Utilizing Negative Information to Track Ground Vehicles through Move-Stop-Move Cycles

Craig S. Agate*, Robert M. Wilkerson and Kevin J. Sullivan

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

ABSTRACT

Ground vehicles can be effectively tracked using a moving target indicator (MTI) radar. However, vehicles whose velocity along the line-of-sight to the radar falls below the minimum detectable velocity (MDV) are not detected. One way targets avoid detection, therefore, is to execute a series of move-stop-move motion cycles. While a target can be acquired after beginning to move again, it may not be recognized as a target previously in track. Particularly for the case of high-value targets, it is imperative that a vehicle be continuously tracked. We present an algorithm for determining the probability that a target has stopped and an estimate of its stopped state (which could be passed to a tasker to schedule a spot synthetic aperature radar (SAR) measurement). We treat a *non-detection* event as evidence that can be used to update the target state probability density function (PDF). Updating the target state PDF using a non-detection event pushes the probability mass into regions of the state space in which the vehicle is either stopped or traveling at a speed such that the range-rate fails the MDV. The target state PDF updated with the non-detection events is then used to derive an estimate of the stopped target's location. Updating the target state PDF using a non-detection event is, in general, non-trivial and approximations are required to evaluate the updated PDF. When implemented with a particle filter, however, the updating formula is simple to evaluate and still captures the subtleties of the problem.

Keywords: move-stop tracking, particle filter, ground targets, 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. The received detections often result from processing the returns from a ground-moving target indicator (GMTI) radar. The GMTI radar is able to reject returns from the background (ground clutter) by exploiting the difference in Doppler between a "moving" vehicle and the "stationary" background. The vehicle motion, however, must be along the line-of-sight (LOS) from the vehicle to the radar. Furthermore, the vehicle speed along the LOS must be above the minimum detectable velocity (MDV); otherwise, the vehicle will escape detection. For example, a vehicle that is stopped or moving *orthogonal* to the LOS from the vehicle to the radar will not be detected. When a tracked vehicle stops, no more detections are received on that vehicle. Without the ability to cope with possibly long periods of time without detections on a vehicle, the track is dropped (i.e., the vehicle is lost). While the vehicle will be tracked when it starts moving again, there is no association of the new track with the previous track, and any intelligence regarding the original vehicle in track is lost. For targets wishing to avoid surveillance, going through several move-stop-move cycles is an effective technique.

In this paper we discuss an algorithm to *continuously* track vehicles through move-stop-move cycles in a road-constrained environment. The algorithm is based on particle filtering methods[1] and utilizes *non-detection* events in order to update the target state density and infer both that the target may have stopped and the target's state. Note that we consider the non-detection event as evidence to actually *update* the target state probability density function (PDF) through Bayes' rule. For a similar approach to utilizing a *non-detection* event, see[2,3] in which the authors use a failed detection to determine whether a tracked aircraft is flying in the blind Doppler zone of the radar.

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

Signal Processing, Sensor Fusion, and Target Recognition XIII, edited by Ivan Kadar,
Proceedings of SPIE Vol. 5429 (SPIE, Bellingham, WA, 2004)
0277-786X/04/$15 · doi: 10.1117/12.542575

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[4, 5] and tracking multiple targets in clutter[6] to which to apply particle filtering methods. Recent work[7] 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 that 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.[1] 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 comprising 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, involves differentiation of the measurement function with respect to the state variables. However, for discrete state variables, the derivative is undefined.

2. PARTICLE FILTERING FOR BAYESIAN STATE ESTIMATION

In the following sections we discuss the particle filtering algorithm as a Bayesian state estimation algorithm by briefly reviewing Bayesian estimation and giving a brief discussion of a basic particle filtering algorithm.

2.1. Bayesian State Estimation

The Bayesian framework for state estimation problems represents a theoretically sound approach that 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

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

where $Z^k = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$ represents a sequence of measurements. The denominator in (1) is merely a normalizing factor. The term $p(\mathbf{x}_k|Z^{k-1})$ represents the PDF of the state at time k conditioned on the past measurements. Since $p(\mathbf{x}_k|Z^{k-1})$ 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

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

where $p(\mathbf{x}_{k-1}|Z^{k-1})$ 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

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{3}$$

where $\mathbf{h}_k(\cdot,\cdot)$ is a possibly time-varying nonlinear function of the unknown state, $\mathbf{x}_k$, and $\mathbf{n}_k$ 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 exist. For example, when (3) and (4) are linear with added Gaussian noise, the state PDF is Gaussian and can be determined using the well-known Kalman filtering algorithm.

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 can be 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,[8,9] a point-mass approximation[10] or Gaussian sum approximation[11] 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. In contrast, the method described next naturally implements a randomly evolving grid based on the likelihoods of the received data.

2.2. Bayesian State 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 rapid increase in computing power and the 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.[1,12,13]

Define $\mathbf{x}_k$ to represent the target's kinematic state (e.g., position, velocity) at time k. Next, suppose that at time $k-1$ we have a set of points and associated weights in the target state space, $\{\gamma_{k-1}^{(i)}, \mathbf{x}_{k-1}^{(i)}\}$ for $i = 1, \ldots, N_s$, that represent *samples* from the updated target state PDF at time $k-1$. Thus, the updated target state PDF at time $k-1$ can be approximated as the following sum of delta functions

$$p(\mathbf{x}_{k-1}|Z^{k-1}) \approx \sum_{i=1}^{N_s} \gamma_{k-1}^{(i)} \delta[\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(i)}] \tag{5}$$

where $\gamma_{k-1}^{(i)}$ is the weight associated with the i^{th} point and $\sum_{i=1}^{N_s} \gamma_{k-1}^{(i)} = 1$. A filtering algorithm is a process by which the set of points representing $p(\mathbf{x}_{k-1}|Z^{k-1})$ is transformed to a set of points, $\{\gamma_k^{(i)}, \mathbf{x}_k^{(i)}\}$ for $i = 1, \ldots, N_s$, representing $p(\mathbf{x}_k|Z^k)$. 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, $p(\mathbf{x}_k|Z^{k-1})$, 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, $\mathbf{z}_k$, to update the set of particles representing the propagated density, $\{\gamma_{k-1}^{(i)}, \mathbf{x}_k^{(i)}\} \sim p(\mathbf{x}_k|Z^{k-1})$ to the set of particles (with updated weights) $\{\gamma_k^{(i)}, \mathbf{x}_k^{(i)}\} \sim p(\mathbf{x}_k|Z^k)$. The process by which the points are transformed is known as importance sampling. Ideally, we would like to sample from the updated density, $p(\mathbf{x}_k|Z^k)$ 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. In the initial implementation 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, $\mathbf{z}_k$, and the set of weighted, propagated points, $\{\gamma_{k-1}^{(i)}, \mathbf{x}_k^{(i)}\}$, the updated set of points is given by $\{\gamma_k^{(i)}, \mathbf{x}_k^{(i)}\} \sim p(\mathbf{x}_k|Z^k)$ where the updated weights are given by

$$\gamma_k^{(i)} = \frac{\gamma_{k-1}^{(i)} p(\mathbf{z}_k|\mathbf{x}_k^{(i)})}{\sum_{j=1}^{N_s} \gamma_{k-1}^{(j)} p(\mathbf{z}_k|\mathbf{x}_k^{(j)})} \tag{6}$$

where $p(\mathbf{z}_k|\mathbf{x}_k^{(i)})$ 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 that are equally weighted.

3. TRACKING GROUND TARGETS THROUGH MOVE-STOP-MOVE CYCLES

Having reviewed the particle filtering approach to Bayesian state estimation in Section 2, we now discuss the particular particle filtering algorithm used to track ground targets as they travel through move-stop-move cycles. As with all filtering algorithms, two distinct stages of the filter are discussed. The first stage in the filtering algorithm is the modeling of the target's dynamic motion and allows us to infer something about the target's state between receiving measurements. The second stage of the filter is the actual updating of the target state PDF using *evidence*, where the evidence is the result from a radar illumination of a region (i.e., a defined sensor footprint).

Before discussing the models for each stage of the filtering process, we must define the state of a ground target. As expected the target state includes the position and velocity of the target. Additionally, in order to cope with the move-stop motion of the targets, we add another discrete random variable to the target state vector that represents the target's *motion state*, where each motion state is characterized by a different motion model. For example, consider a target that has three motion states. The first motion state corresponds to a moving target constrained to a road network, while the second motion state corresponds to a target that moving without the constraint of a road network. The third motion state corresponds to a target that is stopped. We will represent the motion state at time k as the discrete random variable $\mathcal{S}_k$ where

$$\mathcal{S}_k \in \{\mathcal{M}_1 = \text{On-Road Motion}, \mathcal{M}_2 = \text{Off-Road Motion}, \mathcal{M}_3 = \text{Stopped}\} \tag{7}$$

Thus, we can write the target state vector $\mathbf{x}_k$ as

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{r}_k \\ \mathbf{v}_k \\ \mathcal{S}_k \end{bmatrix} \tag{8}$$

where $\mathbf{r}_k \in \mathbb{R}^3$ is the position of the target, and $\mathbf{v}_k \in \mathbb{R}^3$ is the velocity of the target.

3.1. Dynamic Motion Model

Now that the ground target state has been defined, we discuss the dynamic motion model that dictates how the system evolves over time. The general model for the evolution of a dynamic system (in our case, the "system" is a ground vehicle) is given by (4). The introduction of the discrete random variable, $\mathcal{S}_k$, into the ground vehicle state requires a model for the time evolution of not just the position and velocity of the target but the target's motion state as well. Typically, the evolution of a discrete random state is modeled as a Markov process. We model the evolution of the system motion state as a first-order Markov process and define the transition probabilities that govern the probability of transitioning from one motion model to another as

$$P_{i,j}(\mathbf{x}_k) = P(\mathcal{S}_{k+1} = \mathcal{M}_j | \mathbf{x}_k, \mathcal{S}_k = \mathcal{M}_i) \tag{9}$$

For example, $P_{1,3}(\mathbf{x}_k)$ is the probability that a target with state $\mathbf{x}_k$ transitions from moving while constrained to the road network to a stopped state. Note the explicit dependence of the transition probabilities upon the target state (as opposed to static transition probabilities).

In addition to the evolution of the target's motion state, there is a dynamic model for the time-evolution of continous target states (i.e., the position and velocity). As discussed, there is one dynamic model for *each* motion state; therefore, there is a motion model for targets that move while constrained to the road, a motion model for targets that move *without* knowledge of the road network and, finally, the motion model for targets that are stopped. We briefly mention the different motion models but will not go into the details as any reasonable dynamic model will work within the proposed algorithm.

In ground target tracking problems, a road network is commonly employed to aid target state estimation.[14] 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.

An off-road motion model defines the motion of a target that has no knowledge of the road network. Off-road motion models can incorporate terrain-type constraints such as gradients, leading to a higher fidelity model than, for example, a simple model that uses a parasitic acceleration for both the x and y directions.

3.2. Evidential Update of the Target State PDF

Traditionally, any detections received as the result of a sensor dwell are used to update the target state PDF of any tracks within the dwell (or sensor footprint). However, a sensor dwell that returns no detections but contains track estimates within its footprint, still returns useful information that can be used to update the target state PDF of the relevant tracks. For the case of stopped targets, utilizing an "empty" dwell is particularly important for deciding when a track represents a stopped target (so that, for example, the sensor tasker can order a synthetic aperature radar (SAR) image of the area in which the target is believed to have stopped). A radar (more generally a sensor) excites a region on the ground with electromagnetic energy and processes the returned energy. The signal processing of the returned energy may result in "detections" that represent the possible locations of target. We refer to the illumination of a region of the ground as a *sensor dwell* and the illuminated region as the *sensor footprint*. We next derive a procedure for updating the track state using the results of a dwell as evidence.

First, we define the following notation

- Let $\mathcal{D}_k$ be the locus of points in the target state space that is both within the sensor footprint *and* has a direct line of sight to the sensor (e.g., not blocked by terrain or foliage) at time k.
- Let e_k be the evidence that results from processing the returned radar energy in a dwell at time k.
- Let $\mathcal{E}^k$ be the sequence of evidence collected on a target where

$$\mathcal{E}^k = \{e_1, e_2, \ldots, e_k\} \tag{10}$$

- Let v_{min} be the minimum detectable velocity. The MDV is the minimum value that the projection of the target velocity along the line of sight (LOS) must be in order for the MTI radar to distinguish the target from the background.
- Let P_D be the probability of detection given that a target is within the footprint *and* satisifies the minimum detectable velocity (MDV).
- Let $\mathbf{x}_k$ be the target state at time k defined in (8)
- Let $\mathcal{X}$ be the target state space (note that $\mathcal{X} \subset \mathbb{R}^6 \times \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\}$).
- Let $\mathbf{z}_k$ represent the received measurement from a dwell at time k in which a target *was* detected.
- Let Z^k represent a sequence of measurements up through and including time k.
- Let $\mathbf{r}_{s,k}$ represent the sensor position at time k.
- Let η_k represent the *event* that a target was detected during the sensor dwell at time k (while $\neg\eta_k$ represents the event that no target was detected).
- Let $\mathbf{u}_{los}$ represent a unit vector in the direction of the line of sight (i.e., $\mathbf{u}_{los} = \frac{\mathbf{r}_k - \mathbf{r}_{s,k}}{||\mathbf{r}_k - \mathbf{r}_{s,k}||}$).

- Let $p(\mathbf{x}_k|\mathcal{E}^k)$ represent the target state PDF at time k given the sequence of collected evidence $\mathcal{E}^k$. Note that the sequence of evidence could contain actual detection events (and their associated measurements) as well as non-detection events.

We now derive the updated target state PDF using evidence from a sensor dwell. Assume that we have the propagated target state PDF, $p(\mathbf{x}_k|\mathcal{E}^{k-1})$, and we want to use a piece of evidence, e_k, from a sensor dwell to determine $p(\mathbf{x}_k|\mathcal{E}^{k-1}, e_k)$. The first consideration is the nature of the evidence, e_k. The are two possible types of evidence to consider. If no target is detected, then $e_k = \neg\eta_k$. If a target was detected, then $e_k = \{\eta_k, \mathbf{z}_k\}$ where the evidence is both the event, "a target was detected" *and* the actual measurement (e.g., range, range rate and azimuth) associated with that event.

3.2.1. Target State PDF Update for an Empty Dwell

If the sensor dwell resulted in no detected targets, then $e_k = \neg\eta_k$, and we can expand the updated target state PDF using Bayes' rule as

$$p(\mathbf{x}_k|\mathcal{E}^{k-1}, \neg\eta_k) = \frac{P(\neg\eta_k|\mathbf{x}_k, \mathcal{E}^{k-1})p(\mathbf{x}_k|\mathcal{E}^{k-1})}{P(\neg\eta_k|\mathcal{E}^{k-1})} \tag{11}$$

Note that the term, $P(\neg\eta_k|\mathbf{x}_k, \mathcal{E}^{k-1})$, which is akin to the "measurement likelihood" in the normal expansion of the PDF using Bayes' rule, is the *probability* that no detection is received for the track represented by the sequence of evidence, $\mathcal{E}^{k-1}$. Furthermore, note that $\neg\eta_k$ is conditionally independent of $\mathcal{E}^{k-1}$ given the value of the state, $\mathbf{x}_k$. To calculate the probability, $P(\neg\eta_k|\mathbf{x}_k)$, we shall partition the state space into disjoint regions (i.e., the total probability formula).

$$\begin{aligned} P(\neg\eta_k|\mathbf{x}_k) &= P(\neg\eta_k, \mathbf{r}_k \notin \mathcal{D}_k|\mathbf{x}_k) + P(\neg\eta_k, \mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| < v_{min}|\mathbf{x}_k) \\ &+ P(\neg\eta_k, \mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| \geq v_{min}|\mathbf{x}_k) \end{aligned} \tag{12}$$

Note that we have utilized the fact that the state space can be written as the following union of disjoint sets

$$\mathcal{X} = \{\mathbf{r}_k \notin \mathcal{D}_k\} \cup \{\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| < v_{min}\} \cup \{\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| \geq v_{min}\} \tag{13}$$

The expression, $||\mathbf{v}_k^T\mathbf{u}_{los}||$ represents the projection of the target velocity along the line-of-sight from the radar to the target. A typical MTI radar has a minimum detectable velocity (MDV) that an object must satisfy in order for the object to be detected. However, even if an object satisfies this MDV, there may still be a less than unity probability of detection which depends on the radar. The probabilities in (12) can be expanded yielding

$$\begin{aligned} P(\neg\eta_k|\mathbf{x}_k) &= P(\neg\eta_k|\mathbf{r}_k \notin \mathcal{D}_k, \mathbf{x}_k)P(\mathbf{r}_k \notin \mathcal{D}_k|\mathbf{x}_k) \\ &+ P(\neg\eta_k|\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| < v_{min}, \mathbf{x}_k)P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| < v_{min}|\mathbf{x}_k) \\ &+ P(\neg\eta_k|\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| \geq v_{min}, \mathbf{x}_k)P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| \geq v_{min}|\mathbf{x}_k) \end{aligned} \tag{14}$$

We can now simplify the probability in (14) by noting the following. If the target position is either not in the physical footprint of the sensor or is within the footprint but has no direct line of sight to the sensor (perhaps due to terrain blockage), then the probability that the target will *not* be detected is one; therefore, $P(\neg\eta_k|\mathbf{r}_k \notin \mathcal{D}_k, \mathbf{x}_k) = 1$. The term $P(\neg\eta_k|\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| < v_{min}, \mathbf{x}_k)$ is the probability that the target is *not* detected given that its position in the state space puts it within the sensor footprint with a direct line of sight to the sensor but such that the projection of its velocity along the LOS does *not* satisfy the MDV; therefore, its value is one. Finally, the term $P(\neg\eta_k|\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| \geq v_{min}, \mathbf{x}_k)$ is the probability that the target is not detected given that its position in the state space puts it within the sensor footprint and has a direct LOS, and, furthermore, that its velocity satisfies the MDV to be detected. Now even with these requirements met, there is a small probability that the target *still* will not be detected. The probability that a sensor will detect a target given that it is in the field of view and satisfies the MDV, is radar specific and denoted P_D. Incorporating these values into (14) and substituting the result into the target state PDF update equation in (11) yields

$$\begin{aligned} p(\mathbf{x}_k|\mathcal{E}^{k-1}, \neg\eta_k) &\propto \left[P(\mathbf{r}_k \notin \mathcal{D}_k|\mathbf{x}_k) + P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| < v_{min}|\mathbf{x}_k)\right. \\ &+ \left.(1-P_D)\cdot P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| \geq v_{min}|\mathbf{x}_k)\right] p(\mathbf{x}_k|\mathcal{E}^{k-1}) \end{aligned} \tag{15}$$

In general, it is difficult to analytically determine $p(\mathbf{x}_k|\mathcal{E}^{k-1}, \neg\eta_k)$ in (15). The terms in the brackets are all probabilities that are equal to one or zero (excluding the term, $1 - P_D$) depending on whether the given $\mathbf{x}_k$ satisfies the conditions. At this point it is useful to examine (15) for some special conditions in order to satisfy our intuitive sense as to how it *should* behave under particular conditions.

Special Case I: $P(\mathbf{x}_k \in \mathcal{D}_k|\mathcal{E}^{k-1}) \approx 0$
Suppose that practically *all* of the mass density of the predicted state PDF lies outside the sensor footprint. Thus, the following approximations are applicable for all $\mathbf{x}_k$ in the "significant" region of the state space with respect to $p(\mathbf{x}_k|\mathcal{E}^{k-1})$

$$\begin{aligned} P(\mathbf{r}_k \notin \mathcal{D}_k|\mathbf{x}_k) &\approx 1 \\ P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| < v_{min}|\mathbf{x}_k) &\approx 0 \\ P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| \geq v_{min}|\mathbf{x}_k) &\approx 0 \end{aligned} \tag{16}$$

Substituting (16) into (15) results in the following

$$p(\mathbf{x}_k|\mathcal{E}^{k-1}, \neg\eta_k) \approx p(\mathbf{x}_k|\mathcal{E}^{k-1}) \tag{17}$$

For the case where the track estimate is "far" outside the sensor footprint, an non-detection event gives us no information, and the "updated" target state PDF is simply the predicted target state PDF.

Special Case II: $P(\mathbf{x}_k \notin \mathcal{D}_k|\mathcal{E}^{k-1}) \approx 0$
In this case the mass density of the predicted target state PDF is mostly concentrated within the sensor footprint. We can, therefore, assume that $P(\mathbf{r}_k \notin \mathcal{D}_k|\mathbf{x}_k) \approx 0$, and (15) simplifies to

$$\begin{aligned} p(\mathbf{x}_k|\mathcal{E}^{k-1}, \neg\eta_k) &\propto \left[P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| < v_{min}|\mathbf{x}_k)\right. \\ &+ \left.(1-P_D)\cdot P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| \geq v_{min}|\mathbf{x}_k)\right] p(\mathbf{x}_k|\mathcal{E}^{k-1}) \end{aligned} \tag{18}$$

If we further assume a unity probability of detection, then (18) is further simplified to

$$p(\mathbf{x}_k|\mathcal{E}^{k-1}, \neg\eta_k) \propto P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| < v_{min}|\mathbf{x}_k)p(\mathbf{x}_k|\mathcal{E}^{k-1}) \tag{19}$$

The probability, $P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T\mathbf{u}_{los}|| < v_{min}|\mathbf{x}_k)$, in (19) is either zero or one which essentially sets the likelihood to zero for points in the state space representing a target which "should" have been detected. For points in the state space that represent targets moving too slowly to meet the MDV or moving fast but orthogonal (or nearly so) to the line of sight, the probability in (19) is one. Thus, any (probability density) mass in $p(\mathbf{x}_k|\mathcal{E}^k)$ associated with target states that should have been detected is *completely* moved to target states that are consistent with a non-detection event (i.e., targets that are slowly moving or moving orthogonal to the line of sight). With a non-unity probability of detection, the same shifting of mass in the state space occurs but is less complete in the sense that a *portion* of the mass associated with target states that should have been detected is moved to states more consistent with a non-detection event. However, since there is a non-zero, albeit small, probability that a target might not be detected even though it is within the sensor footprint, has a direct line of sight to the sensor and meets the MDV requirement, much but not *all* of the mass density associated with such points will be moved to target states more consistent with the occurrence of a non-detection event.

The probable non-Gaussian nature of the resulting density updated through (15) is one reason it would be difficult to determine the updated PDF without some approximations to the predicted target state PDF. A multiple model approach[15] in which the target state PDF is the sum of two Gaussians (a Gaussian PDF for the moving motion model and a Gaussian PDF for the stopped motion model) can allow an approximate evaluation of (15). However, target states which had a nonzero velocity (but did not satisfy the MDV) would not be handled correctly in the two motion model approach. As alluded to earlier a particle filter is a pointwise approximation of the target state PDF and, therefore, considerably simplifies the calculation of (15) as shown next.

3.2.2. Updating the Target State PDF using a Point-wise Approximation

For a general state estmimation problem, the target state PDF is unlikely to be Gaussian and, therefore, some approximation of the PDF is required. Many approximations to the state PDF have been proposed which rely on a deterministic partition of the state space.[9–11] Another approach to approximating the state PDF is based on a stochastic partition of the state space and falls under the category of sequential Monte Carlo methods (sometimes referred to as *particle filters*[1, 12, 13]). Suppose we have the following approximation of the *propagated* target state PDF

$$p(\mathbf{x}_k|\mathcal{E}^{k-1}) \approx \sum_{i=1}^{N_s} \gamma_{k-1}^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \tag{20}$$

where $\gamma_{k-1}^{(i)}$ represents the mass density associated with the i^{th} point in the state space, $\mathbf{x}_k^{(i)}$, and $\sum_{i=1}^{N_s} \gamma_{k-1}^{(i)} = 1$. Substituting this approximation into the formula for the updated target state PDF in (15), results in

$$p(\mathbf{x}_k|\mathcal{E}^{k-1}, \neg\eta_k) \approx \sum_{i=1}^{N} \gamma_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \tag{21}$$

where the new (unnormalized) weights are given by

$$\begin{aligned} \gamma_k^{(i)} \quad &\propto \quad \Big[P(\mathbf{r}_k \notin \mathcal{D}_k|\mathbf{x}_k^{(i)}) + P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T \mathbf{u}_{los}|| < v_{min}|\mathbf{x}_k^{(i)}) \\ &+ \quad (1 - P_D) \cdot P(\mathbf{r}_k \in \mathcal{D}_k, ||\mathbf{v}_k^T \mathbf{u}_{los}|| > v_{min}|\mathbf{x}_k^{(i)})\Big] \, \gamma_{k-1}^{(i)} \end{aligned} \tag{22}$$

Thus, updating the target state PDF with an empty dwell under a pointwise approximation of the target state PDF is simply a matter of determining whether each point from a finite collection of points falls within particular regions in the state space.

4. RESULTS

A particle filtering algorithm was implemented in C++ for evaluation using a simulation developed at Toyon Research known as SLAMEM™ . With extensive modeling of terrain (DTED), roads, sensors and vehicles (GVS™), SLAMEM™ is an excellent tool for evaluating algorithms. The current particle filtering algorithm was enhanced by incorporating the equations described in this paper so that targets can be tracked through move-stop-move cycles. The algorithm was tested in SLAMEM™ and initially evaluated for feasibility and potential strengths. A vehicle was tracked for one hour using one GMTI radar ($\sigma_\theta = 0.0005^R$, $\sigma_r = 0.289\ m$, $\sigma_{\dot{r}} = 0.289\ m/s$, $v_{min} = 1.0\ m/s$ and the radar revisit was around 10 seconds). The vehicle went through random move-stop cycles. In Figure 1 is a plot of the probability that the vehicle was stopped versus time along with the normalized vehicle's speed versus time during the first 1000 seconds of the simulation. Clearly, the algorithm recognizes when the vehicle stops and continues to track the vehicle once it begins moving again. Shown in Figure 2 is a plot of the position and velocity estimation errors versus time during the first 1000 seconds of the simulation.

5. CONCLUSIONS AND FUTURE WORK

The proposed algorithm shows great promise in terms of being able to track ground targets through move-stop-move cycles. By actually *updating* the target state PDF, we can determine an estimate of the target's stopped location that could be used to schedule other sensor measurements of the target (e.g., optical or SAR). We derived a Bayesian update formula for incorporating new *evidence*, expanding the usual approach which

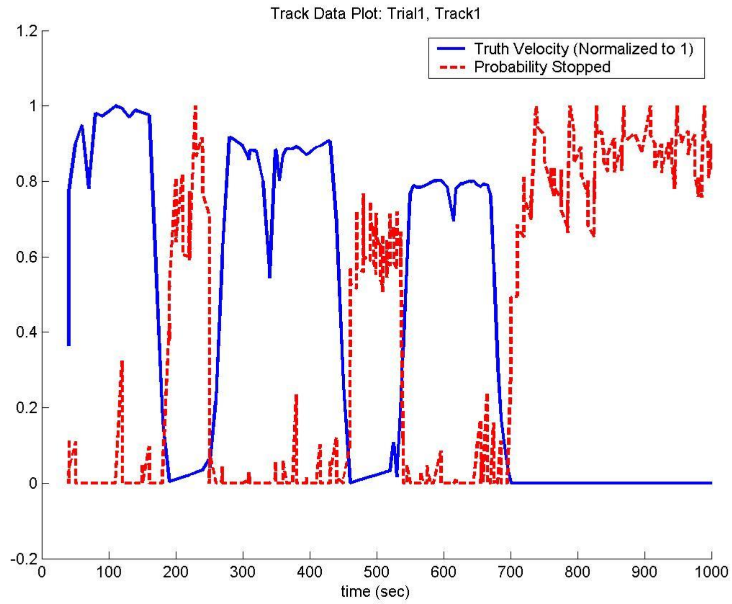

Figure 1. Probability that the vehicle is stopped and the vehicle's (normalized) speed versus time.

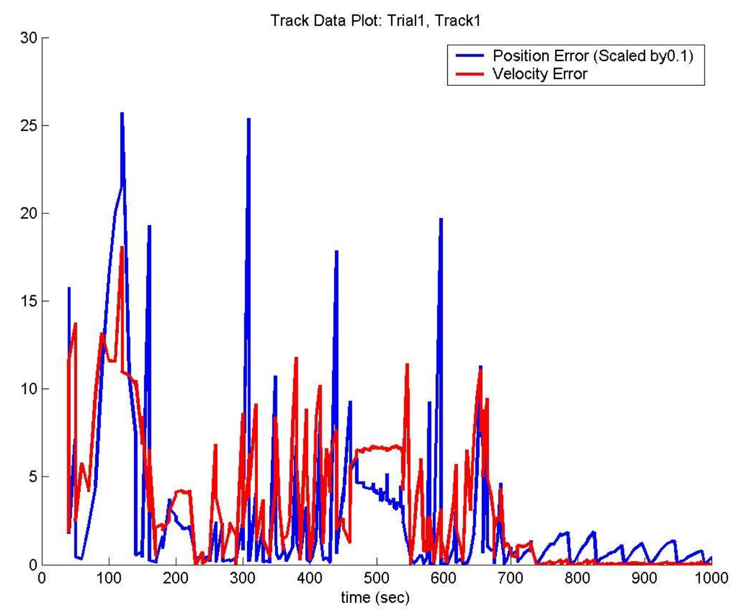

Figure 2. Plot of the position and velocity estimation errors versus time.

utilizes new detections from a radar or other sensor. Using a particle filter and a *point-wise* approximation of the propagated target state PDF, the update becomes straightforward to implement. In fact, while the evidence used here was a *non-detection event*, the mathematics supports the use of any form of evidence to update the target state PDF and is straightforward to implement using a particle filter.

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 and interest.

REFERENCES

1. 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.
2. B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, Boston, MA, 2004.
3. N. Gordon and B. Ristic, "Tracking airborne targets occasionally hidden in the blind Doppler," *Digital Signal Processing* **12**, pp. 383–393, April/July 2002.
4. 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.
5. 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.
6. 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.
7. 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.
8. S. Kramer and H. Sorenson, "Recursive Bayesian estimation using piece-wise constant approximations," *Automatica* **24**(6), pp. 789–801, 1988.
9. 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.
10. R. Bucy and K. Senne, "Digital synthesis of non-linear filters," *Automatica* **7**, pp. 287–298, 1971.
11. H. Sorenson and D. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica* **7**, pp. 465–479, 1971.
12. 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.
13. 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.
14. C. Agate and K. Sullivan, "Particle filtering algorithm for tracking multiple road-constrained targets," in *Proceedings of SPIE Conference on signal processing, sensor fusion and target recognition XII*, **5096**, pp. 256–266, (Orlando, FL), April 2003.
15. 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* **AES-39**, pp. 1098–1103, July 2003.