3-2006

Multiple Model Methods for Cost Function Based Multiple Hypothesis Trackers

Matthew C. Kozak

Multiple Model Methods for Cost Function Based Multiple Hypothesis Trackers

THESIS

Matthew C. Kozak, Captain, USAF

AFIT/GE/ENG/06-29

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/06-29

Multiple Model Methods for Cost Function Based Multiple Hypothesis Trackers

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Matthew C. Kozak, B.S.E.E.

Captain, USAF

March 2006

AFIT/GE/ENG/06-29

Multiple Model Methods for Cost Function Based Multiple Hypothesis Trackers

Matthew C. Kozak, B.S.E.E.

Captain, USAF

Approved:

/signed/	3 Mar 2006
Dr Peter S. Maybeck (Chairman)	date
/signed/	3 Mar 2006
Lt Col Juan Vasquez (Member)	date
/signed/	3 Mar 2006
Dr Meir Pachter (Member)	date

AFIT/GE/ENG/06-29

Abstract

To estimate the state of a maneuvering target in clutter, a tracking algorithm must be capable of addressing measurement noise, varying target dynamics, and clutter. Traditionally, Kalman filters have been used to reject measurement noise, and their multiple model form can accurately identify target dynamics. The Multiple Hypothesis Tracker (MHT), a Bayesian solution to the measurement association problem that retains the probability density function of the target state as a mixture of weighted Gaussians, offers the greatest potential for rejecting clutter, especially when based on an advanced mixture reduction algorithm (MRA) such as the Integral Square Error (ISE) cost function.

This research seeks to incorporate multiple model filters into an ISE cost-function-based MHT to increase the fidelity of target state estimation. Two architectures are tested: replacing the components of an MHT's Gaussian mixture with Multiple Model Adaptive Estimators (MMAEs) or Interacting Multiple Model (IMM) estimators, and replacing the elemental filters of an MMAE or IMM with ISE-based MHTs. Results indicate that all of the proposed multiple model methods can properly identify the maneuver mode of a target, even in dense clutter, and ensure that an appropriately tuned filter is used. During benign portions of flight, this causes significant reductions in position and velocity RMS errors compared to a single-filter MHT. During portions of flight when the mixture mean deviates significantly from true target position, so-called deferred decision periods, the multiple model structures tend to accumulate greater RMS errors than a single-filter MHT, but this effect is inconsequential considering the inherently large magnitude of these errors (a non-MHT tracker would not be able to track during these periods at all). The multiple model MHT structures do not negatively impact track life when compared to a single-filter MHT.

Acknowledgements

I would like to extend my deepest gratitude to Dr. Peter Maybeck, whose patience, encouragement, and expert guidance truly made this thesis possible. I would also like to thank the members of of my committee, Lt Col Juan Vasquez and Dr. Meir Pachter, not only for feedback on my research but also for their countless hours of instruction both in and out of the classroom.

Most importantly, I owe thanks to my wife for her love, support, and understanding throughout the the past few years – I look forward to making up lost time with her.

Matthew C. Kozak

Table of Contents

Page

Abstract . . . iv

Acknowledgements . . . v

List of Figures . . . ix

Notation . . . xiii

I. Introduction . . . 1-1
- 1.1 Motivation . . . 1-1
- 1.2 Research Goal . . . 1-4
- 1.3 Organization . . . 1-4

II. Background . . . 2-1
- 2.1 Introduction . . . 2-1
- 2.2 Tracking Fundamentals . . . 2-1
- 2.3 Kalman Filtering . . . 2-2
  - 2.3.1 Dynamics Design Models . . . 2-3
  - 2.3.2 Propagation . . . 2-4
  - 2.3.3 Measurement Incorporation . . . 2-5
- 2.4 Multiple Model Estimation . . . 2-6
  - 2.4.1 Non-Switching Models . . . 2-6
  - 2.4.2 Ad Hoc MMAE Techniques . . . 2-8
  - 2.4.3 Switching Models . . . 2-11
- 2.5 Multiple Hypothesis Tracking . . . 2-14
  - 2.5.1 Gaussian Mixtures . . . 2-14
  - 2.5.2 Measurement Gating . . . 2-17
  - 2.5.3 State Updates with Measurement Uncertainty . . . 2-17
  - 2.5.4 Mixture Reduction Algorithms . . . 2-19
- 2.6 Cost Functions . . . 2-21
  - 2.6.1 Kolmogorov Variational Distance . . . 2-21
  - 2.6.2 Bhattacharyya Distance . . . 2-23
  - 2.6.3 Integral Square Error . . . 2-25
  - 2.6.4 Kullback-Leibler Distance . . . 2-28
  - 2.6.5 Merging and Pruning . . . 2-28
  - 2.6.6 ISE Complexity . . . 2-30
- 2.7 Multiple Model MHT . . . 2-31
- 2.8 Summary . . . 2-33

Page

III. Simulation Development and Analysis 3-1
- 3.1 Introduction 3-1
- 3.2 Program Architecture 3-1
- 3.3 Truth Generation 3-2
  - 3.3.1 Random Noise Process Simulations 3-2
  - 3.3.2 Loop Models 3-3
  - 3.3.3 FlightGear 3-5
- 3.4 Measurement Generation 3-7
- 3.5 Monte Carlo Simulation 3-11
- 3.6 Filter Modules 3-11
  - 3.6.1 Data Containers 3-12
  - 3.6.2 Simple Kalman 3-13
  - 3.6.3 Multiple Model 3-16
  - 3.6.4 Multiple Hypothesis Module 3-18
- 3.7 Filter Configurations 3-20
  - 3.7.1 Multiple Model of Simple Kalman 3-20
  - 3.7.2 MHT of Simple Kalman 3-21
  - 3.7.3 MacroHypothesis (MHT of Multiple Model) 3-21
  - 3.7.4 Macromixture (Multiple Model of MHT) 3-25
- 3.8 Track Loss Checks 3-27
- 3.9 Probability Marshalling 3-29
- 3.10 Error Estimation and Performance Analysis 3-31
- 3.11 Summary 3-32

IV. Simulation Results 4-1
- 4.1 Introduction 4-1
- 4.2 ISE Implementation 4-1
- 4.3 Loop Models 4-4
  - 4.3.1 Clutter Free 4-5
  - 4.3.2 Clutter 4-12
  - 4.3.3 Loop Model Summary 4-29
- 4.4 Flight Data 4-30
  - 4.4.1 Clutter Free 4-30
  - 4.4.2 Clutter 4-33
  - 4.4.3 F-16 Summary 4-42
- 4.5 Probability Marshalling 4-42
  - 4.5.1 Decoupled Multiple Model Axes 4-43
  - 4.5.2 Broken-Loop Model Probability Marshalling 4-43
  - 4.5.3 F-16 Probability Marshalling 4-47
- 4.6 Summary 4-52

Page

V. Conclusions and Recommendations 5-1
5.1 Restatement of Research Goal 5-1
5.2 Summary of Results 5-1
5.3 Significant Contributions of Research 5-4
5.4 Recommendations for Future Research 5-6

Appendix A. Figure Explanations A-1

Bibliography 1

List of Figures

Figure Page

1.1. MMAE signal flow block diagram. 1-2

1.2. Gaussian mixture reduction 1-3

2.1. Canonical Tracker 2-1

2.2. Non-Switching Multiple Model Filter 2-7

2.3. Multiple Filter Probability Ratios 2-9

2.4. Interacting Multiple Model Filter 2-12

2.5. Deferred Decision Making 2-15

2.6. Effect of Mean Spreading Terms in a Gaussian Mixture 2-16

2.7. Gating Regions for Uncertain Measurements and a Single Target . . 2-18

2.8. Various Cost Functions 2-22

2.9. Preference of Bhattacharyya to Maintain High Probability. 2-24

2.10. Kolmogorov Distance for the Probability Change in an Identical Partition of Two Functions 2-26

2.11. Bhattacharyya Distance for the Probability Change in an Identical Partition of Two Functions 2-26

2.12. Squared Error Distance for the Probability Change in an Identical Partition of Two Functions 2-27

2.13. Average Track Life of Various Merging and Pruning Algorithms . . 2-30

3.1. Constant Velocity Truth Model Driven by White Gaussian Noise . . 3-3

3.2. Constant Velocity Truth Model with Spiral Input Dynamics 3-4

3.3. Broken Loop Injected Dynamics 3-5

3.4. F-16 Truth Flight Segment 3-7

3.5. F-16 Truth Flight Segment Projected onto the Image Plane of a Tracker 3-8

3.6. Probability Starvation From Insufficient Clutter Region 3-9

Figure Page

3.7. Standard Deviation vs Number of Monte Carlo Runs For Cluttered Scenario 3-12

3.8. Kalman Schema and Data Container 3-13

3.9. F-16 Projected Acceleration 3-15

3.10. Multiple Model Schema and Data Container 3-20

3.11. MHT Schema and Data Container 3-21

3.12. Macrohypothesis Schema and Data Container 3-22

3.13. Restart Mechanisms in a Multiple Model Filter 3-24

3.14. Macromixture Schema and Data Container 3-25

3.15. Track Loss in Low Clutter. 3-28

3.16. Dimple Gaussian Multiple Model Probability Calculation 3-31

4.1. ISE Average Track Life 4-2

4.2. Average Track Life Using an Estimate-Based Clutter Region 4-3

4.3. Average Calculation Time Per Epoch 4-3

4.4. Benign Filter: Loop Model Without Clutter 4-6

4.5. Aggressive Filter: Loop Model Without Clutter 4-7

4.6. MMAE Filter: Loop Model Without Clutter 4-8

4.7. Three Filter MMAE: Loop Model Without Clutter 4-10

4.8. IMM Filter: Loop Model Without Clutter 4-11

4.9. MMAE Filter: Broken-Loop Model Without Clutter 4-13

4.10. MMAE Filter: Broken-Loop Model Without Clutter Elemental Filter Probability Flow (Mean $\pm$ 1 Standard Deviation) 4-14

4.11. Benign Filter: Loop Model in Clutter 4-14

4.12. Aggressive Filter: Loop Model in Clutter 4-16

4.13. Macrohypothesis MMAE Filter: Loop Model in Clutter 4-17

4.14. Effects of Clutter on Multiple Model Probability Flow Calculations 4-18

4.15. Three-Filter Macrohypothesis MMAE: Loop Model in Clutter 4-20

Figure Page

4.16. MMAE Filter Probability Flow Visualized Through Propagated Covariance . . . 4-21

4.17. Macrohypothesis IMM Filter: Loop Model in Clutter . . . 4-23

4.18. Macromixture MMAE Filter: Loop Model in Clutter . . . 4-24

4.19. Macromixture IMM Filter: Loop Model in Clutter . . . 4-25

4.20. Macrohypothesis MMAE Filter: Broken-Loop Model in Clutter . . . 4-26

4.21. Macrohypothesis IMM Filter: Broken-Loop Model in Clutter . . . 4-27

4.22. Macromixture MMAE Filter: Broken-Loop Model in Clutter . . . 4-28

4.23. Macromixture IMM Filter: Broken-Loop Model in Clutter . . . 4-29

4.24. Benign CA Filter: F-16 Without Clutter . . . 4-31

4.25. Aggressive CA Filter: F-16 Without Clutter . . . 4-32

4.26. MMAE CA Filter: F-16 Without Clutter . . . 4-33

4.27. MMAE CA Filter: F-16 Without Clutter . . . 4-34

4.28. IMM CA Filter: F-16 Without Clutter . . . 4-34

4.29. Benign CA Filter: F-16 in Clutter . . . 4-35

4.30. Aggressive CA Filter: F-16 in Clutter . . . 4-36

4.31. Macrohypothesis MMAE CA Filter: F-16 in Clutter . . . 4-37

4.32. Macrohypothesis MMAE CA Filter: F-16 in Clutter Probability Flow 4-38

4.33. The Effect of Clutter on Mean Elemental Filter Probability Flow . 4-39

4.34. Macrohypothesis-based IMM CA Filter: F-16 in Clutter . . . 4-39

4.35. Macromixture-based MMAE CA Filter: F-16 in Clutter . . . 4-40

4.36. Macromixture-based IMM CA Filter: F-16 in Clutter . . . 4-41

4.37. F-16 Filter Performance in Clutter . . . 4-42

4.38. F-16 CA MMAE Filter: Separate Axes Without Clutter . . . 4-44

4.39. F-16 CA MMAE Filter: Separate Axes Without Clutter Tracker View 4-44

4.40. Broken-Loop Macrohypothesis MMAE Without Clutter: Probability Flow with and Without Ad Hoc Marshalling Techniques . . . 4-46

4.41. CV Broken-Loop Without Clutter: Temporally Averaged RMS Values 4-47

Figure Page

4.42. Broken-Loop Macrohypothesis MMAE with Clutter: Probability Flow with and Without Ad Hoc Marshalling Techniques 4-48

4.43. CV Broken-Loop in Clutter: Temporally Averaged RMS Values . . 4-49

4.44. F-16 Macrohypothesis MMAE Without Clutter: Probability Flow with and Without Ad Hoc Marshalling Techniques 4-50

4.45. F-16 Flight Without Clutter: Temporally Averaged RMS Errors . . 4-50

4.46. F-16 Macrohypothesis MMAE with Clutter: Probability Flow with and Without Ad Hoc Marshalling Techniques 4-51

4.47. F-16 Flight in Clutter: Temporally Averaged RMS Values (Complete and Benign Regime) 4-52

5.1. 30-Component ISE-Based MRA Running in Heavy Clutter 5-5

5.2. Modified ISE Cost Curve for $\varpi > -2$ 5-7

5.3. Modified ISE Cost Curve for $\varpi < -2$ 5-8

A.1. Example of Target Path, Best Estimate Path, and Estimate Error Ellipses A-1

A.2. Example of Elemental Filter Probability Gauges and Association Gates A-2

A.3. Example of Multi-Epoch View A-3

A.4. Example of Velocity and Acceleration Display A-3

Notation	Usage
$\boldsymbol{z}$, $\boldsymbol{Z}$, etc.	vectors are shown in boldface italic text
$\mathbf{P}$, $\mathbf{H}$, etc.	matrices are shown in boldface roman text
$\hat{\boldsymbol{x}}$, etc.	estimates are indicated using the 'hat' augmentation
$\boldsymbol{x}(t)$	a continuous-time signal, where the indexing t is a continuous variable representing the time in seconds
$\boldsymbol{x}(k)$	a discrete-time signal, where the indexing k is the sample number, and the k-th sample is taken at time t_k
$\hat{\boldsymbol{x}}(k\|k-1)$	the estimate of the signal at sample k, using information only up to the $(k-1)$-th measurement
$\hat{\boldsymbol{x}}(k\|k)$	the estimate of the signal at sample k, using information up to the current time k; the post-update state estimate
$\hat{\boldsymbol{z}}(k\|k-1)$	the predicted value of the measurement at sample k, using information only up to the $(k-1)$-th measurement
$\boldsymbol{Z}^k$	the entire measurement history from sample 1 to sample k
$\boldsymbol{Z}_k$	*all* measurements provided to the system in the k-th set of measurements (i.e., the k-th scan)
$\boldsymbol{z}_j(k)$	the j-th measurement from the k-th set of measurement (i.e., the k-th scan)
$P\{\cdot\}$	the probability of the discrete event specified in $\{\cdot\}$
$f\{\cdot\}$	the probability density function of the continuous parameter specified in the argument $\{\cdot\}$
$\mathcal{N}\{\boldsymbol{x};\boldsymbol{\mu},\mathbf{P}\}$	denotes a Gaussian probability density function for variable $\boldsymbol{x}$, distributed with mean $\boldsymbol{\mu}$ and covariance $\mathbf{P}$: $\mathcal{N}\{\boldsymbol{x};\boldsymbol{\mu},\mathbf{P}\} = \|2\pi\mathbf{P}\|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\mathbf{P}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}$
N_f	the number of filters (or models) in the system
M_j	the event in which model j is in force in a non-switching multiple model system (no time argument is supplied, as this is the non-switching case, in which the model in force does not change with time)
$M_{k,j}$	the event in which model j is in force at sample k in a switching multiple model system
$M^{k,l}$	the l-th model history event in a switching multiple model system — consists of a single time step event (e.g., $M_{k,j}$) for each sample time from 1 to k
$\mathbf{T}(k\|k-1)$	the Markov probability transition matrix for an Interacting Multiple Model (IMM) system
$\hat{\boldsymbol{x}}_j$, $\mathbf{P}_j$	the state estimate and covariance of the j-th filter in a multiple model system, or of the j-th target
$\hat{\boldsymbol{x}}^j$, $\mathbf{P}^j$	the modified state of the j-th model after mixing; provided as the input to the j-th filter in the IMM algorithm
s	the dimension of each measurement from a sensor (always 2 in this research)
$N_z(k)$	the number of measurements in the k-th set (i.e., the k-th scan)

Notation	Usage
$\psi_i(k)$	the i-th association event, proposing that the i-th measurement on the k-th scan is target-originated and the other $N_z(k)-1$ measurements are clutter-originated
$\mathbf{\Psi}_u(k)$	the u-th association *history* event, which contains a joint association event for each scan from 1 to k
$N_h(k)$	the number of association hypotheses in the tracking system after incorporation of the k-th set of measurements
$\mathbf{\Omega}_{N_h}(k)$	the full parameters (weights, means, covariances) of the N_h association hypotheses after incorporation of the k-th set of measurements
$N_r(k)$	the number of association hypotheses at the end of the k-th processing cycle, after hypothesis reduction has been applied
$\bar{\mathbf{\Omega}}_{N_r}(k)$	the parameters of the reduced set of N_r hypotheses

Multiple Model Methods for Cost Function Based Multiple Hypothesis Trackers

I. *Introduction*

The objective of any tracking algorithm is to maintain an accurate estimate of a target's state, even in the presence of measurement uncertainty, unknown targets dynamics, and clutter. A significant body of work exists that addresses each of these problems – Kalman filters are used to reject measurement noise, multiple model techniques can resolve target dynamics, and, recently, multiple hypothesis techniques have emerged as formidable tools in the rejection of false measurements. For the utmost in performance, a modern tracking algorithm should incorporate an optimal synthesis of these techniques.

1.1 Motivation

In order to maintain a lock on a target, a tracker must maintain an internal estimate of the target's state (positions, velocities, and accelerations), updated each cycle with measurement information acquired from a sensor. But, given the inherent inaccuracies in sensing devices, the measurements cannot be used directly, and Kalman filters are traditionally used as an optimal method of fusing noise-corrupted measurements with kinematic aircraft models.

Kalman filters provide the best results when they have accurate information about the target's performance capabilities, such as maneuverability. These characteristics may not be available during processing, and the filter will be forced to use coarse approximations. The widely accepted method for resolving these unknowns, shown in Figure 1.1, is to run several filters in parallel, each tuned to different assumed dynamics. The output of the tracker is a blended estimate of the filters. Several forms of the multiple model filter exist, differing primarily in whether assumed target dynamics states are assumed to be able to switch hypothesized models between update cycles.

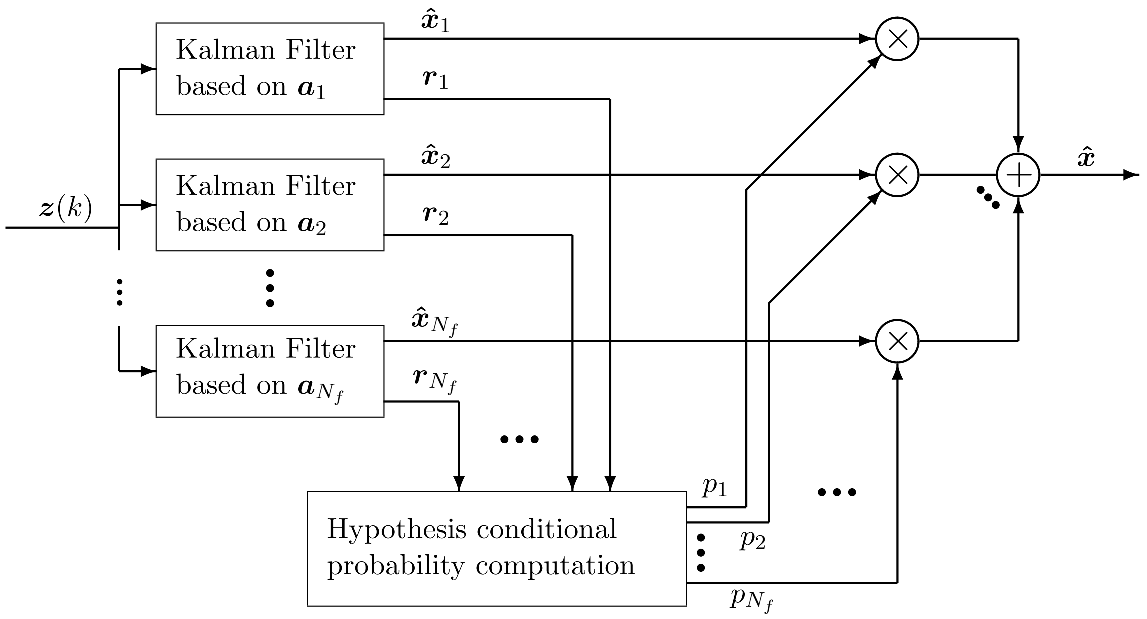

$\hat{\boldsymbol{x}}_1$
Kalman Filter based on $\boldsymbol{a}_1$
$\boldsymbol{r}_1$
$\hat{\boldsymbol{x}}_2$
Kalman Filter based on $\boldsymbol{a}_2$
$\boldsymbol{r}_2$
$\boldsymbol{z}(k)$
$\hat{\boldsymbol{x}}$
$\hat{\boldsymbol{x}}_{N_f}$
Kalman Filter based on $\boldsymbol{a}_{N_f}$
$\boldsymbol{r}_{N_f}$
Hypothesis conditional probability computation
p_1
p_2
p_{N_f}

Figure 1.1: MMAE signal flow block diagram.

Besides measurement and dynamics uncertainty, modern trackers need to deal with measurement uncertainty – whether a measurement is target-originated and, if so, which target. Multiple Hypothesis Trackers (MHT) have emerged as the best solution to this problem [5, 23, 32]. Tractable MHTs are suboptimal implementations of a more rigorous Bayesian solution that maintains a full history of every possible combination of available measurement and target hypothesis data at each time epoch [24,35]. Obviously, the number of solutions grows exponentially with each epoch, and considerable work has been devoted to maintaining only the most meaningful data. Early techniques reduced the number of available hypothesis using *ad hoc* methods such as discarding unlikely hypotheses or fusing all of the available hypotheses into a single estimate each cycle [23]. A more structured approach was proposed that viewed each of the hypotheses as contributing a component to a Gaussian mixture (probability-weighted sum of Gaussians) representation of the Probability Density Function (PDF) in target state space (see Figure 1.2) and reduced the number of components in that approximation using minimized distance, or cost, functions to minimize the differences between Gaussian mixture PDF surfaces before and after that

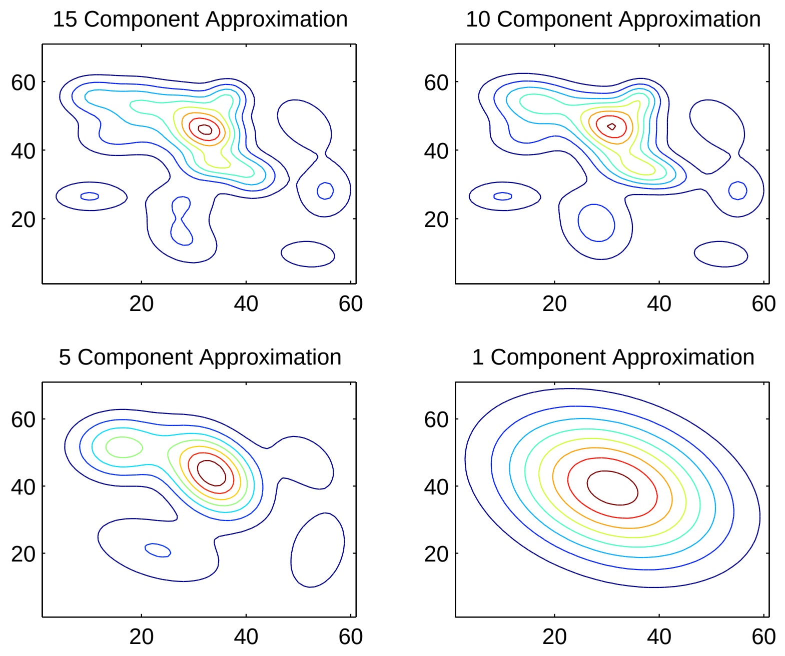


Figure 1.2: Gaussian mixture reduction. Contour plots showing probability distribution for target azimuth and elevation as the fidelity of the representation is degraded by reducing the number of Gaussians in the mixture. Considerable effort has been applied to maintaining the best representation of the original mixture using a minimal number of components.

reduction in number of components [24, 26, 27]. This new procedure provided a considerable improvement in tracking ability in clutter that was significantly improved upon by Williams [35–37], who provided a better cost function based on the Integral Square Error (ISE) distance metric.

The concept of integrating multiple model techniques with multiple hypothesis trackers is not unique, and a substantial amount of research has been made in this arena [14]. Unfortunately, current paradigms do not employ more advanced Mixture Reduction Algorithms (MRAs), most likely due to computational efficiency issues. In a recent study, Smith showed that multiple model techniques could be applied to cost-function-based MHTs and retain a full Gaussian mixture structure between epochs [30, 31]. He accomplished this by maintaining a separate mixture for each elemental filter within a Multiple Model Adaptive Estimator (MMAE) or Interacting Multiple Model (IMM) estimator and basing the multiple model probabilities on approximations of the estimates, covariances, and residuals of those mixtures. He showed that multiple model techniques could be applied to an ISE-based MRA beneficially, but he encountered several problems such as computational loading and measurement association difficulty due to the burden of running several mixtures in parallel. Recently, Torelli et al. [34] proposed an alternate methodology of integrating multiple model structures with MHTs such that each association history hypothesis would spawn a new multiple model filter each epoch. They used an N-scan MRA to limit the number of hypotheses, but their approach could be modified to work with an ISE MRA.

1.2 Research Goal

The goal of this study is to modify existing track-based multiple model techniques so that they can work within an ISE-cost-function-based MHT. The major focus of the research will be to increase the fidelity of the tracker by reducing its root mean square (RMS) miss distance in terms of position and velocity.

While tracking algorithms have been used from everything from schools of fish [28] to individuals in crowds [38], our region of interest involves maneuvering aircraft in clutter. Consequently, the ultimate goal of the simulations will be to provide data quantifying algorithmic performance against realistic flight data using realistic measurement parameters. This should provide considerable insight into capability, since the characteristic behavior of aircraft, namely periods of maneuver separated by long periods of benign flight, lends itself well to a multiple model structure, and tracking such targets in clutter is of substantial importance to the Air Force.

1.3 Organization

Chapter II provides the mathematical foundation for the target tracker by discussing the Kalman filter, multiple model techniques (to include the Multiple Model Adaptive Estimator and the Interacting Multiple Model), and multiple hypothesis trackers with specific emphasis on Williams' ISE mixture reduction algorithm. Chapter III introduces the program architecture necessary to simulate an aircraft target tracker. It describes the methods of generating synthetic and realistic flight data and outlines seven unique tracking architectures that will be used to track the targets from such truth models. Chapter III also includes a discussion of performance enhancing techniques that will be applied to the multiple model structures. Chapter IV presents the results of Monte Carlo simulations for the different path generation techniques, with and without clutter. Specific importance is placed on the improvements in position and velocity errors afforded by the multiple model techniques. Chapter 5 summarizes the results and highlights areas of research that offer potential for future development.

II. Background

2.1 Introduction

Modern tracking algorithms typically perform two basic functions, filtering and measurement association. Filtering is necessary to remove uncertainty in measurements due to sensor errors or changing target dynamics, while measurement association is necessary to handle multiple measurements caused by environmental clutter and targets of no interest, as well as by the target of interest itself. Section 2.3 describes the the Kalman filter, the most commonly used method of rejecting measurement noise and estimating variables beyond those which are directly measured. Because the Kalman filter must be tuned for a certain level of dynamics, and yet targets often exhibit multiple different levels of dynamics, several methods of modifying the Kalman filter to adjust for target maneuver have been proposed. The most widely accepted methods of dealing with target maneuver, multiple model techniques, are outlined in Sections 2.4 and 2.4.3. Section 2.5 discusses the multiple hypothesis tracker, a method of dealing with measurement ambiguity due to clutter.

2.2 Tracking Fundamentals

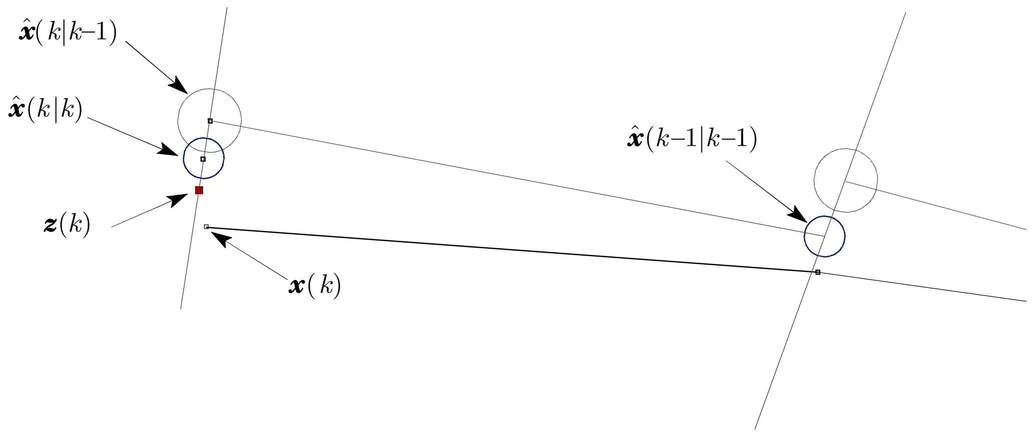

$\hat{\boldsymbol{x}}(k|k{-}1)$
$\hat{\boldsymbol{x}}(k|k)$
$\hat{\boldsymbol{x}}(k{-}1|k{-}1)$
$\boldsymbol{z}(k)$
$\boldsymbol{x}(k)$

Figure 2.1: Canonical Tracker

Figure 2.1 shows two cycles of a simple recursive tracker. Here, the true flight path is $\boldsymbol{x}(k)$ and is a discrete-time representation of a continuous-time process. The target is moving under changing dynamics and produces a noise-corrupted measurement, $\boldsymbol{z}(k)$, at each epoch. An estimated path generated by connecting sequential measurements would be excessively noisy, so modern algorithms use dynamics models within a filter propagation cycle structure to smooth the results. In Figure 2.1, the target state estimate at time k conditioned on all measurements up until time $k-1$, $\hat{\boldsymbol{x}}(k|k-1)$, was generated by taking the previous estimate of location, $\hat{\boldsymbol{x}}(k-1|k-1)$, and propagating it forward using a dynamics model state transition matrix, $\boldsymbol{\Phi}$, that assumes a constant velocity. A multitude of dynamics models exist, each based on different assumptions about the motion of the target and the adequacy of that model as well as the strengths of disturbances affecting that model. These models are presented in Section 2.3.1

At time k, the tracker has two indications of position, one corrupted by the accuracy of the measurement device, and another corrupted by the uncertainty of the target dynamics. The best estimate of true position that the tracker can provide, $\hat{\boldsymbol{x}}(k|k)$, will be a point on a line between the two, as seen in Figure 2.1. The exact interpolation will be based on the ratio of the two uncertainties – more measurement uncertainty would push the estimate towards $\hat{\boldsymbol{x}}(k|k-1)$, while a more maneuverable target or uncertain target dynamics would push the estimate towards the measurement.

The selection of this interpolation can be totally *ad hoc*, yielding the $\alpha-\beta$ tracker (assuming constant-velocity trajectories), or the $\alpha-\beta-\gamma$ tracker (assuming constant-acceleration paths instead) if you wish to estimate acceleration [2, 5]. However, most modern tracking algorithms rely on a Kalman filter to handle propagation and update cycles because it provides a solid mathematical basis and guarantee of optimality for the weighting coefficients.

2.3 Kalman Filtering

As opposed to the simple kinematic trackers listed previously, Kalman filters maintain the conditioned probability density function (PDF) of the target state, conditioned on measurements observed, in a Gaussian form. The Kalman filter provides optimal results given that certain criteria are satisfied: the system must be linear, and the measurement and process noises must be white and Gaussian. The equations for implementing a Kalman filter are presented in the next three sections. For a full discussion of the subject, the reader is referred to the treatment by Maybeck [15].

2.3.1 Dynamics Design Models. Dynamics design models are responsible for describing how the state of a target evolves over time. Such models maintain the position, velocity and sometimes acceleration (and higher order derivative) states of the target and are based on simple kinematic properties. To function within the architecture of a Kalman filter, the models must be linear and the disturbance driving noise must be modelled as white Gaussian noise. While nonlinear dynamics models, such as the coordinated turn-rate model [3, 18], do exist, they would require the use of an extended Kalman Filter (EKF), a variant that allows for minor nonlinearities in model dynamics [16]. The models shown here are presented for two decoupled dimensions, which would correspond to a track projected onto a Cartesian plane as seen by the tracker. Many trackers can incorporate range measurements to model the target in three dimensions using range and angle measurements, but these models break the assumption of linearity, thereby requiring more complex filters such as the EKF.

In the constant velocity (CV) target model, the uncertainty is modelled as noise injected into the acceleration state of the target as shown by the state equation:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}_x(t) \\ \dot{y}(t) \\ \dot{v}_y(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ v_x(t) \\ y(t) \\ v_y(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \tag{2.1}$$

where $\boldsymbol{w}(t)$ represents input noise with statistics:

$$\begin{aligned} E[\boldsymbol{w}(t)] &= \mathbf{0} \\ E[\boldsymbol{w}(t)\boldsymbol{w}^{\mathrm{T}}(t')] &= \mathbf{Q}\,\delta(t-t') \end{aligned} \tag{2.2}$$

Another generalized model assumes that acceleration is a first-order (exponentially time-correlated) Gauss-Markov process, the output of a first-order lag (with lag coefficient $\frac{1}{T}$)

driven by zero-mean white noise, represented by the state equation:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}_x(t) \\ \dot{a}_x(t) \\ \dot{y}(t) \\ \dot{v}_y(t) \\ \dot{a}_y(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{T} \end{bmatrix} \begin{bmatrix} x(t) \\ v_x(t) \\ a_x(t) \\ y(t) \\ v_y(t) \\ a_y(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{T} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{T} \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \tag{2.3}$$

As the time constant in the FOGMA model becomes larger, the $\frac{1}{T}$ value approaches zero, corresponding to the constant acceleration (CA), or white jerk model:

$$\dot{a}_x(t) = 0 + w_1(t) \tag{2.4}$$

and similarly for the y axis [6].

2.3.2 Propagation. Having defined a dynamics model representation and starting from a known state $\hat{\boldsymbol{x}}_0$, the target state at time k can be predicted using:

$$\hat{\boldsymbol{x}}(k|k-1) = \boldsymbol{\Phi}(k, k-1)\hat{\boldsymbol{x}}(k-1|k-1) \tag{2.5}$$

where $\boldsymbol{\Phi}(k, k-1)$ represents the state transition matrix, which defines the changes in state between time $k-1$ and k.

Because we now assume a stochastic system wherein the states, $\hat{\boldsymbol{x}}$, of the target are never certain and are represented as Gauss-Markov processes and Gaussians at any specified time or epoch, their covariances will be computed in a matrix $\mathbf{P}$. Upon predicting the states forward in time, their accuracy will invariably decline, and their new values are calculated as:

$$\mathbf{P}(k|k-1) = \boldsymbol{\Phi}(k, k-1)\mathbf{P}(k-1|k-1)\boldsymbol{\Phi}(k, k-1)^{\mathrm{T}} + \mathbf{Q}_d(k-1) \tag{2.6}$$

where $\mathbf{Q}_d$ is the discrete-time representation of the process noise strength.

2.3.3 Measurement Incorporation. At any given epoch, a measurement may or may not be available. If measurements are available, the Kalman filter will enter an update cycle. In the event that no measurement is available, the cycle will continue with sequential propagation cycles without update, and the covariance will grow accordingly. By the constraints of the Kalman filter, the measurement will be modelled as the sum of a linear transformation of the state corrupted by white, Gaussian noise:

$$\mathbf{z}(k) = \mathbf{H}(k)\boldsymbol{x}(k) + \boldsymbol{v}(k) \tag{2.7}$$

where $\mathbf{H}$ portrays the interrelationship between measurements and state variables, and $\boldsymbol{v}(k)$ has statistics:

$$\begin{aligned} E\{\boldsymbol{v}(k)\} &= \mathbf{0} \\ E\{\boldsymbol{v}(k)\boldsymbol{v}^{\mathrm{T}}(l)\} &= \begin{cases} \mathbf{R}(k) & k = l \\ \mathbf{0} & k \neq l \end{cases} \end{aligned} \tag{2.8}$$

As mentioned in Section 2.2, we desire an interpolation between the propagated estimate and the noise-corrupted measurement. The Kalman filter provides an optimal interpolation (with respect to almost any criteria [15]) through the Kalman filter gain:

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}^{\mathrm{T}}(k)\left[\mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}^{\mathrm{T}}(k) + \mathbf{R}(k)\right]^{-1} \tag{2.9}$$

This gain is used to incorporate the measurement into the propagated state and covariance by:

$$\hat{\boldsymbol{x}}(k|k) = \hat{\boldsymbol{x}}(k|k-1) + \mathbf{K}(k)\left[\boldsymbol{z}_k - \mathbf{H}(k)\hat{\boldsymbol{x}}(k|k-1)\right] \tag{2.10}$$

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{K}(k)\mathbf{H}(k)\mathbf{P}(k|k-1) \tag{2.11}$$

creating the updated state at time instant k. Given that the Kalman filter is recursive, this estimate and error covariance represent the Gaussian PDF of the state of the target conditioned on all of the previous measurements and are the only values that need to be carried into the next filter cycle.

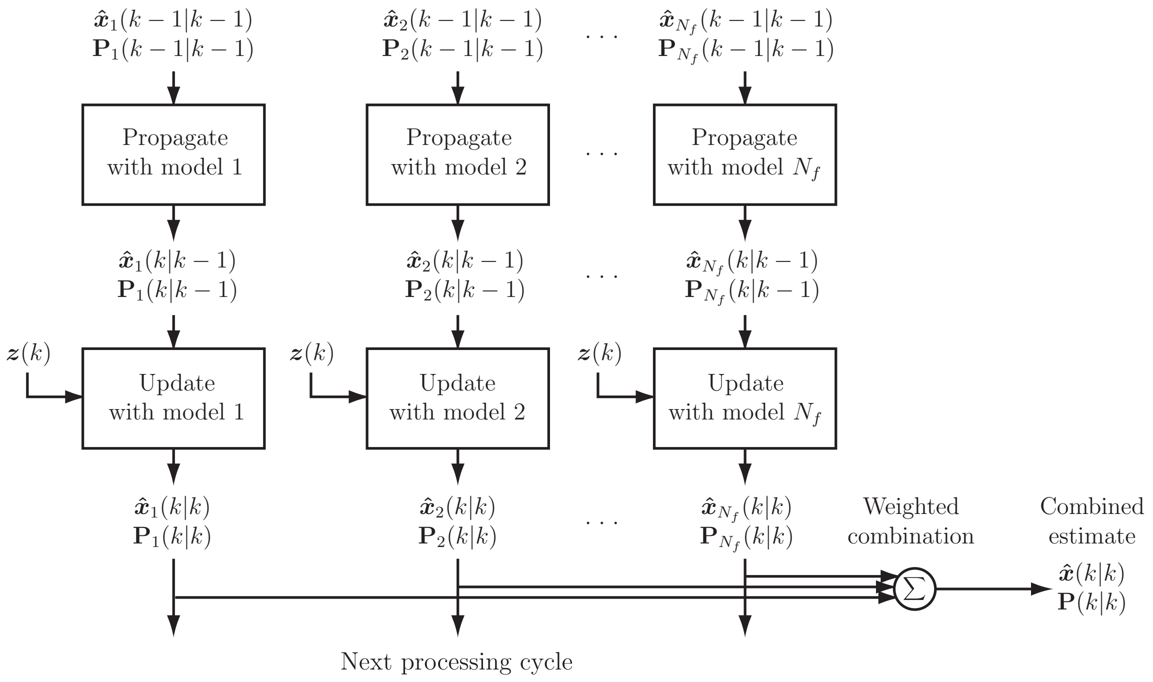
2.4 Multiple Model Estimation

Optimality of the Kalman filter depends on the filter dynamics model adequately portraying the target dynamics model. Unfortunately, the dynamics of a target are usually uncertain, so the appropriate filter tuning at any time is not guaranteed. One could simply tune the process noise for a filter so that it could accommodate all foreseen dynamics levels, but this would sacrifice noise rejection capabilities during time periods in which the target actually exhibits benign trajectories. In fact, aircraft tend to travel in two modes, maneuvering and non-maneuvering. Worse yet, they tend to spend more time in the benign periods so that a high bandwidth filter would be operating outside its design region for the greater portion of time if it were used as a single filter instead of an elemental filter in a multiple model architecture. Various *ad hoc* methods exist to account for uncertainties in $\mathbf{Q}_d$, such as residual "whitening" and covariance matching [16], but the multiple model techniques outlined here are widely regarded as providing superior results.

2.4.1 Non-Switching Models.

Figure 2.2 shows a parallel bank of filters that discretize the maneuver parameter space, a configuration referred to as the Multiple Model Adaptive Estimator (MMAE). A blended estimate of the mean and covariance is acquired using a weighted average of the individual elemental filters (each based on a particular dynamics model). This configuration is referred to as a non-switching model because the assumed target dynamics do not switch modes or filters between cycles, meaning that these filters run completely independently of each other, with the exception of the *ad hoc* techniques in Section 2.4.2.

Considering that the multiple model configuration in Figure 2.2 is based on Kalman filters, the conditional density associated with each filter, j, is represented as a Gaussian and is based on the entire measurement history up to and including the current time, $\boldsymbol{Z}^k = \{\boldsymbol{z}(k), \boldsymbol{z}(k-1), ...\boldsymbol{z}(1)\}$, recursively conditioned on the individual $\mathbf{H}_j$ and $\mathbf{R}_j$ values for that model. In order to obtain a single estimate of the target state at time k, a



$\hat{\boldsymbol{x}}_1(k-1|k-1)$ $\mathbf{P}_1(k-1|k-1)$
$\hat{\boldsymbol{x}}_2(k-1|k-1)$ $\mathbf{P}_2(k-1|k-1)$
$\hat{\boldsymbol{x}}_{N_f}(k-1|k-1)$ $\mathbf{P}_{N_f}(k-1|k-1)$
Propagate with model 1
Propagate with model 2
Propagate with model N_f
$\hat{\boldsymbol{x}}_1(k|k-1)$ $\mathbf{P}_1(k|k-1)$
$\hat{\boldsymbol{x}}_2(k|k-1)$ $\mathbf{P}_2(k|k-1)$
$\hat{\boldsymbol{x}}_{N_f}(k|k-1)$ $\mathbf{P}_{N_f}(k|k-1)$
$\boldsymbol{z}(k)$
Update with model 1
Update with model 2
Update with model N_f
$\hat{\boldsymbol{x}}_1(k|k)$ $\mathbf{P}_1(k|k)$
$\hat{\boldsymbol{x}}_2(k|k)$ $\mathbf{P}_2(k|k)$
$\hat{\boldsymbol{x}}_{N_f}(k|k)$ $\mathbf{P}_{N_f}(k|k)$
Weighted combination
Combined estimate
$\hat{\boldsymbol{x}}(k|k)$ $\mathbf{P}(k|k)$
Next processing cycle

Figure 2.2: Non-Switching Multiple Model Filter

probability weighted average of the filter estimates is formed via:

$$\begin{aligned}
\hat{\boldsymbol{x}}(k|k) &= \sum_{j=1}^{N_f} \mu_j(k)\hat{\boldsymbol{x}}_j(k|k) \\
\mathbf{P}(k|k) &= \sum_{j=1}^{N_f} \mu_j(k)\{\mathbf{P}_j(k|k) + [\hat{\boldsymbol{x}}_j(k|k) - \hat{\boldsymbol{x}}(k|k)][\hat{\boldsymbol{x}}_j(k|k) - \hat{\boldsymbol{x}}(k|k)]^{\mathrm{T}}\}
\end{aligned} \tag{2.12}$$

The μ_j term represents the probability that the particular filter is in force at the given time, conditioned upon the entire available measurement history and is defined as:

$$\mu_j(k) \triangleq P\{M_j|\boldsymbol{Z}^k\} \tag{2.13}$$

where M_j is the j-th assumed mode or model. Separating the measurement history into current and previous yields:

$$\mu_j(k) = P\{M_j|\boldsymbol{Z}^{k-1}, \boldsymbol{z}(k)\} \tag{2.14}$$

and applying Bayes' rule for both $\boldsymbol{z}(k)$ and M_j yields:

$$\begin{aligned}\mu_j(k) &= \frac{f\{M_j, \boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\}}{f\{\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\}} \\ &= \frac{f\{\boldsymbol{z}(k)|M_j, \boldsymbol{Z}^{k-1}\}P\{M_j|\boldsymbol{Z}^{k-1}\}}{f\{\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\}}\end{aligned} \tag{2.15}$$

The denominator in Equation (2.15) can be expanded using the total probability expansion over all models:

$$f\{\boldsymbol{z}(k)|\boldsymbol{Z}^{k-1}\} = \sum_{i=1}^{N_f} f\{\boldsymbol{z}(k)|M_i, \boldsymbol{Z}^{k-1}\}P\{M_i|\boldsymbol{Z}^{k-1}\} \tag{2.16}$$

where N_f is the elemental filters in the structure. This gives the following recursive equation for the model probabilities $\mu_j(k)$:

$$\mu_j(k) = \frac{f\{\boldsymbol{z}(k)|M_j, \boldsymbol{Z}^{k-1}\}\mu_j(k-1)}{\sum_{i=1}^{N_f} f\{\boldsymbol{z}(k)|M_i, \boldsymbol{Z}^{k-1}\}\mu_i(k-1)} \tag{2.17}$$

The denominator is simply the sum of all such numerator terms and ensures that the sum of probabilities is unity, and the conditional density in the numerator is Gaussian and can be evaluated as:

$$f\{\boldsymbol{z}(k)|M_j, \boldsymbol{Z}^{k-1}\} = \frac{1}{(2\pi)^{s/2}|\mathbf{S}_j(k)|^{1/2}}\exp\{\cdot\} \tag{2.18}$$

$$\{\cdot\} = \{-\frac{1}{2}\boldsymbol{r}_j^{\mathrm{T}}(k)\mathbf{S}_j^{-1}(k)\boldsymbol{r}_j(k)\} \tag{2.19}$$

$$\boldsymbol{r}_j(k) = \boldsymbol{z}(k) - \mathbf{H}_j(k)\hat{\boldsymbol{x}}_j(k|k-1) \tag{2.20}$$

$$\mathbf{S}_j(k) = \mathbf{H}_j(k)\mathbf{P}_j(k|k-1)\mathbf{H}_j^{\mathrm{T}}(k) + \mathbf{R}_j(k) \tag{2.21}$$

where s is the measurement dimension.

2.4.2 Ad Hoc MMAE Techniques. Figure 2.3 shows two Gaussians that represent the individual filter estimates of a two-filter MMAE. The red Gaussian has a smaller variance indicating more surety in the estimate than from the other filter. For this de-

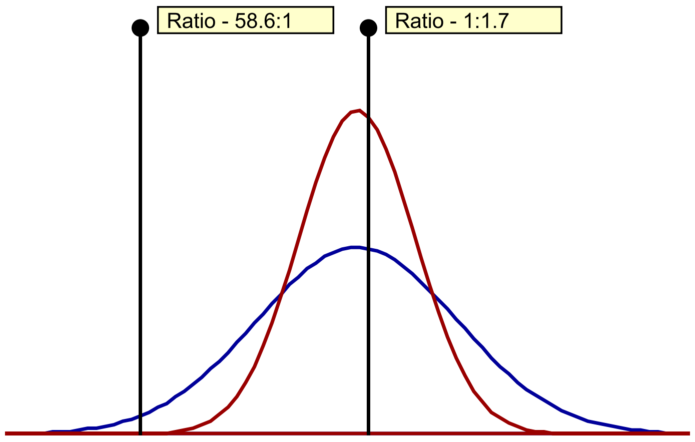


Figure 2.3: Multiple Filter Probability Ratios. The ratio between the PDF values of two Gaussians of the same mean but differing variances is shown for two different realizations (represented by the two vertical lines). The ratio is presented as aggressive:benign and shows that a benign filter is a very poor match to an out of bandwidth measurement, while both the aggressive and benign filters are reasonably good matches for a measurement near their identical means.

velopment, these represent the prototypical benign and aggressive filters. From Equation (2.17), we see that the probability flow between filters depends on the ratio of the filter PDFs evaluated at the incoming measurement. Two measurements, represented as vertical black lines, are shown on Figure 2.3. The measurement on the far left indicates that the aggressive filter has approximately 60 times as much probability as the benign. Equation (2.17) is a recursive function, so the actual flow is also a function of the previous probability values, but the tendency for rapid probability transfer is evident. On the other hand, the alternate measurement near the peaks of both Gaussians indicates that the benign filter has only twice as much probability as the aggressive filter. Consequently, the flow from an aggressive filter to a benign filter will be significantly slower than probability flow in the other direction.

The advantage of MMAEs lies in the ability of a designer to use filters that are too benign for use as a single filter for a dynamic system during its periods of stringent motion. These filters would provide superior filtering when in force while the system actually exhibits benign motion, but the idiosyncracies of MMAE probability flow prevent them from being used for the longest possible periods because of slow natural flow of probability back to benign filters when this is, in fact, appropriate. Korn and Beean [11] proposed an *ad hoc* method to return probability to the more benign filters by periodically resetting the filter probabilities so that the majority of probability is held by the benign filter. If the probabilities are reset when aggressive filter is a closer match to truth than the benign filter, the probability flow should return quickly to the appropriate filter, incurring a minimal amount of error. Otherwise, there is a quicker proper return of probability to the benign filter (when appropriate) than would be accomplished without this method.

Additionally, because of the exponential nature of the PDF and the repetitive implementation of Equation (2.17), after several unfavorable iterations, the μ_j values can become extremely small. It follows that a return to significant probability would require an equal number of favorable iterations. Given the uneven flow caused by probability ratios, this could lead to unacceptable transition times. In fact, on a digital system, the probability could underflow and never return. So, most designers impose lower bounds on filter probabilities, the determination of which is best based on experimental criteria. A higher value of the lower bound will create more agile probability flow, but will allow error from out-of-favor filters to enter inappropriately strongly into the blended estimate of the MMAE. Additionally, in systems with large numbers of filters, a higher lower bound will cause the lower bounded filters to represent enough of the blended estimate to corrupt the result. In practice, a value of .001 seems to offer the best compromise [17]. Note that the lower bound violates the normalization of Equation (2.17), so one must be certain to perform an additional normalization of all modal probabilities after lower bounding to ensure proper summation to unity.

Another *ad hoc* modification required for practical use of the MMAE algorithm is monitoring filters for divergence and resetting them when necessary. Traditionally, the normalized residual $\boldsymbol{r}(k)^T\mathbf{S}(k)^{-1}\boldsymbol{r}(k)$ is used to identify divergent filters. In a well-matched filter, this value should be comparable to the measurement dimension; when it surpasses a certain threshold (to be determined by the designer for each situation), the filter will have its state reset to the blended estimate of the non-divergent filters [16].

2.4.3 Switching Models. Switching models work under the assumption that, between filter cycles, there is a probability, $p_{i,j}$, that the target will transition from model state i to state j. These values are stored in a Markov probability transition matrix:

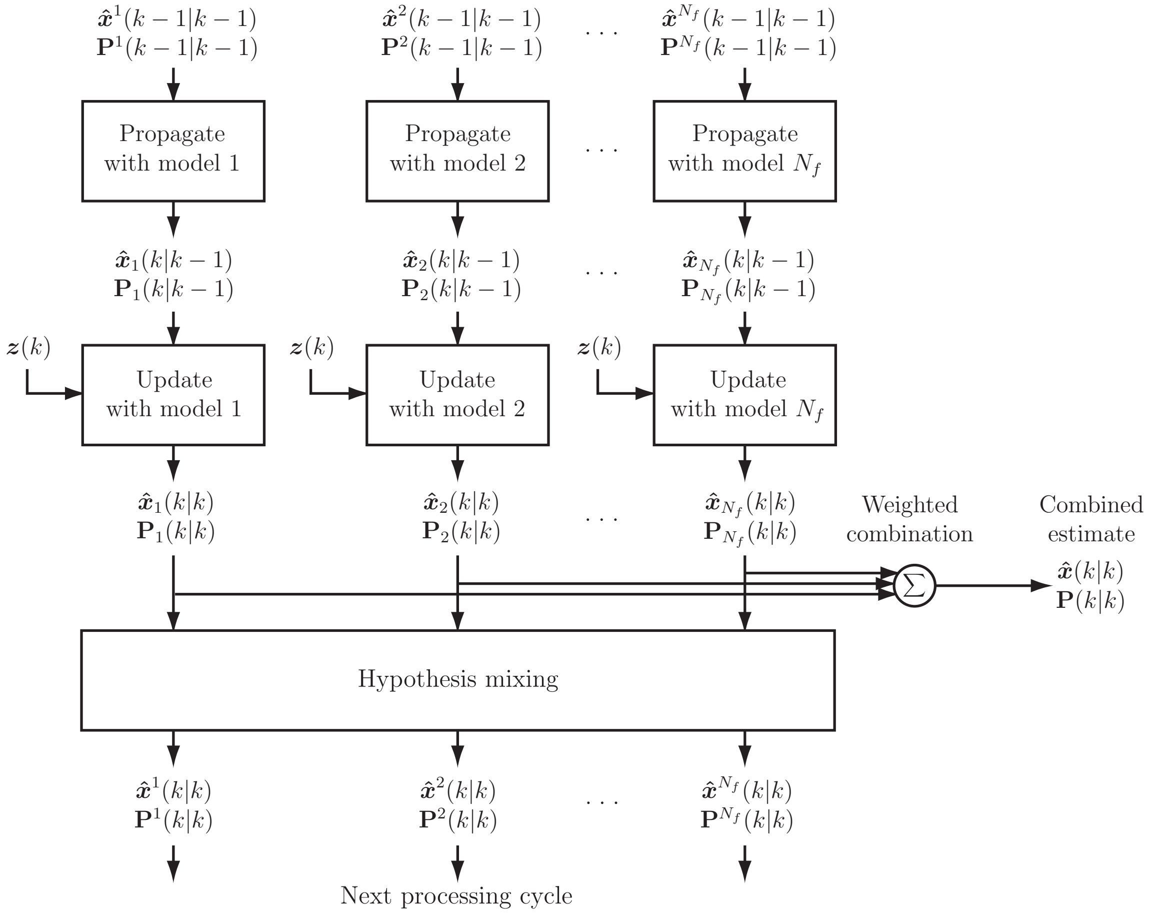
$$\mathbf{T}(k|k-1) = \begin{bmatrix} p_{1,1}(k) & p_{2,1}(k) & \cdots & p_{N_f,1}(k) \\ p_{1,2}(k) & p_{2,2}(k) & \cdots & p_{N_f,2}(k) \\ \vdots & \vdots & \vdots & \vdots \\ p_{1,N_f}(k) & p_{2,N_f}(k) & \cdots & p_{N_f,N_f}(k) \end{bmatrix} \tag{2.22}$$

Being a Markov process, the probability of any model being in force at time instant k, $P\{M_{k,l}\}$, depends only on the previous model probability:

$$\begin{bmatrix} P\{M_{k,1}\} \\ P\{M_{k,2}\} \\ \vdots \\ P\{M_{k,N_f}\} \end{bmatrix} = \mathbf{T}(k|k-1) \begin{bmatrix} P\{M_{k-1,1}\} \\ P\{M_{k-1,2}\} \\ \vdots \\ P\{M_{k-1,N_f}\} \end{bmatrix} \tag{2.23}$$

Because the total probability of reaching a certain mode from all other modes must be unity, the columns of $\mathbf{T}(k|k-1)$ must sum to one. Otherwise, the actual values are up to the discretion of the designer and can be viewed as tuning parameters specific to each application. For the cases of target tracking, the matrices tend to be diagonally dominant.

Considering that each model has the ability to switch to any of the other N_f models at each epoch, the number of models existing at time k will be N_f^k, following an exponential growth curve. Commonly referred to as a full order Markov switching estimator, the full Bayesian solution to switching model dynamics is computationally intractable, but remains the basis for several suboptimal approximations. Namely, the First-Order Generalized Pseudo-Bayesian Estimator (GPB-1) algorithm limits the memory of the model by combining the available models into a single representative model at the end of each processing cycle. This limits the number of models that have to be retained to only N_f. This new estimate is used as the input to the multiple model structure at the beginning of the next cycle. The GPB-1 algorithm is a very coarse approximation to the full order system,



$\hat{\boldsymbol{x}}^1(k-1|k-1)$
$\mathbf{P}^1(k-1|k-1)$
$\hat{\boldsymbol{x}}^2(k-1|k-1)$
$\mathbf{P}^2(k-1|k-1)$
$\hat{\boldsymbol{x}}^{N_f}(k-1|k-1)$
$\mathbf{P}^{N_f}(k-1|k-1)$
Propagate with model 1
Propagate with model 2
Propagate with model N_f
$\hat{\boldsymbol{x}}_1(k|k-1)$
$\mathbf{P}_1(k|k-1)$
$\hat{\boldsymbol{x}}_2(k|k-1)$
$\mathbf{P}_2(k|k-1)$
$\hat{\boldsymbol{x}}_{N_f}(k|k-1)$
$\mathbf{P}_{N_f}(k|k-1)$
$\boldsymbol{z}(k)$
Update with model 1
Update with model 2
Update with model N_f
$\hat{\boldsymbol{x}}_1(k|k)$
$\mathbf{P}_1(k|k)$
$\hat{\boldsymbol{x}}_2(k|k)$
$\mathbf{P}_2(k|k)$
$\hat{\boldsymbol{x}}_{N_f}(k|k)$
$\mathbf{P}_{N_f}(k|k)$
Weighted combination
$\sum$
Combined estimate
$\hat{\boldsymbol{x}}(k|k)$
$\mathbf{P}(k|k)$
Hypothesis mixing
$\hat{\boldsymbol{x}}^1(k|k)$
$\mathbf{P}^1(k|k)$
$\hat{\boldsymbol{x}}^2(k|k)$
$\mathbf{P}^2(k|k)$
$\hat{\boldsymbol{x}}^{N_f}(k|k)$
$\mathbf{P}^{N_f}(k|k)$
Next processing cycle

Figure 2.4: Interacting Multiple Model Filter

and the GPB-2 algorithm tries to increase the fidelity of the estimate by extending the memory length to retain the previous two time steps, which requires maintaining N_f^2 filter models at a given time. A brief summary of these models has been presented here; for a full discussion, the reader is directed to the discussions by Williams [35] and Petrucci [21].

Figure 2.4 shows a block diagram of the Interacting Multiple Model Estimator, an algorithm that achieves comparable performance to the GBP-2 algorithm using only N_f filters. The filter operates under the assumption that the input to the j-th filter should be the best estimate of the state conditioned on the probability that the model j is in force at that time instant.

2.4.3.1 Mixing Process. The structure of the IMM is similar to the MMAE with the addition of the mixing cycle. In fact, if the Markov transition matrix is set to the identity matrix, implying that no mixing occurs, the IMM would be functionally identical to an MMAE without restarts. The mixing process here is summarized from Williams [35]. At the end of every mixing cycle, we desire the mixed estimates and *a posteriori* modal probabilities for recursion into the next cycle. The mixed estimates can be calculated as:

$$\begin{aligned} \hat{\boldsymbol{x}}^i(k|k) &= \sum_{j=1}^{N_f} P\{M_{k,j}|M_{k+1,i}, \boldsymbol{Z}^k\}\hat{\boldsymbol{x}}_j(k|k) \qquad (2.24) \\ \mathbf{P}^i(k|k) &= \sum_{j=1}^{N_f} P\{M_{k,j}|M_{k+1,i}, \boldsymbol{Z}^k\}\{\mathbf{P}_j(k|k) + \\ &\qquad + [\hat{\boldsymbol{x}}_j(k|k) - \hat{\boldsymbol{x}}^i(k|k)][\hat{\boldsymbol{x}}_j(k|k) - \hat{\boldsymbol{x}}^i(k|k)]^T\} \qquad (2.25) \end{aligned}$$

where the mixing probabilities can be calculated as:

$$\begin{aligned} P\{M_{k,j}|M_{k+1,i}, \boldsymbol{Z}^k\} &= \frac{P\{M_{k,j}, M_{k+1,i}|\boldsymbol{Z}^k\}}{P\{M_{k+1,i}|\boldsymbol{Z}^k\}} \\ &= \frac{P\{M_{k+1,i}|M_{k,j}, \boldsymbol{Z}^k\}P\{M_{k,j}|\boldsymbol{Z}^k\}}{P\{M_{k+1,i}|\boldsymbol{Z}^k\}} \\ &= \frac{P\{M_{k+1,i}|M_{k,j}, \boldsymbol{Z}^k\}P\{M_{k,j}|\boldsymbol{Z}^k\}}{\sum_{n=1}^{N_f} P\{M_{k+1,i}|M_{k,n}, \boldsymbol{Z}^k\}P\{M_{k,n}|\boldsymbol{Z}^k\}} \qquad (2.26) \end{aligned}$$

Dropping the $\boldsymbol{Z}^k$ dependency due to the Markov assumption and substituting the Markov mixing matrix $\mathbf{T}_{ij} = P\{M_{k+1,i}|M_{k,j}\}$ yields:

$$P\{M_{k,j}|M_{k+1,i}, \boldsymbol{Z}^k\} = \frac{\mathbf{T}_{ij}P\{M_{k,j}|\boldsymbol{Z}^k\}}{\sum_{n=1}^{N_f} \mathbf{T}_{ij}P\{M_{k,n}|\boldsymbol{Z}^k\}} \qquad (2.27)$$

Finally, applying the likelihood equation garners the final modal probabilities after mixing:

$$P\{M_{k,j}|\boldsymbol{Z}^k\} = \frac{f\{\boldsymbol{z}(k)|M_{k,j}, \boldsymbol{Z}^{k-1}\} \sum_{i=1}^{N_f} P\{M_{k,j}|M_{k-1,i}\}P\{M_{k-1,i}|\boldsymbol{Z}^{k-1}\}}{\sum_{n=1}^{N_f} f\{\boldsymbol{z}(k)|M_{k,n}, \boldsymbol{Z}^{k-1}\}P\{M_{k,n}|\boldsymbol{Z}^{k-1}\}} \qquad (2.28)$$

The output of the estimator is generated by created a blended estimate of the individual filter values as in Equation (2.12).

2.5 Multiple Hypothesis Tracking

In the presence of clutter, the tracker is presented with the problem of data association – determining which measurements are target-originated and which are of no interest. Usually, this cannot be accomplished in a single scan, so the tracker will defer making hard decisions about the nature of any measurement until a later time, when new information has come in to aid the decision. Because of this deferred decision process, the accuracy of the estimate at any time is not certain. The best estimate of an MHT can and will deviate far from the truth, as is shown in Figure 2.5. A fully Bayesian tracker would maintain a track for every possible combination of measurements, ensuring that the correct track is maintained, but such a design is entirely infeasible. Modern multiple hypothesis trackers are suboptimal approximation to the full Bayesian solution that use hypothesis reduction techniques to discard data associations that are unlikely to be correct.

2.5.1 Gaussian Mixtures. The uncertain nature of an MHT's estimate of the target state implies a multi-modal PDF. The Gaussian mixture is an effective way of representing the target state and can be represented as:

$$f\{\boldsymbol{x}(k)|\boldsymbol{Z}^k\} = \sum_{i=1}^{N} p_i(k)\mathcal{N}\{\boldsymbol{x};\hat{\boldsymbol{x}}_i(k|k),\mathbf{P}_i(k|k)\} \tag{2.29}$$

where p_i are the mixture weights of each Gaussian component (calculated by Equation (2.35) or (2.36)) with mean $\hat{\boldsymbol{x}}$ and covariance $\mathbf{P}$. The probability weights are positive and sum to unity. The mixture contains N components with a higher number of components allowing higher fidelity of representation – an important notion that will be developed further in Section 2.5.4. The overall mean and covariance of a Gaussian mixture can be

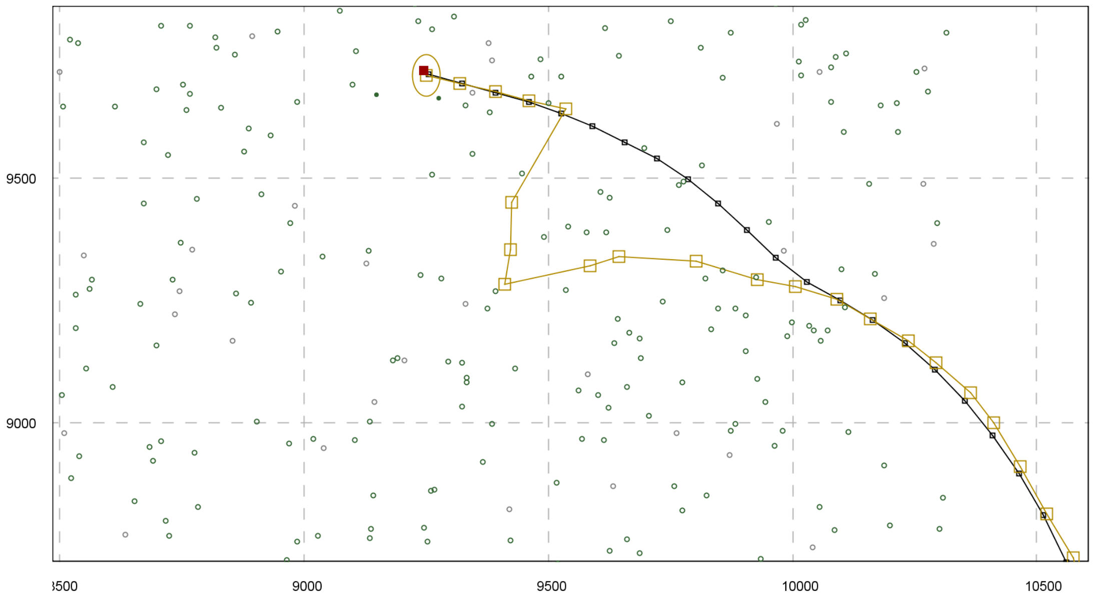


Figure 2.5: Deferred Decision Making. A target, indicated with a black trail, is traversing the field. The best estimate of an MHT, shown in orange, is superimposed. Note that the best estimate deviates by over 500 meters at one point. At the point of separation, the tracker-generated hypotheses based on clutter seemed more likely to be true than the target-originated measurements. But, the tracker still created and maintained hypotheses for the target-originated measurements, even though it deemed them less likely at the time. Later, when the divergent path falls out of favor due to the poor match of the random clutter path to the assumed dynamics of the target, the tracker reverts back to the correct target-originated path.

calculated as:

$$\hat{\boldsymbol{x}}_c = \sum_{i=1}^{N} p_i \hat{\boldsymbol{x}}_i \tag{2.30}$$

$$\mathbf{P}_c = \sum_{i=1}^{N} p_i \left[\mathbf{P}_i + (\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_c)(\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_c)^T\right] \tag{2.31}$$

While Equation (2.30) is a simple weighted average of the mixture means, the overall covariance in Equation (2.31) carries additional "mean-spreading" terms (visualized in Figure 2.6) that increase the mixture covariance based on the dispersion of the individual components from the overall mean.

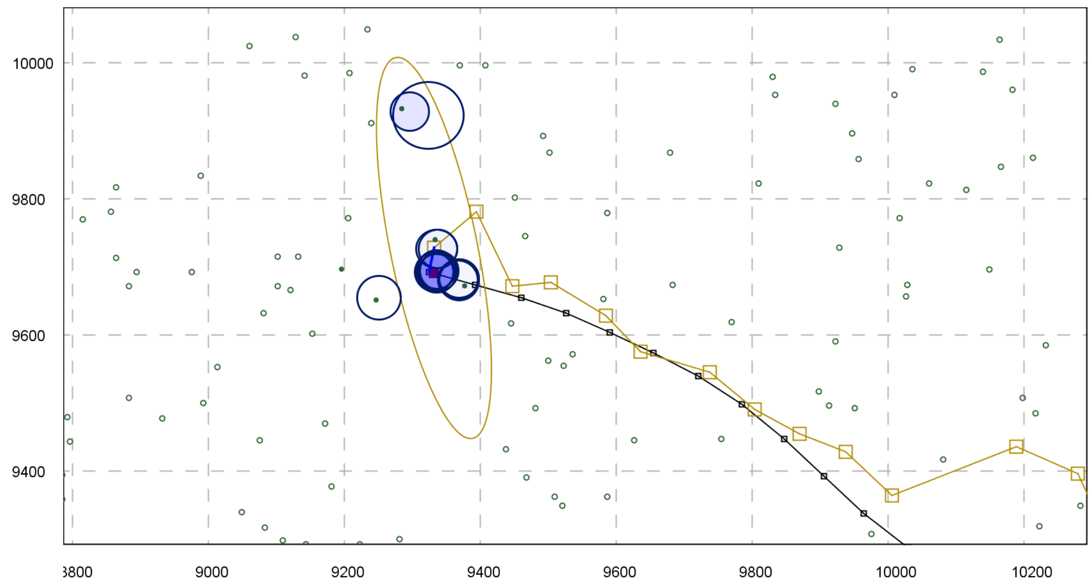


Figure 2.6: Effect of Mean Spreading Terms in a Gaussian Mixture. A Gaussian mixture is composed of individual hypothesis represented as blue ellipses. The approximation, as defined by Equations (2.30) and (2.31), is shown in orange and encompasses the significantly weighted components from the multi-modal distribution (note the low probability component outside the combined estimate) but creates considerable distortion in the PDF by implying probability where none existed before (consider the region below the true position of the target).

2.5.2 Measurement Gating. Measurement gating is employed in nearly all trackers to discard associations that are grossly incorrect for kinematic or statistical reasons. Gating is typically performed by discarding measurements that are too far away from the predicted target location using a Mahalanobis type distance. The distance for the filter i, considering measurement j at time k would be:

$$\gamma \geq [\boldsymbol{z}_j(k) - \hat{\boldsymbol{z}}_i(k|k-1)]^{\mathrm{T}}\mathbf{S}_i(k)^{-1}[\boldsymbol{z}_j(k) - \hat{\boldsymbol{z}}_i(k|k-1)] \tag{2.32}$$

where $\boldsymbol{z}_j$ is the j-th measurement, $\hat{\boldsymbol{z}}_i$ is the predicted measurement for hypothesis i, and γ is a threshold chosen from χ^2 tables to ensure a certain percentage of the PDF hypervolume is selected.

The evaluation of γ yields an accurate elliptical gate, but requires a computationally expensive inverse matrix computation that can be avoided by using first using a square gate that approximates the ellipse. By dividing both sides of Equation (2.32) by γ, we obtain:

$$[\boldsymbol{z}_j(k) - \hat{\boldsymbol{z}}_i(k|k-1)]^{T}[\gamma\mathbf{S}_i(k)]^{-1}[\boldsymbol{z}_j(k) - \hat{\boldsymbol{z}}_i(k|k-1)] \leq 1 \tag{2.33}$$

In our 2D tracking case, this equation represents the intersection of a plane and a 2D Gaussian density surface, yielding an ellipse. The major and minor axes of this ellipse will be the square roots of the eigenvalues of $\gamma\mathbf{S}_i(k)$. The appropriate square gate should be centered on the estimate and have a side of $2\sqrt{\lambda_{max}}$. Figure 2.7 shows the association gates for a tracker. Clutter points outside the square gates will be removed from consideration (a computationally inexpensive test), and the remaining points will be compared to the elliptical gates (at significant computational expense). Only those measurements passing both tests are incorporated.

2.5.3 State Updates with Measurement Uncertainty. Consider that, instead of receiving a single measurement $\boldsymbol{z}(k)$, the tracker receives several measurements $\boldsymbol{Z}^k$. The PDF of the target state is represented as a mixture of Gaussians and can be updated using the standard Kalman filter update equations. The association history event probabilities $P\{\boldsymbol{\Psi}_{u'}(k)|\boldsymbol{Z}^k\}$ which are stored as the mixture probability weights are calculated using

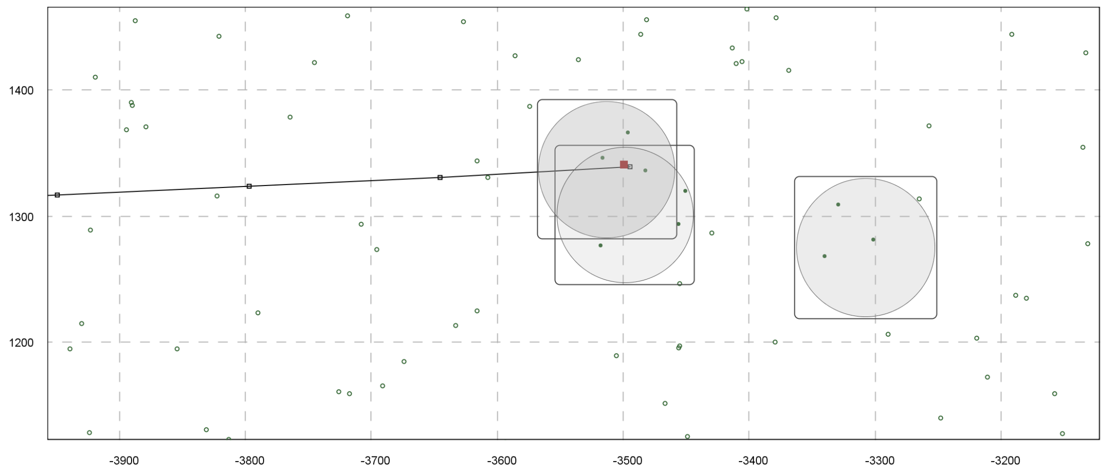


Figure 2.7: Gating Regions for Uncertain Measurements and a Single Target. A target is traversing the field from left to right. The tracker maintains a Gaussian mixture estimation of the target location that is composed of three separate Gaussians, shown here after propagation. Rectangular gates are constructed around the mixture components to discard measurement associations that are extremely unlikely. Then, a higher accuracy elliptical routine is used to create the actual measurement associations from the remaining measurements. Several notable occurrences are displayed here: measurements will fall into the rectangular but not elliptical gates, measurements will be associated with multiple prior hypotheses, and gate regions can be completely disjoint. Hollow circles indicate unassociated measurements, while solid points indicate measurements associated with one or more hypotheses.

Bayes rule:

$$\begin{aligned}P\{\boldsymbol{\Psi}_{u'}(k)|\boldsymbol{Z}^k\} &= P\{\psi_i(k), \boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^k\} = P\{\psi_i(k), \boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}, \boldsymbol{Z}_k, N_z(k)\} \\ &= \frac{P\{\boldsymbol{Z}_k|\psi_i(k), \boldsymbol{\Psi}_u(k-1), \boldsymbol{Z}^{k-1}, N_z(k)\}P\{\psi_i(k)|N_z(k)\}P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\}}{P\{\boldsymbol{Z}_k|\boldsymbol{Z}^{k-1}, N_z(k)\}}\end{aligned} \tag{2.34}$$

This implies two implementation equations, one for updates on measurements falling within a measurement gate:

$$P\{\psi_i(k), \boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^k\} = P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\}f\{\boldsymbol{x}(k)|\boldsymbol{z}_i(k), \boldsymbol{\Psi}_u(k-1)\} \tag{2.35}$$

and another for a missed association hypothesis indicated by the subscript "0" in the following:

$$P\{\psi_0(k), \boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^k\} = P\{\boldsymbol{\Psi}_u(k-1)|\boldsymbol{Z}^{k-1}\}(1 - P_d P_g)\lambda \tag{2.36}$$

where P_d is the probability of the sensor detecting the target (always unity for these simulations), P_g is the probability that the target will be within a gate, and λ is the density of false alarms, or clutter. In Equation (2.35), the propagated hypothesis Gaussian is evaluated at the incoming measurement to generate $f\{\boldsymbol{x}(k)|\boldsymbol{z}_i(k), \boldsymbol{\Psi}_u(k-1)\}$.

2.5.4 Mixture Reduction Algorithms. In a multiple hypothesis scheme in which each prior measurement creates a new hypothesis for each observation, the growth in the number of association hypotheses will be:

$$N_h(k) = N_r(k-1)(N_{Z(k)} + 1) \tag{2.37}$$

where N_r is the number of hypotheses carried into an update cycle (also, the number remaining at the end of the previous reduction cycle) and $N_{Z(k)} + 1$ is the number of new associations (one for each measurement within a gate and an additional association for the case of no detection). The number of hypotheses will quickly become incalculably large and will need to be managed by a mixture reduction algorithm. The goal of any hypothesis reduction algorithm is to reduce the explosion of association history hypotheses while still maintaining enough target information to sustain a track.

In order to approximate the target PDF, one could simply discard, or prune, those hypotheses that have negligible contributions. Several different methodologies [5] include choosing a threshold, ϵ, and discarding hypotheses that have a mixture weight beneath the threshold. Alternately, one could retain only the n best hypotheses (as determined by their mixture weights). Finally, one could discard the lowest weighted components until a certain amount of the total mixture probability has been reduced. These methods work well for low clutter cases and are extremely fast in relation to more complex routines, but do not offer good performance in higher clutter cases [36]. Another method, and one fairly popular in current MHT implementations, is the N-scan memory filter, which combines hypotheses that share a common (defined on the N most recent epochs) association history [29]. If only the current epoch is used, the so-called zero-scan filter will be equivalent to the Probabilistic Data Association (PDA) algorithm [1]. As the length of the memory history increases, the number of maintained hypotheses increases exponentially, as does the computational loading.

Salmond [25–27] proposed two algorithms for mixture reduction. His clustering algorithm starts by finding the component with the highest mixture probability and merges it with components that fall within a certain radius of the cluster center. The square of the clustering distance is defined as:

$$D_{ij}^2 = \frac{p_i p_c}{p_i + p_c} (\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_c)^T \mathbf{P}_c^{-1} (\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_c) \tag{2.38}$$

where the c subscript indicates the cluster center component. As it stands, the algorithm will exit with an arbitrary number of components depending on the distribution of the entering components. Salmond desired a fixed number of components to exit the reduction cycle, so he put the clustering algorithm inside a loop that would iterate while gradually increasing the clustering region size until the appropriate mixture size has been achieved.

The clustering algorithm displays better performance than the previous algorithm, while still operating at a relatively high computational efficiency.

Salmond also generated a cost-function-based mixture reduction algorithm that worked by joining pairs of hypotheses in a mixture that were the "closest" to each other as defined by:

$$d_{ij}^2 = \frac{p_i p_j}{p_i + p_j}(\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j)^T \mathbf{P}^{-1}(\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j) \tag{2.39}$$

where $\mathbf{P}$ is the overall mixture covariance. Originally, Salmond declared a threshold for this distance, $T = 0.001n$ where n is the dimension of the state vector, and would join components until the minimum cost was above this threshold. Notably, this uses an iterative Greedy [33] algorithm to find a traversal of the decision space based on the cost metric.

2.6 Cost Functions

The following sections describe cost functions, which are methods of determining the difference of two functions. These methods can be applied to any functions, but the underlying assumption in this discussion is that we will be comparing two mixtures of Gaussians; specifically, an original mixture is compared to a reduced version (one made of fewer components). The cost function will give an effective measure of the amount of distortion caused by the approximation induced by reducing the number of components in the mixture.

2.6.1 Kolmogorov Variational Distance. Consider the two PDFs shown in Fig 2.8. In this case, a Gaussian mixture of two components is approximated by a single Gaussian. To determine the quality of the representation, one may simply subtract the area represented by the reduced mixture from the area represented by the original mixture. The result, labelled as "raw" and shown above the mixtures, displays the direct difference in mass between the two. For computation, one desires a simple metric, such as a scalar number, that can be easily compared for decision making. This can be accomplished by integrating the difference shown above. Unfortunately, the reduced representation has both positive and negative differences from the original, meaning that a simple integration

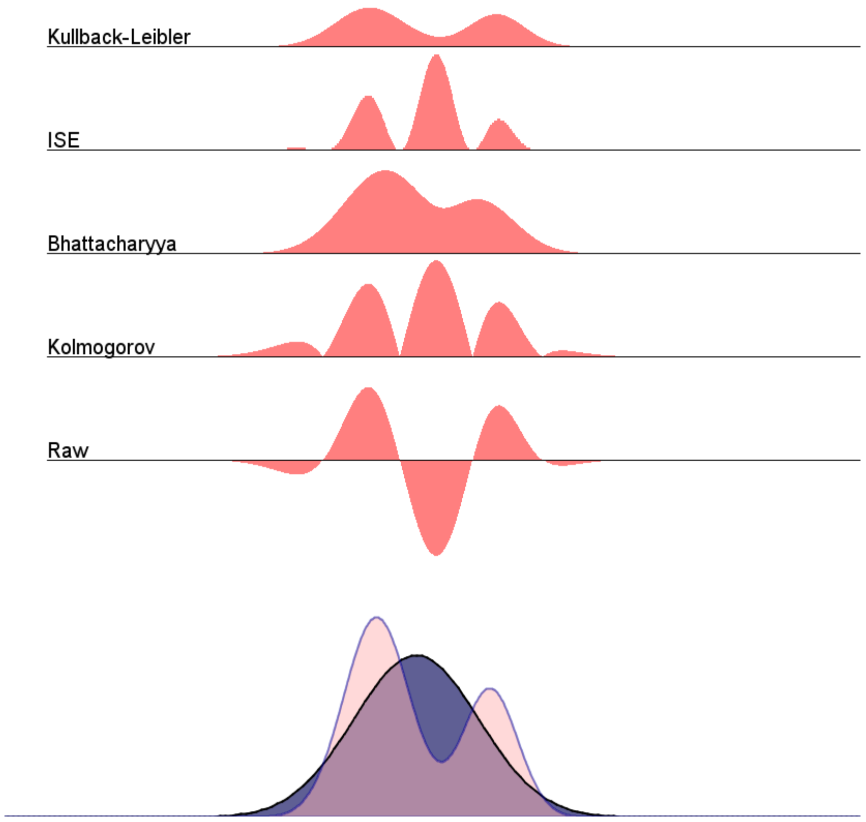


Figure 2.8: Various Cost Functions. Two arbitrary functions are displayed at the bottom of the figure. The difference in their mass as calculated by various cost functions is displayed directly above them.

would allow errors to cancel, an undesirable effect. The obvious solution would be to take the absolute value of the difference before integration, yielding the error representation:

$$J_K = \int |f(x) - g(x)| dx \tag{2.40}$$

known as the Kolmogorov variational distance. Here, the cost will increase as the reduced representation becomes a poorer fit. When the two mixtures are completely disjoint, the cost will be the sum of the mass underneath each of the components (2 in this case of both functions being PDFs; using a multiplicative factor of $\frac{1}{2}$ on (2.40) would equate this to unity). Unfortunately, this equation cannot be resolved in closed form. For the example here, it would be possible to do a piecewise integration, but this becomes extraordinarily burdensome when dealing with the multidimensional Gaussians necessary for target tracking applications.

2.6.2 Bhattacharyya Distance. Since the components of the Gaussian mixture cannot be negative, they will only contribute positive mass upon integration. Therefore, multiplying the functions will still yield a positive number that can be integrated to yield a meaningful distance measure of the form:

$$J_{B^2} = \int f(x)g(x) dx \tag{2.41}$$

This measure would be unbounded, but taking the square root prior to integration would bound the function at unity for similar components, decreasing to zero as the functions become disjoint. This distance measure is known as the Bhattacharyya distance (or Hellinger Affinity Measure) [9] and was used by Lainiotis and Park for Gaussian mixture reduction [13]:

$$J_B = \int \sqrt{f(x)g(x)} dx \tag{2.42}$$

Figure 2.9 shows two PDFs (similar to those of Figure 2.8) represented as several arbitrarily large partitions. The cost differences for each partition as evaluated by the Kolmogorov and Bhattacharyya cost functions are shown directly above the two PDFs. The difference (or error mass) of two partitions is highlighted in blue. Note that the amount of probability in

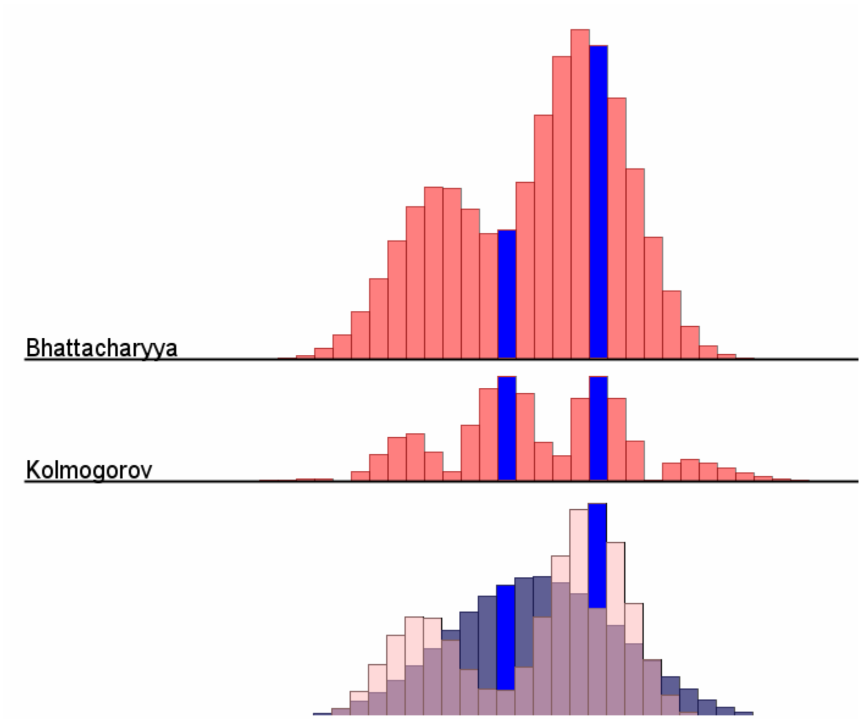


Figure 2.9: Preference of Bhattacharyya to Maintain High Probability.

these differences is the same, as shown by the equal cost assigned to them in the Kolmogorov evaluation, but the cost assigned to them in the Bhattacharyya function is different. In this case, the Bhattacharyya function assigns a higher cost to the difference dealing with higher magnitude probabilities. Given that the Bhattacharyya function is an affinity measure, this implies that the function favors regions of higher probability. In contrast, the Kolmogorov evaluation has no preference for higher or lower probability values, measuring only the absolute difference.

Figures 2.10 and 2.11 show another way of visualizing the cost accrued by the change in probability in a single partition of a PDF. Specifically, if the cost function was evaluated through direct numerical integration by dividing into partitions and evaluating each one individually, the figures show how much cost would be accrued by a change in probability in any given partition. In the left plot, the X and Y axes represent original and reduced probability values of a partition, and the Z axis shows the cost assigned to the corresponding probability change (the right plot is a contour version of the left plot). In these figures, the diagonal line (X=Y) represents no change in probability (the original and reduced partition probabilities are the same). The Kolmogorov function has no cost on the diagonal, with the cost increasing linearly as one moves perpendicularly. Note that the Bhattacharyya function is tilted along the diagonal so that high probability components have a higher cost value, even if there is no change in probability for the partition.

2.6.3 Integral Square Error. Instead of using an absolute value to rectify the negative error costs as in the Kolmogorov evaluation, one can square the values (or use any even integer power), yielding the Integral Square Error metric:

$$J_{ISE} = \int (f(x) - g(x))^2 dx \tag{2.43}$$

Figure 2.12 shows the distance characteristics for the squared error function. Note that the distance increases quadratically as the difference in probability increases, but is not dependent upon the magnitude of probability in the partition.

The major advantage of this distance measure is that it can be evaluated in closed form. Following the derivation of Williams [35], the binomial expansion of Equation (2.43) yields

$$J_{ISE} = \int (f(x)^2 - 2f(x)g(x) + g(x)^2) dx \tag{2.44}$$

This section has described costs in terms of an original function and a comparison function. We are interested in applying this cost function to mixtures of Gaussians, so consider the original function to be a Gaussian mixture prior to a reduction cycle:

$$f\{\boldsymbol{X}(k)|\boldsymbol{\Omega}_{N_h}(k)\} = \sum_{i=1}^{N_h(k)} p_i \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_i, \mathbf{P}_i\} \tag{2.45}$$

and the comparison function will be the same mixture after a reduction cycle:

$$f\{\boldsymbol{X}(k)|\bar{\boldsymbol{\Omega}}_{N_r}(k)\} = \sum_{i=1}^{N_r(k)} \bar{p}_i \mathcal{N}\{\boldsymbol{x}; \bar{\boldsymbol{\mu}}_i, \bar{\mathbf{P}}_i\} \tag{2.46}$$

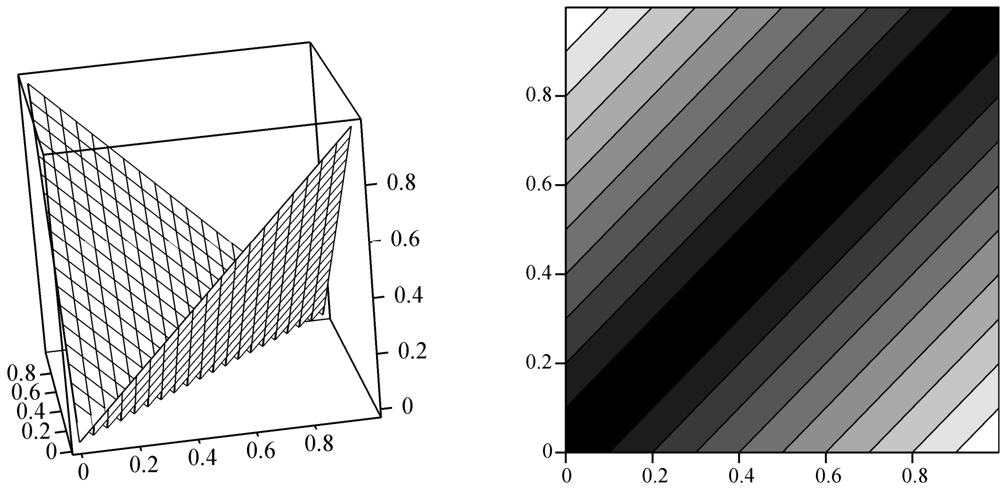


Figure 2.10: Kolmogorov Distance for the Probability Change in an Identical Partition of Two Functions

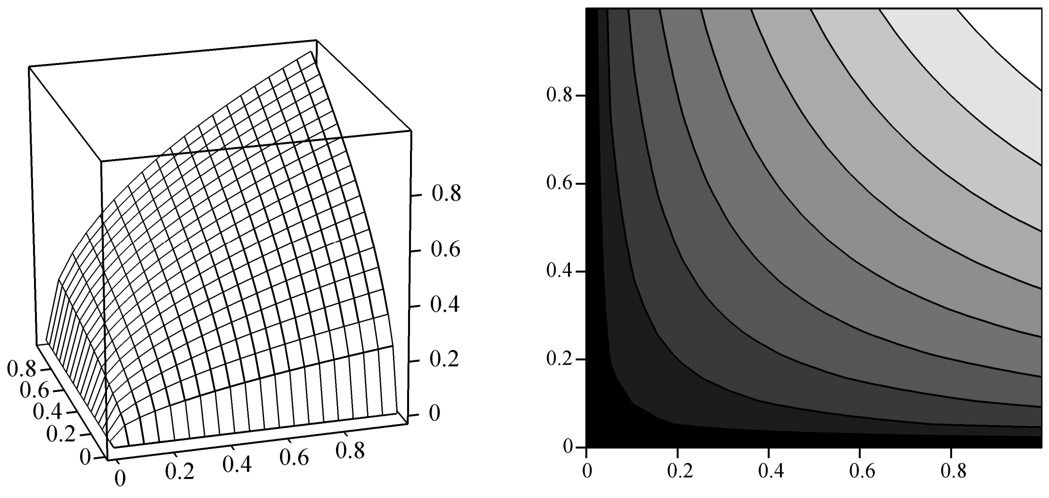


Figure 2.11: Bhattacharyya Distance for the Probability Change in an Identical Partition of Two Functions

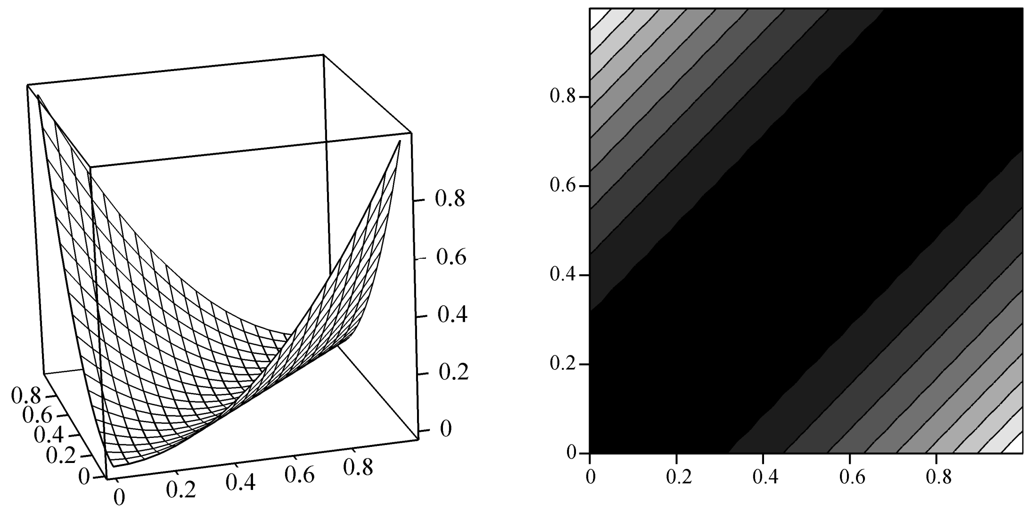


Figure 2.12: Squared Error Distance for the Probability Change in an Identical Partition of Two Functions

Substituting these terms into Equation (2.44) yields:

$$\begin{aligned} J_{ISE} &= \int \sum_{i=1}^{N_h(k)} \sum_{j=1}^{N_h(k)} p_i \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_i, \mathbf{P}_i\} p_j \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_j, \mathbf{P}_j\} \mathrm{d}\boldsymbol{X}(k) \\ &- 2\int \sum_{i=1}^{N_h(k)} \sum_{j=1}^{N_r(k)} p_i \mathcal{N}\{\boldsymbol{x}; \boldsymbol{\mu}_i, \mathbf{P}_i\} \bar{p}_j \mathcal{N}\{\boldsymbol{x}; \bar{\boldsymbol{\mu}}_j, \bar{\mathbf{P}}_j\} \mathrm{d}\boldsymbol{X}(k) \\ &+ \int \sum_{i=1}^{N_r(k)} \sum_{j=1}^{N_r(k)} \bar{p}_i \mathcal{N}\{\boldsymbol{x}; \bar{\boldsymbol{\mu}}_i, \bar{\mathbf{P}}_i\} \bar{p}_j \mathcal{N}\{\boldsymbol{x}; \bar{\boldsymbol{\mu}}_j, \bar{\mathbf{P}}_j\} \mathrm{d}\boldsymbol{X}(k) \end{aligned} \quad (2.47)$$

which is equivalent to (see Williams [35] for a full treatment):

$$\begin{aligned} J_{ISE} &= \sum_{i=1}^{N_h(k)} \sum_{j=1}^{N_h(k)} p_i p_j \mathcal{N}\{\boldsymbol{\mu}_i; \boldsymbol{\mu}_j, \mathbf{P}_i + \mathbf{P}_j\} \\ &- 2\sum_{i=1}^{N_h(k)} \sum_{j=1}^{N_r(k)} p_i \bar{p}_j \mathcal{N}\{\boldsymbol{\mu}_i; \bar{\boldsymbol{\mu}}_j, \mathbf{P}_i + \bar{\mathbf{P}}_j\} \\ &+ \sum_{i=1}^{N_r(k)} \sum_{j=1}^{N_r(k)} \bar{p}_i \bar{p}_j \mathcal{N}\{\bar{\boldsymbol{\mu}}_i; \bar{\boldsymbol{\mu}}_j, \bar{\mathbf{P}}_i + \bar{\mathbf{P}}_j\} \end{aligned} \quad (2.48)$$

The basic building blocks of Equation (2.48) are the multivariate Gaussian evaluation:

$$\begin{aligned} & p_1 p_2 \mathcal{N}\{\boldsymbol{\mu}_1\ \boldsymbol{\mu}_2, \mathbf{P}_1 + \mathbf{P}_2\} \\ & \quad = \quad p_1 p_2 |2\pi(\mathbf{P}_1 + \mathbf{P}_2)|^{-\frac{1}{2}} \exp\{-\tfrac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T (\mathbf{P}_1 + \mathbf{P}_2)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\} \end{aligned} \tag{2.49}$$

2.6.4 Kullback-Leibler Distance. While Williams chose the ISE cost function for use in his research due to the fact that it could be evaluated in closed form, he recommended [35, 37] research into a maximum likelihood function:

$$J_{ML} = \int (f(x) \log g(x) dx \tag{2.50}$$

based on the Kullback-Leibler divergence:

$$J_{KL} = \int f(x) \log \left(\frac{f(x)}{g(x)} \right) dx \tag{2.51}$$

The interpretation of the Kullback-Leibler divergence, the distance between a true PDF and an arbitrary approximation [12], is a natural fit to the posed problem, comparing a reduced mixture to its original, but this distance is, unfortunately, extremely difficult to evaluate due to the logarithm.

2.6.5 Merging and Pruning. The cost evaluation listed in Section 2.6.3 merely gives a measure of difference between two mixtures. In order to use it to reduce a mixture, candidate mixtures must be generated and compared. To generate candidate mixtures, Williams chose to examine the costs related to a sequence of merging and pruning available Gaussian components using the Greedy assignment solution outlined in Algorithm 1.

The most common method of merging Gaussians is by:

$$\text{Mean}: \quad \hat{\boldsymbol{x}}_c = \frac{1}{p_1 + p_2} \{p_1 \hat{\mathbf{x}}_1 + p_2 \hat{\mathbf{x}}_2\} \tag{2.52}$$

$$\text{Covariance}: \quad \mathbf{P}_c = \frac{1}{p_1 + p_2} \left\{ p_1 \mathbf{P}_1 + p_2 \mathbf{P}_2 + \frac{p_1 p_2}{p_1 + p_2} [\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2][\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2]^{\mathrm{T}} \right\} \tag{2.53}$$

Algorithm 1 ISE Algorithm

```
1: function ISEReduce($GaussianMixture, MaxComponents$)
2:     while $ComponentCount > MaxComponents$ do
3:         for all $Components$ do
4:             EvaluateCostofPruning($Component$)
5:         end for
6:         for all $Components$ do
7:             for all $Components$ do
8:                 EvaluateCostofMerging($Component, Component$)
9:             end for
10:        end for
11:        Execute Lowest Cost
12:    end while
13: end function
```

which maintains the mean and covariance of the components. For the cost functions listed previously, these equations will minimize the cost as defined by the Kullback-Leibler (a maximum likelihood) distance, but will not produce the minimum cost representation using the other criteria.

The Greedy algorithm provides a structured method to find the cheapest action at each step, but does not guarantee the optimal traversal will be found. From Figure 2.8, the raw probability mass contribution of a Gaussian can be positive or negative. Thus, one can see that the cost incurred by one merge may be cancelled by a merge at a later step. The Greedy algorithm does not take into account these possibilities.

While an algorithm that predicts several steps into the future to find the cheapest *series* of merges at each time step may be intriguing, the benefits are dubious. The number of possible paths would increase combinatorially, such that a prediction tree with even a minimal number of branches would be entirely intractable. Also, the original intent of Williams' research was to use the Greedy solution as an initialization point and use an iterative solver to modify the mixture to minimize cost further. In the end, his results with the iterated solver were less favorable than with just the Greedy initialization. Williams' reason for the unexpected results was that the ISE cost function is not the best cost metric for the tracking problem. He recommended a maximum likelihood measure based on the Kullback-Leibler divergence as the most meaningful cost function. In fact, the merging operation listed in Equation (2.52) does not create a minimal cost representation for the

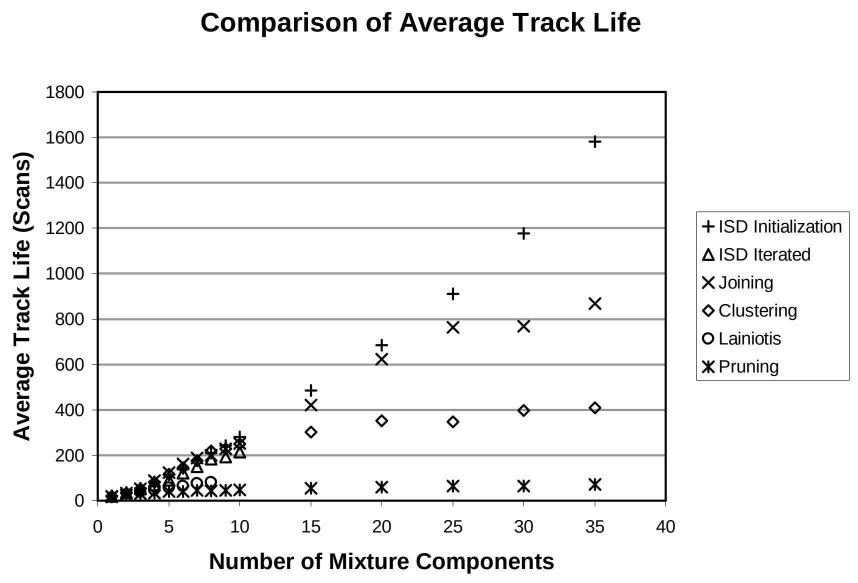


Figure 2.13: Average Track Life of Various Merging and Pruning Algorithms

ISE algorithm, but does minimize the Kullback-Leibler Divergence. Williams concluded that the Greedy ISE performed well because it was a hybrid implementation – it used ISE decisions based on maximum likelihood actions. So, look-ahead schemes that attempt to improve upon the Greedy traversal are not likely to improve performance in a tracking environment.

Figure 2.13 shows the results of a test by Williams comparing the performance of the ISE[1] MRA algorithm against other existing methodologies [35]. The results indicate that the average track life in clutter continues to increase with the number of available components in contrast to the other algorithms, which tend to have performance ceilings on the number of usable components. Additional testing with regard to other criteria [37] maintained that the ISE algorithm provides the best performance compared with any available mixture reduction algorithm.

2.6.6 ISE Complexity. The computational complexity of the algorithm is obvious. Evaluation of Equation (2.48) will yield only the cost for merging a pair of components or pruning a single component. By Algorithm 1, this evaluation will have to be carried

[1]The names Integral Square Error and Integral Square Difference are used interchangeably, because the error is defined as the difference between two functions.

out for every combination of available components and again for each possible deletion. Williams suggests several methods for increasing the computational efficiency of the algorithm without sacrificing accuracy. In [35], he suggests using tables to cache calculated costs and modifying them only when needed. In addition, he proposed a UD Factorization scheme to accelerate the computation of the inverse term in Equation (2.49). In [37] he suggests prepending a modified N-scan filter to the MRA to discard low probability ($p_i < 1\text{x}10^{-6}$) components. Finally, he suggests [37] a scheme that uses cached Gaussian comparison values as lower bounds, so that evaluations are only performed when necessary. In the simulations performed here, implementation of these routines reduced computational times for the algorithm from several hours to a few minutes without impacting accuracy, so the fully accelerated algorithm was used for all trials.

2.7 Multiple Model MHT

The potential for performance improvement offered by a synthesis of the multiple model techniques and multiple hypothesis trackers has fostered considerable amounts of research in the area, although, the many different forms of the MHT with regard to likelihood maintenance and hypothesis control make each implementation unique and difficult to compare.

Dempster, Blackman, and Nichols [7] used an IMM in combination with a track-oriented MHT, which maintains likelihood scores for candidate tracks and forms hypotheses from compatible tracks (those tracks that do not share measurement associations) with likelihood values above a certain threshold. In their study, each hypothesis was propagated and updated using an IMM, but gating and data association were performed using the combined estimate. Their MHT controlled hypotheses and track branches using a pruning technique based on likelihood scores.

Kirubarajan and Bar-Shalom [10] proposed the IMMPDAF, which uses the Probabilistic Data Association (PDA) filter to incorporate multiple measurements probabilistically into a single track. The single track is then propagated and updated using an IMM algorithm. The IMMPDAF is not necessarily an MHT in that it does not maintain multiple hypotheses across scans, rather it forms multiple hypotheses during its update cycle and immediately reduces them to a single component.

Torelli, Graziano, and Farina proposed the I3MHT algorithm [34], which combines a pruning-based MHT with an IMM filter. The IM3HT algorithm maintains a multifold target tree, meaning that each new measurement will spawn a new IMM process from each existing track. Using a variety of turn dynamics from 0 to 5g, they found that the IM3HT algorithm offered improvement over an MHT (equipped with a maneuver detector), with the exception of non-maneuvering modes (0 and .2g) in which the performance suffered from the presence of higher acceleration states with non-zero probability.

Recently, Smith [30, 31] implemented a multiple model tracker that replaced the elemental Kalman filters in an MMAE or IMM architecture with elemental ISE-based MHTs. Smith introduced the concept of pseudo-states, pseudo-covariances, and pseudo-residuals to be used in the IMM mixing process and in forming the blended estimate for both the MMAE an IMM. The pseudo-means and covariances are the mixture best estimates (Equations (2.30) and (2.31)), while the pseudo-residuals are calculated as the mixture-probability-weighted averages of the component residuals:

$$\hat{\boldsymbol{r}}_m(k) = \sum_{u=1}^{N_h(k)} \hat{\boldsymbol{r}}_{m,u}(k) P\{\boldsymbol{\Psi}_u(k)|\boldsymbol{Z}^k\} \tag{2.54}$$

with covariance:

$$\mathbf{S}_m(k) = \sum_{u=1}^{N_h(k)} \left\{\mathbf{S}_{m,u}(k) + [\hat{\boldsymbol{r}}_{m,u}(k) - \hat{\boldsymbol{r}}_m(k)][\hat{\boldsymbol{r}}_{m,u}(k) - \hat{\boldsymbol{r}}_m(k)]^{\mathrm{T}}\right\} P\{\boldsymbol{\Psi}_u(k)|\boldsymbol{Z}^k\} \tag{2.55}$$

where $P\{\boldsymbol{\Psi}_u(k)|\boldsymbol{Z}^k$ corresponds to the individual mixture component probabilities p_i. These two terms are required for the $[\boldsymbol{r}_m^{\mathrm{T}}(k)\ \mathbf{S}_m^{-1}(k)\ \boldsymbol{r}_m(k)]$ quadratic term necessary to calculate modal probabilities within the MMAE or IMM structure. Note that the summation is performed over $N_h(k)$ components, reiterating that the pseudo-residuals are formed prior to mixture reduction.

2.8 Summary

This chapter presented the necessary methods of maintaining the state of maneuvering target in clutter given only noise-corrupted position measurements at discrete intervals in time. Assuming that the target moves in a kinematic fashion, the position can be predicted using one of several dynamics models (Section 2.3.1). Uncertainty in the measurements can be removed using a recursive Kalman filter (Section 2.3), a number of which can be run in parallel in a multiple model architecture, if necessary, to account for uncertain target dynamics (Section 2.4). In the presence of clutter, the target PDF can be propagated and updated as a mixture of Gaussians (Section 2.5), the size of which should be maintained at a desirable level by a mixture reduction algorithm (Section 2.5.4). Currently, the best available reduction algorithm is the Integral Square Error metric (Section 2.6.3).

III. Simulation Development and Analysis

3.1 Introduction

Chapter 2 outlined the theory defining the state-of-the-art in tracking maneuvering targets in clutter. This chapter outlines the methodology for synthesizing those concepts into a working simulator of a tracking environment. Section 3.3 outlines several truth models that will be used for debugging and data collection, to include realistic aircraft data from a flight simulator. Section 3.4 addresses some issues with the standard clutter generation techniques and proposes an alternate method of generation. Sections 3.6 and 3.7 address the component filters of the tracker and how they will be configured with specific attention to the unique ways they partition the data generated by the tracker. The chapter is concluded with a brief discussion on the performance metrics that will be used for analysis in Chapter 4.

3.2 Program Architecture

To ensure consistency of simulation across all of the varied filter configurations, a monolithic simulation architecture was developed – the same binary executable was used for every simulation. This was accomplished by abstracting filter elements into "black box" modules that can be chained together to create different configurations. Algorithm 2 outlines the fundamental steps necessary to implement the tracker, and each step will be fully developed in this chapter and presented in a similar algorithmic format for ease of conveying concepts.

Algorithm 2 Simulation

```
1: while tracking do
2:     GENERATE TRUTH()                    ▷ see Section 3.3
3:     PROPAGATE()                         ▷ see Section 2.3.2
4:     GENERATE MEASUREMENTS()             ▷ see Section 3.4
5:     UPDATE()                            ▷ see Section 2.3.3
6:     CHECK TRACK LOSS()                  ▷ see Section 3.8
7:     PROBABILITY MARSHALLING()           ▷ see Section 3.9
8: end while
```

3.3 Truth Generation

Three methods were used for trajectory generation: random noise processes, mathematical loop models and output from a flight simulator. The random noise process simulations are performed for continuity and verification of the multiple hypothesis tracker with previous research performed by Williams and Salmond, while the loop models are designed for testing multiple model integration. The flight simulator provides the realistic truth data that will highlight the feasibility of these methods in a practical setting.

3.3.1 Random Noise Process Simulations. Williams [35–37] tested his mixture reduction algorithm using a random noise process truth model. The truth trajectory is generated using the CV truth model from Equation (2.1) (alternately presented here in discrete form):

$$\hat{\boldsymbol{x}}(k) = \begin{bmatrix} \hat{p}(k) \\ \hat{v}_x(k) \\ \hat{p}(k) \\ \hat{v}_y(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}(k-1) + \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ T & 0 \\ 0 & \frac{\Delta t^2}{2} \\ 0 & 1 \end{bmatrix} \boldsymbol{w}_d(k-1) \tag{3.1}$$

where $\boldsymbol{w}_d(k)$ is an independent zero-mean white noise process such that:

$$E\{\boldsymbol{w}_d(k)\boldsymbol{w}_d(k)^T\} = \mathbf{Q}_d = q_d\mathbf{I}$$

The time step, Δt, and the process noise variance, q_d, are set to unity. This configuration provides a directed flight path with random jinks. An example can be seen in *RandomNoiseFlight.avi* and Figure 3.1. White Gaussian noise is being added to the velocity state, which is plotted in the square in the upper right corner of the movie and figure. The injected noise can be seen in the movie as a random walk of the velocity value, shown as a moving black circle on the velocity plot[1].

[1]See Appendix A for a full explanation of the symbols in this movie.

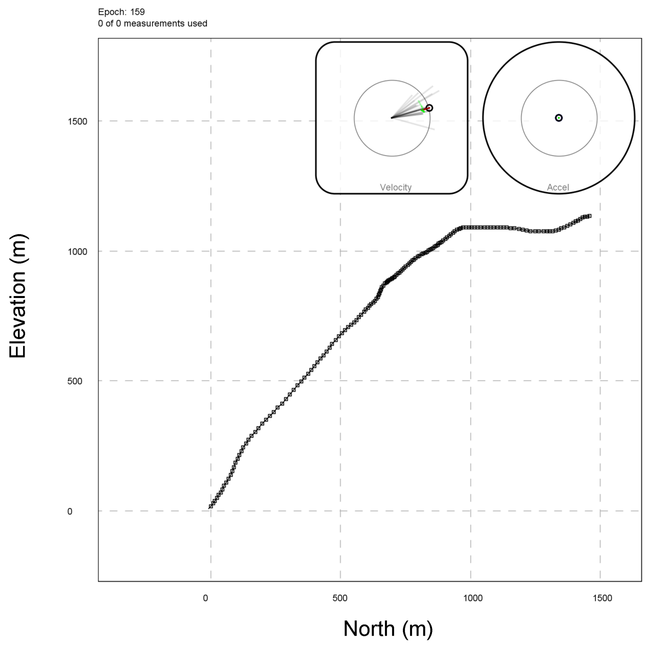


Figure 3.1: Constant Velocity Truth Model Driven by White Gaussian Noise

3.3.2 Loop Models. For testing purposes, a truth model was desired that would produce predictable output in a multiple model based filter. The Φ matrices from the constant-velocity and constant-acceleration truth models were paired with sinusoidal inputs. This is the equivalent of replacing the random noise in Section 3.3.1 with a deterministic input.

$$\boldsymbol{x}(k) = \boldsymbol{\Phi}_{CV}\boldsymbol{x}(k-1) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha(k)\cos(k) \\ \alpha(k)\sin(k) \end{bmatrix} \tag{3.2}$$

or, in the case of a constant acceleration filter:

$$\boldsymbol{x}(k) = \boldsymbol{\Phi}_{CA}\boldsymbol{x}(k-1) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha(k)\cos(k) \\ \alpha(k)\sin(k) \end{bmatrix} \tag{3.3}$$

Over time, the α value is gradually incremented (so the α value over time is a ramp), creating an ever-increasingly g-loaded spiral, the velocity variation of which can be inferred from Figure 3.2.

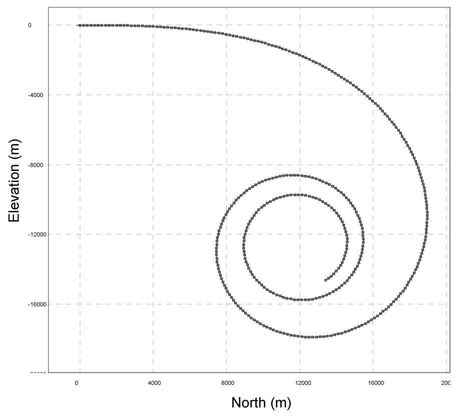


Figure 3.2: Constant Velocity Truth Model with Spiral Input Dynamics

The loop models are designed to display probability flow between elemental filters in a multiple model system. Each filter is tuned to a certain level of process noise, and by increasing the dynamics of the system, the elemental filter will smoothly and predictably flow from filter to filter.

The advantage of the loop model is that it provides an ideal situation for the multiple model structure. Unfortunately, an area where multiple model structures suffer (and thus an area where we would like to demonstrate algorithm performance) is their ability to switch between models when dynamics change abruptly. This effect is most prominent when trying to switch from a filter that assumes aggressive dynamics to a filter assuming benign dynamics. Consequently, an additional truth simulation model was created to address this problem. Specifically, the gradual loop model was modified so that periods of dynamics were interrupted by periods with no dynamic inputs. For all simulations, the input was toggled with a period of 30 epochs. Ideally, probability should transfer in less than 5 epochs, so the 30-epoch period should encompass decisive probability transfer. The magnitude of the dynamics inputs were increased in the same way as before, so this model will present the filter with not only shifts from benign to aggressive, but also abrupt shifts from aggressive to benign and back again. Representative magnitudes of injected dynamics are shown in Figure 3.3. Note that the 50% duty cycle implies that this loop model takes twice as long to reach comparable dynamics levels as the simple loop model.

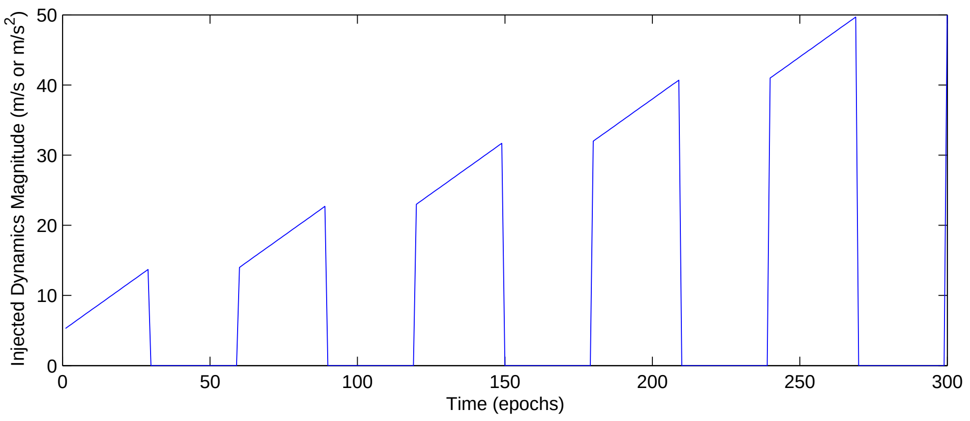


Figure 3.3: Broken Loop Injected Dynamics

3.3.3 FlightGear. The realistic flight data was generated using FlightGear, an open source flight simulator [19]. FlightGear provides a 3D world representation to a pilot, who interacts through a flight stick or yoke. FlightGear also provides a drop-in architecture for several flight dynamics models. The flight data for the simulations was recorded using JSBSim, a 6DOF flight dynamics model that models aerodynamic force and moments using the coefficient buildup method [4]. JSBSim was selected for its ability to offer accurate and realistic flight data – it is thorough enough to model rotational effects on motion, such as the Coriolis effect and centrifugal acceleration.

JSBSim provides several aircraft models spanning all varieties of aircraft. For path generation, General Dynamics F-16 was selected to an aggressive flight model. Roughly

10 minutes of flight data was recorded at 4 epochs per second. Previous research has favored a one second sample period, most likely due to ease of implementation and simpler calculation. For realistic aircraft, a one second sample period is relatively long. The aircraft can maneuver a significant amount in one second, and acceleration states cannot be meaningfully estimated [5]. Additionally, lower sample frequencies create larger measurement gates, which will increase the number of associations per cycle and degrade MHT performance. Consequently, a shorter sample period was selected, allowing higher clutter densities and the ability to estimate acceleration states. A shorter sample period is not unrealistic considering that the simulation loosely models visual trackers and agile radar beams, which tend to have higher sample rates than 1Hz. The simulation was modified so that recorded data could also be undersampled by skipping measurements, allowing flexibility in selecting the sample rate, which has a significant impact on the growth of uncertainty between epochs. In fact, the final time step for the bulk of F-16 simulations was .5 seconds, accomplished by using every other recorded data point.

At the increased sample rate, each run captured roughly 2000 epochs of data. Processing the entirety of this data for every run would have been computationally expensive and would likely not garner more information than testing smaller pieces. Consequently, representative portions were selected from the flight data. Figure 3.4 shows the 175 second time period selected for the F-16 simulations.

JSBSim has the ability to log coordinates in several different reference frames, but does not output all of them in a consistent reference frame. For example, position is written out in Longitude-Latitude-Altitude (LLA), while velocities and acceleration are provided in a North-East-Down (NED) frame. It is possible to convert between frames, but the conversions introduces noise corruption that is noticeable in the final analysis. The noise corruption is most likely due to the limited precision of the ASCII output logs and the difference in magnitude of the measurements. To alleviate this problem, only the velocity data in the NED frame was recorded for each flight, and a Kalman filter based on a simple Newtonian state transition matrix and no measurement noise was used to estimate the position and acceleration states. This provided consistent noise-free values of all truth

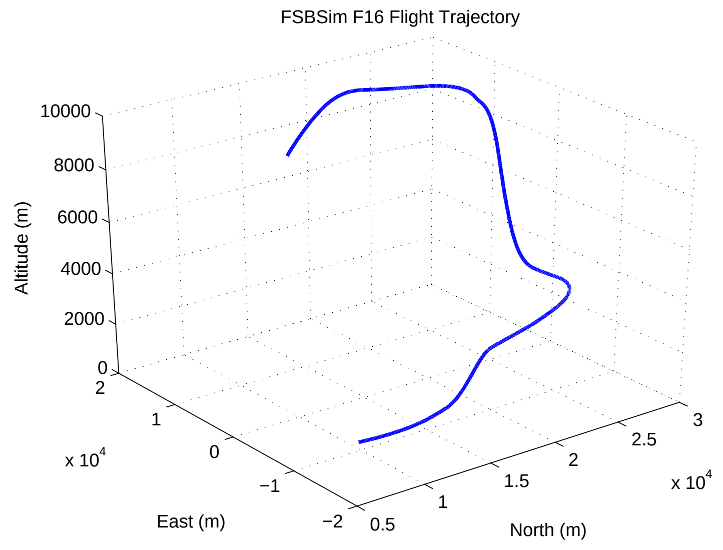


Figure 3.4: F-16 Truth Flight Segment

states. Originally, a Kalman smoother was considered to reconstruct the values, but the single pass filter was sufficient, most likely due to the perfect measurements.

JSBSim logs data in three dimensions, yet the simulator only deals with two dimensions. Consequently, the east data is simply discarded. This effectively creates an orthographic projection, which is a reasonable constraint for the linear filters being employed here and implies that the tracker is a considerable distance away from the target so that nonlinear camera effects such as perspective and angular falloff are negligible. Figure 3.5 shows the F-16 flight path as projected onto the view plane of the observer.

3.4 Measurement Generation

During measurement generation, clutter points are generated within space using a Poisson distribution. For all of the scenarios here, the probability of detection, P_d, is one, so a noise-corrupted truth measurement will always be added to the clutter-originated measurements. For the flight scenarios, the target-originated measurement is corrupted with white Gaussian noise with statistics shown in Equations (2.8) and $\mathbf{R} = 100m^2$. Williams generated measurements inside a variably sized box centered on the truth states [35]. The box was sized large enough that hypotheses could deviate from the truth path

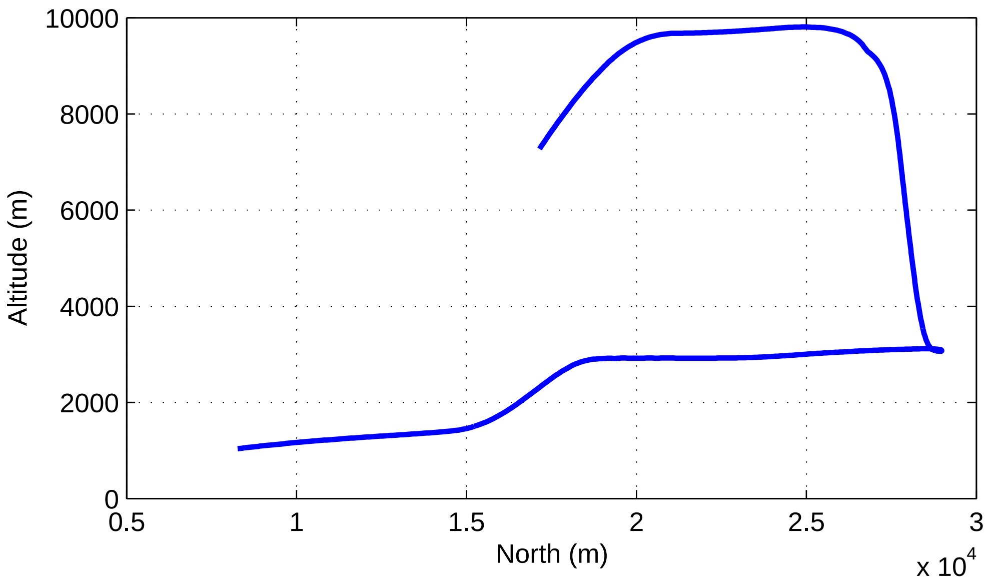


Figure 3.5: F-16 Truth Flight Segment Projected onto the Image Plane of a Tracker

and be deceived by clutter points for several time periods until the randomness of the noise causes the probability associated with an incorrect hypothesis to extinguish. Originally, the box was sized to be $200\sqrt{R}$ units per side. Keeping the same clutter density of $\lambda = 0.012\, pts/m^2$ as Salmond[2], this yields approximately 480 clutter measurements per epoch. Later in his research, Williams discovered that hypotheses could carry probability outside the edge of the box and artificially disintegrate from lack of updates, as shown in Figure 3.6 and movie *EdgeImpact.avi*. To remedy this, he expanded the size of the box to $2000\sqrt{R}$, the end result indicating that edge extinction did aid track life negligibly for ISE and more so for other algorithms such as the Salmond Joining algorithm.

To negate these effects, the measurement clutter for this simulation was generated around the best estimate of the filter predicted forward to the current epoch. The intended effect is that probability deviating from truth would pull the measurement region away from truth much like a diverging agile beam radar. In this case, edge extinction would tend to occur in hypotheses far from the estimate, not in cases far from the truth. Thus, the

[2]Clutter density, as defined by Salmond, is given in measurements per unit hypervolume. For this research, the dimensionality will always be two, and meters are the assumed coordinate system units. Thus, all clutter densities are given as points per meter squared

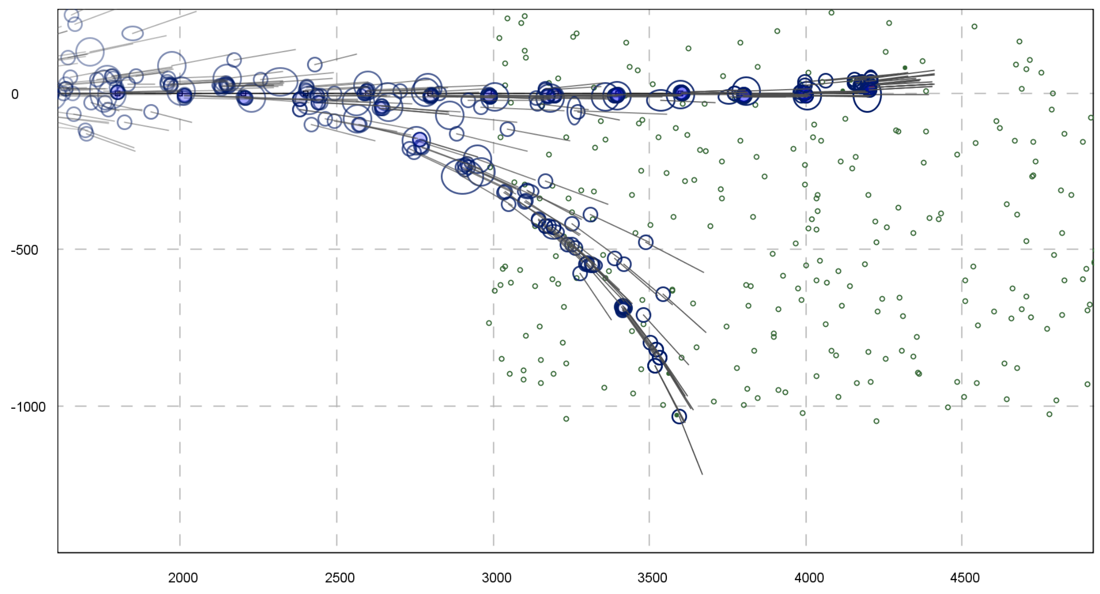


Figure 3.6: Probability Starvation From Insufficient Clutter Region. 13 consecutive epochs of a tracker are displayed. Clutter points are generated in a square region centered on the target, which is traversing the field horizontally from left to right. Note that a series of hypotheses based on clutter have branched downward from the truth trajectory and attempt to propagate outside the clutter region. Without measurements in their association gates, the hypotheses do not update, causing them to lose probability (see Equation (2.36)), and are removed in subsequent mixture reduction cycles. This reduction introduces artificial bias because the clutter field should have infinite dimension so that hypotheses cannot lose probability by entering a region that inappropriately has no clutter.

true track is as likely to be erroneously deleted as a spurious track. In practice, the large measurement error used for these simulations created a clutter region large enough that edge extinction was not noticeable. A performance comparison of the standard, extended, and estimated clutter regions is presented in Section 4.2.

The choice of clutter density greatly affects performance and results. A density of $\lambda = 0.012\, pts/m^2$ creates high enough clutter to demonstrate the remarkable ability of cost-function-based trackers, but the focus of this research is improving RMS tracking accuracy, not track life. Thus, the clutter density was varied until track loss occurred in roughly 10% of the cases, indicating that the algorithm is running near the edge of its performance capability, while still yielding enough complete runs to garner meaningful sample statistics information. In the simulations presented here, the λ value was set at .0001. As target maneuverability increases, the required process noise to track the target also increases, yielding larger gates and more associations per epoch. Consequently, the acceptable clutter values for a more maneuverable target will be smaller than for a less maneuverable target. Additionally, the magnitude of the process noise value will increase with the sample rate, so the largest acceptable clutter density will be inversely proportional to the sample rate.

The number of components remaining in the Gaussian mixture at the end of each mixture reduction cycle was set at 15 for all simulations. In his original work, Salmond suggests 10 components as an appropriate number for good performance [24]. Williams showed that those algorithms did in fact garner little additional performance past 15 components. Also, Williams found that the ISE performance did not surpass that of the Salmond Joining filter until more than 10 components were retained [37]. The simulations here are based primarily on the ISE cost function, which gains improvement exponentially as additional components are added. Obviously, with an apparently uncapped algorithm, some limit needs to be chosen, and 15 was recognized as a good balance of performance and computation time.

It should be noted that, during testing, runs were conducted at a debug setting of 10 components and a clutter density of $\lambda = .00005\, pts/m^2$, which yielded roughly 80% track loss. This provided rapid turnaround time on simulations. In preparation for final simulations, the number of elements was increased to 15 components and a new set point for the clutter density was explored. The algorithm was maintaining 100% track life for all runs even after the clutter density had been increased sixfold ($\lambda = .0003 pts/m^2$). This reiterates that a marginal increase in the number of available hypotheses yields a significant increase in tracking ability. Unfortunately, at this density the number of divergent hypotheses increased to the point that fidelity results were being masked, so the clutter density was reduced to $\lambda = .0001\, pts/m^2$ and the time step was increased to $\Delta t = .5$. Increasing the time step increases the propagation covariance, thereby creating larger gates that will create more measurement associations at a given clutter density. This yielded an acceptable solution with 90% track loss.

3.5 Monte Carlo Simulation

Excepting the random noise process scenarios, all truth methods are deterministic in that each Monte Carlo run differs only in the measurement noise and generated clutter. Accordingly, the Monte Carlo results converge on a solution relatively quickly. Figure 3.7 shows the time-averaged standard deviation of a cluttered case and indicates that the sample statistics computed by the Monte Carlo analysis have settled to an acceptable value (representative of the true underlying statistics) in approximately 20 runs. Since the simulations performed here desire 10% track loss, at least 23 runs are necessary to assure meaningful data. All runs were performed at over double that number (50), so we have complete confidence that the Monte Carlo sample statistics are truly representative of actual statistical properties. Calculation of statistics included data from the divergent runs up until 7 epochs before track loss.

3.6 Filter Modules

The propagate and update methods in Algorithm 2 of Section 3.2 are methods of a base filter, which can be any of three basic filters outlined in this section: the Kalman filter, a multiple model filter, and a multiple hypothesis filter. All filters implement three basic methods, `propagate`, `update`, and `getBestEstimate`. These "black box" implementations are referred to as modules that can be arranged in different configurations to implement

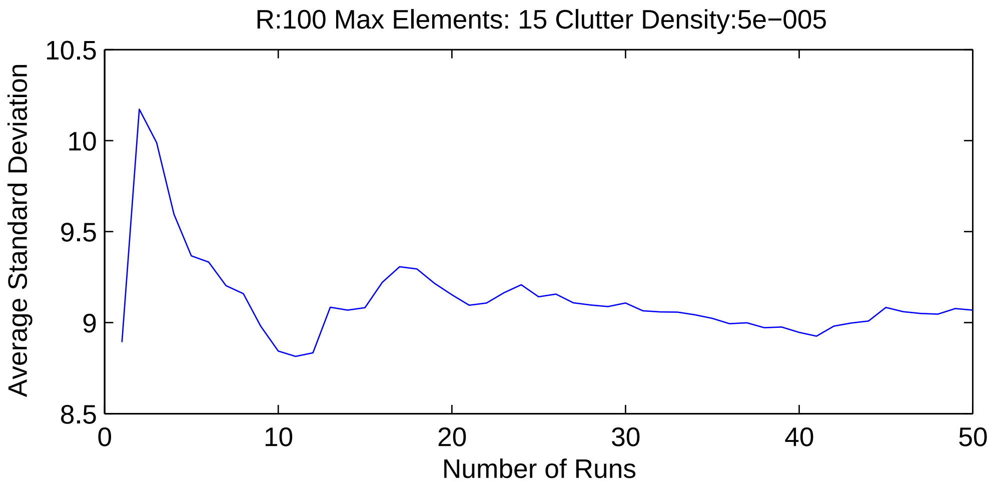


Figure 3.7: Standard Deviation vs Number of Monte Carlo Runs For Cluttered Scenario

the desired filter structures. Additionally, the filters are memoryless, meaning they cannot store data from one epoch to the next – this is an important restriction for the multiple model methods that recursively update elemental filter probabilities. The more advanced multiple model structures here use one filter to update several different hypotheses each cycle, each having its own set of elemental filter probabilities. Consequently, those values must be stored with each hypothesis. Because all of the filters support these requirements, they can be chained together to create different configurations that handle the data structures in specific ways.

3.6.1 Data Containers. For these simulations, the fundamental storage unit is the hypothesis, Ω, which maintains the means, $\hat{\boldsymbol{x}}$, and covariances, $\mathbf{P}$, of a single hypothesis. Additionally, it must maintain two probabilities: p, the probability weight assigned to this component in the Gaussian mixture (for handling clutter), and μ, the multiple model probability computed within and MMAE or IMM structure (for handling varying target dynamics)[3]. The multiple model architecture proposed here will spawn a new multiple model process for each measurement association generated in the update cycle. For com-

[3]The multiple model probability is only necessary when the hypothesis is part of a macrohypothesis structure (to be described more fully in Section 3.7.3), but the values are calculated and stored in the single filter case anyway. The single filter tracker can be viewed as a multiple model filter with only one filter. This abstraction aids in streamlining the code.

putational efficiency, the same instantiation of the multiple model filter in the computer's memory will be used for all updates. Consequently, the multiple model classes must not retain any variables because they are general purpose classes – any variables stored in the multiple model class would be instantly written over by the next measurement association. But, the multiple model concept requires that the modal probability be available for recursion. Thus, the modal probability is stored in the hypothesis to which it applies. This construct modifies the merging equations presented in Section 2.6.5 with the addition of:

$$\mu_{\mathbf{c}} = \frac{1}{p_1 + p_2} \{p_1 \mu_1 + p_2 \mu_2\} \tag{3.4}$$

For computational efficiency, $\mathbf{S}^{-1}$ is also maintained. This value is necessary during the gating equation (recall Equation (2.32)), but is calculated during the propagation cycle. Caching the value reduces the cost of calculating the elliptical gates – in fact, the implementation here did not employ the square gate approximation because the gating computation time was negligible for the given clutter densities.

3.6.2 Simple Kalman. The simple Kalman filter module is the atomic filter for the system. It always represents the end of the filter chain because it contains no subfilters. The simple Kalman filter can act alone in a tracking environment, so it is listed here as both a module (a fundamental "black box" filter) and a configuration (an arrangement of one or more modules). It has no inherent ability to handle clutter and its dynamics are fixed, so it would rarely be used alone. This filter is commonly used as a subfilter to some of the more advanced filters presented later. Figure 3.8 presents a block diagram of the filters and its storage scheme.

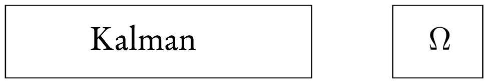


Figure 3.8: Kalman Schema and Data Container

Each Kalman filter is initialized with an assumed dynamics model that yields the $\mathbf{\Phi}$ and $\mathbf{Q}_d$ matrices. As mentioned in Chapter 2, several truth models can be used to predict the dynamics of an aircraft. The constant velocity and the constant acceleration models were used for the bulk of the simulations, although others exist. According to Blackman [5], the constant velocity model performed as well as the more complex Singer model in simulation, despite the Singer model being tuned to the specific aircraft. The constant acceleration model has only one tuning parameter, the discrete-time process noise variance q_d. Blackman [5] recommends using $.5\Delta a_m \leq q_d \leq \Delta a_m$ where Δa_m is the maximum acceleration increment expected over the sampling interval, although true tuning is typically a heuristic process. The constant acceleration model is used in the F-16 simulations for its ease of tuning and by inspection of the recorded acceleration states – the tested flight segment maintained accelerations long enough that the appropriate FOGMA time constant for Equation (2.3) would have been sufficiently large to imply the adequacy of a constant acceleration filter.

Considering that the loop models represent continually increasing dynamics, any filter tuning value will match truth eventually. Consequently, the strength of the process noises were set to match 1g and 3g maneuvers for the benign and aggressive filters, respectively. Having the truth data for the realistic flight models before actual simulation made the filter tuning rather trivial. The acceleration magnitude of the F-16 as viewed by the tracker is shown in Figure 3.9. These values directly represent the process noise required for a constant velocity filter at any given instant since the noise enters at the acceleration level. Given that noise enters the constant acceleration filter at the jerk state, one would tune the constant acceleration filter based on those values instead. From this data, the benign and aggressive CA F-16 filters were tuned to $.5m/s^3$ and $4m/s^3$ jerk levels.

The methods of the simple Kalman filter, presented in Algorithm 3, implement the standard filter equations given in Section 2.5. Additionally, certain variables such as $\mathbf{S}^{-1}$ and the multiple model PDF value, which are necessary for the MHT gating algorithm (see Equation (2.32)) and multiple model probability flow (see Equation (2.18)), are calculated here and cached for later use. For implementation simplicity, these methods are always calculated even though certain filter configurations may not require them: configurations not requiring these values run in fractions of seconds and the additional computation time is negligible.

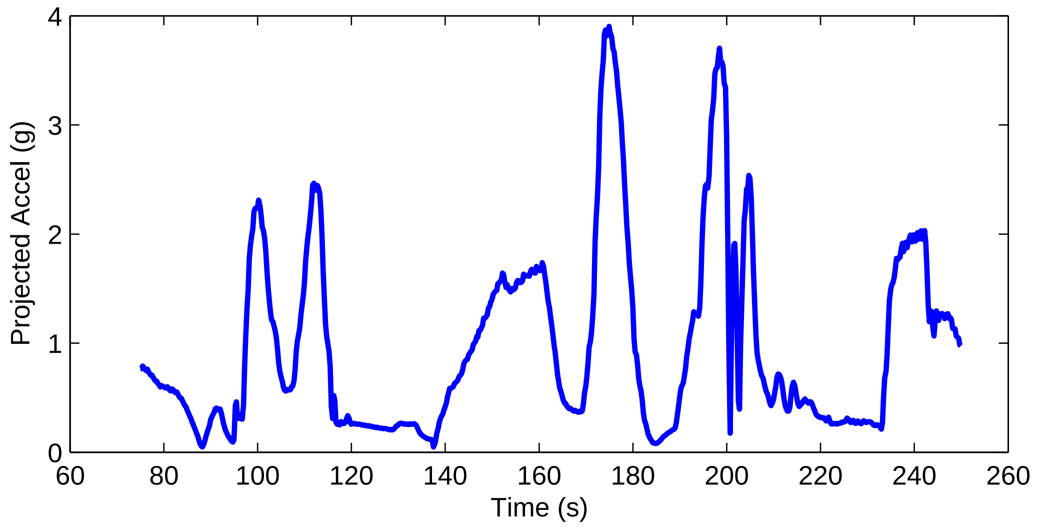


Figure 3.9: F-16 Projected Acceleration

Algorithm 3 Simple Kalman

```
1: function PROPAGATE(mixture)
2:     PROPAGATEMEANS()                                ▷ see Equation (2.5)
3:     PROPAGATECOVARIANCES()                          ▷ see Equation (2.6)
4:     CACHEVARIABLES()
5: end function

6: function UPDATE(mixture)
7:     UPDATEMEANS()                                   ▷ see Equation (2.10)
8:     UPDATECOVARIANCES()
9:     CALCULATEMULTIPLEMODELPROBS()                   ▷ see Section 2.4.1
10: end function
```

3.6.3 Multiple Model. The Multiple Model module is an abstract filter class containing two or more filter models[4] with differing dynamics meant to deal with maneuvering targets. This module is an abstraction and will always be realized as either an IMM or MMAE module.

The primary function of this module is to iterate over every elemental filter and ensure that each element is propagated and updated. The exact method for accomplishing this depends on the form of the data structure and will be fully explained in the pertinent sections.

Algorithm 4 Multiple Model

```
function PROPAGATE(mixture)
    for all $subfilters$ do
        $subfilters \rightarrow$ PROPAGATE( )
    end for
end function

function UPDATE($mixture, measurements$)
    for all $subfilters$ do
        $subfilters \rightarrow$ UPDATE(mixture,measurements)
    end for
end function
```

3.6.3.1 MMAE. The MMAE implementation of the multiple model filter is responsible for normalizing computed probabilities, performing restarts, and lower bounding. Note that the MMAE and IMM implementations have identical propagation steps, which are handled through the multiple model implementation. The MMAE and IMM are subclasses of the Multiple Model class, meaning that they retain all of the Multiple Model functions and augment them with their own.

The setting for the lower bounds will affect the speed of probability flow to effect filter transitions. Experience has shown that .001 provides agile flow without corrupting the blended estimate with excessive data from the mismatched filters.

[4]In fact, the multiple model filter can contain only one filter, but the results will be identical to the simple Kalman model.

Algorithm 5 MMAE

```
 1: function PROPAGATE(mixture)
 2:     super → PROPAGATE( )                 ▷ call the propagate method in Algorithm 4
 3: end function

 4: function UPDATE(mixture, measurements)
 5:     super → UPDATE( )                    ▷ call the update method in Algorithm 4
 6:     NORMALIZEPROBS()                                     ▷ see Section 2.4.2
 7:     LOWERBOUND()                                         ▷ see Section 2.4.2
 8:     RESTARTDIVERGENT(mixture, measurements)              ▷ see Section 2.4.2
 9:     GENERATEBLENDEDESTIMATE()                            ▷ see Equations (2.12)
10: end function
```

Traditionally, restarts are initiated using the measure $[\boldsymbol{r}^T(k)\boldsymbol{A}^{-\mathbf{1}}(k)\boldsymbol{r}(k)]$ quadratic term. Due to the variable caching of this architecture[5], only the multiple model PDF evaluation is available at the restart cycle of the tracker. This is the quadratic term after multiplying by $-\frac{1}{2}$, exponentiation, and applying the leading term (Equation (2.18)), so it still provides meaningful information on filter performance. Additionally, it provides a meaningful interpretation of when filters are diverging because it is an evaluation of a density function at the location of the measurement. For these simulations, the restart threshold was set at .00001.

3.6.3.2 IMM. The IMM implementation is responsible for performing the elemental filter intermixing. The different forms of the Markov matrix provide the methodology for tuning an IMM. Although many different forms exist, this research will use a Markov model probability state transition matrix of the form:

$$T(k|k-1) = \begin{bmatrix} \eta & \gamma & \cdots & \gamma \\ \gamma & \eta & \cdots & \gamma \\ \vdots & \vdots & \ddots & \gamma \\ \gamma & \gamma & \gamma & \eta \end{bmatrix} \tag{3.5}$$

[5]The quadratic term could be used, but it would require storing more data in the hypothesis structure. Due to cache and speed concerns, it is desirable to keep this structure as small as possible.

where $\gamma = \frac{1-\eta}{N_f - 1}$. For these simulations, $\eta = .9$. In the case of a macromixture, a tracker configuration in which the individual elemental filters of the multiple model structure are based on Gaussian mixtures rather than simple Gaussians (outputs of individual Kalman filters), the intermixing process will require the pseudo-states of the mixtures, so they are calculated prior to mixing.

Algorithm 6 IMM

```
1:  function PROPAGATE($mixture$)
2:      $super \rightarrow$ PROPAGATE( )          ▷ call the propagate method in Algorithm 4
3:  end function

4:  function UPDATE($mixture, measurements$)
5:      $super \rightarrow$ UPDATE( )             ▷ call the update method in Algorithm 4
6:      if $macroMixture$ then
7:          CALCULATEPSEUDOSTATES()           ▷ see Equations (2.54) and (2.55)
8:      end if
9:      GENERATEBLENDEDESTIMATE()             ▷ see Equation (2.12)
10:     DOMIXING()                            ▷ see Section 2.4.3.1
11: end function
```

3.6.4 Multiple Hypothesis Module. This filter implements a probability-weighted Gaussian mixture multiple hypothesis tracker. This is the module responsible for gating, creating new association hypothesis, and mixture reduction. This is the only module that has the ability to add or delete hypotheses from the data structure. This module has one submodule, which can be either a simple Kalman filter of Section 3.6.2 or a Multiple Model filter of Section 3.6.3.

The MHT handles associations through the gating method on line 9. This method uses cached variables from the update cycle to implement elliptical gating, as described in Section 3.7.3.1. Williams presented a method for two-step gating that can increase gating efficiency, but only elliptical gating was implemented here[6].

Note that all hypotheses in the mixture are propagated via step 3 of Algorithm 7, but only those hypotheses with gates containing measurements are updated. As updates

[6]Williams implemented the more efficient routine to handle the massive number of clutter points generated by the extended clutter region. As described in Sections 3.4 and 4.2, the extended clutter region was not used, so the more efficient routine was not necessary.

Algorithm 7 Multiple Hypothesis Filter

```
1: function PROPAGATE($mixture$)
2:     for all $mixturecomponents$ do
3:         PROPAGATE()
4:     end for
5: end function

6: function UPDATE($mixture, measurements$)
7:     for all $mixturecomponents$ do
8:         for all $measurements$ do
9:             if INSIDEGATE($component, measurement$) then
10:                $newComponent \leftarrow$ CLONECOMPONENT($component$)
11:                UPDATE($newComponent, measurement$)
12:                CALCUPDATEDPROB($newComponent$)            ▷ see Equation (2.35)
13:            end if
14:        end for
15:        CALCMISSEDPROB($component$)                        ▷ see Equation (2.36)
16:    end for
17:    if $sizeMixture > maxSize$ then
18:        REDUCE($mixture$)                                  ▷ see Algorithm 1
19:    end if
20: end function
```

reduce the covariance of a hypothesis, those hypotheses that are not updated will maintain their assumed dynamics with growing covariances (see Figure 3.15). Upon a successful association, the reference hypothesis is duplicated (note step 10 of Algorithm 7: a duplicate or "clone" is generated), updated and added to the mixture. This will always preserve the original hypothesis as a missed detection. The hypothesis probability is updated either inside the gating loop using Equation (2.35) or outside the loop using the missed hypothesis probability using Equation (2.36).

The last step of the algorithm involves hypothesis reduction. Noting that two nested loops could theoretically create an association for each pair of previous hypotheses and new measurement, the potential for growth is exponential. A simple check is performed to see if the size of the mixture is greater than the maximum number of components, and the mixture is passed to a mixture reduction algorithm if so. Any mixture reduction function can be used, but the ISE cost function was used for all of the simulations presented here.

3.7 Filter Configurations

Having defined the basic modules, the different complex configurations are presented here.

3.7.1 Multiple Model of Simple Kalman. Figure 3.10 represents a conventional multiple model filter in which the elemental filters are Kalman filters. As before, this filter has no ability to deal with clutter, but now has the ability to adjust to changing target dynamics. The results of this filter represent the baseline best performance of a filter configuration in a clutter-free environment.

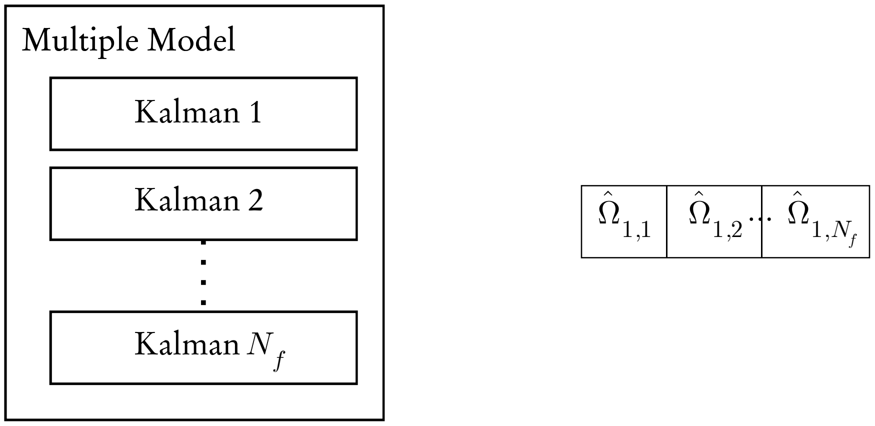


Figure 3.10: Multiple Model Schema and Data Container

The data structure is an array of hypotheses, with each hypothesis corresponding to the assumed dynamics model as defined by the elemental Kalman filter. This structure will be referred to as a *macrohypothesis*. The filter gets a best estimate of its states by performing a blended estimate of the single macrohypothesis using Equation (2.12). This configuration can exist as either an IMM or an MMAE.

Restarts are initiated by analyzing the PDF values stored in each hypothesis. These are the filter-propagated covariances evaluated at the measurements (Equation (2.18)). If the PDF is too low, below 1x10^{-5} in these simulations, the mean and covariance of the divergent hypothesis are set to the blended estimate of the remaining hypotheses. Restarts and lower bounds are only performed by the MMAE version of this filter. The IMM

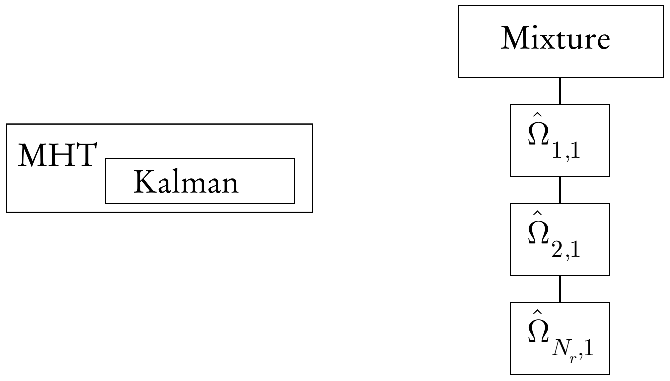


Figure 3.11: MHT Schema and Data Container

incarnation of this filter performs intermixing of the hypotheses as described in Section 2.4.3.1.

3.7.2 MHT of Simple Kalman. Figure 3.11 represents a multiple hypothesis module based on a single Kalman filter dynamics model. In this configuration, there is a single dynamics model that is used to process the hypotheses, each of which represents an assumed association history. This arrangement is capable of dealing with clutter, but not necessarily changing target dynamics. This configuration corresponds to the tests run by Salmond [24] and Williams [35].

In this implementation, the Gaussian mixture is stored as a single linked list of hypotheses. The size of the mixture will be based on the number of associations created in the update cycle; i.e., there is only one type of Kalman filter (based on one dynamics model) but there are N_r separate Kalman filters of that type within the MHT, one for each assumed association history. As this number is unknown until after the update cycle, a variably sized structure that allows rapid inserts is required. Hence, a linked list is used. The best estimate of state for this filter is generated by via Equations (2.30) and (2.31).

3.7.3 MacroHypothesis (MHT of Multiple Model). Figure 3.12 is similar to the previous structure except the multiple hypothesis tracker is using a multiple model based elemental filter instead of a Kalman filter. From Section 3.6.3, a multiple model scheme requires separate states to be maintained for each filter. Here, the data structure is a linked list of macrohypotheses. This configuration is heretofore referred to as a *macrohypothesis-based MHT*. This structure is capable of dealing with both clutter and changing target dynamics. To generate a best estimate of the state of this configuration, the MHT module takes the weighted mean of the macrohypotheses using their blended estimates.

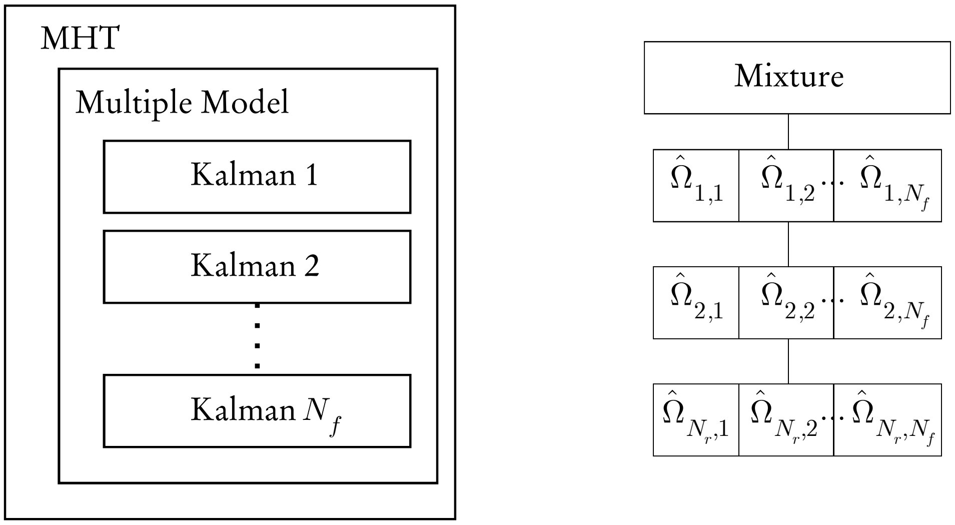

MHT
Multiple Model
Kalman 1
Kalman 2
Kalman N_f
Mixture
$\hat{\Omega}_{1,1}$ $\hat{\Omega}_{1,2}$ ... $\hat{\Omega}_{1,N_f}$
$\hat{\Omega}_{2,1}$ $\hat{\Omega}_{2,2}$ ... $\hat{\Omega}_{2,N_f}$
$\hat{\Omega}_{N_r,1}$ $\hat{\Omega}_{N_r,2}$ ... $\hat{\Omega}_{N_r,N_f}$

Figure 3.12: Macrohypothesis Schema and Data Container

3.7.3.1 Gating. Traditionally, when multiple model structures are used in gating, the gate size is defined by the most dynamic filter [6]. Here, gates are constructed around the individual elements of a macrohypothesis, and each hypothesis is updated if a measurement falls into any of these gates. Therefore, the gating area can be assumed to be a union of the individual gate areas corresponding to different dynamics models[7]. Figure 3.13 shows the effect of restarts on a tracker. The figures show several successive epochs of a target traversing the field from right to left being tracked by a two-filter MMAE. The solid black line represents the true trajectory of the target, the blue ellipses represent the 3σ boundaries of the position state estimates, and velocities are indicated as gray vectors originating at the estimate centers. The squares represent the gates of the MMAEs, where the large square represents a filter assuming aggressive dynamics and

[7]This gate unioning is dissimilar from the unioning performed by Smith, as he was unioning across mixtures, i.e., unioning the gates corresponding to every elemental filter in the MHT structure, each being associated with a specific association history hypothesis.

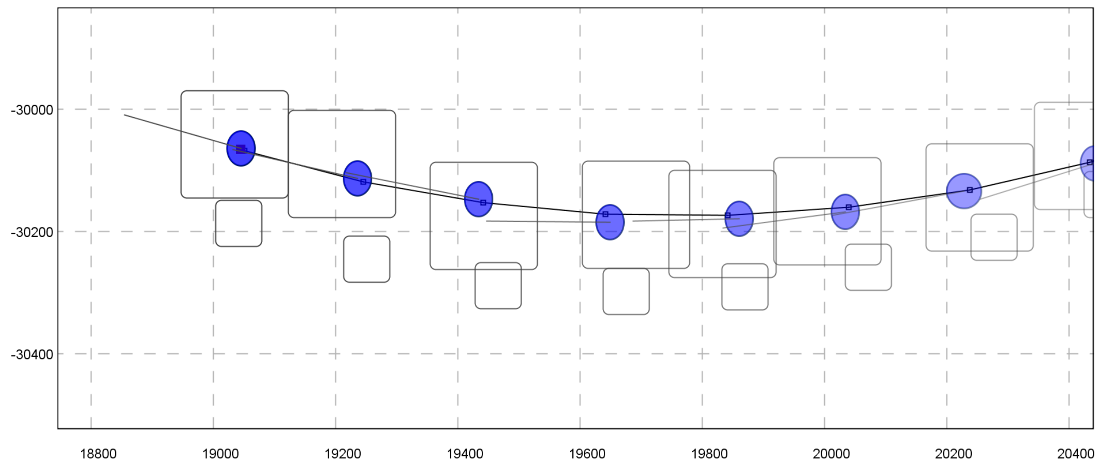
the smaller square represents a filter assuming benign dynamics. Figure 3.13(a) shows a multiple model structure in which the two filter estimates have diverged considerably. In this example, the gates create a disjoint partition space, which would require a per-gate measurement association scheme. As indicated in Figure 3.13(b), a multiple model filter with a well-tuned restart mechanism tends to keep the benign filter within the domain of the aggressive filter. Note the labelled restart. In the previous time epoch, the benign filter was diverging from the aggressive filter and a PDF evaluation based on an incoming measurement triggered a restart. So, even though the measurements are gated using the separate hypothesis gates, if the restart mechanism is working properly, the covered area will effectively be the larger gate, so the more complex gate unioning should not be necessary.

3.7.3.2 Mixture Reduction Algorithm. After the MHT algorithm updates each component in order, it may reduce the mixture. In this case, the mixture is composed of macrohypotheses, and the reduction occurs using a representative mixture composed of the single Gaussian representations of each of these hypotheses. Effectively, the best estimate of each hypothesis is generated and that representation is used in the reduction cycle. A pruning operation will discard the entire macrohypothesis, which will remove N_f hypotheses from the mixture. A merging cycle will merge the hypotheses and their corresponding filter models using augmented merge equations (see Algorithm 8). Note that the cached variable $\mathbf{S}^{-1}$ is not merged because it has already been used in the gating equation at this point and no longer needs to be retained. The restart process for

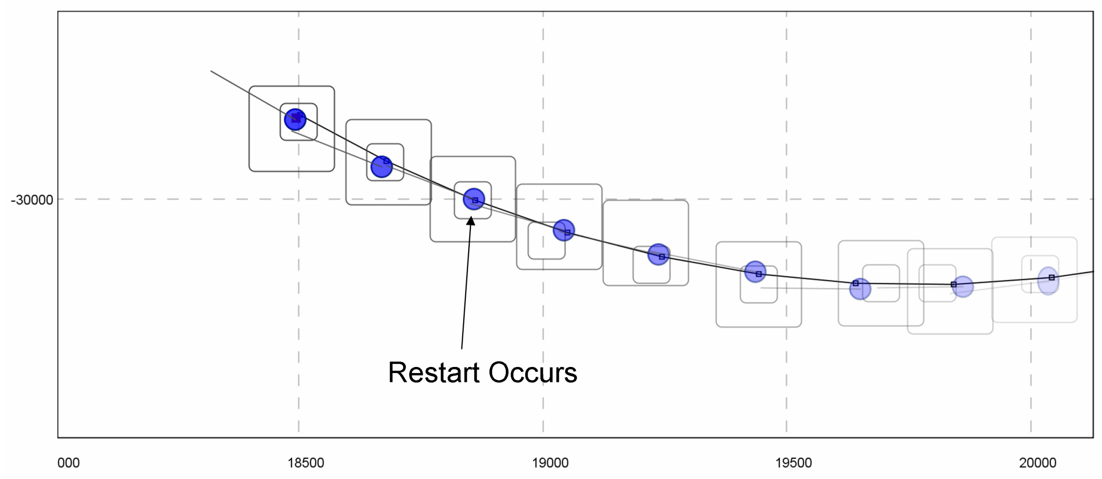
Algorithm 8 Macrohypothesis Merging

```
1: function MERGE($\Omega_i, \Omega_j$)
2:     for $i = 1 : N_f$ do
3:         MERGE($\Omega_{i,i}, \Omega_{j,i}$)        ▷ see Equations (2.52) and (3.4)
4:     end for
5: end function
```

a single macrohypothesis was defined in Section 3.7.1. Considering that the data structure for this configuration is a linked list of macrohypotheses, the restart cycle involves traversing the list and performing the same restart method on each component independently. Remember that the μ values are stored within each macrohypothesis, so they act



(a) Gates Generated in MMAE Without Restarts



(b) Coincident Gates in an MMAE Employing Aa Restart Mechanism

Figure 3.13: Restart Mechanisms in a Multiple Model Filter

as independent multiple model structures. The intermixing process for the IMM version is handled similarly.

3.7.4 Macromixture (Multiple Model of MHT). Figure 3.14 is similar to Figure 3.10 except the Kalman-based elemental filters have been replaced with multiple hypothesis filters; it is denoted as a *macromixture.* This structure corresponds to the design proposed by Smith [30]. Referring to Section 3.7.2, we see that each elemental filter will maintain its own mixture, which will be stored as a linked list. So, this structure maintains a single linked list per filter.

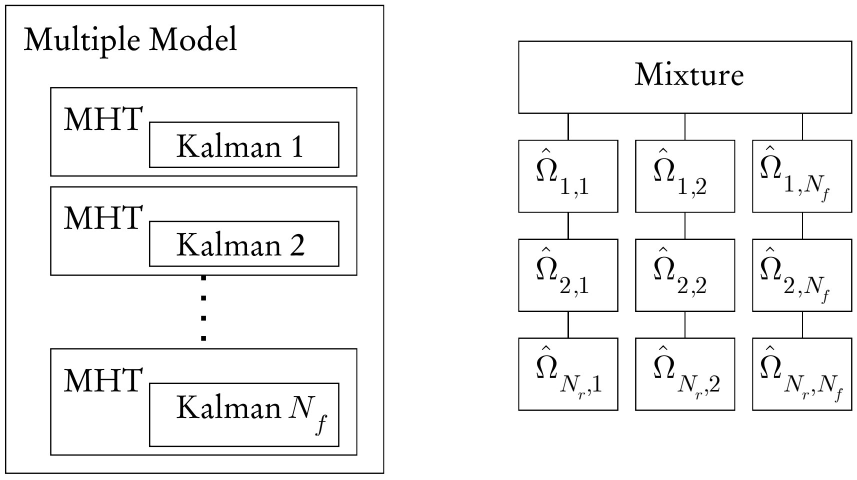


Figure 3.14: Macromixture Schema and Data Container

In the macrohypothesis structure in the previous section, the multiple model probabilities, mu, were calculated for each macrohypothesis as though it were the only existing hypothesis – it had no inherent knowledge of the other components. In this structure though, multiple model probabilities need to be calculated for each mixture based on their pseudo-residuals, the probability-weighted average of the mixture hypotheses' individual residuals (see Equation (2.54)).

3.7.4.1 Best Estimates. To generate a best estimate of the state of this configuration, the weighted mean of each mixture will be generated, forming pseudo-states and pseudo-covariances. These pseudo-states will then be blended using the elemental filter probabilities to form one estimate.

3.7.4.2 Restarts (MMAE only). In his original work, Smith [30] recommended a restart method dubbed "re-centering the cloud," whereby he took an entire divergent mixture and centered it on the best estimate of the remaining non-divergent mixtures [30]. His intent was to maintain a distribution in the mixture so that the mixture would not need to repopulate after a restart. In this implementation, the divergent mixture is discarded and replaced with the best estimate of the non-divergent mixture. It was found that the covariance of the best estimate is typically large enough to encompass enough measurement that the mixture will fully repopulate very quickly. Additionally, the new mixture would now be based on measurements from the assumed area of the target. Therefore, the "re-centering the cloud" restart mechanism was not implemented in this research.

3.7.4.3 Mixing (IMM only). The mixing process for this configuration is fairly complex. The problem lies in the methodology for blending mixtures via the process in Section 2.4.3.1. In this configuration, the multiple model functions have been realized by manipulating the pseudo-states of the mixtures. In the mixing process, the calculation of $\hat{\boldsymbol{x}}^i$ is difficult because it exists only as the best estimate of a Gaussian mixture. So, each Gaussian within a mixture (based on a single filter) is mixed against the pseudo-states of the other mixtures. The pseudo-states are approximations of the entire mixture, and using them as a source for a mixing process will distribute the approximation distortion to the other mixture elements. Most likely due to these approximations, the macromixture form of the IMM performed the poorest in the simulations.

3.7.4.4 Mixture Reduction. Note that this configuration will perform a reduction cycle for each mixture (corresponding to a specified dynamics model) at each epoch. Considering that the reduction process is one of the most computationally expensive steps, performing it multiple times per epoch can quickly lead to significant computational loading.

3.8 Track Loss Checks

In cluttered cases, the filter can lose track. Traditionally, track loss is declared when a target-originated measurement has not been incorporated for the previous five time periods [20, 30, 35]. This can be interpreted as meaning that the current filter estimate is based primarily on clutter and is meaningless or wrong. In addition, most trackers implement an additional check to see if the combined (overall algorithm) estimate is within 10σ of the truth measurement. Experience shows that the combined estimate can be significantly far away from truth while still incorporating the truth measurement. This is the essence of deferred decision making. Williams [35–37] and Smith [30, 31] used a modified criterion that declared track loss when *all* hypotheses yielded target estimates that were more than 10σ from truth to account for this. Visual inspection of several cases involving track loss in clutter indicated that these additional criteria were problematic and not necessarily beneficial. Hypotheses that do not update on the truth measurement tend to follow trajectories away from the flight path or are updated on clutter, inciting a random walk as in Figure 3.15. As such, any reacquisition is often based solely on chance. Reacquisition is more likely in very low clutter, as the covariance will grow with each missed update until it encapsulates truth, but the rich measurement field of a cluttered environment precludes this form of reacquisition.

In fact, the longest example of track loss and reacquisition found in simulation was an instance of two missed updates before reacquisition. As outlined in Section 4.2, the average track loss scenario was recreated using the more stringent no-update-in-five-epochs track loss criterion without the additional distance checks, and the results were identical. Additionally, the primary focus of this research does not deal specifically with track life, so a more rigorous definition does not impact results. In fact, deferred decision-making tends to allow the blended estimate to wander far from truth, making the performance benefits of using multiple filters to track differing target maneuver modes difficult to see. The multiple model structure will still benefit the (dynamics model) hypotheses tracking truth, but if those hypotheses do not have sufficient mixture probabilities to contribute significantly to the overall mixture state estimate, the performance benefit will be masked by the erroneous hypotheses.

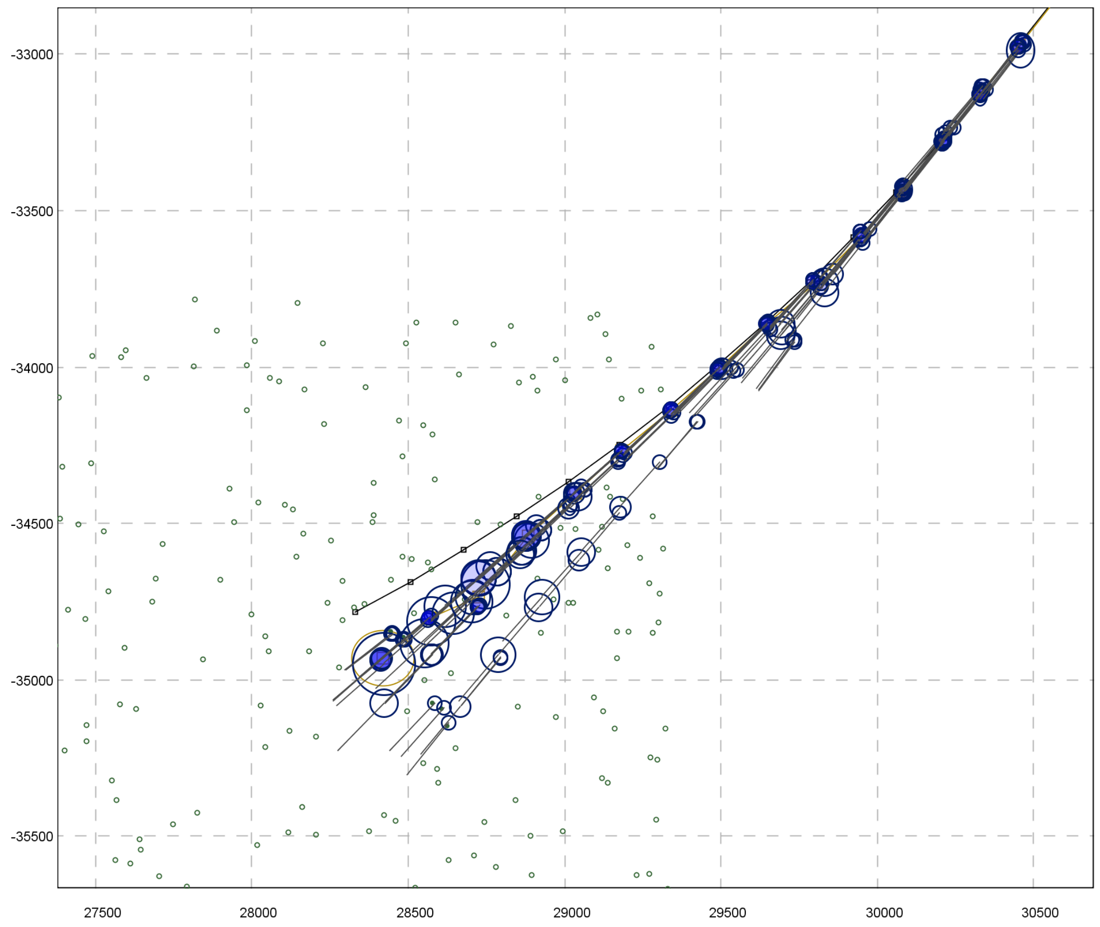


Figure 3.15: In this case, the truth measurement, indicated by the solid black line, has not been incorporated into the mixture for five time periods. The mixture, moving under the assumption of a constant velocity, continues in a straight line, falling away from truth. Reacquisition is unlikely unless the true target location maneuvers into the path of the mixture.

3.9 Probability Marshalling

The speed of probability flow back and forth between filters in multiple model systems is unequal depending on the assumed dynamics of the filters. This can be detrimental in filter applications in which a filter with poor noise rejection capabilities (i.e., based on an aggressive dynamics model) is in force longer than necessary because of its inability to hand off the probability to the more benign filters. Korn and Beean [11] proposed a mechanism that resets filter probabilities at a regular interval. In cases without clutter, such a probability reset (forcing all of the probability to the benign filter) that *incorrectly* removes probability from the filter assuming more aggressive dynamics will be quickly rectified, so the technique is generally useful. However, the same occurrence in clutter can have devastating effects.

Mismatched filter dynamics in clutter can cause the filter to diverge and even lose track, so care must be taken not to reset the filter when it is *correctly* assuming an aggressive dynamics model state. Fortunately, in the case of Gaussian mixtures, considerable insight as to the state of the target can be derived from the qualities of the mixture. Target maneuver can initiate deferred decisions in the multiple hypothesis structure that cause the Gaussian components in the mixture to spread out. If the separated hypotheses carry considerable mixture weight, they can cause the covariance of the mixture to increase via the mean spreading terms in Equation (2.31). So, simple observation of the filter-computed covariance can indicate whether the target is maneuvering. To exploit this fact, the Korn & Beean resets were modified so that they would *not* reset the mixture if the covariance is above a certain threshold. The threshold is easy to identify – a non-maneuvering target tracked under the assumption of aggressive dynamics would have a filter-computed covariance equal to the steady state filter-computed covariance of the most aggressively tuned filter. The only way that the computed covariance can be greater than this value is in the presence of mean spreading. In the simulations run here, the threshold was keyed off the most aggressive computed covariances of the velocity states, although the position states could be used as well. In the tuning process, the velocity states were found to be more conservatively tuned than the position states, so the probability reset threshold was somewhat higher, allowing a small amount of mixture separation before the process blocked elemental filter probability resets.

Figure 3.16 revisits the probability calculation example given in Section 2.4.2. Here, an *ad hoc* technique has been applied to reduce increase the ratio of benign probability to aggressive probability in their region of overlap by modifying the probability calculations so that:

$$f\{\boldsymbol{z}(k)|M_j, \boldsymbol{Z}^{k-1}\} = \begin{cases} f\{\boldsymbol{z}(k)|M_j, \boldsymbol{Z}^{k-1}\} & \text{if } f\{\boldsymbol{z}(k)|M_j, \boldsymbol{Z}^{k-1}\} \geq f\{\boldsymbol{z}(k)|M_i, \boldsymbol{Z}^{k-1}\} \\ \alpha f\{\boldsymbol{z}(k)|M_j, \boldsymbol{Z}^{k-1}\} & \text{otherwise} \end{cases} \tag{3.6}$$

where j denotes a filter with a covariance greater than filter i, and α is a scalar value that determines the amount of penalty to apply to redundant probability from Gaussians with higher covariance. A value of .5 was found to be an acceptable value and will be discussed further in Section 4.5.2. The conditional probability functions are the output of Kalman filters, and are Gaussians, but the modification applied here destroys that assumption and the volume under the CDF no longer sums to one. In practice, this is of little consequence, as the individual probabilities are normalized before being used in the multiple model probability calculations (Equation (2.13)). This technique imposes a procedural constraint on the system in that the individual filter PDFs need to be calculated in order of increasing covariance (aggressiveness) because the conditional PDF evaluations require knowledge of the more benign filter PDFs. While no reason exists that this method cannot be used in concert with the Korn & Beean method, the two methods accomplish the same purpose, and function better alone. The method presented here will be referred to as the "dimpled Gaussian" filter for its characteristic shape.

The final technique does not necessarily manipulate probability. Rather, it decouples the axes of the tracker so that maneuvers in one axis do not cause the other axis to change modes. Until now, the elemental filter probability has been retained in μ. Now, consider that two probabilities are maintained, μ_x and μ_y, one for each axis (assuming a 2D tracker). These values are calculated by evaluating Equation (2.18) separately for each row in the $\mathbf{H}$ matrix (and the corresponding elements of $\boldsymbol{z}$). Additionally, the blended estimate equations need to be modified to handle per axis elemental filter probabilities through

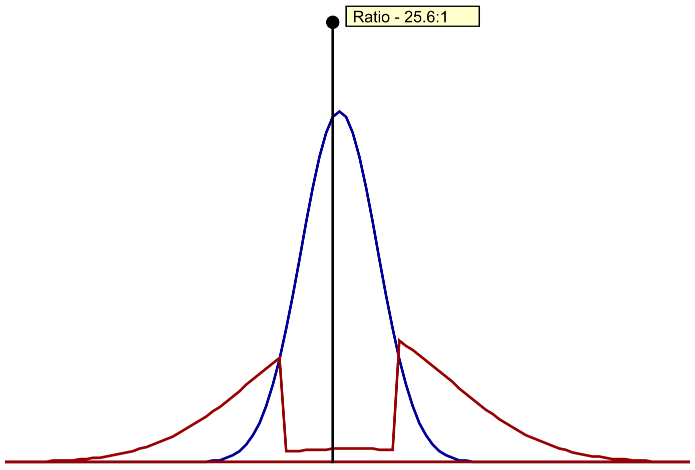


Figure 3.16: Dimple Gaussian Multiple Model Probability Calculation. An *ad hoc* technique is applied to reduce the overlap between two Gaussians functioning in a multiple model structure. A simple test determines if the current filter's PDF is greater than that of a filter with a smaller covariance. If so, the PDF for the filter with the larger covariance is divided by a scalar, the magnitude of which determines the magnitude of effect on probability flow.

block multiplication:

$$\hat{\boldsymbol{x}}(k|k) = \sum_{j=1}^{N_f} \begin{bmatrix} \mu_{x,j}(k) & 0 \\ 0 & \mu_{y,j}(k) \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}}_{x,j}(k|k) \\ \hat{\boldsymbol{x}}_{y,j}(k|k) \end{bmatrix} \tag{3.7}$$

where $\hat{\boldsymbol{x}}_{x,j}(k|k)$ are only those filter states pertaining to the X axis of the j-th filter. Note that this assumes a model structure where the axes are actually decoupled. Many trackers are based on angle distance measurements, and this technique would not work in those cases.

3.10 Error Estimation and Performance Analysis

The performance of the multiple model algorithms can be quantified in several ways. Most directly, one can observe the magnitude of the miss distance, the difference in the estimated state and the true state. The RMS miss distance for both the position and velocity states will be used as a common metric for comparing performance between different filter configurations. Also, a Kalman filter gives the best results when its filter dynamics match the actual dynamics. Therefore, one can observe the multiple model probability values and see if the multiple model architecture is assigning elemental filter probability values appropriately. If so, the filter should offer better performance in rejecting measurement noise, thereby reducing the RMS error.

3.11 Summary

Chapter 3 outlined a multiple model multiple hypothesis tracker. The tracker generates target trajectories using a random noise process (Section 3.3.1), synthetic loop models (Section 3.3.2), or the FlightGear Flight Simulator (Section 3.3.3). The simulator is designed to run in several configurations: a simple multiple hypothesis tracker (Section 3.7.2), a multiple model structure of Gaussian mixtures (Section 3.7.4), and a Gaussian mixture of multiple model components (Section 3.7.3). Additionally, several elemental filter probability marshalling techniques are implemented, such as a modification of the Korn & Beean method that uses mixture analysis to avoid resets during dynamic periods, a split filter that maintains a different multiple model probability for each axis, and the dimpled Gaussian filter that artificially modifies the PDF of a benign filter to affect the benign-to-aggressive PDF ratio.

IV. Simulation Results

4.1 Introduction

Chapter 3 documented an MHT tracking architecture capable of tracking synthetic models and realistic flight data in clutter using several different filter configurations. The results of those simulations are presented here, in addition to a revisiting of the Williams' simulations for code verification.

4.2 ISE Implementation

The track life simulation originally conducted by Williams was reaccomplished using a new code base to verify the performance of the ISE implementation. Figure 4.1 compares the results with Williams' original and the simulations by Petrucci [21], who explored variants of the ISE algorithm. All three simulations provide identical results and demonstrate the repeatability of the ISE performance and validate the implementation performed here. Note that the version performed here used the simplified track loss criterion discussed in Section 3.8, yet still garnered the same results.

As stated in Section 3.4, the clutter region for the following simulations was generated around the best estimate of the mixture propagated forward to the current epoch. The size of the clutter region was maintained at $200\sqrt{\mathbf{R}}\lambda$ meters, yielding 480 expected clutter measurements per epoch. Figure 4.2 shows the average track life based on this clutter region compared to the original truth-centered clutter region and an extended truth-centered clutter region that has 16 times the area[1]. The average track life values are more pessimistic than before, but are in closer agreement with the values generated using the extended clutter region. This demonstrates that the standard clutter region imposes an artificial track measurement starvation phenomenon that yields and erroneously favorable indication of algorithm performance, whereas the extended clutter region and proposed estimate-based region both provide accurate portrayals of real performance potential.

[1]Williams used an extended clutter region that had 100 times the area of the original, generating 48,000 measurements per epoch. Visual inspection of the simulations indicated that the smaller extended region used here was sufficient, and produced results very similar to those of Williams.

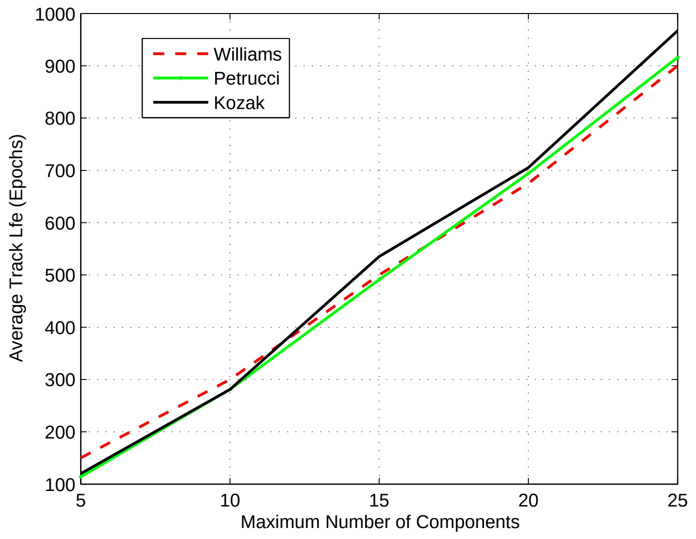


Figure 4.1: ISE Average Track Life

Even though the moving clutter region yielded similar results, it does not work in the same fashion as the extended clutter region. In the case of the extended region, false hypotheses are pruned from the mixture after poor update cycles, never from running outside the region of generated clutter. In the moving clutter region case, hypotheses are artificially terminated by leaving the clutter region, but now the truth measurements are just as likely to be lost by leaving the region. Considering that acceptable performance was obtained using the estimate centered clutter region, this method of clutter generation was used in the remainder of the following simulations.

Figure 4.3 shows the average calculation time per epoch. Notice that the computational loading increases quadratically with the number of components. Considering that the simulations performed here assumed a measurement sample frequency of either one or two Hz and used 15 components, one might conclude that the algorithm could perform in real time. Unfortunately, the values are average computation times, and the speed of the algorithm often depends on the acceleration techniques outlined in Section 2.6.6, which tend to become less effective as the components in the Gaussian mixture become more disjoint due to clutter or target maneuvers. Consequently, the loading of this algorithm is not fixed per epoch and should still be assumed to be non-real time.

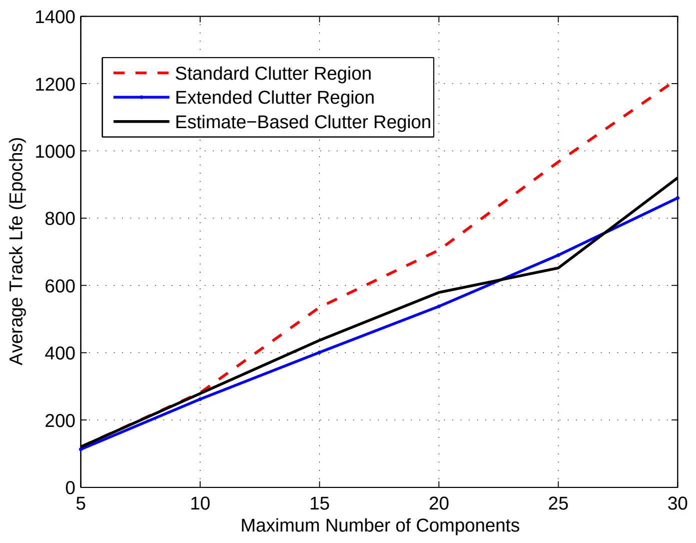


Figure 4.2: Average Track Life Using an Estimate-Based Clutter Region

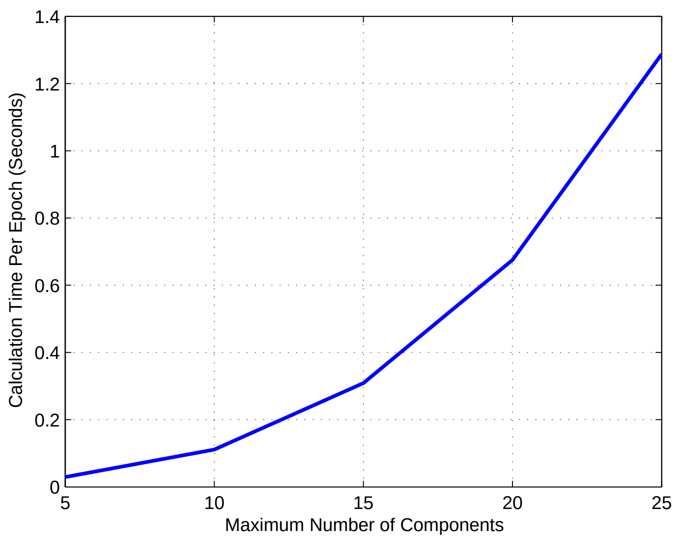


Figure 4.3: Average Calculation Time Per Epoch

4.3 Loop Models

The two loop models presented in Section 3.3.2 are presented here to examine the performance of both multiple model structures and certain probability flow techniques in an easily observable environment. Considering the nature of the loop models, gradually increasing dynamics, the expected performance is distinct periods of probability flow to the appropriately tuned filter, with transition periods between. The broken-loop models should illustrate flow characteristics in the transition periods predicted by the discussion in Section 2.4.2, specifically that flow will be more sluggish moving from an aggressive filter to a benign filter than vice versa.

The results of Monte Carlo simulations in this and every following section are presented in a standard eight-plot format, the first example of which can be seen in Figure 4.4. The top four plots are error plots for the X and Y position and velocity states. The three thin black lines running through the center of the plot are the mean error $\pm 2\sigma$ of all of the viable (still tracking the target) filters at each time epoch. In a well-tuned filter, the mean error should be zero. The thick gray lines indicate the filter computed $0 \pm 2\sigma$ values. If the filter is properly tuned to the observed target dynamics, the computed filter standard deviations should agree with the true standard deviations. Each error plot also has a pair of dashed red lines, which indicate the $0 \pm 2\sigma$ values of the most aggressive filter in the tracker running at steady state in no clutter. This is useful in mixture analysis and will be employed in the modified Korn & Beean method of Section 4.5. If the filter-computed standard deviation for an MHT, macrohypothesis, or macromixture algorithm at any time is greater than the steady state standard deviation, then the computed standard deviation is being influenced heavily by mean spreading terms, indicating that the mixture is in a deferred decision making process. The RMS error plots are beneath the four X and Y error plots and indicate the mean miss distance of the tracker best estimate, calculated using only filter values (from all of the Monte Carlo simulations) that are still tracking the target. The temporal average of the RMS error is displayed above the plots and is a fairly good indicator of the performance of the filter as a whole, although the large errors accrued by MHTs during a deferred decision-making process can make the average difficult to interpret. The lower left plot shows the elemental filter probability flow (the mean μ

values for each filter) and will always be unity in single-filter scenarios. In the case of a macromixture structure, this will simply show the elemental filter probability (as calculated by the mixture pseudo-residuals) for each mixture. In the case of a macrohypothesis structure, this will show the mean elemental filter probability of each filter model for every macrohypothesis. The mean will be weighted by the individual mixture probabilities, so it does not necessarily indicate the probability flow for the correct hypotheses, only those that carry the majority of the mixture probability (these will not be in agreement during a deferred decision-making process). The final plot on the lower right shows the total number of runs, always starting at 50 for these simulations, contributing to the statistics at every epoch. It should be consulted to gain a good impression of loss-of-track difficulties. Runs are removed from the simulation 7 epochs (chosen as padding to prevent errors from entering the calculations) before track loss is declared. From Section 3.5, statistics are assumed to be valid if more than 25 runs are contributing.

4.3.1 Clutter Free. Uncluttered scenarios are presented here to familiarize the reader with characteristics of a multiple model filter running in ideal situations and to provide a baseline for best performance against clutter. The addition of clutter will always degrade performance.

Figure 4.4 shows the performance of a single benign constant-velocity (CV) filter against the simple loop model. A filter can be expected to perform well if the filter-computed covariance is equivalent to the actual covariance. In all of the error estimates, it is obvious that the filter is poorly matched past epoch 100. The magnitude of dynamics increases linearly with time, so the poor filter performance implies that the filter can only handle dynamics with magnitude less than those incurred prior to epoch 100. Because this case involves no clutter, the filter does not lose track, so the error manifests itself as offset (the mean errors should be zero). While the filter does not lose track, the estimate is, at times, significantly far from truth. The bias is highly correlated with the sinusoidal input of the truth model and shows how the loop model effectively alternates input dynamics between the axes. Overall, this can be said to be a poor filter for the truth case, but this

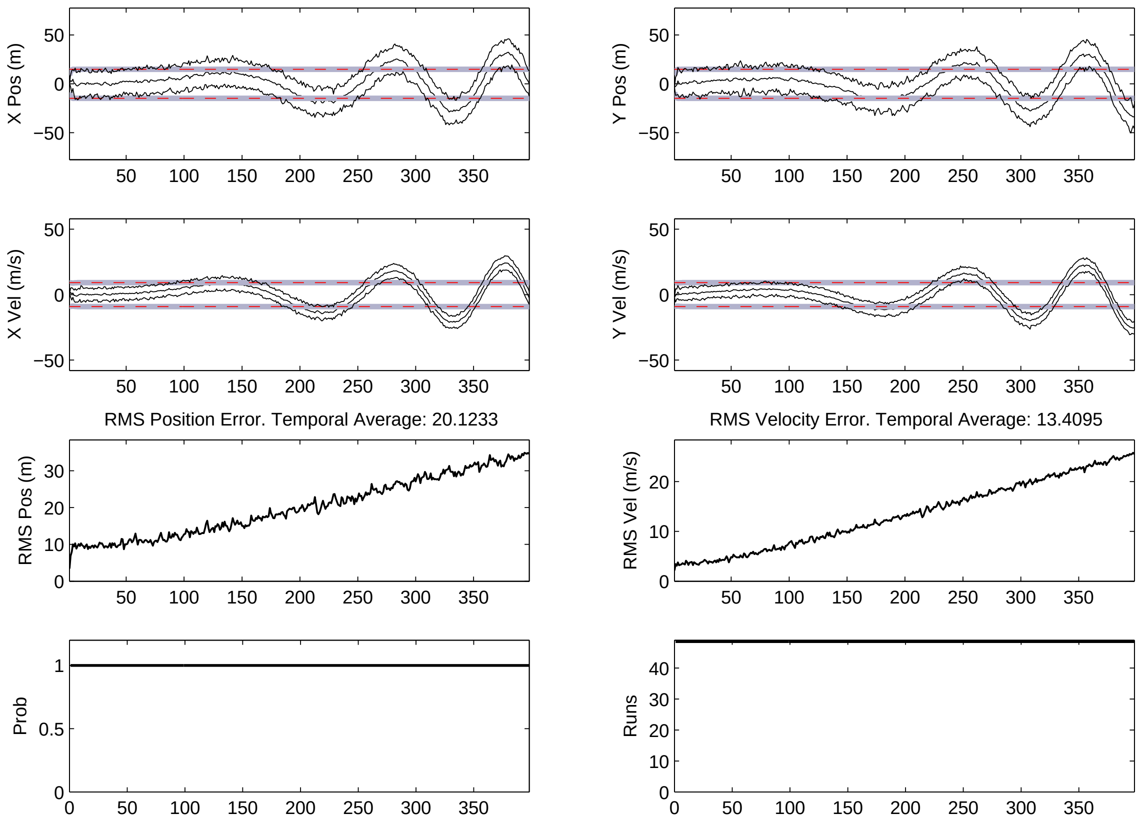


Figure 4.4: Benign Filter: Loop Model Without Clutter

filter does offer substantial noise rejection in the low dynamics regime for which it is tuned (1g maneuvers).

Figure 4.5 shows the aggressive constant-velocity filter performance versus the simple loop model without clutter. Here, the filter is a better match for the truth dynamics for a greater length of time into the loop simulation, as can be seen in the more closely matched computed and actual filter covariances. Still, the filter is a suboptimal match past epoch 200. Notice that the RMS error does not increase as drastically with the harshening dynamics as in the benign filter scenario, which indicates a better match of the filter-assumed process noise strength to actual target dynamics.

Figure 4.6 shows the results of an MMAE composed of the two filters shown in Figures 4.4 and 4.5. As mentioned before, the benign filter is a poor fit past epoch 100, and probability can be seen flowing to the aggressive filter shortly thereafter. While the probability flow plot displays the filter in force at a particular time, the flow can also be

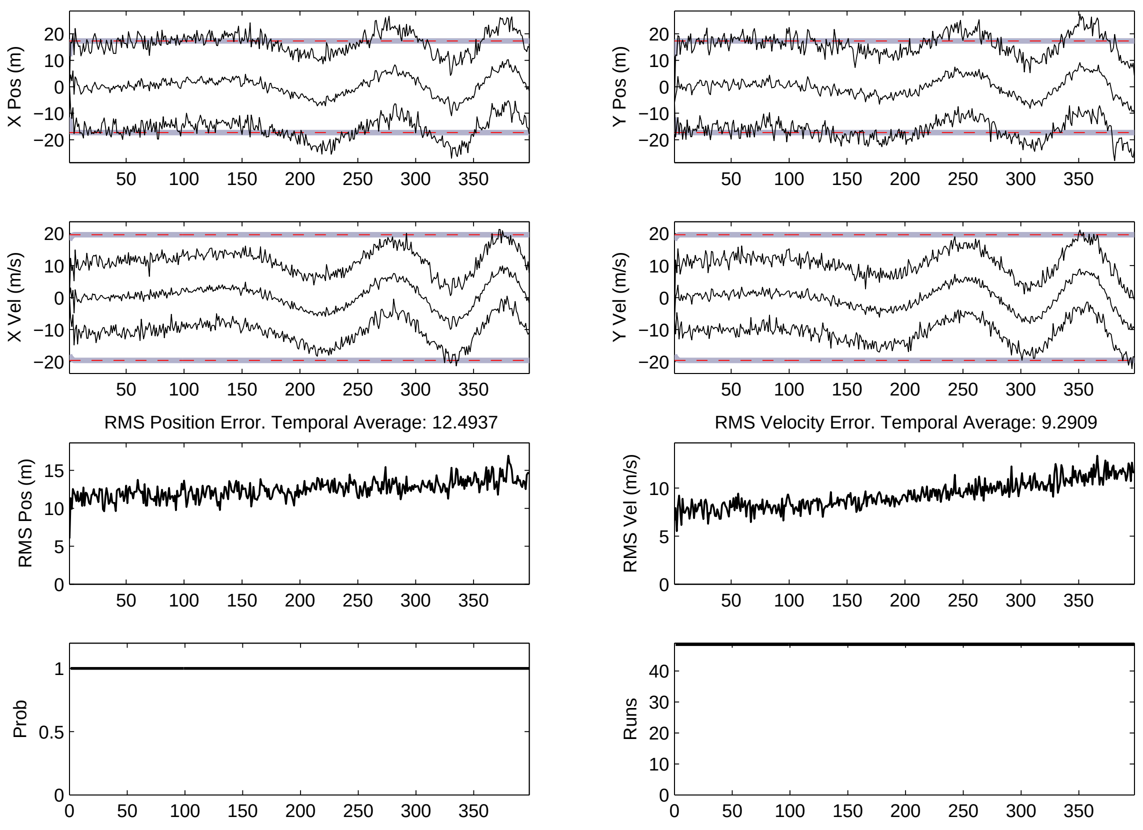


Figure 4.5: Aggressive Filter: Loop Model Without Clutter

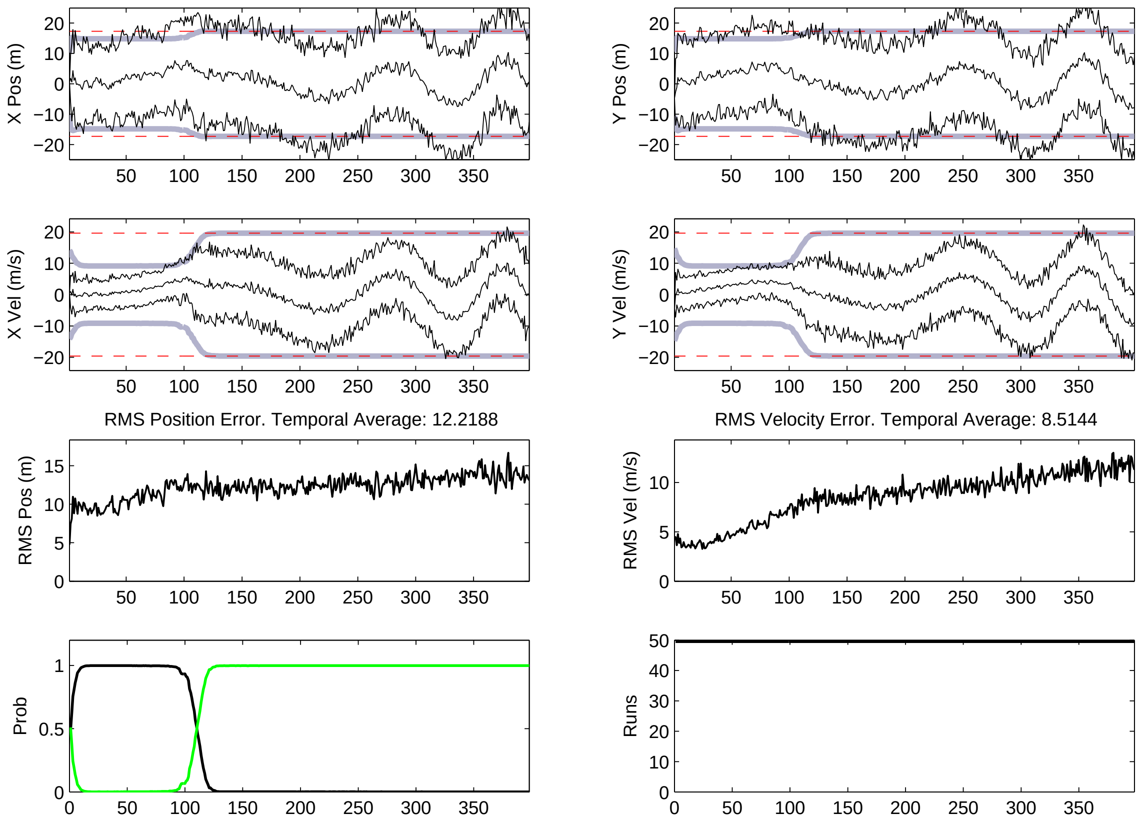


Figure 4.6: MMAE Filter: Loop Model Without Clutter

inferred from the filter-computed standard deviation, represented by the thick gray lines in the plots. Additionally, the dominant filter will dictate the covariance (and thus the filter-computed standard deviation portrayed in the figure) at the end of the propagation cycle. So, by visualizing the propagated covariances, one can also see probability flow (see *SimpleMMAE.avi*). Note that the average RMS error in both position and velocity is smaller than in the case of either elemental filter running alone.

The discretization of parameter space in this example is not optimum – ideally, we would desire transition at epoch 200 (the middle of the simulation in time) so that the maximum benefit is derived from both elemental filters. This would require increasing the q values on both filters. Given that the truth model is based on constantly increasing dynamics, proper parametrization would depend solely on the length of the simulation. For example, if Figure 4.6 were cropped at epoch 200, the filters would seem to be well selected. The filter tunings selected here were chosen to give good performance in the three-filter and broken-loop simulations; consequently, the performance here is somewhat suboptimal.

Figure 4.7 represents an MMAE containing 3 elemental filters. Here, the filters can be described as well tuned. Note that non-dominant filters at any time maintain a probability of .001, the lower bound, while not well matched to observed dynamics. This probability flow is desirable as it shows that the MMAE is capable of identifying the appropriate filter to use at a given time, thereby preventing poorly matched filters from contributing significantly to the blended result. Unfortunately, this performance is not observed in cluttered runs and will be fully developed on Figure 4.15. Additionally, the lower bounding implies that the filters are well separated in parameter space. If the filters were too close together, the probability would not decisively flow to any one filter. Rather, the mean probability value would be split between two of them at a time (or even among all three). This is seen in the transition epochs – during these periods, the filters are equally well matched to the dynamics, and the result is a smooth interpolation between them (see epochs 100-150 and 300-350).

Figure 4.8 shows an IMM implementation using a non-preferential Markov transition matrix defined in Section 3.6.3.2. Because of the intermixing process, the multiple model probability flow does not exhibit definitive swings to a specific filter. Rather, the flow represents a "degree of fit" for each model. Again, there is smooth transition between the filters as the dynamics increase. Notice that, as the benign filter falls out of favor, it approaches a lower bound of .1. This is expected, given that the transition matrix for this simulation (Equation (3.5)) uses .1 for its non-identity mixing values, essentially meaning that even when a filter is poorly matched to truth dynamics, the Markov matrix still assumes there is a 10% probability of transition. Again, the average RMS errors are smaller than in the individual cases.

Figure 4.9 shows the MMAE performance when used against the broken-loop model. This truth model is meant to highlight rapid transitions between dynamics levels and display the characteristic multiple model responses to quick shifts in dynamics. Namely, the transition from a benign filter to a more aggressive filter is fairly quick (see epoch 300 in the lower left plot of Figure 4.9), but the transition to a more benign filter is much

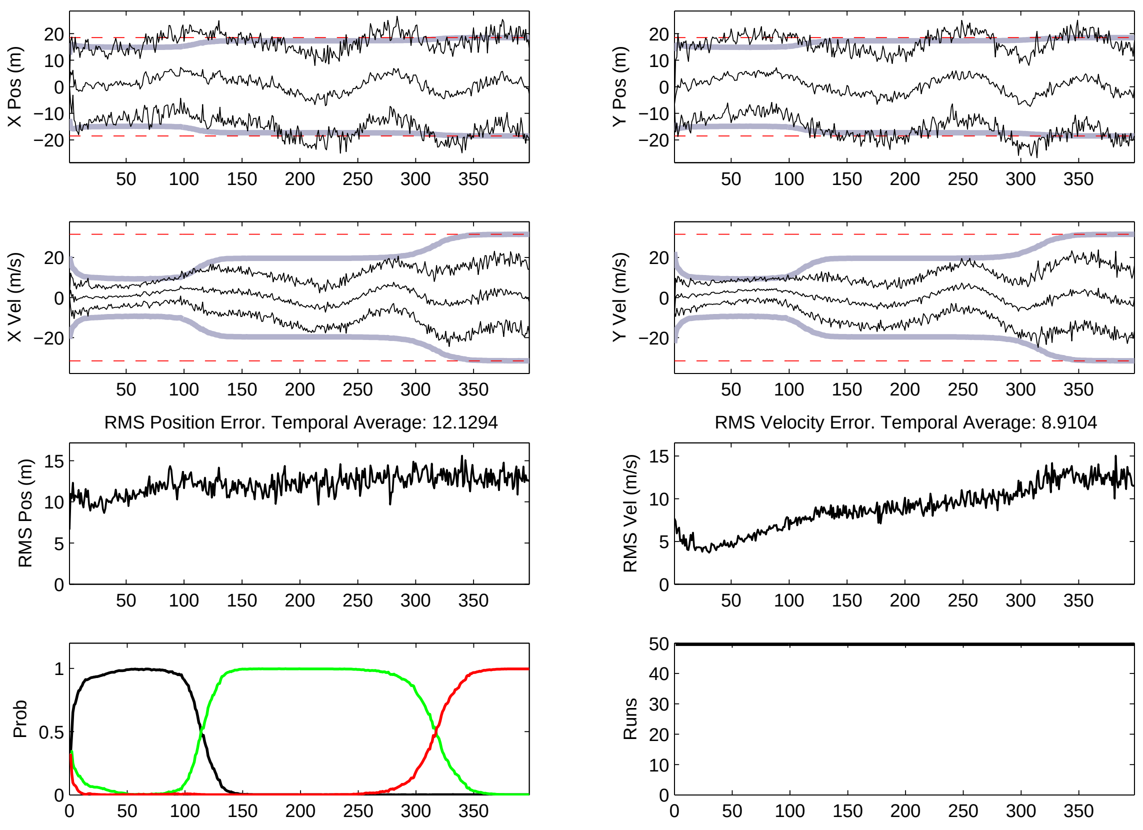


Figure 4.7: Three Filter MMAE: Loop Model Without Clutter

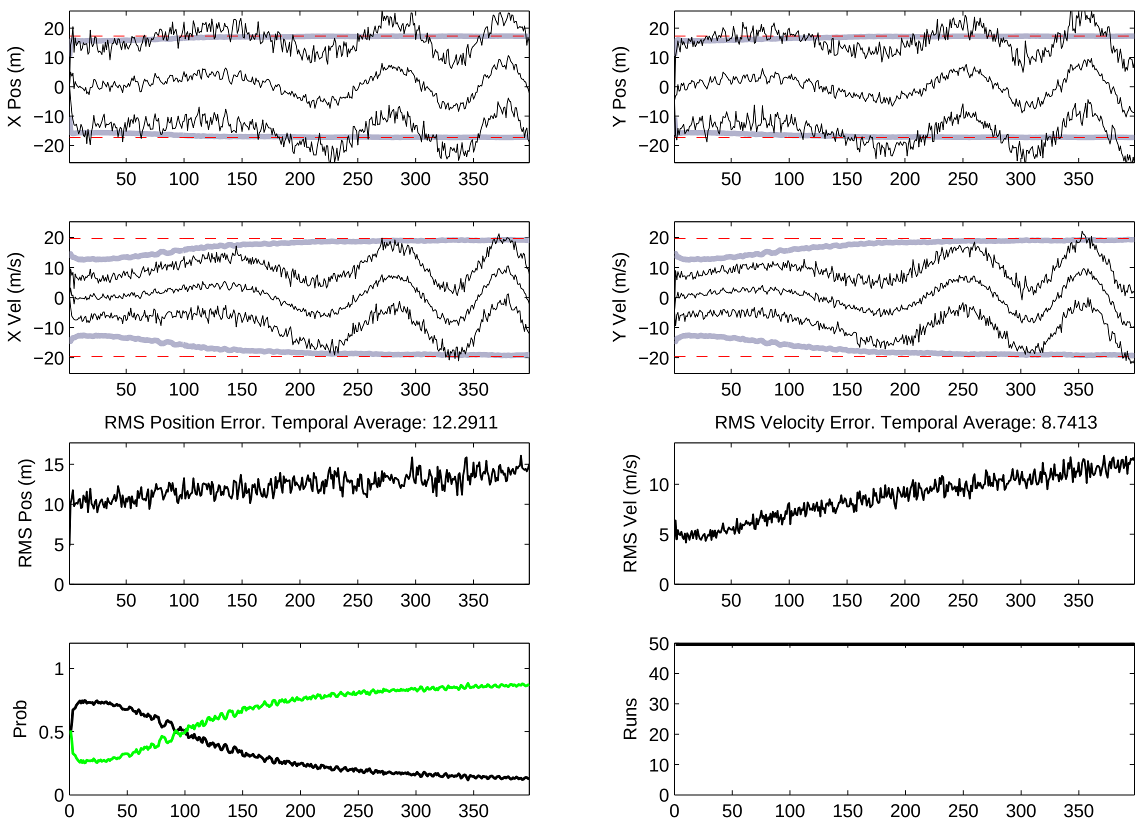


Figure 4.8: IMM Filter: Loop Model Without Clutter

slower (see epoch 330 in the lower left plot of Figure 4.9). The lower plot of Figure 4.10 shows the actual dynamics, which can be compared to the individual elemental filter flows above it to see the estimator's lag in identifying dynamics mode changes. Some methods for resolving this are addressed in Section 4.5.

Figure 4.10 shows the elemental filters compared against the input dynamics. Note the mean probability flow transitions always lag behind the dynamics transitions by several epochs. The exact length and rapidity of change are determined by the aggressiveness of the maneuver and the parameter spacing of the filters. Notice the filter transition due to dynamics over epochs 180-210 is slow enough that full transfer does not occur. At this time, the truth dynamics are fairly well matched for either filter and definitive flow does not occur. The broken-loop dynamics increase at half the rate of the loop model dynamics because of the 50% duty cycle, so the magnitude of the dynamics at this transition is the equivalent dynamics of the loop model over epochs 90-105. Referring to Figure 4.6, we see that this is a period of marginally better fit for the aggressive filter, so the transitions should be sluggish. By the next few cycles, the dynamics more definitively match the aggressive filter, and the flow is quicker and more decisive.

4.3.2 Clutter. For all of the scenarios in this section, clutter was added at a density of $\lambda = .00005\, pts/m^2$, which yields approximately 200 clutter points per epoch. 15 Gaussian mixture components were carried out of every reduction cycle using an ISE-based mixture reduction algorithm (MRA).

Figure 4.11 shows the benign filter running against the loop model in clutter. Even though the simulation was run until epoch 400, there was total track loss on all simulations occurring around epoch 160. This can be seen in the lower right plot of 4.11, which shows the total number of runs tracking at each epoch. In this case, the target outmaneuvered the filter, which would normally be unacceptable for a single filter tracking a target, but this filter is intended to be used in a multiple model structure with a more aggressive filter that will assume control during maneuvers that would deceive this filter.

Statistics for Monte Carlo simulations that lose runs due to track loss are calculated using only those remaining runs that do not exhibit track loss, so the RMS performance

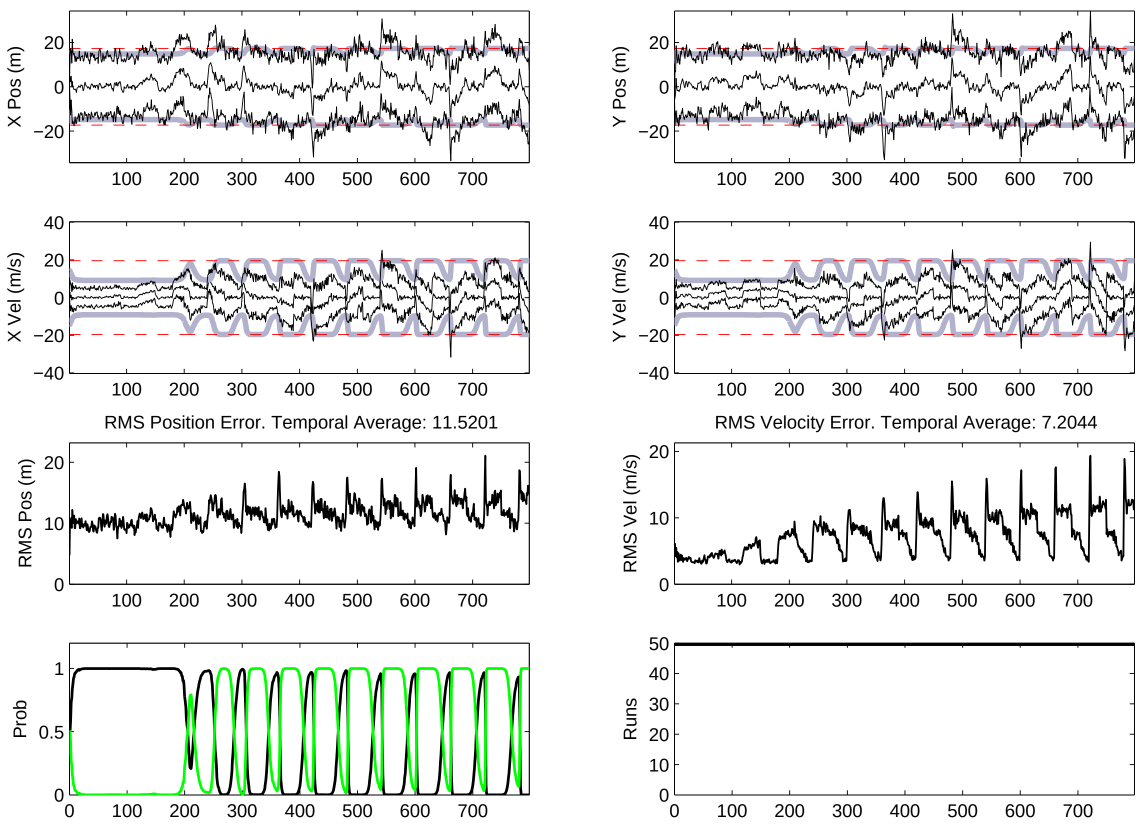


Figure 4.9: MMAE Filter: Broken-Loop Model Without Clutter

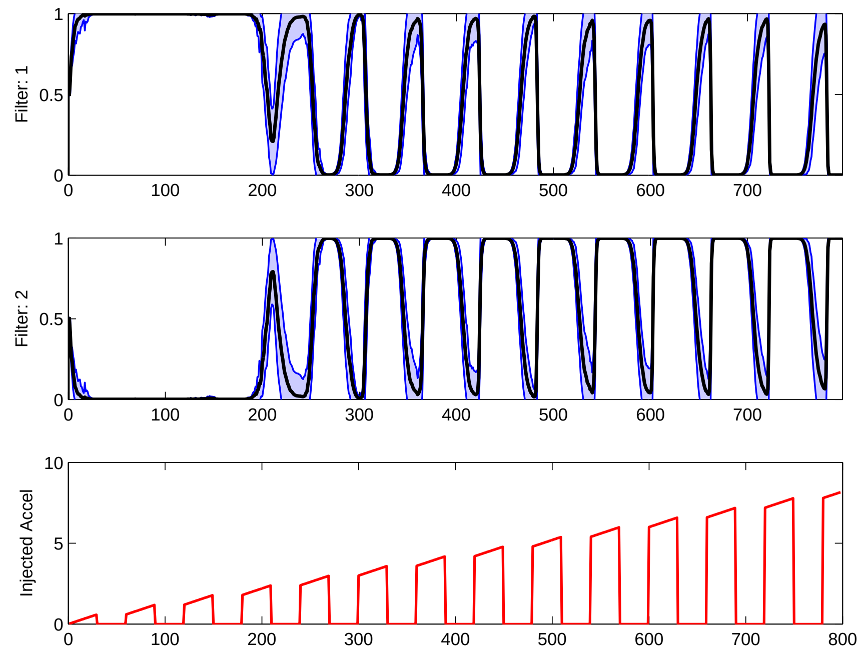


Figure 4.10: MMAE Filter: Broken-Loop Model Without Clutter Elemental Filter Probability Flow (Mean ± 1 Standard Deviation)

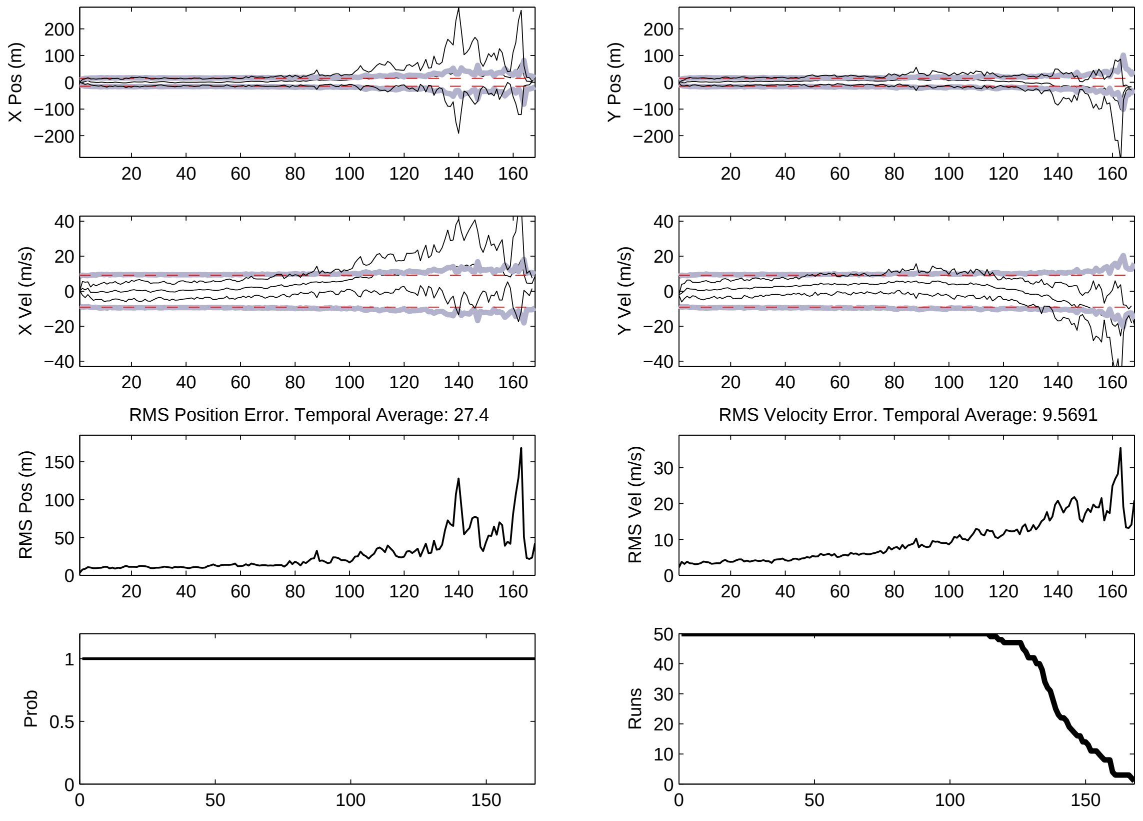


Figure 4.11: Benign Filter: Loop Model in Clutter

numbers for this filter configuration are artificially high. In comparing this filter to others, remember that the average RMS errors are only based on times when this filter was actually capable of tracking the target. To compare this run to the others based on the RMS temporal average, one would need to truncate their evaluation at the epoch where this filter no longer has sufficient contributing runs for meaningful statistics (around epoch 125 in this case, as seen in the lower right plot of Figure 4.11). When considering the full 400 epochs used for the other cases, this filter cannot be compared directly because it is incapable of tracking the target. Consequently, this filter's performance should be considered an impossible lower bound for any of the more aggressively tuned filters running against the defined dynamics and clutter density.

Additionally, note that the mean spreading terms that increase the filter covariance (see discussion in Section 2.5.1 and Equation (2.31)) can be viewed here, especially in epochs 125-150 in the X position error. Component separation distances are used to inflate the covariance of the blended mixture. Therefore, computed filter covariances that are larger than the precalculated covariance for all of the base filters indicate that the mixture is "spread out," or that the hypotheses are not in agreement. This property will be exploited in the probability marshalling processes of Section 4.5. As noted before, this occurs here right before total track loss on all of the runs.

Figure 4.12 shows the aggressive filter run against the loop model in clutter. As in the previous case, the mean spreading terms become visible in the inflated covariance values near the end of the simulation. Note that half of the runs have lost track by the end of the simulation. As the dynamics are always increasing, total track loss is guaranteed if the simulation is run long enough. From Figure 4.5, the RMS error statistic is usually flat for the region in which the filter is well tuned, and increases linearly with the magnitude of the dynamics outside that region. In Figure 4.12, the RMS errors increase proportionally with the target dynamics up until epoch 250 and then begin to increase much more dramatically. This is caused by the filter being deceived by clutter – the filter is finding clutter points with more favorable residuals than the highly maneuverable target. The filter does make associations with the target, but they are assigned lower mixture probabilities, causing the mixture estimate to wander from the truth. Through the deferred decision making

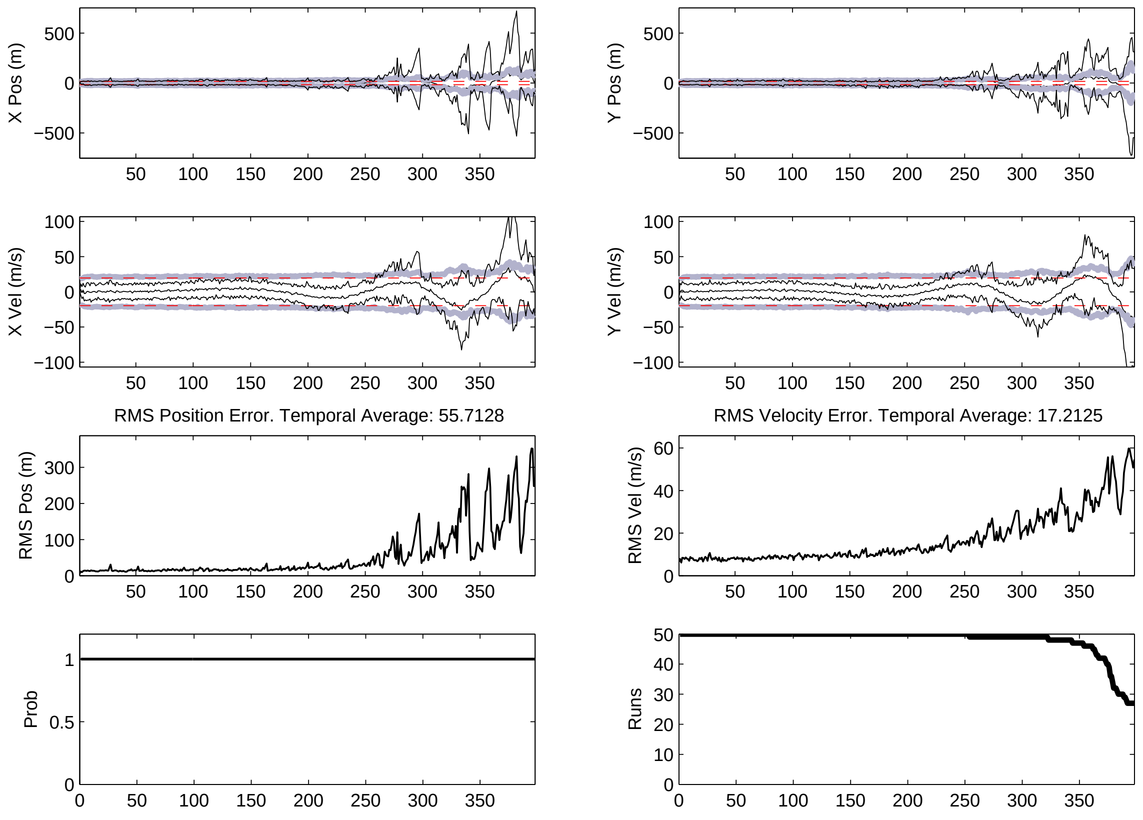


Figure 4.12: Aggressive Filter: Loop Model in Clutter

process, probability may eventually flow back to hypotheses based on truth, but not after significant RMS error has been accrued.

Figure 4.13 shows a simulation using the macrohypothesis (defined in Section 3.7.3) version of the MMAE. Notice that probability flow is clear and well defined as in the uncluttered case (Figure 4.6), but the flow crossover point occurs somewhat earlier than in the clutter-free case. This is explained in that the probability flow is based on the individual filter residuals. In the clutter-free case, the residuals are always based on the true measurement. On the other hand, the cluttered case will have hypotheses based on clutter-originated residuals. Clutter points are likely to be far away from the truth, and will look more dynamic to the elemental filter probability calculation. Clutter will create a "dynamics floor" in that the target will never appear to be entirely benign because there will always be clutter-originated residuals that appear dynamic.

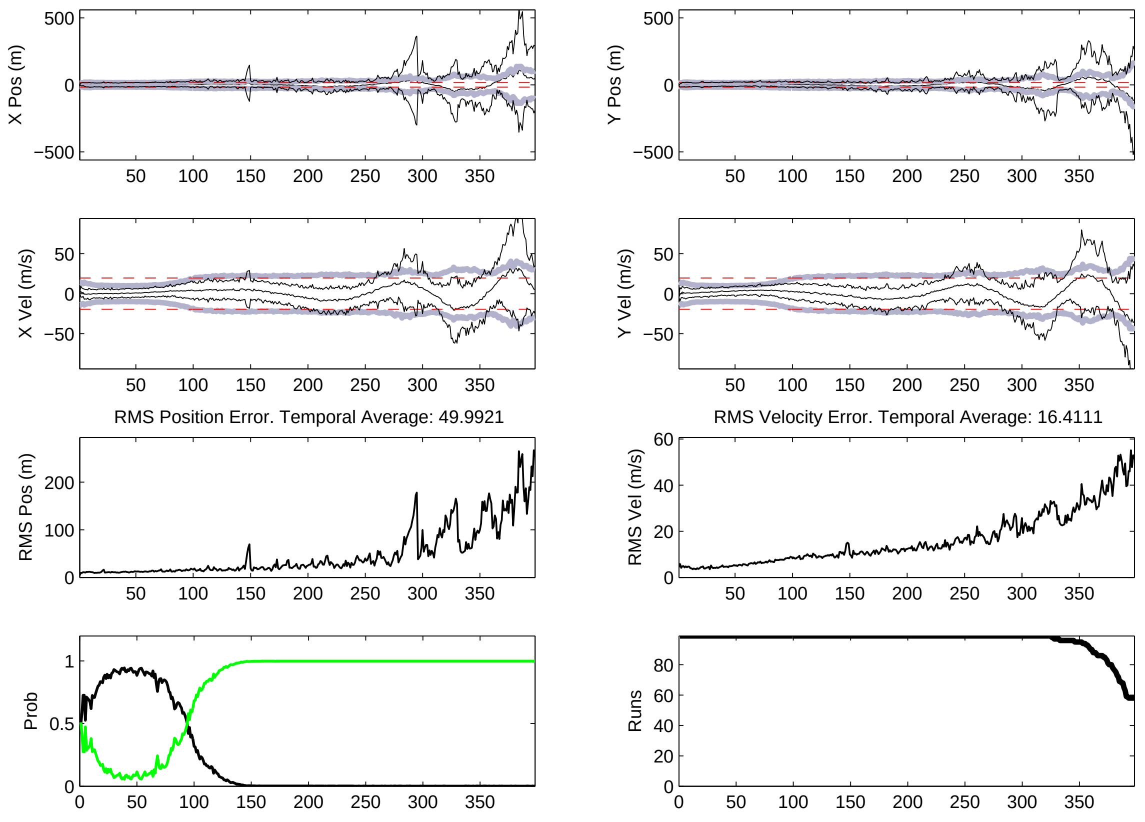
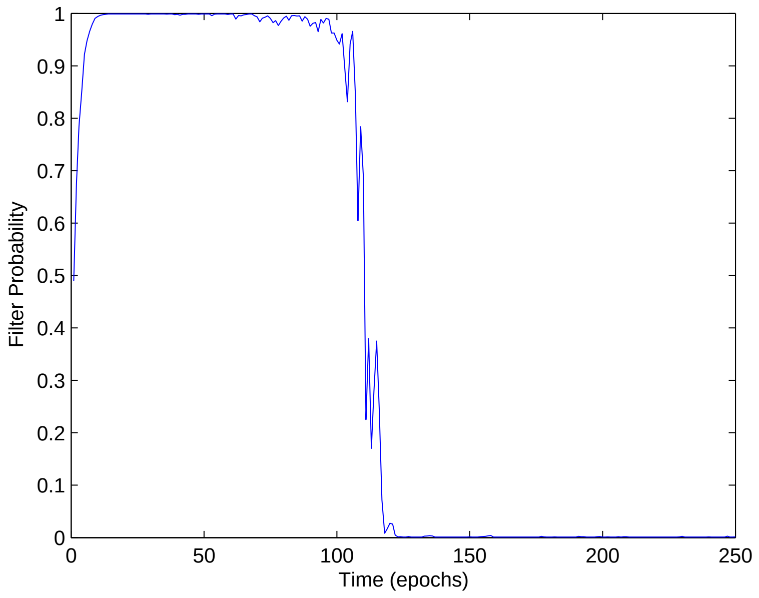


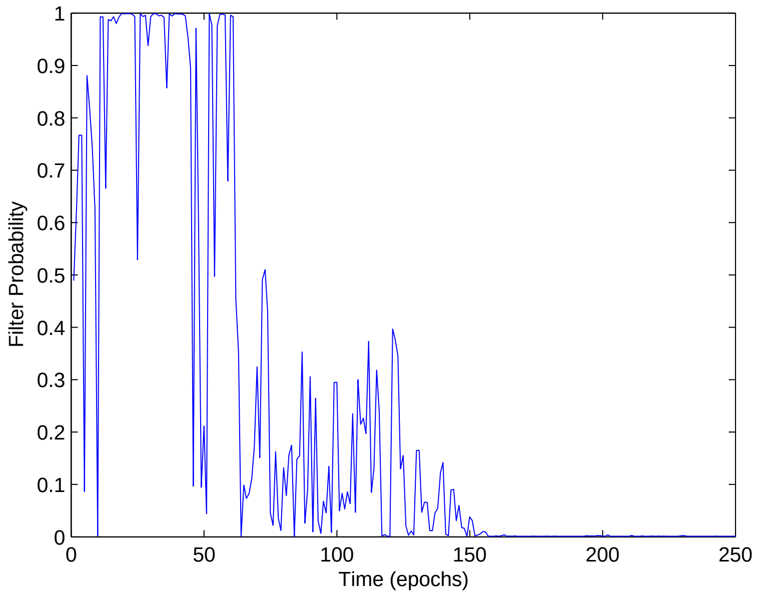
Figure 4.13: Macrohypothesis MMAE Filter: Loop Model in Clutter

Note that the smooth filter probability flow shown here (recall that mean probability values are being plotted) is an "artifact" of the Monte Carlo process. Individual probability flows are, from epoch to epoch, erratic and very noisy. This can be observed in the variance of the probability flow plot. Figure 4.14 compares probability flow of the benign filters in cluttered and uncluttered cases for a single representative run. Given that the truth dynamics are smoothly increasing, one can expect a smooth linear interpolation. Figure 14(a) is fairly smooth, with variations caused entirely by measurement noise. The flow in Figure 14(b) is based on residuals corrupted not only by measurement noise but also by association uncertainty, and is significantly more corrupted.

Figure 4.15 shows a three-filter MMAE running against the loop model in clutter. This run was specifically designed to test the structure's integration with larger numbers of elemental filters. Positively, note that the third filter effectively assumes control during the highest dynamics portion of the run and mitigates the track loss shown before. Also note



(a) Benign Filter: Probability Flow Without Clutter



(b) Benign Filter: Probability Flow with Clutter

Figure 4.14: Effects of Clutter on Multiple Model Probability Flow Calculations

that the RMS errors increase proportional to the dynamics, and the effect of deferred decision induced errors is significantly reduced. This explains why this configuration performs significantly better (in an RMS error sense) than the others.

Unfortunately, the probability flow plots show that the more aggressive filters begin to absorb probability from the benign filter even during the low dynamics portion of the run. Contrast this to Figure 4.7 in which the out of favor elemental filters were lower bounded. This is caused by the structure of this MHT – the more dynamic filters inherently have larger gates. As explained in Section 3.7.3.1, all of the hypotheses in a macrohypothesis are effectively updated on the largest gate, allowing more clutter points into the update cycle. Consequently, the MHT structure will create more false associations, each with reasonably acceptable residuals for the filter based on the most aggressive dynamics model. Incorporating more clutter points with larger residuals will increase the "dynamics floor" and pull probability away from the benign filter. This also explains why the benign filters are, appropriately, lower bounded during high dynamic periods. Their residuals will tend to be pessimistically large in the case of clutter. Figure 4.14 reiterates this point, as the benign filter flow is corrupted when it matches truth (the low dynamics portion of flight), but not when it is entirely out of favor. In fact, the elemental filter probability is at the lower bounds when the truth dynamics are more aggressive than the assumed filter dynamics.

As in Section 4.6, the probability flow can also be visualized in the propagated filter covariances. Figure 4.3.2 shows the difference in propagated covariances, shown as light gray ellipses, between a model operating under benign assumptions and another operating under aggressive assumptions (see movie *LoopClutterMMAE.avi*). The probability bars in the upper left corner show the elemental filter probabilities, μ. The velocity plot in the left shows the velocity state as a black hollow dot, and the 15 estimates (each corresponding to a Gaussian mixture component) as line vectors. The acceleration plot, the circle in the upper right, shows no value because the constant-velocity filter does not estimate this state. Note that filters based on more aggressive dynamics tend to spawn more hypotheses. It is important to note that the gates used during benign flight are the same size as those used

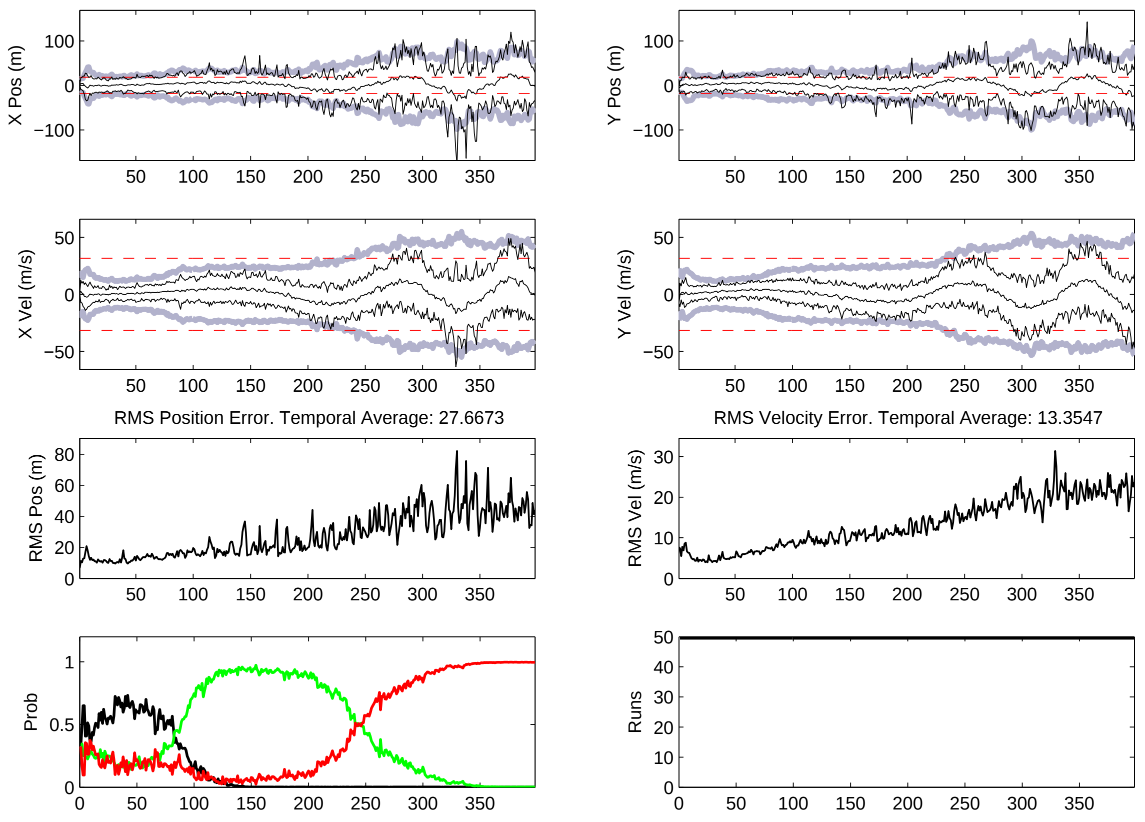
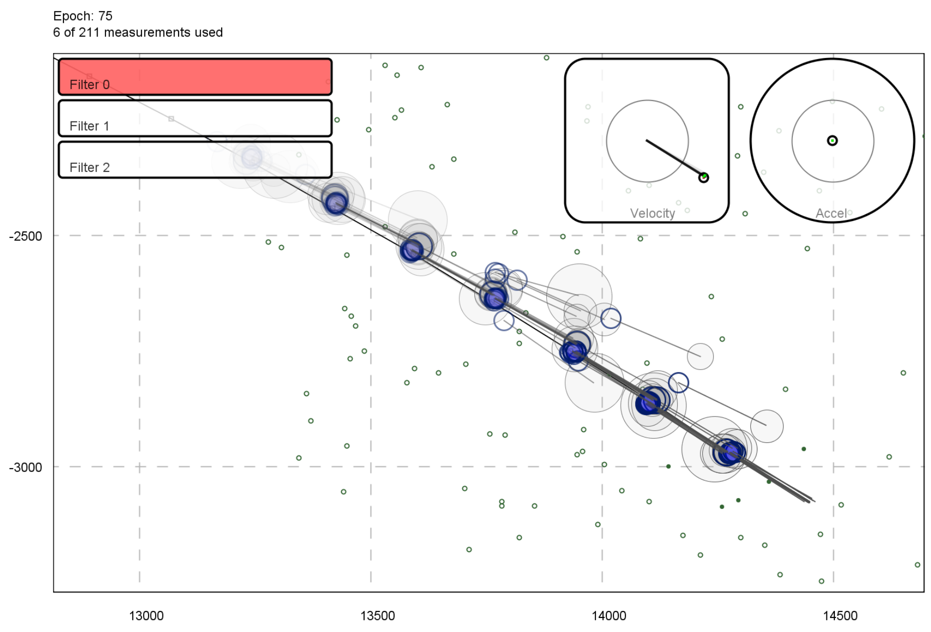
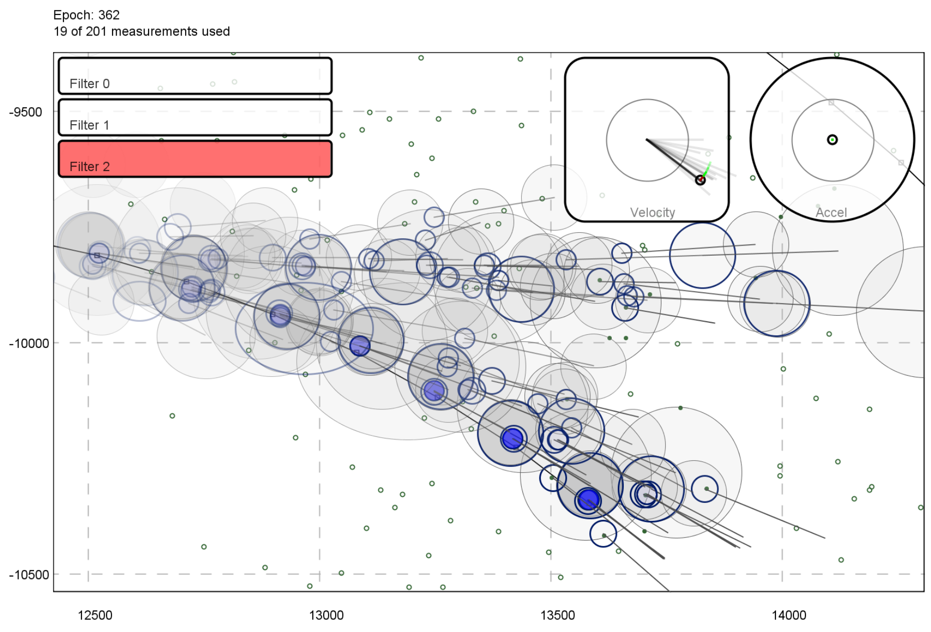


Figure 4.15: Three-Filter Macrohypothesis MMAE: Loop Model in Clutter



(a) MMAE Operating Under Benign Filter Influence



(b) MMAE Operating Under Aggressive Filter Influence

Figure 4.16: MMAE Filter Probability Flow Visualized Through Propagated Covariance

during more aggressive flight[2], so the additional hypothesis are not a product of varying gate size. Rather, the dynamic filters seem to create more associations because the mixture probabilities assigned to the new hypotheses are higher. Referring to Equation (2.35), the mixture probabilities are a function of the hypothesis covariance – when they are based on a more aggressive filter, they will be larger, thereby assigning higher probabilities to clutter points that are farther away. The same number of associations are occurring in Figure 4.16(a) and Figure 4.16(b), but higher uncertainty in filter dynamics allows more of the new hypotheses to survive the reduction cycle. This affects the computational performance of the filter. The speed optimizations covered in Section 2.6.6 become less effective when the probabilities are more evenly distributed. The most obvious example is the optimization that immediately discards low probability ($p_i < 1\text{x}10^{-6}$) components – if the covariances are high enough that all associations are assigned mixture probabilities greater than $1\text{x}10^{-6}$, this modification has no effect.

Figure 4.17 shows the performance of the IMM filter. This filter also suffers from deferred decision induced errors after epoch 250. The probability flow is exactly as before, but the equilibrium point (the point where either filter is an adequate representation of a measurement and probability flow will be split evenly between them) occurs earlier, as in the case of the MMAE.

Figure 4.18 shows the performance of the macromixture-based MHT presented in Section 3.7.4. The elemental filter probability flow transfers as expected, but the equilibrium point occurs later than before in the no-clutter case of Figure 4.6. The transition from benign to aggressive is much slower. In fact, the transition begins at the same time as for the MMAE filter, but takes nearly twice as long. The sluggish transfer is most likely due to the probability flow being based on pseudo-residuals – the probability-weighted averages of each mixture's residuals. Decisive flow does not occur until the majority of the components in each mixture are in agreement. This filter also suffers from deferred decision error after epoch 250.

[2]The exception is gates generated by a hypothesis the epoch after a missed update. The hypothesis will go through consecutive propagation cycles without updates, which increases the covariance, thereby creating a larger gate.

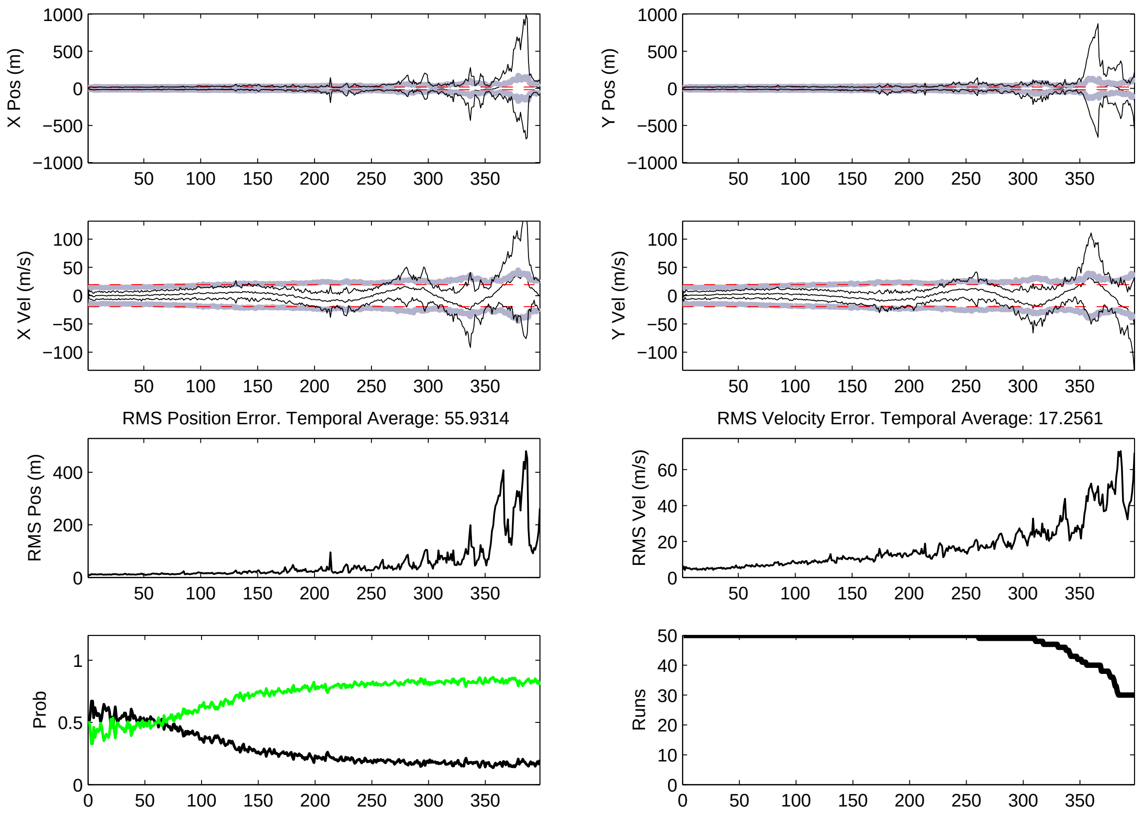


Figure 4.17: Macrohypothesis IMM Filter: Loop Model in Clutter

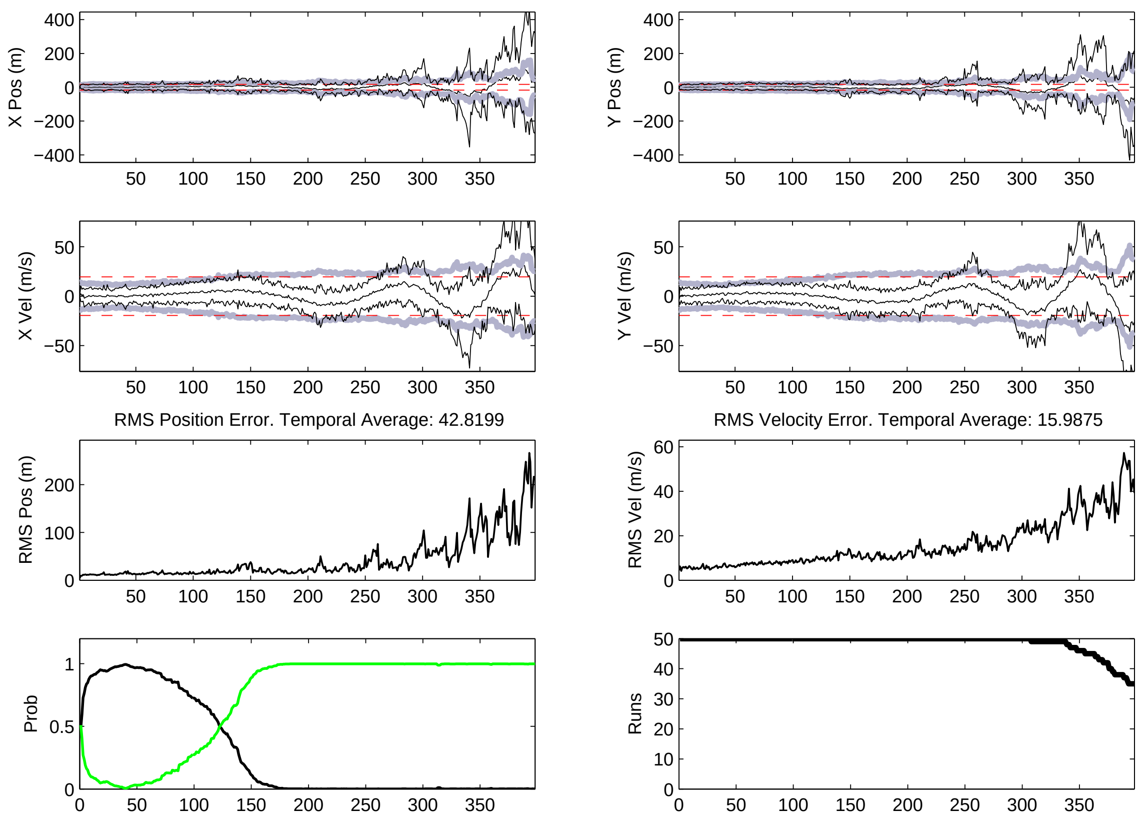


Figure 4.18: Macromixture MMAE Filter: Loop Model in Clutter

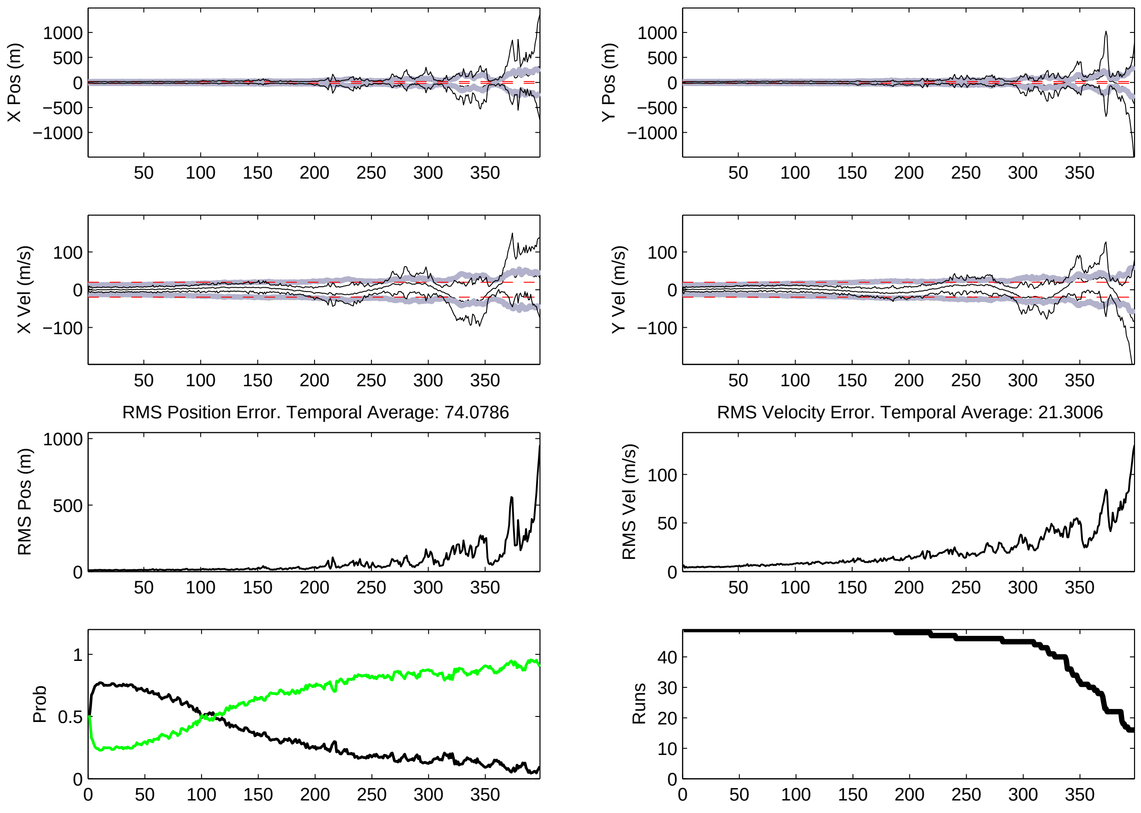


Figure 4.19: Macromixture IMM Filter: Loop Model in Clutter

Figure 4.19 shows the performance of the macromixture-based IMM. This configuration exhibits total track loss around epoch 250 (as seen in the lower right plot of Figure 4.19) – the same time that other filters begin to rely on the deferred decision process to maintain track. The poor tracking ability is most likely due to distortion of the mixtures caused by the intermixing process. Each epoch, the individual components of the mixtures are blended against the pseudo-states of the other mixtures. This significantly distorts the components, and is devastating with regard to track quality for maneuvering targets in heavy clutter.

Figure 4.20 shows the performance of the macrohypothesis-based MMAE against the broken-loop model. To achieve the same dynamics level as the other loop models, this model has been run for 800 epochs. The mean probability flow plot distinctly shows transfer to the appropriate filter. Again, the flow to the benign state occurs much more slowly than the flow from benign to aggressive. After epoch 300, the probability can flow

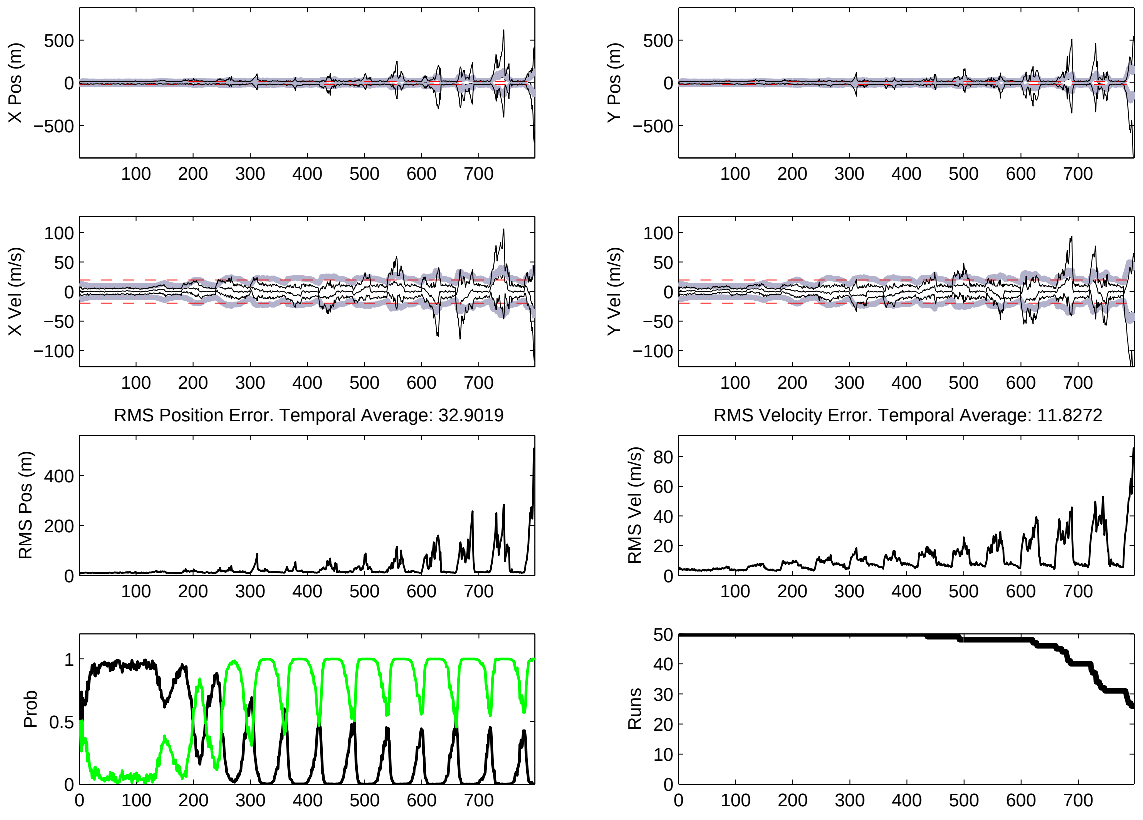


Figure 4.20: Macrohypothesis MMAE Filter: Broken-Loop Model in Clutter

completely to the aggressive filter, but the 30 epoch cycle time is insufficient for flow to return to the benign filter. Deferred decision error, error caused when the majority (with regard to mixture probability) of the mixture components are updated on clutter-originated measurements, is especially visible in the last four maneuver cycles[3], and is displayed as spikes in the RMS errors and error standard deviations and significantly larger than average filter computed covariances.

Figure 4.21 shows the performance of the macrohypothesis-based IMM against the broken-loop model. Similar to the previous model, the IMM can correctly identify the maneuvering and non-maneuvering dynamics cycles, and the clutter-induced error increases dramatically as the aggressive filter becomes mismatched to the ever-increasing dynamics.

[3]Since the dynamics are gradually increasing, these kinds of errors do not manifest themselves until the later cycles

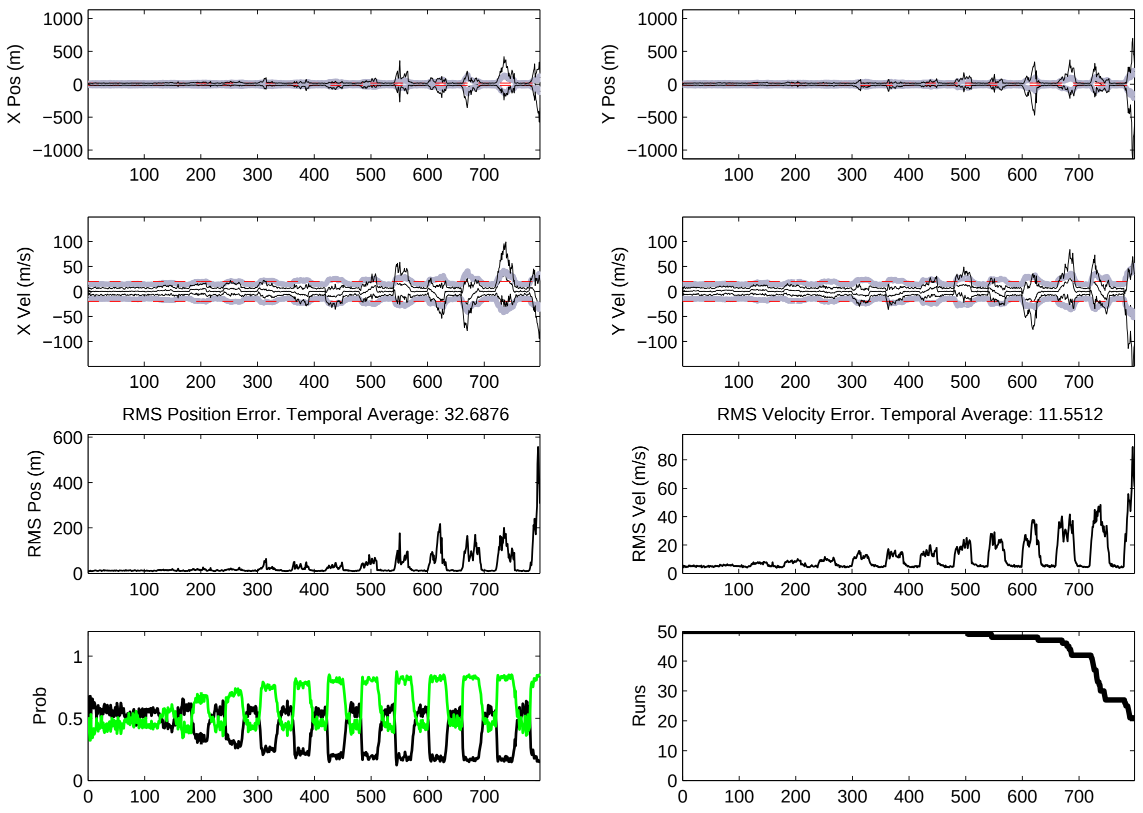


Figure 4.21: Macrohypothesis IMM Filter: Broken-Loop Model in Clutter

Interestingly, the probability flow for the IMM model is rapid in both directions, and the flow makes complete transitions in filter favor within the 30 epoch cycle time.

Figure 4.22 shows the performance of the macromixture-based MMAE against the broken-loop model. Again, the filter correctly identified the appropriate filter for the true dynamics mode and was able to complete the transitions from benign to aggressive within the 30 epoch cycle time, whereas the macrohypothesis MMAE version of the filter could only attain 50% transitions in the same time. This implies that filter flow is nearly twice as fast from aggressive to benign in this case as in the previous case.

Figure 4.23 shows the performance of the macromixture-based IMM against the broken-loop model. The filter correctly identifies the appropriate target maneuver mode, as can be seen in the mean probability flow. The transitions from mode to mode are more rapid than the MMAE version of this filter, as would be expected based on the results of Figures 4.20 and 4.21. Note the precipitous loss of contributing runs near the

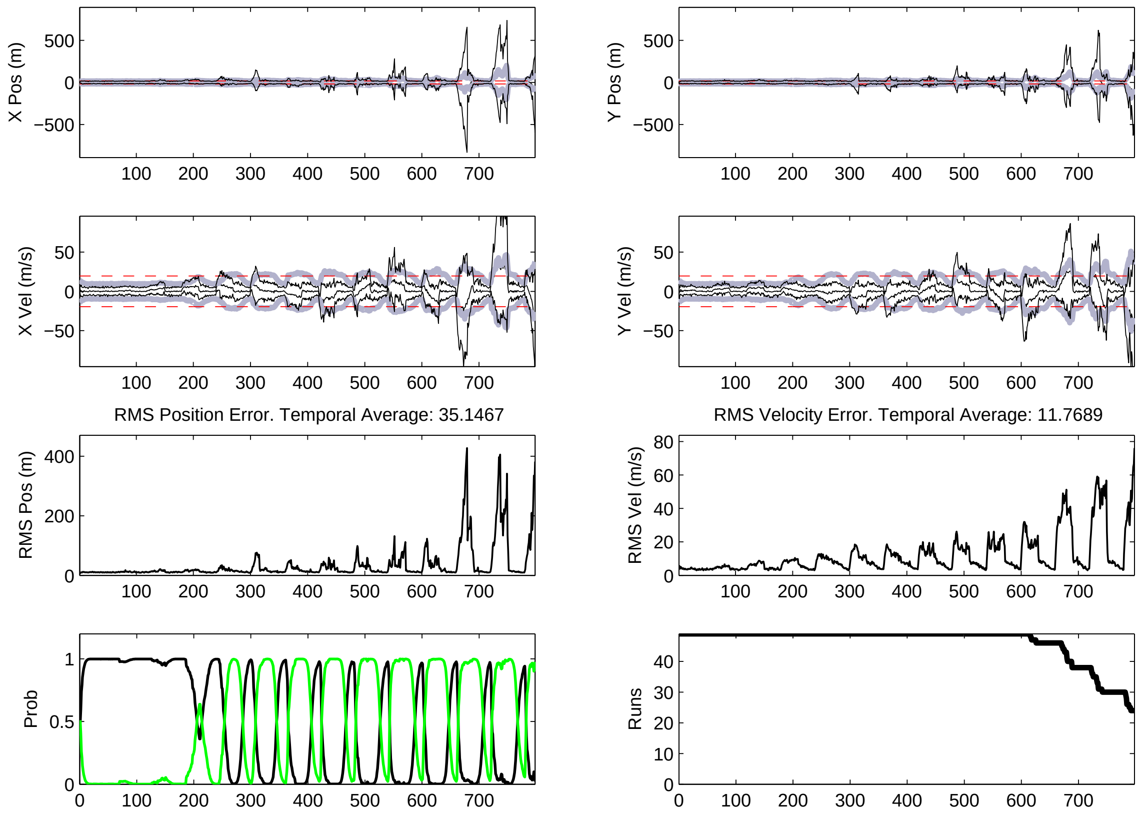


Figure 4.22: Macromixture MMAE Filter: Broken-Loop Model in Clutter

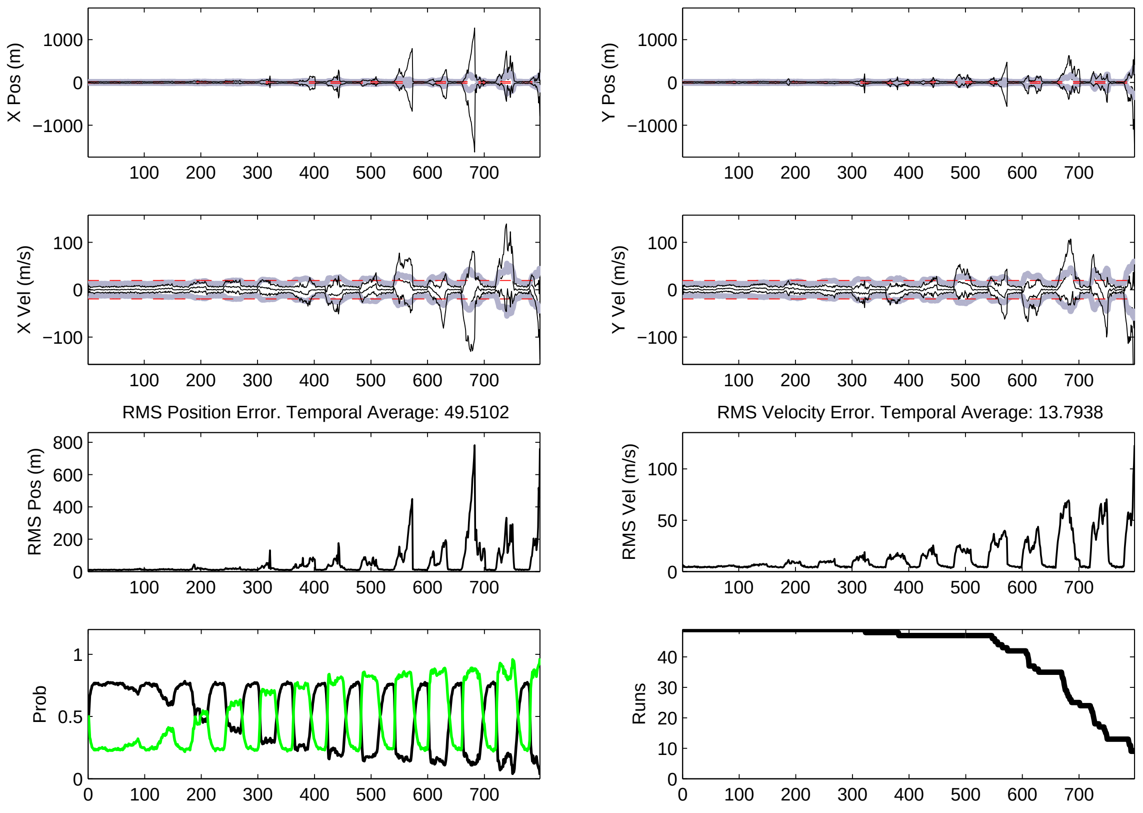


Figure 4.23: Macromixture IMM Filter: Broken-Loop Model in Clutter

end. This configuration had the worst average track life of the available configurations. Additionally, this configuration has the worst deferred decision process errors, with RMS error magnitudes peaking to twice that of other configurations.

4.3.3 Loop Model Summary. The two varieties of loop models are presented here using the constant velocity models. The simulations show that, within a multiple hypothesis structure, a multiple model architecture can correctly identify the true dynamics at a given time, even in the presence of clutter and maneuvers, and create an estimate of the target state by blending the estimates of two or more elemental filters. During the cluttered scenarios, the loop models were run for sufficient length to cause the majority of the runs to lose track, indicating that the ISE-based MRA had exhausted its capability. In clutter, the MHT exhibited two regimes of operation: tracking with a well-matched filter that produced acceptable results (the multiple model versions split this regime between their elemental filters) and tracking with a mismatched filter that leads to gross errors due to the deferred decision making process. Within the matched regime, the multiple model techniques reduced RMS errors, compared to an equivalent single filter design (equivalent in that it can track the same target dynamics without track loss), because they insured that the appropriate filter models were in force. Outside that regime, the multiple model techniques seemed to magnify the deferred decision errors, most likely due to their benign filters forming favorable residuals from clutter-originated measurements.

4.4 Flight Data

The following simulations were performed using recorded flight data from an F-16 flight (see Section 3.3.3) and a filter bank composed of constant acceleration filters. All simulations use the exact same flight data, which was selected because it displayed distinct periods of maneuver and benign flight.

4.4.1 Clutter Free. Figure 4.24 shows the results of the recorded F-16 flight path shown in Figure 3.4 using a constant-acceleration (CA) filter tuned for benign ($.5m/s^3$; see Section 3.6.2) dynamics. As expected, the aircraft displays distinct periods of benign and aggressive flight. The filter performs well for the benign periods for which it is tuned, but performs poorly for aggressive periods, which can be observed as offsets in the Y positions and velocities around epochs 50-100 and 200-250. Also, note that the majority of highly dynamic maneuvers are seen in the vertical (Y) direction. The aircraft does perform maneuvers in the X direction, but with the exception of the maneuver at epoch 250, they are adequately covered by the benign filter. The separation of dynamics level in the axes is addressed more fully in Section 4.5.

Figure 4.25 shows the filter tuned for more aggressive ($4m/s^3$) dynamics against the F-16 flight data. The filter-computed standard deviations and the actual standard deviations are very well matched, and the position and velocity means show almost no offset, even during the periods of maneuver (epochs 50-100 and 200-250). This filter adequately handles all of the maneuver dynamics, but offers only marginal noise rejection.

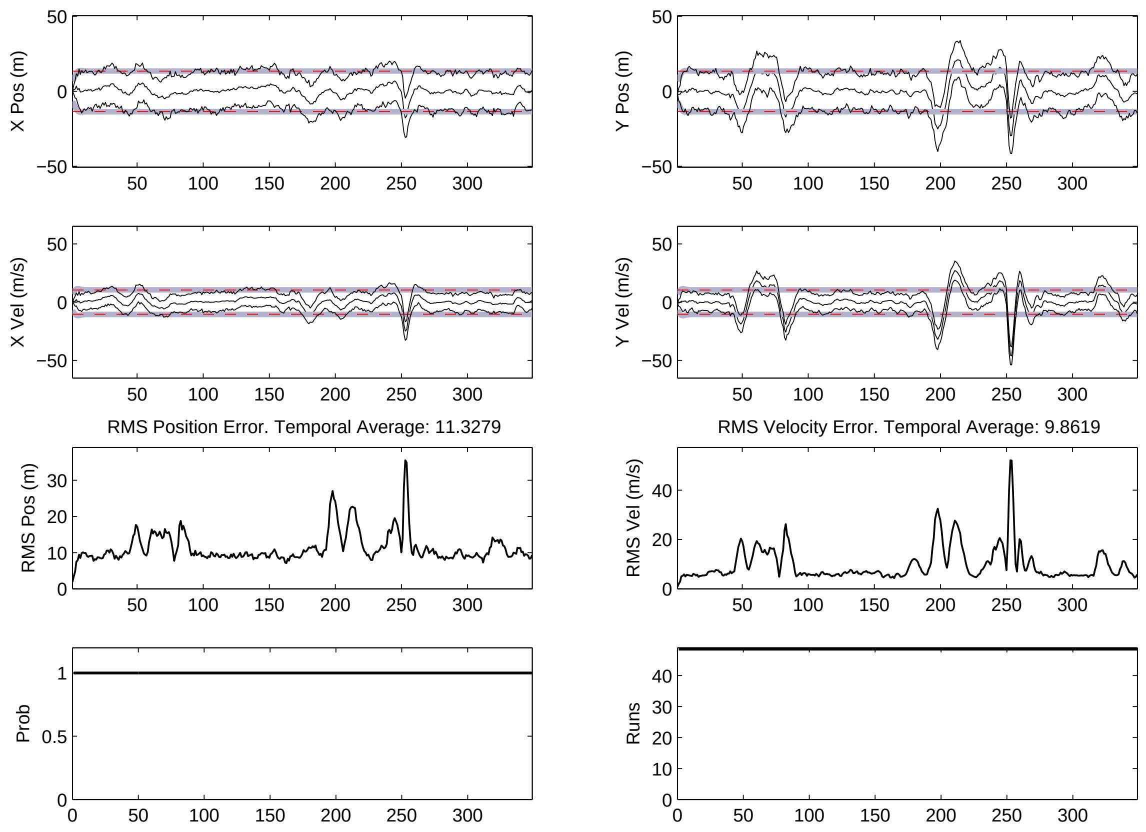


Figure 4.24: Benign CA Filter: F-16 Without Clutter

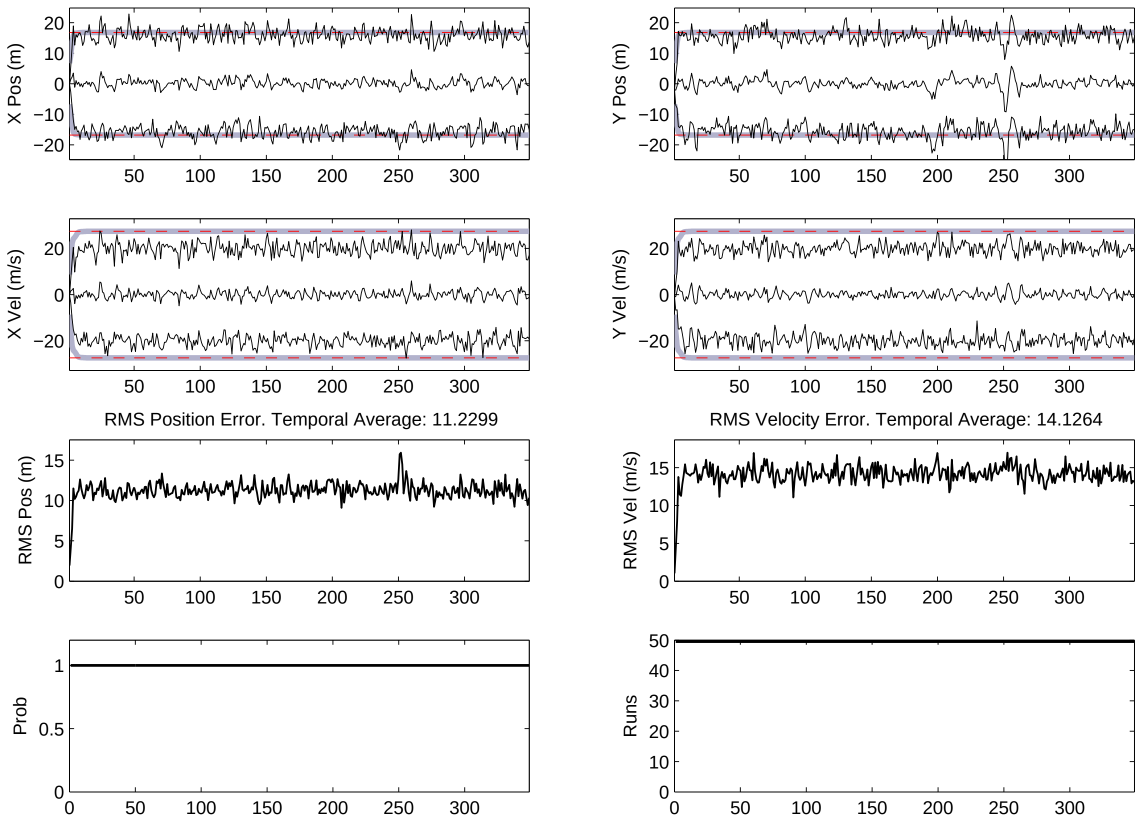


Figure 4.25: Aggressive CA Filter: F-16 Without Clutter

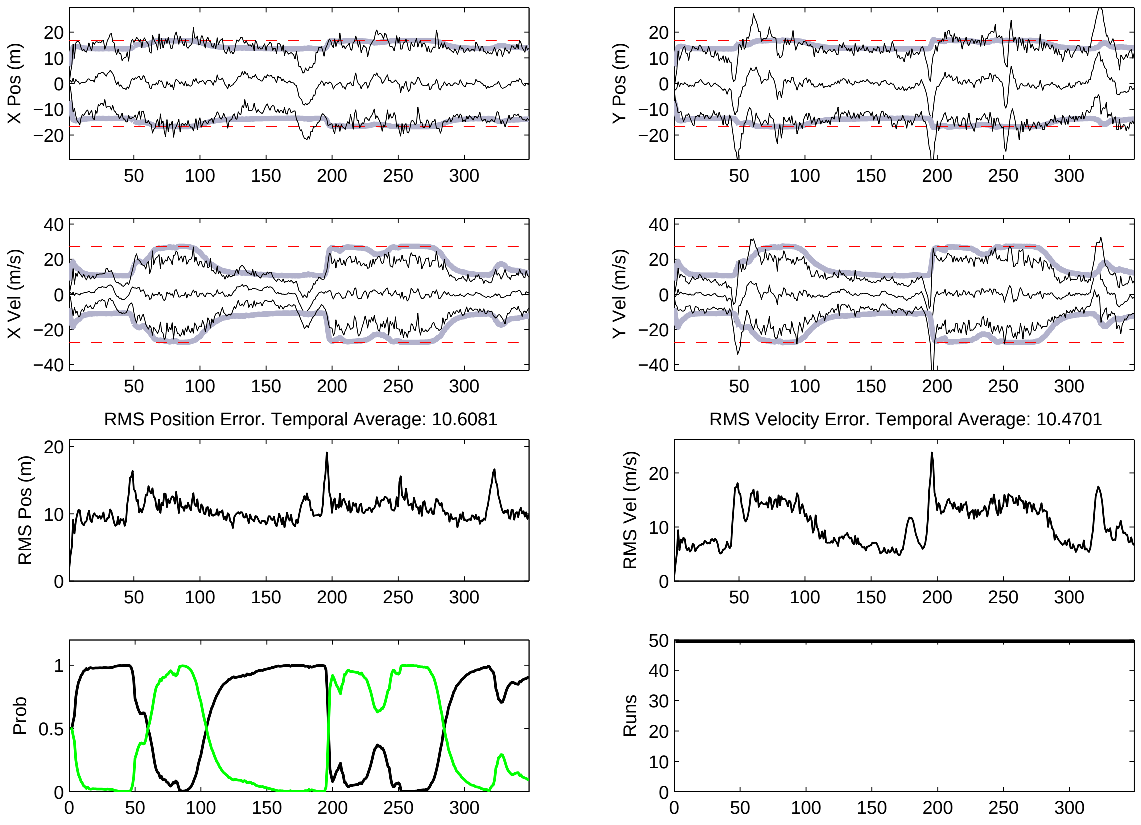


Figure 4.26: MMAE CA Filter: F-16 Without Clutter

The combined performance of the two previous filters in an MMAE configuration is shown in Figure 4.26. In the lower left plot, and in Figure 4.27, notice the distinct probability flow between the filters during maneuvers.

The F-16 performance against an IMM filter composed of the benign and aggressive filters is shown in Figure 4.28. As in all of the previous IMM cases, the probability flow is non-decisive, but the maneuvers are still apparent, especially the aggressive maneuver at epoch 250.

4.4.2 Clutter. The following figures use the same filter tunings used in Section 4.4.1, but now the simulations include Poisson distributed clutter points at a density of $\lambda = .0001\, pts/m^2$, yielding 400 expected clutter points per epoch. Again, this clutter density was chosen so that track loss would occur in approximately 10% of the runs, indicating that the algorithm is running near its full performance capability.

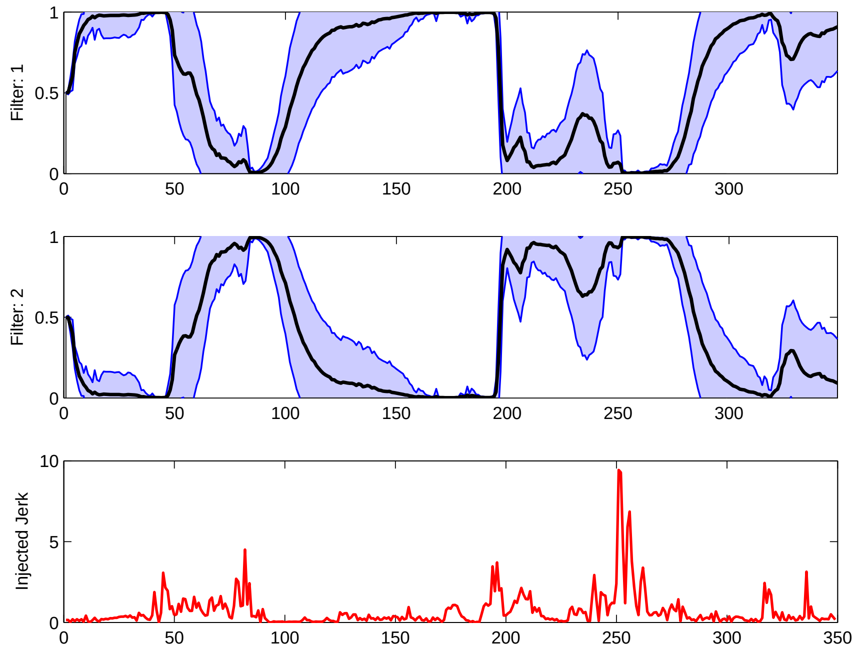


Figure 4.27: MMAE CA Filter: F-16 Without Clutter

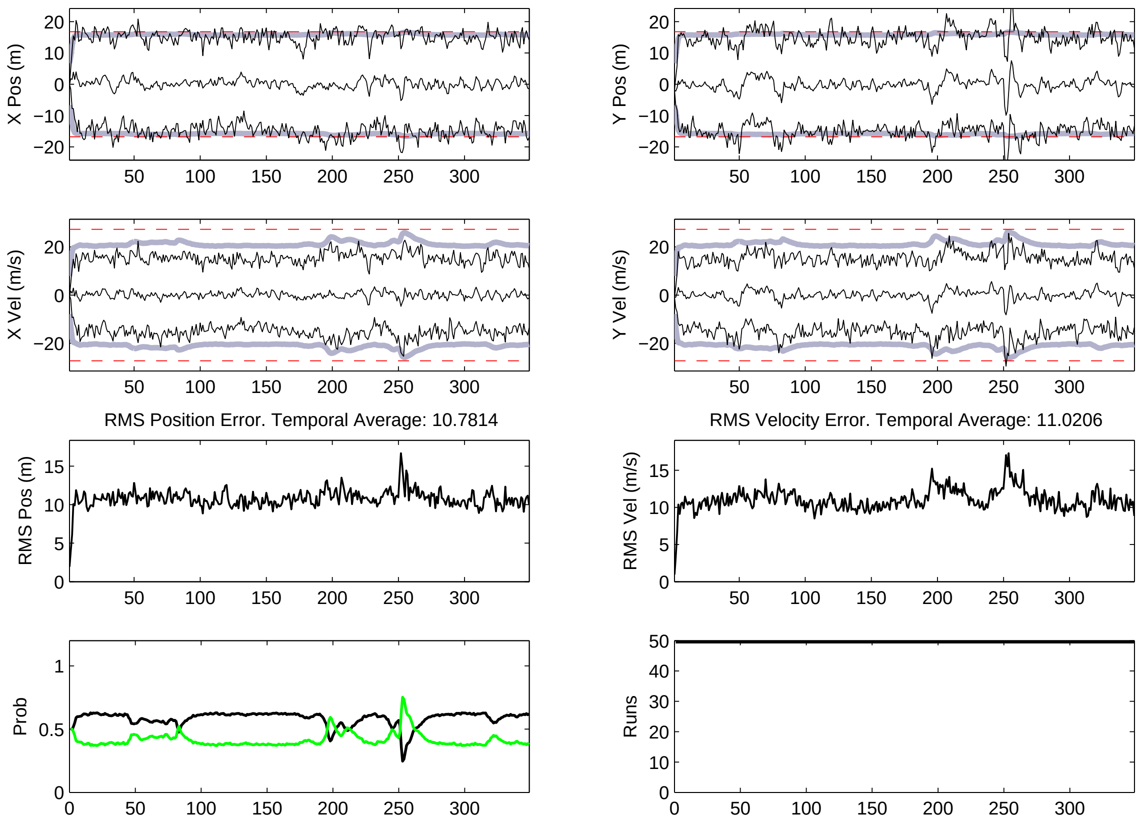


Figure 4.28: IMM CA Filter: F-16 Without Clutter

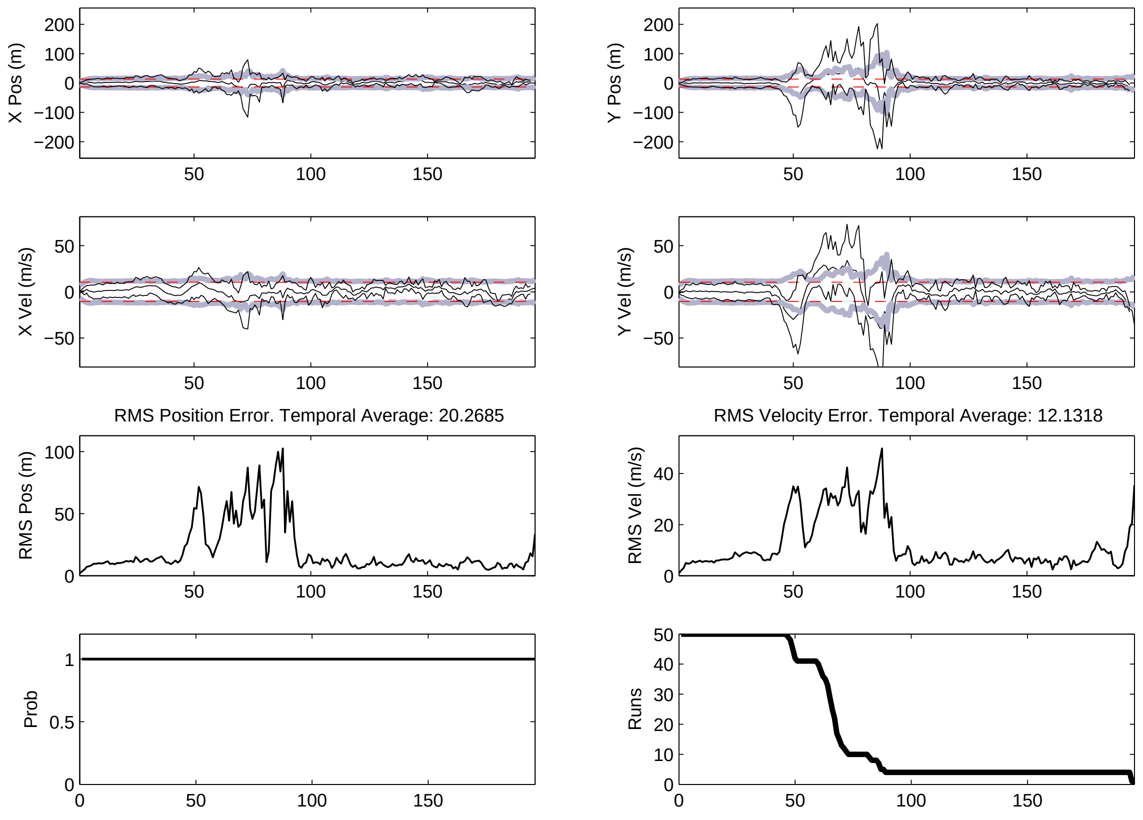


Figure 4.29: Benign CA Filter: F-16 in Clutter

Figure 4.29 shows the performance of the benign filter in clutter. As expected, total track loss occurs during the first significant maneuver. As before, in the periods before track loss in which the benign filter is a poor match for the target dynamics, significant RMS errors are incurred from the deferred decision making process.

Figure 4.30 shows the aggressively tuned filter. As shown in the track loss plot, the goal of 10% track loss has been achieved, indicating that the mixture fidelity is the minimum necessary for this clutter density. The filter is generally well tuned for the dynamics, with a notable exception at epoch 250, where there is a distinct spike in the RMS errors. Note that this is the same filter (CA based on assumed dynamics of $4m/s^3$) as found in Figure 4.25, which did not display an error of this magnitude during the maneuver. Apparently, the filter is being deceived by clutter during the maneuver and flying off track, thereby incurring a significant RMS error penalty. Attempts to alleviate this problem by doubling and even tripling the filter's process noise strength resulted in even poorer

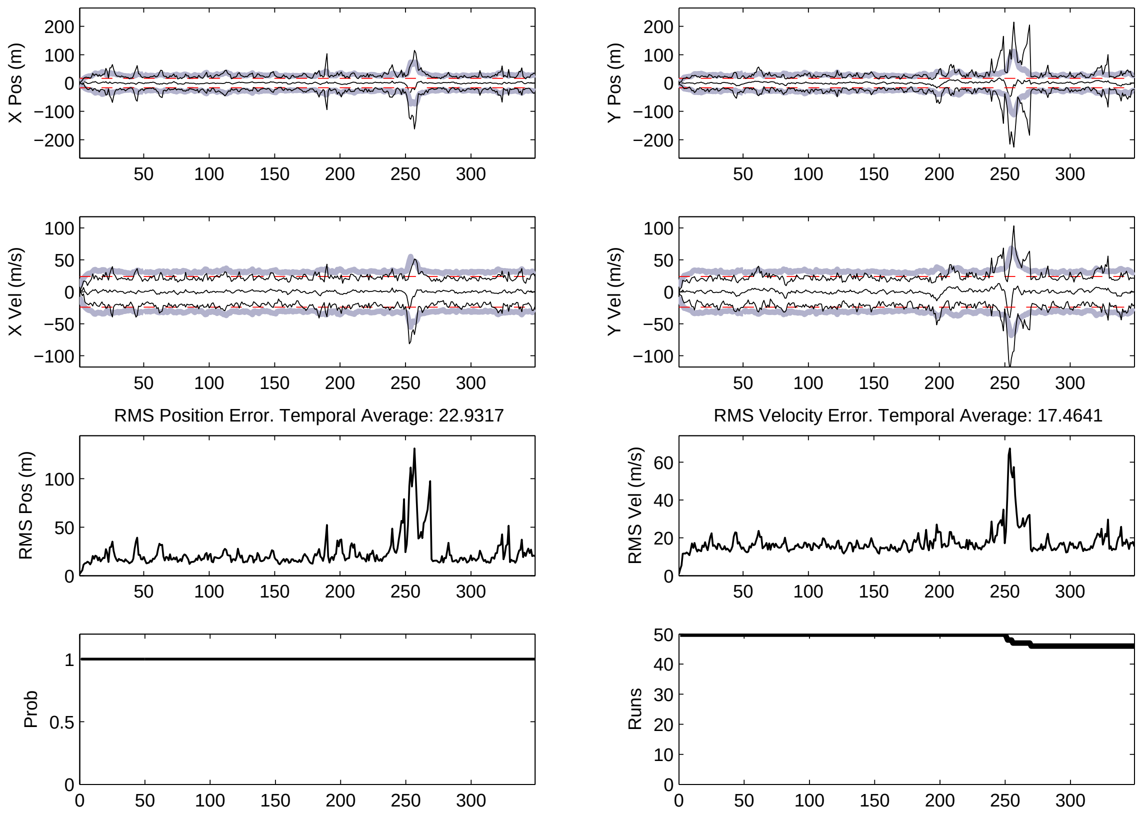


Figure 4.30: Aggressive CA Filter: F-16 in Clutter

performance[4]. Effectively, the clutter density is high enough that, during this maneuver, random clutter points will always be closer to the propagated position than truth. The only way of avoiding the error would be to reduce the clutter density or increase the sample rate. This implies that, for a given clutter density and sample rate, there exists a level of maneuver above which a filter cannot be tuned to avoid the overwhelming possibility of a deferred decision error.

Figure 4.31 shows the macrohypothesis MMAE configuration that allows 15 Gaussian components to exit the reduction cycle at each epoch. From the lower right plot, the filter maintains track with the majority (86%) of the runs, losing the majority during the harsh maneuver at epoch 250, displaying that the MMAE structure has not negatively impacted the clutter rejection capability of the filter. The lower left plot, expanded in Figure 4.32,

[4]The fact that increasing the process noise on the filter provided no benefit is not surprising because the filter is tuned appropriately for the maneuver.

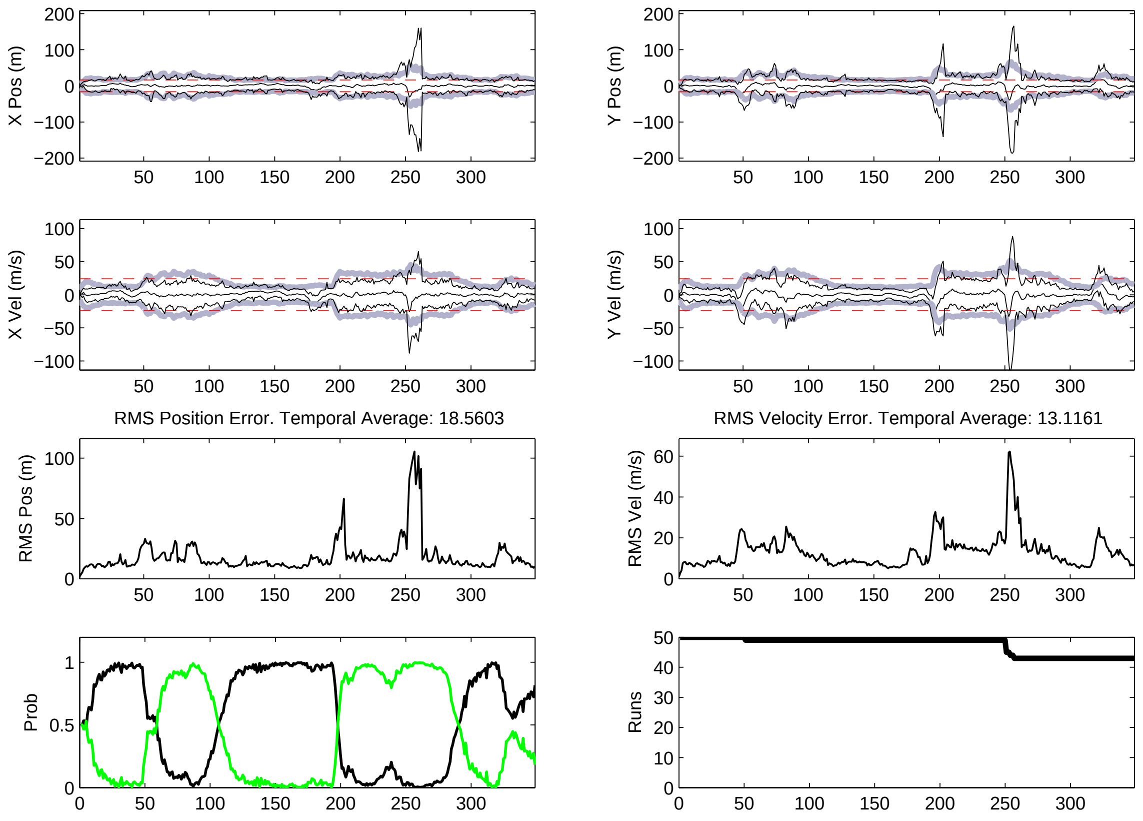


Figure 4.31: Macrohypothesis MMAE CA Filter: F-16 in Clutter

shows that transitions from benign to aggressive occur swiftly at the appropriate times (note the transition at epoch 200) while the transitions from aggressive to benign are much slower (note the gradual transition from epoch 90-125). Also note from Figure 4.31 the magnitude of the RMS errors at epoch 250 – nearly twice those of any other period. Coincidentally, the mean filter probability is still aggressive from the maneuver at epoch 200 prior to the onset of the maneuver at epoch 250. Section 4.5 discusses methods that attempt to return probability to the benign filters more quickly – in this particular situation, those methods impact performance because they will have directed probability back to the benign case, meaning that the probability will have to flow back to the aggressive filter at the onset of the second maneuver, whereas in the case without the probability marshalling, the probability would already be on the aggressive filter from the previous maneuver.

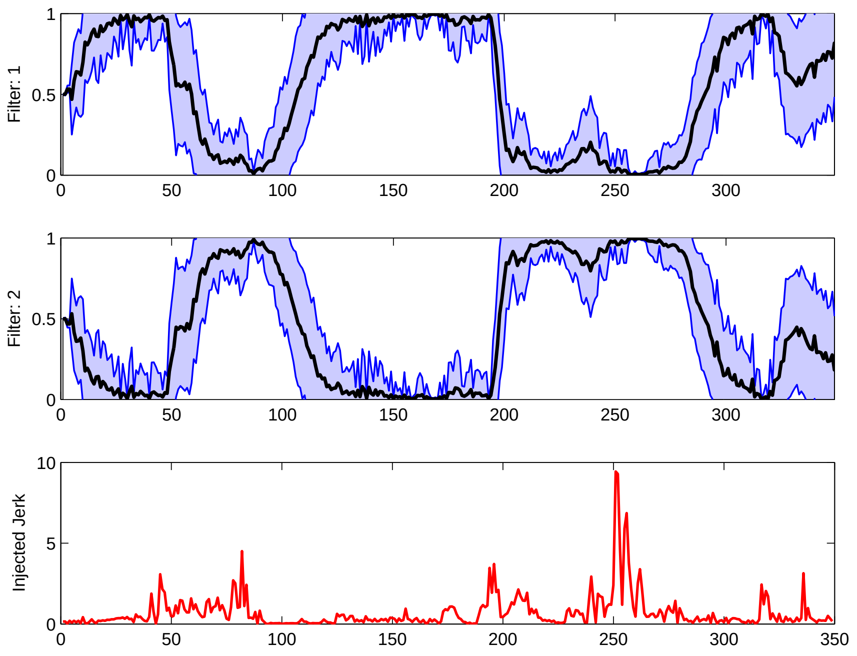
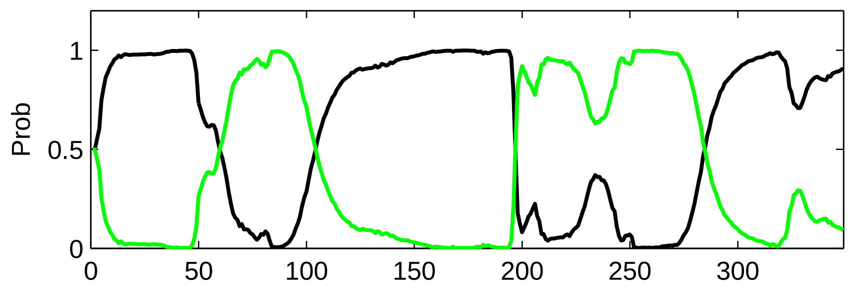


Figure 4.32: Macrohypothesis MMAE CA Filter: F-16 in Clutter Probability Flow

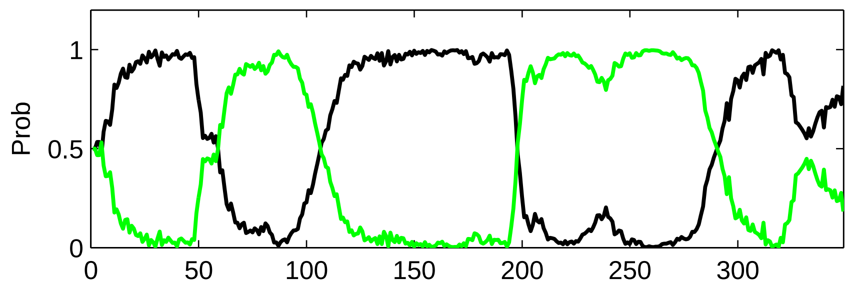
Figure 4.32 shows the mean $\pm 1\sigma$ plots for the mean elemental filter probability flow in the macrohypothesis MMAE filter case. Figure 4.33 compare the clutter-free results directly to the cluttered case, and the mean flow is nearly identical. In fact, the only difference is the increased magnitude of the standard deviation, caused by the clutter corruption and explained in the discussion of Figure 4.14.

Figure 4.34 shows the performance of the macrohypothesis IMM filter against the F-16 in clutter. As in the previous case, the multiple model structure can identify the appropriate filter for the observed target dynamics. This filter also displays the characteristic deferred decision errors during the harsh maneuvers – more so than the macrohypothesis MMAE version.

Figure 4.35 shows the performance of the macromixture MMAE against the F-16 in clutter. Note that the mean elemental filter flow is less decisive than in the macrohypothesis versions of the tracker, but the filter still correctly identifies appropriate truth dynamics model.



(a) Macrohypothesis MMAE Mean Probability Flow Without Clutter



(b) Macrohypothesis MMAE Mean Probability Flow with Clutter

Figure 4.33: The Effect of Clutter on Mean Elemental Filter Probability Flow

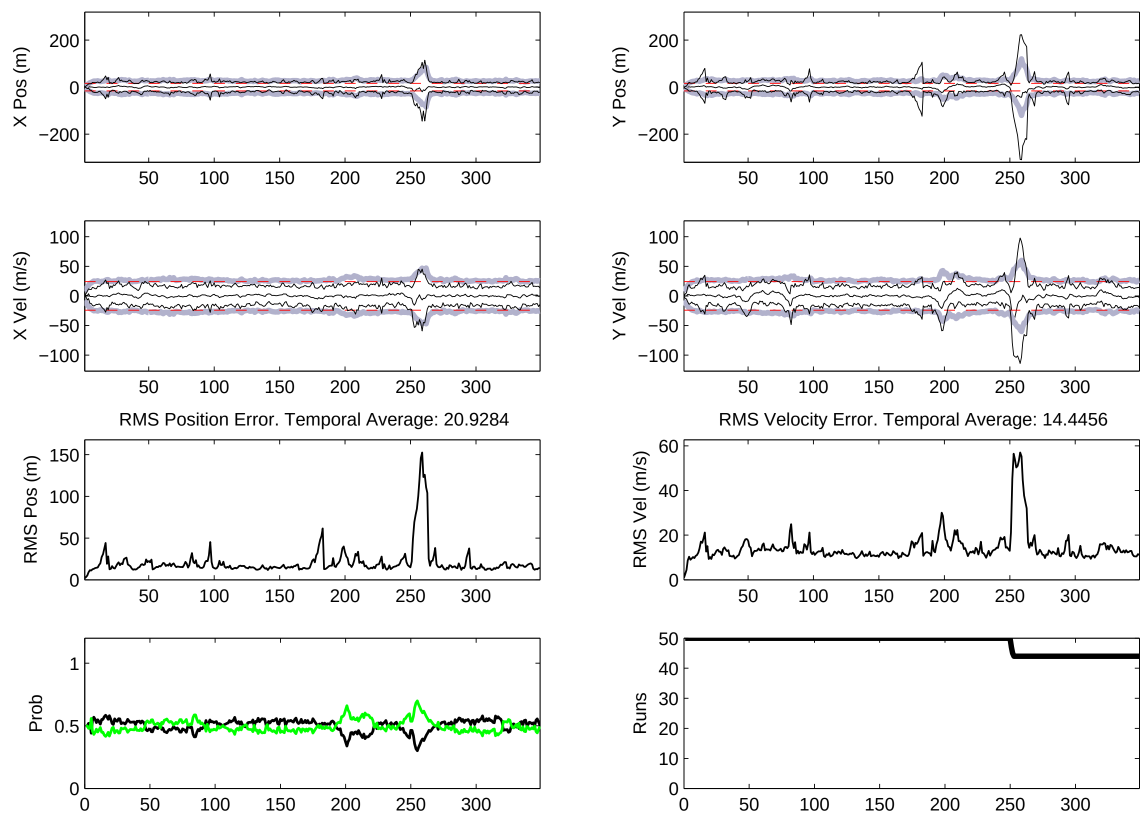


Figure 4.34: Macrohypothesis-based IMM CA Filter: F-16 in Clutter

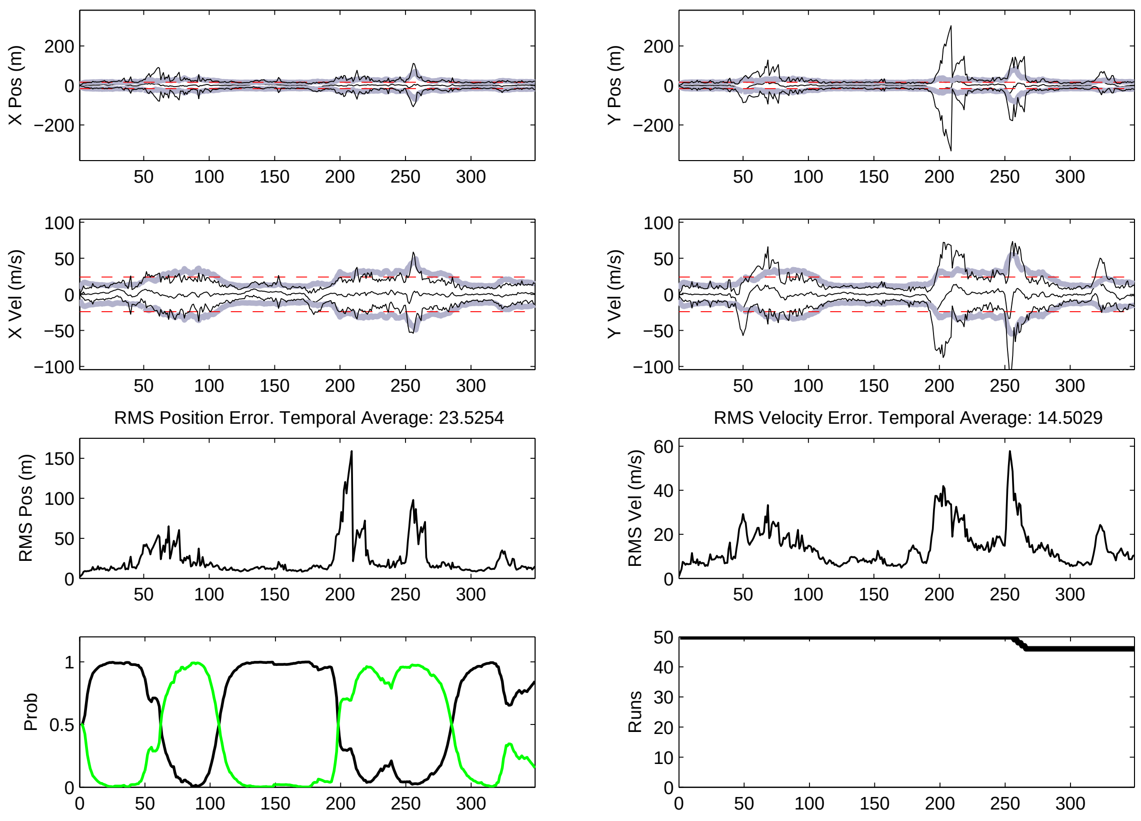


Figure 4.35: Macromixture-based MMAE CA Filter: F-16 in Clutter

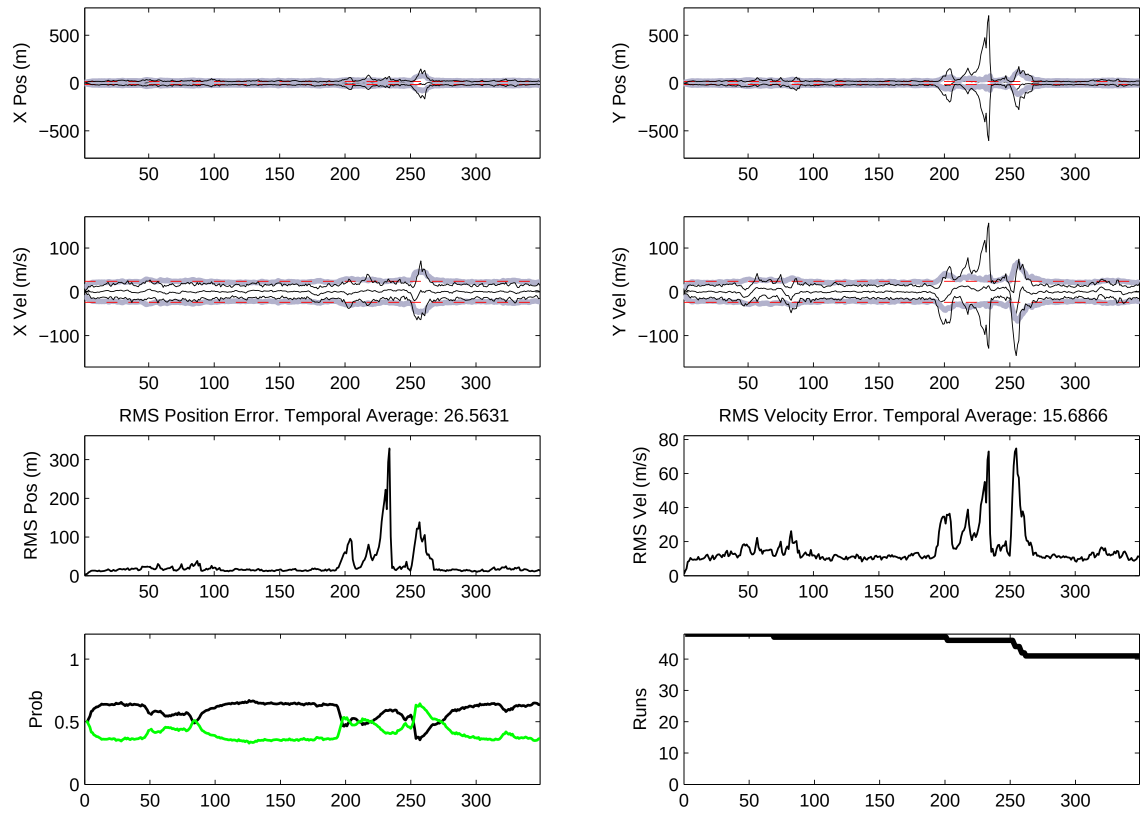
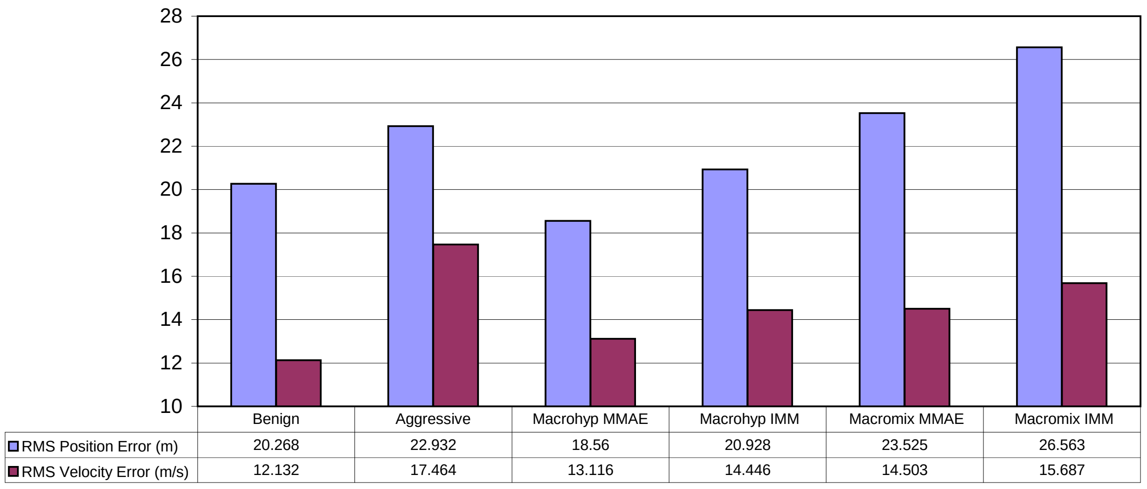


Figure 4.36: Macromixture-based IMM CA Filter: F-16 in Clutter

The macromixture IMM, presented in Figure 4.36, is the worst performing multiple model structure in these runs, but does provide intriguing data. This filter configuration exhibited total failure in the loop model test in clutter in half the time as the other configurations, yet displayed competitive (although still worse than the MMAE version in Figure 4.35) track life in this scenario. Remember that the loop model dynamics were driven well past the level for which the individual filters were tuned and the macromixture IMM only displayed massive track loss after the dynamics exceeded the individual filter tuning levels. In this scenario, the filters were tuned to the observed dynamics, so the tracker was never placed into a mismatched dynamics regime. This would imply that the macromixture form of the IMM severely hinders the ISE deferred decision process, which is easily understood considering that the intermixing process mixes individual hypothesis against the pseudo-states of the other mixtures.



	Benign	Aggressive	Macrohyp MMAE	Macrohyp IMM	Macromix MMAE	Macromix IMM
🟦 RMS Position Error (m)	20.268	22.932	18.56	20.928	23.525	26.563
🟥 RMS Velocity Error (m/s)	12.132	17.464	13.116	14.446	14.503	15.687

Figure 4.37: F-16 Filter Performance in Clutter. Position (blue) and velocity (purple) RMS error temporal average for various multiple model configurations.

4.4.3 F-16 Summary. Figure 4.37 shows a comparison of the temporally averaged RMS errors in position and velocity for the four varieties of multiple model filters and the aggressive and benign filters of which they were composed. Remember that the benign filter exhibited total track loss, so the values on the chart should be viewed as the lowest errors possible. Note that the velocity errors for the multiple model structures have improved over the results for the non-adaptive aggressive filtering, and that position states, with the exception of those of the macromixture IMM, have also improved. The macrohypothesis MMAE is the best performer in this case and has performance closer to the unattainable benign values than to the aggressive filter. Referring to the previous figures for the multiple model structures, it is evident that the harsh maneuver at epochs 200 and 250 have caused deferred decision errors in all of the multiple model filters. The aggressive filter has also accrued some error from these maneuvers, but not on the same order, yet the multiple model filters still outperform the aggressive filter. This implies that the performance during the benign period was even better than is being portrayed in the chart.

4.5 Probability Marshalling

The configurations in this section were designed to test the capability of *ad hoc* techniques of speeding the return of multiple model probability to a benign state after a maneuver. The specific techniques tested were splitting the filter axes, Korn and Beean restarts, and the dimpled Gaussian filter.

4.5.1 Decoupled Multiple Model Axes. Figure 4.38 shows the results of a simulation using the recorded F-16 data and an MMAE without clutter, with decoupled elemental filter probabilities as described in Section 3.9. The mean elemental filter flow in the lower left plot still shows the flow as though it were calculated normally, but the internal elemental filter flows have been separated by axis, and the flow can be easily seen in the computed filter covariances shown in the position and velocity error plots. Notice that the Y axis elemental filter probability transfers to a more aggressive filter from epochs 50-125, but the X axis remains matched to the benign filter. This can be seen in the movie *SplitMMAE.avi* and in Figure 4.39, in which the aircraft enters a vertical climb, but does not significantly accelerate in the X axis – the maneuver is contained entirely in the Y axis. Figure 4.39 shows 5 epochs of an MMAE running with decoupled axes. The propagated covariances are represented as gray ellipses (with shading getting lighter with each previous epoch), and the vertical elongation indicates that the Y-axis is based on more aggressive filter dynamics than the X-axis. The benefit of this technique is minimal but still observable in this case. The additional filtering of one state does provide a slight decrease in the RMS errors, especially in the velocity state.

4.5.2 Broken-Loop Model Probability Marshalling. The broken-loop dynamics model was specifically designed to test elemental filter probability flow rates. Consequently, it is used here to demonstrate efficacy of the various probability marshalling techniques. The runs were truncated at 500 epochs because the probability flow becomes repetitive after epoch 400. Figure 4.40 shows the mean elemental filter flow of four variants of the macrohypothesis MMAE without clutter. The standard mean probability flow is provided as a reference, and it shows the uneven transition rates of the filter. Note that the aggressive filter (shown in green) is in force for longer than the benign filter (shown in black) even though the truth model spends an equal amount of time in either state. Figure 40(b) shows the Korn & Beean resets. Note that resets occasionally occur during an aggressive state and the mean filter probability quickly returns to the aggressive filter. Figure 40(c) shows

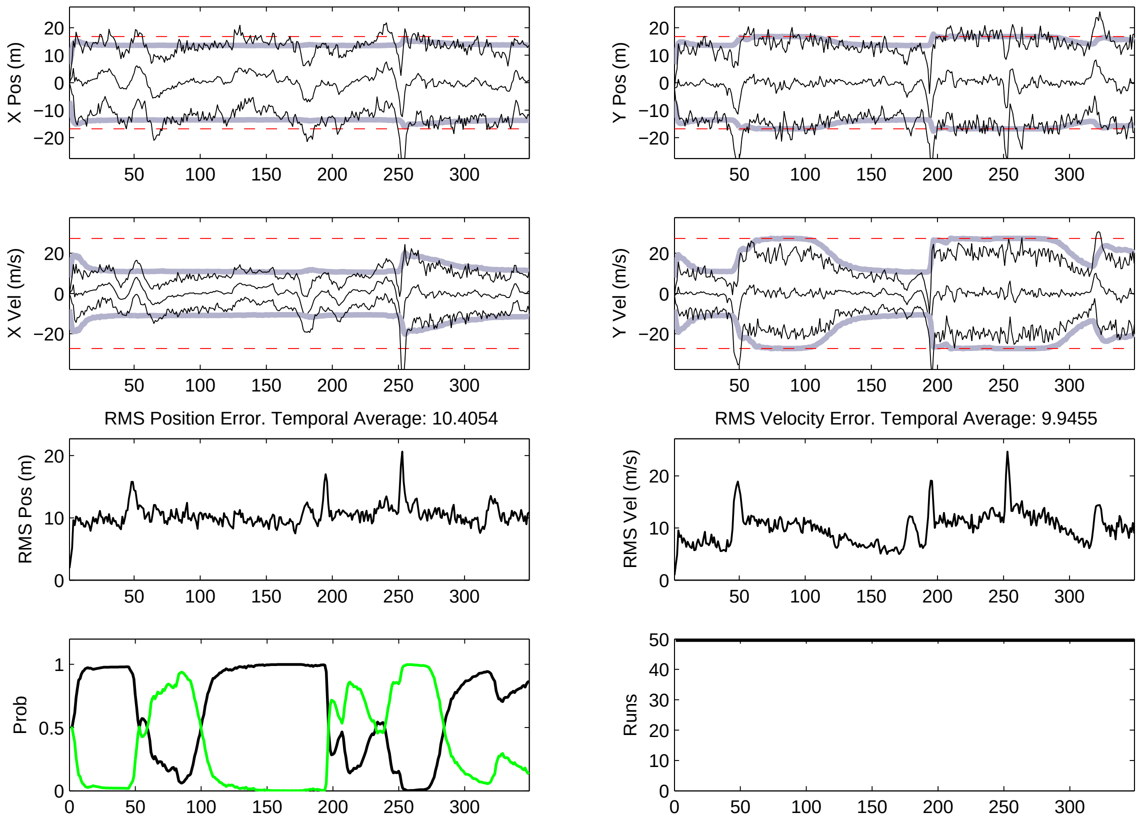


Figure 4.38: F-16 CA MMAE Filter: Separate Axes Without Clutter

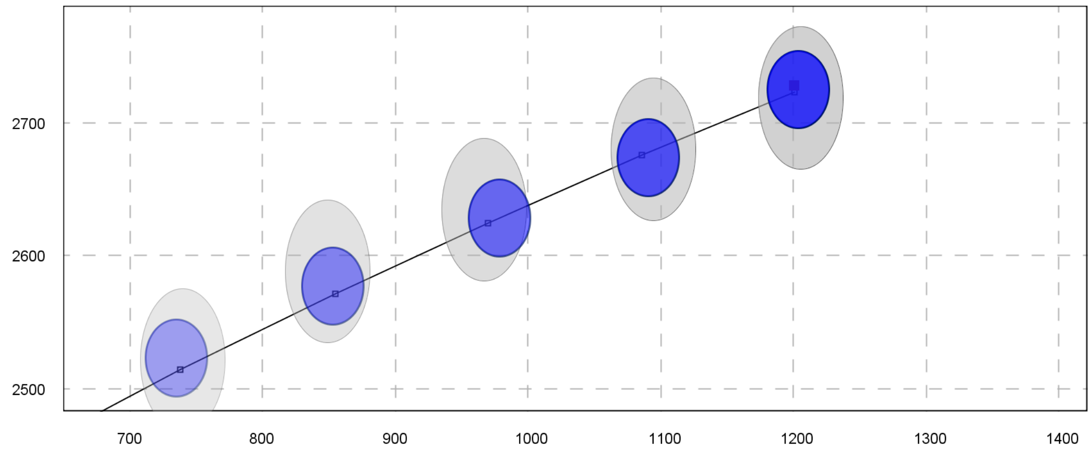


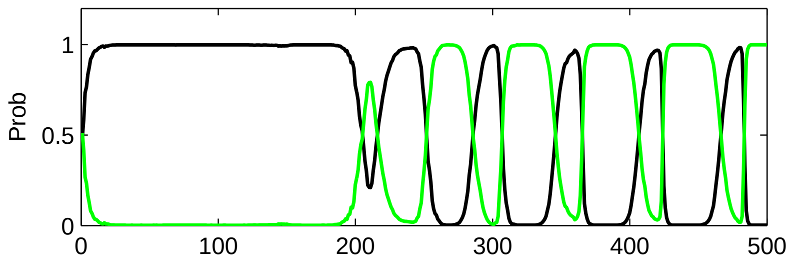
Figure 4.39: F-16 CA MMAE Filter: Separate Axis Without Clutter. The target has recently entered a vertical climb. The propagated covariances are represented as gray ellipses (with shading getting lighter with each previous epoch). Notice that the propagated covariances, which are a good indicator of the assumed dynamics of the filter in force, are elongated in the Y axis. This indicates that the Y axis is in a more aggressive mode.

the dimpled Gaussian flow with $\alpha = .1$ (refer to Equation (3.6)). Here the transfer to the benign state is rapid, but flow towards the aggressive filter has been severely hampered. Figure 40(d) shows the dimpled Gaussian with $\alpha = .5$. In this case, the transitions are rapid in both directions. The two dimpled cases show that the α tuning parameter should be selected carefully, as it can have a negative effect. Additionally, note that the dimpled Gaussian has absorbed the partial transition from benign to aggressive that occurs around epoch 210. At this time period, the filters are both a relatively close match to the target dynamics and the measurements are near the PDF crossover point of the Gaussians. The dimpled Gaussian creates a jump discontinuity in the PDF of the aggressive filter at this point, so the measurements that would normally have carried a higher aggressive PDF are artificially forced to the benign filter.

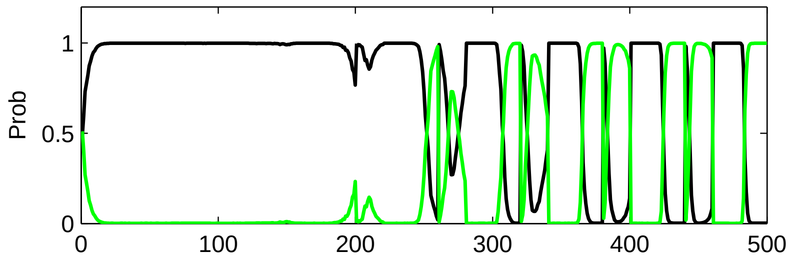
The RMS error values for the loop without clutter are shown in Figure 4.41. The RMS errors behave as expected, the Korn & Beean method reduces velocity errors, but has higher position error, most likely due to errors caused by probability restarts at inopportune (resetting an appropriately matched filter) times. The dimpled Gaussian techniques provide a definite reduction in error.

Figure 4.42 shows modified probability marshalling techniques against the loop models in clutter. Notice the addition of the modified Korn & Beean reset filter (Figure 42(c)), described in Section 3.9, that analyzes the mixture to determine whether a probability restart should be performed. Note that the modified Korn & Beean filter successfully identifies the maneuver periods and does not initiate probability restarts during them. The mean probability flow results in clutter are similar to the uncluttered cases, indicating that the probability marshalling techniques work effectively in the MHT structure. Like the modified Korn & Beean filter, the dimpled Gaussian does not distort the transitions, and has a faster transfer from aggressive to benign than the other filters.

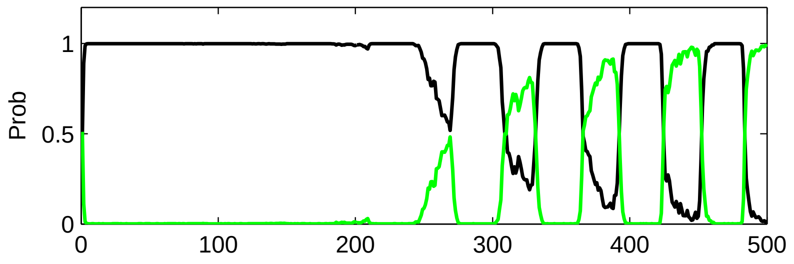
The RMS error values for the cluttered broken-loop are shown in Figure 4.43. Surprisingly, the dimpled Gaussian filter, which displayed the most desirable probability flow, did not perform the best. In fact, the modified Korn & Beean was the best performer in this case. The dimpled Gaussian filter seems more susceptible to the deferred decision



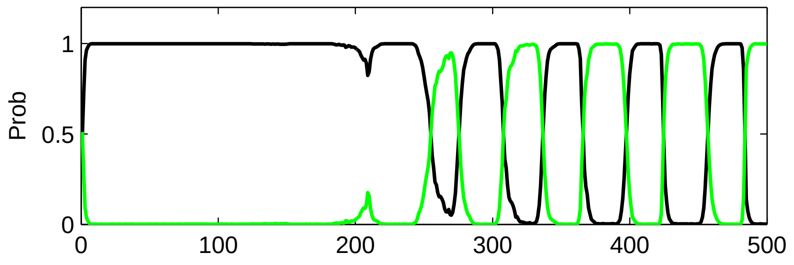
(a) Standard MMAE Flow



(b) Korn & Beean MMAE Flow

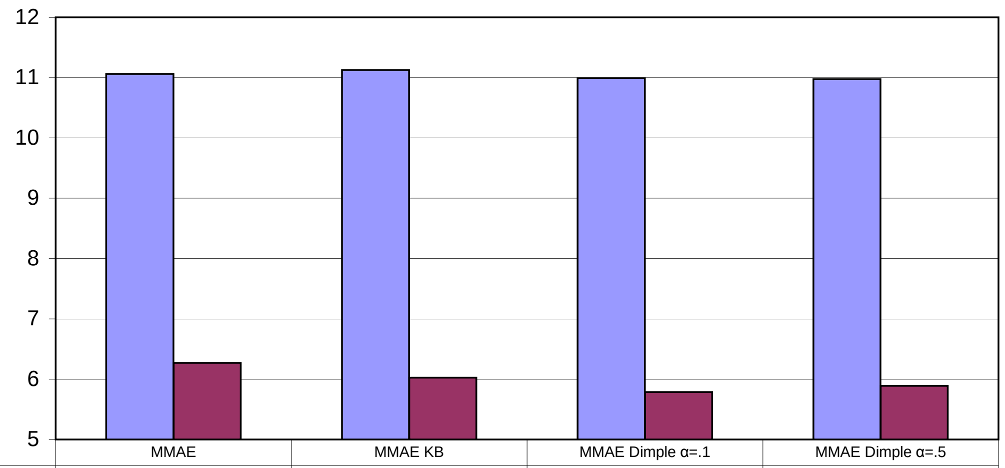


(c) Dimple MMAE Flow $\alpha = .1$



(d) Dimple MMAE Flow $\alpha = .5$

Figure 4.40: Broken-Loop Macrohypothesis MMAE Without Clutter: Probability Flow with and Without Ad Hoc Marshalling Techniques



	MMAE	MMAE KB	MMAE Dimple α=.1	MMAE Dimple α=.5
▨ RMS Position Error (m)	11.06	11.126	10.991	10.974
■ RMS Velocity Error (m/s)	6.2688	6.0241	5.7873	5.8879

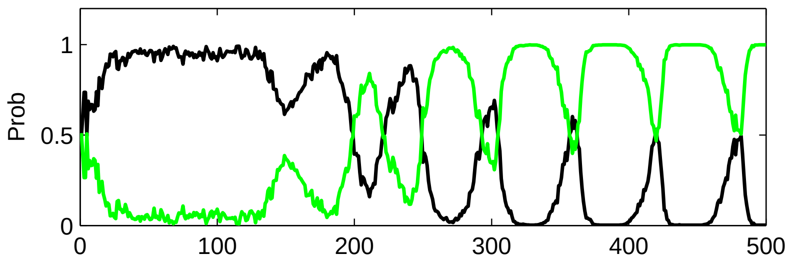
Figure 4.41: CV Broken-Loop Without Clutter: Temporally Averaged RMS Values. Comparison of the standard MMAE, an MMAE using Korn & Beean Restarts (KB), and dimpled Gaussian filters for two α values.

errors than the other filters, most likely because the benign filter will latch onto a clutter point, causing the filter based on that hypothesis to diverge, more readily than before.

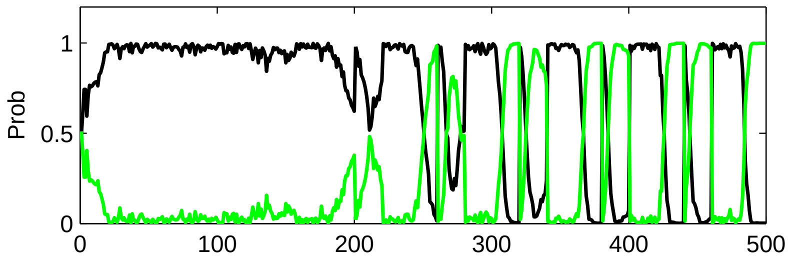
4.5.3 F-16 Probability Marshalling. The Korn and Beean and dimpled Gaussian techniques were tested against the F-16 flight data and the results are compared to the unaltered flow without clutter in Figure 4.44 and with clutter in Figure 4.46.

The RMS error values for the F-16 flight without clutter are shown in Figure 4.45. The split axis technique has been applied to some of the filters here, but was omitted from the loop models – the split axis technique relies on maneuvers occurring in a single axis while the loop models were designed to spread dynamics across axes, so they were omitted before. Note the significant error improvements over the baseline MMAE that all of the techniques provide. Again, the only decrease in performance was the Korn & Beean filter with respect to position. The best performer in this case was the dimpled Gaussian with decoupled axes.

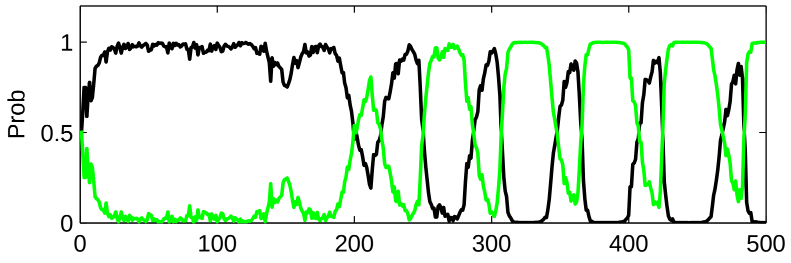
Figure 4.46 shows the mean filter flows for the F-16 in clutter. The filters react as before, demonstrating excellent model mode identification and increased model transition speed. Note that the modified Korn & Beean method does not differ much from the refer-



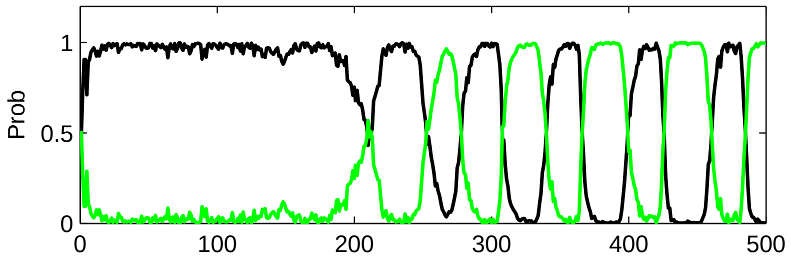
(a) Standard MMAE Flow



(b) Korn & Beean MMAE Flow

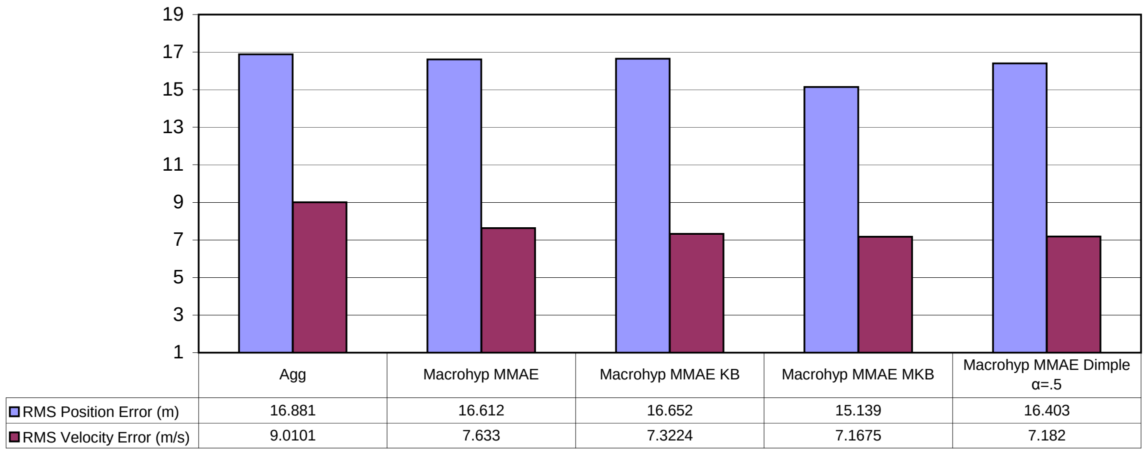


(c) Modified Korn & Beean MMAE Flow



(d) Dimple MMAE Flow $\alpha = .5$

Figure 4.42: Broken-Loop Macrohypothesis MMAE with Clutter: Probability Flow with and Without Ad Hoc Marshalling Techniques



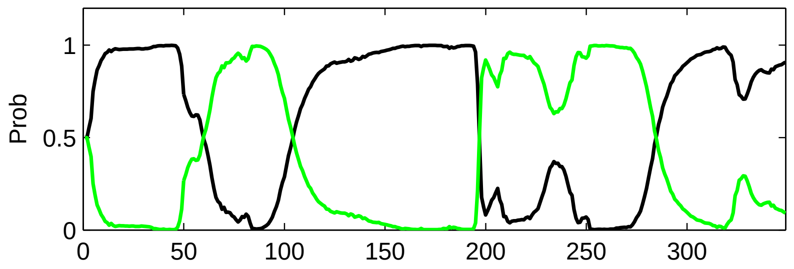
	Agg	Macrohyp MMAE	Macrohyp MMAE KB	Macrohyp MMAE MKB	Macrohyp MMAE Dimple α=.5
🟦 RMS Position Error (m)	16.881	16.612	16.652	15.139	16.403
🟥 RMS Velocity Error (m/s)	9.0101	7.633	7.3224	7.1675	7.182

Figure 4.43: CV Broken-Loop in Clutter: Temporally Averaged RMS Values. Comparison of the standard MMAE, an MMAE using Korn & Beean restarts (KB), Modified Korn & Beean restarts (MKB), and the dimpled Gaussian filter.

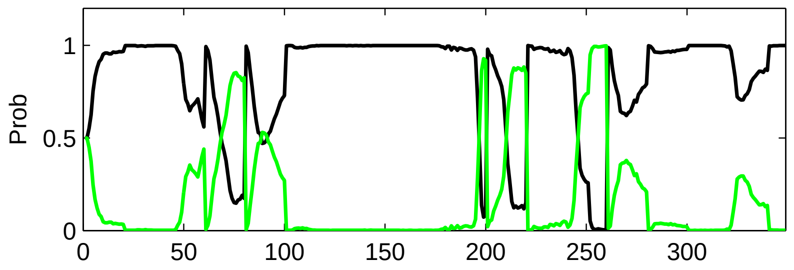
ence MMAE flow. The method tries not to adjust the mixture when it detects spreading in the mixture. In a dynamic model in clutter, this may preclude probability restarts from happening most of the time. Of course, the penalty for remaining in an aggressive state longer than absolutely necessary is minor compared to a benign restart during an aggressive period.

The temporally averaged RMS error values for the F-16 flight with clutter ($\lambda = .0001 pts/m^2$) are shown in Figure 4.47. These plots have additional values, the temporally averaged RMS errors for only the benign period of flight from epochs 100-175. These were included to insure the reader that the puzzlingly large errors in the MMAE position states are due to deferred decision errors during maneuvers. Note that, during the benign regime, every MMAE outperforms the aggressive filer. With regard to velocity error, some are significantly lower than that of the aggressive filter. But, notice that the position errors are much larger in some when considering the entire period for averaging. The dimpled Gaussian provides best performance in the benign regime, but worst overall. Again, the addition of a benign mode to the multiple model MHT will increase its tendency to be deceived by clutter.

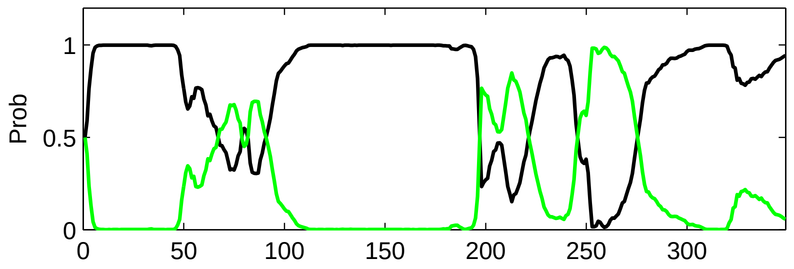
The probability marshalling techniques were only tested on the MMAE multiple model configuration because the intermixing process of the IMM configuration facilitates



(a) Standard MMAE Flow

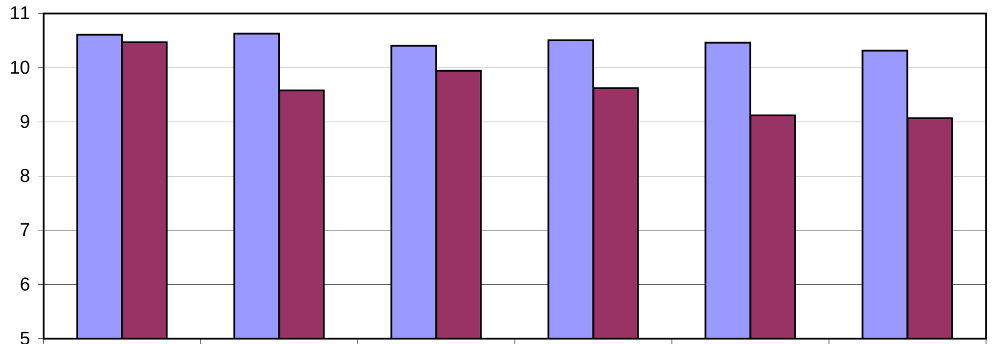


(b) Korn & Beean MMAE Flow



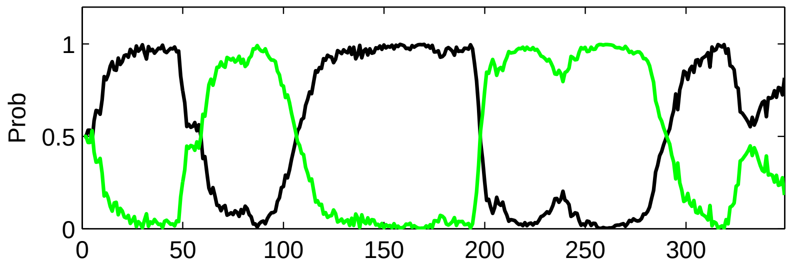
(c) Dimple MMAE Flow

Figure 4.44: F-16 Macrohypothesis MMAE Without Clutter: Probability Flow with and Without Ad Hoc Marshalling Techniques

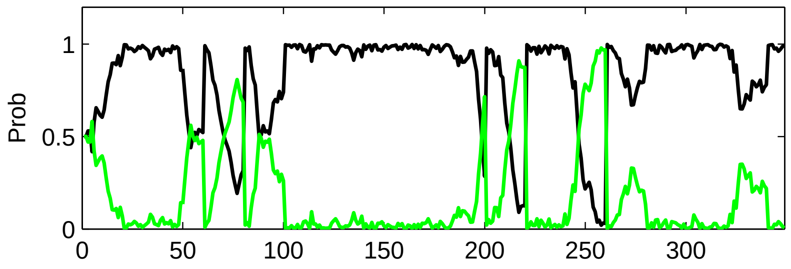


	F-16 MMAE	F-16 MMAE KB	Macrohyp MMAE Split	Macrohyp MMAE Dimple	Macrohyp MMAE Split KB	Macrohyp MMAE Dimple Split
🟦 RMS Position Error (m)	10.608	10.629	10.405	10.508	10.46	10.316
🟥 RMS Velocity Error (m/s)	10.47	9.5815	9.9455	9.6216	9.1215	9.0663

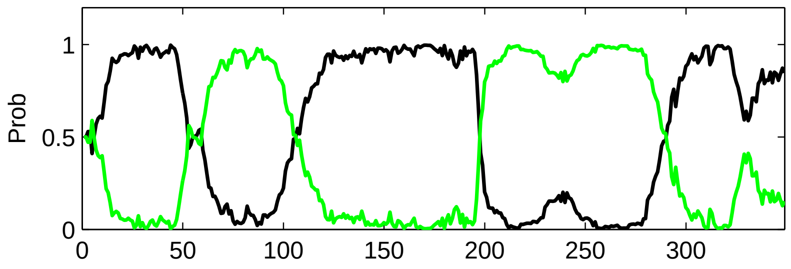
Figure 4.45: F-16 Flight Without Clutter: Temporally Averaged RMS Errors. Performance comparisons with Korn & Beean restarts (KB), split axes, and dimple Gaussian filters.



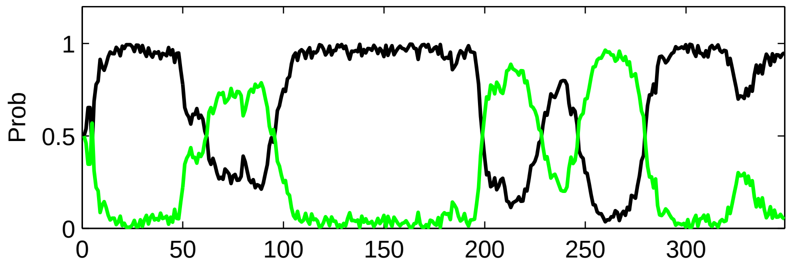
(a) Standard Macrohypothesis MMAE Flow in Clutter



(b) Korn & Beean Macrohypothesis MMAE Flow in Clutter

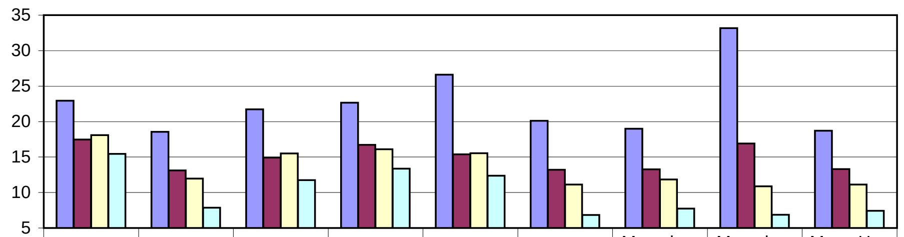


(c) Modified Korn & Beean Macrohypothesis MMAE Flow in Clutter



(d) Dimple MMAE Macrohypothesis Flow in Clutter

Figure 4.46: F-16 Macrohypothesis MMAE with Clutter: Probability Flow with and Without Ad Hoc Marshalling Techniques



	Agg	Macrohyp MMAE	Macrohyp IMM	Macromix MMAE	Macromix IMM	Macrohyp MMAE KB	Macrohyp MMAE MKB	Macrohyp MMAE Dimple	MacroHyp MMAE Split
RMS Position Error (m)	22.932	18.56	21.71	22.659	26.615	20.101	18.987	33.153	18.706
RMS Velocity Error (m/s)	17.464	13.116	14.923	16.71	15.384	13.194	13.263	16.906	13.288
Benign Regime RMS Position(m)	18.082	11.957	15.494	16.096	15.55	11.113	11.831	10.89	11.117
Benign Regime RMS Velocity (m/s)	15.44	7.8495	11.746	13.359	12.376	6.827	7.7267	6.8597	7.4324

Figure 4.47: F-16 Flight in Clutter: Temporally Averaged RMS Values (Complete and Benign Regime)

rapid bidirectional flow. Comparing performance on the broken loop model in clutter, shown in Figures 4.20 (MMAE) and 4.21 (IMM), it can be seen that the IMM exhibits sufficiently agile probability flow in the case of rapidly changing dynamics. In fact, this is the basic assumption of the IMM – that the target mode will switch between filters.

4.6 Summary

Section 4.2 reaccomplished the track life simulations performed by Williams [35–37], verifying that the simulator implementation was correct. Additionally, these tests showed that a clutter region based on predicted estimates of the states would provide more realistic data than if the clutter region was based on the truth position, which would create unrealistically optimistic results. Section 4.3 tested constant-velocity filter models against a truth model with ever increasing dynamics. Both with and without clutter, the multiple model structures displayed ability to identify the appropriate dynamics mode for the target. Section 4.4 tested the same models against realistic flight data, garnering similar results: the multiple model structures correctly identified the appropriate maneuver mode and were able to reduce RMS error in both position and velocity. Section 4.5 discussed several techniques for ensuring agile probability flow from aggressive to benign filters, such as a modified Korn & Beean restart, a dimpled Gaussian filter, and splitting the tracker axes. These methods worked well without clutter, but incurred significant deferred decision errors in trials with clutter. Closer inspection revealed the performance of the benign periods of these simulations had improved, but the peak error had increased enough to remove the benefit, when viewed from a temporally averaged RMS error metric.

V. Conclusions and Recommendations

5.1 Restatement of Research Goal

The goal of this research has been to integrate multiple model techniques (to resolve alternate possible dynamics modes for a target) with a cost-function-based multiple hypothesis tracker (to address clutter) and test its performance using realistic data. Two multiple model structure were tested, the Multiple Model Adaptive Estimator (MMAE) and the Interacting Multiple Model (IMM). Two possible schemes for integrating the multiple model structure exist. The method proposed by Smith [30, 31] runs several MHTs in parallel and uses multiple model techniques to blend the pseudo-states generated by each mixture to form a result. This MMAE or IMM structure of ISE-based Gaussian mixtures is referred to as a macromixture architecture. The new architecture presented here attempts to modify the commonly accepted track-based multiple model MHT architecture to function inside a Gaussian mixture based MHT by modifying the Gaussian mixture to maintain composite hypotheses (addressed by multiple model techniques) instead of individual Gaussians (handled by single Kalman filters instead). This ISE-based Gaussian mixture structure of MMAEs or IMMs is referred to as a macrohypothesis architecture. The performance capabilities of the two schema are compared by running both of them against synthetic loop models and realistic target data generated by a flight simulator.

5.2 Summary of Results

Two loop models were developed as test environments with easily predictable multiple model functionality, one with gradually increasing maneuver dynamics and another with alternating periods of increasing dynamics and benign flight. Four multiple model variants were tested: the MMAE/IMM variants of the macrohypothesis structure and the MMAE/IMM variants of the macromixture structure. Each configuration showed correct probability flow: as the magnitude of target dynamics was gradually increased, the elemental filter probability smoothly flowed from the filter assuming benign dynamics to the filter assuming aggressive dynamics. As the target dynamics increased past the tuning levels of the filters, the filter entered a regime of operation in which the tracking was more a factor of the deferred decision-making than of filter capability.

A three-filter MMAE was tested against the loop model, displaying that the configuration is capable of incorporating several filters at one time, but not as effectively in clutter as without clutter. Gating is performed as a union of the individual macrohypotheses, and the larger gates will incorporate more clutter points. The more aggressive filter will generate favorable residuals from the clutter points and deny probability from the benign filter. This process is not reciprocal – in the case of an aggressive truth model, the benign filters were at the lower bound.

The filters were run against the broken loop model showing that, even in clutter, all four multiple model structures were capable of identifying the appropriate target model. The IMMs and macromixture structures displayed faster reaction times to the broken-loop's step changes in dynamics than the macrohypothesis MMAE.

The F-16 flight data provided a spectrum of maneuvers: benign flight, high-G time-correlated turns, and sudden jinking maneuvers. Even in heavy clutter, all four multiple model structures were capable of identifying the appropriate truth model. Significantly, the multiple model structures were capable of maintaining track on the target even when a single benignly tuned filter was shown to be incapable of tracking the maneuver. The fact that the elemental filter probabilities regularly flowed to the benign filter indicates that the multiple model filters were offering filtering capabilities that would be impossible in single model structures.

The correct probability flow implies that the tracking performance will improve, and a metric was devised to quantify the improvement. Namely, the RMS errors were averaged temporally to yield a scalar indicator of tracking error. While the F-16 data seemed to indicate that the multiple model structures offered a clear advantage over a single-filter configuration, the loop model performance hinted at some idiosyncracies of the multiple model MHT integration. The deferred decision-making process of the MHT is unavoidable and, in fact, is essential to the performance of the algorithm. The errors incurred during periods of deferred decision-making are several orders of magnitude higher than those of a well-behaved mixture and severely skew the temporally averaged RMS error. Also, multiple model structures tended to exacerbate the errors generated during the deferred decision process. Consequently, the temporally averaged RMS errors were generated for only a benign portion of flight as well as the entire time interval of interest. Over this subset of simulated trajectory results, the multiple model methods showed a clear and distinct advantage over the single filters.

MMAEs are characteristically slow in returning elemental filter probability to benign filters after the target exits an aggressive maneuvering mode, causing the MMAE to remain in an incorrectly aggressive filter mode after the cessation of maneuver. Any time spent with the assumed filter dynamics mismatched to the actual target dynamics will degrade the performance potential of the multiple model structure. Consequently, several methods were tested in an attempt to return probability to the benign filter as quickly as possible. A modification of the Korn & Beean [11] probability restarts, a split multiple model probability system, and an *ad hoc* Gaussian PDF evaluation were tested against flight data both with and without clutter. Without clutter, every technique displayed increased elemental filter agility and returned probability more quickly and decisively to the appropriate elemental filter. In all cases, the temporally averaged RMS errors decreased. In the cluttered simulations, the filter probability flow improved, yet the temporally averaged RMS errors increased. Upon closer inspection, it was found that the probability flow techniques magnified the errors caused by the deferred decision-making process of the MHT. With regard to a benign portion of flight, those segments in which deferred decision-making did not occur, the probability marshalling techniques did reduce RMS errors.

While the multiple model structures can identify the appropriate target dynamics, their performance penalty during maneuvers tends to negate their potential. The intent of this research was to augment the remarkable capabilities of the ISE-mixture reduction algorithm with a multiple model structure, but when the ISE algorithm is running at its peak, i.e., in extremely heavy clutter, it relies on a deferred decision-making process that a multiple model structure cannot aid – target dynamics cannot be extracted from a random walk in clutter. The performance of the multiple model algorithm in moderate clutter should not be ignored though, and application of these structures should be considered when operating in an environment with moderate clutter or low dynamics.

While the macrohypothesis MMAE emerged as the best performer in the F-16 flight test, the other forms should not be discounted. The IMMs were run with a simplistic Markov mixing matrix, and one can assume their performance could be enhanced by proper tuning. The macromixture IMM displayed the worst performance, which can be attributed to the approximations made in the mixing process by the pseudo-states and pseudo-residuals.

5.3 Significant Contributions of Research

This research demonstrates that the macrohypothesis multiple model configuration can perform effectively within an ISE-cost-function-based multiple hypothesis tracker. Specifically, the multiple model configuration is capable of identifying the appropriate target dynamics mode and basing its internal filters on those dynamics. The new filter structures were shown to perform as well or better than the existing macromixture multiple model MHT schema in a direct comparison.

The efficacy of the ISE algorithm and its multiple model variants were shown against realistic flight data recorded from a simulation of an F-16. The flight data demonstrated highly time-correlated maneuvers that aided the performance of the multiple model structure; namely, the target spent long periods in a benign state punctuated by brief highly dynamic maneuvers. Compared to a single filter MHT, the multiple model MHT had significantly better performance during benign periods, which outweighed performance degradation (compared to a single filter MHT) during intense maneuvers. Figure 5.1 shows the flight path of a 30-component Gaussian mixture in extremely dense clutter. For the majority of the flight time, the best estimate of the tracker, shown in orange, is in agreement with the true target trajectory, shown as a black line. During these periods, the multiple model structures presented here will improve the accuracy of the best estimate. The gross position error occurring near the beginning of the trajectory is an example of the deferred decision process that is inherent in the functioning of an MHT algorithm. During these periods, a multiple model MHT structure will wander farther and

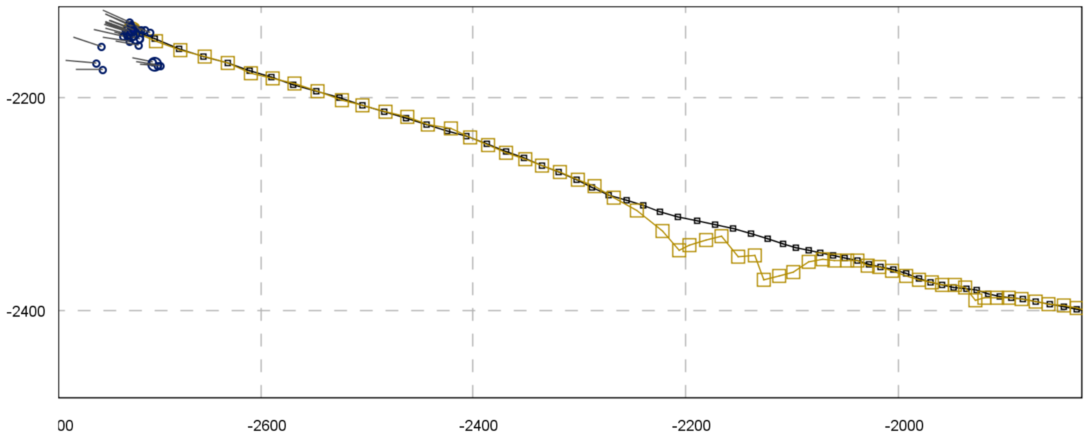


Figure 5.1: 30-Component ISE-Based MRA Running in Heavy Clutter. Here, the target traverses the field from right to left. The clutter density is high enough, $\lambda = .012\, pts/m^2$, that it cannot be meaningfully displayed. The best estimate deviates from the truth significantly for a portion of the flight. This research has shown that multiple model techniques can reduce RMS errors during periods in which the best estimate is coincident with the truth, but multiple model techniques also have a tendency to exacerbate the error caused by clutter-induced deviations. The worthiness of multiple model techniques will depend upon the situation with specific regard to maximum error tolerance.

recover more slowly than a single filter MHT. Thus, the decision of whether a multiple model structure is superior depends on several criteria such as the clutter density (higher density will cause more deferred decision errors), target maneuverability (more dynamic targets will also cause more deferred decision errors), and the maximum allowable error of the tracking solution. Multiple model MHTs have greater error during a deferred decision-making period when compared to a single-filter MHT, but these errors can be two orders of magnitude greater than normal[1] errors. The filter is hardly tracking with any fidelity during deferred-decision periods, so the better performance of the single filter models during these periods is irrelevant considering the performance sacrifices they make during benign periods.

Several *ad hoc* techniques were proposed for marshalling the elemental filter probability into the most beneficial states for the given target dynamics. The Korn & Beean

[1]An MHT will normally have the majority of its probability centered on true target location. The exception, a deferred decision period, occurs when the mixture mean deviates from the target location.

probability restart technique was modified so that it avoided disturbing a dynamic mixture inappropriately by analyzing the mixture covariance prior to probability restarts. Additionally, a modified Gaussian PDF evaluation ("dimpled" Gaussian method) was proposed that speeds probability flow from aggressive to benign filters. While this filter incurred the most deferred decision error in the cluttered runs, its performance during the uncluttered runs showed significant improvement in both probability flow agility and RMS error.

5.4 Recommendations for Future Research

The probability marshalling techniques in Section 4.5 were all effective in increasing the proportion of time that the tracker spent with the appropriate filter model carrying the majority of the probability. Unfortunately, these techniques incurred significant performance penalties during transitions from benign to aggressive filter modes. This is due to probability flow being hampered in both directions due to clutter. Additionally, all of these techniques are geared towards pushing probability toward the benign filter. The probability marshalling techniques may be more effective if they were modified to increase probability flow in both directions, not just in the direction of the benign filter. The modified Korn & Beean method was shown to be successful in identifying "quality" of the mixture and, consequently, the maneuver mode of the target. Perhaps, this trigger could be used as a maneuver detector to signal that the mixture needs to be more "aggressive."

Section 2.6 discussed cost functions in terms of the cost contribution per partition. The Bhattacharyya function was shown to incur a higher cost if the relative magnitude of probabilities were higher, indicating an affinity for higher probability regions in cost evaluation. The ISE algorithm was shown to have no such affinity, but, by modifying the leading scalar on the cross term:

$$J_{MISE} = \int (f(x)^2 - \varpi f(x)g(x) + g(x)^2)dx \tag{5.1}$$

the ISE algorithm can be "tilted" to induce a preference or penalty for higher probability partitions. Figure 5.2 shows the effect of increasing ϖ (from an equilibrium point of -2). In this case, the cost function should tend to retain probability created by hypotheses with

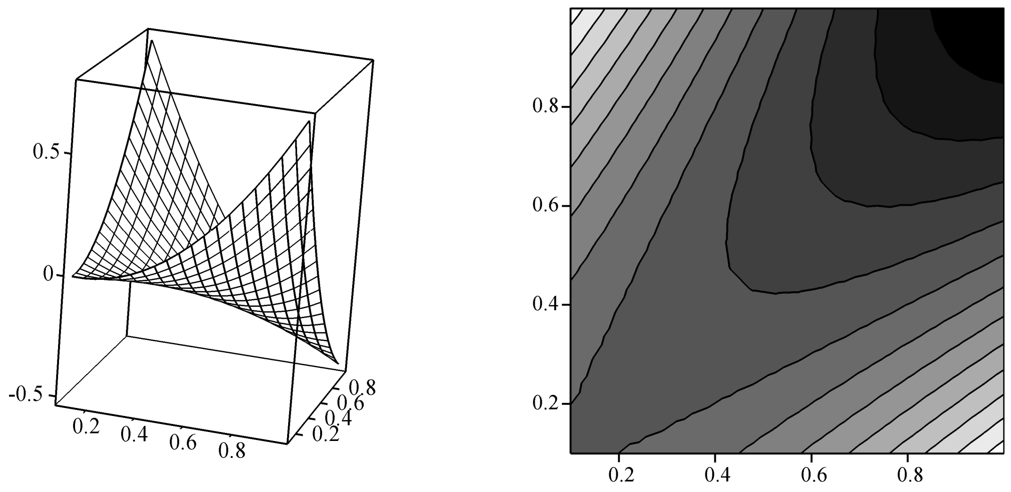


Figure 5.2: Modified ISE Cost Curve for $\varpi > -2$. In limited testing, this ISE variant tended to discard probability from components that have a lower initial probability contribution, which resulted in a negative impact on both track life and RMS errors.

a lower weight (thereby retaining hypotheses with a lower mixture probability). Initial testing at a value of $\varpi = -1.7$ indicated that lower probability components were being retained – the mixture became more spread out and component weights became more evenly distributed. At low clutter densities, the track life was unaffected and RMS errors displayed marginal improvement, but, at higher clutter densities, the modified ISE track life performance began to suffer significantly. The ability of this form to function properly seems dependant upon proper selection of the ϖ value, which appears to depend upon the number of components in the mixture and the density of the clutter. While initial experimentation with this modified algorithm was not as fruitful as expected, its ability to function predictably in lower clutter levels indicates that similar *ad hoc* shaping techniques may have potential for affecting algorithmic performance.

Finally, the computational complexity of the ISE algorithm remains a significant barrier to application. As discussed in Section 4.2, the computation time during these simulations was not real-time. Considering that the simulations only used 15 components and the ISE algorithm has the performance capacity to employ more, every attempt should be made to reduce the computation time of the algorithm. Williams identified the matrix inverse portion of the multivariate Gaussian evaluation as the most expensive computa-

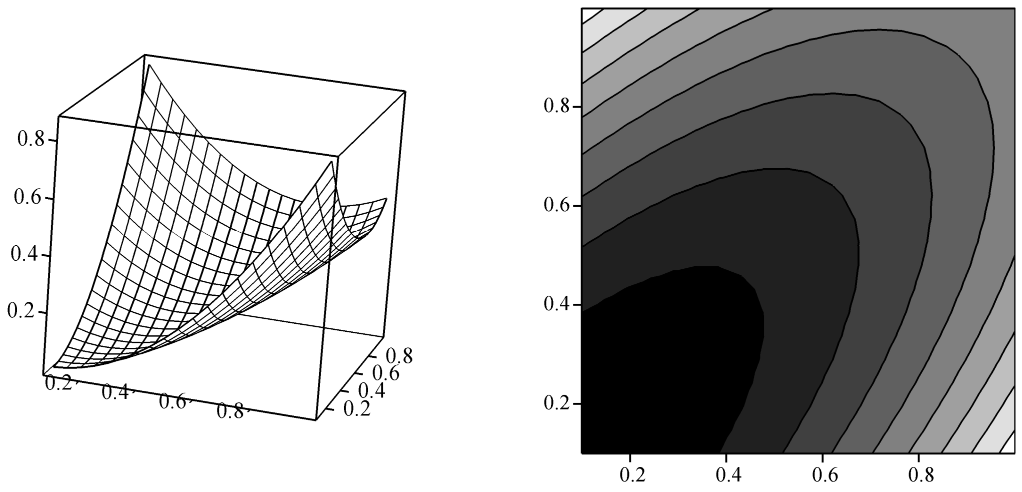


Figure 5.3: Modified ISE Cost Curve for $T < -2$. In limited testing, this ISE variant tended to retain low probability components.

tional portion of the ISE algorithm and implemented a UD-factorization to avoid it [35]. The UD-factorization was performed in code using a Numerical Recipes [22] algorithm.

In recent years, video games have driven the computer industry to develop video graphics cards that are highly specialized for parallel matrix operations, a fundamental element of creating 3D worlds. A state-of-the-art graphics processor in 2006 has roughly 300 million transistors, twice as many as a computer CPU. A recent paper by Galoppo et al. [8] discusses the solution of dense linear systems through image rasterization on inexpensive graphics hardware and uses the LU-decomposition as a speed benchmark. They found that a graphics processor can perform linear algebra functions faster than the ATLAS package [8], a highly optimized matrix library. While this technique has several drawbacks [2], video cards have an unusually fast evolution rate, usually a new product line every year, and may present a viable alternative in the near future. With the possibility of parallel calculation (current video cards have 16 pipelines), processing times could drop drastically. Additionally, an optimized parallel Gaussian sampling routine could allow other cost functions, such as the Kullback-Leibler function, to be evaluated numerically by partitioning the mixture space and sampling at regular intervals.

[2]Graphics Processing Units (GPUs) are currently limited to 32-bit calculations. Additionally, the bus bandwidth between the video card and the cpu is fairly limited, making it difficult to retrieve solutions from the card in a timely fashion.

Throughout the document and on the included movie files, the output of a track visualization tool is used to explain the operation of the tracker. Several output examples are shown here to explain the various symbols. Figure A.1 shows a cluttered field of view with the target trajectory presented as a black line with square ticks marking each epoch on the path. The orange path near the target trajectory is the tracker's best estimate of the target trajectory. The blue ellipses indicate the 3σ boundaries of the individual components (hypotheses) in the Gaussian mixture. The ellipses are not rotated – they merely indicate rough position and magnitude. Their mixture weight is indicated by their opacity, with darker blue indicating higher mixture weight. Measurements are shown as either hollow circles (unassociated measurements) or filled circles (associated measurements). The numbers next to the measurements indicate how many of the hypotheses formed associations with the measurement.

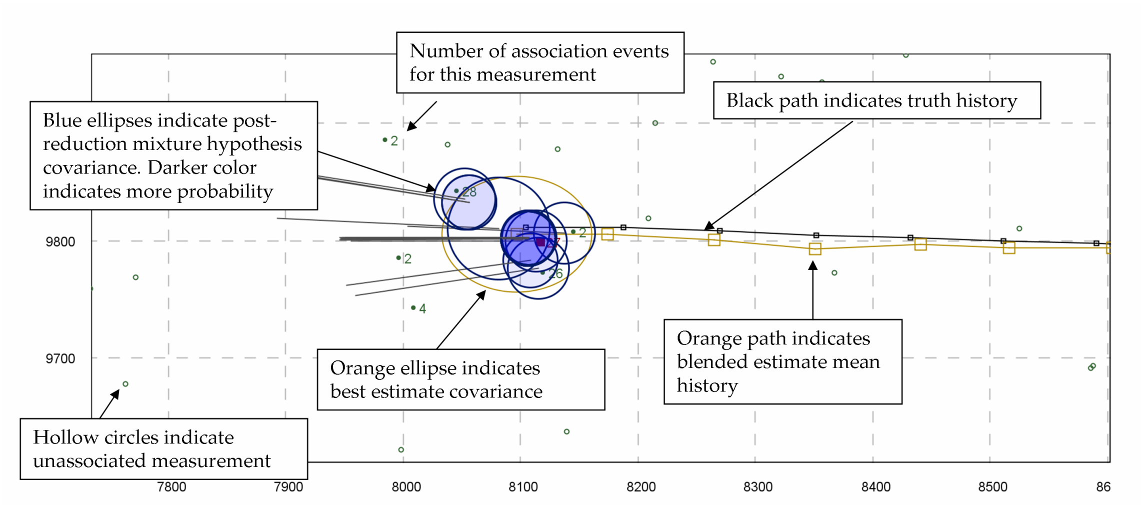


Figure A.1: Example of Target Path, Best Estimate Path, and Estimate Error Ellipses

Figure A.2 shows a multiple model structure, indicated by the elemental filter probabilities in the upper left corner. Additionally, the measurement association gates are shown as rectangles. There are elliptical gates inside the rectangular gates, but they are not shown to avoid cluttering the display. Red ellipses indicate post-update mixture components.

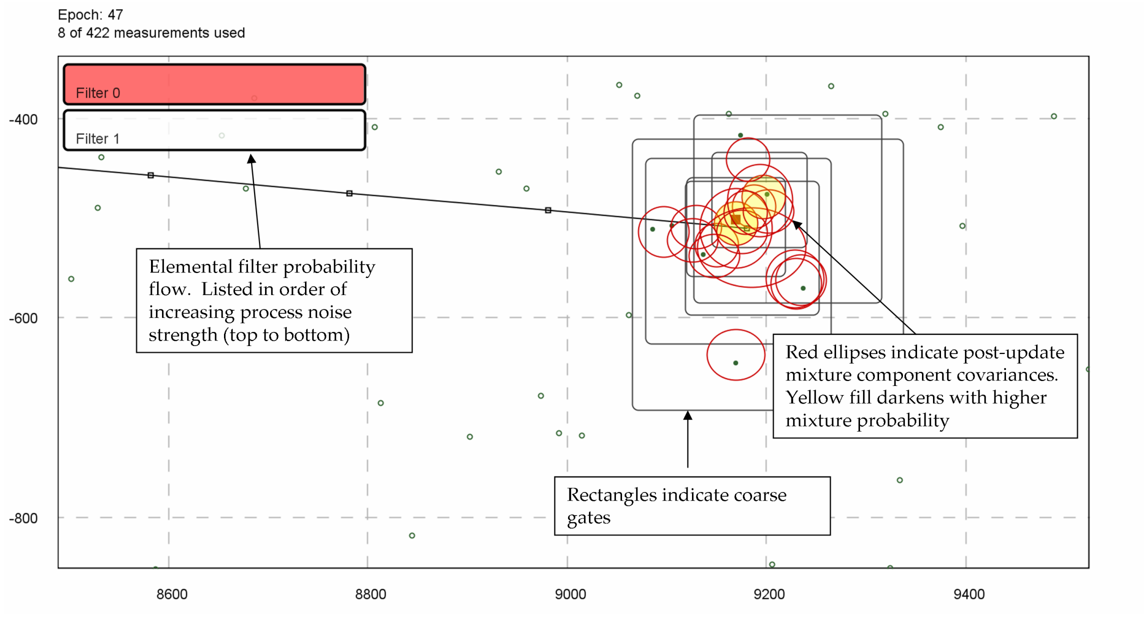


Figure A.2: Example of Elemental Filter Probability Gauges and Association Gates

Figure A.3 shows several epochs simultaneously, which allows the development of false trails and deferred decision to be seen more easily. Epochs get more faint with age relative to the number of epochs being shown. Gray ellipses indicate post-propagation mixture components.

Figure A.4 shows the values of the target velocity and acceleration states, indicated as black circles in space, along with the tracker's estimate of those states, represented as black lines pointing to the actual values. The estimate line opacity is equivalent to the mixture weight so that unlikely estimates are nearly invisible. The magnitude of the axes in these plots have been normalized to the maximum values attained during the simulation. Consequently, they may not be equal in magnitude.

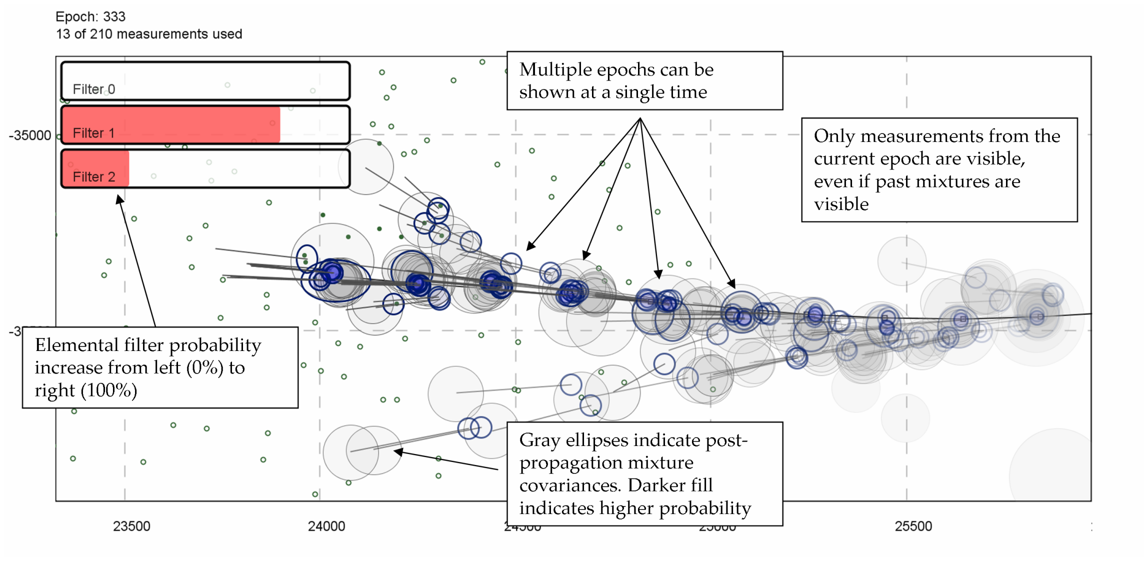


Figure A.3: Example of Multi-Epoch View

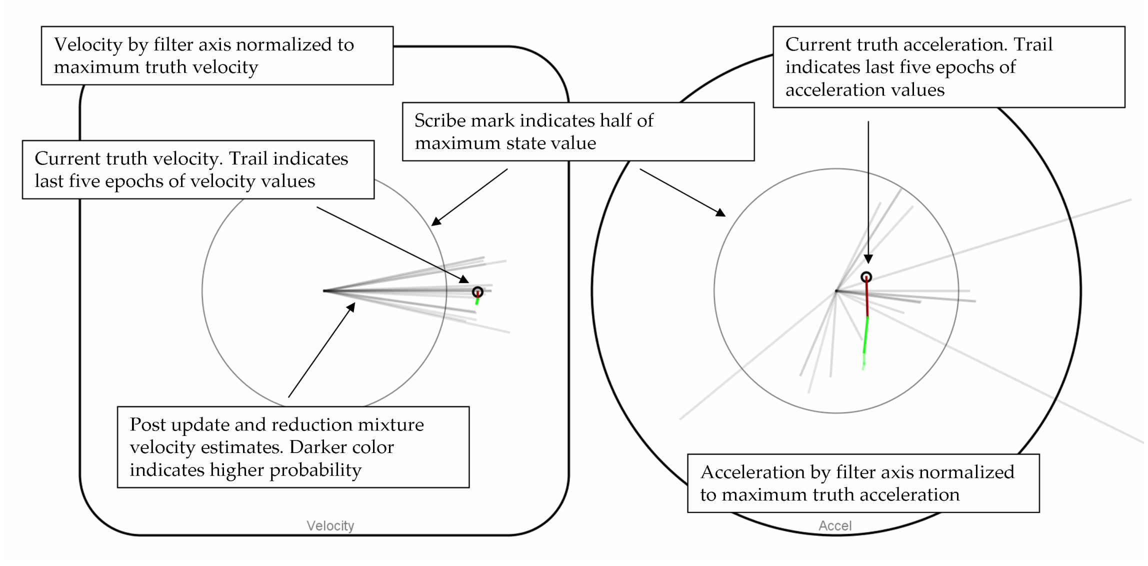

Velocity by filter axis normalized to maximum truth velocity
Current truth velocity. Trail indicates last five epochs of velocity values
Scribe mark indicates half of maximum state value
Post update and reduction mixture velocity estimates. Darker color indicates higher probability
Velocity
Current truth acceleration. Trail indicates last five epochs of acceleration values
Acceleration by filter axis normalized to maximum truth acceleration
Accel

Figure A.4: Example of Velocity and Acceleration Display

Bibliography

1. Bar-Shalom, Y. and E. Tse. "Tracking in A Cluttered Environment with Probabilistic Data Association". *Automatica*, 11:451–460, 1975.

2. Bar-Shalom, Yaakov and Thomas E. Fortmann. *Tracking and Data Association.* Academic Press, Inc., Orlando, FL, 1988.

3. Bar-Shalom, Yaakov and Xiao-Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques.* YBS Publishing, Storrs, CT, 1995.

4. Berndt, Jon S. "JSBSim: Open Source Flight Dynamics Model in C++". URL `http://jsbsim.sourceforge.net`.

5. Blackman, Samuel S. *Multiple-Target Tracking with Radar Applications.* Artech House, Norwood, MA, 1986.

6. Blackman, Samuel S. and Robert Popoli. *Design and Analysis of Modern Tracking Systems.* Artech House, Norwood, MA, 1999.

7. Dempster, R.J., S.S. Blackman, and T.S. Nichols. "Combining IMM Filtering and MHT Data Association for Multitarget Tracking". *Proceedings of the 29th Southeastern Symposium on System Theory*, 123–127. IEEE Press, Cookeville, TN, March 1997.

8. Galoppo, Nico, Naga Govindaraju, Michael Henson, and Dinesh Manocha. "LU-GPU: Efficient Algorithms for Solving Dense Linear Systems on Graphics Hardware". URL `http://gamma.cs.unc.edu/LU-GPU/lugpu05.pdf`.

9. Kailath, Thomas. "The Divergence and Bhattacharyya Distance Measures in Signal Selection". *IEEE Transactions on Communication Theory*, COM-15(1):52–60, February 1967.

10. Kirubarajan, T., Y. Bar-Shalom, W.D. Blair, and G.A. Watson. "IMMPDAF for Radar Management and Tracking Benchmark with ECM". *IEEE Transactions on Aerospace and Electronic Systems*, 34, No. 4:1115–1134, October 1998.

11. Korn, J. and L. Beean. *Application of Multiple Model Adaptive Estimation Algorithms to Maneuver Detection and Estimation.* Technical report, Alphatech, Inc., Burlington, MA, June 1983.

12. Kullback, Solomon. *Information Theory and Statistics.* Dover Publications, Mineola, NY, second edition, 1997.

13. Lainiotis, D.G. and S.K. Park. "On Joint Detection, Estimation and System Identification: Discrete Data Case". *International Journal of Control*, 17(3):609–633, March 1973.

14. Li, X. Rong and Vesselin P. Jilkov. "Survey of Maneuvering Target Tracking. Part V. Multiple-Model Methods". *IEEE Transactions on Aerospace and Electronic Systems*, 41, No. 4:1255–1321, October 2005.

15. Maybeck, Peter S. *Stochastic Models, Estimation, and Control*, volume 1. Navtech, Arlington, VA, 1994.

16. Maybeck, Peter S. *Stochastic Models, Estimation, and Control*, volume 2. Navtech, Arlington, VA, 1994.

17. Maybeck, Peter S. "EENG768 Multiple Model Adaptive Estimation", 2005. Lecture Notes. Air Force Institute of Technology, Wright-Patterson Air Force Base, OH.

18. Maybeck, Peter S. et al. "Proceedings of NAECON". *Investigation of Constant Turn-Rate Dynamics for Airborne Vehicle Tracking*, 896–903. Artech-House, Norwood, MA, May 1982.

19. Olson, Curtis L. "FlightGear Flight Simulator". URL `http://www.flightgear.org`.

20. Pao, Lucy Y. "Multisensor Multitarget Mixture Reduction Algorithms for Tracking". *Journal of Guidance, Control, and Dynamics*, 17(6):1205–1211, November–December 1994.

21. Petrucci, David J. *Gaussian Mixture Reduction For Bayesian Target Tracking in Clutter*. Master's thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, December 2005. AFIT/GE/ENG/06-01.

22. Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, second edition, 1992.

23. Reid, Donald B. "An Algorithm for Tracking Multiple Targets". *IEEE Transactions on Automatic Control*, AC-24(6):843–854, December 1979.

24. Salmond, David J. *Mixture Reduction Algorithms for Uncertain Tracking*. Technical Report 88004, Royal Aerospace Establishment, Farnborough, UK, January 1988. DTIC Number ADA197641.

25. Salmond, David J. "Mixture Reduction Algorithms for Target Tracking". *IEE Colloquium on State Estimation in Aerospace and Tracking Applications*, 7/1–7/4. IEE Publishing, London, UK, December 1989.

26. Salmond, David J. *Tracking in Uncertain Environments*. Technical Memorandum AW 121, Royal Aerospace Establishment, Farnborough, UK, September 1989. DTIC Number ADA215866. Taken from a D Phil thesis of the University of Sussex.

27. Salmond, David J. "Mixture Reduction Algorithms for Target Tracking in Clutter". *SPIE Signal and Data Processing of Small Targets*, 1305:434–445, April 1990.

28. Schell, Chad and Stephen P Linder. "Experimental Evaluation of Tracking Algorithms used for the Determination of Fish Behavioural Statistics". URL `www.cs.dartmouth.edu/ spl/publications/2005/IEEE%20Oceanic.pdf`.

29. Singer, R.A., R.G. Sea, and K.B. Housewright. "Derivation and Evaluation of Improved Tracking Filters for Use in Dense Multi-target Environments". *IEEE Transactions on Information Theory*, IT-20(4):423–832, July 1974.

30. Smith, Brian D. *Multiple Model Adaptive Estimator Target Tracker For Maneuvering Targets in Clutter*. Master's thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, March 2005. AFIT/GE/ENG/05-18.

31. Smith, Brian D. and Peter S. Maybeck. "Multiple Model Tracker Based on Gaussian Mixture Reduction for Maneuvering Targets in Dense Clutter". *Proceedings of the Seventh International Conference on Information Fusion*, 40–47. International Society of Information Fusion, Philadelphia, PA, July 2005.

32. Streit, Roy L. and Tod E. Luginbuhl. *Probabilistic Multi-Hypothesis Tracking*. Technical report, Naval Undersea Warfare Center Division, Newport, RI, 1995.

33. T. Cormen, R. Rivest, C. Leiserson. *Introduction to Algorithms*. MIT Press, Boston, MA, 1990.

34. Torelli, R., A. Graziano, and A. Farina. "IM3HT Algorithm: A Joint Formulation of IMM and MHT for Multitarget Tracking". *European Journal of Control*, volume 5, 46–53. 1999.

35. Williams, Jason L. *Gaussian Mixture Reduction for Tracking Multiple Maneuvering Targets in Clutter*. Master's thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, March 2003. AFIT/GE/ENG/03-19.

36. Williams, Jason L. and Peter S. Maybeck. "Cost-Function-Based Gaussian Mixture Reduction for Target Tracking". *Proceeding of the Sixth International Conference on Information Fusion*, 1047–1054. International Society of Information Fusion, Cairns, Australia, July 2003.

37. Williams, Jason L. and Peter S. Maybeck. "Cost-Function-Based Hypothesis Control Techniques for Mulitple Hypothesis Tracking". *Proceedings of the SPIE Annual International Defense and Security Symposium, Vol. 5428*, 167–179. SPIE Press, Orlando, FL, April 2004.

38. Zhao, Huijing and Ryosuke Shibasaki. "A Novel System for Tracking Pedestrians Using Multiple Single-Row Laser-Range Scanners". *IEEE Transactions On Systems, Man, and Cybernetics*, 35(2):283–291, March 2005.

NONPRINT FORM

1. Type of Product: Compact Disk	**2. Operating System/Version:** Windows XP	**3. New Product or Replacement:** New	**4. Type of File:** Simulation executables and data files

5. Language/Utility Program:

C++/Java

6. # of Files/# of Products: 738 files/ 4 programs + 1 data set	**7. Character Set:** ISO9660	**8. Disk Capacity:** 700MB
	9. Compatibility:	**10. Disk Size:** 350MB

11. Title:

Multiple Model Methods For Cost Function Based Multiple Hypothesis Trackers

12. Performing Organization: Air Force Institute of Technology Graduate School of Engineering and Management, Bldg. 641 (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	**13. Performing Report #:** AFIT/GE/ENG/06-29	**14. Contract #:**
		15. Program Element #:
16. Sponsor/Monitor: Maj Todd E. Combs, USAF, Ph.D. (703) 696-9548 Program Manager, Optimization & Discrete Mathematics AFOSR/NM Suite 325, Room 3112 875 Randolph St. Arlington, VA 22203-1768	**17. Sponsor/Monitor # Acronym:**	**19. Project #:**
	18. Sponsor/Monitor #:	**20. Task #:**
		21. Work Unit #:

22. Date: 06 Mar 06	**23. Classification of Product:** Unclassified
24. Security Classification Authority:	**25. Declassification/Downgrade Schedule:**

26. Distribution/Availability:

Approval for public release; distribution is unlimited.

NONPRINT FORM

27. Abstract:

Multiple hypothesis trackers (MHTs) are widely accepted as the best means of tracking targets in clutter. This research seeks to incorporate multiple model Kalman filters into an Integral Square Error (ISE) cost-function-based MHT to increase the fidelity of target state estimation. Results indicate that the proposed multiple model methods can properly identify the maneuver mode of a target in dense clutter and ensure that an appropriately tuned filter is used. During benign portions of flight, this causes significant reductions in position and velocity RMS errors compared to a single-filter MHT. During portions of flight when the mixture mean deviates significantly from true target position, so-called deferred decision periods, the multiple model structures tend to accumulate greater RMS errors than a single-filter MHT, but this effect is inconsequential considering the inherently large magnitude of these errors (a non-MHT tracker would not be able to track during these periods at all). The multiple model MHT structures do not negatively impact track life when compared to a single-filter MHT.

28. Classification of Abstract: Unclassified		**29. Limitation of Abstract:** Unlimited		
30. Subject Terms: Multiple hypothesis tracker, Multiple model adaptive estimator, Interacting multiple model, Kalman filter, Gaussian mixture, Target tracking		**30a. Classification of Subject Terms:** Unclassified		
31. Required Peripherals:				
32. # of Physical Records:	**33. # of Logical Records:**	**34. # of Tracks:**		
35. Record Type:	**36. Color:**	**37. Recording System:**		
38. Recording Density:	**39. Parity:**	**40. Playtime:**		
41. Playback Speed:	**42. Video:**	**43. Text:**	**44. Still Photos:**	**45. Audio:**
46. Other:				
47. Documentation/Supplemental Information:				
48. Point of Contact and Telephone Number: Dr. Peter S. Maybeck (ENG) 937) 255–3636, x4581 Peter.Maybeck@afit.edu				

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)* 23–03–2006	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Aug 2004 — Mar 2006

4. TITLE AND SUBTITLE

Multiple Model Methods For Cost Function Based Multiple Hypothesis Trackers

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Matthew C Kozak, Capt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management, Bldg. 641 (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/06-29

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Maj Todd E. Combs, USAF, Ph.D. (703) 696-9548
Program Manager, Optimization & Discrete Mathematics
AFOSR/NM
Suite 325, Room 3112
875 Randolph St.
Arlington, VA 22203-1768 email: todd.combs@afosr.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Multiple hypothesis trackers (MHTs) are widely accepted as the best means of tracking targets in clutter. This research seeks to incorporate multiple model Kalman filters into an Integral Square Error (ISE) cost-function-based MHT to increase the fidelity of target state estimation. Results indicate that the proposed multiple model methods can properly identify the maneuver mode of a target in dense clutter and ensure that an appropriately tuned filter is used. During benign portions of flight, this causes significant reductions in position and velocity RMS errors compared to a single-filter MHT. During portions of flight when the mixture mean deviates significantly from true target position, so-called deferred decision periods, the multiple model structures tend to accumulate greater RMS errors than a single-filter MHT, but this effect is inconsequential considering the inherently large magnitude of these errors (a non-MHT tracker would not be able to track during these periods at all). The multiple model MHT structures do not negatively impact track life when compared to a single-filter MHT.

15. SUBJECT TERMS

Multiple hypothesis tracker, Multiple model adaptive estimator, Interacting multiple model, Kalman filter, Gaussian mixture, Target tracking

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Peter S. Maybeck (ENG)
U	U	U	UU	155	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255–3636, x4581 Peter.Maybeck@afit.edu