\gamma_k^j = \frac{1}{C} l_k^{j} \sum_{i=1}^{c} p_{i}^k \gamma_k^{j-1}$ where $c = \sum_{j=1}^{c} l_k^{j} (\sum_{i=1}^{c} p_{i}^k \gamma_k^{j-1})$</td>
</tr>
<tr>
<td>6.</td>
<td>For $n=1, \ldots, N$ and $j=1, \ldots, m$, calculate the combined weights $w_{k}^{(n)} = \frac{1}{\alpha} \sum_{j=1}^{m} \gamma_{k}^{j} w_{k}^{(n),j}$, where $\alpha = \sum_{n=1}^{N} w_{k}^{(n)}$</td>
</tr>
<tr>
<td>7.</td>
<td>Generate a new set ${x_{k}^{(n)}}<em>{n=1}^{N}$ by resampling where $p(x</em>{k}^{(n)} = x_{k}^{(n)} = w_{k}^{(n)}$</td>
</tr>
<tr>
<td>8.</td>
<td>For $n=1, \ldots, N$, predict new particles according to $x_{k+1}^{(n)} = F x_{k}^{(n)} + G v_k^{(n)}$, where $v_k^{(n)}$ is drawn from the process noise pdf</td>
</tr>
<tr>
<td>9.</td>
<td>Increase $k$ and continue from step 2</td>
</tr>
</tbody>
</table>

### 3.2 IMM particle filter via the time update

It is clear that the same concept as in Section 3.1 can be used for handling hard constraints via the time update in (9). One important aspect when incorporating constraints via (9) is how to construct the process noise model. The actual implementation of the process noise dynamics is further discussed in Section 4. The handling of constraint models via the time update implies that (17) is no longer a feasible assumption. Instead the predicted particles are conditioned on the model $j$ according to $x_{k}^{(n),j} = p(x_{k+1} \mid \Psi_k^j, Z_k)$. Since the predicted particles will belong to different modes, it is not reasonable to approximate (11) using particles generated from resampling with replacement among previously predicted particles originating from a common transitional prior.

There are several possible approaches to overcome this problem. One method would be to resample with replacement among predicted particles conditioned on model $j$, which means that the approximated posterior pdf obtained from the resampling step will also be conditioned on model $j$. These conditioned posterior densities are then used as the foundation in the interaction and combination stages. An obvious drawback using this approach is the fact that a sample of sufficiently large size needs to be drawn from each mode, which means that the samples have to be passed trough the time consuming resampling step. In [13] this problem is solved by simply avoiding the resampling step and instead applying a hybrid version of the bootstrap filter. The mixing probabilities are then calculated with standard IMM using approximated continuous pdf and MMS estimates. Another benefit of utilizing the hybrid bootstrap filter is that the degeneracy of the effective number of particles introduced by using resampling may be prevented. However, the mixture reduction algorithm (see, [14]) suggested in [13] can be rather time consuming, if the additional threshold values established in the clustering calculations are not tuned with care.

The IMM particle filter considered in this section, and which will be further investigated in Section 4, is based on the algorithm suggested in [2]. This strategy, which originally is designed for jump Markov systems, will hereafter be referred to as the (E)fficient IMM-PF. Instead
of using a Gaussian sum approach, as in [7], the conditional pdf for the state in mode $j$ is approximated by the particles obtained from a resampling step. The EIMM-PF algorithm, including the interaction stage, is described in more detail in [2]. It should be emphasized that within the IMM framework, updating constraint models via (9) implies that a mixing process needs to be performed to produce the posterior density at $k-1$ for each model $j$. This interaction is done before the filtering step. In the simulation study in Section 4 particles representing the constrained model that are predicted outside the constrained region $\Psi$ are mapped on to particles that are inside the region.

### 3.3 Multiple model particle filter

In this section the multiple-model particle filter (MM-PF) approach for solving the hard constraint problem is briefly described. We let an augmented state vector be defined as $s_k = [x_k^T \ r_k^T]$, where $r_k$ represents the model (regime) in effect during the current sampling period. The sequence of regimes and the transitional probabilities in set to $\pi_j = \pi_j(r_k)$, with a speed of 4 m/s along the path shown as dots in Figure 1. There is a field where the vehicle will take off the road. A target is turning right on to a smaller road. Thereafter the target is turning left (after 200 seconds) and once again moving in a northerly direction. Suddenly, when 275 seconds have elapsed, the target is travelling off the road. The off-road motion is interrupted after 375 seconds, when the target is turning right on to a road again. All target manoeuvres are preceded by natural accelerations and brakings.

The simulation scenario is depicted in Figure 1, where the target trajectory is indicated by the dots.

The sensor is fixed and located at the origin. We assume that there are no false alarms and no missed detections. Accuracies of the azimuth measurement is $\sigma_\phi=0.05$ rad, and for the range $\sigma_r=4$ m. It is assumed that the true measurement noise variances are known. The measurements are received at a sampling rate of $T=5$ seconds. It is assumed that dynamics of the target is represented by the model (1)-(4). The same process noise parameters, $Q = \text{diag}[0.05^2 \ 0.05^2 \ 0.05^2 \ 0.01^2]$, and initial covariance matrix, $P_0 = \text{diag}[0.5^2 \ 0.5^2 \ 0.01^2 \ 0.01^2]$, were used for all algorithms. The number of particles were $N=1000$ in all simulation runs, and the scenario lasts for 450 seconds. Two modes where used in the multiple-model algorithms. The first mode corresponds to a road motion model, whereas the second mode corresponds to off-road motion. Thus, for the EIMM-PF algorithm there are 500 particles in each mode. To evaluate the performance, a root mean square error RMSE analysis for the position is performed over 100 simulations.

In the MC simulation runs the transition matrix, where the first state is on-road motion and the second state is off-road motion, is given for all multiple-model algorithms as

$$ P_{i,j} = \begin{bmatrix} P_{on} & 1-P_{on} \\ 1-P_{off} & P_{off} \end{bmatrix}, $$

where $P_{on}$ is the probability that a target travelling on the road will stick to the road at next time step. Thus, $P_{off}$ is the corresponding regime probability for the off-road motion. It is assumed that the a priori probability that a target will get off the road is relatively low, and the probability that a target travelling off-road will enter on to a road is equal to probability that the target will keep the off-road motion. Hence, we have that $P_{off} = 0.5$, whereas
$P_{on}$ is considered as a design variable and different values for $P_{on}$ will be investigated in the simulation study.

In the first simulation runs the applied process noise in the propagation step was chosen the same for all algorithms. Figure 2 shows a snapshot of the MLM-PF algorithm as the target is moving along one road segment. From the figure it can be seen that most of the particles are spread along the road. However, since the off-road probability mode is not zero, some particles also appear outside the road.

Figure 2. A snapshot of the MLM-PF after 125 seconds.

In Figures 3 and 4 the mean RMSE values for 100 MC simulation runs using $P_{on} = 0.97$ are presented (the simulation period is split into two parts for illustrative reasons). With this choice of $P_{on}$ the RMSE for the MLM-PF algorithm has a better behaviour than the other algorithms. This is also illustrated in Figure 5, where the overall RMSE performance is presented in the upper plot. The “total” RMSE for the position is defined as

$$RMSE_{tot} = \frac{1}{N_t} \sum_{t=1}^{N_t} RMSE(t),$$

where $t$ is the sample index and the number of samples is $N_t = 90$. Figure 5 shows that the MLM-PF algorithm outperforms the other algorithms for higher values of $P_{on}$. In Figure 5, 30 MC simulation runs for each value of $P_{on}$ have been used, and the tracking performance where the time period for the true target off-road motion is excluded from the RMSE measure is also shown (lower plot). The results indicate that with the chosen process noise the optimal value of $P_{on}$ for the MLM-PF algorithm is approximately 0.97. For the other two multiple model algorithms it seems that values larger than 0.9 give worse tracking performance compared with the standard PF, also when the target is travelling on the road.

The different mode probabilities for the MLM-PF algorithm using $P_{on} = 0.97$ for one simulation run are illustrated in Figure 6. In Figure 7 the effective sample size, defined as $N_{eff} = 1/\sum_{m=1}^{M} (w_i^{(m)})^2$, for the multiple-model approaches is presented. For the EIMM-PF algorithm the effective sample size is calculated as the sum of the effective sample size from each mode. The results in Figure 7 show that the $N_{eff}$ is significantly lower for the MLM-PF algorithm compared with the other algorithms. However, even if there is a correlation between the tracking performance and the effective sample size (compare with the results in Figures 3 and 4) the degradation of the tracking performance is not as severe as for the MM-PF. The results also indicate that the particle clouds, especially for the MM-PF algorithms, become degenerated around mode transitions and at manoeuvres. This is probably the explanation to the less favourable performance shown in Figure 5.
One possibility to improve the behaviour of the EIMM-PF and the MM-PF algorithms would be to use directional process noise for the models corresponding to the on-road mode. This will be exploited next.

In the second simulation study a directed process is applied to the EIMM-PF and MM-PF algorithms. An important design variable in PF is the process noise. The behaviour of the EIMM-PF and MM-PF algorithms are also investigated using process noise that is constructed in such a way that samples are propagated only in directions along the roads for the on-road model. The statistical meaning of such an approach is that the variance along the road is high and the uncertainty orthogonal to the road is low. Hence, if $\sigma_{\text{along}}^2$ denotes the variance along the road and $\sigma_{\text{ortho}}^2$ denotes the variance orthogonal to the road. Then, the values $\sigma_{\text{along}} = 0.1$ and $\sigma_{\text{ortho}} = 0$ are used in the simulations. Directional process noise is also implemented in the MLM-PF algorithm in such a way that the directional process noise above is applied in the propagation step on particles positioned on the road. In Figure 8 the overall RMSE$_{\text{tot}}$ performance for different values of $P_{\text{on}}$ is presented, using 30 MC simulation runs for each value of $P_{\text{on}}$. The results indicate that the performance of the EIMM-PF and MM-PF approaches can be enhanced by using the concept of directional process noise, whereas the behaviour of the MLM-PF method seems to be worse when applying process noise in certain directions. The best performance for the MLM-PF algorithm is obtained when non-directional process noise is used by choosing $P_{\text{on}} = 0.98$ (RMSE$_{\text{tot}} = 2.4$). For the EIMM-PF and MM-PF methods the best performance is achieved when directional process noise is applied, and when the on-road regime probability is approximately $P_{\text{on}} = 0.9$ (RMSE$_{\text{tot}}=2.6$) and $P_{\text{on}} = 0.8$ (RMSE$_{\text{tot}}=2.7$), respectively.

The mean effective sample size for the whole time period and for all MC simulations, where $P_{\text{on}} = 0.97$, is given in Table 2. In the table the mean $N_{\text{eff}}$ using directional process noise is compared with the results from the MC simulations without directional process noise. The result shows that the $N_{\text{eff}}$ is increased for all algorithms when directional process noise is utilized. However, in contrast to the other methods it seems that this is not improving the tracking performance for the proposed MLM-PF method. It is worth mentioning that the introduction of directional process noise in the algorithms also increases the computational burden.

<table>
<thead>
<tr>
<th>Method</th>
<th>MLM</th>
<th>EIMM</th>
<th>MM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directional noise</td>
<td>194</td>
<td>303</td>
<td>616</td>
</tr>
<tr>
<td>Non-directional noise</td>
<td>134</td>
<td>265</td>
<td>555</td>
</tr>
</tbody>
</table>

Figure 5. RMSE for different $P_{\text{on}}$.

Figure 6. Mode probabilities for MLM-PF with $P_{\text{on}}=0.97$.

Figure 7. Effective sample size, $N_{\text{eff}}$, with $P_{\text{on}}=0.97$.

Figure 8. RMSE for different $P_{\text{on}}$ with directional process noise.
5 Conclusions

This paper has developed a novel particle filter algorithm for handling multiple models under hard constraints. The different modes in the filter are represented by constrained likelihood models, and the method is therefore referred to as the multiple likelihood model particle filter (MLM-PF). One important benefit with the MLM-PF algorithm is that the same set of particles is used for all modes. The performance of the MLM-PF algorithm has been evaluated, and compared with other multiple model PF based methods, in a ground target road network simulation study. It is shown that the suggested MLM-PF algorithm has in general an excellent tracking performance, and under some conditions outperforms the other multiple model PF methods. Moreover, since the implementation of the constrained models is conducted in the measurement update, there is no need to sample from complicated process noise pdf.

In order to amplify the number of effective samples for the on-road motion mode, the concept of the so-called directional process noise was applied. The effective sample size was increased for all algorithms. For the proposed MLM-PF algorithm this didn’t improve the tracking performance as it did for the other investigated algorithms.

Future work will include addressing the data association problem for multi-target situations. Furthermore, it is of great interest to study the proposed algorithm under more difficult conditions, i.e. to add false alarms and missed detection situations, as well as to evaluate the method using real data.

ACKNOWLEDGEMENT

This work has been partly financially supported by the Swedish and Dutch MOD representatives, as a European MOU, ERG no. 01, research and technology project. This support is gratefully acknowledged.

References