Maneuvering target tracking using an unbiased nearly constant heading model

P.A. Kountouriotis  
EEE Department  
Imperial College London  
SW7 2AZ, UK  
Email: pk201@imperial.ac.uk

Simon Maskell  
QinetiQ  
St. Andrews Road, Malvern  
Worcestershire, WR14 3PS, UK  
Email: srmaskell@QinetiQ.com

Abstract—This paper addresses the problem of modeling maneuvering target motion in tracking applications. Moving targets typically follow deterministic straight-line or curved trajectories, with minor deviations due to random disturbances. As a result, modeling target motion typically involves the derivation of state transition functions based on the laws of kinematics, with the addition of uncertainty terms in the form of random noise to compensate for model mismatch. Although it is possible to construct quite accurate models, there is a trade-off between model simplicity (and, thus, ease of implementation) and model accuracy. In this paper, we present a model for target motion that is based on a Brownian description of the target’s speed and heading, which allows the derivation of closed form expressions for the exact first two moments of the propagated probability density function of the target’s state vector. We outline the design of tracking algorithms based on this model, and demonstrate its effectiveness in dealing with maneuvering targets based on simulations.

I. INTRODUCTION

Modeling target motion typically involves the derivation of state transition functions based on physics, i.e. Newton’s laws of motion, with the addition of uncertainty terms in the form of Gaussian noise disturbances to cater for model mismatch. The most common kinematic models are the (nearly) constant velocity (CV) and (nearly) constant acceleration (CA) models, which, as their names suggest, describe the motion of a target moving along a straight path with constant velocity or under a constant accelerating force. These ‘nonmaneuver’ models assume that target motion is uncoupled across coordinates, and are very popular due to their simplicity ([5]). They rest on the principle that targets typically follow deterministic trajectories, with minor deviations due to random disturbances on the velocity or acceleration components respectively.

Other types of trajectories can also be considered, where, for example, a turn can be described as a circular, or semi-circular path in the state space. Models that describe such motion, in contrast to the previous ones, introduce coupling between the target’s motion coordinates, and provide a more ‘natural’, albeit more complicated, description of the target’s kinematics. The most representative example in this class of models is the coordinated turn (CT) model, which assumes that the target of interest moves with constant speed and constant turn rate. Again, deficiencies in the model are compensated by the addition of suitable process noise.

Several other models are also available, such as curvilinear and 3D motion models, but have not received as much attention as the aforementioned ones, as their nonlinearity requires the use of nonlinear filtering algorithms, which can pose difficult challenges to the practitioner. For more details of such models, the reader is referred to [12].

On the other hand, the CV and CA models, as well as CT models when the turn rate is a known constant, have the distinct advantage of being linear1. The linearity property of CV models, combined with the empirical fact that targets spend most of their time traveling along straight line segments, makes them highly desirable in tracker design, as they allow for the simple application of the Kalman filter. If, however, such a simple tracker is to be used for tracking a maneuvering target, the ‘inappropriateness’ of the model to describe more complicated trajectories has to be compensated for with the addition of a higher process noise intensity, at the expense of increased steady-state error along the linear segments of the trajectory [4].

Alternatively, and perhaps more appropriately, target maneuvers can be thought of as switches among a set of different target motion regimes. Better tracking performance can then be obtained via the use of multiple model algorithms, which are in essence banks of Kalman filters employing different state transition models, which switch among the various models based on the likelihood of each new measurement. Again, several methods have been proposed for this task, with the most popular being the generalized pseudo-Bayesian estimator (GPB) ([11], [7]) and the interacting multiple model estimator (IMM) ([3], [6], [13]). These algorithms make use of a set of models, typically a CV model for the linear parts of the trajectory and a CT model for the maneuvering part, or, perhaps more commonly, two (or more) CV models with varying levels of process noise intensity, to produce estimates of the target’s state vector that are more accurate than those obtained via simpler ‘single-model’ algorithms.

In this paper, we approach the problem of modeling target maneuvers from a slightly different perspective. Instead of considering Cartesian velocities, we work with the polar

1If the turn rate in the CT model is, however, unknown, the model becomes nonlinear and necessitates the use of approximations, usually via linearization techniques, in order to construct tracking algorithms.
where the measurements are available at discrete times, the proposed model. Simulation results are presented in Section II. Section III proceeds with headings model and the associated moment propagation equations (SDEs): where the corresponding Fokker-Planck partial differential equation was numerically solved on a grid, and filtering was performed via numerical grid-based methods which are, however, very computationally demanding. For related methods see also [15].

Instead of solving the Fokker-Planck equation corresponding to our model, we derive closed form expressions for the exact first two moments of the propagated probability density function of the state vector, starting with a Gaussian prior, which allows for the use of a Kalman filter-like algorithm to perform the tracking. Our model is unbiased in the sense of the work presented in [14]; namely, the formulae that we obtain for the predicted mean and covariance of the state vector involve appropriate multiplicative correction terms that cater for the implicit polar-to-Cartesian transformation of the velocity part of the state vector.

The paper is structured as follows. The (nearly) constant heading model and the associated moment propagation equations are described in Section II. Section III proceeds with an outline of four tracking algorithms that make use of the proposed model. Simulation results are presented in Section IV, followed by concluding remarks in Section V.

II. THE UNBIASED NEARLY CONSTANT HEADING (CH) MODEL

We consider a target, moving on a two-dimensional plane, whose speed $s_t$ and heading $\phi_t$ evolve according to (independent) Brownian motions. We define the state vector $x_t = [x_t, y_t, s_t, \phi_t]^T$ at time $t$ consisting of the Cartesian position, the speed and the heading of the target, which evolve according to the following system of stochastic differential equations (SDEs):

$$
\begin{align*}
\frac{dx_t}{dt} &= s_t \cos \phi_t dt \\
\frac{dy_t}{dt} &= s_t \sin \phi_t dt \\
\frac{ds_t}{dt} &= \sigma_s dB_t \\
\frac{d\phi_t}{dt} &= \sigma_\phi dW_t
\end{align*}
$$

(1)

In the above, $B_t$ and $W_t$ denote independent standard Brownian motions (i.e. $B_t \sim \mathcal{N}(0, t), W_t \sim \mathcal{N}(0, t)$).

In order to employ the above model in tracking applications, where the measurements are available at discrete times, the equations (1) have to be discretized. This is done by re-writing the SDEs in integral form as follows:

$$
\begin{align*}
x_{t+h} &= x_t + \int_t^{t+h} s_u \cos \phi_u du \\
y_{t+h} &= y_t + \int_t^{t+h} s_u \sin \phi_u du \\
s_{t+h} &= s_t + \sigma_s (B_{t+h} - B_t) \\
\phi_{t+h} &= \phi_t + \sigma_\phi (W_{t+h} - W_t),
\end{align*}
$$

(2)

where $h$ is the sampling interval. Note that, in the above set of equations, we assume that the state vector at the initial time $t$, $x_t$, is a random vector, distributed as $\mathcal{N}(\bar{x}_t, P_t)$.

According to the model of equations (2), and given the mean and covariance, $\bar{x}_t$ and $P_t$, of the state vector $x_t$ at the initial time $t$, it is a straightforward exercise to calculate the first two moments of $x_{t+h}$ at time $(t + h)$. For ease of notation, we will let $t = 0$ throughout the rest of this section.

Thus, given the initial mean $\bar{x}_0 = (E[x_0], E[y_0], E[s_0], E[\phi_0])^T$ and covariance $P_0$, the predicted mean, $\bar{x}_h$, and covariance, $P_h$, of $x_h$ can be calculated as follows:

Let

$$
I_1 = \frac{2}{\sigma_\phi} (1 - e^{-\sigma_\phi^2 h/2})
$$

(3)

$$
I_2 = \frac{2}{\sigma_\phi} (2 - 2e^{-\sigma_\phi^2 h/2} - e^{-\sigma_\phi^2 h/2} \sigma_\phi h)
$$

(4)

$$
J_1 = \frac{1}{3 \sigma_\phi^2} (3 - 4e^{-\sigma_\phi^2 h/2} + e^{-2\sigma_\phi^2 h})
$$

(5)

$$
J_2 = \frac{2}{\sigma_\phi} (-2 + 2e^{-\sigma_\phi^2 h/2} + \sigma_\phi^2 h)
$$

(6)

$$
K_1 = \frac{1}{18 \sigma_\phi^6} (9 - 16e^{-\sigma_\phi^2 h/2})
$$

$$
+ 7e^{-2\sigma_\phi^2 h} + 6e^{-2\sigma_\phi^2 h} \sigma_\phi^2 h
$$

(7)

$$
K_2 = \frac{1}{\sigma_\phi^6} (8 - 8e^{-\sigma_\phi^2 h/2} - 4\sigma_\phi^2 h + \sigma_\phi^4 h^2)
$$

(8)

Then, using equation (2), it can be shown that

$$
\hat{x}_h = \bar{x}_h + I_1 (E[s_0 \cos \phi_0], E[s_0 \sin \phi_0], 0, 0)^T
$$

and

$$
P_h = \begin{pmatrix} P_{h}^{11} & P_{h}^{12} \\ P_{h}^{21} & P_{h}^{22} \end{pmatrix}.
$$

(10)

The $P_{h}^{ij}$ in the above equation are $2 \times 2$ sub-matrices. In particular,

$$
P_{h}^{22} = \begin{pmatrix} \text{cov}(s_0) + \sigma_s^2 h & \text{cov}(s_0, \phi_0) \\ \text{cov}(s_0, \phi_0) & \text{cov}(\phi_0) + \sigma_\phi^2 h \end{pmatrix},
$$

(11)

while $P_{h}^{12}$ is given by

$$
P_{h}^{12} = \begin{pmatrix} E[x_h s_h] - E[x_h] E[s_h] & E[x_h \phi_h] - E[x_h] E[\phi_h] \\ E[y_h s_h] - E[y_h] E[s_h] & E[y_h \phi_h] - E[y_h] E[\phi_h] \end{pmatrix}
$$

(12)

The derivation of equations (9)-(14) is quite lengthy, and is based on fundamental properties of Brownian motion and multivariate normal random variables; a detailed proof will be provided in a forthcoming paper.
where
\[
E[x_h s_h] = E[x_0 s_0] + E[z_0] \cos \phi_0 I_1 + \sigma^2 \cos \phi_0 I_2
\]
\[
E[y_h s_h] = E[y_0 s_0] + E[z_0] \sin \phi_0 I_1 + \sigma^2 \sin \phi_0 I_2
\]
\[
E[x_h \phi_h] = E[x_0 \phi_0] + \frac{E[z_0 \cos \phi_0]}{\sin \phi_0} I_1 - \sigma^2 \cos \phi_0 I_2
\]
\[
E[y_h \phi_h] = E[y_0 \phi_0] + \frac{E[z_0 \sin \phi_0]}{\sin \phi_0} I_1 + \sigma^2 \cos \phi_0 I_2.
\]

Finally,
\[
P_{h}^{11} = \begin{pmatrix}
E[x_h^2] - (E[x_h])^2 & E[x_h y_h] - E[x_h] E[y_h] & E[y_h^2] - (E[y_h])^2
\end{pmatrix}
\]
in which
\[
E[x_h y_h] = E[x_0 y_0] + E[z_0 \sin \phi_0] (J_1 + J_2)
\]
\[
E[x_h^2] = E[x_0^2] + E[z_0^2 \cos^2 \phi_0] (J_1 + J_2)
\]
\[
E[y_h^2] = E[y_0^2] + E[z_0^2 \sin^2 \phi_0] (J_1 + J_2)
\]

The unspecified terms of the form \(E[z_0 \cos \phi_0],\)
\(E[z_0^2 \cos \phi_0],\) \(E[z_0 \sin \phi_0],\) \(E[z_0^2 \sin \phi_0],\) \(E[z_0 \cos \phi_0],\) \(E[z_0^2 \cos \phi_0],\) \(E[z_0 \sin \phi_0],\) \(E[z_0^2 \sin \phi_0],\) etc., appearing in the above equations, can be calculated by making use of the following formulae:
\[
E[\cos \phi_0] = e^{-\sigma^2_0 / 2} \cos \phi_0
\]
\[
E[\sin \phi_0] = e^{-\sigma^2_0 / 2} \sin \phi_0
\]
\[
E[z_0 \cos \phi_0] = (\hat{z}_0 \cos \phi_0 - \text{cov}(z_0, \phi_0) \sin \phi_0) e^{-\sigma^2_0 / 2}
\]
\[
E[z_0 \sin \phi_0] = (\hat{z}_0 \sin \phi_0 + \text{cov}(z_0, \phi_0) \cos \phi_0) e^{-\sigma^2_0 / 2}
\]
\[
E[z_0 w_0 \cos \phi_0] = e^{-\sigma^2_0 / 2} \cos \phi_0 [\hat{z}_0 \hat{w}_0 + \text{cov}(z_0, w_0)]
\]
\[
E[z_0 w_0 \sin \phi_0] = e^{-\sigma^2_0 / 2} \sin \phi_0 [\hat{z}_0 \hat{w}_0 + \text{cov}(z_0, w_0)] + \text{cov}(z_0, w_0)
\]
\[
E[z_0 \cos \phi_0] = e^{-\sigma^2_0 / 2} \cos \phi_0 [\hat{z}_0 \hat{w}_0 + \text{cov}(z_0, w_0)] + \cos \phi_0 \text{cov}(z_0, w_0)
\]
\[
E[z_0 \sin \phi_0] = e^{-\sigma^2_0 / 2} \sin \phi_0 [\hat{z}_0 \hat{w}_0 + \text{cov}(z_0, w_0)] + \sin \phi_0 \text{cov}(z_0, w_0)
\]

where the variables \(z_0\) and \(w_0\) stand for any component of the state vector \((x_0, y_0, s_0, \text{or } \phi_0)\), as appropriate in each case, and \(\sigma^2 \phi_0\) is used to denote \(\text{cov}(\phi_0)\). Notice that the terms involving quadratic trigonometric functions, such as \(E[z_0^2 \cos^2 \phi_0]\) and \(E[\cos^2 \phi_0]\), can be expanded via standard trigonometric identities; the resulting expressions involving \(2\phi_0\) (instead of \(\phi_0\)) are then given by (14), by replacing \(\phi_0\) with \(2\phi_0\), \(\text{cov}(z_0, \phi_0)\) with \(2\text{cov}(z_0, \phi_0)\) and \(\sigma^2 \phi_0\) with \(4\sigma^2 \phi_0\).

We point out that, even though the resulting density \(p(x_h)\) of the state vector at time \(h\) is non-Gaussian, the above expressions for its mean and covariance are exact.

### III. Filter Design with the CH Model

In this Section we outline four alternative filtering algorithms that make use of the model of Section II. As the model (2) is nonlinear and non-Gaussian, all algorithms provide approximations to the conditional density of the target, given the set of available measurements.

#### A. Kalman filter (KF-CH)

It is straightforward to design a Kalman filter-like algorithm using the results of Section II, by employing equations (9)-(10) in the prediction step of the algorithm, and the standard Kalman filter formulae for the measurement update step.

Thus, assume that at time \(t - 1\) a Gaussian approximation to the conditional density of the state is available, with mean \(\hat{x}_{t-1|t-1}\) and covariance \(P_{t-1|t-1}\). When a new measurement becomes available at time \(t\), the mean and covariance of the state vector are propagated to the current time via equations (9)-(10) to obtain the predicted estimates \(\hat{x}_{t|t-1}\) and \(P_{t|t-1}\), which are then updated to incorporate the latest measurement via the Kalman filter update formulae.

Such an approach approximates the (non-Gaussian) predicted density \(p(x_t|z_{1:t-1})\) with a moment-matched Gaussian, which allows for the application of the standard Kalman filter update equations, resulting in a Gaussian posterior density (which is, itself, an approximation to the conditional density of the state).

#### B. Extended Kalman filter (EKF-CH)

An alternative approach for constructing a ‘moment-matched’ algorithm is to linearize the non-linear state transition model of Section II via Taylor expansion in the extended Kalman filter framework ([9]). For the model (2), this corresponds to the following equations for the predicted mean and covariance of the state vector:

\[
\hat{x}_{t|t-1} = \hat{x}_{t-1|t-1} + h \text{cov}(\hat{x}_{t-1|t-1}, \sin(\hat{\phi}_{t-1|t-1})) + (I + 3A) h^2 / 2 + AQ^2 h^3 / 3
\]

\[
P_{t|t-1} = (I + A h) P_{t-1|t-1} + (I + A h) P_{t-1|t-1} (I + A h) + (Q + A Q^2 + A Q^3) h^3 / 3
\]

\[
A = \begin{pmatrix}
O_{2 \times 2} & F \\
O_{2 \times 2} & O_{2 \times 2}
\end{pmatrix}
\]

\[
F = \begin{pmatrix}
\cos(\hat{\phi}_{t-1|t-1}) & \hat{s}_{t-1|t-1} \sin(\hat{\phi}_{t-1|t-1}) \\
\sin(\hat{\phi}_{t-1|t-1}) & \hat{s}_{t-1|t-1} \cos(\hat{\phi}_{t-1|t-1})
\end{pmatrix}
\]
where \( Q = \text{diag}(0,0,\sigma^2_x,\sigma^2_o) \), \( h \) is the sample period and \( O_{2 \times 2} \) is a 2-by-2 matrix of zeros.

The measurement update equations are then given by the standard Kalman filter formulae (assuming linear measurements).

C. Particle filter (PF-CH)

To cater for the nonlinear and non-Gaussian structure of the problem, it is relatively simple to construct a sampling-importance-resampling (SIR) particle filter for the model of Section II by employing an Euler discretization ([11]) of the stochastic differential equation (2) to propagate the \( i \)-th particle \( \mathbf{x}_t^i = (x_t^i, y_t^i, s_t^i, \phi_t^i)^T \) as

\[
\begin{align*}
x_{t+1}^i &= x_t^i + h s_t^i \cos \phi_t^i \\
y_{t+1}^i &= y_t^i + h s_t^i \sin \phi_t^i \\
s_{t+1}^i &= s_t^i + \sigma_s \sqrt{h} n_t^i \\
\phi_{t+1}^i &= \phi_t^i + \sigma_{\phi} \sqrt{h} w_t^i,
\end{align*}
\]

where \( h \) is the sample period and \( n_t^i \) and \( w_t^i \) are independent standard normal random variables (\( n_t^i, w_t^i \sim \mathcal{N}(0,1) \)).

The other steps in the algorithm are standard (see, for example, [2]). We note that it is also possible to perform a multi-step Euler discretization for propagating the particles; simulation results, however, showed that such a modification did not result in any gains in performance.

D. Interacting Multiple Model filter (IMM-CH)

The Kalman filter implementation (KF-CH) of the constant heading model can be easily incorporated in an Interacting Multiple Model algorithm ([3]), so as to efficiently handle different modes of target motion during the tracking process. The IMM tracker can either use a standard (nearly) constant velocity Kalman filter (KF-CV) for the linear segments of the track and a KF-CH module to handle target maneuvers, or, alternatively, employ two KF-CH modules with different noise intensity parameters \( \sigma_s \) and \( \sigma_{\phi} \). In the former case, the velocity part of the state estimate produced by the KF-CV module needs to be transformed into its polar representation (i.e. speed and heading) during the ‘mixing’ and ‘combination’ stages of the algorithm — this task is performed via standard linearization techniques.

IV. SIMULATIONS

To assess the performance of trackers employing the proposed constant heading model, we consider a scenario that has featured in an earlier comparative study ([4]), in which a target, flying westbound at a constant speed of 120 m/s, executes two maneuvers of 0.2 \( g \) and 0.6 \( g \) respectively, as shown in Figure 1.

To simplify the evaluation, as in [4], we consider a sensor situated at the origin, taking noisy measurements of the target’s Cartesian position every \( h = 5 \) seconds. That is, the measurements available at time \( t \), \( \mathbf{z}_t \), are given by

\[ \mathbf{z}_t = (x_t, y_t)^T + \mathbf{n}_t, \]

where \( \mathbf{n}_t, t = 1, \ldots, T \), is a sequence of i.i.d. Gaussian random variables \( \mathcal{N}(0, \sigma^2 I_{2 \times 2}) \).

The following algorithms are compared in the simulations:

- **PF-CH**: a particle filter employing 20000 particles, using a first order Euler scheme in the discretization of (2) as described in Section III-C
- **KF-CH**: a Kalman filter using the CH model by using the results of Section II in the prediction step
- **EKF-CH**: a continuous-discrete extended Kalman filter, linearizing equation (1) in the prediction step, as described in Section III-B
- **KF-CV**: a Kalman filter employing the standard nearly constant velocity model to describe the motion of the target
- **IMM-CT**: an IMM (interacting multiple model) filter, switching between a KF-CV and a coordinated turn (CT) model, as described in [4]
- **IMM-L**: an IMM filter, composed of two KF-CV modules with different process noise intensity parameter
- **IMM-CVCH**: an IMM filter, making use of a KF-CV module for the linear segments of the track and a KF-CH module for the maneuvering parts, as described in Section III-D
- **IMM-CHCH**: an IMM filter comprised of two KF-CH modules.

The KF-CV, IMM-CT and IMM-L filter parameters were chosen as in [4]; namely, the KF-CV algorithm used process noise of standard deviation \( \sigma = 1 m/s^2 \) (as a compromise between the maximum acceleration and constant velocity portions of the track), the IMM-L algorithm employed \( \sigma = 0.1 m/s^2 \) for the uniform motion and \( \sigma = 2 m/s^2 \) for the maneuvers respectively, and the IMM-CT algorithm employed \( \sigma = 0.1 m/s^2 \) in the KF-CV module and \( \sigma = 0.5 m/s^2 \) and \( \sigma_{\omega} = 0.2 \) in the CT module.

The KF-CH, EKF-CH and PF-CH algorithms used process noise parameters \( \sigma_s = 1 m/s \) for the speed component of
the velocity and \( \sigma_\phi = 0.16rad/s \) for the heading, so as to accommodate target maneuvers, while the IMM-CVCH used \( \sigma = 0.2m/s^2 \) in the KF-CV module and \( \sigma_x = 1m/s \) and \( \sigma_\phi = 0.16rad/s \) in the KF-CH module. Finally, the modules of the IMM-CHCH assumed \( \sigma_x = 0.1m/s \) and \( \sigma_\phi = 0.03rad/s \) to model the uniform motion, and \( \sigma_x = 1m/s \) and \( \sigma_\phi = 0.16rad/s \) for the maneuvering portions of the track.

All IMM filters assumed a mode transition probability matrix

\[
\pi = \begin{pmatrix}
0.95 & 0.05 \\
0.10 & 0.90
\end{pmatrix}
\]

and were initialized with the mode probability vector \( \mu = (1, 0)^T \).

Three variations of the scenario were examined, by allowing the measurement noise standard deviation \( \sigma \) to take values of 100\( m \), 250\( m \) and 500\( m \) respectively. Figures 2(a)-2(f) show the RMS position errors for each case, averaged over 100 Monte Carlo trials.

In particular, Figures 2(a), 2(c) and 2(e) show the RMS error for the PF-CH, KF-CH, KF-CV, EKF-CH and IMM-CT algorithms. All ‘single-model’ CH algorithms significantly outperform the KF-CV method, and exhibit a constant steady-state error during the whole track, regardless of whether the target is moving along a straight line or undergoing a turn. This steady-state error is, however, slightly higher than that achieved by the KF-CV algorithm during the linear parts of the track.

The KF-CH algorithm matches the performance of the PF-CH method in all cases, for a fraction of the computational cost (see Table I for the computational cost\(^3\) of each algorithm), showing that the implicit Gaussian approximation due to the application of the KF equations does not adversely affect tracking performance. The EKF-CH algorithm also reaches the same levels of RMSE as the KF-CH and PF-CH methods; other simulations showed, however, that the algorithm may sometimes suffer from instability and is therefore not preferable over the KF-CH version. The IMM-CT algorithm outperforms the filters that are based on the constant heading model during the linear parts of the track, but fails to do so when the target executes a maneuver, even though it improves very significantly on the KF-CV estimates.

The ‘multiple-model’ algorithms are compared in Figures 2(b), 2(d) and 2(f). As reported in [4], the IMM-CT algorithm improves on the performance of the IMM-L during target maneuvers, while the IMM-CVCH significantly outperforms both filters during the second, sharper turn by approximately 25%. The IMM-CHCH filter outperforms the IMM-CVCH algorithm during the middle leg of the track, but exhibits slower convergence in the initial stage of the tracking process.

### V. Conclusions

We have outlined the constant heading model for describing maneuvering target motion, and provided closed-form expressions for the exact first two moments of the time-propagated state vector under the assumption of a Gaussian prior.

Using these results, we constructed four different filtering algorithms making use of the model, and assessed their performance in one scenario of interest. Simulation results indicate that algorithms based on the constant heading model significantly outperform a Kalman filter employing a CV model with inflated process noise during the maneuvering parts of the track, at the cost of an increase in the steady-state error during the linear segments. In fact, the resulting trackers exhibit a constant steady-state error along the entire track, irrespective of whether the target is turning or moving along a straight line.

In addition, our Kalman filter-like algorithm was generally found to be more robust to a linearized version based on the Extended Kalman filter framework, while it achieves the same level of performance as a high order particle filter constructed according to the same transition model, showing that the implicit Gaussian approximation due to the application of the Kalman filter equations does not adversely affect tracking performance. Finally, an IMM filter making use of the constant heading model in place of the coordinated turn model is shown to further reduce estimation errors that are associated with an IMM filter employing the CT model during target maneuvers.

Extensive simulations, as well as full details on the derivation of the results presented here, will be given in a forthcoming paper. Investigation of alternative discretization schemes for the construction of particle filter algorithms, using methods as in [8], will also be the subject of future work.

### References


### Table I

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CPU time (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF-CH</td>
<td>3.9554</td>
</tr>
<tr>
<td>KF-CH</td>
<td>0.0312</td>
</tr>
<tr>
<td>KF-CV</td>
<td>0.0492</td>
</tr>
<tr>
<td>EKF-CH</td>
<td>0.0471</td>
</tr>
<tr>
<td>IMM-CT</td>
<td>0.0609</td>
</tr>
<tr>
<td>IMM-L</td>
<td>0.0558</td>
</tr>
<tr>
<td>IMM-CVCH</td>
<td>0.0857</td>
</tr>
<tr>
<td>IMM-CHCH</td>
<td>0.1002</td>
</tr>
</tbody>
</table>

**TABLE I

AVERAGE CPU TIME PER MONTE-CARLO RUN**

\(^3\)All simulations were performed on a computer equipped with an Intel Xeon Quad-Core 2.53GHz processor and 12GB of RAM.
Fig. 2. Estimation Error


