Particle filtering algorithm for tracking multiple road-constrained targets

Craig S. Agate* and Kevin J. Sullivan

Toyon Research Corporation, 75 Aero Camino Drive Suite A, Goleta, CA USA 93117-3139

ABSTRACT

We propose a particle filtering algorithm for tracking multiple ground targets in a road-constrained environment through the use of GMTI radar measurements. Particle filters approximate the probability density function (PDF) of a target's state by a set of discrete points in the state space. The particle filter implements the step of propagating the target dynamics by simulating them. Thus, the dynamic model is not limited to that of a linear model with Gaussian noise, and the state space is not limited to linear vector spaces. Indeed, the road network is a subset (not even a vector space) of $\mathbb{R}^2$. Constraining the target to lie on the road leads to adhoc approaches for the standard Kalman filter. However, since the particle filter simulates the dynamics, it is able to simply sample points in the road network. Furthermore, while the target dynamics are modeled with a parasitic acceleration, a non-Gaussian discrete random variable noise process is used to simulate the target going through an intersection and choosing the next segment in the road network on which to travel. The algorithm is implemented in the SLAMEM^{TM} simulation (an extensive simulation which models roads, terrain, sensors and vehicles using GVS^{TM}). Tracking results from the simulation are presented.

Keywords: particle filter, ground targets, constrained estimation, nonlinear filtering

1. INTRODUCTION

Traditional ground-target tracking algorithms use a set of detections obtained by a low-range-resolution radar to track targets in a region of interest. The tracking algorithms typically employ a Kalman filter or an EKF (Extended Kalman Filter) to facilitate state estimation. Modern radars are capable of providing HRR (high-range-resolution) radar data which, in addition to kinematic information, yield signature or feature information. The resulting signature is a function of both the target class (i.e., type of object) and the orientation of the target relative to the radar (i.e., the aspect from which the target is illuminated). Since the orientation is related to the kinematic state of the target, the information contained in the signature not only aids estimation of a target's class, but also aids estimation of the target's kinematic state. The coupling between a target's kinematics and signature has traditionally been ignored with the signature relegated to estimation of a target's ID (identification) and/or aiding the association of measurements to tracks (i.e., signature-aided tracking[1]). While target identification and SAT (signature-aided tracking) are crucial aspects of situational awareness, any target identification and tracking algorithm should strive for the maximum extraction of information from a target's signature. Thus, full utilization of the information in the feature/signature and kinematic measurements should enhance a joint tracking and ID algorithm in the following ways

- Orientation-dependent features aid the estimation of a target's kinematic state.
- Kinematic measurements aid estimation of class type through, for example, maneuverability and maximum velocity constraints
- Features help identify targets as belonging to a particular class or as a particular target previously in track.
- Features aid the data association problem by contributing to the likelihood that a detection originated from a target in track.

*cagate@toyon.com; phone (805)968-6787x176; fax (805)685-8089; www.toyon.com

- Target identification supports more accurate dynamic modeling of a target as opposed to using a generic dynamic model to represent a variety of target types.

While our *ultimate* goal is the development of a joint tracking and identification algorithm using GMTIHRR measurements based on particle filtering[2] methods, in this paper we discuss the first step in the development of the algorithm. We have developed a particle filtering algorithm which tracks ground targets in a road-constrained environment.

Particle filtering methods have, in recent years, been investigated as a means of handling the difficult nonlinear estimation problem. Since particle filters are essentially non-parametric density estimators, there is no constraint on the underlying structure of the probability density function (PDF). State estimation problems in which nonlinear dynamic models, nonlinear measurement models or non-Gaussian noise processes are present pose fundamental difficulties when applying standard Kalman filtering techniques. Alternatively, particle filtering techniques are well-suited to problems in which nonlinearities and non-Gaussian noise processes exist. Many researchers have sought out nonlinear problems, such as bearings-only tracking[3,4] and tracking multiple targets in clutter[5] to which to apply particle filtering methods. Recent work[6] investigates the application of particle filters for Jump Markov Linear Systems, which are linear systems whose parameters evolve according to a Markov process.

While much attention has focused on applying particle filters to problems with nonlinear dynamics or measurements, there are other problems in which particle filters exhibit great potential. For example, the joint estimation of continuous random variables and discrete random variables is extremely problematic with standard techniques. The typical approach is based on the assumption that the continuous components and discrete components of the state can be estimated separately, an assumption only valid if the measurements used by the estimator are separable into those that rely on the discrete component of the state and those that rely on the continuous component of the state. Feature measurements such as HRR profiles, however, are a function of both the continuous state of the target (position and velocity) and the discrete state of the target (target ID). Particle filters also elegantly handle problems in which the state space of the target is constrained to a subset of some vector space. Targets constrained by terrain or a road network are examples of state-constrained estimation problems. Due to their unique properties, particle filtering methods offer a new and exciting approach to these and other nonlinear estimation problems.

A particle filter approximates the PDF of an unknown random variable by a set of samples (often called a particle cloud) in the state space. Thus, its accuracy is a function of the number of particles propagated by the filter. While the number of particles which can be propagated is limited by computing resources, the rapid increase in computing power and the *embarrassingly parallelizable* property of the particle filtering algorithm, coupled with the following exciting attributes of the algorithm warrant further research and consideration

- **Not constrained by linear dynamics, linear measurements or Gaussian PDFs** Particle filtering algorithms do not require linearization of dynamic equations or measurement equations. Furthermore, PDFs for the measurement likelihoods are not required to be Gaussian.[2] Updating the target state PDF simply requires values of the measurement likelihood function evaluated at discrete points in the state space. The implications of this are that the algorithm effortlessly incorporates measurements from a variety of sensor types (from HRRGMTI type measurements which provide range, range rate and bearing as well as a signature, to ESM type sensor measurements which provide a bearing in addition to a signature used to characterize the target class). Again, no linearization of the measurement model and subsequent calculation of Jacobians is required.

- **Elegantly handles constrained estimation problems (e.g. road-constrained networks)** In a constrained estimation problem the possible values of an unknown random vector are constrained to a subset of some standard vector space (such as $\mathbb{R}^2$). There is significant information in the constraints and utilizing them should yield significantly better estimation results. Depending upon the geometry of the subset, however, incorporating the constraint into the problem formulation for a standard Kalman filtering algorithm can be difficult and lead to ad-hoc solutions. On the other hand, the underlying geometry of the state space, which may not even be a proper vector space, has little effect on the implementation of the particle filter; the only requirement being the ability to sample points from this space. An excellent example of this type of problem is the tracking of ground targets in a road-constrained network. Some trackers incorporate road-clamping into a Kalman filtering algorithm which, while improving the performance of the standard Kalman filter, is awkward and typically requires the use of ad-hoc steps.

- **Naturally handles a state vector comprising both discrete and continuous variables** Underlying a joint tracking and identification algorithm is a state vector which comprises a continuous part (the kinematic state of the target) and a discrete part (the class to which the target belongs). Since the filter propagates samples of the state space, the underlying geometry of the state space imposes no constraints on the implementation of the algorithm. Implementation of the EKF requires calculating Jacobian matrices which, in turn, involve differentiation of the measurement function with respect to the state variables. However, for discrete state variables, the derivative is undefined.

2. PARTICLE FILTERING FOR GROUND TARGET TRACKING

In the following sections we discuss the particle filtering algorithm applied to the ground target tracking problem. Since particle filters implement the Bayesian solution to state estimation, we briefly review Bayesian estimation and the theory behind particle filtering. Subsequently, we define the target dynamic state and measurement model.

2.1. Bayesian State Estimation

The Bayesian framework for state estimation problems represents a theoretically sound approach which entails determining the probability density function (PDF) of an unknown random vector. The following formula, referred to as Bayes' Rule, provides a means by which the prior PDF of some unknown random time-varying parameter, $\mathbf{x}_k$, is updated using a likelihood function

$$p(\mathbf{x}_k|Y^k) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|Y^{k-1})}{p(\mathbf{y}_k|Y^{k-1})} \tag{1}$$

where $Y^k = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ represents a sequence of measurements. The denominator in (1) is merely a normalizing factor. The term $p(\mathbf{x}_k|Y^{k-1})$ represents the PDF of the state at time k conditioned on the past measurements. Since $p(\mathbf{x}_k|Y^{k-1})$ is not conditioned on the current measurement, it is often referred to as the *predicted* state density or the *propagated* state density. The predicted state density is given by

$$p(\mathbf{x}_k|Y^{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|Y^{k-1})d\mathbf{x}_{k-1} \tag{2}$$

where $p(\mathbf{x}_{k-1}|Y^{k-1})$ is the updated state PDF from the last iteration. Thus, (1) and (2) together allow the propagation and updating of the target state density at time $k-1$ with a measurement at time k. The measurements are related to the unknown parameter through the following measurement equation

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k) \tag{3}$$

where $\mathbf{h}_k(\cdot,\cdot)$ is a possibly time-varying nonlinear function of the unknown parameter, $\mathbf{x}_k$, and $\mathbf{v}_k$ is typically assumed to be a sample of a white noise measurement process whose PDF is known. The state is assumed to evolve according to the following general dynamic equation

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{w}_k) \tag{4}$$

where, again, $\mathbf{f}_k(\cdot,\cdot)$ may be a time-varying nonlinear function of the state and $\mathbf{w}_k$ is typically modeled as a white noise process with a known PDF. The difficulty in analytically determining the PDF of $\mathbf{x}_k$ is that only for special cases does a closed-form (i.e., finite) description of exist. When (3) and (4) take on the special linear forms

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}_k \mathbf{x}_k + \mathbf{\Gamma}_k \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{5}$$

where $\mathbf{w}_k$ and $\mathbf{v}_k$ are samples from a Gaussian white noise process, then the PDFs are all Gaussian and, thus, are completely characterized by the first and second moments (i.e., mean and covariance). For the linear case modeled as in (5), the resulting Bayes' implementation is the well-known Kalman Filter. However, for a plethora of other cases (e.g., nonlinear, non-Gaussian), the Kalman filter is inappropriate. While some success has been achieved by applying the EKF (Extended Kalman filter), derived by linearizing the measurement equation and/or dynamic equation, the behavior of the EKF is often unpredictable. The difficulty in applying the EKF lies in the underlying assumption that the PDF of the unknown parameter (i.e., target state) can be sufficiently approximated by a Gaussian density function, a dubious assumption, particularly for the case of multimodal PDFs.

When linearization of the measurement model and/or dynamic model is not feasible or leads to unsatisfactory results, another approach to the implementation of Bayes' methods is the approximation of the true PDF using a model density. Various approximations include a piece-wise constant approximation,[7,8] a point-mass approximation[9] or Gaussian sum approximation[10] of the true target state posterior PDF. These *deterministic* methods all involve approximating the PDF of the state on a fixed grid over the state space and, therefore, suffer from the *curse of dimensionality*. To cope with the computational complexity of these fixed-grid approaches, sub-optimal pruning of basis functions (i.e., points in the grid) is usually required. In contrast, the method described next naturally implements a randomly evolving grid based on the likelihoods of the received data.

2.2. Bayesian Estimation through Particle Filtering

Stochastic sampling methods randomly sample points in the state space and determine the value of the target state PDF at these points. Referred to generally as Monte Carlo methods, the computation required for their implementation in real-time systems precluded them from use except in statistical batch processing applications. With the revolution in computing power over the last decade and advent of parallel computing structures (particle filtering methods are easily parallelizable), Monte Carlo methods have experienced a resurgence in the field of estimation and filtering. Particle filters approximate the PDF of an unknown random quantity by a weighted sum of delta functions. We will give a brief description of particle filtering here but urge the reader to consult various references for a more in-depth description.[2,11,12]

First, suppose we have a set of points and associated weights in the target state space, $\{w_{k-1}^{(i)}, \mathbf{x}_{k-1}^{(i)}\}$ for $i = 1, \ldots, N_s$, which represent samples from the updated target state PDF at time $k-1$. We can write this approximation using a weighted sum of delta functions as

$$p(\mathbf{x}_{k-1}|Y^{k-1}) \approx \sum_{i=1}^{N_s} w_{k-1}^{(i)} \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(i)}) \tag{6}$$

where $w_{k-1}^{(i)}$ is the weight associated with the i^{th} point and $\sum_{i=1}^{N_s} w_{k-1}^{(i)} = 1$. A filtering algorithm is a process by which the set of points representing $p(\mathbf{x}_{k-1}|Y^{k-1})$ is transformed to a set of points, $\{w_k^{(i)}, \mathbf{x}_k^{(i)}\}$ for $i = 1, \ldots, N_s$, representing $p(\mathbf{x}_k|Y^k)$. The particle filter accomplishes this in two steps. The first step is to transform the points representing the updated density at time $k-1$ into a set of points representing the predicted density, $p(\mathbf{x}_k|Y^{k-1})$, by *simulating* the evolution of the target state as dictated by (4). Thus, each sample point, $\mathbf{x}_{k-1}^{(i)}$ is substituted into (4), along with a value for the process noise (obtained by sampling from the known process noise distribution). The ramifications of *simulating* the state dynamics are that the dynamics may be highly nonlinear, and the process noise distribution may be non-Gaussian without affecting the efficacy of the algorithm. In fact, a closed-form mathematical description of the target dynamics is not required. Any means by which we can emulate the evolution of the state is sufficient to propagate the target state PDF forward in time.

The second step in the filtering process is to utilize a measurement at time k, $\mathbf{y}_k$, to update the set of particles representing the propagated density, $\{w_{k-1}^{(i)}, \mathbf{x}_k^{(i)}\} \sim p(\mathbf{x}_k|Y^{k-1})$ to the set of particles (with updated weights) $\{w_k^{(i)}, \mathbf{x}_k^{(i)}\} \sim p(\mathbf{x}_k|Y^k)$. The process by which the points are transformed is known as importance sampling. Ideally, we would like to sample from the updated density, $p(\mathbf{x}_k|Y^k)$ directly. Obviously, this is typically not possible since the whole objective here is to approximate the *unknown* updated density. However, through importance sampling, we sample points from a *known* distribution and weight them in order to simulate samples from the unknown distribution. We have chosen to use the propagated state PDF as the importance density which leads to a simple weighting scheme of the samples. Specifically, given a measurement, $\mathbf{y}_k$, and the set of weighted, propagated points, $\{w_{k-1}^{(i)}, \mathbf{x}_k^{(i)}\}$, the updated set of points is given by $\{w_k^{(i)}, \mathbf{x}_k^{(i)}\} \sim p(\mathbf{x}_k|Y^k)$ where the updated weights are given by

$$w_k^{(i)} = \frac{w_{k-1}^{(i)} p(\mathbf{y}_k|\mathbf{x}_k^{(i)})}{\sum_{j=1}^{N_s} w_{k-1}^{(j)} p(\mathbf{y}_k|\mathbf{x}_k^{(j)})} \tag{7}$$

where $p(\mathbf{y}_k|\mathbf{x}_k^{(i)})$ is the measurement likelihood given that the target state is represented by the i^{th} particle, $\mathbf{x}_k^{(i)}$. Unfortunately, it is known that over time the particle weights will degenerate such that one particle will dominate. To avoid this degeneracy, the updated, weighted distribution of points is *resampled* to yield a new set of points which are equally weighted.

Note that to update the particle weights, only knowledge of the measurement likelihood function is required. The actual form of the measurement equation, given by (3), is not necessary. Without requiring a closed-form mathematical description of the measurements, the particle filter can naturally incorporate feature measurements into the filtering process. The dependence of the measured features on the target state is often difficult, if not impossible, to describe using a closed-form mathematical function. However, measured feature data with corresponding known values of the target state are often available from which a measurement likelihood function can be built. As was mentioned previously, the goal of this research is the development of a joint tracking and identification algorithm which incorporates both kinematic and feature measurements.

2.3. GMTI Measurement Model

Any filtering algorithm requires a measurement model relating the measurements to the target states. Typically, a closed-form mathematical function is sought relating the target states to the measurements. We have seen that updating the target state PDF simply requires a measurement likelihood function and not the actual function relating target states to the measurements. In the case of the GMTI radar measurements both the measurement function given by (3) and the measurement likelihood function are known. For feature measurements, such as HRR profiles, the measurement function (3) is generally not available. However, given a collection of measurements for which the target state is known, measurement likelihoods can be constructed. In this section we discuss the kinematic measurement likelihood that was utilized by the particle filtering algorithm for tracking ground targets.

The kinematic measurements at time k processed by the filtering algorithm are the range, range rate and azimuth ($[r_k \ \dot{r}_k \ \phi_k]^T$) and are nonlinear functions of the target state and the known sensor position at time k, $[x_{s,k} \ y_{s,k} \ z_{s,k}]^T$. The measurements are assumed corrupted with additive white Gaussian noise. Thus, the likelihood functions are Gaussian with a mean given by the particular nonlinear function relating the target state to the measurements. The 3-dimensional MTI measurement can be modeled using a 2-dimensional tracking scenario assuming that the elevation angle is obtained by terrain clamping. Thus, the measurements are the range from the target to the sensor when projected onto the plane in which the target lies, the azimuth and the range rate which is the closing speed. The kinematic measurement likelihood is given by

$$p(\mathbf{y}_k|\mathbf{x}_k) = p(r_k, \dot{r}_k, \phi_k|\mathbf{x}_k) = p(r_k|\mathbf{x}_k)p(\dot{r}_k|\mathbf{x}_k)p(\phi_k|\mathbf{x}_k) \tag{8}$$

where we have assumed that the range, range rate and azimuth measurement errors are uncorrelated. Note that the target state (in a local flat plane) is given by

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} \tag{9}$$

Furthermore, we assume that each individual density in (8) is Gaussian and given by

$$\begin{aligned} p(r_k|\mathbf{x}_k) &= \frac{1}{\sqrt{2\pi\sigma_r^2}} \exp\left(\frac{-[r_k - \sqrt{(x_k - x_{s,k})^2 + (y_k - y_{s,k})^2}]^2}{2\sigma_r^2} \right) \\ p(\dot{r}_k|\mathbf{x}_k) &= \frac{1}{\sqrt{2\pi\sigma_{\dot{r}}^2}} \exp\left(\frac{-\left[\dot{r}_k - \frac{\dot{x}_k(x_k - x_{s,k}) + \dot{y}_k(y_k - y_{s,k})}{\sqrt{(x_k - x_{s,k})^2 + (y_k - y_{s,k})^2}}\right]^2}{2\sigma_{\dot{r}}^2} \right) \\ p(\phi_k|\mathbf{x}_k) &= \frac{1}{\sqrt{2\pi\sigma_\phi^2}} \exp\left(\frac{-\left[\phi_k - \tan^{-1}\left(\frac{y_k - y_{s,k}}{x_k - x_{s,k}}\right)\right]^2}{2\sigma_\phi^2} \right) \end{aligned} \tag{10}$$

Recall that updating the target state PDF with a measurement involves updating the weight of each particle by calculating the likelihood that the measurement originated from a target having the state given by the particular particle. The kinematic measurement likelihood for the particle is given by substituting the particle state and measurement into (10).

2.4. Target Dynamic Model

In the tracking of ground targets, a road network is commonly employed to aid the target state estimation problem. By restricting the position of the target to lie within the road network, improved state estimation should be possible. Additionally, the speed of the ground vehicles can also be constrained to lie on an interval such as $[-v_{max}, v_{max}]$. Constraints such as these aid the estimation problem by reducing the set over which the unknown random parameters lie. However, incorporating the mathematical descriptions of these constraints into the standard estimation formulation can be difficult and awkward. For the particle filtering algorithm, the structure of the underlying state space imposes no constraints on its implementation. It is only necessary that we can sample points from that state space. In this section we formulate the road-constrained tracking problem and show how naturally these constrained subsets are incorporated into the particle filtering algorithm. The number of particles directly affects the accuracy with which the particle filtering algorithm represents the optimal Bayesian solution. Incorporating constraints into the estimation problem, reduces the size of the state space and, thus, less particles are required.

The road network can be modeled as a set of line segments (of any shape) $\{l_1, l_2, \ldots, l_L\}$ and a set of nodes $\{n_1, n_2, \ldots, n_M\}$ representing the intersections of the roads. The position of a target can be described by an integer representing the line segment on which the target lies and a positive real number representing the distance from the starting node of that line segment to the target's position on the segment. A simplified model of a vehicle's velocity assumes the velocity is tangent to the road at the position of the vehicle on the road network. Thus, given a position of the vehicle on the road network, one real number in the interval is sufficient to describe the vehicle's velocity. Mathematically, then, the target's state vector can be represented by $\mathbf{x}_k = [l_k \; d_k \; v_k]^T$ where $l_k \in \{l_1, l_2, \ldots, l_L\}$ is the line segment on which the target lies at time k, $d_k \in [0, d_{max}]$ is the distance from the beginning node on segment l_k, and v_k is the speed of the target. Originally, the state comprised these three variables. However, a few problems highlighted the need to allow a target to vary from the road network slightly:

- A road segment represented the center of the road, and vehicles would drive off the center line (in a lane). The particles were constrained to lie on the center line, however. The result was equivalent to a road bias.
- While the vehicle was only slightly (approximately 2.25 meters) off the center of the road for vehicles simulated in SLAMEM^{TM}, other test cases had varying off-road biases. For example, *real* measurements of GPS-equipped vehicles driving around in an area were acquired. Knowing the region in which the vehicles were observed, a road network for the area was used along with the vehicle trajectories, calculated using GPS measurements, to build a test case in SLAMEM^{TM}. Due to errors in the road network or other unknown errors, the vehicles would sometimes be off the road network segments by 20 meters.

Due to the presence of these unmodeled errors, track breaks were common. The measurement noise fed to the filter could have been increased to account for these errors, but the noise is not additive Gaussian noise. Therefore, we decided to add another variable to the target state representing the off-road distance. The off-road distance was modeled as a variable whose maximum value was bounded. Including the off-road distance as a state variable results in the target state $\mathbf{x}_k = [l_k \; d_k \; \delta_k \; v_k]^T$ where $\delta_k \in [-\delta_{max}, \delta_{max}]$ and $\delta_{max} > 0$.

The dynamic model is fairly straightforward. A Gaussian random vector, $\mathbf{w}_k \in \mathbb{R}^2$, is generated containing an acceleration process noise and another process noise variable which allows the off-road distance, δ_k, to vary slightly over time. Denoting the random process noise vector elements as $\mathbf{w}_k = [w_{k,1} \; w_{k,2}]^T$ in which $w_{k,1}$ is the on-road acceleration process noise, and $w_{k,2}$ is the road-offset process noise, we now mathematically describe the target dynamic model for the on-road case. The propagation of the target's speed is given by

$$v_{k+1} = \begin{cases} \min(v_{max}, v_k + \Delta t_k w_{k,1}) & v_k + \Delta t_k w_{k,1} > 0 \\ \max(-v_{max}, v_k + \Delta t_k w_{k,1}) & v_k + \Delta t_k w_{k,1} < 0 \end{cases} \tag{11}$$

where the speed is constrained, and Δt_k is the time until the next detections are processed. Mathematically describing d_{k+1} is more difficult. Ostensibly, d_{k+1} is given by

$$d_{k+1} = d_k + \Delta t_k v_k + \frac{\Delta t_k^2}{2} w_{k,1} \tag{12}$$

However, if d_{k+1} is greater than the distance from the current position to the end of the current line segment, the target will lie on one of the next segments in the road network. Let ϵ_0 be the distance from the current target position to the end of segment l_k. Furthermore, suppose d_{k+1} is such that the particle travels through J more road segments, each with a length of ϵ_j, before stopping on the $J+1$ segment. Then d_{k+1} is the distance from the end of the $J+1$ segment and is given by

$$d_{k+1} = d_k + \Delta t_k v_k + \frac{\Delta t_k^2}{2} w_{k,1} - \sum_{j=0}^{J} \epsilon_j \tag{13}$$

The state variable representing the line segment, l_{k+1} in the road network on which the particle lies at time $k+1$, depends on whether the target moves far enough to leave the current segment l_k (i.e., reach an intersection in the road network). Again, if $d_{k+1} > \epsilon_0$ (where d_{k+1} is given by (12)), then the target will continue on a different line segment.

The off-road distance variable has a straightforward equation describing its time evolution

$$\delta_{k+1} = \begin{cases} \min(\delta_{max}, \delta_k + \Delta t_k w_{k,2}) & \delta_k + \Delta t_k w_{k,2} > 0 \\ \max(-\delta_{max}, \delta_k + \Delta t_k w_{k,2}) & \delta_k + \Delta t_k w_{k,2} < 0 \end{cases} \tag{14}$$

Another process noise variable is often required during the state evolution phase of the filtering process. The situation requiring a process noise variable arises when a particle propagates to an intersection. To determine, upon reaching the intersection, on which road the target continues, a discrete random variable is drawn. The distribution of this discrete random variable is uniform over a set of integers, each integer representing a different line segment (i.e., road) at the next intersection. Suppose, for example, the propagated position of the target goes past the next intersection, and there are three possible roads to take at the intersection. We number the roads 1, 2 and 3 and choose one of these numbers randomly, each with a probability of $\frac{1}{3}$ (or with some other distribution if prior knowledge of traffic patterns indicates that certain directions are more likely than others). This type of dynamic model is trivial to implement with the particle filter. At time k we have a scattering of particles along the road network representing the PDF of the positions of the targets. Using the described model, these particles are propagated along the road network to yield a scattering of particles on the road network at time $k+1$ representing the propagated PDF of the target positions.

We have given a fairly complete description of the particle filtering algorithm and its initial implementation in this effort. Next, we examine some results demonstrating the filter's feasibility and potential to effectively address the ground target tracking problem.

3. PARTICLE FILTERING ALGORITHM PERFORMANCE

The particle filtering algorithm described was implemented in C++ for evaluation using a simulation developed at Toyon Research known as SLAMEM^{TM}. With extensive modeling of terrain (DTED), roads, sensors and vehicles (GVS^{TM}), SLAMEM^{TM} is an excellent tool for evaluating algorithms. The particle filtering algorithm was incorporated into SLAMEM^{TM} and initially evaluated for feasibility and potential strengths. The following are some of the conditions under which the algorithm was evaluated

- Two airborne sensors standing off about 90 km illuminate the region of interest (approximately $200km^2$).
- 10 targets were tracked for 1 hour.
- Kinematic measurement errors were $\sigma_r = 0.2m, \sigma_\phi = 0.001^R, \sigma_{\dot{r}} = 0.2m/s$.
- The PDF (probability density function) modeling the state of each object was represented by 10,000 particles.
- Each radar had a revisit time of approximately $30sec$.

An example of the performance of the particle filtering algorithm is shown in Figure 1. The position error and speed error for each Monte Carlo run are plotted where the errors represent the average over all tracks and all updates for each trial. For a rough comparison, a Kalman filter was run on the same case. The comparison should be taken as merely establishing the potential of the particle filter for this problem, since a true comparison of different filters would consider many performance measures and not simply the track accuracy.

3.1. Multi-Modal Target State PDF Representation

When the PDF of the target state is assumed to be Gaussian, representing the PDF with a mean and covariance is sufficient. For the non-Gaussian case, however, representing the PDF by a mean and covariance is an approximation whose accuracy depends on the actual form of the PDF. A particular case in which the Gaussian approximation is unsuitable is that in which the target state PDF is multi-modal. In the road-constrained ground target tracking problem addressed here, a multi-modal PDF occurs when targets travel through intersections. Any tracking algorithm must propagate the target state through intersections and cope with the inherent uncertainty in choosing which road segment the target takes. In Kalman filtering algorithms, a typical approach is to propagate the state along all possible road segments and *choose* the propagated path which "best" matches the next measurement. Making such a hard decision is often not desirable. Alternatively, a more complex multi-hypothesis type of approach may be considered to avoid making a hard decision after just one measurement.

A particle filter represents the target state PDF by a collection of particles, each representing a possible value of the target's state. In the propagation step, each particle propagates by simulating the dynamics of the target. Any randomness is simulated by drawing a random variable. For example, in our ground target tracking problem we have modeled the unknown on-road acceleration with Gaussian process noise. Thus, to propagate in time, a random Gaussian variable is drawn, and the particle's next position and velocity are determined using the drawn random variable as the constant acceleration. For vehicles that travel on the road, there is another

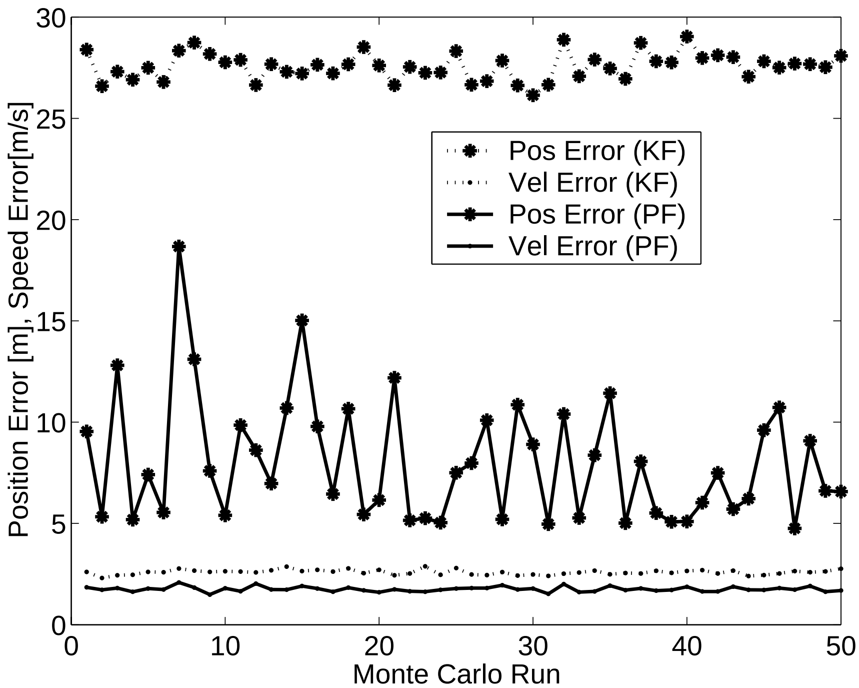

Figure 1. Average position and velocity error at each Monte Carlo run for a Kalman filter and the particle filter.

noise process in addition to the acceleration noise. When a particle approaches an intersection, the direction that the particle takes through the intersection is uncertain. Thus, one process noise model is to randomly choose one road segment at an intersection as the one on which the particle continues. The random decision may be uniform over all possible road segments, or, alternatively, the probability of choosing a particular segment may be a function of the type of object the particle represents and prior information about landmarks near the intersection. For example, if there is a military base down one road segment, then if the particle being propagated represents a military vehicle, there is a higher probability that the particle will choose that segment over other road segments. Through this mechanism, the dynamics are coupled to the target type and allow inference on the class type merely by observing the dynamics of the object in track.

For our initial implementation of the particle filter, road segments at an intersection are randomly chosen but uniform across all segments. When a track is propagated through an intersection, some particles will travel down each of the different segments, resulting in multiple clouds of particles (i.e., a multi-modal PDF). As measurements are received and processed, particles are weighted according to the measurement likelihoods and particles traveling down one segment (the one on which the actual target traveled) will, naturally, have higher likelihoods and will, thus, be more strongly represented in the updated target state PDF. The radar location with respect to the target determines the shape of the updated target state PDF. The updated PDF may immediately revert to a unimodal density, or only after several measurements will the PDF return to a unimodal density. In Figure 2 we illustrate an example of a situation in which a multi-modal PDF exists after passing through an intersection. The image on the left in Figure 2 shows the target after it has passed through the intersection and the collection of propagated points spreading out along the different road segments. The image on the right shows the updated and resampled set of points after incorporating a radar measurement. The radar geometry is such that with the large cross-range error, points on both road segments have the potential to have generated the measurement, although points on the upper road segment are favored (since the density of points is higher than on the lower segment). The ability to represent non-Gaussian and, in particular, multi-modal densities highlights one of the strengths of the particle filter which is inherently a non-parametric density estimator.

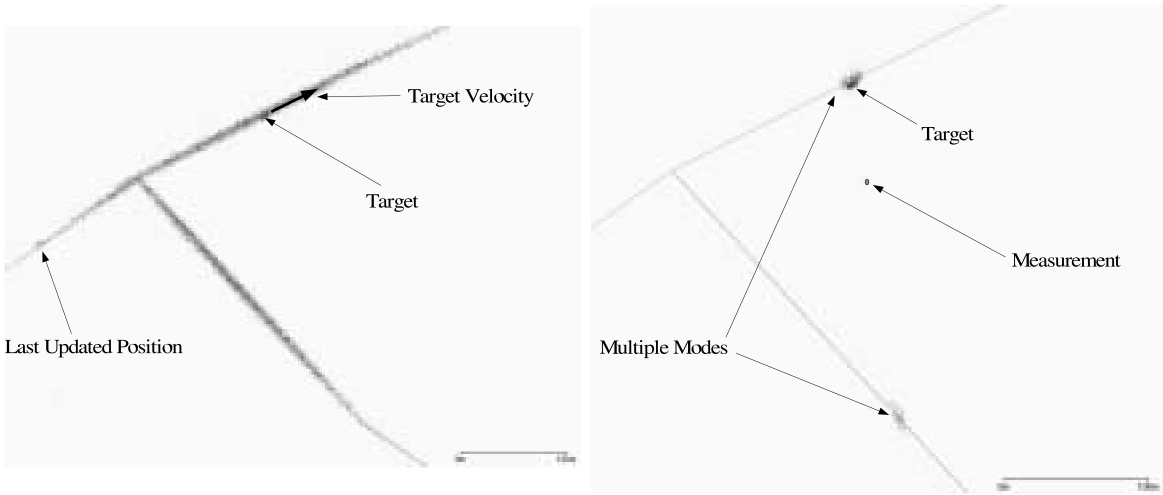

Figure 2. Snapshots of the particle cloud propagating through an intersection and after one update. Note that two particle clouds remain after the update representing a multi-modal target state PDF.

4. CONCLUSIONS AND FUTURE WORK

As stated previously, the ultimate goal of this research is to develop a particle filtering algorithm which tracks and identifies ground targets. In this work we have shown the initial results of a particle filtering algorithm for tracking ground targets when constrained to travel on a road network. Progress has already been made in incorporating high-range resolution (HRR) radar profiles into the algorithm for both state estimation and identification. Future work involves handling both on-road and off-road vehicles, tracking vehicles through move-stop-move cycles, as well as other real-world issues.

ACKNOWLEDGMENTS

The work presented here was supported under contract No. F33615-02-M-1219 by Air Force Research Labs (AFRL). The authors would like to express their sincere thanks to the Program Manager Dr. Jeffery Layne for his support.

REFERENCES

1. C. Agate and K. Sullivan, "Signature-aided tracking using association hypotheses," in *Proceedings of SPIE Conference on signal processing, sensor fusion and target recognition XI*, **4729**, (Orlando, FL), April 2002.
2. N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F* **140**, pp. 107–113, April 1993.
3. S. Arulampalam and B. Ristic, "Comparison of the particle filter with range-parameterised and modified polar EKFs for angle-only tracking," in *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, **4048**, pp. 288–299, (Orlando, FL), April 2000.

4. X. Lin, T. Kirubarajan, Y. Bar-Shalom, and S. Maskell, "Comparison of EKF, pseudomeasurement and particle filters for bearing-only target tracking problem," in *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, **4728**, pp. 240–250, (Orlando, FL), April 2002.
5. N. Gordon, "A hybrid bootstrap filter for target tracking in clutter," *IEEE Transactions on Aerospace and Electronic Systems* **AES-33**, pp. 353–358, Jan. 1997.
6. A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Transactions on Signal Processing* **SP-49**, pp. 613–624, March 2001.
7. S. Kramer and H. Sorenson, "Recursive Bayesian estimation using piece-wise constant approximations," *Automatica* **24**(6), pp. 789–801, 1988.
8. R. A. Iltis, "State estimation using an approximate reduced statistics algorithm," *IEEE Transactions on Aerospace and Electronic Systems* **AES-35**, pp. 1161–1172, October 1999.
9. R. Bucy and K. Senne, "Digital synthesis of non-linear filters," *Automatica* **7**, pp. 287–298, 1971.
10. H. Sorenson and D. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica* **7**, pp. 465–479, 1971.
11. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing* **SP-50**, pp. 174–188, Feb. 2002.
12. A. Doucet, N. de Freitas, and N. Gordon, "An introduction to sequential monte carlo methods," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, eds., pp. 3–14, Springer, New York, NY, 2001.