Efficient Particle Filtering for Road-Constrained Target Tracking

Yang Cheng
Department of Mechanical and Aerospace Engineering
University at Buffalo, State University of New York
Amherst, NY 14260
Email: cheng3@buffalo.edu

Tarunraj Singh
Department of Mechanical and Aerospace Engineering
University at Buffalo, State University of New York
Amherst, NY 14260
Email: tsingh@buffalo.edu

***Abstract*— The variable-structure multiple model particle filtering approach for state estimation of road-constrained targets is addressed. The multiple models are designed to account for target maneuvers including "move-stop-move" and motion ambiguity at an intersection; the time-varying active model sets are adaptively selected based on target state and local terrain condition. The hybrid state space is partitioned into the mode subspace and the target subspace. The mode state is estimated based on random sampling; the target state as well as the relevant likelihood function associated with a mode sample sequence is approximated as Gaussian distribution, of which the conditional mean and covariance are deterministically computed using nonlinear Kalman filtering. The importance function for the sampling of the mode state approximates the optimal importance function under the same Gaussian assumption of the target state.**

I. Introduction

A Notable characteristic of ground target tracking is that prior nonstandard information such as target speed constraints, road networks, and so forth can be exploited in the tracker to reduce the uncertainty of target motion and provide better estimates of the target state[1]. A tracker that ignores or is unable to make use of this additional source of information can only attain limited performance. In the cases of low signal-to-noise ratio, the incorporation of such constraint information is essential to successful tracking.

Multiple model estimation is widely used in the tracking community to tackle motion uncertainty. The interacting multiple model estimator[2] is one of the best known multiple model estimators. Recent applications of multiple model estimators to ground target tracking were presented in Ref. [1], [3], [4], [5], [6]. Kirubarajan and Bar-Shalom noted that for ground target tracking a multiple model estimator with fixed structure has to consist of a large number of models, owing to the many possible motion modes and various road constraints[3]. It is not only computationally undesirable but also potentially results in highly degraded estimates (due to the excessive "competition" among the many models). In order to overcome this problem, they proposed an adaptive or variable-structure interacting multiple model estimator for ground target tracking[3]. The basic idea is that the active model set varies in an adaptive manner and thus only a small number of active models are needed to be maintained at each time. Following the same idea of the variable structure interacting multiple model estimator, a variable structure multiple model particle filter was proposed for ground target tracking[1]. Simulation results showed that the particle filtering based approach has remarkably better error performance. The reasons for the superiority of this particle filtering based approach, as noted in Ref. [1], is that with particles or random samples the simulation-based particle filter is able to incorporate more accurate dynamics models and estimate non-Gaussian distributions (e.g., at an intersection) more accurately than the Kalman-filtering-based interacting multiple model estimator. The superiority of multiple model particle filter over the interacting multiple model estimator within the fixed structure multiple model framework was demonstrated in Ref. [7].

Multiple model estimation falls into the category of nonlinear filtering even if every single model is a linear system with Gaussian noise. A sufficient statistic of the hybrid state distribution with a fixed dimension is thus impossible. Moreover, the complexity of the optimal multiple model estimator increases exponentially with time[1]. Both the interacting multiple model estimator and the particle filter are suboptimal nonlinear filtering algorithms that maintain constant complexity and computational expense. The former maintains a constant number (i.e., the number of models) of Kalman filters while the latter maintains a constant number of (the most likely) particle trajectories. Such sub-optimality is inevitable for practical purposes.

One of the major concerns of the application of particle filtering to target tracking is its efficiency and computational expense. The convergence rate of ideal Monte Carlo sampling is only of the order of $\mathcal{O}(1/\sqrt{N})$, where N is the number of particles[7]. In particular, the bootstrap filter[1], [7] used in multiple model target tracking is simple to implement but is also known for its inefficiency. A large number of particles and therefore high computational expense are usually required in order to attain certain accuracy and robustness. The main objective of this paper is to derive an efficient particle filter for road-constrained targets. The basic idea is to reduce the whole sampling space of the multiple model system to the mode subspace by marginalization over the target subspace and choose better importance function for mode state sampling. The theoretical foundation of optimal particle filtering for ideal jump Markov linear Gaussian systems is given in Ref. [8], [9]. Of particular interest in the theory is that when the multiple model system is a jump Markov linear Gaussian system, the target state conditioned on a mode sample sequence is exactly Gaussian and therefore can be analytically determined using Kalman filtering; the mode state is the only part that needs to be estimated using particle filtering; because the mode state is of finite discrete values, the optimal importance function for the mode state can be used.

For our application the linear Gaussian assumption of each single model only approximately holds due to the nonlinear observation models. Hence, the sub-optimality of the proposed particle filter also results from the Gaussian approximation about the target state and the relevant likelihood function associated with each mode sample sequence. This particle filter is a combination of sampling approximation and analytic approximation and may outperform the particle filters solely based on sampling approximation when the Gaussian approximation holds to a satisfactory degree. Were the Gaussian approximation of the target state severely violated, a more general framework purely based on sampling approximation[10] would be desired. An analogous particle filter with similar algorithmic structure and Gaussian approximation was successfully applied in a very different context of multiple target data association[11]. As an aside, we note that when applied to nonlinear models, the interacting multiple model estimator makes the same Gaussian approximation about the target states. Furthermore, it approximates a Gaussian mixture with a single Gaussian distribution in the merging step.

The ground moving target indicator (GMTI) radar can provide detections on moving ground targets over a large region and has become an extremely useful sensor for surveillance of ground targets[1]. A special problem with GMTI tracking of ground targets is that when a target's radial velocity (along the line of sight from the sensor) falls below the minimum detectable velocity, the target cannot be detected by the sensor[4]. Thus, an evasive target can use the so-called "move-stop-move" strategy, in which it deliberately stops or moves at a very low speed for some time before accelerating again, to avoid detection by the sensor. The inference of target stop from the event of lack of detection is not trivial and can at best be done in a probabilistic manner because multiple causes may account for the same lack of detection. For example, in addition to target stop, the target may not be detected due to the less-than-one probability of detection. Under the assumption of confirmed tracks, Ref. [4] developed a simple but effective strategy to tracking the move-stop-move maneuver by augmenting the mode set with a stopped target model. More complex strategy involving data association was presented in Ref. [6]. Based on the assumption that the correct data association is known, we develop a strategy similar to Ref. [4] but in the context of particle filtering.

The organization of the paper proceeds as follows. First, the dynamics and observation models for road-constrained target are presented. Then the particle filtering theory for jump Markov systems is reviewed. Finally, an efficient particle filter is derived and compared with the bootstrap filter within the variable structure multiple model framework.

II. Dynamics and Observation Models

A. Target Dynamics Model

In road-constrained target tracking, knowledge of the road network, for example, in terms of the endpoint positions and directions of the road segments, is assumed to be available to the tracker. The road information incorporated in the dynamics models imposes constraints on feasible moving directions of on-road targets and thus greatly reduces motion uncertainty.

The road-constrained target is modeled as a point mass moving in a road-network on the horizontal plane. Its acceleration is modeled as a stochastic process. A natural local decomposition of the road-constrained target motion is the motion along the road and that perpendicular to the road, with the dominant motion being the former. The target motion on a road segment is best described in the local Cartesian coordinate system fixed to the road segment. The origin (reference point) of the coordinate system is chosen as an endpoint of the segment, while the x- and y-axes are aligned along and perpendicular to the road direction, respectively. In terms of local coordinates x^L and y^L, the continuous-time linear Gaussian target dynamics are described by

$$\ddot{x}^L(t) = w_x^L(t) \tag{1}$$

and

$$\ddot{y}^L(t) + 2\xi\omega_n\dot{y}^L(t) + \omega_n^2 y^L(t) = w_y^L(t) \tag{2}$$

where ξ is the damping factor and ω_n is the frequency. They are tuned to guarantee strictly stable off-road dynamics, which is preferred for tight control of y^L. The process noise $w_x^L \sim \mathcal{N}(w_x^L; 0, \sigma_{w_x}^2)$ and $w_y^L \sim \mathcal{N}(w_y^L; 0, \sigma_{w_y}^2)$ are Gaussian and white. The notation $\mathcal{N}(x; \hat{x}, \sigma^2)$ refers to a Gaussian distribution of x with mean $\hat{x}$ and variance σ^2. Note that different second-order models are used for the two orthogonal directions. In a more general setting, the models may even be of different orders. Higher order models for x^L may be used, for example, when the target is deemed highly maneuverable. On the other hand, reduced order models for y^L may be used, for example, when the off-road motion is considered much less significant or the observation accuracy is modest. In the limiting case, simply setting $y^L = 0$ for all the time would be sufficient.

Now suppose the present local coordinate frame is fixed on a directed road segment $\overrightarrow{AB}$, then the target is said to remain on the road segment if $0 \le x^L \le |\overrightarrow{AB}|$; it is said to leave the segment via the endpoint A if $x^L < 0$ and via the endpoint B if $x^L > |\overrightarrow{AB}|$. When the target leaves the road segment, the new segment or segments (when A or B is an intersection) it enters is determined by examining the adjoining segment or segments. After a new road segment is determined, the local coordinate system switches to the new segment. Subsequently, the new coordinate x^L is adjusted to $|x^L|$ or $x^L - |\overrightarrow{AB}|$, depending whether the target leaves the previous segment via A or via B. (The adjustment of x^L may be repeated if more than one road segment is passed, which is the case that the road segments in the database are of small length or high resolution.) For sake of simplicity, y^L is assumed to remain unchanged during the change of segment.

Under the approximation that the process noise is piecewise constant over a sampling interval T, the "equivalent" discrete-time dynamics model is obtained, given by

$$\mathbf{x}_k^L = \Phi^L \mathbf{x}_{k-1}^L + G^L \mathbf{w}_{k-1}^L \tag{3}$$

with

$$\Phi^L = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & \phi_{22} & 0 & \phi_{24} \\ 0 & 0 & 1 & 0 \\ 0 & \phi_{42} & 0 & \phi_{44} \end{bmatrix}, G^L = \begin{bmatrix} T^2/2 & 0 \\ 0 & g_{22} \\ T & 0 \\ 0 & g_{42} \end{bmatrix} \tag{4}$$

$\mathbf{x}_k^L = \left[(\mathbf{p}_k^L)^T \quad (\mathbf{v}_k^L)^T \right]^T = \left[x_k^L \quad y_k^L \quad \dot{x}_k^L \quad \dot{y}_k^L \right]^T$, and $\mathbf{w}_k^L = \left[w_{xk}^L \quad w_{yk}^L \right]^T$. The constants ϕ_{22}, ϕ_{24}, ϕ_{42}, ϕ_{44}, g_{22}, and g_{42} are functions of ξ and ω_n. When $\xi = \omega_n = 0$, that is, the same model is used for the motions of the two orthogonal directions, the matrices become

$$\Phi^L = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, G^L = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix} \tag{5}$$

The knowledge about the position and velocity of the target with respect to a predefined global coordinate system is needed as well. To convert the local representations $\mathbf{x}_k^L$ to the global representations $\mathbf{x}_k$, defined by $\mathbf{x}_k = [\mathbf{p}_k^T, \mathbf{v}_k^T]^T$, the following relations are used:

$$\mathbf{p}_k - \mathbf{p}_k^O = C(\psi)\mathbf{p}_k^L \text{ and } \mathbf{v}_k = C(\psi)\mathbf{v}_k^L \tag{6}$$

where $\mathbf{p}_k^O$ is the known global coordinates of the reference endpoint of the chosen road segment and $C(\psi)$ is a two dimensional rotation matrix defined by

$$C(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \tag{7}$$

with ψ being the angle from the x-axis of the global coordinate system to the road direction. The resulting global dynamics model is given by

$$\mathbf{x}_k = \Phi \mathbf{x}_{k-1} + (I_{4\times4} - \Phi)\mathbf{x}_k^O + G\mathbf{w}_{k-1}^L \tag{8}$$

where

$$\Phi = \begin{bmatrix} C(\psi) & 0_{2\times2} \\ 0_{2\times2} & C(\psi) \end{bmatrix} \Phi^L \begin{bmatrix} C^T(\psi) & 0_{2\times2} \\ 0_{2\times2} & C^T(\psi) \end{bmatrix} \tag{9}$$

$$G = \begin{bmatrix} C(\psi) & 0_{2\times2} \\ 0_{2\times2} & C(\psi) \end{bmatrix} G^L, \mathbf{x}_k^O = \begin{bmatrix} \mathbf{p}_k^O \\ \mathbf{0}_{2\times1} \end{bmatrix} \tag{10}$$

B. Observation Models

The observations are assumed to be obtained from a GMTI sensor with known positions $\mathbf{p}_k^S$. More importantly, the data association problem is assumed to have been solved correctly so that only the valid tracks are processed. The observation model of a detected target is given by[1]

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}'_k) + \boldsymbol{\nu}_k \tag{11}$$

$$\mathbf{h}(\mathbf{x}'_k) = \begin{bmatrix} \rho_k \\ \theta_k \\ \dot{\rho}_k \end{bmatrix} = \begin{bmatrix} \sqrt{(x'_k)^2 + (y'_k)^2} \\ \text{ATAN2}\,(y'_k, x'_k) \\ \frac{x'_k \dot{x}'_k + y'_k \dot{y}'_k}{\sqrt{(x'_k)^2 + (y'_k)^2}} \end{bmatrix} \tag{12}$$

where $\mathbf{x}'_k = \mathbf{x}_k - \left[(\mathbf{p}_k^S)^T \quad \mathbf{0}_{1\times2} \right]^T$ is the relative position from the sensor to the target, and ρ_k, θ_k, and $\dot{\rho}_k$ denote range, azimuth, and Doppler, respectively. The function ATAN2 is the four-quadrant arc tangent function. For sake of simplicity, the observation noise is assumed to be Gaussian and white: $\boldsymbol{\nu}_k \sim \mathcal{N}(\boldsymbol{\nu}_k; \mathbf{0}_{3\times1}, R)$. For the above model, the likelihood function $p(\mathbf{y}_k|\mathbf{x}_k) = p(\mathbf{y}_k|\mathbf{x}'_k) = \mathcal{N}(\mathbf{y}_k; \mathbf{h}(\mathbf{x}'_k), R)$. More complex likelihood function can be calculated based on this.

The event of no detection can be regarded as a general observation in the sense that the mode of the target can be inferred in a probabilistic manner even if the target is not detected. The likelihood functions associated with the detection event are summarized as follows

$$\begin{aligned} &\Pr(\text{detected}|\text{stopped}) = 0 \\ &\Pr(\text{detected}|\text{moving normally}) = P_D \\ &\Pr(\text{undetected}|\text{moving normally}) = 1 - P_D \\ &\Pr(\text{undetected}|\text{stopped}) = 1 \\ &\Pr(\text{undetected}|\text{moving perpendicular to LOS}) = 1 \\ &\Pr(\text{undetected}|\text{hidden from the sensor}) = 1 \end{aligned} \tag{13}$$

where LOS stands for "line of sight."

Of most interest is to infer from "no detections" whether the target is stopped. Under our assumptions the target cannot be stopped if detected by the sensor. When lack of detection is observed for a number n of consecutive sampling intervals, it is very unlikely that the target is in "normal" motion because $(1 - P_D)^n$, the probability of a "normally' moving target not being detected for n consecutive intervals, is very small (assuming P_D is close to one). In other words, the lack of detection is unlikely due to $P_D < 1$. If the "singular" case that the target is moving perpendicular to the line of sight from the target to the sensor or the target is hidden from the sensor, for example, it enters a tunnel, can be successfully excluded based on target state estimates, we will be able to assert that the target is very likely to be stopped. The first singular case can be easily eliminated if more than one GMTI sensor or other sensors are available. In practice, the probability of the occurrence of this case is not high because of the motion of the GMTI sensor. When the target enters a long tunnel, however, there is no way to predict whether it will be stopped or keep moving on. (In the worst case, the track will be lost.) Since very limited information about the target motion is contained in the events of no detection, when the “singular” cases cannot be eliminated and thus additional hypotheses have to be maintained, the ambiguity about the target mode dramatically increases and the inference becomes much more inaccurate.

III. Particle Filters For Jump Markov Systems

A general jump Markov system can be described by the following state-space model:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k): \quad \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, r_k, \mathbf{w}_{k-1}) \tag{14}$$

$$p(\mathbf{y}_k|\mathbf{x}_k, r_k): \quad \mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, r_k, \boldsymbol{\nu}_k) \tag{15}$$

$$r_k \sim p(r_k|r_{k-1}) \tag{16}$$

where the hybrid state vector consists of the base state $\mathbf{x}_k$ and the mode state r_k. The base state vector $\mathbf{x}_k$ and the observation vector $\mathbf{y}_k$ are continuous-valued; the mode state r_k is discrete-valued and the discrete values are denoted by $\{1, 2, \ldots, \mathcal{S}\}$. The dynamics function $\mathbf{f}(\cdot)$ and the observation function $\mathbf{h}(\cdot)$ may be nonlinear. The process noise $\{\mathbf{w}_k\}$ and the observation noise $\{\boldsymbol{\nu}_k\}$ are assumed to be white noise. When $\mathbf{f}$ and $\mathbf{h}$ are linear and $\mathbf{w}_k(r_k) \sim \mathcal{N}(\mathbf{w}_k(r_k); \mathbf{0}, Q(r_k))$ and $\boldsymbol{\nu}_k(r_k) \sim \mathcal{N}(\boldsymbol{\nu}_k(r_k); \mathbf{0}, R(r_k))$ as well as the initial base state $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \hat{\mathbf{x}}_0, P_0)$ are Gaussian noise, the general jump Markov system reduces to the jump Markov linear Gaussian system, given by

$$\mathbf{x}_k = \Phi(r_k)\mathbf{x}_{k-1} + G(r_k)\mathbf{w}_{k-1}(r_k) \tag{17}$$

$$\mathbf{y}_k = H(r_k)\mathbf{x}_k + \boldsymbol{\nu}_k(r_k) \tag{18}$$

$$r_k \sim p(r_k|r_{k-1}) \tag{19}$$

The Markov transition model $r_k \sim p(r_k|r_{k-1})$ may be generalized to $r_k \sim p(r_k|\mathbf{R}_{k-1}, \mathbf{Y}_{k-1})$, where $\mathbf{R}_k \equiv \{r_1, ..., r_k\}$ and $\mathbf{Y}_k \equiv \{\mathbf{y}_1, ..., \mathbf{y}_k\}$.

The objective of particle filtering for jump Markov systems is to estimate recursively the joint posterior distribution $p(\mathbf{X}_k, \mathbf{R}_k|\mathbf{Y}_k)$ or $p(\mathbf{x}_k, r_k|\mathbf{Y}_k)$ with $\mathbf{X}_k \equiv \{\mathbf{x}_1, ..., \mathbf{x}_k\}$. Of practical interest is the marginal distribution $p(\mathbf{x}_k|\mathbf{Y}_k)$, obtained from $p(\mathbf{x}_k, r_k|\mathbf{Y}_k)$ by standard marginalization. With weighted particles $\{r_k^{(i)}, \mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$, they are approximated as[9]

$$p(\mathbf{x}_k, r_k|\mathbf{Y}_k) \approx \sum_{i=1}^{N} w_k^{(i)} \delta_{\mathbf{x}_k^{(i)}, r_k^{(i)}}(d\mathbf{x}_k, r_k) \tag{20}$$

and

$$p(\mathbf{x}_k|\mathbf{Y}_k) \approx \sum_{i=1}^{N} w_k^{(i)} \delta_{\mathbf{x}_k^{(i)}}(d\mathbf{x}_k) \tag{21}$$

where δ is the point-mass delta function.

The particles and their associated weights are recursively updated by 1) sampling $r_k^{(i)}$ and $\mathbf{x}_k^{(i)}$ from a certain importance function $q(\mathbf{x}_k, r_k|\mathbf{X}_{k-1}^{(i)}, \mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_k)$ and 2) updating the importance weights according to

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(i)}, r_k^{(i)}) p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, r_k^{(i)}) p(r_k^{(i)}|r_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)}, r_k^{(i)}|\mathbf{X}_{k-1}^{(i)}, \mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_k)} \tag{22}$$

Resampling step that eliminates particles with low importance weights and multiplies particles with high importance weights should be added in the filter cycle in order to suppress the degeneracy problem[1].

The bootstrap filter for jump Markov systems corresponds to the special choice of the importance function as

$$q(\mathbf{x}_k, r_k|\mathbf{X}_{k-1}^{(i)}, \mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)}, r_k) p(r_k|r_{k-1}^{(i)}) \tag{23}$$

The importance weights are then updated using

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(\mathbf{y}_k|\mathbf{x}_k^{(i)}, r_k^{(i)}) \tag{24}$$

The bootstrap filter makes few assumptions about the state-space model and employs little of the structure of jump Markov systems. The sampling scheme as shown above is simple to implement but can be inefficient. The efficient particle filter for general jump Markov systems presented in Ref. [10] greatly improves the efficiency of sampling $r_k^{(i)}$ and $\mathbf{x}_k^{(i)}$ by using an importance function that makes better use of the Markov structure. That is, because r_k can only take on a finite number of values and therefore the mode transition from $r_{k-1}^{(i)}$ to r_k can only have a finite number of possibilities, the marginalization over the mode subspace as required by a better importance function can be implemented with summation.

When the system is a jump Markov linear Gaussian one, the benign structure makes it possible to design more efficient particle filter based on Rao-Blackwellization. The idea of Rao-Blackwellized particle filtering is to reduce the sampling space as much as possible by analytic marginalization. For jump Markov linear Gaussian systems, the technique is based on a partition of the joint distribution, given by[9]:

$$p(\mathbf{X}_k, \mathbf{R}_k|\mathbf{Y}_k) = p(\mathbf{X}_k|\mathbf{R}_k, \mathbf{Y}_k) p(\mathbf{R}_k|\mathbf{Y}_k) \tag{25}$$

Because $p(\mathbf{X}_k|\mathbf{R}_k, \mathbf{Y}_k)$ is exactly Gaussian in this case, it can be sufficiently represented by its mean and covariance. (This statement does not hold any more when r_k is dependent on $\mathbf{x}_{k-1}$.) Given $\mathbf{R}_k$ and $\mathbf{Y}_k$, the conditional mean and covariance can be determined in closed form using Kalman filtering. Thus, only the mode distribution $p(\mathbf{R}_k|\mathbf{Y}_k)$ needs to be estimated using particle filtering. The marginal distribution $p(\mathbf{R}_k|\mathbf{Y}_k)$ satisfies a recursion[9]:

$$p(\mathbf{R}_k|\mathbf{Y}_k) = p(\mathbf{R}_{k-1}|\mathbf{Y}_{k-1}) \frac{p(\mathbf{y}_k|\mathbf{Y}_{k-1}, \mathbf{R}_k) p(r_k|r_{k-1})}{p(\mathbf{y}_k|\mathbf{Y}_{k-1})} \tag{26}$$

where $p(\mathbf{y}_k|\mathbf{Y}_{k-1}, \mathbf{R}_k)$ can be regarded as the “likelihood” of a mode state sequence and can be analytically determined. Based on the recursion, a particle filter scheme for $p(r_k|\mathbf{Y}_k)$ can be designed, in which the filtering distribution $p(r_k|\mathbf{Y}_k)$ is approximated with the weighted particles $\{r_k^{(i)}, w_k^{(i)}\}_{i=1}^N$ as

$$p(r_k|\mathbf{Y}_k) \approx \sum_{i=1}^{N} w_k^{(i)} \delta_{r_k^{(i)}}(r_k) \tag{27}$$

and the filtering distribution $p(\mathbf{x}_k|\mathbf{Y}_k)$ is then approximated as a Gaussian mixture

$$
\begin{aligned}
p(\mathbf{x}_k|\mathbf{Y}_k) &\approx \sum_{i=1}^{N} w_k^{(i)} p(\mathbf{x}_k|\mathbf{R}_k^{(i)}, \mathbf{Y}_k) \\
&= \sum_{i=1}^{N} w_k^{(i)} \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^{(i)}, P_k^{(i)})
\end{aligned}
\tag{28}
$$

where $p(\mathbf{x}_k|\mathbf{R}_k^{(i)}, \mathbf{Y}_k)$, the posterior distribution of $\mathbf{x}_k$ for an individual mode sequence $\mathbf{R}_k^{(i)}$, is exactly Gaussian and the associated mean $\hat{\mathbf{x}}_k^{(i)}$ and covariance $P_k^{(i)}$ are determined using Kalman filtering. Since the distribution $p(\mathbf{x}_k|\mathbf{R}_k^{(i)}, \mathbf{Y}_k)$ is exactly determined (equivalent to using an infinite number of samples of $\mathbf{x}_k$), the approximation of the filter only comes from the particle representation of $p(r_k|\mathbf{Y}_k)$.

For jump Markov linear Gaussian systems, samples of r_k can be drawn from the optimal importance function, given by

$$
p(r_k|\mathbf{R}_{k-1}, \mathbf{Y}_k) = \frac{p(\mathbf{y}_k|r_k, \mathbf{R}_{k-1}, \mathbf{Y}_{k-1}) p(r_k|r_{k-1})}{p(\mathbf{y}_k|\mathbf{R}_{k-1}, \mathbf{Y}_{k-1})} \tag{29}
$$

Then the associated normalized importance weights are

$$
\begin{aligned}
w_k^{(i)} &\propto w_{k-1}^{(i)} \frac{p(\mathbf{y}_k|r_k^{(i)}, \mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_{k-1}) p(r_k^{(i)}|r_{k-1}^{(i)})}{q(r_k^{(i)}|\mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_k)} \\
&= w_{k-1}^{(i)} p(\mathbf{y}_k|\mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_{k-1})
\end{aligned}
\tag{30}
$$

where

$$
p(\mathbf{y}_k|\mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_{k-1}) = \sum_{j=1}^{\mathcal{S}} p(\mathbf{y}_k|j, \mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_{k-1}) p(j|r_{k-1}^{(i)}) \tag{31}
$$

In the summation,

$$
\begin{aligned}
& p(\mathbf{y}_k|j, \mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_{k-1}) \\
&= \mathcal{N}\left(\mathbf{y}_k; H(j)\hat{\mathbf{x}}_k^{-(i,j)}, H(j)P_k^{-(i,j)}H^T(j) + R(j)\right)
\end{aligned}
\tag{32}
$$

with $\hat{\mathbf{x}}_k^{-(i,j)}$ and $P_k^{-(i,j)}$ being the mean and covariance of the prediction $p(\mathbf{x}_k|j, \mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_{k-1})$, respectively. Because the importance weights $p(\mathbf{y}_k|\mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_{k-1})$ do not depend on r_k, it is possible to select the fittest particle trajectories $\mathbf{R}_{k-1}^{(i)}$ based on $\mathbf{y}_k$ before samples of $r_k^{(i)}$ are drawn.

Because random samples are used to represent $p(r_k|\mathbf{Y}_k)$ while exact means and covariances are used to represent $p(\mathbf{x}_k|\mathbf{R}_k, \mathbf{Y}_k)$, the full representation for the distribution of the hybrid state is composed of $\{r_k^{(i)}, \hat{\mathbf{x}}_k^{(i)}, P_k^{(i)}, w_k^{(i)}\}_{i=1}^N$, where $\hat{\mathbf{x}}_k^{(i)}$ and $P_k^{(i)}$ are deterministically updated given $r_k^{(i)}$.

IV. Efficient Particle Filter for Road-Constrained Target Tracking

In this section, two variable-structure multiple-model particle filters for road-constrained target tracking are presented. The bootstrap filter serves as the baseline algorithm and the proposed efficient particle filter is compared with it. Both filters are designed based on the same assumptions and models.

A. Multiple Models for Road-Constrained Target

Multiple dynamics models are used to account for the motion uncertainty due to target maneuver. We assume the motion mode state $r = 1$ corresponds to the cruise mode, $r = 2$ corresponds to the maneuver mode, and $r = 3$ corresponds to the stopped mode. According to the idea of variable-structure multiple-model approach, the cruise and the maneuver modes are active all the time, whereas the stopped mode is active only when there is no detection. The stopped mode is added to the active mode set when the target is no longer detected and removed after the target is detected again.

The target dynamics models for different target modes have the same linear Gaussian structure, given by Eqs. (3) and (8). For the maneuver model, large process noise along the road is used, which is of the order of the magnitude of the maximum acceleration. Much smaller process noise is used in the cruise model. In the stopped model, the process noise is set to zero. The target velocity in the stopped target model is also set to zero. The road information in terms of the reference point $\mathbf{p}_k^O$ and the direction ψ of the road segment is incorporated in the dynamics models through $\mathbf{x}_k^O$, Φ and G.

The transition of the motion mode r is assumed to occur only at sensor sampling instants and is governed by the transition probability matrix P, whose elements are defined by

$$
\mathrm{p}_{ij} = p(r_k = j|r_{k-1} = i) \tag{33}
$$

where p_{ij} satisfy $\sum_{j=1}^{\mathcal{S}} \mathrm{p}_{ij} = 1$, with $\mathcal{S}$ is the number of active modes (also the number of columns of the matrix P). For sake of simplicity, constant P is used. The active mode set may be $\{1, 2\}$ or $\{1, 2, 3\}$. Hence, four transition matrices in total are needed. The initial guess of the transition matrices are calculated based on the sojourn time of the modes[4].

Because the knowledge of the present road segment is required for the propagation of the dynamics models, a pointer p_k pointing to the road segment the target is on at time t_k is used as an auxiliary mode state. All the information about the present road segment, such as its endpoints, directions, and neighbors, is indexed in the road database via the pointer p_k. The update of p_k is determined by the propagated target position. The sequence p_k is not a Markov chain because of its target state dependence. If after propagation the target remains on the present road segment, p_k does not change its value; if the target leaves the present road segment to enter a new segment, p_k points to the new segment, too. When there is only one new segment to enter, the pointer is updated without ambiguity. However at an intersection where more than one road meets, it is uncertain which road segment the target would enter. Then all the hypotheses have to be considered and thus p_k and $\mathbf{x}_k$ in the particle filter have to be updated in a probabilistic manner. The ambiguity can only be eliminated after new observations are available. If no prior knowledge about the route or destination of the target is available, then it is reasonable to assign identical probability to each hypothesis. Suppose the number of roads to enter is L, the probability to enter any road segment is $1/L$.

A single observation model of a detected target is used, as given by Eq. (11). In other words, $p(\mathbf{y}_k|\mathbf{x}_k, r_k = 1) = p(\mathbf{y}_k|\mathbf{x}_k, r_k = 2)$. When the target is detected, the likelihood of a moving mode can be computed using $\mathbf{y}_k$ and the system model; the likelihood of the stopped mode is zero. (The probability of detection P_D may be incorporated in the likelihood of the moving mode, but it is not necessary since P_D is a common factor among the stopped and moving modes.) When the target is not detected, the likelihood functions of different modes are determined using Eq. (13).

B. Bootstrap Filter

The joint distribution of $\mathbf{x}_k$ and r_k is approximated by weighted particles $\{r_k^{(i)}, \mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$. The pointers $p_k^{(i)}$ to the present road segment are also assigned to particles. Thus, the full particle representation is given by $\{r_k^{(i)}, \mathbf{x}_k^{(i)}, p_k^{(i)}, w_k^{(i)}\}_{i=1}^N$. The outline of a filter cycle of the bootstrap filter for road-constrained target tracking is given in Table I.

- For $i = 1, ..., N$,
  - determine the active motion mode set for i^{th} particle at t_k
  - determine the transition probability matrix $\mathrm{P}_k^{(i)}$
  - sample
$$r_k^{(i)} \sim p(r_k|r_{k-1}^{(i)})$$
  - propagate $\mathbf{x}_{k-1}^{(i)}$ through the model specified by mode $r_k^{(i)}$ and road segment $p_{k-1}^{(i)}$ to generate $\mathbf{x}_k^{*(i,l)}$, where $l = 1, \ldots, L^{(i)}$, with $L^{(i)}$ the number of feasible road segments at time t_k. (if the target does not cross an intersection, $L^{(i)} = 1$; if the target crosses an intersection, $L^{(i)} > 1$.)
  - if $L^{(i)} > 1$, draw $\mathbf{x}_k^{(i)}$ from $\mathbf{x}_k^{*(i,l)}$ randomly (the probability of $\mathbf{x}_k^{*(i,l)}$ is $1/L^{(i)}$).
  - determine $p_k^{(i)}$ according to $\mathbf{x}_k^{(i)}$.
- For $i = 1, \ldots, N$,
  - evaluate the likelihood $\Lambda_k^{(i)}$:
    * if the target is not detected
      · if $r_k^{(i)} = 3$, $\Lambda_k^{(i)} = 1$
      · if $r_k^{(i)} = 1$ or 2 AND the target is hidden or moves perpendicular to the line of sight, $\Lambda_k^{(i)} = 1$
      · if $r_k^{(i)} = 1$ or 2 AND the target is in normal motion, $\Lambda_k^{(i)} = 1 - P_D$
    * if the target is detected
      · if $r_k^{(i)} = 3$ OR the the target is hidden or moves perpendicular to the line of sight, $\Lambda_k^{(i)} = 0$
      · if $r_k^{(i)} = 1$ or 2, $\Lambda_k^{(i)} = p(\mathbf{y}_k|\mathbf{x}_k^{(i)}, r_k^{(i)})$
  - evaluate and normalize the importance weights
$$w_k^{(i)} \propto w_{k-1}^{(i)} \Lambda_k^{(i)}$$
- Reampling: Multiple/Discard particles $\{r_k^{(i)}, \mathbf{x}_k^{(i)}, p_k^{(i)}, w_k^{(i)}\}_{i=1}^N$ with respect to high/low importance weights $w_k^{(i)}$ to obtain N new particles $\{r_k^{(i)}, \mathbf{x}_k^{(i)}, p_k^{(i)}, w_k^{(i)}\}_{i=1}^N$ with equal weights.

TABLE I

BOOTSTRAP FILTER FOR ROAD-CONSTRAINED TARGET TRACKING

C. Efficient Particle Filter

The efficient particle filter for road-constrained target tracking is designed based on the optimal particle filtering theory for jump Markov linear Gaussian systems. Analytic approximation is made for the target state distribution mainly because although the target dynamics is modeled as linear system driven by Gaussian white noise, the observation model is nonlinear. Hence, the conditional distribution $p(\mathbf{x}_k|\mathbf{R}_k^{(i)}, \mathbf{Y}_k)$ is not strictly Gaussian. It is, however, still approximated by a Gaussian distribution $\mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_k^{(i)}, P_k^{(i)}\right)$ whose mean $\hat{\mathbf{x}}_k^{(i)}$ and covariance $P_k^{(i)}$ are an approximate sufficient statistic and are updated using unscented Kalman filtering. The details of the unscented Kalman filter can be found in Ref. [12]. For nonlinear filtering problems, when the parameters of the unscented transformation are appropriated tuned, the unscented Kalman filter can yield better estimation results than the extended Kalman filter. The likelihood $p(\mathbf{y}_k|r_k, \mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_{k-1})$ used for recursive sampling of r_k is also calculated based on Gaussian approximation. That is,

$$\begin{aligned} p(\mathbf{y}_k|r_k, \mathbf{R}_{k-1}^{(i)}, \mathbf{Y}_{k-1}) &\approx p(\mathbf{y}_k|r_k, \hat{\mathbf{x}}_{k-1}^{(i)}, P_{k-1}^{(i)}) \\ &\approx \mathcal{N}\left(\mathbf{y}_k; \hat{\mathbf{y}}_k^{(i)}, P_{yk}^{(i)}\right) \end{aligned} \quad (34)$$

Given r_k, $\hat{\mathbf{x}}_{k-1}^{(i)}$, and $P_{k-1}^{(i)}$, the mean $\hat{\mathbf{y}}_k^{(i)}$ and covariance $P_{yk}^{(i)}$ of $\mathbf{y}_k$ are estimated using standard unscented transformation.

The full particle representation is given by $\{r_k^{(i)}, \hat{\mathbf{x}}_k^{(i)}, P_k^{(i)}, p_k^{(i)}, w_k^{(i)}\}_{i=1}^N$, where $\hat{\mathbf{x}}_k^{(i)}$ and $P_k^{(i)}$ are deterministically updated given $r_k^{(i)}$ and $p_k^{(i)}$. The outline of a filter cycle of the efficient particle filter for road-constrained target tracking is given in Table II.

V. SIMULATION RESULTS

The main objective of this section is to demonstrate the feasibility and efficiency of the proposed efficient particle filter. The main simulation parameters are given as follows: the sampling interval is 5 seconds; the standard deviations of the range, azimuth, and Doppler are 20 m, 0.01 rad/s, and 1 m/s, respectively; the probability of detection is $P_D = 0.85$; the minimum detectable velocity is 2 m/s. For sake of simplicity, the GMTI sensor is assumed to be stationary at [-1000 m, 1000 m], denoted by the circle in Fig. 1. A road network used for simulation is illustrated in Fig. 1. The route of the target is from segment $\overrightarrow{AB}$ to segment $\overrightarrow{DE}$ via $\overrightarrow{BC}$ and $\overrightarrow{CD}$. The points B, C, and D are intersections. The modes of target motion alternate between acceleration, cruise, deceleration, and stop. Only one target is considered because under the assumption that the correct data association is known, the tracking of multiple targets simply reduces to the separate tracking of individual targets. The equivalent target positions in Cartesian coordinates directly converted from the range ρ_k and azimuth θ_k observations of a typical run are plotted in Fig. 1 as the dots in the neighborhood of the road. At positions where the velocity of the target is smaller than the minimum detectable velocity, the target is not detected for deterministic reasons. At other points, the target may still be undetected

- For $i = 1, \ldots, N$,
  - determine the active motion mode set for i^{th} particle at t_k
  - determine the transition probability matrix $\mathrm{P}_k^{(i)}$
  - For $j = 1, \ldots, \mathcal{S}$, where $\mathcal{S}$ is the number of active motion modes (hypotheses),
    - $*$ propagate $\hat{\mathbf{x}}_{k-1}^{(i)}$ and $P_{k-1}^{(i)}$ through the model specified by mode j and road segment $p_{k-1}^{(i)}$ to generate $\hat{\mathbf{x}}_k^{*(i,j,l)}$ and $P_k^{*(i,j,l)}$, where $l = 1, \ldots, L^{(i,j)}$(if the target does not cross an intersection, $L^{(i,j)} = 1$; if the target crosses an intersection, $L^{(i,j)} > 1$); determine $p_k^{*(i,j,l)}$ according to $\hat{\mathbf{x}}_k^{*(i,j,l)}$ and $P_k^{*(i,j,l)}$
  - For $j = 1, \ldots, \mathcal{S}$ and for $l = 1, \ldots, L^{(i,j)}$, evaluate the likelihood $\Lambda_k^{(i,j,l)}$:
    - $*$ if the target is not detected
      - · if $j = 3$, $\Lambda_k^{(i,j,l)} = 1$
      - · if $j = 1$ or 2 AND the target is hidden or moves perpendicular to the line of sight, $\Lambda_k^{(i,j,l)} = 1$
      - · if $j = 1$ or 2 AND the target is in normal motion, $\Lambda_k^{(i,j,l)} = 1 - P_D$
    - $*$ if the target is detected
      - · if $j = 3$ OR the target is hidden or moves perpendicular to the line of sight, $\Lambda_k^{(i,j,l)} = 0$
      - · if $j = 1$ or 2, compute $\Lambda_k^{(i,j,l)} = \hat{p}(\mathbf{y}_k | \hat{\mathbf{x}}_k^{*(i,j,l)}, P_k^{*(i,j,l)})$ based on Gaussian approximation
- For $i = 1, \ldots, N$, compute

$$w_k^{(i)} \propto w_{k-1}^{(i)} \sum_{j=1}^{\mathcal{S}} \sum_{l=1}^{L^{(i,j)}} \Lambda_k^{(i,j,l)} p(j|r_{k-1}^{(i)})/L^{(i,j)}$$

- Resampling: Multiple/Discard $\{r_{k-1}^{(i)}, \hat{\mathbf{x}}_k^{*(i,j,l)}, P_k^{*(i,j,l)}, p_k^{*(i,j,l)}, \Lambda_k^{(i,j,l)}\}_{i=1}^N$ with respect to high/low importance weights $w_k^{(i)}$ to obtain N new $\{r_{k-1}^{(i)}, \hat{\mathbf{x}}_k^{*(i,j,l)}, P_k^{*(i,j,l)}, p_k^{*(i,j,l)}, \Lambda_k^{(i,j,l)}\}_{i=1}^N$ with equal weights.
- For $i = 1, \ldots, N$, sample

$$(r_k^{(i)}, l^{(i)}) \sim \hat{p}(\mathbf{y}_k | \hat{\mathbf{x}}_k^{*(i,r_k,l)}, P_k^{*(i,r_k,l)}) p(r_k | r_{k-1}^{(i)})/L^{(i,j)}$$

where $\hat{p}(\mathbf{y}_k | \hat{\mathbf{x}}_k^{*(i,j,l)}, P_k^{*(i,j,l)}) = \Lambda_k^{(i,j,l)}$; set

$$\hat{\mathbf{x}}_k^{-(i)} = \hat{\mathbf{x}}_k^{*(i,r_k^{(i)},l^{(i)})},\ P_k^{-(i)} = P_k^{*(i,r_k^{(i)},l^{(i)})},\ p_k^{(i)} = p_k^{*(i,r_k^{(i)},l^{(i)})}$$

- For $i = 1, \ldots, N$, update $\hat{\mathbf{x}}_k^{(i)}$, $P_k^{(i)}$ from $\hat{\mathbf{x}}_k^{-(i)}$, $P_k^{-(i)}$ based on Gaussian approximation, and update $p_k^{(i)}$ according to $\hat{\mathbf{x}}_k^{(i)}$ and $P_k^{*(i,j,l)}$

TABLE II

EFFICIENT PARTICLE FILTER FOR ROAD-CONSTRAINED TARGET TRACKING

due to $P_D < 1$. The results of a typical run of the efficient particle filters with 50 particles are presented in Figs. 2 and 3. The RMS error of the estimated position over this run is about 20 m. The RMS velocity error is about 2.4 m/s. The filter shows good tracking ability. The position errors (relative to the true positions) of the position observations and the position estimates given by the particle filter are also compared in Fig. 2. It can be seen that in the position observations the errors perpendicular to the road are quite large, with the peak value 180 m. For this reason, the position estimates in the off-road direction are tightly constrained from tracking the off-road observations too closely. In our particle filters, such constraint is imposed using dynamics models that are strictly stable in the direction perpendicular to the road, for example, with $\xi = 1$ and $\omega_n = 0.2$, so that the particle cloud is restricted well within the road. The spikes in the position and velocity errors are unlikely to be avoided in all the runs. Here are some typical cases in which the spikes will occur.

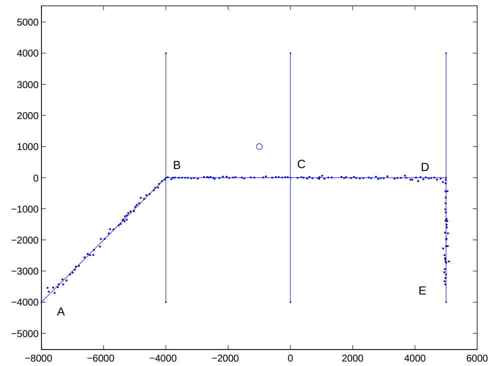

Fig. 1. Road Network

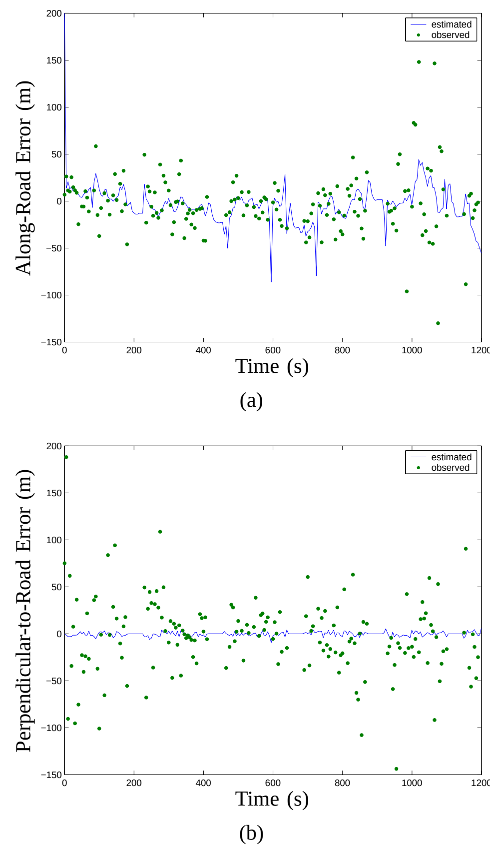

Fig. 2. Time History of Position Errors

- The filters are usually tuned so that a target stop can be quickly detected from lack of detection. However, if during a number of (e.g., 3) consecutive sampling intervals the target is accelerating from the stopped mode but no observations are available due to $P_D < 1$, the

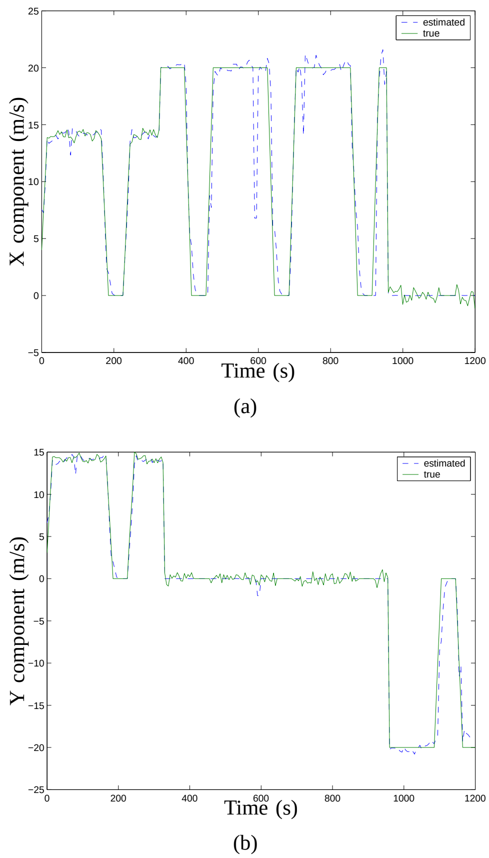

Fig. 3. Time History of True and Estimated Velocities

tracker tends not to response and the delayed response gives rise to spikes.

- If at a sampling point the target is crossing the intersection but no observations are available or the observations are close to the intersection, the tracker cannot effectively eliminate the road ambiguity around the intersection until more post-crossing observation data are processed. But before that it has to maintain all the hypotheses of the road segments around the intersection, which again leads to spikes in estimation errors. The spike around time $t = 590s$ corresponds to this case.
- Observation outliers and the singular cases can also lead to imprecise judgment on target maneuvers.

Now the performance of the efficient particle filter with 50 particles is compared with that of the bootstrap filter with 1000 particles. The estimation accuracy is measured by RMS errors over 30 runs. The Matlab function CPUTIME that "returns the CPU time in seconds that has been used by the MATLAB process since MATLAB started" is used to roughly measure the computational expense of the algorithms. The RMS errors of the efficient particle filter with 50 particles are 17.5m and 2.1m/s. The RMS errors of the bootstrap filter with 1000 particles are 20.0m and 2.2 m/s. The average CPUTIME for the efficient particle filter is about 42 seconds and that for the bootstrap filter is about 150 seconds. Although the bootstrap filter has less computational expense per particle than the efficient particle filter, the CPUTIME taken by the bootstrap filter with 1000 particles turns out to be about four times longer than that taken by the efficient particle filter with 50 particles. From the above comparisons, a modest conclusion can be drawn that the efficient particle filter with 50 particles can achieve performance similar to that of the bootstrap filter with 1000 particles with much less computational expense.

VI. Conclusions

For road-constrained targets, the incorporation of road information into the dynamics models can greatly reduce the target motion uncertainty. A variable-structure, multiple-model framework is used to address target maneuvers along the road. The proposed efficient particle filter is an approximation to the optimal particle filter for jump Markov linear Gaussian systems. The main approximation of the filter is the Gaussian assumption about the conditional target state distribution given a mode sequence and observations. The efficient particle filter with 50 particles yields satisfactory simulation results. Compared with the standard bootstrap filter, the proposed efficient particle filter involves much less computation for similar accuracy and robustness.

References

[1] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA: Artech House Publishers, 2004.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. New York, NY: John Wiley and Sons, Inc., 2001.

[3] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, and I. Kadar, "Ground target tracking with variable structure imm estimator," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 1, pp. 26 – 46, January 2000.

[4] T. Kirubarajan and Y. Bar-Shalom, "Tracking evasive move-stop-move targets with a gmti radar using a vs-imm estimator," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1452 – 1457, October 2003.

[5] C. Kreucher and K. Kastella, "Multiple model nonlinear filtering for low singal ground target applications," in *Proceedings of SPIE Conference on Aerosense, Signal Processing, Sensor Fusion and Target Recognition X*, vol. 4380, April 2001, pp. 256–266.

[6] L. Lin, Y. Bar-Shalom, and T. Kirubarajan, "New assignment-based data association for tracking move-stop-move targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 714 – 725, April 2004.

[7] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer-Verlag, 2001.

[8] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump markov linear systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 613 – 624, March 2001.

[9] N. de Freitas, R. Dearden, F. Hutter, R. Morales-Menéndez, J. Mutch, and D. Poole, "Diagnosis by a waiter and a mars explorer," in *Proceedings of the IEEE*, vol. 92, no. 3, March 2004, pp. 455 – 468.

[10] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump markov systems. application to time-varying autoregressions," *IEEE Transactions on Signal Processing*, vol. 51, no. 7, pp. 1762 – 1770, July 2003.

[11] S. Särkkä, A. Vehtari, and J. Lampinen, "Rao-blackwellized monte carlo data association for multiple target tracking," in *Proceedings of FUSION 2004, The 7th International Conference on Information Fusion*, Stockholm, June 2004.

[12] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proceedings of the IEEE*, vol. 92, March 2004, pp. 401–422.