A Comparison of Several Filters for Maneuvering Targets

Peter Abeles
Intelligent Automation Inc.
Rockville, MD U.S.A.
pabeles@i-a-i.com

Abstract — In this study, the performance of eleven different filters are compared against a simulated maneuvering target. To provide applicability to a wide range of problems, a diverse set of metrics are considered. To ensure rigorous testing, a combination of randomly generated and fixed extreme trajectories are used. The effects of maneuvers are extracted by processing the results conditionally on maneuver type, sojourn time and filter phase.

Keywords: Filter, Maneuver, Manoeuvre, MTT, Evaluation, Performance, Consistency

1 Introduction

Maneuvering target tracking (MTT) is an important problem with applications in numerous fields, such as: air traffic control, robotics, surveillance, and defense. A maneuvering target has the potential to change its behavior unexpectedly. The unknown future target behavior is what makes MTT a difficult problem.

To track a target, its state is estimated using a filter. Deciding which filter to use is a challenging task, partially because of the number of options. To intelligently select a filter, its effects on the system as a whole and its performance relative to other filters need to be understood.

Comparative studies show the relative performance of several filters in one or more scenarios. Recent comparative studies [1,2] and comparisons performed in numerous other works [3–6], either focused on a small set of filters or a single type of filter. State estimation accuracy and maneuver classification accuracy are the most commonly evaluated metrics in those studies.

In this paper, the results of a study on eleven diverse filters is presented against a simulated maneuvering target. While not completely comprehensive, more filters are considered than in previous studies. These filters represent both popular and less common algorithms and strategies used in MTT today.

Most of the metrics used and how they are computed are not typical. In addition to error metrics, several other metrics are computed that provide insight for various MTT related problems. For example, multi-sensor fusion, hand-offs, classification, and closely spaced objects. Metrics are computed conditionally on maneuver type, maneuver sojourn time, and filter phase. This clearly shows how target maneuvers effects filter performance.

MTT is a diverse field and a single study can not provide a definitive answer as to the best filter to use in all scenarios. Many real-world issues (e.g. non-linearity and sensor variability) are intentionally omitted in this study for sake of simplicity. When taken in account, the same general trends observed in this study are often still true.

Performance Metrics: A filter is a specific implementation of an algorithm, which estimates the target’s state given a set of measurements. Sequential algorithms are popular because of their computational efficiency. There are numerous metrics [7] for quantifying filter performance.

In this study, filter performance is evaluated by first using the following basic metrics: “mean” accuracy, outlier accuracy, containment, gating region size, bias, transient phase length, maneuver classification accuracy, and relative runtime. Second, by considering maneuver sojourn time conditional performance, the worst-case and steady-state performance metrics are found.

Accuracy is a measure of the difference between an estimated state and its true value. The term “mean” accuracy, in this paper, refers to any metric which attempts to evaluate how well a filter will perform most of the time. Outlier accuracy refers to a metric that evaluates how poorly a filter can be expected to perform. In MTT, the outlier performance is primarily driven by a filter’s response to maneuvers.

In a fault tolerant system, which can survive several bad estimates, outlier accuracy is less important and “mean” accuracy more important. This type of system

---

1The source code for both the filters and evaluation tools used in this paper are available at http://sourceforge.net/projects/jtargettracking/.
is common in MTT. Outlier accuracy is important in fragile systems, where a single bad state estimate can cause a failure. This is the case when a hand-off is performed between sensors and with targeting systems. Consistency [4,8] is how accurately a filter describes the uncertainty of its estimate. This uncertainty is represented by a probability distribution function (PDF). Poor consistency will cause assignment issues that degrade track performance.

The state estimate’s PDF plays an important role in resolving the assignment ambiguity. The more confined the PDF is, the better it is at resolving ambiguity. This is especially true when dealing with: point and small targets (since there are few features that can aid in association), closely spaced objects, and multi-sensor fusion.

A gating region [6] defines what measurements can be considered for association with a particular track. Measurements outside of a gating region are pruned. By pruning unlikely associations, the efficiency of assignment algorithms is improved. Gating region size is a metric that evaluates the PDF’s effectiveness for resolving ambiguity.

A consistent filter will have a known containment. Containment refers to the expected fraction of true target observations that are contained inside the gating region. Observed containment of a gating region can be used as a metric to gauge how accurate the PDF is at describing the uncertainty. If the empirically computed containment is different than the expected value, the quality of the association will degrade.

Another measure of consistency is bias. Bias is defined as the difference between estimate’s true expected value and the reported expected value. A biased estimate increases the probability of otherwise unlikely failure conditions. A filter will typically become biased after a maneuver has occurred.

After a maneuver, the filter’s performance enters a transient phase. Knowing the transient phase length (or recovery time [6]) is important for system designers. It can be used to decide when a maneuver type estimate can be trusted or how many missed associations should be allowed.

Real-time classification of a target’s maneuvers can help defense systems classify and prioritize targets. Many filters detect maneuvers automatically as part of their design. A filter with superior maneuver classification accuracy is desirable for these systems.

An algorithm’s relative runtime is the length of its processing time relative to other algorithms. One algorithm might have better accuracy than another, but if it requires more than the available resources, it can not be used. The relative runtime of an algorithm is highly implementation and hardware dependent.

Performance conditional on a specific target behavior can expose systematic failure conditions. From conditional results the worst-case and steady-state performance are found. This allows a system designer to know the most likely performance and how poorly it can perform. From a practical point of view, it is often impossible to know what conditions are currently valid. Depending on the system requirements, this can force a system to be designed for the worst-case performance.

**Filter Design:** When designing a filter for MTT, it is essential to understand how it reacts to maneuvers. A maneuver event refers to the instantaneous change in the target’s behavior. The target’s mode refers to a specific behavior that is performed over a period of time. A maneuver can either refer to the maneuver event or the target’s mode, depending on the context.

A filter’s reaction to maneuvers can be broken up into two phases: transient, and steady-state. The transient phase begins immediately after a maneuver and the steady-state after the filter has had time to stabilize. While in the transient phase, it will typically have the largest errors and consistency issues (but not always, as will be shown). How long the transient phase lasts depends on the maneuver, sensors, and the filter used.

As a general rule, if a filter has good performance in the steady-state phase, it will have poor performance in the transient phase and vice versa. For a filter to perform well in the steady-state phase, it has to make strong accurate assumptions. Stronger assumptions result in larger differences between the filter’s model and the target’s behavior in the transient phase.

The design of a filter will significantly affect its performance and great care needs to be taken when selecting a particular filter for a specific task. Three different design strategies for MTT filters are discussed: hard decision, soft decision, and pessimistic. The difference between these strategies is in how they detect and estimate maneuvers.

In this paper, a hard decision filter computes its output using a single, strong assumption and discards information when the assumption is changed. A soft decision filter continuously estimates the current maneuver and does not discard information. A pessimistic filter does not attempt to determine the current maneuver, instead it assumes one is always about to happen.

Hard decision filters typically perform best while the filter is in a steady-state and produce large errors immediately after maneuvers. Soft decision filters exhibit neither the worst nor the best performance. Pessimistic filters are not a typical approach and they have the unusual property of performing the best immediately after a maneuver and the worst in the steady-state phase.

Which category a filter belongs in can be ambiguous. A single filter can employ elements of different strategies. This is often necessary to achieve acceptable performance. The same algorithm can be configured into
different filters that conform to different strategies, see Section 3.3 for an example.

**Evaluation Strategy:** The purpose of evaluating a filter is to determine if its performance is sufficient to perform a specific task. A critical aspect of evaluating filters in MTT is determining what the effects of maneuvers are on performance. If a filter is known to perform poorly during maneuvers, then a tracking system needs to be designed to handle that.

The most common approach found in the literature for evaluating maneuvering targets, is to consider a small set of simulated trajectories. The results are typically plotted as a function of simulation time for each trajectory and summarized in tables [1,2,4–6]. In those works, summary tables show metrics computed from samples of: one or more entire trajectories, segments of a single trajectory, or a weighted sets of trajectory segments.

Filter performance can vary significantly for each trajectory. It is well understood that to accurately tune and evaluate a filter numerous trajectories are needed. Intuitively, simulating a large number of trajectories would provide a good set of trajectories to evaluate a filter with, as was done in [5]. The advantage of computing results from a single trajectories is that some specific filter responses are detectable that would be washed out if results are computed from multiple trajectories.

In this paper, analysis is performed by using both randomly generated and fixed extreme trajectories. The fixed extreme trajectories show the filters’ performance against selected pathological cases. For all trajectories, the performance conditional on maneuver type, filter phase, and maneuver sojourn time are computed.

Conditional performance is computed by separating the results into different bins using the maneuver features listed above. There are two primary benefits to this approach: results associated with different features do not influence each other, and results from multiple trajectories and maneuvers are easily combined. This allows causal relationships to be easily and precisely identified.

The following notation is used: $T$ is the length of a time step, the subscript $k$ indicates the value at time step $k$, $\|a - b\|^2_2 = (a - b)'B(a - b)$, $N(x, P)$ is a normal distribution, $U(a,b)$ is a uniform distribution, and $F'$ represents the transpose of matrix $F$.

## 2 Experimental Setup

Monte-Carlo (MC) simulation is used to evaluate the filters’ performance. The target’s initial state, measurement noise, maneuver times, and maneuver magnitudes are all randomly drawn from their respective distributions. A total of 4000 MC trials are performed for each trajectory type: the random set and each fixed extreme trajectory. Each MC trial is run for a total of 100 simulated seconds with a fixed time step of $T = 0.1s$.

A simulated target with piecewise constant acceleration is considered. Many measurements are required before maneuvers become statistically significantly in sensor space. During this time, large errors can accumulate. This is similar to many real world scenarios. While not perfectly analogous, the target described below draws inspiration from aircraft, such as missiles and airplanes, that vary their acceleration during flight.

The target alternates between extended periods of coasting (no acceleration) and brief periods of acceleration. This behavior is common among maneuvering targets since acceleration uses more resources.

### 2.1 Target Model

**Kinematics:** The target’s kinematics is described by a linear system:

$$
\begin{bmatrix}
    p_k \\
    v_k
\end{bmatrix} = \begin{bmatrix}
    1 & \Delta \\
    0 & 1
\end{bmatrix}
\begin{bmatrix}
    p_{m-1} \\
    v_{m-1}
\end{bmatrix} + \begin{bmatrix}
    \frac{1}{2} \Delta^2 \\
    \Delta
\end{bmatrix} a_m
$$

(1)

where $x_k = [p_k, v_k]'$ is the target’s state, $\Delta$ is the current maneuver’s sojourn time, $[p_{m-1}, v_{m-1}]'$ is the target’s state at the end of the previous maneuver, and the current control acceleration is $a_m$. A sensor makes noisy position measurements:

$$
z_k = p_k + n_z
$$

(2)

where $n_z \sim N(0, \sigma^2_{n_z})$ is independent zero-mean white Gaussian noise. The target’s initial state $x_0$ is drawn from an independent multivariate Gaussian distribution $\bar{x}_0 \sim N(\bar{x}_0, P_0)$.

$$
\bar{x}_0 = \begin{bmatrix}
    0 & m \\
    2m & \tau
\end{bmatrix} \quad P_0 = \begin{bmatrix}
    20^2 m^2 & 0 \\
    0 & 5^2 (m/s)^2
\end{bmatrix}
$$

$$
\sigma^2_{n_z} = 20^2 m^2
$$

**Control:** When in the coasting mode, $a_m = 0$, otherwise the target is in the accelerating mode. Maneuvers are allowed to occur at any time, including between measurements. The target is not allowed to transition into the same mode multiple times in a row. The maneuver sojourn time for coasting and accelerating modes is $T_c$ and $T_m$ respectively. These are drawn from uniform distributions when a new maneuver begins. The accelerating mode’s acceleration magnitude, $a_c$, is selected when it first enters the mode and remains constant until it leaves. This control mechanism can be described as a semi-Markov process.

In addition to randomly generated trajectories, four fixed extreme trajectories are also evaluated individually. Targets in these trajectories always perform maneuvers with acceleration and sojourn times that are either at their maximum or minimum allowed values.
\( T_c = U(5, 40)s \quad T_m = U(1, 9)s \quad a_c = U(5, 50)m/s^2 \)

### 3 Filter Discussion

A high level subjective summary of the approaches taken by the filters is shown in Table 1. “Adaptive Kinematics” indicates that a filter can change its kinematic model. “Hard” and “Soft” indicate how extreme of a hard or soft decision approach the filter takes. “Transition Accuracy” and “Kinematic Accuracy” indicate how accurately the maneuver event and mode are modeled, respectively. “Transition Process” describes what sort of process models the maneuver sequence. “Relative Runtime” shows how long the filters’ runtime is relative to each other.

#### 3.1 Tuning

How a filter is tuned can dramatically change its performance. Tuning a filter for steady-state performance will typically make transient performance worse and vice versa. In this study, all of the filters are tuned empirically, first for containment and second for “mean” position/velocity accuracy across the randomly generated trajectory scenario. The reason containment is used in this study, is that in many systems, if a filter has very good “mean” accuracy, but is inconsistent, it can fail due to association issues. Common practices that could introduce unintentionally biased results are ignored. For instance, the plant noise in many of the filters is set to zero. It is common practice to inflate the plant noise because of potential unanticipated target behaviors and numerical issues.

#### 3.2 Filter Kinematic Models

Two different kinematic models are defined below: constant control acceleration and dynamic acceleration. The former assumes the acceleration is known and nearly constant. The latter allows acceleration to be estimated and change with time.

- **Constant Acceleration Model**

\[
\begin{align*}
    x_k &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} \frac{1}{2} T^2 \\ T \end{bmatrix} a_c + w_c \\
    \end{align*}
\]

where \( x_k = [p_k, v_k]' \), \( w_c \sim N(0, Q_c) \), \( Q_c = \Gamma_c \sigma_c^2 \Gamma_c' \), and \( \Gamma_c = \left[ \frac{1}{2} T^2, T \right] \).

- **Dynamic Acceleration Model**

\[
\begin{align*}
    x_k &= \begin{bmatrix} 1 & T & \frac{1}{2} T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} x_{k-1} + w_a \\
    \end{align*}
\]

where \( x_k = [p_k, v_k, a_k]' \), \( w_a \sim N(0, Q_a) \), \( Q_a = \Gamma_a \sigma_a^2 \Gamma_a' \), and \( \Gamma_a = \left[ \frac{1}{2} T^2, T, 1 \right] \).

### 3.3 Kalman Filter

A Kalman filter (KF) [4,9] is an optimal minimum mean square error (MMSE) estimator. It provides the basic structure that many filters build upon in MTT. As an estimator for maneuvering targets, its modeling capabilities are limited. It can only describe a target using a single set of kinematic equations. The system equations used by the discrete-time KF are shown below:

\[
\begin{align*}
    x_k &= F_{k-1} x_{k-1} + G_{k-1} u_{k-1} + w_{k-1} \\
    z_k &= H_k x_k + v_k \\
    \end{align*}
\]

where \( F_k, G_k, \) and \( u_k \) are known, \( w_k \sim N(0, Q_k) \) and \( v_k \sim N(0, R_k) \) are mutually independent known zero-mean white Gaussian noise processes. Where \( w_k \) is referred to as the plant or process noise of the system.

#### 3.3.1 PVA

A common way to apply a KF to accelerating targets is to approximate maneuvers with plant noise on the acceleration. Target’s kinematics is modeled by Eq. 4. This amounts to a soft decision strategy, since the maneuver (acceleration) is constantly being estimated.

#### 3.3.2 PVcA

The KF can also be configured to employ a pessimistic strategy. By using Eq. 3, the acceleration is set to a fixed value. The acceleration’s value is selected so that it would minimize the maximum acceleration error when a maneuver occurs.

### 3.4 H\(_\infty\) Filter

The H\(_\infty\) (H-infinity or minimax) filter is designed to be robust against noise modeling inaccuracies. A KF requires strong assumptions about the statistical nature of noise processes. In practice, noise models are often approximate because of the nature of the problem or the lack of engineering resources. A KF provides an optimal solution, but does not bound errors. A H\(_\infty\) filter guarantees that the cost function \( J_1 \) is bounded by the inequality \( J_1 \leq 1/\theta \), see [10] for more details.

In a related paper, the performance of a KF and H\(_\infty\) are compared in [11] for an accelerating maneuvering target. It was found that both filters required tuning to achieve optimal performance. After tuning, both filters had comparable performance.

#### 3.4.1 PVA\(_\infty\)

This filter is analogous to the PVA filter, since both use Eq. 4 for their kinematics. Unlike the PVA filter, there are multiple tuning parameter (plant noise and \( \theta \)) values that have the desired containment.

The H\(_\infty\) filter’s problem formulation does not provide an error covariance matrix. However, the H\(_\infty\) filter can be configured such that it is identical to a KF, see [10]. In this paper, the noise processes are Gaussian and the filter is configured in a similar way to a KF. To allow...
Table 1: A subjective overview of approaches taken by each filter. Larger values indicates that a filter is more accurate or is a more “pure” follower of a strategy. M stand for Markov.

<table>
<thead>
<tr>
<th></th>
<th>Adaptive Kinematics</th>
<th>Multi-Hypothesis</th>
<th>Hard</th>
<th>Soft</th>
<th>Transition Accuracy</th>
<th>Kinematic Accuracy</th>
<th>Transition Process</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>PVcA</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>1.0</td>
</tr>
<tr>
<td>PVA∞</td>
<td>0</td>
<td>10</td>
<td>2</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>1.2</td>
</tr>
<tr>
<td>IMM_d</td>
<td>✓</td>
<td>✓</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>M</td>
<td>27.7</td>
</tr>
<tr>
<td>IMM_2</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>9</td>
<td>5</td>
<td>5</td>
<td>M</td>
<td>5.5</td>
</tr>
<tr>
<td>MM-k</td>
<td>✓</td>
<td>✓</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>M</td>
<td>119.5</td>
</tr>
<tr>
<td>ME-kS</td>
<td>✓</td>
<td>✓</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>M</td>
<td>114.9</td>
</tr>
<tr>
<td>ME-kH</td>
<td>✓</td>
<td>✓</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>M</td>
<td>89.2</td>
</tr>
<tr>
<td>VD</td>
<td>✓</td>
<td>9</td>
<td>1</td>
<td>8</td>
<td>9</td>
<td></td>
<td></td>
<td>5.5</td>
</tr>
<tr>
<td>DDL</td>
<td>✓</td>
<td>7</td>
<td>3</td>
<td>7</td>
<td>9</td>
<td></td>
<td></td>
<td>49.0</td>
</tr>
</tbody>
</table>

all the metrics to be computed, it is assumed (without proof) that an error covariance matrix is provided.

### 3.5 Variable Dimension Filter

The Variable Dimension (VD) Filter [12] is a hard decision type filter. It operates in one of two modes at any given time, either non-maneuvering (quiescent) or maneuvering. When a target mode change is detected, it re-initializes the filter, changes the kinematics, and, if applicable, reprocesses measurements in its history window.

#### 3.5.1 VD

The non-maneuvering and maneuvering models are Eq. 3 with $a_c = 0$ and Eq. 4, respectively. Since the accelerating mode only terminates when the acceleration estimate becomes statistically equivalent to zero, a large plant noise is needed while using the accelerating model.

Both maneuver onset and offset are detected using the same fading memory tests suggested in [12]. However, the single measurement tests discussed in [12] are not used due an excessive number of false positives.

### 3.6 Double Decision Logic

A key feature of the Enhanced Variable Dimension filter, proposed in [13], is the reduction of false positives, by using double decision logic (DDL) for maneuver detection. DDL works by first using a simple less accurate test to identify when a maneuver might have occurred. This creates a new internal filter that represents a new maneuver. Second, a more rigorous test is used that compares the performance of the two internal filters simultaneously across multiple measurements. If the second test detects a maneuver, the new internal filter becomes active and information associated with the other one is discarded.

#### 3.6.1 DDL

The DDL algorithm used in this paper is inspired by [13]. It switches between two kinematic models. Coasting is modeled by Eq. 3 with $a_c = 0$. Accelerating is modeled by Eq. 4. The first test is a fading memory average innovation test and the second is an innovation maximum likelihood ratio test.

The coasting filter is given an initial state using past observations, as was done in [13]. For the accelerating filter, the initial acceleration is set to the expected value with the appropriate variance.

### 3.7 Input Estimation

The Input Estimation (IE) filter [14] models a maneuver as a control input in Eq. 5. When a new measurement arrives, the acceleration of the target is estimated using a least squares (LS) algorithm from the N most recent measurements. The LS algorithm assumes a constant acceleration within its window.

A $\chi^2$ test is used on the estimated acceleration to determine if a maneuver has occurred. The filter’s output is then corrected using the estimated acceleration. If a maneuver is not detected, the control input is set to zero and, if needed, the output is recomputed.

#### 3.7.1 IE

In this implementation of the IE algorithm, the acceleration is estimated at every time step. The kinematics is modeled using Eq. 3. The covariance is only adjusted when a large maneuver is detected. Typically, this is at maneuver onset. A non-zero plant noise helps to compensate for the acceleration not being constant inside its window.

### 3.8 Interacting Multiple Model

The Interacting Multiple Model (IMM) [15] and related [16] algorithms, have been a focus of recent re-
search in sequential filters for MTT. The IMM approximates maneuvers as a finite discrete set of \( N \) internal models/filters/hypotheses. The IMM is a suboptimal algorithm that approximates an optimal hypothesis merging strategy at a depth of two.

Each internal filter is assigned a probabilistic weight that is dynamically computed based on measurement innovations and a Markov process. This weight determines the model’s influence on the output and its influence on the other models. A Markov matrix \( \pi \in \mathbb{R}^{N \times N} \) describes the probability that the target will switch between the different models. The output is computed as a mixture of Gaussians of its internal filters.

There are several different popular ways to design an IMM filter. A similar problem is discussed in [2], where three different implementations are compared. Approaches, similar to the IMM\(_d\) and IMM\(_2\) algorithms, did well in maneuver equalized and unequalized performance, respectively.

### 3.8.1 IMM\(_d\)

The target is modeled as having \( M \) discrete acceleration levels and a single model for coasting. There are a total of \( N = M + 1 \) models, where each model uses Eq. 3 with different values for \( a_c \). The first (coasting) model has \( a_c^1 = 0 \) and the accelerating models are set to:

\[
a_c^n = (a_{max} - a_{min})(n-2)/(M-1) + a_{min}.
\]

Where \( a_{min} \) and \( a_{max} \) is the minimum and maximum acceleration the target can perform. The Markov matrix is set to:

\[
\pi = \begin{bmatrix}
p_c & \frac{1-p_c}{M} & \cdots & \frac{1-p_c}{M} \\
1-p_a & p_a & 0 & 0 \\
\vdots & 0 & \ddots & 0 \\
1-p_a & 0 & 0 & p_a
\end{bmatrix}
\] (7)

where \( p_c = \exp(-T/\tau_c) \) and \( p_a = \exp(-T/\tau_a) \). The values of \( \tau_c \) and \( \tau_a \) are set to the expected sojourn times, in the coasting and accelerating models respectively. The computation of \( p_c \) and \( p_a \) conform to the suggested guidelines in [4].

### 3.8.2 IMM\(_2\)

This filter is composed of two models that have different state dimensions. The acceleration is modeled by Eq. 4, and coasting is modeled with Eq. 3 with \( a_c = 0 \). The Markov matrix is set to:

\[
\pi = \begin{bmatrix}
p_c & 1-p_c \\
1-p_a & p_a
\end{bmatrix}
\] (8)

When merging hypotheses, the coasting model is treated as having an acceleration and acceleration variance of zero.

### 3.9 Multiple Model k-Best

The Multiple Model k-Best filter uses the same internal model as the IMM algorithm. The primary difference between the two algorithms is in how they handle hypothesis maintenance. In this filter, there is no interaction between hypotheses. Pruning is used to ensure that at most \( k \) hypotheses are allowed at any time.

#### 3.9.1 MM-k

The same Markov process and kinematic models used in the IMM\(_d\) filter are used in this filter. Tuning this filter for containment proved difficult. Increasing the plant noise degraded other metrics. Increasing the number of hypotheses significantly degrading runtime performance. The output is a mixture of Gaussians of all the hypotheses.

### 3.10 Maneuver Event k-Best

The MM approach can only coarsely model processes that are best described by discontinuous changes to parameters with a continuous domain. To overcome this weakness, Maneuver Event (ME) filters model maneuvers as a change in kinematics and a discontinuous change to the state.

ME k-best is the same as MM k-best, except for how hypotheses are generated. When creating a new hypothesis, ME creates the initial state by applying a state transformation function to the parent hypothesis:

\[
x_k^+ = G^{i,j}(x_k^-)
\] (9)

where \( x_k^- \) and \( x_k^+ \) are, respectively, the estimated state immediately before and after the hypothesized maneuver, \( G^{i,j} \) is the state transition function from model \( i \) to \( j \). The covariance is also adjusted to account for uncertainty related to the maneuver. A ME filter that uses a semi-Markov process is proposed in [17].

#### 3.10.1 ME-kS

Two kinematics models are used: Eq. 3 with \( a_c = 0 \) and Eq. 4, for the coasting and accelerating models respectively. Unlike MM-k, arbitrary accelerations can be estimated. The same Markov process used in IMM\(_2\) is used in this filter.

A transition from coasting to accelerating augments the state with an acceleration term, that is initially set to the expected value with the appropriate variance. Transitioning from accelerating to coasting truncates the state by removing the acceleration term. Other transitions do not change the state. For ease of implementation, it is assumed that maneuvers occurred just after new measurements arrive.

The same issues that MM-k experienced with containment were also experienced with this filter. The output is a mixture of Gaussians of all the hypotheses.

#### 3.10.2 ME-kH

Same as ME-kS, except that the output is the most likely hypothesis.
4 Evaluation Metrics

Simulation results are processed in several different conditional bins: \{all\}, \{maneuver type\}, \{maneuver type and sojourn time\}, and \{maneuver type and filter steady-state phase\}. In each of these bins the basic metrics (defined below) are computed.

Both 50\% and 95\% Euclidean percentile errors are computed for position and velocity. These metrics (P50, P95, V50, V95) represent the “mean” and outlier error metrics. Percentile error is used instead of the popular root mean square error (RMSE) and peak error. RMSE has no direct physical interpretation that system designers can use, see [7]. Peak error is undesirable because it is unbounded. As a result, peak error’s value increases with sample size (percentile error converges) and has no meaningful physical interpretation.

The size of the gating region is defined as the volume of a hyper-ellipsoid, specified by equation:

\[ R^G = \{ z \in \mathbb{R}^N : \| \hat{z} - z \|^2_{S^{-1}} \leq r_G \} \]  

(10)

where \( \hat{z} - z \) is the innovation, \( S \) is the innovation covariance, and \( r_G \) is a threshold that determines the size of the gating region. The PDF’s ability to resolve ambiguity is evaluated using the mean volume of the gating region (\( V^G \)). Since the measurements have 1-DOF, this can be computed easily using equation:

\[ V^G = \frac{1}{N} \sum_{i} ^N 2 \sqrt{r_G S_i} \]  

(11)

where \( N \) is the number of samples in the bin and \( S_i \) is the innovation variance for a sample. When the threshold of this gating region is set to \( r_G = 3.84 \) the expected containment is 95\%, the desired value of the containment metric.

The position bias \( B^{(p)} \) is computed using equation:

\[ B^{(p)} = \sum_{i} ^N [\hat{x}^{(p)} - x^{(p)}] \]  

(12)

where \( \hat{x}^{(p)} \) is the updated position estimate, and \( x^{(p)} \) is the true position. If a filter is unbiased, then the expected value is zero.

Maneuver classification accuracy is found by comparing the estimated maneuver to the true maneuver. The target is declared as being in the coasting mode if: a) IMM_d/IMM_z: the coasting model has a weight of more than 50\%, b) MM-k/ME-kS: the set of all coasting hypotheses has a weight of more than 50\%, c) ME-kH: if the most likely hypothesis is coasting, d) PVA/PVA∞: the acceleration is less than 2.5 m/s², e) VD/DDL/IE: the active model is coasting, and f) PVcA does not estimate the maneuver.

From the maneuver and sojourn time conditional basic metrics results, the transient phase length (\( \tau_{tran} \)), worst-case, and steady-state metrics are computed. The worst-case metrics are: maximum position 95\% error (MP95), maximum velocity 95\% error (MP95), minimum containment (MiCnt), and maximum magnitude position bias (MB\(^{(p)} \)). The steady-state metrics are: minimum position 50\% error (MiP50), and minimum velocity 50\% error (MiV50). When the transient phase ends is ambiguous and subjective. To make it less subjective, an automated algorithm is used to compute \( \tau_{tran} \).

5 Results

<table>
<thead>
<tr>
<th></th>
<th>P50</th>
<th>P95</th>
<th>V50</th>
<th>V95</th>
<th>Cont B(^{(p)} )</th>
<th>( V^G )</th>
<th>MA</th>
</tr>
</thead>
<tbody>
<tr>
<td>PVcA</td>
<td>8.3</td>
<td>23.6</td>
<td>16.1</td>
<td>36.7</td>
<td>0.950</td>
<td>4.9</td>
<td>112.1</td>
</tr>
<tr>
<td>PVA</td>
<td>7.1</td>
<td>20.9</td>
<td>7.6</td>
<td>27.4</td>
<td>0.950</td>
<td>-0.0</td>
<td>108.3</td>
</tr>
<tr>
<td>PV∞</td>
<td>7.0</td>
<td>21.1</td>
<td>6.6</td>
<td>31.4</td>
<td>0.950</td>
<td>-0.1</td>
<td>108.4</td>
</tr>
<tr>
<td>IMM(_d)</td>
<td>4.9</td>
<td>16.9</td>
<td>2.3</td>
<td>23.1</td>
<td>0.950</td>
<td>0.1</td>
<td>104.3</td>
</tr>
<tr>
<td>IMM(_z)</td>
<td>5.0</td>
<td>17.7</td>
<td>2.3</td>
<td>25.0</td>
<td>0.950</td>
<td>-1.0</td>
<td>104.7</td>
</tr>
<tr>
<td>MM-k</td>
<td>3.7</td>
<td>20.0</td>
<td>0.8</td>
<td>29.5</td>
<td>0.942</td>
<td>-0.5</td>
<td>102.3</td>
</tr>
<tr>
<td>ME-kS</td>
<td>4.5</td>
<td>19.9</td>
<td>1.6</td>
<td>28.0</td>
<td>0.941</td>
<td>1.8</td>
<td>102.2</td>
</tr>
<tr>
<td>ME-kH</td>
<td>4.4</td>
<td>21.7</td>
<td>0.9</td>
<td>31.7</td>
<td>0.933</td>
<td>1.2</td>
<td>100.2</td>
</tr>
<tr>
<td>VD</td>
<td>3.9</td>
<td>20.8</td>
<td>0.8</td>
<td>29.1</td>
<td>0.939</td>
<td>-1.3</td>
<td>102.2</td>
</tr>
<tr>
<td>DDL</td>
<td>4.0</td>
<td>23.1</td>
<td>0.8</td>
<td>33.4</td>
<td>0.934</td>
<td>-0.6</td>
<td>101.0</td>
</tr>
<tr>
<td>IE</td>
<td>4.5</td>
<td>20.9</td>
<td>1.1</td>
<td>30.8</td>
<td>0.935</td>
<td>-1.3</td>
<td>100.1</td>
</tr>
</tbody>
</table>

Table 2: Comparison of filter performance computed from all the samples in the randomly generated trajectory scenario. Effects of maneuvers are washed out in these results.

Tables 2 and 3 summarize the results from the randomly generated trajectories scenario. Results from the different fixed extreme trajectories are not shown due to space limitations. By comparing these two tables, it is clear that conditional performance needs to be considered to understand the affects of maneuvers. The plots in Figure 1 show the results as a function of sojourn time conditional on maneuver type.

As expected, the hard decision type filter had the best steady-state performance, but the worst worst-case performance. The opposite was true of the pessimistic filter, where the performance improved after maneuvers. Soft decision type filter tended to have middle of the road performance.

After a maneuver all filters, except for PVcA, experienced a large spike in their bias. PVcA has a large steady-state bias, but the magnitude of that bias is less than the peak bias of the other filters. Many of the filters also have bias in their steady-state phase. Some of the steady-state bias is “left over” bias due to the ambiguous end of the transient phase. In MH filters, this bias is caused (for the most part) by the need to consider unlikely hypotheses that skew the acceleration estimate. ME-kH is less biased than ME-kS, because
<table>
<thead>
<tr>
<th>(\tau_{\text{tran}})</th>
<th>Coasting Mode Only</th>
<th>Steady State</th>
<th>Worst Case</th>
<th>Accelerating Mode Only</th>
<th>Coasting Mode Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCIe</td>
<td>1.3</td>
<td>8.0</td>
<td>16.9</td>
<td>8.5</td>
<td>24.0</td>
</tr>
<tr>
<td>PVA</td>
<td>4.9</td>
<td>0.714</td>
<td>6.5</td>
<td>6.1</td>
<td>6.6</td>
</tr>
<tr>
<td>PCIe</td>
<td>6.1</td>
<td>0.818</td>
<td>6.2</td>
<td>4.8</td>
<td>6.4</td>
</tr>
<tr>
<td>IMM(_d)</td>
<td>3.1</td>
<td>0.997</td>
<td>4.0</td>
<td>1.5</td>
<td>4.3</td>
</tr>
<tr>
<td>IMM(_d)</td>
<td>3.1</td>
<td>0.998</td>
<td>4.0</td>
<td>1.4</td>
<td>4.3</td>
</tr>
<tr>
<td>MM-k</td>
<td>4.5</td>
<td>0.993</td>
<td>1.7</td>
<td>0.1</td>
<td>2.9</td>
</tr>
<tr>
<td>ME-kS</td>
<td>5.1</td>
<td>0.976</td>
<td>2.1</td>
<td>0.7</td>
<td>3.5</td>
</tr>
<tr>
<td>ME-kH</td>
<td>5.1</td>
<td>0.981</td>
<td>2.0</td>
<td>0.1</td>
<td>3.4</td>
</tr>
<tr>
<td>VD</td>
<td>3.9</td>
<td>0.980</td>
<td>1.8</td>
<td>0.1</td>
<td>3.1</td>
</tr>
<tr>
<td>DDL</td>
<td>5.1</td>
<td>0.924</td>
<td>1.8</td>
<td>0.1</td>
<td>3.0</td>
</tr>
<tr>
<td>IE</td>
<td>7.1</td>
<td>0.995</td>
<td>2.9</td>
<td>0.5</td>
<td>3.3</td>
</tr>
</tbody>
</table>

Table 3: Comparison of sojourn time conditional filter performance, from randomly generated trajectories.

its output comes from a single hypothesis and not a mixture. Hard decision filters’ bias largely comes from incorrectly determining the target’s mode.

In filters with dynamic kinematics, the gating region size vary as a function of sojourn time. For Multiple Hypothesis (MH) filters, increased ambiguity in the transient phase is the cause of the spike after a maneuver. In the steady-state, plant noise from the most likely/active hypothesis determine the gating region size.

To ensure containment and have acceptable errors after maneuvers, conditional performance must be considered when tuning the filters in MTT. By tuning the filters across all samples, filters with nearly perfect containment in Table 2 have too high containment in the steady-state phase and low worst-case containment in Table 3. The only filter which comes close to having the desired worst-case containment is PCIe.

One strategy, that is some times employed with MH filters, is to consider each hypothesis individually during association. While not considered in this study, it increases the chance of association during maneuvers, and the effective gating region size.

In the extreme trajectory scenarios, differences between the filters is sometimes exaggerated or reduced. With a few exceptions, the relative rankings of the filters stayed the same as the randomly generated scenario. These scenarios often exposed the brittleness of hard decision filters and other peculiar quirks of filters in general. For example, some filters had trouble classifying the maneuver type for small accelerations, while others did not.

For the most part, the results of this study confirmed results or statements made in the literature, when there is past work to compare to. There are some notable exceptions. In past comparisons of KF and IMM filters, the KF always has higher outlier errors [3,4,6]. However in Table 3 the IMM has higher worst-case error in the acceleration mode. In [12], the VDF filter was shown to be superior to the IE filter. In this study it is ambiguous which filter has the best performance.

The most likely cause of these discrepancies is how the filters are tuned. In those studies (either stated or assumed) filters were tuned to minimize “mean” error, but in this study they were tuned primarily for con-
tainment. Tuning for containment reduces outlier error and increase “mean” error.

As in [11], it was also found that a KF and a $H_\infty$ can be tuned to have similar performance. However, the additional observation was made that the $H_\infty$ filter can be tuned to have containment, and better “mean” performance, which the KF can not.

In MTT, minimizing worst-case performance requires different filter designs than minimizing steady-state performance does. Most of the literature has focused on reducing “mean” error. This has the counter intuitive result of more “advanced” filters having worse worst-case performance than more “primitive” ones. PVcA was included in this study to illustrate this point.

6 Conclusion

The performance of a diverse set of filters has been considered against a simulated maneuvering target. Several of the filters are not typical and many had not been compared before in the literature. By comparing and contracting these filters, interesting observations into the nature of filter design for MTT were made.

Considering a larger set of metrics provided insight into filter performance for several MTT related problems. When there was related past work in the literature, this study usually confirmed their results.

The evaluation methodology used in this study improved upon the techniques used in past studies. It provided a more precise understanding of the effects of maneuvers. This evaluation methodology can be used to perform better tuning and evaluation in simulation.

References


Figure 1: Performance metrics as a function of sojourn time conditional on maneuver type for random scenario. Some plots are cropped for clarity.