

A Comparison of Tracking Algorithms for Supermaneuverable Targets

Chris Kreucher, Kristine Bell *Fellow, IEEE*, and David Sobota

Abstract — Modern aircraft are capable of maneuvers exceeding those possible by purely aerodynamic design. This capability, called supermaneuverability, includes rapid changes in acceleration and high-G turns that are not feasible from traditional aircraft. Furthermore, newer aircraft often have a low radar cross-section (RCS) profile and/or RCS which varies rapidly with look angle. This paper summarizes the results of a performance evaluation of several conventional and emerging tracking approaches for supermaneuverable targets. The algorithms have been evaluated with respect to target maneuverability, along the continuum of mild maneuvers to supermaneuvers and as a function of target RCS. From this analysis, we draw the following broad conclusions: For high-SNR stable RCS targets exhibiting modest maneuvering, conventional Kalman filter (KF) trackers work well and are computationally appealing. For modern targets that exhibit a low RCS profile, high scintillation, and/or high maneuverability, KF approaches fail and more sophisticated approaches are required. While particle filter (PF) trackers are more computationally demanding than the KF, they provide a tracking capability not achievable by KF methods. This paper provides several case studies that backup these conclusions.

Index Terms—Tracking, Radar, Nonlinear Filtering, Supermaneuverable Targets, Particle Filtering, Kalman Filtering

I. INTRODUCTION

MODERN aircraft are capable of supermaneuverability, a term that includes actions such as extremely tight turns, and rapid changes in acceleration. Examples of such maneuvers include the Herbst supermaneuver [1] and the spiral and loaded turn maneuvers [2], as illustrated in Figure 1 and Figure 2. In addition, newer aircraft often have a low radar cross-section (RCS) profile and/or have an RCS which varies rapidly with look angle.

These traits pose problems to conventional Kalman filter (KF)-based trackers, which assume linearized models with additive Gaussian noise processes and require data to be processed to provide “detections.” Consequently, systems that use these trackers will break lock. In contrast, state-of-the-art tracking approaches such as the particle filter (PF) allow nonlinear state and measurement models, do not require additive Gaussian noise processes, and can use track-before-

detect (TBD) measurement models to achieve significantly improved performance.

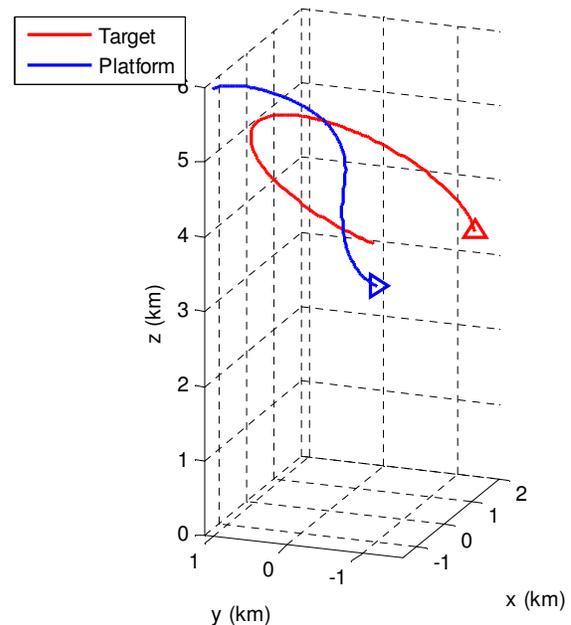


Figure 1. Simulation of a loaded turn maneuver.

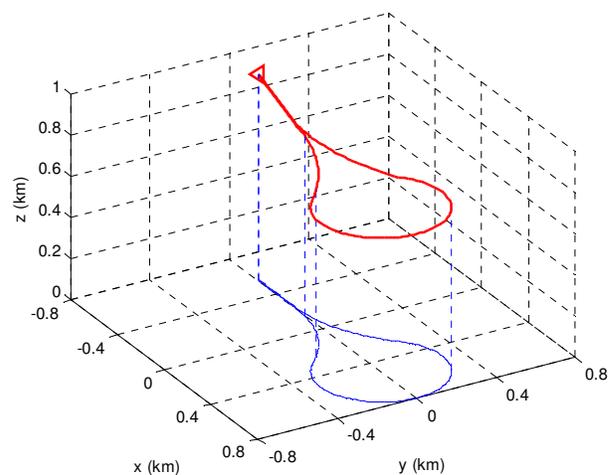


Figure 2. Simulation of a Herbst supermaneuver.

This paper describes an evaluation of both conventional and modern tracking approaches with supermaneuverable targets. Similar comparisons [17][4] have been made on other model problems in the past. We focus on single-target track

This work was supported by Air Force contract FA8650-14-M-1774.

Chris Kreucher is with Integrity Applications Incorporated in Ann Arbor, MI 48108 (e-mail: ckreuche@umich.edu).

Kristine Bell is a Senior Scientist with Metron Scientific Solutions, Reston VA 20190 (e-mail: bell@metsci.com).

maintenance, which allows us to highlight the aspects unique to supermaneuverable targets, rather than challenges associated with multiple targets and track initialization. The algorithms are scored in terms of tracking error and computational requirements. The trackers include the extended Kalman filter (EKF), the unscented Kalman filter (UKF), the standard PF with resampling (PFR), and PF with homotopy flow (PFH). Our simulations include targets exhibiting supermaneuverability, low RCS, specular scintillation properties, and combinations of the preceding. Target measurements are made using a tracking Radar simulation.

A summary of our findings is as follows:

1. For high-SNR stable RCS targets exhibiting with modest maneuvers, KF-based trackers work well and are computationally appealing.
2. For modern low RCS, highly scintillating, maneuverable targets, KF approaches fail and more sophisticated tracking approaches are required. While PF trackers are more computationally demanding than the KF methods, they provide a capability not achievable by KF methods.
3. PFs succeed because they can more accurately model target motion uncertainty, as well as to more fully exploit target measurements through non-thresholded data.

This paper proceeds as follows. First, Section II describes supermaneuverable targets and our modeling approach. Next, in Section III, we describe our measurement simulation. Third, in Section IV, we describe the tracking algorithms evaluated in this work. Next, Section V summarizes results of the evaluation both in terms of compute requirements and tracking performance. Finally, Section VI concludes.

II. SUPERMANEUVERABLE TARGETS

Our approach to supermaneuverable target simulation and modeling is to combine conventional kinematic models piecewise. We do not restrict model parameters to values achievable with conventional aircraft. This section defines our notation and reviews our base models, the constant velocity (CV) and coordinated turn (CT) kinematic models.

Let \mathbf{x}_k represent the 3D state of a target at time t_k , i.e.,

$$\mathbf{x}_k \equiv [x_k \quad \dot{x}_k \quad y_k \quad \dot{y}_k \quad z_k \quad \dot{z}_k]^T \quad (1)$$

The CV and coordinated turn with known turn parameters (CTK) models are linear models of the form:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} \quad (2)$$

For the CV model, \mathbf{F}_{CV} is a concatenation of submatrices of the form $\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$. For a horizontal CT in the x - y plane with

known turn rate ω , we use the two-dimensional model given in [3]-[6], augmented with constant (zero) velocity motion in the z dimension. To simplify our treatment, we do not consider out-of-plane motion or planar motion in an arbitrary (i.e., not oriented in the x - y plane) here. The transition matrix thus has the form:

$$\mathbf{F}_{HCT}(\omega) = \begin{bmatrix} 1 & \frac{\sin(\omega\Delta t)}{\omega} & 0 & -\frac{1-\cos(\omega\Delta t)}{\omega} & 0 & 0 \\ 0 & \cos(\omega\Delta t) & 0 & -\sin(\omega\Delta t) & 0 & 0 \\ 0 & \frac{1-\cos(\omega\Delta t)}{\omega} & 1 & \frac{\sin(\omega\Delta t)}{\omega} & 0 & 0 \\ 0 & \sin(\omega\Delta t) & 0 & \cos(\omega\Delta t) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

In the limit $\omega \rightarrow 0$, this horizontal CTK model reduces to the CV model. This horizontal plane CTK can be extended to arbitrary planar motion by specifying two additional parameters that define the plane of motion.

In the motion models used by the tracking algorithms, we include process noise, giving the nearly constant velocity (NCV) and nearly CTK models:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_k, \quad (4)$$

where \mathbf{v}_k is the process noise. If we assume the process noise $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$, the transition density is

$$q(\mathbf{x}_k | \mathbf{x}_{k-1}) = \frac{1}{\sqrt{|2\pi\mathbf{Q}_k|}} \exp\left\{-\frac{1}{2}[\mathbf{x}_k - \mathbf{F}\mathbf{x}_{k-1}]^T \mathbf{Q}_k^{-1} [\mathbf{x}_k - \mathbf{F}\mathbf{x}_{k-1}]\right\}. \quad (5)$$

We use the continuous white noise acceleration (CWNA) model [4] for the process noise, with independent accelerations along each of the axes. The process noise covariance matrix has the form:

$$\mathbf{Q} = \sigma_q^2 \begin{bmatrix} \frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 & 0 & 0 & 0 & 0 \\ \frac{1}{2}\Delta t^2 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 & 0 & 0 \\ 0 & 0 & \frac{1}{2}\Delta t^2 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ 0 & 0 & 0 & 0 & \frac{1}{2}\Delta t^2 & \Delta t \end{bmatrix} \quad (6)$$

To model supermaneuverable targets, we will combine these primitives with appropriate parameters. The combinations are described in Section V.

III. RADAR MODEL

Our analysis of supermaneuverable target tracking uses simulated radar measurements to drive the trackers. This section briefly describes our radar measurement model. We assume a pulse-Doppler tracking radar which is manipulated with Fourier processing to create a range-Doppler map (RDM). We further assume the phased array radar has engaged tracking mode and steers in azimuth and elevation angle so as to approximately follow the target. The tracking function is performed by pulsing at the target and processing the returns to exploit time-of-flight and Doppler shift to yield information about target range and range rate (radial velocity).

For the purposes of analysis, we simulate an X-band radar [14] with the parameters shown in Table 1.

Table 1: Modeled Radar Parameters

| Parameter | Notation | Value |
|----------------------------|------------------|---------------|
| Center frequency | f_c | 8 GHz |
| Wavelength | l | 3.75 cm |
| IF bandwidth | BW | 5 MHz |
| Pulse repetition frequency | PRF | 20 KHz |
| Number of pulses | N_p | 56 |
| Coherent Pulse Interval | CPI | 0.0028 s |
| Range Resolution | ΔR | 30 m |
| Range Rate Resolution | $\Delta \dot{R}$ | 6.7 m/s |
| Range Ambiguity | R_{amb} | 7500 m |
| Range Rate Ambiguity | \dot{R}_{amb} | ± 188 m/s |
| Beamwidth (azimuth) | $\Delta \phi$ | 60° |
| Beamwidth (elevation) | $\Delta \theta$ | 60° |
| Transmit Power | P_t | 4.7 kW |
| Noise temperature | T_0 | 300°K |
| Noise figure | NF | 2 |

Denote the state of the radar platform as

$$\mathbf{x}_{p,k} \equiv \begin{bmatrix} x_{p,k} & \dot{x}_{p,k} & y_{p,k} & \dot{y}_{p,k} & z_{p,k} & \dot{z}_{p,k} \end{bmatrix}^T \quad (7)$$

The target's state relative to the platform state is defined as:

$$\mathbf{x}_{r,k} = \mathbf{x}_k - \mathbf{x}_{p,k}. \quad (8)$$

The range and range rate to the platform are given by:

$$R_k = \sqrt{x_{r,k}^2 + y_{r,k}^2 + z_{r,k}^2} \quad \text{and} \quad \dot{R}_k = \frac{x_{r,k}\dot{x}_{r,k} + y_{r,k}\dot{y}_{r,k} + z_{r,k}\dot{z}_{r,k}}{\sqrt{x_{r,k}^2 + y_{r,k}^2 + z_{r,k}^2}}, \quad (9)$$

and the target's azimuth and elevation angles are specified as:

$$\phi_k = \arctan\left(\frac{y_{r,k}}{x_{r,k}}\right) \quad \text{and} \quad \theta_k = \arctan\left(\frac{z_{r,k}}{\sqrt{x_{r,k}^2 + y_{r,k}^2}}\right). \quad (10)$$

A. Range-Doppler Maps

The radar output will be modeled as an $N_R \times N_{\dot{R}}$ RDM such as that shown in Figure 3.

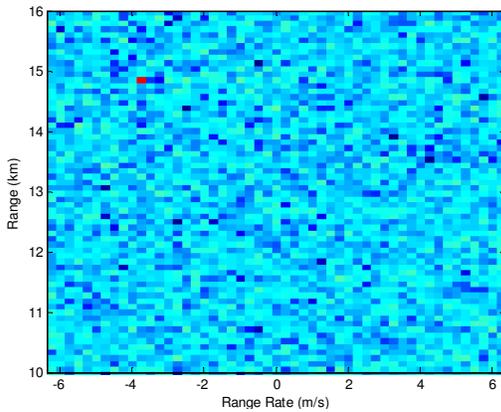


Figure 3: A Range Doppler Map

Each pixel in the RDM corresponds to a particular (ambiguous) range and range rate (Doppler) and is a measurement of either background (noise/clutter) or

background plus target. The RDM surface is characterized by its pixel spacing and extent, which are determined by the radar system parameters. The range and range-rate ambiguities (i.e., the extent of the surface in these directions) are given as [15]:

$$R_{amb} = \frac{c}{2PRF} \quad \text{and} \quad \dot{R}_{amb} = \frac{c PRF}{4f_c}. \quad (11)$$

The range rate ambiguity is two-sided. The range and range rate resolutions (i.e., pixel spacings) are given as

$$\Delta R = \frac{c}{2BW} \quad \text{and} \quad \Delta \dot{R} = \frac{c}{2f_c CPI} \quad (12)$$

where CPI captures the number of pulses integrated,

$$CPI = \frac{N_p}{PRF} \quad (13)$$

Let $\lceil \cdot \rceil$ denote rounding up to the nearest integer. The RDM surface then has N_R range, spaced ΔR apart, where

$$N_R = \left\lceil \frac{R_{amb}}{\Delta R} \right\rceil = \left\lceil \frac{BW}{PRF} \right\rceil \quad (14)$$

and $N_{\dot{R}}$ pixels in range rate, spaced $\Delta \dot{R}$ apart, where

$$N_{\dot{R}} = \left\lceil \frac{2\dot{R}_{amb}}{\Delta \dot{R}} \right\rceil = \lceil PRF \cdot CPI \rceil = N_p \quad (15)$$

The target is assumed small relative to the range spacing, thus the target is restricted to one pixel. Pixel statistics will depend on whether the pixel is a target-containing or not. In pixels that do not contain a target, we assume thermal noise is the dominant factor entering the radar front-end. We do not consider clutter targets. The statistics are modeled as Rayleigh. The PDF of the noise-only pixels is then given by:

$$p_N(z) = \frac{z}{\eta} \exp\left\{-\frac{z^2}{2\eta}\right\} \quad (16)$$

where the noise power η is the product of Boltzmann's constant, the system noise temperature, the receiver noise bandwidth, and the noise figure,

$$\eta = kT_0 \cdot BW \cdot NF \quad (17)$$

The PDF of the signal plus noise at the output is Rician [15], i.e., the PDF of the target-containing pixels is given by:

$$p_S(z) = \frac{z}{\eta} \exp\left\{-\frac{z^2 + A^2}{2\eta}\right\} I_0\left(\frac{zA}{\eta}\right) \quad (18)$$

Where the amplitude of the pulse echo, A , is computed using standard radar modeling. For a pulse transmitted with power P_t , the received power is

$$P_{r,k} = \frac{P_t G(\delta\phi_k, \delta\theta_k)^2 \lambda^2 \sigma_k}{(4\pi)^3 R_k^4}, \quad (19)$$

where $G(\delta\phi_k, \delta\theta_k)$ is the antenna gain in the direction of the target and σ_k is the target RCS. Therefore, the amplitude of the received echo is given by:

$$A_k = \frac{\sqrt{P_t} G(\delta\phi_k, \delta\theta_k) \lambda \sqrt{\sigma_k}}{(4\pi)^{3/2} R_k^2} \quad (20)$$

The antenna gain is specified as follows. For a radar pointing at (ϕ_p, θ_p) , the difference between the pointing and target angles is $\delta\phi_k = \phi_p - \phi_k$ and $\delta\theta_k = \theta_p - \theta_k$. For a rectangular $L_x \times L_y$ phased array radar, the beam pattern is given by [15]:

$$G(\delta\phi_k, \delta\theta_k) = \left| \frac{4\pi L_x L_y}{\lambda^2} \text{sinc}\left(\frac{L_x}{\lambda} \sin \delta\theta_k \cos \delta\phi_k\right) \text{sinc}\left(\frac{L_y}{\lambda} \sin \delta\theta_k \sin \delta\phi_k\right) \right| \quad (21)$$

The target RCS is specified as follows. For the non-fluctuating case (referred to as Swerling 0), $\sigma_k = \sigma_{av}$ is a constant. For targets with signatures that fluctuate, σ_k is a random variable with its own probability distribution. The four cases used in the literature are called the Swerling models. In this analysis, we considered the Swerling I model, in which the RCS varies from scan-to-scan according to an Exponential distribution [15]:

$$p(\sigma_k) = \frac{1}{\sigma_{av}} \exp\left\{-\frac{\sigma_k}{\sigma_{av}}\right\} \quad (22)$$

The RDM pixels are used in one of two ways by the filters: Either the RDM is thresholded and the detections are passed to the filter, or the raw (non-thresholded) RDM is used directly.

B. Detection Measurements

The ‘‘detect-and-then-track’’ method first distills the RDM surface to a small set of threshold exceedances (detections). The trackers then use these detections in the measurement update. All KF-based trackers require this operation. The PF trackers can use this but are not required to. The threshold determines the probability of detection, P_D , and the probability of false alarm, P_{FA} .

The detect-and-then-track method does not fully exploit the measurements because of the hard decisions. At low SNR this will lead to tracker failure through the following mechanism. In order to achieve a low false alarm rate, the detection threshold must be high. In low SNR, this leads to a low detection probability. As such, trackers become data starved and fail. However, when SNR is high and the target is not rapidly maneuvering, detect-and-then-track algorithms are not measurement starved and thus succeed in a computationally attractive manner.

Assuming we have correctly identified the target pixel, we then have a coarse measurement of range and range rate. Implicitly, we also have a measurement of azimuth and elevation because the target is on the RDM. This implies that we are in the main beam of the radar in those dimensions. The measurements (modulo ambiguities) are related to the target state by

$$\mathbf{z}_k = \begin{bmatrix} \hat{R}_k & \dot{\hat{R}}_k & \hat{\phi}_k & \hat{\theta}_k \end{bmatrix}^T = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k \quad (23)$$

where

$$\mathbf{h}(\mathbf{x}_k) = \begin{bmatrix} R_k & \dot{R}_k & \phi_k & \theta_k \end{bmatrix}^T. \quad (24)$$

The measurement error will be modeled as a zero-mean Gaussian random vector $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{R}_k)$, thus the likelihood function of the thresholded measurements is

$$L_T(\mathbf{z}_k | \mathbf{x}_k) = \frac{1}{\sqrt{|2\pi\mathbf{R}_k|}} \exp\left\{-\frac{1}{2}[\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k)]^T \mathbf{R}_k^{-1} [\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k)]\right\}$$

We assume the covariance matrix has the form:

$$\mathbf{R}_k = \begin{bmatrix} \Delta R^2/12 & 0 & 0 & 0 \\ 0 & \Delta \dot{R}^2/12 & 0 & 0 \\ 0 & 0 & \Delta \phi^2/12 & 0 \\ 0 & 0 & 0 & \Delta \theta^2/12 \end{bmatrix} \quad (25)$$

To use this nonlinear measurement model in the EKF, we require the 4x6 Jacobian, which has the form:

$$\tilde{\mathbf{H}}(\mathbf{x}) = [\nabla_{\mathbf{x}} \mathbf{h}^T(\mathbf{x})]^T = \begin{bmatrix} \frac{x_r}{R} & 0 & \frac{y_r}{R} & 0 & \frac{z_r}{R} & 0 \\ \frac{\dot{x}_r}{R} - \frac{x_r \dot{R}}{R^2} & \frac{x_r}{R} & \frac{\dot{y}_r}{R} - \frac{y_r \dot{R}}{R^2} & \frac{y_r}{R} & \frac{\dot{z}_r}{R} - \frac{z_r \dot{R}}{R^2} & \frac{z_r}{R} \\ -\frac{y_r}{R_G} & 0 & \frac{x_r}{R_G} & 0 & 0 & 0 \\ -\frac{x_r z_r}{R^2 R_G} & 0 & -\frac{x_r z_r}{R^2 R_G} & 0 & \frac{R_G}{R^2} & 0 \end{bmatrix},$$

C. Track-before Detect Measurements

The TBD method instead uses all of the pixels in the RDM as a means of updating the tracker. Define $z_k(i,j)$ as the ij th RDM pixel and \mathbf{z}_k as the collection of the $N_R N_{\dot{R}}$ RDM pixels. The update is done by computing the probability of \mathbf{z}_k conditioned on the target state \mathbf{x}_k , i.e.,

$$L_N(\mathbf{z}_k | \mathbf{x}_k, A_k) = \prod_{i=1}^{N_R} \prod_{j=1}^{N_{\dot{R}}} p(z_k(i,j) | \mathbf{x}_k, A_k) \propto \frac{p_S(z_k(i_{R_k}, j_{\dot{R}_k}) | A_k)}{p_N(z_k(i_{R_k}, j_{\dot{R}_k}))}$$

Let pixel $(i_{R_k}, j_{\dot{R}_k})$ correspond to x_k . For the Rician and Rayleigh models here, we find

$$L_N(\mathbf{z}_k | \mathbf{x}_k, A_k) \propto \exp\left(\frac{-A_k^2}{2\eta}\right) I_0\left(\frac{z_k(i_{R_k}, j_{\dot{R}_k}) A_k}{\eta}\right). \quad (26)$$

In contrast to the detect-and-then-track approach, here all the measurements are used, not just the threshold exceedances. In addition, the raw measurement value is used, not just the fact that it has exceeded the threshold.

IV. TRACKING ALGORITHMS

This section describes several tracking algorithms and their application to supermaneuverable targets. These include the standard linear KF, the EKF, UKF and the fully nonlinear PFR and PFH trackers. For the detection filters, we use

probabilistic data association (PDA) to handle missed detections and false alarms.

A. Kalman Filter

The KF assumes linear AWGN motion and thresholded measurements. The models are of the form:

$$\begin{aligned}\mathbf{x}_k &= \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_k \\ \mathbf{z}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k,\end{aligned}\quad (27)$$

where $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$ and $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{R}_k)$. The initial state distribution is $\mathbf{x}_0 \sim N(\boldsymbol{\mu}_0, \mathbf{P}_0)$. The KF iterates through the well-known two step motion and measurement update [3].

B. Extended Kalman Filter

The EKF is a KF-based tracking algorithm that uses first-order analytical linearization. The EKF assumes nonlinear AWGN motion and measurement models of the form:

$$\begin{aligned}\mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{v}_k \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k,\end{aligned}\quad (28)$$

where $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$ and $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{R}_k)$. The EKF linearizes \mathbf{f} and \mathbf{h} about x at each time step as

$$\tilde{\mathbf{F}}_k \equiv \left[\nabla_{\mathbf{x}} \mathbf{f}^T(\mathbf{x}) \right]_{\mathbf{x}=\boldsymbol{\mu}_{k-1}^+}^T \quad (29)$$

And

$$\tilde{\mathbf{H}}_k \equiv \left[\nabla_{\mathbf{x}} \mathbf{h}^T(\mathbf{x}) \right]_{\mathbf{x}=\boldsymbol{\mu}_k^-}^T \quad (30)$$

and then uses the same two-step KF iteration [3].

C. The Unscented Kalman Filter

The UKF is a KF-based tracking algorithm that uses a small set of samples to compute second order statistics (means and covariance matrices) used in the KF. It approximates the (presumed) Gaussian predicted and posterior densities by a set of deterministically chosen sample points that preserve the mean and covariance matrix through nonlinear transformations (up to second order). It does not require evaluation or computation of Jacobians. Full details of the UKF can be found in [5].

D. Resampling Particle Filter

In a PF [4], the distribution is represented by a set of N particles and weights $\{x^n, w^n\}$. In the motion update step, particles are propagated according to the motion model, while the weights remain fixed. In the measurement update step, the particles remain fixed while the weights are updated by Bayes' rule, which involves multiplying the weights by the likelihood function and normalizing.

Particle degeneracy occurs when the likelihood function is concentrated and the particle weights are mostly zeros [7]. The standard remedy is resampling, in which the particles with the largest weights are replicated in proportion to their weights, and the weights of the resampled particles are all set to be equal to $1/N$. The resampling PF can use either detections or the unthresholded RDM pixels as measurements.

We report on both versions, which we refer to as particle filter resampling thresholded (PFRT) and particle filter resampling non-thresholded (PFRN).

E. Homotopy Flow Particle Filter

The resampling PF is a computationally intensive process that incurs performance degradation in many practical problems. The homotopy particle flow method has been developed to implement the information update in a particle filter in an entirely different manner [7]-[13]. In the homotopy method, the weights remain fixed, while the particles themselves "flow" to the appropriate regions of the target state space that provide a good representation of the posterior. Several versions of the particle flow method have been developed in [7]-[13] and related papers, and remarkable performance and computational gains over resampling filters have been reported.

Closed form solutions for the particle flow function exist only in a few special cases (e.g. detection measurements with a linear Gaussian measurement model). Practical implementations based on the linear Gaussian flow function for nonlinear models have been recently reported [7]-[13]. We study the PFH with detection measurements and a nonlinear Gaussian measurement model. The implementation is based on [13] for linear Gaussian measurements, but using second order statistics computed from the particles.

F. Measurement Association

In the detect-and-then-track algorithm descriptions above, we assumed that the detection threshold was fixed and there was exactly one detection measurement available to the tracker. In practice there may be missed detections and/or false detections due to noise and clutter, and there needs to be some validation and association of measurements to the target. In this paper we use PDA [3] to perform this function. More sophisticated approaches which vary the threshold and/or provide amplitude information could also be used [3][16].

G. Interacting Multiple Model (IMM) Algorithms

IMM algorithms assume that the target follows a jump Markov model in which the target obeys one of M motion models, or "modes." Let M_k denote the mode at time t_k . The target is modeled to switch between modes with transition probabilities defined as:

$$\pi_{ij} \equiv \Pr\{M_k^j | M_{k-1}^i\} = \Pr\{M_k = j | M_{k-1} = i\} \quad (31)$$

In the IMM approach, a separate filter is maintained for each mode and the mode probability is tracked along with the kinematic state. The posterior density is expressed as a weighted sum of mode-conditioned posterior densities

$$p^+(\mathbf{x}_k) = \sum_{j=1}^M p^+(\mathbf{x}_k | M_k^j) \eta_k^j \quad (32)$$

where the posterior mode probability η_k^j is defined as:

$$\eta_k^j \equiv \Pr\{M_k^j | \mathbf{Z}_k\} \quad (33)$$

V. RESULTS

This section describes our evaluation of the algorithms outlined above in terms of tracking ability and computational requirements. The simulations include targets that exhibit supermaneuverability, low RCS, and specular scintillation properties. We summarize the factors we investigated here.

Tracking Approaches. We studied the EKF, UKF, PFH, PFRT, and PFRN. The PF algorithms can be adjusted to trade performance for computation and we have analyzed that trade.

Motion Models. We studied both the NCV model and the IMM model with an NCV mode and two CTK modes. Each of these models are coupled with each of the tracker types.

Target Maneuvers. We studied several types of target maneuvers. We present results for a maneuver which consists of straight line motion followed by a constant turn-rate maneuver followed again by straight line motion. The turn rate is varied from modest to supermaneuverable.

A. The Baseline Scenario

The baseline scenario shown in Figure 4. It consists of a linear motion segment for $k = 1$ to 25, a horizontal plane CT maneuver for $k = 26$ to 125, and another linear motion segment for $k = 126$ to 200.

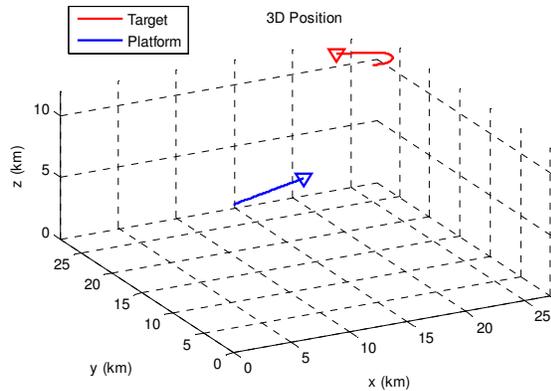


Figure 4. Scenario 1: Baseline Maneuver by Target

B. Computation Time

The algorithms are implemented in MatLab using semi-optimized code. As such, the timing is inherently susceptible to over-estimation due to imperfect implementation. We have taken care to determine the main sources of time using MatLab's profiler and eliminate any of those that are obviously optimizable. In all cases, the main computational drivers that remain after our optimization effort appear to be implemented as efficiently as possible.

There are two basic ways to empirically measure computational complexity in MatLab : either simply using the execution time, or using MatLab's internal CPU time. Both of these measures are imperfect gauges of algorithm complexity. Total execution time can be heavily influenced by how parallelized the algorithm is, and whether MatLab can recognize opportunities for parallelization. On the other hand, CPU time does not account for non-CPU activities (e.g., data manipulation and indexing) that are important to gauge an algorithm's complexity. Fortunately, in this analysis we found

that both metrics told the same story qualitatively about the relative requirements of the different trackers. As such, we settled on using execution time as a gauge.

EKF. The EKF is based on estimating the mean and covariance. It is implemented entirely as a set of (small) matrix manipulations. As such, it the fastest and least memory intensive algorithm we studied. The EKF algorithm runs in real-time even with high-rate radar measurements.

UKF. The UKF algorithm is also based on estimating the mean and covariance of the target state. It differs from the EKF by its use of sigma points, which are an empirical approach to linearization. Like the EKF, the UKF is mainly a set of (small) matrix manipulations. The UKF algorithm takes about 3x as long as the EKF. This is related to the number of sigma-points in the implementation.

PFs. PF algorithms use discrete samples (particles) to represent the target probability density. As such, they can be significantly more computationally complex than KF-based algorithms. PF algorithms involve a series of computations that are not just matrix manipulations like the KF-based algorithms. Care has been taken to ensure that the algorithms are implemented efficiently so that we have provided an accurate estimate of their computational complexity. Tracking performance improves with the number of particles.

C. Tracker Benchmarking, NCV Model

With this as background, Figure 5 shows curves giving the tracking performance versus compute time for a high-SNR scenario using the NC tracking model. The PF algorithms are characterized by running with varying numbers of particles. The compute time is normalized to the EKF (i.e., EKF compute time = 1.0). The EKF performance is shown by the black line and the UKF is given by the green line. We find that the PFH algorithm has significant computation advantages over PFRT and PFRN algorithms. These advantages come directly from its efficient placement of particles which allows it to avoid the resampling step.

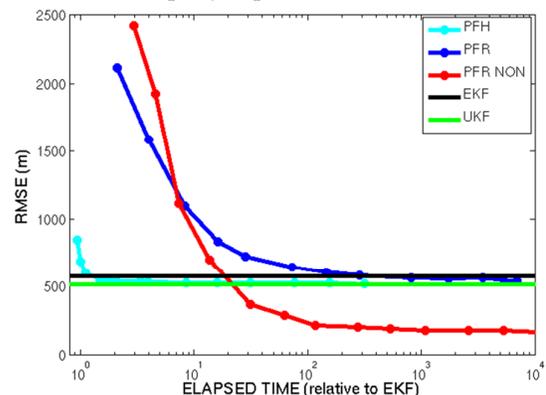


Figure 5. RMSE v compute time for the tracking algorithms.

Figure 5 illustrates several attributes of the filters:

- In this high-SNR, modest maneuverability scenario all of the detection-based filters (EKF, UKF, PFH, and PFRT) achieve roughly the same performance.

- The PFH algorithm has computational requirements only 2-3x more than the KF algorithms.
- The PFRN and PFRT algorithms have computational requirements 2 to 3 orders of magnitude higher than the KF algorithms.
- The PFRN is able to achieve better performance because it exploits the measurements more effectively.
- At high SNR/low maneuverability, all trackers work well
- At low SNR/low maneuverability and high SNR/high maneuverability the detection-based filters degrade significantly but still provide some tracking performance. The non-thresholded filter continues to perform well.
- At low SNR/high maneuverability all of the detection-based filters fail catastrophically while the non-thresholded filter continues to perform well.

As a further analysis, Figure 6 shows the performance as the maneuverability of the target is varied from modest (1.6 degrees/second) to extreme (16 degrees/second) and the SNR is varied from high (7 dB) to low (3 dB). This figure illustrates the following:

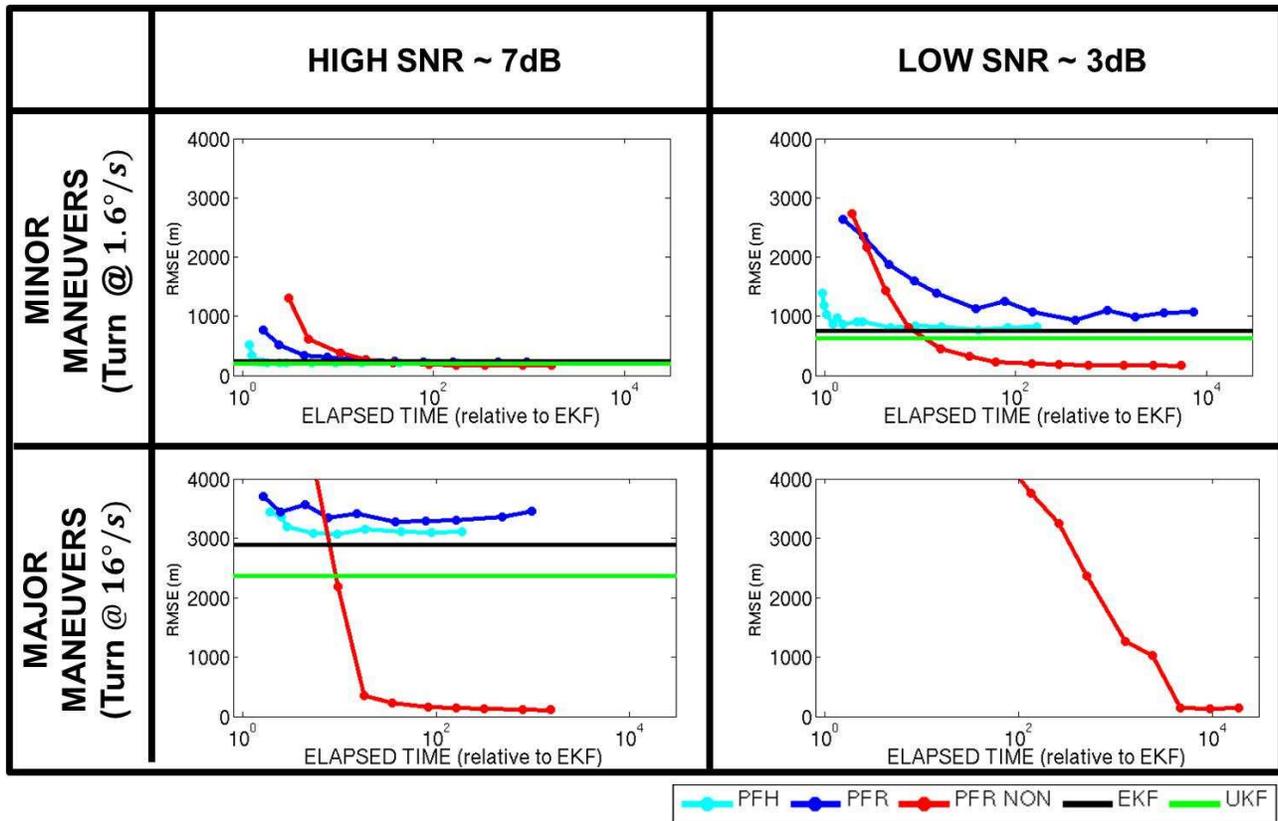


Figure 6. Algorithm Performance vs. SNR and Maneuverability

D. Tracker Benchmarking, IMM Model

In addition, we benchmarked the trackers using the IMM algorithm with an NCV mode ($\omega=0$) and two CTK modes with fixed known turn rates, $\omega = \pm 16$. The IMM algorithm calculates the probability of each mode. Figure 7 shows typical mode estimates for one trial.

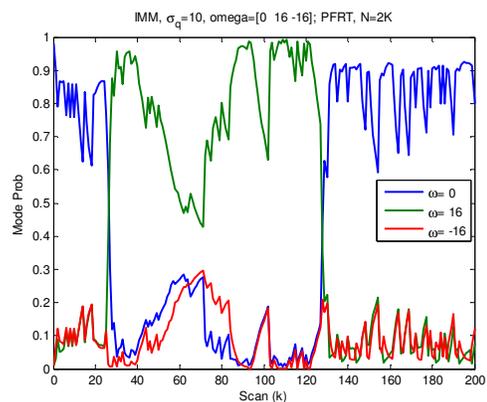


Figure 7. Typical Mode Probabilities for PFRT IMM Algorithm

Figure 8 shows the performance versus compute time for the IMM algorithm on a high SNR (left) and low SNR (right) scenario. At high SNR, all algorithms track well. At low SNR, the detection-based algorithms degrade severely while the non-thresholded particle filter continues to perform well.

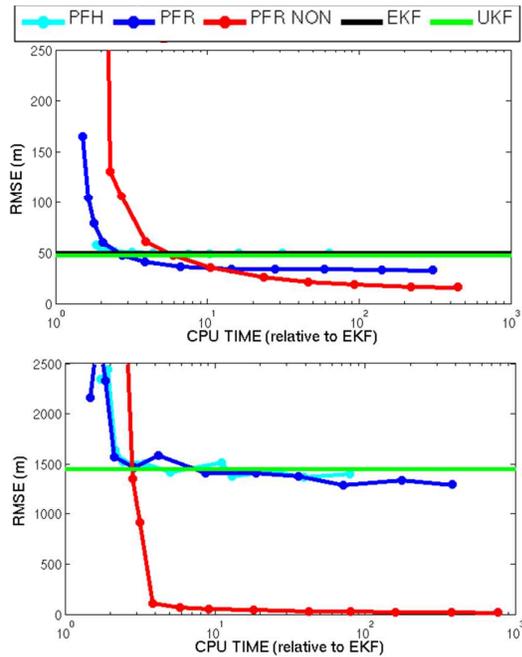


Figure 8. RMSE vs. Computation Time for IMM Algorithms. Top: High SNR. Bottom: Low SNR.

VI. CONCLUSIONS

This report has described a Phase I evaluation of tracking algorithms for a low RCS supermaneuverable target. Our major findings are as follows:

1. For high-SNR stable RCS targets with modest maneuvers, KF-based trackers work well and are computationally appealing.
2. For modern targets with low RCS, high scintillation, and/or high maneuverability, KF approaches fail and more sophisticated tracking approaches are required. While the PF trackers are significantly more computationally demanding than the KF methods, they provide a tracking capability not achievable by the KF methods.

3. The PF algorithms work because they are able to more accurately model the target motion uncertainty through non-Gaussian estimates of the target state distribution, as well as to more fully exploit target measurements through non-thresholded data.

REFERENCES

- [1] Repperger, D. W., "A study of supermaneuverable flight trajectories through motion field simulation of a centrifuge simulator," *Journal of Dynamic Systems, Measurement, and Control, Trans. of the ASME*, vol. 114, pp. 270-277, January 1992.
- [2] "NASA - Past Projects F-18 HARV", [Online]. Available at: <http://www.nasa.gov/centers/dryden/history/pastprojects/HARV/rd.html>.
- [3] Bar-Shalom, Y. and Li, X. R., **Multitarget-Multisensor Tracking: Principles and Techniques**, Storrs, CT: YBS Publishing, 1995.
- [4] Ristic, B., Arulampalam, S., and Gordon, N., **Beyond the Kalman Filter: Particle Filters for Tracking Applications**, Boston, MA: Artech House, 2004.
- [5] Julier, S., Uhlmann, J., and Durrant-White, H. F., "A new method for nonlinear transformation of means and covariances in filters and estimators", *IEEE Trans. Automatic Control*, vol. 45, no. 3, pp. 477-482, March 2000.
- [6] Li, X. R., and Jilkov, V. P., "Survey of maneuvering target tracking. Part I: Dynamic models," *IEEE Trans. Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333-1364, Oct. 2003.
- [7] Daum, F. and Huang, J., "Particle degeneracy: root cause and solution," *Proc. SPIE 8050, Signal Processing, Sensor Fusion, and Target Recognition XX*, 80500W, May 2011.
- [8] Daum, F. and Huang, J., "Nonlinear filters with log-homotopy," *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, San Diego, CA, USA, Sept. 2007.
- [9] Daum, F., Huang, J., and Noushin, A., "Exact particle flow for nonlinear filters," *Proc. SPIE 7697, Signal Processing, Sensor Fusion, and Target Recognition XIX*, 769704, April 2010.
- [10] Choi, S., Willett, P., Daum, F., and Huang, J., "Discussion and application of the homotopy filter," *Proc. SPIE 8050, Signal Processing, Sensor Fusion, and Target Recognition XX*, 805021, May 2011.
- [11] Ding, T. and Coates, M. J., "Implementation of the Daum-Huang exact-flow particle filter," *2012 IEEE Statistical Signal Processing Workshop*, Ann Arbor, MI, pp. 257-260, Aug. 2012.
- [12] Jilkov, V. P., Wu, J., and Chen, H., "Performance comparison of GPU-accelerated particle flow and particle filters," *2013 IEEE Conference on Information Fusion*, Istanbul, Turkey, pp. 1095-1102, July 2013.
- [13] Bell, K. L. and Stone, L. D., "Implementation of the homotopy particle filter in the JPDA and MAP-PF multi-target tracking algorithms," *2014 IEEE Conference on Information Fusion*, Salamanca, Spain, July 2014.
- [14] Streetley, M., ed. *Jane's Radar and Electronic Warfare Systems 2011-2012*, Jane's Information Group, 2012.
- [15] Skolnik, M. I., **Introduction to Radar Systems**, McGraw-Hill, 2003.
- [16] Willett, P., Niu, Ruixin, and Bar-Shalom, Y., "Integration of Bayes Detection with Target Tracking", *IEEE Transactions on Signal Processing*, vol. 49, no. 1, January 2001.
- [17] Davey, S. J., Rutten, M. G., Cheung, B., "A comparison of detection performance for several Track-Before-Detect algorithms", *IEEE Information Fusion Conference*, pp. 1-8, July 2008.