DEFENCE
SCIENCE & TECHNOLOGY

Apr 2003

DSTO

Estimation of Manoeuvring Targets using Hybrid Filters

Jason J. Ford & Peter G. Hunter

DSTO-TN-0488

Estimation of Manoeuvring Targets using Hybrid Filters

Jason J. Ford & Peter G. Hunter

Weapons Systems Division
Systems Sciences Laboratory

DSTO–TN–0488

ABSTRACT

This report examines the problem of estimating the location, velocity and manoeuvre of a manoeuvring target (from range, bearing and pose information). The report considers a non-linear modeling technique in which the target is represented as a hybrid system (a combination of discrete and continuous valued states) and considers new associated approaches such as the polymorphic estimator.

Although simulation studies were performed, the polymorphic estimator had serious numeric problems that suggested the estimator should not be used until the approach is refined. This report is intended to facilitate further discussion and development of the approach (if deemed necessary); hence, it includes details of the assumptions made and the Matlab^{TM} implementation.

APPROVED FOR PUBLIC RELEASE

Published by

DSTO Systems Sciences Laboratory
PO Box 1500
Edinburgh, South Australia, Australia 5111

Telephone: (08) 8259 5555
Facsimile: (08) 8259 6567

AR No. AR-012-723
April, 2003

APPROVED FOR PUBLIC RELEASE

Estimation of Manoeuvring Targets using Hybrid Filters

EXECUTIVE SUMMARY

Modern guided weapons are frequently required to operate in a complex environment that often involve highly complicated behaviours. In these types of engagements, assumptions of linearity no longer hold and model uncertainties in the form of unmeasured aerodynamic coefficients and complex non-linear aerodynamics are common. The target filter is an important sub-system of any guidance loop that estimates the required target and engagement information. To improve the performance of this target filter in challenging environments involving manoeuvring target, a full understanding of any non-linearities present is required.

The aim of this report is to investigate the manoeuvring target filtering problem, to examine the importance of mode measurements and to examine a particular filtering approach. A review of existing filtering results is provided before introducing a non-linear filtering approach known as hybrid filtering. Three possible filtering approaches are examined in simulation studies: the extended Kalman filter, the interacting multiple model filter and a new hybrid filtering approach. The simulation studies suggest that mode measurements may improve target filtering performance but the studies do not support the use of the examined hybrid filtering approach. Some refinement of this hybrid filtering approach is required.

An improved understanding of filtering techniques is required to aid support of present upgrade programs involving the guidance loops of new air-to-air and standoff missile systems. This understanding is necessary for the support of future weapon procurement and upgrade programs.

Authors

Jason J. Ford
Weapons Systems Division

Jason Ford joined the Guidance and Control group in the Weapons Systems Division in February 1998. He received B.Sc. (Mathematics) and B.E. degrees from the Australian National University in 1995. He also holds the PhD (1998) degree from the Australian National University. His thesis presents several new on-line locally and globally convergent parameter estimation algorithms for hidden Markov models (HMMs) and linear systems.

His research interests include: adaptive signal processing, adaptive parameter estimation, risk-sensitive techniques, non-linear filtering and applications of HMM and Kalman filtering techniques.

Peter G. Hunter
Weapons Systems Division

Peter Hunter joined the Guidance and Control Group in the Weapons Systems Division in September 1999. Peter holds a B.Sc. (Physics) from University of Adelaide (1994) and Associate Diploma (Communication Electronics) from the Royal Melbourne Institute of Technology (1974). Before employment at DSTO, Peter's career highlights include 12 months stationed at Mawson Antarctic for the Bureau of Meteorology, construction of prototype RF and digital circuitry with Codan Pty Ltd, Adelaide, and developing the electronic equipment calibration and maintenance program for the EEE Department, University of Adelaide.

Recently, Peter has been involved in the support of trials, data analysis, and the evaluation of various missile filtering and guidance algorithms.

Contents

1 Introduction **1**

2 Target Filtering Problem **3**

2.1 The Filtering Problem 3

2.2 Continuous State Model 3

2.3 Hybrid System Model 4

3 The Extended Kalman Filter **5**

4 Hybrid System Filtering **7**

4.1 Interacting Multiple Model Filter 7

4.2 Polymorphic Estimator 8

4.3 Other Filtering Approaches 10

5 Implementation Issues **11**

5.1 Direct Issues 11

6 Simulation Studies **13**

6.1 Stand Alone Filter Performance 13

6.2 Filters in a Missile Guidance Loop 14

6.3 Simulation Conclusions 15

7 Conclusion **15**

References **16**

Appendices

A Simulation Environment **18**

A.1 Path generation: Overview 18

A.2 Path Generation: Sub-functions 19

B Filtering Evaluation Code **21**

C Evaluation of Guidance Performance Code **25**

D Matlab^{TM} Syntax Headers **26**

E Simulation Parameter Values **36**

E.1 Stand-Alone Filter Evaluation 36

E.2 Filters in Guidance Loop 36

1 Introduction

Precision guidance of weapon systems is a computationally and conceptually demanding problem [2]. Historically, due to real-time computing constraints, major approximations in the control design process have been necessary. Recent advances in missile sub-systems mean that modern guided weapons have significantly improved computational capacity and hence various modelling approximations are being reconsidered [2].

One sub-problem of the missile guidance problem is estimation of the target position and velocity (and other quantities) from measurements such as range and bearing to the target. A common approach involves developing a system model of the relationship between the target state (position and velocity say) and the measurements available. Once a system model has been obtained, the target estimation problem can then be posed as a model based filtering (or optimal filtering) problem.

Across many fields of study, one of the more famous and commonly used system models describes the relationship between the states of a system and system measurements as a linear Gauss-Markov system with Gaussian noises. This model assumes linear dynamic behaviour of the internal system state (with perhaps an additive Gaussian noise disturbance) and measurements that are noisy linear functions of the state.

This linear Gauss-Markov system assumption is popular because, although most systems have no finite dimensional optimal filter, the Kalman filter has been shown to be the optimal filter for such a system [1, 5]. In this context, optimality is in the minimum mean squares sense and a filter is finite dimensional if it can be implemented using a finite number of statistics which can be calculated using a finite number of recursions. Because the Kalman filter is a finite dimensional optimal filter it has been applied to a large variety of filtering problems.

In a typical interceptor-target engagement, measurements are the relative range and bearing to the target (and perhaps pose information). These measurements are non-linearly related to the relative position and velocity of the target in cartesian co-ordinates. Because of the non-linearity in the measurement process, the Kalman filter is not appropriate for this problem (even under the assumption that the state dynamics of the target are Gauss-linear). Further, if the target is performing manoeuvres then a Gauss-linear model of the target dynamics is no longer appropriate.

For general non-linear problems, when a finite-dimensional optimal filter is not possible, sub-optimal numeric or approximate approaches must be used. The simplest approach is to use an extension of the Kalman filter known as the extended Kalman filter (EKF) [5, 1]. The EKF involves linearisation of the non-linear model about the current operation point. The EKF approach, although appealing due to its similarity to the Kalman filter, does not perform well in many non-linear filtering problems.

Some non-linear systems can be well represented by a non-linear model known as the hybrid system model (a model that contains a mixture of continuous and discrete valued states). Close to optimal finite dimensional filters have been designed that perform better than EKF approaches on these systems.

In this report we compare three non-linear filtering approaches to the problem of estimating the state of a manoeuvring target. The three approaches are a simple EKF approach (included as a benchmark), the interacting multiple model (IMM) filter [9], and the polymorphic estimate (PME) [18]. The IMM and PME are filtering approaches designed on an approximate hybrid model representation of the manoeuvring target. An important issue is whether these hybrid system models are realistic representations of manoeuvring target dynamics.

The key aim of this report is to examine the performance of these three filtering approaches in a manoeuvring target filtering problem to evaluate the apparent advantages of a hybrid system filtering approach. The second aim of this report is to examine these filters as part of a missile guidance loop. A third aim is to examine the influence of pose information on a missile guidance loop.

The report is structured as follows: In Section 2, we introduce several possible dynamic models and then introduce the target filtering problem. In Section 3, the extended Kalman filter solution to the problems is presented. In Section 4, the interacting multiple model filter and the polymorphic estimator are presented. In Section 5, some implementation issues are discussed. In Section 6, simulation results are presented. Finally, in Section 7, some conclusions are presented.

2 Target Filtering Problem

In this section we present the manoeuvring target tracking problem. Many formulations of the target tracking problem are commonly used. The formulation presented below allows several different types of dynamic models to be considered in a similar formulation. We first introduce the filtering problem, then introduce two target models.

2.1 The Filtering Problem

The manoeuvring target filtering problem stated in the broadest terms is to determine information about the state, usually denoted x_k, and perhaps the target manoeuvre, denoted here u_k^T, from measurements up until time k. In the context of the target tracking problem, we are usually interested in estimating the position, velocity and perhaps manoeuvre of the target from range and bearing measurements.

For the purpose of this report we are going to consider filtering and estimation from a model based perspective where estimation is according to a conditional mean criteria (we will limit our interest to statistics of the state such as the mean and variance).

An exact study of the properties of the presented filters is beyond the scope of this report. Various practical approaches are considered in the following sections but none of these filters are optimal. We will examine the proposed approaches according to their performance in a target tracking problem.

2.2 Continuous State Model

Consider the following non-linear model for engagement with a manoeuvring target:

$$\begin{aligned} x_{k+1} &= a_k(x_k, u_k^T, u_k^I, v_k) \\ y_k &= c_k(x_k, w_k) \end{aligned} \tag{2.1}$$

where $x_k \in R^N$ is a state describing both the target and interceptor dynamics (or perhaps the relative dynamics between the two), $u_k^T \in \mathcal{U}(x_k)$ is the manoeuvre performed by the target which is an unmeasured input, u_k^I is the manoeuvre performed by the interceptor which is assumed to be known, $y_k \in R^M$ is the observation, v_k is a process noise process, and w_k is a measurement noise process. The set $\mathcal{U}(x_k)$ is the complete set of target manoeuvres that can be performed from state x_k.

This model is generic enough to include general target manoeuvres, general measurement processes, and complicated aerodynamics. The problem considered in this report is estimation of the state, x_k, from the measurements y_k. The general filtering problem for the above non-linear system with unmeasured u_k^T has not been solved in analytic form (there are several numeric approaches such as the particle filter which are outside the scope of this report and not considered here).

In some situations it is reasonable to use a more restrictive stochastic process to describe u_k^T and to assume linear state dynamics as follows:

$$\begin{aligned} x_{k+1} &= Ax_k + B^T u_k^T + B^I u_k^I + v_k \\ y_k &= c_k(x_k) + n_k \end{aligned} \qquad (2.2)$$

where $A \in R^{N \times N}$, v_k and n_k are Gaussian noise processes, and u_k^T is a stochastic process dependent on x_k and u_{k-1}^T.

This filtering problem is also difficult and no analytic solution exists. The model (2.2) is appropriate for an extended Kalman filter solution to the target estimation problem. We will discuss this approach later but first we introduce further restrictions on the input that lead to a hybrid system model.

2.3 Hybrid System Model

In this subsection we consider a further restriction on u_k^T that leads to a hybrid system model and leads to two alternative filtering approaches. Consider the case that $\mathcal{U}(x_k)$ is restricted to be a finite set (whose elements correspond to distinct possible target accelerations) and further that u_k^T is a Markov process. With this finite set restriction, assuming linear state dynamics, the following model is obtained:

$$\begin{aligned} u_k^T & \quad \text{is a first order Markov chain} \\ x_{k+1} &= \bar{A}(u_k^T)x_k + B^I u_k^I + v_k \\ y_k &= c_k(x_k) + w_k. \end{aligned} \qquad (2.3)$$

where u_k^T is a finite process that takes values from a discrete set (of size N_U) whose values approximate the behaviour of the target. Here each $\bar{A}(.)$ indexed by u_k^T describes the target dynamics for each of the possible target manoeuvres. For example, $\bar{A}(0) = A$ where 0 indicates the target is not performing a manoeuvre etc. It should be noted that this target acceleration model, (2.3), only holds when the possible target control actions are restricted to perpendicular accelerations.

This system is considered a hybrid system model because it contains a mixture of continuous valued and discrete valued states (x_k and u_k^T respectively).

The motivation in considering a hybrid system approximation for the continuous system (2.1) stems from the knowledge that "close to optimal" filtering solutions for hybrid systems exist. The meaning of "close to optimal" will be discussed in a later section. A key question is whether a "close to optimal" solution on the hybrid system approximation achieves better performance than an approximate solution based on a continuous system model. This issue is investigated in Section 6 via simulation studies.

The system model (2.3) is appropriate for both the interacting multiple model and the polymorphic estimator (both are discussed in Section 4).

3 The Extended Kalman Filter

The Kalman filtering is the optimal filter for a linear Gauss-Markov system. System (2.2) is linear Gauss-Markov when $A(u_k) = A_k$, $c_k(x_k) = C_k x_k$ and v_k,w_k are independent sequences of Gaussian noise. The Kalman filter is optimal in the sense that it produces estimates, $\hat{x}_{k|k}$, that minimise

$$E[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})'|y_0, \ldots, y_k].$$

The extended Kalman filter extends the concept of the Kalman filter to a non-linear system model via linearisation. The extended Kalman filter (analogous to the Kalman filter) calculates a state estimate, a covariance matrix, *a priori* state estimate and *a priori* covariance matrix at each time instant ($\hat{x}_{k|k}$,$P_{k|k}$, $\hat{x}_{k|k-1}$ and $P_{k|k-1}$ respectively).

To develop the extended Kalman filter for this problem we first assume that $\hat{u}_k^T$ is available (this assumption will be relaxed later). Let us define the following quantities:

$$A_k \triangleq A(\hat{u}_k^T), \quad \text{and} \tag{3.1}$$

$$C_k \triangleq \left.\frac{\partial c_k(X)}{\partial X}\right|_{X=\hat{x}_{k|k-1}}. \tag{3.2}$$

Here $A_k \in R^{(N\times N)}$ and $C_k \in R^{(M\times N)}$.

Let us also introduce matrices Q_k^* and R_k^* which are the covariance matrices for noises w_k and v_k.

The extended Kalman filter is implemented using the following equations [1, 5]:

$$\begin{aligned}
\hat{x}_{k|k-1} &= a_{k-1}(\hat{x}_{k-1|k-1}, \hat{u}_k^T, u_k^I, v_k) \\
P_{k|k-1} &= A_{k-1}P_{k-1|k-1}A'_{k-1} + Q_k^* \\
K_k &= P_{k|k-1}C'_k\left[C_kP_{k|k-1}C'_k + R_k^*\right]^{-1} \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k\left[y_k - c_k(\hat{x}_{k|k-1})\right] \\
P_{k|k} &= P_{k|k-1} - K_kC_kP_{k|k-1}
\end{aligned} \tag{3.3}$$

Given a sequence of measurements, y_k, the extended Kalman filter provides a sequence of estimates, $\hat{x}_{k|k}$, and an estimate of error covariance $P_{k|k}$. Although in non-linear estimation problems higher order moments can be significant, the EKF only keeps track of 1st and 2nd moment information ($\hat{x}_{k|k-1}$,$P_{k|k-1}$).

The above implementation assumes that an estimate of the target manoeuvre is available. Often a target manoeuvre estimate will not be available and a common strategy is to then assume the target is not manoeuvring. This sort of approximation about the target manoeuvre significantly affects the performance of the filter and is the prime motivation for considering the two filters described in the next section.

Another possible alternative is to include the target manoeuvre as part of the system state. However, when formulated this way the system model becomes highly non-linear and the extended Kalman filter itself (which is based on linearisations) is likely to be unstable when initialised poorly.

4 Hybrid System Filtering

In this section two separate approaches to the filtering problem for hybrid systems are discussed. The first filtering approach considered here is the interacting multiple model filter which is based on the concept of a running bank of filters (one for each of the possible manoeuvre modes) and then combine the output of each filter to obtain an estimate of the target state. The second filter considered is the Polymorphic Estimators which is based on a sub-optimal approximation to the optimal solution of the hybrid system filtering problem.

4.1 Interacting Multiple Model Filter

The interacting multiple model filter is a filtering approach (based on the model (2.3)) that involves running a sub-filter for each of the distinct manoeuvres modelled [9]. At each time step, each individual sub-filter uses any new measurements, mode probability information and previous sub-filter outputs to generate a new estimate based on the sub-filter's assumed manoeuvre. Then the output all these sub-filters is combined according to certain probability information to produce a new current overall target estimate. This is explained in more detail below.

Let $\hat{x}^i_{k-1|k-1}$ and $P^i_{k-1|k-1}$ be the outputs of the ith sub-filter at time $k-1$ and let $\bar{A}^{ji}$ be the transition probability from manoeuvre i to manoeuvre j. Also, let μ^j_{k-1} denote the estimated probability of being in mode j at time $k-1$.

Then

$$\begin{aligned}
\mu^{i|j}_{k-1} &= c^j \bar{A}^{ji} \mu^i_{k-1} \\
\hat{x}^{j0}_{k-1|k-1} &= \sum_{i=1}^{N+U} \hat{x}^i_{k-1|k-1} \mu^{i|j}_{k-1} \\
P^{j0}_{k-1|k-1} &= \sum_{i=1}^{N_U} \mu^{i|j}_{k-1} \left(P^i_{k-1|k-1} + (\hat{x}^i_{k-1|k-1} - \hat{x}^{j0}_{k-1|k-1})(\hat{x}^i_{k-1|k-1} - \hat{x}^{j0}_{k-1|k-1})' \right) \quad (4.1)
\end{aligned}$$

where $\hat{x}^{j0}_{k-1|k-1}$ and $P^{j0}_{k-1|k-1}$ are the inputs to the jth sub-filter at the kth cycle used below, $\mu^{i|j}_{k-1}$ is the probability that a particular transition of manoeuvre occured, and c^j is a normalisation constant (from $\mu^{i|j}_{k-1}$ over i).

Then the N_U sub-filters iterate as follows

$$\begin{aligned}
\hat{x}^j_{k|k-1} &= a_{k-1}(\hat{x}^{j0}_{k-1|k-1}, u^T_k = j, u^I_k, v_k) \\
P^j_{k|k-1} &= A_{k-1}(j)P^{j0}_{k-1|k-1}A_{k-1}(j)' + Q^j *_k \\
K^j_k &= P^j_{k|k-1}C'_k\left[C_kP^j_{k|k-1}C'_k + R^j *_k\right]^{-1} \\
\hat{x}^j_{k|k} &= \hat{x}^j_{k|k-1} + K^j_k\left[y_k - c_k(\hat{x}^j_{k|k-1})\right] \\
P^j_{k|k} &= P^j_{k|k-1} - K^j_kC_kP^j_{k|k-1}
\end{aligned} \tag{4.2}$$

where C_k was defined in the last section and $A_k(j) \triangleq A(u^T_k = j)$.

The mode probability can be calculated as

$$\begin{aligned}
\Lambda^j_k &= P\left(y_k|u_k = j, \hat{x}^{j0}_{k-1|k-1}, P^{j0}_{k-1|k-1}\right) \\
\mu^j_k &= c\Lambda^j_k\sum_{i=1}^{N_U}\bar{A}^{ji}\mu^i_{k-1}
\end{aligned} \tag{4.3}$$

where c is a normalisation constant.

The overall estimate of the overall scheme at any time is given by

$$\begin{aligned}
\hat{x}_{k|k} &= \sum_{j=1}^{N_U}\hat{x}^j_{k|k}\mu^j_k \\
P_{k|k} &= \sum_{j=1}^{N_U}\mu^j_k\left(P^j_{k|k} + (\hat{x}^j_{k|k} - \hat{x}_{k|k})(\hat{x}^j_{k|k} - \hat{x}_{k|k})'\right)
\end{aligned} \tag{4.4}$$

Note that the IMM filter keeps track of only 1st and 2nd order moment quantities $(\hat{x}^j_{k|k}, P^j_{k|k})$.

4.2 Polymorphic Estimator

Unlike the EKF and IMM filters presented above, the polymorphic estimator keeps track of several higher order statistics to improve the quality of state estimates for the hybrid system (2.3). By studying the optimal filtering problem for (2.3) and making approximations in equations involving 4th and 5th order moments, filters with improved performance can be obtained. We provide no other details about the development of the PME and the reader is directed to [18] for any more information. Here we simply present the filter.

To simplify the presentation, we consider the PME as a two step process: a time update step, and a measurement update step.

Time Update

In between measurements, the statistics of the hybrid system evolve according to the underlying dynamics of the system. For this reason, the time update step of the PME is typically done via numeric integration of the following differential equation. In general, the evolution of these equation can not be implemented as difference equations.

$$\begin{aligned}
\mu_{t+\delta t} &= \bar{A}\mu_t \\
\frac{d}{dt}\hat{x}_t &= \sum_{i=1}^{N_U} A_t(i)\left(\hat{x}_t\mu_t^i + P_{x\mu}^{(.i)}\right) \\
\frac{d}{dt}P_{x\mu} &= P_{x\mu}\bar{A}_c + \sum_{i=1}^{N_U} A_t(i)\left(\hat{x}_t P_{\mu\mu}^{(i.)} + Px_{\mu\mu^i} + P_{x\mu}\mu_t^i\right) \\
\frac{d}{dt}P_{xx} &= \sum_{i=1}^{N_U} A_t(i)\left(\hat{x}_t P_{\mu x}^{(i.)} + Px_{xx\mu^i} + P_{xx}\mu_t^i\right) + R(i)\mu_t^i \\
\frac{d}{dt}P_{xx\mu^m} &= \sum_{i=1}^{N_U} A_t(i)P_{(x\mu^i)x\mu^m} + P'_{(x\mu^i)x\mu^m}A_t(i)' + P_{xx\mu^i}Q^{im}
\end{aligned} \tag{4.5}$$

where $A^{(i.)}$ is the ith row of A and $A^{(.i)}$ is the ith column of A. Note that

$$\begin{aligned}
P_{\mu\mu} &= diag(\mu_t) - \mu_t\mu_t' \\
P_{x\mu} &= P'_{\mu x} \\
P_{x\mu\mu^i} &= P_{x\mu^i}(e_i - \mu_t)' + \mu_t^i P_{x\mu} \\
P_{(x\mu^i)x\mu^m} &= P_{xx\mu^i\mu^m} + \mu_t^i P_{xx\mu^m} + \hat{x}_t P_{\mu^i x\mu^m} - P_{x\mu}^{(.i)}\left(P_{x\mu}^{(.m)}\right)' \\
P_{\mu^i x\mu^m} &= P_{x\mu\mu^m}^{(.i)} \quad \text{and} \\
P_{xx\mu^i\mu^m} &= \begin{cases} P_{xx\mu^m} + P_{xx}\mu^m - \mu^m P_{xx\mu^i} - \mu^i P_{xx\mu^m} - P_{xx}\mu^i\mu^m & \text{if } i = m \\ -\mu^m P_{xx\mu^i} - \mu^i P_{xx\mu^m} - P_{xx}\mu^i\mu^m & \text{otherwise} \end{cases}
\end{aligned} \tag{4.6}$$

where $diag(X)$ is the diagonal matrix with X on its diagonal.

Measurement Update

If either range and bearing or pose information becomes available, then the state estimate of the hybrid system filter can be updated as shown in the following two sub-sections. Note that if both types of measurements become available at the same time instant then the update steps can be performed sequentially. Although the following equations are generally implemented as discrete time recursions, to ensure conformity with the time update equations, we have expressed the updates as changes from $\hat{x}_t^-$ to $\hat{x}_t^+$ etc. at the discrete time instants that measurements occur.

Mode Measurements Assuming that there are measurements of the target pose y_t^m, this information can be incorporated into the filter estimates. Here $y_t^m \in \{e_1, \ldots\}$ is a discrete value representation of the target pose (for example, the target orientation in the yaw-plane).

$$\begin{aligned}
\bar{y}_t^m &= \bar{A}^m y_t^m \\
\mu_t^+ &= \mu_t^- * \bar{y}_t^m; \qquad \Delta\mu = \mu_t^+ - \mu_t^- \\
\hat{x}_t^+ &= \hat{x}_t^- + P_{x\mu}\bar{y}_t^m; \quad \Delta x = P_{x\mu}\bar{y}_t^m \\
P_{x\mu} &= P_{x\mu} - \Delta x \Delta\mu' + \sum_{i=1}^{N_U} P_{x\mu\mu^i}\bar{y}_t^m(i) \\
P_{xx} &= P_{xx} - \Delta x \Delta x + \sum_{i=1}^{N_U} P_{xx\mu^i}\bar{y}_t^m(i) \qquad (4.7)
\end{aligned}$$

where the ijth element of $\bar{A}^m$ is the probability of being in mode i given that the target pose is j, $\bar{y}_t^m$ is a pseudo measurement of target mode, and $*$ is the operation of element-by-element multiplication.

Range/bearing Measurements Range and bearing information, y_t, can be used to update the target state information as follows

$$\begin{aligned}
\gamma_x &= P_{xx}C'(CP_{xx}C' + R)^{-1} \\
\hat{x}_t^+ &= \hat{x}_t^- + \gamma_x y_t \\
P_{x\mu} &= P_{x\mu} - \gamma_x C P_{x\mu} \\
P_{xx} &= P_{xx} - \gamma_x(CP_{xx}C' + R)\gamma_x' \qquad (4.8)
\end{aligned}$$

4.3 Other Filtering Approaches

There are many other non-linear filtering approaches that may be considered appropriate for this problem. The robust Kalman filtering approach may be considered an alternative to the extended Kalman filtering approach [16]. More success can be obtained from non-model based approaches such as Monte Carlo approaches [15] and the particle or auxiliary particle filter approach [17].

Also of interest is the collection of approaches to the hybrid filtering problem. There are many sub-optimal hybrid filtering approaches available and this paper only had time to examine one. Some of the other hybrid filtering approaches many be worth considering.

None of these alternative approaches were examined in the context of this simulation study, but previous work suggests that the particle filter approach may offer some advantages (once the technique has become well developed). Further comment is outside the scope of this report.

5 Implementation Issues

This section discusses some of the implementation issues associated with the above filters. In the first sub section we will discuss some issues directly. Further information about the Matlab^{TM} implementation and the syntax and header files can be found in Appendix A.

5.1 Direct Issues

Implementation of the extended Kalman filter has been covered by many other authors (See [5] for example), and implementation issues with respect to the IMM are well understood [9] so this section will concentrate on issues related to the Polymorphic Estimator.

Numeric Stability

In our studies, the Polymorphic Estimator was found to have serious numeric stability problems. The most significant of these problems was the instability in the time update step of the filter. The time update step requires numeric integration of a non-linear system of equations. In our implementation of the filter, a Euler approximation was used for integration purposes.

It was found that unless the step-size was reduced to $h \leq 0.001$ the filter would often update P_{xx} and other matrices to non-positive definite matrices (non-valid updates). While decreasing the step-size did reduce the number of non-valid updates, the incidence of non-valid updates was not completely removed. When this positive definite property is lost, the filter quickly diverges.

Alternative implementations, that were not tried, may have reduced the incidence of non-valid updates these include: reformulation of the update equations to maintain the positive definiteness property, and the use of other numeric integration techniques.

Instead, the implemented code checks for non-valid updates of the matrices and attempts to correct the updates and keep the filter stable. In simulation studies in Section 6, simulations involving situations where the PME had extensive numeric problems have been excluded. Hence, the reader must consider the results as best case results, indicative of the sort of performance that may be possible for a hybrid system filtering approach, rather than an endorsement of the PME filter.

The numeric stability problems on the PME were found to be significant enough to exclude use of the filter in a practical environment.

Target Motion Assumptions

The target is assumed to exhibit piece-wise perfect linear and perfect turning motion. The turn rate of the target was assumed to be perfectly known by the filter, but the timing of turning events is not.

These target motion assumptions as quite unrealistic. In real problems the turning rates will neither be known nor be constant. An extensive investigation of situations involving model mismatch is required before any hybrid system filtering approach could be applied to a real problem. Such examinations have been performed for the IMM filter [9]. However, in this study, the computation effort required to implement the PME filter was too significant for us to test a large set of engagements and hence performance during mismatch was not examined.

Inteceptor Motion Assumptions

A simple augmented proportional navigation guidance law without either guidance or autopilot lags was used to evaluate the filters as part of a guidance loop.

Matlab^{TM} Implementation

Details of the Matlab^{TM} implementation of the filters are presented in Appendix A

6 Simulation Studies

In this section we examine the three presented filters in an engagement against a target performing a series of manoeuvres. We first consider the error performance of the filters (under no guidance) and then we investigate the miss distance performance when these filters are used in a missile guidance loop.

6.1 Stand Alone Filter Performance

To examine the performance of the filters, the EKF, IMM and PME were used to filter data in a total of 1249 simulations. Details of the parameter values used in these simulations are provided in Appendix A. The EKF with knowledge of the actual target mode was also simulated. Because all the filters in this study are EKF based the EKF filter with knowledge of the actual target mode provides a lower bound on performance.

As discussed in the previous section, the PME filter sometimes has numeric difficulty in the P_{xx} update step. The implemented version of the PME filter monitors the P_{xx} update step and provides a count of the number of bad update steps. The simulation studies suggest that when more than 5 bad P_{xx} updates occur the performance of the P_{xx} can be quite poor.

Although, the performance of the PME was quite unsatisfactory, some selected results are presented in the tables below to demonstrate what performance gains might be possible if the numeric problems can be resolved. In Table 1, the average error performance of the filters are compared on the 75 data sets for which the PME had no bad P_{xx} updates.

The average mean error values presented in the table is the average error over the last 100 points of the selected simulation run. The variance of error supplied in the table is the variance of individual simulation errors across the selected set. In Table 2, the average error performance of the filters are compared on the 244 data sets for which the PME had one or less bad P_{xx} updates. The other 1005 runs have greater than one bad update.

Table 1: Filter results with no bad P_{xx} updates

Filter	Average Mean Error	Variance of Error
EKF(truth)	33.1 m	82.9 m^2
EKF	569.0 m	446.9 m^2
IMM	118.3 m	1380.9 m^2
PME	72.6 m	332.9 m^2

Table 2: Filter results with one or less bad P_{xx} updates

Filter	Average Mean Error	Variance of Error
EKF(truth)	32.6 m	70.2 m^2
EKF	574.7 m	1400.2 m^2
IMM	116.0 m	992.7 m^2
PME	75.9 m	1257.3 m^2

6.2 Filters in a Missile Guidance Loop

To evaluate the performance of the PME and the influence of having pose measurements of the target available in the missile guidance problem, the performance of the PME with various amounts of target pose information was compared to the performance of a guidance loop using an EKF that knows the present target manoeuvre.

Table 3 shows the achieved miss-distance in 5 guidance loop configurations. Details of the parameter values used in these simulations are provided in Appendix A. To examine how the availability of mode measurements influence the performance, the PME was examined with four different amounts of mode information. The scenarios considered were:

- the PME with mode measurements for the complete simulation,
- the PME with no mode measurements available,
- the PME with mode measurements available when range was less than 1500 m, and
- the PME with mode measurements available when range was less than 750 m.

The achieved miss-distances and the count of P_{xx} bad updates are provided in Table 3.

Due to the large number of P_{xx} bad updates it is not possible to determine the influence of mode measurement on guidance performance.

Table 3: Guidance Results

Filter	Miss distance	P_{xx} Bad Updates
EKF(truth)	0.8514 m	n/a
PME(full mode)	7.9520 m	2
PME(no mode)	30.6579 m	5500
PME(1500m mode)	9.0632 m	5501
PME(750m mode)	10.6382 m	5500

6.3 Simulation Conclusions

The numeric problems of the PME are a very serious limitation. Due to the very long time it took to perform the simulations (2 weeks for a full simulation set) it was not possible to work towards a numerical stable implementation of the filter. However, by looking at the performance of the filter in the sub-set of simulations that P_{xx} was updated correctly, it seems the filter may offer performance advantages over the IMM and EKF approaches. This may also suggest that mode measurements can involve target filter performance.

The simulations on the performance of the filter in a missile guidance problem were not conclusive because of the numeric problems. The study done here was not complete and significant work is required on the PME filter before it could be considered fully examined.

7 Conclusion

This report provides a description of recent work examining the use of hybrid filters including the Polymorphic Estimator (PME) for the estimation of manoeuvring targets. Details of the filtering problem and standard approaches such as the extended Kalman filter (EKF) and the interacting multiple model (IMM) filters were provided. The value of mode measurement (or target pose information) was examined and simulation studies described.

The PME, as presented in the literature, was found to have very serious numeric problems that hindered a full evaluation of the hybrid filtering approach. The simulation studies suggested that the PME (and the use of mode measurements) may offer performance advantages over the well known EKF and IMM approaches; however, significantly more work (including the development of numerically stable versions of the filter) is required before the examination of the PME could be considered complete.

References

1. B.D.O. Anderson and J.B. Moore, *Optimal Filtering*, Prentice Hall, New Jersey, 1979.

2. F.A. Faruqi, "Optimum Precision Guidance - Development of Target Interceptor Kinematics Model and Optimum Performance Index", *DSTO Series Report*, File: J 95D5-17-223 - to be issued.

3. P.S. Maybeck, *Stochastic Models, Estimation, and Control*, Vol. 1, Academic Press, New York, 1979.

4. A.H. Sayed and T. Kailath, "A State-Space Approach to Adaptive RLS Filtering," *IEEE Signal Processing Magazine*, pp. 18-60, July 1994.

5. R.G. Brown and P.Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 2nd Ed., John Wiley & Sons, New York, 1992.

6. A.H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic, New York, 1972.

7. B.F. La Scala, R.R. Bitmead and M.R. James, "Conditions for stability of the Extended Kalman Filter and Their Application to the Frequency tracking Problem," *Mathematics of Control, Signals, and Systems*, Vol. 8 pp. 1-26, 1995.

8. K. Reif, S. Günther, E. Yaz and R. Unbehauen, "Stochastic Stability of the Discrete-Time Extended Kalman Filter," *IEEE Trans. on Automatic Control*, Vol. 44, No. 4, April 1999.

9. Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press, Boston, 1988.

10. S.C. Krammer and H.W. Sorenson, "Recursive Bayesian estimation using piece-wise constant approximations," *Automatica*, Vol. 24, pp. 789-801, 1988.

11. H. Nijmeijer and A.J. van der Schaft, *Nonlinear Dynamical Control Systems*, Springer, New York, 1990.

12. I. Marceels and J. W. Polderman, *Adaptive Systems: An Introduction*, Birkhauser, Boston, 1996.

13. L. Ljung, *System Identification: Theory for the User*, Prentice Hall, London, 1987.

14. T.M. Cover and J.A. Thomas, *Elements of Infromation Theory*, John Wiley & Sons, New York, 1991.

15. B.P. Carlin, N.G. Polson, D.S. Stoffer, "A Monte Carlo approach to nonnormal and nonlinear state space modeling," *Journal of the American Statisitcal Assocation*, Vol. 87, pp. 640-647, 1992.

16. I. R. Petersen and A. V. Savkin, *Robust Kalman filtering for signals and systems with large uncertainties*, Birkhuser, Boston Mass., 1999.

17. A. Doucet, N. de Freitas, N. Gordon, "Sequential Monte Carlo Methods in Practice," Springer-Verlag, 2001.

18. D.D. Sworder and J.E. Boyd, *Estimation Problems in Hybrid Systems*, Cambridge University Press, Cambridge, 1999.

Appendix A Simulation Environment

In these appendices we describe the operation of the filter in terms of the notation used to implemented the filter in the Matlab^{TM}. The Matlab^{TM} code was used to examine the PME, IMM and EKF filters. The filters were examined in two problems: As a stand alone filter (representing measurements from the origin), and as part of a missile guidance loop. Below we provide a discussion of the implementation of each simulation separately.

A.1 Path generation: Overview

A trajectory of a missile is simulated in Matlab^{TM}. At each time instant, the kinematic state of the missile is represented by a state vector, $[x; y; vx; vy]$, (x,y position and x,y velocity) which are stored as columns in the array, Xk.

The missile in the simulation performs a series of manoeuvres, including straight sections and turning left or turning right segments. In the code presented above the missile target performs 5 manoeuvres on its flight toward the origin :

- turn left (5sec),
- straight (6sec),
- turn right (5sec),
- straight(3sec) and
- turn left (5sec).

The target begins at an initial state, x_s (with units m and m/s) and all turns are with the same, $\pm phi$ turn rates (units rad/s).

Simulated measurements of the target from the coordinate origin are also generated, including:

- A simulated radar producing noisy measurements of range and bearing.
- A simulated imager (imager1.m) that measures the target orientation under noise. Target orientation is specified to a 30° sector (bin) numbered $1 - 12$ from North.

The radar measurements and orientation are combined and produce a vector of measurements and are stored in Ymk ([range;bearing;bin]). The mode (or current manoeuvre) of the target is recorded as *modev*.

A.2 Path Generation: Sub-functions

In the implementation, the whole trajectory of the missile is simulated by combining simulated trajectories for each distinct manoeuvre. For each manoeuvre the function asm.m generates a discrete series of states ($xk1$) with sampling time (h), typically $0.001s$. Internally, the complete path state record is stored as the vector Sk. Measurements are available with sampling time (hm), typically 0.1 s, in the vector Ymk.

The initial state (xkm) is evolved at each time instant k using the following state equation:

$$xk1 = Ah \times xkm + G \times w$$

where xkm is the state vector for the previous sample time; Ah is a 4×4 matrix selected to produce a path which is straight or turning left or right at typically 0.2rad/s; G is the covariance matrix of the noise components; and w is the process noise, set to zero for all simulations.

At each measurement instant, i, a rectangular - polar coordinate conversion is performed giving the target range

$$Ck(1,i) = \sqrt{(Sk(1, i \times ns))^2 + (Sk(2, i \times ns))^2};$$

and bearing

$$Ck(2,i) = atan(Sk(2, i \times ns), Sk(1, i \times ns));$$

where $ns = hm/h$ must be an integer and i is the measurement interval (hm) index. To these range and bearing values measurements noise is added:

$$Vk(1,i) = \sqrt{rvar} \times randn; \quad \text{(Range measurement noise)}$$

and

$$Vk(2,i) = \sqrt{tvar} \times randn; \quad \text{(Bearing measurement noise)}$$

Here $rvar$ is the range variance, $tvar$ is the bearing variance and $randn$ is Matlab's normal distributed random number generation. The radar measurements are recorded in $Ymk(1:2,i) = Ck(:,i) + Vk(:,i);$.

For each measurement, i, imager1.m works out the orientation of the target (θ) and assigns a bin number (bin). Imager errors are introduced randomly to simulate the following :

- Total occlusion (UDE - 5% chance) - selects any bin number at random.
- Nearest neighbour (NNE - 5% chance) - assigns one of the bins either side of the correct one.

The output bin is recorded as $Ymk(3, i)$.

At the completion of each manoeuvre the state record (Xk), observations (Ymk), mode record ($modev1$) are stored. The final state (xkm) of each manoeuvre segment becomes the initial state of the next manoeuvre. The successive Xk, Ymk and modev1 are concatenated to produce a complete state, measurement and mode record for the path.

Appendix B Filtering Evaluation Code

The following subsections describe the various routines used to compare the EKF, IMM and PME filter on the target filtering problem.

initcommon.m Contains initialization and definitions for a number of parameters and matrices which are common to all the filters. These include sample time (h), measurement time (hm), turn rate (phi) and state transition matrices (Ac, Ad). The following vectors and matrices are also initialised: initial target state (xs),initial target state estimate (xhs), measurement and process noise covariance matrices (Rn, Qn), and the prior state error covariance matrices (Pn).

The quantity Ac is $(4 \times 4 \times 3)$ 3d-matrix where the third index corresponds to the selection of a 2d-matrix representing the continuous-time state transition matrices corresponding to turning left, straight flight, and turning right respectively. The quantity Ad is similarly used to represent three discrete-time state transition matrices.

The measurement and process noise covariance matrices (Rn, Qn) are individually stored for EKF, IMM and PME filters. The prior state error covariance matrix, Pn, is initialised with errors of 100 m in position and 10 m/s in velocity for all runs (unless a larger state error is input via xhs).

EKF filters The EKF algorithm, as described in (3.3) was implemented with notation changes to allow for the restricted text format options in the Matlab^{TM} code editor. These changes are:

Maths Terms	Matlab^{TM} Term	Notes
$\hat{x}_{k\|k-1}$	xh	
$\hat{x}_{k\|k}$, $\hat{x}_{k-1\|k-1}$	xhm	xhm is the output of one iteration ($\hat{x}_{k\|k}$) and also the input of the loop($\hat{x}_{k-1\|k-1}$).
A_{k-1}	Ahm	
$P_{k\|k-1}$	Pm	
$P_{k\|k}$, $P_{k-1\|k-1}$	Pm	Note: Pm is both the output of one iteration ($P_{k\|k}$) and the input of the next loop ($P_{k-1\|k-1}$).

C_k	C	
K_k	Kk	
y_k	Ymk	
$c_k\hat{x}_{k\|k-1}$	$yhat$	
R_k	R	
Q_k	0	we assume there is no process noise

At each measurement instant the appropriate state transition matrix $Ad(:,:,i)$ is selected (assuming the EKF has access to the true target manoeuvre state. The variable prf is used to switch between two measurement situations. If $prf = 10$ then the measurement update occurs at hm. If $prf =$ any other value then the measurement update occurs at $10 \times hm$. The variable $xhatv$ is the complete record of the estimated state output of the filter, at the rate dictated by hm.

For the ekfc.m version, no knowledge of the target manoeuvre is assumed, so the state transition matrix corresponding to straight flight $Ad(:,:,2)$ is used.

IMM filter The IMM filter, as described in (4.1)-(4.4), was implemented with notation changes to allow for the restricted text format options in the Matlab^{TM}. These changes are:

Maths Terms	Matlab^{TM} Term	Notes
$\mu_{i\|j}(k-1)$	$mu2$	Equation (4.1)
$\hat{x}^{j0}_{k-1\|k-1}$	xoj	Equation (4.1)
$P^{j0}_{k-1\|k-1}$	Poj	Equation (4.1)
$\hat{x}^{j}_{k\|k-1}$	xjh	Equation (4.2)
$P^{j}_{k\|k-1}$	Pj	Equation (4.2)
K^{j}_{k}	Kk	Equation (4.2)
$\hat{x}^{j}_{k\|k}$	$xhjk$	Equation (4.2)
$P^{j}_{k\|k}$	Pjk	Equation (4.2)
Λ^{j}_{k}	Ljk	Equation (4.3)

μ_k^j	mu	Equation (4.3)
$\hat{x}_{k\|k}$	xkk	Equation (4.4)
$P_{k\|k}$	Pkk	Equation (4.4)

It is assumed in the initial definition of the mode probability (mu) that the the target is equally likely to initially be in one of the three possible modes (straight path, turning left or right). It is also assumed that the target will continue in the same mode (manoeuvre) or the target can switch, with equal probability, to any adjacent manoeuvre mode (ie. the model does not allow for direct transition between left and right turn modes). This mode switch is random and described by a probability transition matrix introduced in the PME description below.

PME filter The PME filter, as described in (4.5) -(4.8), was implemented with notation changes to allow for the restricted text format options in the Matlab^{TM}. These changes are:

Maths Terms	Matlab^{TM} Term	Notes
$\mu_{t+\delta t}$	fhm	Time update equations
mu_t	fh	
$\frac{d}{dt}\hat{x}_t$	$dx \times deltaT$	
$\frac{d}{dt}P_{x\mu}$	$dPxf \times deltaT$	
$\frac{d}{dt}P_{xx}$	$dPxx \times deltaT$	
$\frac{d}{dt}P_{xx\mu^m}$	$dPxxfm \times deltaT$	
$P_{\mu\mu}$	Pff	
$P_{x\mu}$	Pxf	
$P_{x\mu\mu^i}$	$Pxffi$	
$P_{(x\mu^i)x\mu^m}$	$Pxfixfm$	
$P_{\mu^i x\mu^m}$	$Pfixfm$	
$P_{xx\mu^i\mu^m}$	$Pxxfifm$	
$\tilde{y}_t^m$	ybm	Measurement update equations

μ_t^+	fh	
$\hat{x}_t^+$	xh	
Δx	dxh	
$P_{x\mu}$	Pxf	
P_{xx}	Pxx	
γ_x	Kk	Range/bearing measurements update equations

This filter is based on the algorithm presented in by Boyd and Sworder. A section of Boyd and Sworder's version of the Matlab^{TM} code is incorporated using wrapping code to translate between the different variable names used in their implementation.

In defining a hybrid system description appropriate for the PME filter several design assumptions were made. It was assumed that the target was able to perform 3 types of manoeuvre (straight flight, left turn, right turn). The probability of transitioning between manoeuvre modes is described by the (3×3) transition probability matrix Aa . The elements of Aa are such that $A(i, j)$ represents the probability of changing from mode j to mode i.

At each time instant the target orientation is one of twelve values (corresponding to the orientation bins). The probability of changing orientation bin, when that target is in a particular manoeuvre mode, is described by 3 (12×12) matrices, As, Al and Ar.

It was decided to use a composite discrete state vector, fh, to represent both the present manoeuvre mode and orientation of the target. This vector represents all combinations of the 12 orientations and 3 manoeuvre modes so fh is a (36×1) vector. The transition probability matrix, A, required by the PME is hence a (36×36) matrix made up from Aa, As, Al and Ar.

The PME filter also requires an observation mapping matrix, Pb, which describes the probability of receiving ybm given that fh has a particular value. Because the orientation observation is independent of the target manoeuvre, the mapping matrix, Pb, contains three repeated blocks, Po.

Appendix C Evaluation of Guidance Performance Code

To evaluate the filter as part of a guidance loop we simulated a missile guidance problem. In the missile guidance loop, the missile observes the target through a range and bearing measurements. The missile uses these measurements to estimate the state of the target and then uses these state estimates to guide towards the target.

The guidance algorithm used in these guidance loop simulations is the augment proportional navigation algorithm:

$$\bar{u}_t = -3(V_c\dot{\lambda} + \frac{1}{2}d). \tag{C1}$$

where d is the target acceleration perpendicular to the line-of-sight line connecting the missile and the target; V_c is the closing velocity; $\dot{\lambda}$ is the line-of-sight rate seen by the missile; and $\bar{u}_t$ is the control action.

In previous sections we have introduced all the information necessary to construct the state estimates so this is not repeated here. The control algorithm described above was implemented with notation changes to allow for the restricted text format options in the Matlab^{TM} code. These changes are:

Maths Terms	Matlab^{TM} Term
$\bar{u}_t$	u
V_c	Vc
$\dot{\lambda}$	Osd
d	$Tlat$

Appendix D Matlab^{TM} Syntax Headers

In this section we introduce the headers of the Matlab^{TM} code used examine various filters.

Stand Alone Filters Figures D1 and D2 demonstrate the interconnection of Matlab^{TM} files used to examine the filters in a stand alone implementation.

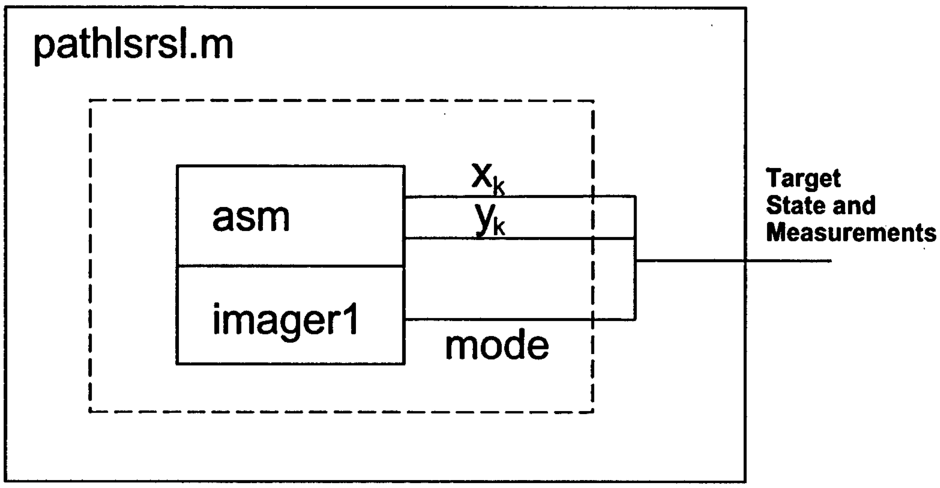

Figure D1: Matlab^{TM} files for data generation in stand alone filtering simulation

Path Generation

- pathlsrsl
- asm
- imager1

```
%Syntax:        : function [Xk,Ymk,modev,xho,vk,phi,t,hm,h]=pathlsrsl(dummy);
%Description    : Generates a target path in 2-d space(x-y)and encompasses 5 manoeuvres -
%                left turn(5sec),straight(6sec),right turn(5sec),straight(3sec),left turn(5sec)
%Inputs         : nil (all necessary variables are set by initcommon)
%Outputs        : Xk-path state vector (4x(T/h)) (T=total path time (sec), h=sample time (sec))
%                   Xk(1,:)=x position, Xk(2,:)= y position,Xk(3,:)=x velocity, Xk(4,:)=y velocity
%               : Ymk-radar measurements
%                   Ymk(1,:)=range(m), Ymk(2,:)= bearing(deg.)
%                   Ymk(3,:)=target direction of travel(bin number, of 30deg opening)
%               : modev-record of the target mode(straight path or turning left/right)at
```

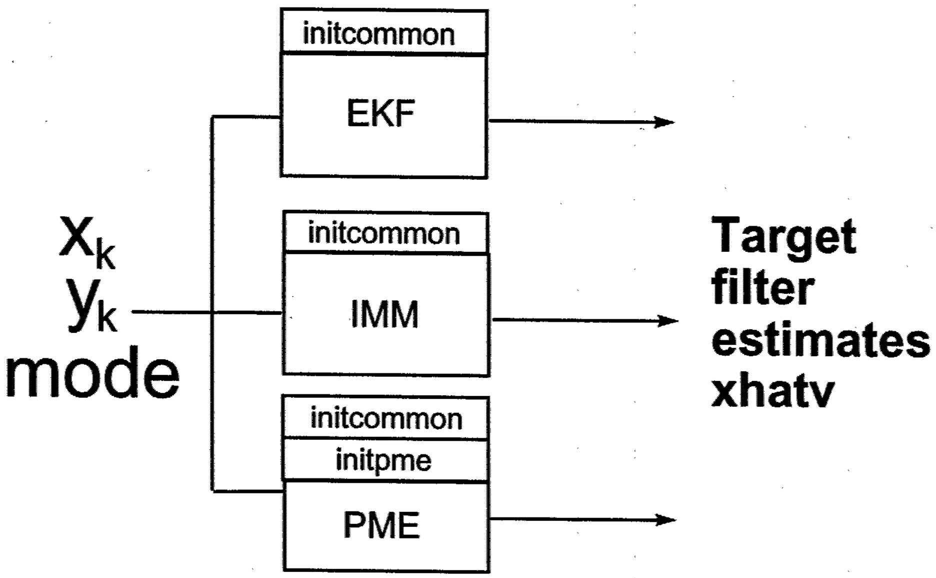

Figure D2: Matlab^{TM} files for filters in stand alone filtering simulation

```
%                                                              each sample point
%                : xho- estimate of the state vector at the beginning of the run
%                : vk- process and measurement noise variance vector
%                   vk(1)= range measurement noise variance,vk(2)=bearing measurement noise variance
%                   vk(3)=x component process noise variance,vk(4)= y component process noise variance
%                : phi= turn rate (rad/sec) - +ve =left turn,-ve = right turn of last manoeuvre.
%                : t= Total flight time
%                : hm=measurement interval (sec)
%                : h=sample interval (sec)
%Calls           : asm.m, initcommon.m
%Authors         : Jason Ford, Peter Hunter
%Modifications   : final version 14-11-01
%Bugs            : nil
%Assumptions     : Target has constant velocity segments and constant turning rate segments.

%Syntax:         : function [Xk,Ymk,xkm,nm,modev1]=asm(phi,xkm,vk,t,hm,h);
%Description     : Generates a target path in 2-d space(x-y)encompassing 1 manoeuvre -
%                  left turn (phi=+0.2 rad/sec),straight(phi=0 rad/sec) or right turn(phi=-0.2 rad/sec)
%Inputs          : phi= turn rate (rad/sec) - +ve =left turn,-ve = right turn
%                : xkm = initial state vector for this manoeuvre iteration.
%                : vk- process and measurement noise variance vector
%                   vk(1)= range measurement noise variance,vk(2)=bearing measurement noise variance
%                   vk(3)=x component process noise variance,vk(4)= y component process noise variance
%                : t= manoeuvre time
%                : hm=measurement interval (sec)
%                : h=sample interval (sec)
%Outputs         : Xk-path state vector (4x(T/h)) (T=total path time (sec), h=sample time (sec))
%                   Xk(1,:)=x position, Xk(2,:)= y position,Xk(3,:)=x velocity, Xk(4,:)=y velocity
```

```
%               : Ymk-radar measurements
%                 Ymk(1,:)=range(m), Ymk(2,:)= bearing(deg.)
%                 Ymk(3,:)=target direction of travel(bin number, of 30deg opening)
%               : xkm - (4x1) initial state vector for the manoeuvre
%               : nm - the number of measurements in the manoeuvre
%               : modev1-record of the target mode(straight path or turning left/right)at
%                                                               each sample point
%Calls          : imager1.m
%Authors        : Jason Ford, Peter Hunter
%Modifications  : final version 15-11-01
%Bugs           : nil
%Assumptions    : nil


%Syntax:        : [bin]=imager1(Ymk,Vy,Vx,Y,X,i);
%Description    : The target direction is located in one of twelve 30 deg.intervals(labelled
%                                               cw from North as {1,2,3,..,12})
%                 Randomly occurring allocation errors are introduced:
%                    -NNE - nearest neighbour error- placed into an adjacent bin due to noise etc.
%                    -UDE - uniformly distributed error - occlusion of target makes any bin number
%                                                                    a possibility
%Inputs         : Ymk-not used
%               : Vy - target velocity - y direction
%               : Vx - target velocity - x direction
%               : Y - target location - y direction
%               : X - target location - x direction
%               : i - not used
%Outputs        : bin - the direction of target travel as classified {1:12} (noisy result)
%Calls          : nil
%Authors        : Jason Ford, Peter Hunter
%Modifications  : final version 14-11-01
%Bugs           : nil
%Assumptions    : Target can be represented by 1 of twelve attitude bins.
%               : Assumptions on the nature of the error.
```

Filters

- pme
- imm
- ekf

```
%Syntax:        : [xhatp,Pxxv,normrecord,modeestv,binestv]=pme(Ymk,Xk);
%Description    : implements a Polymorphic Estimator algorithm to produce an estimate
%                 of position and velocity of the target derived from noisy range
%                 and bearing measurements from a simulated tracking radar.
%Inputs         : Ymk-radar measurements from the simulation origin.
%                   Ymk(1,:)=range(m), Ymk(2,:)= bearing(deg.)
%                   Ymk(3,:)=target direction of travel(bin number, of 30deg opening)
%               : Xk-target state (x,y,x-velocity,y-velocity) and only used for the initial Pxxv
%Output         : xhatp (4xT) vector of position estimate (x,y, x-velocity, y-velocity)
%               : Pxxv(4xT)-vector of Pxx diag elements (used by plotdata routine).
```

```
%               : normrecord- debuging info.
%               : modeestv(1xT)-vector of manu. mode estimates.
%               : binestv(1xT)-vector of orientation mode estimates.
%Calls          : initpme
%Authors        : Jason Ford, Peter Hunter
%Modifications  : final version 22-11-01
%Bugs           : The time update step of the algorithm is not particularly stable.
%                Small time step sizes (ie. h) seem to improve the preformance but
%                bad updates of Pxx vector are possible (if this happens the code resets
%                some quantities in the filter to try to stop the filter diverging).
%Assumptions    : numeric intergration of continuous-time equations.


%Syntax:        : [xhati,Pkkv,muv,xh1,xh2,xh3]=imm(Ymk);
%Description    : Interacting Multiple Mode filter (IMM)- is a filter function used to estimate
%                 the location of a manoeuvring target.It uses measurement data as well as manoeuvre
%                 type (straight or turning path) to estimate the path of the target.
%                 It uses the mode estimation to switch between individual EKF algorithms to better
%                                                                             track the target.
%Inputs         : Ymk-radar measurements from the simulation origin.
%                   Ymk(1,:)=range(m), Ymk(2,:)= bearing(deg.)
%                   Ymk(3,:)=target direction of travel(bin number, of 30deg opening)
%               :modev (1xT/h) vector which logs the actual manoeuvre at each sample point
%Outputs        :xhati (4xT) vector of estimates  (x,y, x-velocity, y-velocity)
%               :Pkkv combined state covariance matrix
%               :muv mode probability for each modeal filter
%               :xh1 state estimate of filter 1
%               :xh2 state estimate of filter 2
%               :xh3 state estimate of filter 3
%Calls          : initcommon
%Authors        : Jason Ford, Peter Hunter
%Modifications  : final version 20-11-01
%Bugs           : nil
%Assumptions    : Based on hybrid system mode of dynamics.

%Syntax:        : function [xhatv]=ekf(Ymk,modev,prf);
%Description    : implements an Extended Kalman Filter to produce an estimate of position and
%                 velocity of the target derived from noisy range and bearing measurements from
%                 a simulated tracking radar. The filter has the target manoeuvre at each
%                 sample point (straight path or turning left or right at 0.2 rad/sec)
%Inputs         : Ymk-radar measurements from simulation origin
%                   Ymk(1,:)=range(m), Ymk(2,:)= bearing(deg.)
%                   Ymk(3,:)=target direction of travel(bin number, of 30deg opening)
%               : modev (1xT/h) vector which logs the actual manoeuvre at each sample point
%               : prf - measurement interval (sec) (pulse repetition frequency of the radar system)
%Output         : xhatv (4xT) vector of position estimate (x,y, x velocity, y velocity)
%Calls          : initcommon
%Authors        : Jason Ford, Peter Hunter
%Modifications  : final version 20-11-01
%Bugs           : nil
%Assumptions    : nil
```

Initialisation files

- initcommon

- initpme

```
%Syntax:        :initcommon
%Description    : contains the initialization values for all filters
%Inputs         : nil
%Outputs        : there are no explicit outputs but the following variables are initialised:
%                   - h (sample interval(s)),phi(turn rate(rad/sec)),hm
%                   - xs,xhs
%                   - Ac,Ad,Rn,Qn,Pn
%                   - for EKF  : EKFQ,EKFR,EKFP
%                   - for IMM  : IMMQ,IMMR,IMMP
%                   - for PME  : Rx,Rxx,Pxx
%Calls          : nil
%Authors        : Jason Ford, Peter Hunter
%Modifications : final version 14-11-01
%Bugs           : nil
%Assumptions    : nil


%Syntax:        : initpme
%Description    : Contains constants and initialization data for the pme.m program
%Inputs         : nil
%Outputs        : there are no explicit outputs but the following variables are initialised:
%                   - statenum,modenum,binnum,measbin,bigmode
%                   - hm,minfh
%                   - Aa,As,Al,Ar,Ai
%                   - Po,Q,Rx,Rxx.
%Calls          : initcommon
%Authors        : Jason Ford, Peter Hunter
%Modifications : final version 27-11-01
%Bugs           : nil
%Assumptions    : nil
```

Guidance Evaluation Figure D3 describes the structure of the Matlab^{TM} code used to simulate a missile guidance loop using an extended Kalman filter. The Matlab^{TM} code for incorporating the PME filter is similar.

Matlab Files

- comparefilters
- guidanceekf
- guidamcepme
- ekfscon
- pmelloop

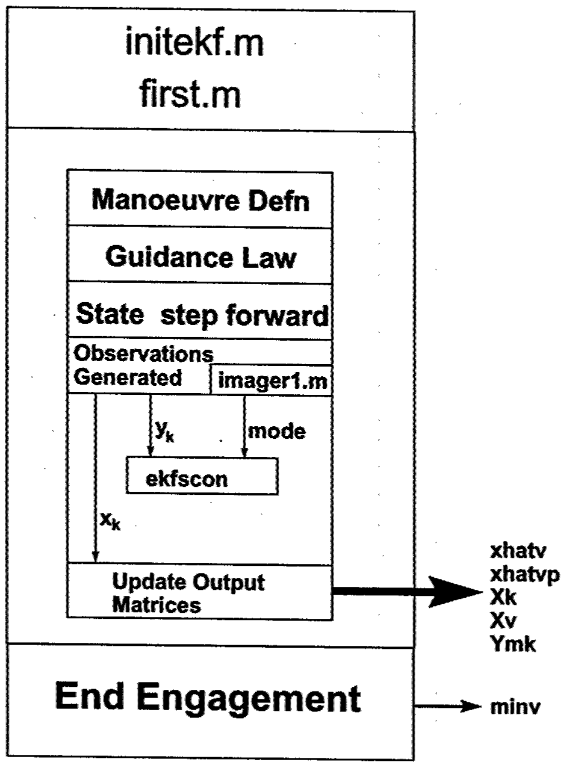

Figure D3: Matlab^{TM} files for guidance loop simulation using the extended Kalman filter.

- imager1
- first
- initekf
- initpme

```
%comparefilters
%Syntax:         : comparefilters
%Description     : runs a number of control/filter algorithms to compare the minimum miss distances (minv)
%Inputs          : nil
%Outputs         : MISSmin - vector of the minimum miss distances for each filter
%                : MeanMISSminv - for multiple runs this is the average result for each filter
%                : additional outputs with prefixes :EKF...(c),PMEMOD...,PMENM...,PMESM1500...,
%                                                   PMESM750...
%                  - ekf filter :Pm,Xk,xhatv,Xv,Clatv,Tlatv
%                  - pme filter :Pxxv,Xk,xhatv,pxxcount,pxxvcount,badextraps,Xv,Clatv,Tlatv
%
%Calls           : guidanceekf, guidancepme, first
```

```
%Authors       : Jason Ford, Peter Hunter
%Modifications : 16/2/01 : save workspace to comparef4a.mat
%                21/2/01 : save workspace to comparef4b.mat
%                28/2/01 : save workspace to comparef4c.mat  h=0.0001 s=3
%                12/7/01 : save workspace to comparef4f.mat  h=0.001, ekf, ekfc, pmemod
%                18/7/01 : save workspace to comparef4g.mat  h=0.001 s=10 all except ekfc
%                2/10/01 : save workspace to comparef4a2.mat h=0.0001, s=1, run=10, only EKF &
%                                                                                    PMEMOD
%                3/10/01 : save workspace to comparef4a3.mat as for comparef4a2 except all filters
%                27/11/01: final version , renamed comparefilters
%Bugs           : nil
%Assumptions    : nil

%Syntax:        : [Pmt,Xv,Xk,Ski,Ymk,modev,xhatv,xhatvp,Rangev,Tlatv,minv,Clatv]=guidanceekf(timesteps,ti0);
%Description    : A simulated engagement between an incoming, manoeuvring target and a radar guided
%                 interceptor. The radar range and bearing measurements are filtered by an EKF, with
%                 the whole operating within an APN guidance loop.
%Globals        : B, Q, R, Bt;
%Inputs         : timesteps - the number of time sample points = realtime/h.
%               : ti0 - interceptor start heading (rad)
%Outputs        : Pmt - diagonal elements of the covariance matrix. It is a diagnostic output to
%                        enable construction of the 1 sigma circles at each measurement point
%                        of the filter output.
%               : Xv - relative state between interceptor and target
%               : Xk - target state
%               : Ski - actual target state
%               : Ymk - measurement sequence (radar range and bearing measurements)
%               : modev - the target mode vector
%               : xhatv - estimate of target state
%               : xhatvp - one-step ahead estimate of target state
%               : Rangev - target-interceptor separation(used for testing guidance loop)
%               : Tlatv - perpendicular component of the target acceleration(testing variable)
%               : minv - minimum miss distance between the target and interceptor
%               : Clatv - Vector of guidance commands
%Calls          : initekf
%               : ekfscon.m
%               : imager1.m
%Authors        : Jason Ford, Peter Hunter
%Modifications  : final version 27-11-01
%Bugs           : nil
%Assumptions    : interceptor position is prefectly known...  in real case would need INS solution


%Syntax:        : [Xv,Xk,Pxxv,Ski,Ymk,modev,xhatv,xhatvp,Rangev,Tlatv,minv,pxxcount,pxxmcount,
%                 badextraps,Clatv]=guidancepme(timesteps,ti0,moderangemax);
%Description    : A simulated engagement between an incoming, manoeuvring target and a radar
%                 guided interceptor.  The radar range and bearing measurements are filtered
%                 by a PME, with the whole operating within an APN guidance loop.
%Globals        : B, Q, R, Bt
%               : TIMEUPDATE MODEUPDATE BASEUPDATE RADAR_ALL_TIME
%               : xh fh Pxx Pxf Pxxfi xhm fhm Pxxm Pxfm Pxxfim modest
%               : A Rx Rxx W Q Pb
%               : pxxcount badextraps pxxmcount
%               : bigmode E
%               : h P_preset
%Inputs         : timesteps - the number of time sample points = realtime/h.
%               : ti0 - interceptor start heading (rad)
%               : moderangemax - the maximum distance (m) over which the imager provides information
```

```
%Outputs       : Xv - relative state between interceptor and target
%              : Xk - target state
%              : Pxxv- diagonal elements of the covariance matrix. Used to construction
%                      one sigma circles at each measurement point of the filter output.
%              : Ski - actual target state
%              : Ymk - measurement sequence (radar range and bearing measurements)
%              : modev - the target mode vector
%              : xhatv - estimate of target position
%              : xhatvp - Debug variable
%              : Rangev - target-interceptor separation(used for testing guidance loop)
%              : Tlatv - perpendicular component of the target acceleration(testing variable)
%              : minv - minimum miss distance between the target and interceptor
%              : pxxcount  - Error count
%              : pxxmcount - Error count
%              : badextraps  - Error count
%              : Clatv - Vector of guidance commands
%Calls         : initpme
%              : pme1loop.m
%              : imager1.m
%Authors       : Jason Ford, Peter Hunter
%Modifications : final version 27-11-01
%Bugs          : nil
%Assumptions   : interceptor position is prefectly known...  in real case would need INS solution

%Syntax:       :[xhp,xhm,Pm]=ekfscon(Ahm,xhm,Pm,ymk,radarpoints,i,u,xik);
%Description   : EKF section - one iteration only
%Globals       : B, Q, R, Bt;
%Inputs        : Ahm - initial state transition matrix
%              : xhm - initial target state (k-1|k-1)
%              : Pm - initial covariance martix (k-1|k-1)
%              : Ymk-radar measurements from the simulation origin.
%                  Ymk(1,:)=range(m), Ymk(2,:)= bearing(deg.)
%                  Ymk(3,:)=target direction of travel(bin number, of 30deg opening)
%              : radarpoints - helps determine if a measurement update is required
%              : i - time k
%              : u - control action of the interceptor.
%              : xik - initial interceptor state (k-1|k-1)
%Outputs       : xhp - state vector - after time update (k-1|k)
%              : xhm - measuremnet updated state (k|k)
%              : Pm - initial covariance matrix for (k+1)
%Calls         : nil
%Authors       : Jason Ford, Peter Hunter
%Modifications : final version 27-11-01
%Bugs          : nil
%Assumptions   : nil


%Syntax:       :[dummy]=pme1loop(Ymk,k,xik,radarpoints,modepoints,moderangemax,Range);
%Description   : is a single time iteration of the PME filtering algorithm
%Globals       : TIMEUPDATE MODEUPDATE BASEUPDATE RADAR_ALL_TIME
%              : xh fh Pxx Pxf Pxxfi xhm fhm Pxxm Pxfm Pxxfim modest
%              : A Rx Rxx W Q Pb
%              : pxxcount badextraps pxxmcount
%              : bigmode E
%              : h P_preset
%Inputs        : Ymk-radar measurements from the simulation origin.
%                  Ymk(1,:)=range(m), Ymk(2,:)= bearing(deg.)
%                  Ymk(3,:)=target direction of travel(bin number, of 30deg opening)
```

```
%               : radarpoints - number of radar measurements
%               : modepoints - number of mode observations
%               : moderangemax - maximum range (m) over which the image information is available
%               : Range
%Outputs        : nil explicitly (output via global variables)
%Calls          : nil
%Authors        : Jason Ford, Peter Hunter
%Modifications  : final version 27-11-01
%Bugs           : Same bugs as PME filter. Numerical unstable.
%Assumptions    : Same as the PME filter.

%Syntax:        : [bin]=imager1(Ymk,Vy,Vx,Y,X,i);
%Description    : The target direction is located in one of twelve 30 deg.intervals(labelled cw
%                                                  from North as {1,2,3,..,12})
%                 Randomly occurring allocation errors are introduced:
%                    -NNE - nearest neighbour error- placed into an adjacent bin due to noise etc.
%                    -UDE - uniformly distributed error - occlusion of target makes any bin number
%                        a possibility
%Globals        : nil
%Inputs         : Ymk-not used
%               : Vy - target velocity - y direction
%               : Vx - target velocity - x direction
%               : Y - target location - y direction
%               : X - target location - x direction
%               : i - not used
%Outputs        : bin - the direction of target travel as classified {1:12} (noisy result)
%Calls          : nil
%Authors        : Jason Ford, Peter Hunter
%Modifications  : final version 27-11-01
%Bugs           : nil
%Assumptions    : nil


%Syntax:        :first
%Description    : contains the initialization values for all filters evaluated in the control loop
%                    environment
%Globals        : nil
%Inputs         : nil
%Outputs        : there are no explicit outputs but the following variables are initialised:
%                   - Basic System Parameters :h ,phi,hm,modeh,radarh
%                   - Sample Numbers/sec      :modepoints,radarpoints,controlpoints
%                   - Initial State vectors   :ta,xs,ti0,xis
%                   - Noise variances         :pxvar,pyvar,rvar,tvar
%                   - Debugging Switches      :TIMEUPDATE,MODEUPDATE,BASEUPDATE,RADAR ALL TIME
%                   - Iteration Parameters    :s,runnum,seedoff
%Calls          : nil
%Authors        : Jason Ford, Peter Hunter
%Modifications  : final version 27-11-01
%Bugs           : nil
%Assumptions    : nil


%Syntax:        :initekf
%Description    : contains the initialization values for EKF filter
%Globals        : B, Q, R
%Inputs         : nil
%Outputs        : there are no explicit outputs but the following variables are initialised:
%                   -phi
```

```
%                 -xhs
%                 -Ac,Ad,Aint,Aintd,Bt,Bi
%                 -Pn,Qn,Rn
%                 -EKFP,EKFQ,EKFR
%Calls          : nil
%Authors        : Jason Ford, Peter Hunter
%Modifications  : final version 27-11-01
%Bugs           : nil
%Assumptions    : nil


%Syntax:        : initpme
%Description    : contains the initialization values for the guidancepme program
%Globals        : nil
%Inputs         : nil
%Outputs        : there are no explicit outputs but the following variables are initialised:
%                 -statenum,modenum,binnum,measbin,bigmode
%                 -pxxcount
%                 -f,fh,fhm,minfh
%                 -w
%                 -xhm,xhs,xh,xkm,xo,xho,ym,ybm
%                 -Aa,As,Al,Ar,A,Ah,Ahm,Ac,Ad,Ai,Aint,Aintd
%                 -Bt,Bi,C,Ck,E,G
%                 -Pn,Po,Pb1,Pb,Pxf,Pxffi,Pxx,Q,Qn,Rn,Rx,Rxx
%                 -MKL,Pxx_p,pixx_p,Pxx_m,pixx_m,phi_m,phi_p,Pxp_m,
%                 -Pxp_m,Pxp_p,x_p,x_m,badextraps
%Calls          : first
%Authors        : Jason Ford, Peter Hunter
%Modifications  : final version 27-11-01
%Bugs           : nil
%Assumptions    : nil
```

Appendix E Simulation Parameter Values

E.1 Stand-Alone Filter Evaluation

The following parameter values were used in the stand-alone filter evaluation:

$h = 0.001s$	sample time
$hm = 0.01s$	measurement time
$xs = [36000, 36000, -300, -300]$	initial target state
$xsh = [36100, 36100, -310, -310]$	initial target state estimate
$phi = 0.2rad/s$	turn rate

$Aa =$	mode transition probability matrix (3×3).
If in straight flight mode:	
0.98	straight path will continue
0.01	change to left (or right) turn mode
If in a turn flight mode:	
0.99	that the turn will continue
0.01	change to straight flight mode.
Al, As and Ar=	Probability matrix for manoeuvre induced bin number change, each (12×12).
As	straight path gives no change (Identity matrix).
Al	left turn - probability of transition to next left bin is 0.05.
Ar	right turn - probability of transition to next right bin is 0.05
fh	bin transition indicator (36x1). Initialised so that $fh(8) = 1$, $fh(i) = 0$ for all other i

E.2 Filters in Guidance Loop

$h = 0.0001$

$hm = h$

$radarh = 1$

$controlh = 0.01$

$phi = 0.2rad/s$	
$rvar = (40m)^2$	target range measurement variance
$tvar = (1.75 * 10^{-3}rad/s)^2$	target bearing measurement variance
$timesteps = 100,000$	
$xs = [2000, 1000, 660 \times cos(225), 660 \times sin(225)]$	
$[100, 100, 10, 10]$	initial state estimate errors
$ta = 225°$	initial target heading
$xis = [0, 0, 1000 \times cos(75), 1000 \times sin(75)]$	
$ti0 = 75°$	interceptor initial bearing

DISTRIBUTION LIST

Estimation of Manoeuvring Targets using Hybrid Filters

Jason J. Ford and Peter G. Hunter

AUSTRALIA

DEFENCE ORGANISATION

S&T Program

Chief Defence Scientist
FAS Science Policy
AS Science Corporate Management
Director General Science Policy Development
} shared copy

Counsellor Defence Science, London (Doc Data Sheet)
Counsellor Defence Science, Washington (Doc Data Sheet)
Scientific Adviser to MRDC Thailand (Doc Data Sheet)
Scientific Adviser Joint
Navy Scientific Adviser (Doc Data Sheet and Distribution list only)
Scientific Adviser - Army (Doc Data Sheet and Distribution list only)
Air Force Scientific Adviser
Director Trials

Systems Sciences Laboratory

Chief of Weapons Systems Division (Doc Data Sheet and Distribution List Only)
Research Leader: RLAWS (Doc Data Sheet and Distribution List Only)
Task Manager: Farhan Faruqi
Author(s): Jason J. Ford, Peter G. Hunter

DSTO Library and Archives

Library Edinburgh 2 copy
Australian Archives

Capability Systems Staff

Director General Maritime Development (Doc Data Sheet only)
Director General Land Development
Director General Aerospace Development (Doc Data Sheet only)

Knowledge Staff

Director General Command, Control, Communications and Computers (DGC4) (Doc Data Sheet only)

Army

ABCA National Standardisation Officer, Land Warfare Development Sector, Puckapunyal (4 copies)

SO (Science), Deployable Joint Force Headquarters (DJFHQ) (L), Enoggera QLD
(Doc Data Sheet only)

Intelligence Program
DGSTA Defence Intelligence Organisation
Manager, Information Centre, Defence Intelligence Organisation

Defence Libraries
Library Manager, DLS-Canberra
Library Manager, DLS - Sydney West (Doc Data Sheet Only)

UNIVERSITIES AND COLLEGES
Australian Defence Force Academy
Library
Head of Aerospace and Mechanical Engineering
Serials Section (M list), Deakin University Library, Geelong, VIC
Hargrave Library, Monash University (Doc Data Sheet only)
Librarian, Flinders University

OTHER ORGANISATIONS
National Library of Australia
NASA (Canberra)

OUTSIDE AUSTRALIA

INTERNATIONAL DEFENCE INFORMATION CENTRES
US Defense Technical Information Center, 2 copies
UK Defence Research Information Centre, 2 copies
Canada Defence Scientific Information Service, 1 copy
NZ Defence Information Centre, 1 copy

ABSTRACTING AND INFORMATION ORGANISATIONS
Library, Chemical Abstracts Reference Service
Engineering Societies Library, US
Materials Information, Cambridge Scientific Abstracts, US
Documents Librarian, The Center for Research Libraries, US

INFORMATION EXCHANGE AGREEMENT PARTNERS
Acquisitions Unit, Science Reference and Information Service, UK
Library - Exchange Desk, National Institute of Standards and Technology, US

SPARES (5 copies)

Total number of copies: 35

Page classification: UNCLASSIFIED

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION **DOCUMENT CONTROL DATA**	1. CAVEAT/PRIVACY MARKING

2. TITLE	3. SECURITY CLASSIFICATION
Estimation of Manoeuvring Targets using Hybrid Filters	Document (U) Title (U) Abstract (U)
4. AUTHORS	5. CORPORATE AUTHOR
Jason J. Ford & Peter G. Hunter	Systems Sciences Laboratory PO Box 1500 Edinburgh, South Australia, Australia 5111

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO–TN–0488	AR-012-723	Technical Note	April, 2003

8. FILE NUMBER	9. TASK NUMBER	10. SPONSOR	11. No OF PAGES	12. No OF REFS
9505-23-110	LRR 01/219	DSTO	37	18

13. URL OF ELECTRONIC VERSION	14. RELEASE AUTHORITY
http://www.dsto.defence.gov.au/corporate/reports/DSTO–TN–0488.pdf	Chief, Weapons Systems Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved For Public Release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SOUTH AUSTRALIA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS

No Limitations

18. DEFTEST DESCRIPTORS

Target acquisition Filters (Mathematics)	Nonlinear systems

19. ABSTRACT

This report examines the problem of estimating the location, velocity and manoeuvre of a manoeuvring target (from range, bearing and pose information). The report considers a non-linear modeling technique in which the target is represented as a hybrid system (a combination of discrete and continuous valued states) and considers new associated approaches such as the polymorphic estimator.

Although simulation studies were performed, the polymorphic estimator had serious numeric problems that suggested the estimator should not be used until the approach is refined. This report is intended to facilitate further discussion and development of the approach (if deemed necessary); hence, it includes details of the assumptions made and the Matlab^{TM} implementation.

Page classification: UNCLASSIFIED