

ADAPTIVE ESTIMATION TECHNIQUES FOR RESIDENT SPACE OBJECT  
CHARACTERIZATION

by

Jamie LaPointe

---

Copyright © Jamie LaPointe 2016

A Thesis Submitted to the Faculty of the  
DEPARTMENT OF AEROSPACE & MECHANICAL ENGINEERING

In Partial Fulfillment of the Requirements  
For the Degree of

MASTER OF SCIENCE  
WITH A MAJOR IN AEROSPACE ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

2016

## STATEMENT BY AUTHOR

The thesis titled “Adaptive Estimation Techniques for Resident Space Object Characterization” prepared by Jamie LaPointe has been submitted in partial fulfillment of requirements for a master’s degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that an accurate acknowledgement of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: \_\_\_\_\_

## APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

\_\_\_\_\_  
*David Gaylor*  
*Associate Professor of Practice*

*Defense Date:*  
12/1/2016

# Acknowledgments

First I need to thank my wife and kids for putting up with me devoting lots of family time to the task of completing this research. I had to make many sacrifices and their loving support and understanding allowed me to get through to the end of this process.

I need to thank my thesis advisor, Dr. David Gaylor, for all of his kind and extremely helpful support during my long research journey. He taught and guided me through most of what I now know about the field of astrodynamics and greatly expanded my knowledge of state estimation and adaptive estimation techniques. He understood the time constraints placed on me by working and having a family, yet still believed in me. He introduced me to this unique field where it rekindled my love for the space environment. I am very thankful for his support.

Dr. Linares was gracious enough to let me use his MATLAB<sup>®</sup> codes and models. The simulation created for this thesis utilizes the MMAE formulation and many other pieces of this code. A summary of what was changed and added is in Section 3.6. His work gave me a head start in my own research. He was extremely helpful in sharing his codes, teaching me what his research had accomplished, and allowing me to extend it. He was always available to answer any questions I may have had about either his model or MMAE.

Lastly, I need to thank my coworkers and employer for allowing me to work a modified schedule and quasi-part-time. They were very understanding, supportive, and allowed me

to put in less than 100% of my effort at work during this time period. Without everyone's support, this research would not have been possible. For that I am extremely grateful.

# Contents

<b>List of Figures</b>	<b>8</b>
<b>List of Tables</b>	<b>10</b>
<b>1 Abstract</b>	<b>12</b>
<b>2 Introduction</b>	<b>14</b>
2.1 Thesis Outline . . . . .	20
<b>3 Methods</b>	<b>21</b>
3.1 Sequential Estimators . . . . .	21
3.1.1 Extended Kalman Filter . . . . .	22
3.1.2 Unscented Kalman Filter . . . . .	26
3.2 Adaptive Estimation Techniques . . . . .	33
3.2.1 Multiple Model Adaptive Estimation . . . . .	33
3.2.2 Mixture of Experts . . . . .	35
3.2.3 Hierarchical Mixture of Experts . . . . .	40
3.3 Dynamic/Process Models . . . . .	43
3.3.1 Solar Radiation Pressure . . . . .	44
3.3.2 Gravity . . . . .	47

---

3.3.3	State Propagation . . . . .	49
3.4	Measurement Models . . . . .	52
3.4.1	Bidirectional Reflectance Distribution Model . . . . .	53
3.4.2	Charge Coupled Device Noise Model . . . . .	57
3.4.3	Apparent Magnitude Model . . . . .	61
3.4.4	Angle Measurement Model . . . . .	64
3.5	Use of HME with UKF Experts . . . . .	65
3.5.1	UKF Details . . . . .	65
3.5.2	UKF Attitude Details . . . . .	66
3.5.3	MMAE & Single-Layer ME Details . . . . .	69
3.5.4	Dual-Layer HME Details . . . . .	70
3.6	Simulation . . . . .	72
<b>4</b>	<b>Results</b>	<b>74</b>
4.1	Tuning . . . . .	74
4.2	Experiment Setup . . . . .	75
4.3	Experiment 1 - Size Detection . . . . .	78
4.4	Experiment 2 - Surface Material Reflectance Detection . . . . .	84
4.5	Experiment 3 - Dual Layer HME - Size & Reflectance . . . . .	87
4.6	Experiment 4 - Size Detection - Truth not in bank . . . . .	95
4.6.1	4a - Truth Bounded By Filter Bank - Close . . . . .	96
4.6.2	4b - Truth Bounded By Bank - Further Away . . . . .	98
4.6.3	4c - Truth Outside Bank . . . . .	103
4.7	Experiment 5 - Dual Layer HME - Maneuver Time Detection . . . . .	104
4.7.1	Experiment 5b - In-Track Maneuver Detection . . . . .	109
4.8	Experiment 6 - Dual Layer HME - Compare Methods . . . . .	114

---

4.9 Experiment 7 - Dual Layer HME Monte Carlo . . . . .	120
4.10 Summary Results . . . . .	123
<b>5 Conclusions</b>	<b>124</b>
5.1 Future Work . . . . .	125
<b>Bibliography</b>	<b>128</b>

# List of Figures

2.1	Monthly Number of Cataloged Objects in Earth Orbit by Object Type . . . .	16
3.1	MMAE Design . . . . .	34
3.2	General ME Design . . . . .	36
3.3	General HME Design . . . . .	42
3.4	Solar Radiation Pressure Geometry . . . . .	44
3.5	Specular vs Diffuse Reflectivity . . . . .	46
3.6	Geocentric Latitude . . . . .	48
3.7	BRDF Geometry . . . . .	54
3.8	Microfacet Rough Surface . . . . .	54
3.9	Diffuse reflectance of the substrate under specular coating . . . . .	55
3.10	Fresnel Reflectance . . . . .	56
3.11	Basic Earth RSO Sun Geometry . . . . .	62
4.1	Experiment 1 - RSO State Estimation Errors . . . . .	80
4.2	Experiment 1 - Apparent Magnitude Measurement & Residuals . . . . .	82
4.3	Experiment 1 - MMAE & ME Weights . . . . .	83
4.4	Experiment 2 - Apparent Magnitude Residuals . . . . .	85
4.5	Experiment 2 - MMAE & ME Weights . . . . .	86

---

4.6	Experiment 3 - Selected Nominal Indices . . . . .	89
4.7	Experiment 3 - HME TLGN Gating Weights . . . . .	90
4.8	Experiment 3 - HME Macromode Gating Weights . . . . .	91
4.9	Experiment 3 - Nominal Model Am Residuals . . . . .	92
4.10	Experiment 3 - MMAE Weights . . . . .	92
4.11	Experiment 3 - RSO Attitude & Rate State Estimation Errors . . . . .	93
4.12	Experiment 3 - RSO Position & Velocity State Estimation Errors . . . . .	94
4.13	Experiment 4a - MMAE & HME Gating Weights . . . . .	97
4.14	Experiment 4a - Measurement Residuals . . . . .	98
4.15	Experiment 4b - MMAE & HME Gating Weights . . . . .	100
4.16	Experiment 4b - BAM Measurement Residuals . . . . .	102
4.17	Experiment 4c - MMAE & HME Gating Weights . . . . .	104
4.18	Local VNB Frame . . . . .	105
4.19	Experiment 5 - Selected MM Nodes . . . . .	107
4.20	Experiment 5 - HME Gating Weights . . . . .	108
4.21	Experiment 5 - MMAE Weights . . . . .	109
4.22	Experiment 5b - Weights . . . . .	112
4.23	Experiment 5b - Measurement Residuals . . . . .	113
4.24	Experiment 5b - Position Errors . . . . .	113
4.25	Experiment 5b - Velocity Errors . . . . .	114
4.26	RSO Apparent Magnitude Measurement . . . . .	114
4.27	Experiment 6 - HME Selected Nominal Index . . . . .	118
4.28	Experiment 6 - HME MM Gating Weights . . . . .	119
4.29	Experiment 6 - HME TLGN Gating Weights . . . . .	120
4.30	Experiment 7 - HME Selected Nominal Index . . . . .	122

# List of Tables

3.1	EKF algorithm	23
3.2	EKF Variables Part 1	24
3.3	EKF Variables Part 2	25
3.4	Tuning Variables	27
3.5	Scaling Parameters	27
3.6	UKF Weights	27
3.7	UKF Sigma Points	28
3.8	UKF Algorithm w/ Full Augmented State	31
3.9	UKF Algorithm w/ Unaugmented State	32
3.10	ME algorithm	37
3.11	HME TLGN algorithm	41
3.12	Normalized Legendre Zonal Coefficients	48
4.1	UKF Tuning Values	75
4.2	HME Tuning Values	75
4.3	Dual Layer HME Threshold Values	75
4.4	Orbital Elements of RSO	76
4.5	UKF Initial State $1\sigma$ Errors	76

---

4.6	Experiment 1 - Size Options . . . . .	78
4.7	Experiment 2 - Surface Material Reflectance Options . . . . .	84
4.8	Experiment 3 MM1 - Nominal Model . . . . .	87
4.9	Experiment 3 MM2 - Reflectance Model . . . . .	88
4.10	Experiment 3 MM3 - Size Model . . . . .	88
4.11	Simulation Computer Specs . . . . .	95
4.12	Experiment 4a - Size Options . . . . .	96
4.13	Experiment 4b - Size Options . . . . .	99
4.14	Experiment 4b - Select Areas & RMS values . . . . .	102
4.15	Experiment 5 MM1 - Nominal Model . . . . .	106
4.16	Experiment 5 MM2 - Maneuver at 50 minutes . . . . .	106
4.17	Experiment 5 MM3 - Maneuver at 100 minutes . . . . .	106
4.18	Experiment 5 MM4 - Maneuver at 150 minutes . . . . .	106
4.19	Experiment 5b MM1 - Nominal Model . . . . .	109
4.20	Experiment 5b MM2 - maneuver at 50 minutes . . . . .	110
4.21	Experiment 5b MM3 - maneuver at 100 minutes . . . . .	110
4.22	Experiment 5b MM4 - maneuver at 150 minutes . . . . .	110
4.23	Original HME TLGN algorithm . . . . .	115
4.24	Experiment 6 - Enhanced HME Learning Rate Parameters . . . . .	116
4.25	Experiment 6 MM1 - Nominal Model . . . . .	116
4.26	Experiment 6 MM2 - Reflectance Model . . . . .	117
4.27	Experiment 6 MM3 - Size Model . . . . .	117
4.28	Experiment 7 MM1 - Nominal Model . . . . .	121
4.29	Experiment Conclusion Summary . . . . .	123

# Chapter 1

## Abstract

This thesis investigates using adaptive estimation techniques to determine unknown model parameters such as size and surface material reflectivity, while estimating position, velocity, attitude, and attitude rates of a resident space object. This work focuses on the application of these methods to the space situational awareness problem.

This thesis proposes a unique method of implementing a top-level gating network in a dual-layer hierarchical mixture of experts. In addition it proposes a decaying learning parameter for use in both the single layer mixture of experts and the dual-layer hierarchical mixture of experts. Both a single layer mixture of experts and dual-layer hierarchical mixture of experts are compared to the multiple model adaptive estimation in estimating resident space object parameters such as size and reflectivity. The hierarchical mixture of experts consists of macromodes. Each macromode can estimate a different parameter in parallel. Each macromode is a single layer mixture of experts with unscented Kalman filters used as the experts. A gating network in each macromode determines a gating weight which is used as a hypothesis tester. Then the output of the macromode gating weights go to a top level gating weight to determine which macromode contains the most

---

probable model. The measurements consist of astrometric and photometric data from non-resolved observations of the target gathered via a telescope with a charge coupled device camera. Each filter receives the same measurement sequence. The apparent magnitude measurement model consists of the Ashikhmin Shirley bidirectional reflectance distribution function. The measurements, process models, and the additional shape, mass, and inertia characteristics allow the algorithm to predict the state and select the most probable fit to the size and reflectance characteristics based on the statistics of the measurement residuals and innovation covariance. A simulation code is developed to test these adaptive estimation techniques. The feasibility of these methods will be demonstrated in this thesis.

## Chapter 2

# Introduction

The primary purpose of this thesis is to compare and contrast the use of a multiple model adaptive estimation (MMAE) and hierarchical mixture of experts (HME) in estimating characteristics of a resident space object (RSO). In addition, this work develops a new dual-layer HME-based algorithm using UKF experts to process astrometric and photometric data to determine the size and reflectivity of a RSO as well as the time, magnitude, and direction of maneuvers. This work also demonstrates the usefulness of a decay function for the HME learning parameter.

Space Situational Awareness (SSA) involves detecting, tracking, identifying, and characterizing RSOs. Knowing properties such as size, shape, configuration, rotational dynamics (attitude and angular velocity), and surface properties such as specular and diffuse albedo (reflectivity) of a RSO are necessary in accurate prediction of its motion and useful in classifying and identifying them.

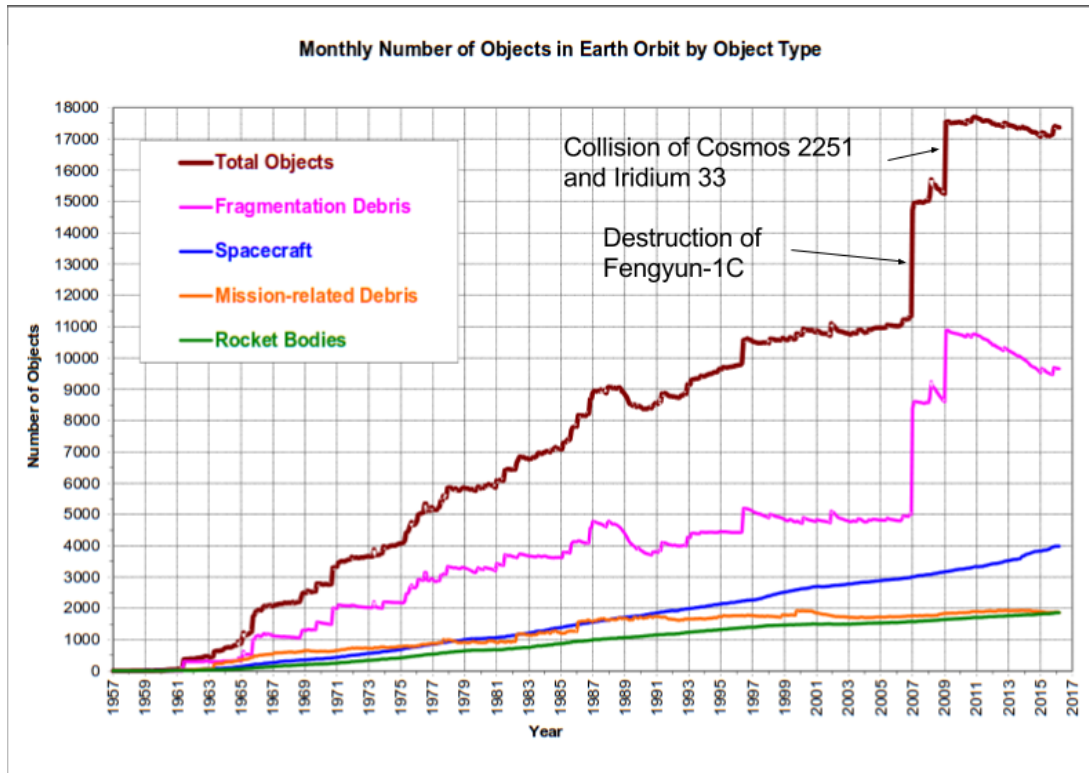
Figure 2.1 shows the amount of detected orbital debris by type as a function of time. Two events in particular worsened the situation of space debris greater than 10 cm. The first was the decommissioning of the Chinese weather satellite Fengyun-1C via a Chinese

anti-ballistic missile test on 11 January 2007. This created a debris cloud of more than 3,400 pieces larger than 10 cm [1]. This is the largest debris cloud yet tracked and will create a significant collision threat to the other space objects in low earth orbit. This debris poses significant risk to satellites and other spacecraft in Earth orbit [2–5]. The second and fourth largest event was the collision and breakup of Cosmos 2251 and Iridium 33 spacecraft in February 2009 [1]. The other interesting item of note from the figure is the slight decrease in debris from about 2011 to around the end of 2015. This is attributed to atmospheric drag and high solar activity in 2014 causing some items in low Earth orbit to burn up in the Earth’s atmosphere [6]. The high solar activity was temporary and is cyclical on multi-year cycles. At the current level of detected orbital debris decrease, it will take decades to reduce it significantly, assuming no additional debris is added.

Impacts to spacecraft from this debris can cause significant damage due to the hyper-velocities involved with orbits. This damage can cause malfunctions, destruction, or complete loss of the asset possibly resulting in a significant loss of functionality and/or money. In the worst case, it can cause loss of human life if it impacts the International Space Station, space capsules, or other objects carrying humans [8].

It is critical that methods be developed that can better predict the effects of non-gravitational forces, resulting in better orbit estimation and prediction. Non-gravitational forces such as solar radiation pressure depend on the solar flux and specific characteristics of a RSO such as shape, size, mass, orientation, rotational rates, orbit parameters, and surface material properties such as albedo. In addition, knowing these same RSO characteristics can help identify and classify RSOs.

Filtering algorithms for state estimation, such as the extended Kalman filter (EKF) [9] and unscented Kalman filter (UKF) [10] have been used for tracking RSOs. However, Kalman filters require an accurate knowledge of the dynamics and measurement model pa-



Monthly Number of Cataloged Objects in Earth Orbit by Object Type: This chart displays a summary of all objects in Earth orbit officially cataloged by the U.S. Space Surveillance Network. "Fragmentation debris" includes satellite breakup debris and anomalous event debris, while "mission-related debris" includes all objects dispensed, separated, or released as part of the planned mission.

Figure 2.1 Monthly Number of Cataloged Objects in Earth Orbit by Object Type [7]

rameters and uncertainties. Incorrect modeling can lead to large errors or filter divergence. For the case of RSO tracking, the size, shape, attitude, and other characteristics that affect the dynamics are often not known. Furthermore, the RSO itself may perform unknown maneuvers. Thus, we do not always have accurate knowledge of the dynamics necessary for successfully using the Kalman filter for RSO tracking.

The MMAE [11] was modified by Chaer et al. to use a neural network approach, which is referred to as a mixture of experts (ME) [12]. Similar to MMAE, the ME is composed of a finite number of filters running in parallel. Each filter's process/measurement models

uses a different realization of the unknown parameters. The filters can include different process models, process noise, and measurement noise. The ME selects the best filter using a neural network, i.e. a gating network capable of assigning a specific weight to each expert in order to provide the optimal mixture of filters. The ME adapts individual filters to better match incoming data via a feedback loop using a gating network to weight the filter outputs [12]. This work defines a time-based decay function as opposed to a constant learning rate. This helps to allow for faster convergence of the a-priori probability to the best model at the beginning and slower convergence to a new best model as time goes on.

Within this framework, this thesis defines a HME [13] that is designed to include multiple layers and groupings. Crain applied the HME as part of a real-time interplanetary orbit determination monitoring algorithm for detecting and identifying changes in the spacecraft dynamic and measurement environments. He conducted experiments of the Mars Pathfinder cruise trajectory environment using simulated range and Doppler measurements. These experiments demonstrated the ability of the HME to identify three environment macromodes, i.e. 1) unmodeled impulsive maneuvers, 2) changes in the solar radiation pressure dynamics, and 3) changes in the measurement noise strength [13].

MMAE using a bank of UKFs was shown to determine the shape of a RSO while also estimating its inertial position, velocity, and rotational dynamics using astrometric and photometric data in the following references [14–17]. Each UKF in the MMAE bank contains a different hypothesis of the characteristic such as a shape estimate. The photometric measurement model uses a bidirectional reflectance distribution function (BRDF) to model the amount of light reaching the observer as a function of inertial position, attitude, the Sun’s position, the observer’s position, the Sun’s luminosity, and the surface material properties of the RSO. The MMAE uses the measurement residuals and innovation covariance from the UKF banks and computes weights corresponding to the conditional probabilities

derived from Bayes' rule in a likelihood probability as described in [14]. The UKF with the highest likelihood probability is the shape most probable to either match or be the best fit to the truth model.

Gaylor et al. demonstrated the use of HME using EKFs to detect changes in a RSO's characteristics and attitude profile detection in [18, 19]. The experiments performed in these papers by Gaylor et al. are:

- Detect size change
- Detect reflectivity change
- Detect shape change
- Determine maneuver time
- Determine maneuver direction and magnitude
- Determine attitude profile such as nadir or sun pointing
- Detect attitude profile change
- Detect a change in spin rate

This work improves upon the work of Gaylor et al. in many aspects such as:

- Utilizing a HME with UKF
- Comparing the results of HME with MMAE
- Determining feature characteristics from a RSO
- Utilizing a CCD error model
- Utilizing the Ashikhmin-Shirley BRDF

- Utilizing only a single observer

The use of either a single layer ME or a dual-layer HME using Kalman filters as the experts was used in references [12, 13, 20]. The HME is similar to the MMAE in its use of a bank of UKFs which contain different hypothesis. However, the HME uses a neural network to compute the gating weights and the MMAE uses Bayes rule. In addition, HME is useful in estimating multiple parameters in parallel as shown in [20].

There are many methods of estimating attitude of a space object. A quick survey can be found in reference [21].

When using quaternions in an UKF the issue of how to efficiently and accurately maintain normalization arises especially within the sigma points of an UKF. One method used by Crassidis is the generalized Rodriguez parameters (GRPs), the details of which can be found in the following references [21–23]. The basic idea is that the state contains the 4-vector quaternion and the error state covariance contains the 3-vector Euler error angles. However, during propagation, the quaternion errors are converted to GRPs, and then back to quaternions when finished to help ensure quaternion normalization and accuracy.

Another method developed by Challa, Moore, and Rogers uses quaternions without any transformation as described in [24].

Light curve data is the time-varying sensor wavelength-dependent apparent magnitude of photons reflected from an object along the line-of-sight to an observer. The apparent magnitude of a RSO is a function of its size, orientation, position in space relative to the Sun and observer vectors, the solar flux of the Sun, and surface material properties. Consequently, one should be able to estimate these characteristics using appropriate algorithms. The asteroid community has shown how using light curve data can help determine an object's characteristics as shown in references [25–29]. Satellites and other RSOs are different than asteroids in many respects such as shape, size, surface material properties,

orbital properties, attitude dynamics, inertia, etc. However, similar techniques can still be used to determine an RSO's characteristics.

If an accurate reflectance model of the object in question can be found, then it is possible to estimate some characteristics and features of the RSO such as shape, size, solar radiation pressure, attitude, and angular rates as demonstrated in references [13–15, 30–32]. Thus it is important to have a numerical model of this reflectance. A popular choice which was developed by the computer graphics community is that of BRDF. An analysis of different BRDF models is described in detail in reference [33].

## 2.1 Thesis Outline

Chapter 3 gives a review of the methods used in the adaptive estimators. These methods include an overview of sequential estimators for non linear models and explains both the EKFs and UKFs. Next it goes into an overview of MMAE and HME adaptive estimation techniques explored in this research. Then it goes on to explain the dynamic process models of solar radiation pressure, gravity potential field, and the overall state propagation model. Next, this thesis describes the measurement and measurement error models used in this research such as the BRDF, CCD error model, apparent magnitude, and angle measurements. Finally, this thesis explains how it all fits together into the UKF along with MMAE and HME models.

Chapter 4 presents results of 7 experiments.

Chapter 5 presents conclusions and recommendations for future work.

## Chapter 3

# Methods

This work uses these methods to demonstrate some of the power of processing photometric and astrometric data from non-resolved images and running them through a bank of UKFs setup as hypothesis testers with proper process and measurement models and then feeding these UKF outputs to either a MMAE, single layer ME, or dual-layer HME to determine RSO characteristics. The models are as detailed as necessary to accomplish the goals set forth in this research and allows the focus to be on the adaptive estimation techniques. In a production environment or when using real data, the models would need to be higher fidelity and contain proper process noise models in the filters.

### 3.1 Sequential Estimators

Sequential estimators are generally used to estimate hidden Markov states and filter noisy measurements. In addition, sequential estimators have been used as the experts or inputs into adaptive estimation techniques. The primary benefit of a sequential estimator are the relatively low computational requirements and the ability to process measurements as they come in as compared to full Monte Carlo Bayesian and batch estimators. In addition,

sequential estimators allow for prediction of the state either with or without a measurement. These abilities lend the sequential estimator for use in real time systems.

### 3.1.1 Extended Kalman Filter

This section briefly summarizes the EKF. Additional resources can be found in books from Maybeck [34], Gelb [35], Tapley, Shultz, and Born [36], Brown and Hwang [37], and Crassidis and Junkins [38].

The EKF is a nonlinear extension of the Kalman filter (KF). The EKF can be used to estimate the hidden states of either a discrete or continuous time non-linear dynamic system.

The EKF propagates the states and converts the predicted state to the predicted measurement via nonlinear functions, while using linearized first-order approximations for updating the error covariances. In general it begins with an initialization of the state and the state's error covariance. It then begins a loop consisting of a predict or time update (timeup) and a correction or measurement update (measup) as shown in Table 3.1.

The state is moved forward in time via a non-linear function  $f$ , while the error is moved forward in time via a linearized error state transition Jacobian matrix  $\Phi$ . Similarly, the predicted state is transformed to the predicted measurement via the possibly non-linear measurement function  $h$ , while the linearized measurement Jacobian matrix,  $H_k$ , transforms the Kalman gain to update the state error covariance during a measurement update. In addition, it is used to transform the state error covariance to its measurement equivalent to update the measurement innovation covariance.

The process noise,  $v_k$ , consists of unmodeled dynamics such as a model using a 6<sup>th</sup> degree and order gravity model. In this case, the process model would contain the expected noise from higher order effects. The measurement noise,  $u_k$ , consists of errors in the

**formulation:**

$$x_{k+1} = f(x_k, u_k) + w_k \quad (3.1)$$

$$z_k = h(x_k) + v_k \quad (3.2)$$

**initialization:**

$$\text{Initial state estimate: } \hat{x}_k = E[\hat{x}_0^-] \quad (3.3)$$

$$\text{Initial state error covariance: } P_k = P_0^- = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \quad (3.4)$$

**time update/prediction:**

$$\text{Predict state forward in time: } \hat{x}_{k+1}^- = f(\hat{x}_k, u_k) \quad (3.5)$$

$$\text{Predict error covariance forward in time: } P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k \quad (3.6)$$

**measurement update/correction:**

$$\text{measurement residual: } \tilde{y}_k = z_k - h(\hat{x}_k^-) \quad (3.7)$$

$$\text{Innovation covariance: } S_k = H_k P_k^- H_k^T + R_k \quad (3.8)$$

$$\text{Kalman gain: } K_k = P_k^- H_k^T S_k^{-1} \quad (3.9)$$

$$\text{Update state: } \hat{x}_k = \hat{x}_k^- + K_k \tilde{y}_k \quad (3.10)$$

$$\text{State error covariance: } P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (3.11)$$

After initialization, then a loop consisting of the time update and measurement update continues ad infinitum

[36–38]

Table 3.1 EKF algorithm

---

$x_k$	= (nx1) process state vector at time $t_k$
$\hat{x}_k^-$	= $\hat{\phantom{x}}_k^-$ is the estimate and - is our best estimate prior to assimilating the measurement
$w_k$	= (nx1) process noise – assumed to be a white sequence with known covariance structure $Q_k$
$u_k$	= (nx1) control vector at time $t_k$
$v_k$	= (mx1) measurement noise – assumed to be a white sequence with known covariance structure $R_k$ and having zero cross correlation with $w_k$
$z_k$	= (mx1) measurement vector at time $t_k$
$f(x_k, u_k)$	= Compute the predicted state from the previous element – usually a non-linear function
$h(x_k)$	= Compute the predicted measurement from the predicted state – potentially a non-linear function
$e_k^-$	= $x_k - \hat{x}_k^-$ = state estimation error
$E[\phantom{x}]$	= Expectation
$P_k$	= (nxn) state error covariance matrix at time $t_k$ = $E[e_k e_k^T] = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T]$
$K_k$	= (nxm) Kalman gain matrix at time $t_k$
$R_k$	= (mxm) measurement noise covariance matrix at time $t_k$ = $E[v_k v_k^T]$
$Q_k$	= (nxn) process noise covariance matrix at time $t_k$ = $E[w_k w_k^T]$
$\tilde{y}_k$	= (mx1) innovation or measurement residual at time $t_k$
$S_k$	= (mxm) Innovation (or measurement residual) covariance matrix at time $t_k$

---

Table 3.2 EKF Variables Part 1

$H_k = (m \times n)$  linear approximation measurement Jacobian - giving the ideal (noiseless) connection between the measurement and the state vector at time  $t_k$

$$= \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_{k+1}^-}$$

$\Phi_k = (n \times n)$  linear approximation state transition Jacobian relating  $x_k$  to  $x_{k+1}$

$$= \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_k, u_k}$$

---

Table 3.3 EKF Variables Part 2

measurement itself. Every measurement instrument that exists contains some error. In general, it is expected that both the process and measurement noise are white and Gaussian (e.g. unbiased, zero mean, and normal distribution) and are uncorrelated. However, biases or non-zero mean and cross-correlation can sometimes be dealt with, assuming that they can be modeled and/or estimated.

The EKF during the timeup step only uses the mean of the previous state, not a weighted sum of a sample of points within the  $1\sigma$  uncertainty [39]. In addition, the state error and measurement covariances are determined by linearizing the dynamic equations via the  $\Phi_k$  and  $H_k$  Jacobian matrices respectively. The EKF provides a first order approximation to the optimal terms [40]. Thus, in general, it performs best over short time steps where the local linearity or first order approximation is valid and its errors are minimized [41].

### 3.1.2 Unscented Kalman Filter

The UKF was invented by Julier and Uhlmann in 1997 [41] to create a better filter for non-linear systems. Like the EKF, the UKF is a sequential state estimator and can be used in discrete time non-linear dynamic systems. The UKF improves upon the EKF in nonlinear environments by providing a second-order Taylor series approximation in the error covariances while being equivalent computationally [42]. Thus, it too lends itself to use in real-time systems. In addition, it makes an attempt at taking the expectation of the state,  $E[x_k]$ , via the use of weights and the sigma points. According to Wan and Van Der Merwe, the two main problems with EKF are that the state distribution is propagated analytically through the first order linearization of the non-linear system and the EKF only uses the mean approximation of the state rather than attempting an expectation of several points lying within the uncertainty [42]. Thus, the UKF improves upon the two main problems of the EKF, while maintaining its benefits.

Both the EKF and UKF are generally used to estimate hidden Markov chain states and filter noisy measurements. They both assume that process and measurement noise are zero-mean, Gaussian, and white. Both approximate the a-posterior mean and covariance for either non-linear processes or measurement functions (or both). However, the EKF is a first-order Taylor series approximation, while the UKF provides an estimate accurate to the second-order for any non-linearity. Most importantly, the UKF does not require Jacobian transformations. For some highly non-linear systems, the Jacobians can be very difficult or impossible to calculate analytically. For these, additional numerical approximations would be needed in an EKF.

The UKF has a few additional tuning variables, in addition to the process and measurement noise models and initial error state covariance of the standard EKF, used to calculate the weights and determine the spread of the sigma points as can be seen in Tables 3.4 to 3.6.

---

$\alpha$  = determines the spread of sigma points around the mean of  $x_k$  (3.12)

$\beta$  = used to incorporate prior knowledge of the distribution of  $x_k$  (3.13)

$\kappa$  = additional tuning value (3.14)

---

Table 3.4 Tuning Variables

---

$L$  = dimension of the random variable  $x_k$  (3.15)

$\lambda = \alpha^2 (L + \kappa) - L$  (3.16)

$\gamma = \sqrt{L + \lambda} = \sqrt{\alpha^2 (L + \kappa)}$  (3.17)

---

Table 3.5 Scaling Parameters

---

$w_0 = \frac{\lambda}{L + \lambda} = 1 - \frac{L}{\alpha^2 (L + \kappa)}$  (3.18)

$W_0 = \frac{\lambda}{L + \lambda} + 1 - \alpha^2 + \beta = 2 - \frac{L}{\alpha^2 (L + \kappa)} - \alpha^2 + \beta$  (3.19)

$w_i = W_i = \frac{1}{2(L + \lambda)} = \frac{1}{2\alpha^2 (L + \kappa)}$  (3.20)

---

Table 3.6 UKF Weights

Typical values for (Eq 3.12) are  $1 \leq \alpha \leq 10^{-4}$ . If the distribution of  $x_k$  is Gaussian, then the optimal value for (Eq 3.13) is  $\beta = 2$ . Usually, (Eq 3.14) is just set to  $\kappa = 0$  [42]. According to Julier and Uhlmann, if  $x_k$  is Gaussian, then a good starting value for (Eq 3.16) is  $L + \lambda = 3$  [41]. Using the equality of (Eq 3.16) and doing some algebra one

comes to a starting value to try for (Eq 3.12):

$$\alpha = \sqrt{\frac{3}{L + \kappa}} \text{ or if } \kappa = 0, \text{ then} \quad (3.21)$$

$$\alpha = \sqrt{\frac{3}{L}} \quad (3.22)$$

To come to the actual best performing value of (Eq 3.12) requires Monte Carlo simulations while varying its value around the starting point in (Eq 3.22) and within the typical range as mentioned in the previous paragraph.

Note that for the weights in Table 3.6, the lower case  $w$  as in  $w_0$  and  $w_i$  are the mean and sigma points weight values while the upper case  $W$  as in  $W_0$  and  $W_i$  are the covariance weight values.

Setting the sigma points is done through the unscented transformation. The unscented transformation is a method of generating points of a random variable which undergoes a nonlinear transformation [41]. Assume we have a set of  $2L$  sigma points with mean  $\hat{x}$  and with covariance  $P_{xx}$ . Each of these sigma points undergo a non-linear transformation to yield  $\hat{y}$  and covariance  $P_{yy}$ . These sigma points are *not* drawn at random but with a deterministic algorithm as shown in Table 3.7.

---


$$\chi_0 = \hat{x} \quad (3.23)$$

$$\chi_i = \hat{x} + \left( \sqrt{(L + \lambda) P_{xx}} \right)_i \quad i = \{1, \dots, L\} \quad (3.24)$$

$$\chi_{i+L} = \hat{x} - \left( \sqrt{(L + \lambda) P_{xx}} \right)_i \quad i = \{1, \dots, L\} \quad (3.25)$$


---

Table 3.7 UKF Sigma Points

In Table 3.7,  $\chi_0$  is the mean point and  $\chi_i$  are the  $2L$  sigma points.  $\sqrt{P_{xx}}$  is the lower

diagonal of the square root matrix. Since  $P_{xx}$  is a symmetric positive definite matrix, the lower diagonal of the square root matrix can be solved for using the efficient Cholesky decomposition. The Cholesky decomposition has a cost of about  $L^3/3$  flops (floating point operations), which is half of the cost of the LU decomposition [43]. If the inputs are Gaussian for all non-linearities, then the unscented transform is accurate up to  $3^{rd}$  order. Otherwise, if the inputs are non-Gaussian for any of the non-linearities, then the unscented transform is accurate up to  $2^{nd}$  order [42]. Additional computational optimizations may be possible as mentioned in [44].

The canonical implementation of the UKF involves creating an augmented state and state covariance matrix. The augmented state contains the regular mean state augmented with the mean process and measurement noise:  $x_k^a = \begin{bmatrix} x_k & w_k & v_k \end{bmatrix}$ . The mean process noise,  $w_k$ , and mean measurement noise,  $v_k$ , are usually just zero. It is very common for the measurement noise to be independent and purely additive. If this is true, then an alternative implementation of the augmented state is  $x_k^a = \begin{bmatrix} x_k & w_k \end{bmatrix}$ . A third alternative is that the process noise too is independent and purely additive, then the augmented state is just the unaugmented state  $x_k^a = [x_k]$ . Below shows the three alternatives of the augmented state covariance matrices:

$$P_k = \begin{bmatrix} P & P_{xw} & P_{xv} \\ P_{wx} & Q & P_{wv} \\ P_{vx} & P_{vw} & R \end{bmatrix}$$

$$P_k = \begin{bmatrix} P & P_{xw} \\ P_{wx} & Q \end{bmatrix}$$

$$P_k = \begin{bmatrix} P \end{bmatrix}$$

The primary benefit of the alternative implementations are the reduction of the dimension of the state vector and by extension the dimension of the state covariance matrix along with a reduction in the number of sigma points.

The first step in the UKF process is determining which of the three alternative augmented matrices should be attempted. The most common scenario is one in which the measurement noise is both purely additive and independent. The next special and often found case is one where the process noise is also purely additive and independent. The fully augmented option is shown in Table 3.8, while the last option of no augmented state is shown in Table 3.9. The middle option ( $x_k^a = \begin{bmatrix} x_k & w_k \end{bmatrix}$ ) was not shown as it should be clear what to do in this case.

In all cases, the first step is always starting with an initial state estimate vector and error state covariance matrix as shown in (Eq 3.26) and (Eq 3.27). If using an augmented state, then set it up with (Eq 3.28) and (Eq 3.29). Note that for the augmented state the mean of the process and measurement noise are nearly always zero.

Next begins the predict/correction loop of all sequential estimators. The time update begins with the sigma points being setup (Eq 3.30) as is shown in Table 3.7. Note, this does require computing the square root matrix every time update, which can be computed more efficiently with the Cholesky decomposition. Then, running all  $2L + 1$  states (mean +  $2L$  sigma points where  $L$  is the dimension of the augmented state vector) through the non-linear process model (Eq 3.31) to propagate the states to current time. The  $2L + 1$  states are then summed together with weights applied to give the predicted state (Eq 3.32). The state covariance is moved forward in time by the summation of the weights applied to the self outer product of the expectation of the difference of the sigma points and predicted state (Eq 3.33). Next, the predicted sigma points are transformed to the predicted sigma measurements by the non-linear measurement model (Eq 3.34). Finally, in the time update,

**initialization:**

$$\hat{x}_k = E \left[ \hat{x}_0^- \right] \quad (3.26)$$

$$P_k = P_0^- = E \left[ (x_k - \hat{x}_k) (x_k - \hat{x}_k)^T \right] \quad (3.27)$$

$$\hat{x}_0^a = \left[ \left( \hat{x}_0^- \right)^T \quad 0_v^T \quad 0_w^T \right]^T \quad (3.28)$$

$$P_k^a = E \left[ (x_k^a - \hat{x}_k^a) (x_k^a - \hat{x}_k^a)^T \right] \quad (3.29)$$

**time update/prediction:**

$$\chi_k^a = \left[ \chi_{0,k}^a \quad \chi_{i,k}^a \quad \chi_{i+L,k}^a \right] \quad (3.30)$$

$$\hat{\chi}_{k+1} = f \left( \hat{\chi}_k^a, u_k \right) \quad (3.31)$$

$$\hat{x}_{k+1}^- = \sum_{i=0}^{2L} w_i \hat{\chi}_{i,k+1}^a \quad (3.32)$$

$$P_{xx,k+1}^- = \sum_{i=0}^{2L} W_i \left[ \hat{\chi}_{i,k+1}^a - \hat{x}_{k+1}^- \right] \left[ \hat{\chi}_{i,k+1}^a - \hat{x}_{k+1}^- \right]^T \quad (3.33)$$

$$\Psi_{k+1} = h \left( \hat{\chi}_{k+1} \right) \quad (3.34)$$

$$\hat{y}_{k+1}^- = \sum_{i=0}^{2L} w_i \Psi_{i,k+1} \quad (3.35)$$

**measurement update/correction:**

$$\tilde{y}_k = z_k - \hat{y}_k^- \quad (3.36)$$

$$P_{yy} = \sum_{i=0}^{2L} W_i \left[ \hat{\Psi}_{i,k+1} - \hat{y}_{k+1}^- \right] \left[ \hat{\Psi}_{i,k+1} - \hat{y}_{k+1}^- \right]^T \quad (3.37)$$

$$P_{xy} = \sum_{i=0}^{2L} W_i \left[ \hat{\chi}_{i,k+1} - \hat{x}_{k+1}^- \right] \left[ \hat{\Psi}_{i,k+1} - \hat{y}_{k+1}^- \right]^T \quad (3.38)$$

$$K_k = P_{xy} P_{yy}^{-1} \quad (3.39)$$

$$\hat{x}_k = \hat{x}_k^- + K_k \tilde{y}_k \quad (3.40)$$

$$P_{xx,k} = P_{xx,k}^- - K_k P_{yy} K_k^T \quad (3.41)$$

Table 3.8 UKF Algorithm w/ Full Augmented State

**initialization:**

$$\hat{x}_k = E \left[ \hat{x}_0^- \right] \quad (3.42)$$

$$P_k = P_0^- = E \left[ (x_k - \hat{x}_k) (x_k - \hat{x}_k)^T \right] \quad (3.43)$$

**time update/prediction:**

$$\chi_k = \left[ \chi_{0,k} \quad \chi_{i,k} \quad \chi_{i+L,k} \right] \quad (3.44)$$

$$\hat{\chi}_{k+1} = f(\hat{\chi}_k, u_k) \quad (3.45)$$

$$\hat{x}_{x+1}^- = \sum_{i=0}^{2L} w_i \hat{\chi}_{i,k+1} \quad (3.46)$$

$$P_{xx,k+1}^- = \sum_{i=0}^{2L} W_i \left[ \hat{\chi}_{i,k+1} - \hat{x}_{k+1}^- \right] \left[ \hat{\chi}_{i,k+1} - \hat{x}_{k+1}^- \right]^T + Q_k \quad (3.47)$$

$$\Psi_{k+1} = h(\hat{\chi}_{k+1}) \quad (3.48)$$

$$\hat{y}_{k+1}^- = \sum_{i=0}^{2L} w_i \Psi_{i,k+1} \quad (3.49)$$

**measurement update/correction:**

$$\tilde{y}_k = z_k - \hat{y}_k^- \quad (3.50)$$

$$P_{yy} = \sum_{i=0}^{2L} W_i \left[ \hat{\Psi}_{i,k+1} - \hat{y}_{k+1}^- \right] \left[ \hat{\Psi}_{i,k+1} - \hat{y}_{k+1}^- \right]^T + R_k \quad (3.51)$$

$$P_{xy} = \sum_{i=0}^{2L} W_i \left[ \hat{\chi}_{i,k+1} - \hat{x}_{k+1}^- \right] \left[ \hat{\Psi}_{i,k+1} - \hat{y}_{k+1}^- \right]^T \quad (3.52)$$

$$K_k = P_{xy} P_{yy}^{-1} \quad (3.53)$$

$$\hat{x}_k = \hat{x}_k^- + K_k \tilde{y}_k \quad (3.54)$$

$$P_{xx,k} = P_{xx,k}^- - K_k P_{yy} K_k^T \quad (3.55)$$

Table 3.9 UKF Algorithm w/ Unaugmented State

the expected measurement is the summation of the weights applied to each expected sigma measurement (Eq 3.35).

The measurement update begins with the calculation of the measurement residual, which, like the EKF, is the difference between the actual measurement and predicted measurement (Eq 3.36). Next, the innovation covariance is computed by taking the summation of the weights applied to the self outer product of the expectation of the difference between the predicted sigma measurements and the predicted measurement (Eq 3.37). The cross correlation matrix is computed by taking the summation of the weights applied to the outer product of the expectation of the difference of the sigma points and predicted state and the expectation of the difference between the predicted sigma measurements and the predicted measurement (Eq 3.38). The Kalman gain is computed by multiplying the cross correlation matrix with the inverse of the innovation covariance matrix (Eq 3.39). Next, the corrected state is computed by adding the predicted state with the product of the Kalman gain and the measurement residual (Eq 3.40). Finally, the state error covariance is corrected by taking the difference between the predicted state error covariance and the Kalman gain applied to the innovation covariance (Eq 3.41).

## 3.2 Adaptive Estimation Techniques

There are a number of adaptive estimation techniques and learning algorithms. This thesis focuses specifically on two adaptive estimation techniques: MMAE and HME.

### 3.2.1 Multiple Model Adaptive Estimation

This section provides a brief overview of the MMAE. See [11, 45–48] for more detailed information. Figure 3.1 shows the basic design of the MMAE process.

The MMAE consists of a bank of sequential estimators. Each filter is independent

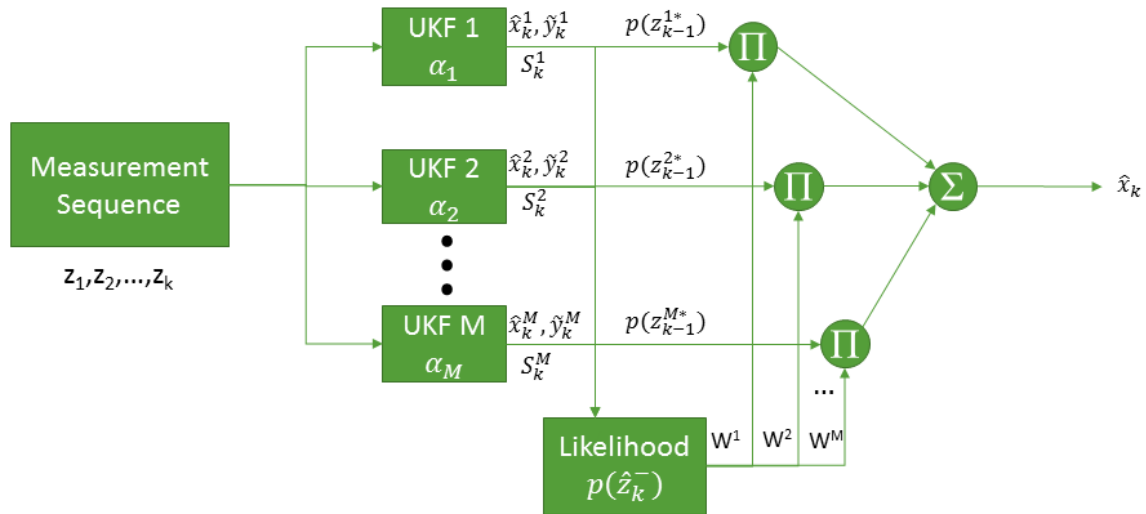


Figure 3.1 MMAE Design

and typically differs by one or more of the following: process model, measurement model, process noise covariance, or measurement noise covariance. The MMAE is recursive and finds weights via Bayes' rule:

$$p(\alpha_i | z_k^*) = \frac{p(z_k^* | \alpha_i) p(\alpha_i)}{p(z_k^*)} = \frac{p(z_k^* | \alpha_i) p(\alpha_i)}{\sum_{j=1}^M p(z_k^* | \alpha_j) p(\alpha_j)}, i = 1, 2, \dots, M \quad (3.56)$$

In (Eq 3.56), we are after the probability density function (pdf) of the unknown parameters contained in the vector  $\alpha_i$  given the set of measurements up to and including time  $k$ ,  $z_k^*$ . It can then be shown that the following recursive equation follows

$$p(\alpha_i | z_k^*) = p(\alpha_i | \hat{z}_k^-) p(\alpha_i | z_{k-1}^*) \quad (3.57)$$

[45]

In Figure 3.1, the weights from the output of the Likelihood function are  $W_i =$

$p(\alpha_i | \hat{z}_k^-)$ .

If we make the assumption that both  $x$  and  $z$  are Gaussian processes, then (Eq 3.57) becomes

$$p(\alpha_i | z_k^*) = \frac{1}{\sqrt{|2\pi S_k^i|}} \exp \left\{ -\frac{(\tilde{y}_k^i)^T (S_k^i)^{-1} (\tilde{y}_k^i)}{2} \right\} p(\alpha_i | z_{k-1}^*), i = 1, 2, \dots, M \quad (3.58)$$

Where,  $|a|$  is the determinant of  $a$  and  $\exp(b)$  is the exponential of  $b$ . Recall too that  $\tilde{y}_k^i$  is the  $i^{\text{th}}$  measurement residual and  $S_k^i$  is the  $i^{\text{th}}$  measurement innovation covariance from the sequential filter.

The conditional mean estimate is found by the following summation of the weighted pdf's

$$\hat{x}_k = \sum_{i=1}^M p(\alpha_i | z_k^*) \hat{x}_k^i \quad (3.59)$$

As can be noticed from (Eq 3.58), the models with a combination of lower residual and tighter variance is given a higher probability of being the best model.

### 3.2.2 Mixture of Experts

This section provides a brief overview of the ME. Additional details of the ME and its extension the HME can be found in [13, 18, 19, 49]. Figure 3.2 shows the basic design of the generalized ME process. The total output of the ME is

$$\hat{x}_k = \sum_{i=1}^M g_i \hat{x}_k^i \quad (3.60)$$

Where  $g_i$  is the gating weight of the  $i^{\text{th}}$  expert and  $M$  is the total number of nodes.

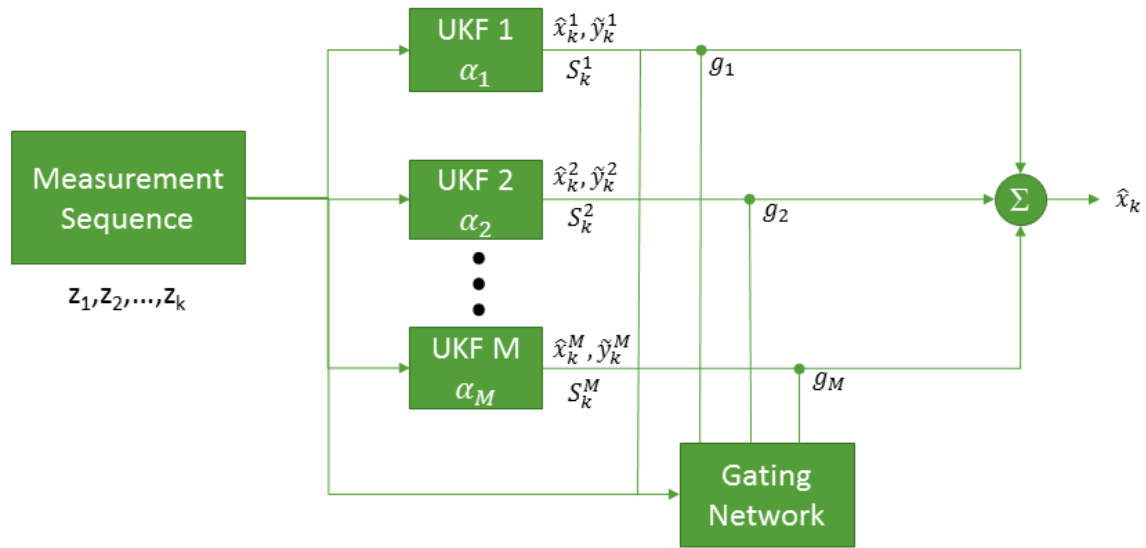


Figure 3.2 General ME Design

The probability distribution of  $z_i$  for the parallel bank of experts is given by:

$$f(z) = \sum_{i=1}^M f(z)_i g_i \quad (3.61)$$

The optimal vector of  $g_i$  maximizes the distribution of (Eq 3.61).

In the ME each expert represents a node or neuron in the network. The algorithm shown in Table 3.10 starts by initializing the synaptic weights,  $a_i$ , to the multiplicative inverse of the number of experts in the bank (Eq 3.62). The  $i$  represents the value for a specific node/neuron/expert and has range  $i = \{1, 2, \dots, M\}$ . Next, the learning rate for each neuron is initialized (Eq 3.63). The initial learning rate is a tuning factor. Higher values allow the gates to converge to a solution quicker than lower values. However, a value too high can cause the a-priori probability gating weights to be given a high confidence too soon. This can cause a cascade of different gating weights getting high probabilities early on prior to the UKFs converging [49]. In general, all neurons have the same initial

**Initialization:**

$$a_0 = \frac{1}{M} \quad (3.62)$$

$$\eta_i = \eta_{i,0} \quad (3.63)$$

**ME Loop:**

$$u_i = z^T a_{i,k-1}, \quad \text{weight inputs option or} \quad (3.64)$$

$$= a_{i,k-1}, \quad \text{unit node alternative} \quad (3.65)$$

$$g_i = \frac{e^{u_i}}{\sum_{j=1}^M e^{u_j}} \quad (3.66)$$

$$f(z_k | \alpha_i) = \frac{1}{\sqrt{|2\pi S_{i,k}|}} \exp \left\{ \frac{-\tilde{y}_{i,k}^T S_{i,k}^{-1} \tilde{y}_{i,k}}{2} \right\} \quad (3.67)$$

$$h_i = \frac{f(z_k | \alpha_i) g_i}{\sum_{j=1}^M f(z_k | \alpha_j) g_j} \quad (3.68)$$

$$\eta_{i,k} = \Upsilon(\eta_{i,k-1}, t) \quad (3.69)$$

$$a_{i,k} = a_{i,k-1} + \eta_{i,k} (h_i - g_i) \quad (3.70)$$

Table 3.10 ME algorithm

learning rate.

After initializing the ME, the measurement loop begins after the experts have run. In the case of using sequential filter experts, the ME uses the measurement residuals of each neuron,  $\tilde{y}_{i,k}$ , and the innovation covariance,  $S_{i,k}$ . The ME loop begins with computing the pre-activation per the synaptic neuron (Eq 3.64). The ME algorithm typically adds weights to the inputs [12, 50]. However, the sequential filter method used in the simulation uses the alternative method which is to activate the neuron with the synaptic weight (Eq 3.65) as was done in reference [13].

The gating weights,  $g_i$ , are computed via the softmax function (Eq 3.66). The softmax function takes the exponential of the current node's 'activated neuron' (just the synaptic weight in this work) normalized against the sum of all the nodes' activated neurons.

Next, we compute the a-posteriori probability of the  $i^{\text{th}}$  neuron. This is done by first computing the  $i^{\text{th}}$  neuron's conditional probability,  $f(z_k|a)$ , (Eq 3.67). The neuron's conditional probability is computed by taking the exponential of the negative of half of the measurement's statistical distance and dividing this by the square root of the determinant of  $2\pi$  times the innovation covariance.

Then, the a-posteriori probability can be computed (Eq 3.68). It is the current node's conditional probability weighted against the gating weight normalized against the sum of all the nodes' conditional probabilities weighted against their gating weights.

The Learning Rate can be constant, be time varying, or some other possible function (Eq 3.69). The learning rate is used to determine how much to adjust the synaptic weights based on the difference between the a-posteriori probability and gating weights. In the simulation, the learning rate is set to a time based decay function with the decay rate, initial learning rate, and minimum learning rate as the two tuning variables of the ME:

$$\Upsilon(\eta_{i,k-1}, t) = (\eta_{i,k-1} - m_\eta) e^{-(1-h_i)\lambda t} + m_\eta \quad (3.71)$$

Where  $\lambda$  is a user defined exponential decay factor and  $m_\eta$  is the minimum learning rate value. This function begins with a default leaning rate paratmeter,  $\eta_0$ . As is usual with decay functions:

- if  $\lambda > 0$ , then (Eq 3.71) decays exponentially and  $\lambda$  represents a decay rate
- if  $\lambda = 0$ , then (Eq 3.71) is constant and neither decays nor grows
- if  $\lambda < 0$ , then (Eq 3.71) grows exponentially and  $\lambda$  represents a growth rate; thus, this selection makes no sense in this context.

When  $\lambda < 0$ , then this function will decay the initial learning paramter rate exponentially as a function of time asymptotically approaching the minimum learning parameter rate,  $m_\eta$ .

The simulation model uses the decay function for the learning parameter to allow a sort of convergence time for the UKF experts to converge to the best fit model. A high learning rate at the beginning allows the filters to select the most probable model using the a-priori probability result quickly. Then, over time, to make it more difficult to select a different model. This is to help account for bad noise forcing a switch to a new model. After the learning rate has decayed, it takes much more evidence to force the a-priori probability to select a new best fit model.

Finally, the synaptic weight is updated based on the learning rate applied to the difference between the a-posteriori probability and a-priori probability gating weights (Eq 3.70).

The gating weights,  $g_i$ , can be thought of as a-priori knowledge that a given model produced the proper inputs. The  $h_i$  are the a-posteriori probabilities as they contain the

model's measurement statistical distance from the expected value. The gating weights are initially all given equal weight. This is to initialize a balanced learning period, with no bias to a particular expert. The gating weights also have the following properties:

$$0 < g_i < 1 \quad (3.72)$$

$$\sum_{i=1}^M g_i = 1 \quad (3.73)$$

In general, the a-priori probability tends towards unity for the current best model. However, if conditions change such that the performance of the current best filter degrades, then the ME gating weights will detect this and select the next best model that fits the current conditions [12].

### 3.2.3 Hierarchical Mixture of Experts

This section provides a brief overview of the HME.

The HME contains a bank of ME macromodes. Each ME macromode follows the same algorithm as discussed in the previous Section 3.2.2 and shown in Table 3.10 and Figure 3.2. The results of each ME macromode goes to a top level gating network (TLGN) and produces the resultant output as shown in Figure 3.3. The algorithm of the TLGN is similar to the GN of each individual ME macromode. However, the TLGN requires a slightly different algorithm for computing  $h_l$  as shown in Table 3.11.

In the HME equations the  $i$  index refers to a specific node within the ME macromode, the  $l$  index refers to a specific ME macromode, and  $L$  refers to the total number of macromodes. Each ME macromode can contain an independent number of nodes, including only a single node.

The HME begins with initializing its top-level synaptic neuron (Eq 3.74) and learning

**Initialization:**

$$a_0 = \frac{1}{L} \quad (3.74)$$

$$\eta = \eta_{l,0} \quad (3.75)$$

**HME Loop:**

$$u_l = a_{l,k-1}, \quad (3.76)$$

$$g_l = \frac{e^{u_l}}{\sum_{j=1}^L e^{u_j}} \quad (3.77)$$

$$n_l = \text{Idx}(\max(g_{l,i})) \quad (3.78)$$

$$f(z_k|\alpha_l) = \frac{1}{\sqrt{|2\pi S_{n_l}|}} \exp\left\{-\frac{\tilde{y}_{n_l}^T S_{n_l}^{-1} \tilde{y}_{n_l}}{2}\right\} \quad (3.79)$$

$$h_l = \frac{f(z_k|\alpha_l) g_l}{\sum_{j=1}^L f(z_k|\alpha_j) g_j} \quad (3.80)$$

$$\eta_{l,k} = \Upsilon(\eta_{l,k-1}, t) \quad (3.81)$$

$$a_{l,k} = a_{l,k-1} + \eta_{l,k} (h_l - g_l) \quad (3.82)$$

Table 3.11 HME TLGN algorithm

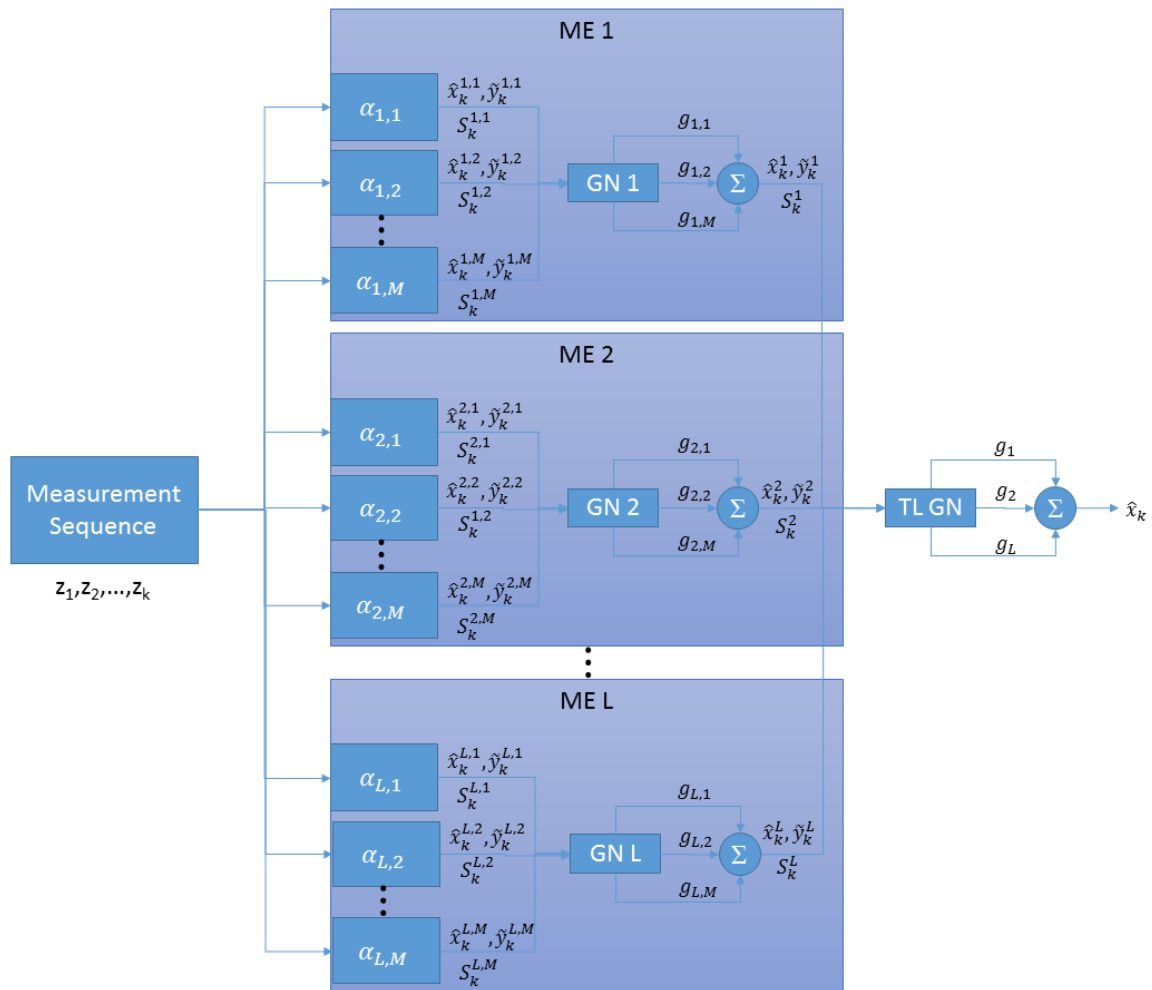


Figure 3.3 General HME Design

rate (Eq 3.75). Then it begins the HME loop. The HME loop begins with updating the top-level neuron,  $u_l$ , using the unit node alternative (Eq 3.76). Then, it updates the gating weight,  $g_l$ , via the softmax function (Eq 3.77).

In (Eq 3.78), the function  $Idx(max(g_{l,i}))$  determines the node index within the ME macromode  $l$  which contains the greatest gating weight. This is the node within the ME macromode  $l$  which has the highest likelihood. The index  $n_l$  represents the node within

the ME macromode  $l$  with the highest probability of matching the measurements.

If the gating weights of a macromode have been reset to  $1/M_l$ , then the TLGN a-posteriori probability is not updated as there cannot be a  $\max(g_{l,i})$  yet. This occurs at initial startup of the HME and anytime the macromode is reset, such as when a nominal model is updated as explained later in Section 3.5.4.

The  $l^{\text{th}}$  macromode neuron's conditional probability,  $f(z_k|a_l)$ , is computed by using the statistics of the neuron in the  $l^{\text{th}}$  macromode with the highest gating weight,  $g_{l,j}$  as shown in (Eq 3.79).

The a-posteriori probability is next computed as shown in (Eq 3.80). This computation is the product of the conditional probability and the TLGN macromode gating weight all normalized with the rest of the macromodes.

The normalization process ensures the correct probability statistics such that the sum of the probabilities for all the macromodes equals 1 for the a-priori probability gating weight,  $\sum_{l=1}^L g_l = 1$ , and the a-posteriori probability,  $\sum_{l=1}^L h_l = 1$  [12].

Finally, the learning parameter,  $\eta_l$  (Eq 3.81), and the synaptic weights,  $a_l$  (Eq 3.82), are updated in the top-level as in the ME algorithm. The best fit or most probable model which matches the statistics of the input is the macromode with the highest TLGN gating weight and the node within that macromode with the highest GN gating weight.

### 3.3 Dynamic/Process Models

This section summarizes the dynamic and process models utilized in the algorithms. These are the models utilized by the UKF to generate the accelerations used to propagate the state in time.

### 3.3.1 Solar Radiation Pressure

The solar radiation pressure (SRP) model used in the simulation is from [51].

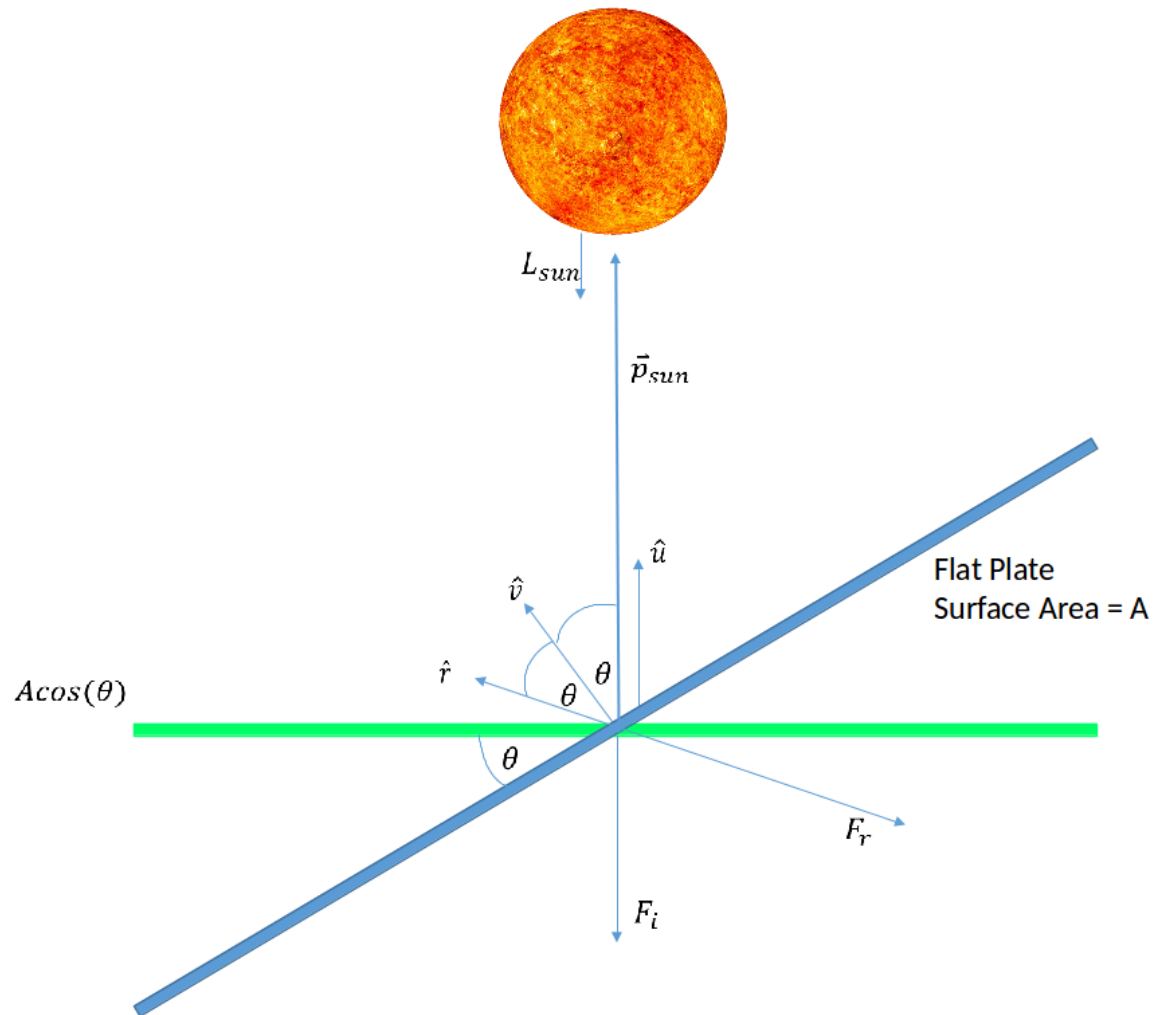


Figure 3.4 Solar Radiation Pressure Geometry

SRP is a pressure force exerted on a body by electro-magnetic (EM) radiation generated by the Sun. It is thus directly proportional to the solar flux and the exposed area to the EM radiation along with the surface material properties of the RSO. The acceleration due to SRP is shown in (Eq 3.83). This is a summation of the acceleration computed for each

flat plate in the model, where  $S$  is the total number of sides. This model assumes a constant solar luminosity and uses the average accepted value of  $L_{sun} = 3.828 \times 10^{26} J/s$  [52]. The solar flux reaching the object is  $W = L_{sun}/4\pi |\vec{p}_{sun}|^2$ , where  $\vec{p}_{sun}$  is the full range vector from the object to the Sun.  $m$  is the mass of the object and  $c = 2.99792458 \times 10^8 m/s$  is the speed of light in a vacuum. In an operational system, a more accurate model of a dynamic solar flux would be used along with corrections from space weather monitoring services. In addition, this model neglects the albedo of the Earth as well as infrared radiation from the Earth.

$$a_{srp_i} = \frac{-W}{mc} A_i \cos^2(\theta_i) \left[ 2\delta\beta\hat{r}_i + \frac{2}{3}(1-\delta)\beta\hat{v}_i + (1-\delta\beta)\hat{u}_i \right] \quad (3.83)$$

$$a_{srp} = \sum_i^S a_{srp_i} \quad (3.84)$$

The geometry of a flat plate with SRP can be seen in Figure 3.4.  $\hat{u}$  is a unit vector of  $\hat{p}_{sun}$  which points toward the Sun. The  $\hat{v}$  is the unit vector normal to the surface of the flat plate. The positive normal is defined in the shape model and represents outside the RSO. The  $\hat{r}$  vector is the direction the specular reflectance ray would take if the flat plate were a perfect mirror. In addition, the angle,  $\theta$ , between  $\hat{u}$  and  $\hat{v}$  is the same as that between  $\hat{v}$  and  $\hat{r}$ . The projected area is  $A \cos(\theta)$ . If the cosine of the incident angle,  $\cos(\theta)$ , is less than zero or the RSO is within Earth's shadow, then the object is not facing the Sun and that side is not used in the SRP calculation.

Reflected light is separated into specular and diffuse reflection as shown in Figure 3.5. Specular reflection is the reflection expected from mirrors and other highly reflective smooth surfaces. While diffuse reflection is the reflection expected from subsurface reflection. When modeling reflectance it is useful to determine the fraction of light which experiences specular

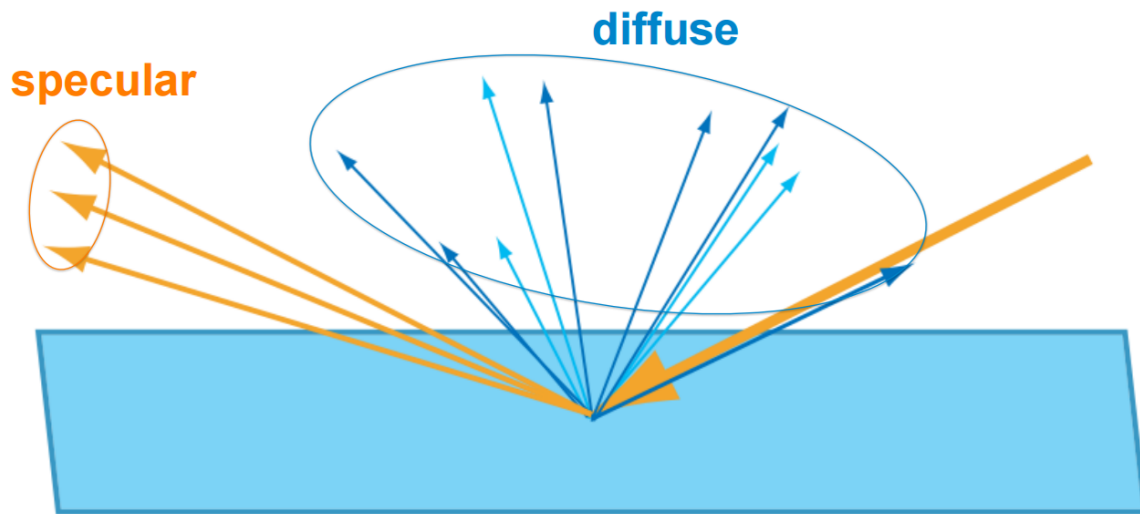


Figure 3.5 Specular vs Diffuse Reflectivity  
[53]

and diffuse reflection with the remaining light being absorbed.

The total force applied is a combination between the amount of light specular reflection and diffuse reflection versus the amount that is absorbed.  $\beta$  is the coefficient of reflection or the fraction of total light being reflected, which includes both specular and diffuse reflection. Thus,  $1 - \beta$  is the fraction of total light absorbed.  $\delta$  is the ratio of specular versus diffuse reflection,  $\delta = R_{spec}/R_{spec} + R_{diff}$ .  $R_{spec} = \{x \in \mathbb{R} : 0 \leq x \leq 1\}$  is a color that specifies specular reflectance at normal incidence.  $R_{diff} = \{x \in \mathbb{R} : 0 \leq x \leq 1\}$  is a color that specifies diffuse reflectance of the substrate under the specular coating. Thus,  $\delta\beta$  is the fraction of light that is specular reflection and  $(1 - \delta)\beta$  is the fraction of light that is diffuse reflection. Then, algebra shows that the totality of light is a combination of the specular reflection, diffuse reflection, and absorption:  $\delta\beta + (1 - \delta)\beta + (1 - \beta) = 1$ .

The SRP induces a moment on the RSO body which causes torque:

$$T_{srp_i} = (mI_B\hat{v}_i) \times (d_{BI}a_{srp_i}) \quad (3.85)$$

$$T_{srp} = \sum_{i=1}^S T_{srp_i} \quad (3.86)$$

In (Eq 3.85),  $m$  is the mass of the RSO.  $I_B$  is the geometric center of each side with respect to the center of mass in the body frame. In this simplified model, the RSO is a rectangular cuboid with constant density.  $\hat{v}_i$  is the outward facing unit normal vector to the surface, in RSO body frame.  $d_{BI}$  is the direction cosine matrix (DCM) which rotates from ECI frame to RSO body frame. Finally,  $a_{srp}$  is the acceleration due to SRP.

### 3.3.2 Gravity

This model uses a 6<sup>th</sup> degree zonal model for gravity.

$$U = \frac{\mu}{r} \left[ 1 - \sum_{n=2}^6 J_n \left( \frac{a}{r} \right)^n P_n(\sin \theta') \right] \quad (3.87)$$

[54, 55]

The gravitational aspherical potential function is shown in (Eq 3.87).

$$\mu = GM_{\oplus} = 3986004.418 \times 10^8 \frac{m^3}{s^2} \quad (3.88)$$

Earth's gravitational constant value is shown in (Eq 3.88), where  $G$  is the universal gravitational constant and  $M_{\oplus}$  is the mass of Earth [56]. The magnitude of the vector from Earth's center of mass to the RSO's center of mass is  $r = |\hat{r}|$ . The semi-major axis of the earth is  $a = 6.3781370 \times 10^6$  meters [56].  $\theta'$  is the geocentric latitude of the RSO along

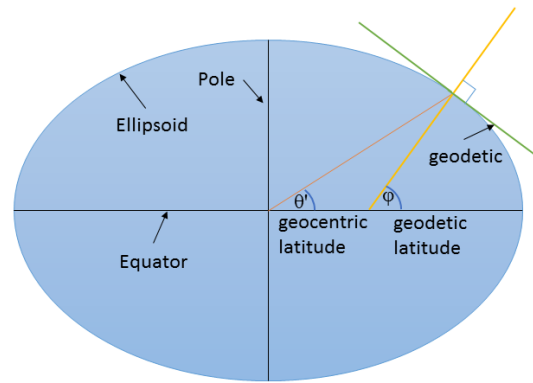


Figure 3.6 Geocentric Latitude

$$C_{2,0} = -.484165371736 \times 10^{-3}$$

$$C_{3,0} = .957254173792 \times 10^{-6}$$

$$C_{4,0} = .539873863789 \times 10^{-6}$$

$$C_{5,0} = .685323475630 \times 10^{-7}$$

$$C_{6,0} = -.149957994714 \times 10^{-6}$$

---

Table 3.12 Normalized Legendre Zonal Coefficients  
[56]

the  $\hat{r}$  vector. Figure 3.6 shows a greatly exaggerated ellipsoid to point out the difference between the geocentric latitude and the geodetic latitude.

$J_n$  are the denormalized zonal spherical harmonic coefficients mathematically modeling Earth's shape. The computation of  $J_n$  starts with getting the normalized Legendre zonal coefficients,  $C_n$ , for zonal order 0 and degree  $n = \{2, 3, 4, 5, 6\}$ . These values are listed in Table 3.12.

The algorithm to convert from normalized zonal spherical harmonic coefficients,  $C_n$ , to denormalized,  $J_n$ , can be found in (Eq 3.89). Where,  $n = \{2, 3, 4, 5, 6\}$  is the degree,

$m = 0$  is the order, and  $k = 1$  is a coefficient related to the order.

$$J_{n,0} = \frac{C_{n,0}}{\left(\frac{(n+m)!}{(n-m)k(2+n+1)}\right)^{\frac{1}{2}}} \text{ for } n \geq 2 \quad (3.89)$$

[54]

$P_n$  is the  $n^{\text{th}}$  Legendre polynomial function. The computation of the zonal Legendre polynomial functions are shown in (Eq 3.93).

$$\gamma = \sin \theta' \quad (3.90)$$

$$P_1(\gamma) = 1 \quad (3.91)$$

$$P_2(\gamma) = \gamma \quad (3.92)$$

$$P_n(\gamma) = \frac{(2n-1)\gamma P_{n-1}(\gamma) - (n-1)P_{n-2}(\gamma)}{n}, \text{ for } n \geq 2 \quad (3.93)$$

### 3.3.3 State Propagation

Let  $\vec{x}$  be the state vector as shown in (Eq 3.94).  $\vec{q}$  is the quaternion representing an attitude vector on the unit sphere in four-dimensional space and represents a rotation from ECI to RSO body frame.  $\vec{\omega}$  is the attitude rate vector of the three Euler angles of  $[\psi, \theta, \phi]$  or yaw, pitch, roll in RSO body frame.  $\hat{r}$  is the position vector in Cartesian coordinates in geocentric pseudo-inertial ECI frame.  $\dot{\hat{r}}$  is the velocity vector also in Cartesian coordinates in geocentric pseudo-inertial ECI frame.

$$\vec{x} = \begin{bmatrix} \vec{q} \\ \vec{\omega} \\ \vec{r} \\ \dot{\vec{r}} \end{bmatrix} \quad (3.94)$$

The differential equation showing how the state changes with time is shown in equation (Eq 3.95). The quaternion dynamics are shown in (Eq 3.96), attitude rate dynamics in (Eq 3.97), there are no position dynamics in this differential equation (Eq 3.98), and finally the velocity dynamics are shown in (Eq 3.99).

$$\dot{\vec{x}} = \begin{bmatrix} \dot{\vec{q}} \\ \dot{\vec{\omega}} \\ \dot{\vec{r}} \\ \ddot{\vec{r}} \end{bmatrix} \quad (3.95)$$

$$\dot{\vec{q}} = \frac{1}{2}\xi\omega \quad (3.96)$$

$$\dot{\vec{\omega}} = J^{-1}(T_{srp} - \omega \times (J\omega)) \quad (3.97)$$

$$\dot{\vec{r}} = \dot{\vec{r}} \quad (3.98)$$

$$\ddot{\vec{r}} = a_g + a_{srp} \quad (3.99)$$

$$J = \frac{m}{12} (s_2^2 + h^2, s_1^2 + h^2, s_1^2 + s_2^2) \quad (3.100)$$

The quaternion dynamics in (Eq 3.96) is  $1/2$  times the quaternion matrix times the attitude rate. An approximation is taken in this model that the attitude rate is constant during the delta time,  $\Delta t$ , of integration. A more complicated and more accurate model could be used to take into account the changing attitude rate. The attitude matrix is the standard matrix representation of a quaternion. The derivation of the attitude quaternion dynamics is explained in more detail in reference [57].

The attitude rate equation is described in (Eq 3.97).  $J$  is the moment of inertia  $3 \times 3$  diagonal tensor in RSO body and its computation is shown in (Eq 3.100).  $m$  is the mass of the RSO,  $s$  is the length of sides 1 and 2, and  $h$  is the length of the height dimension. The attitude rate equation is simply the inverse of the moment of inertia tensor times the sum of the torque applied by solar radiation pressure,  $T_{srp}$ , and the outer product of the current angular rate and the product of the moment of inertial matrix and the current angular rate.

There are no position dynamics or velocity update in this differential equation model. The velocity and position are updated via numerical integration of the accelerations.

The velocity dynamics or acceleration update is simply the sum of accelerations due to the gravitational potential of the Earth and solar radiation pressure.

The dynamics are then integrated between measurement update time  $dt$  in seconds. The numerical integration algorithm used here is the RK4 (fixed step size, 4<sup>th</sup> order) Runge-Kutta methods as shown in (Eq 3.101).

$$x_{n+1} = x_n + \frac{\Delta t}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (3.101)$$

$$k_1 = f(x_n, m) \quad (3.102)$$

$$k_2 = f\left(x_n + k_1 \frac{\Delta t}{2}, m\right) \quad (3.103)$$

$$k_3 = f\left(x_n + k_2 \frac{\Delta t}{2}, m\right) \quad (3.104)$$

$$k_4 = f(x_n + k_3 \Delta t, m) \quad (3.105)$$

$$f(x_n, m) = \dot{\vec{x}} \quad (3.106)$$

[58]

$x_n$  is the current state,  $x_{n+1}$  is the propagated state.  $\Delta t$  is the time between the last propagation and the current time. Finally,  $m$  is the additional model parameters such as the features and characteristics which we are estimating and searching for. This can include the shape models, mass and density models, albedo (reflectivity) models, maneuver estimates, etc.

### 3.4 Measurement Models

The UKFs in the simulation take as measurement inputs the photometric and astrometric data of an RSO from non-resolved observations in GEO. The observations are modeled as being taken from a telescope and a charge coupled device (CCD) camera. The simulation makes use of the Ashikhmin-Shirley bidirectional reflectance distribution function for the albedo model along with the standard computation of the apparent bolometric magnitude. Finally, a WGS-84 model is used to convert topographic azimuth and elevation to geocentric psuedo-inertial ECI measurements.

### 3.4.1 Bidirectional Reflectance Distribution Model

The BRDF determines how reflected radiance is distributed in terms of the distribution of incident radiance [33]. The BRDF was developed by the computer graphics community to synthesize a realistic image of a scene to the viewer [33]. The BRDF is a function of two vectors, the incident light ray striking the object and the vector from the observer to the object. The BRDF contains parameters related to the material reflectance properties. A few assumptions of the BRDF are:

- light is not polarized
- fluorescence is not modeled
- phosphorescence is not modeled

Fluorescence is the electromagnetic radiation emitted with a longer wavelength as a result of incident radiation. Phosphorescence is light emitted by an object similar to fluorescence but on a longer timescale.

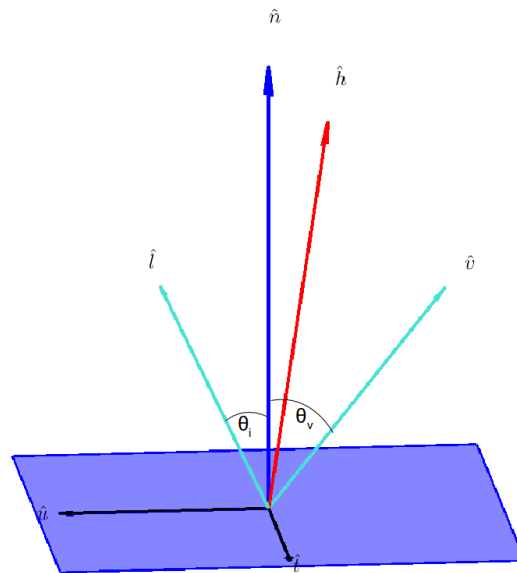


Figure 3.7 BRDF Geometry

The basic geometry of light striking a single point on a flat surface can be seen in Figure 3.7. In this figure,  $\hat{l}$  is the vector to the incident light source,  $\hat{v}$  is the vector to the viewer,  $\hat{h}$  is the halfway vector between  $\hat{l}$  and  $\hat{v}$ ,  $\hat{n}$  is the surface unit normal vector pointing to the outside of the RSO, and  $\hat{u}$  and  $\hat{t}$  represent tangent vectors on the surface. Vectors  $\hat{u}$ ,  $\hat{t}$ , and  $\hat{n}$  form a local right-handed orthogonal coordinate system. Vectors  $\hat{l}$ ,  $\hat{h}$ , and  $\hat{v}$  all lie in the same plane which usually does not include  $\hat{n}$ .

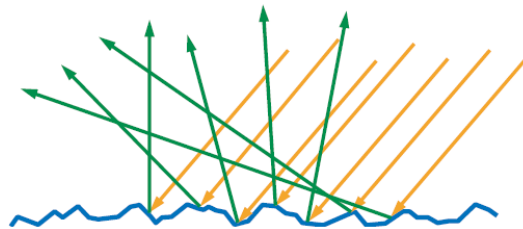


Figure 3.8 Microfacet Rough Surface

[53]

The BRDF models typically differentiate between specular and diffuse reflectivity. A

simple example of the two is shown in Figure 3.5. Diffuse reflection is usually modeled as reflected in nearly all outside directions and is a function of both surface roughness and subsurface reflection. Microfacet theory models the surface as non-optically flat where the roughness is small to the observer but large to the wavelength of light. Then each point is modeled as a perfect mirror and reflects the incoming incident light into a single outgoing direction as shown in Figure 3.8. A source of diffuse reflectivity is from subsurface reflection as shown in Figure 3.9. In this figure, the larger orange line represents the incident light ray, the three orange reflected rays represent the specular reflection, and the scattering of blue light rays represents the diffuse reflection.

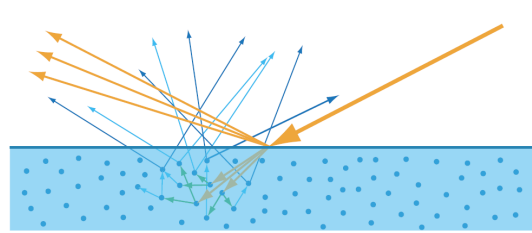


Figure 3.9 Diffuse reflectance of the substrate under specular coating [53]

$$f_s(\hat{l}, \hat{v}) = \frac{F(\hat{l}, \hat{v}) G(\hat{l}, \hat{v}, \hat{h}) D(\hat{h})}{4(\hat{n} \cdot \hat{l})(\hat{n} \cdot \hat{v})} \quad (3.107)$$

$f_s(\hat{l}, \hat{v})$  in (Eq 3.107) is the general microfacet specular BRDF, where the vectors are from Figure 3.7.

$F(\hat{l}, \hat{v})$  is the Fresnel reflectance function. Fresnel reflectance is the fraction of incoming light that is specularly reflected from an optically flat surface of a given substance. Fresnel reflectance is dependent on the angle of incidence,  $\theta_i$ , which itself is dependent on the light source vector and the surface (e.g. microfacet) normal. Fresnel reflectance for different

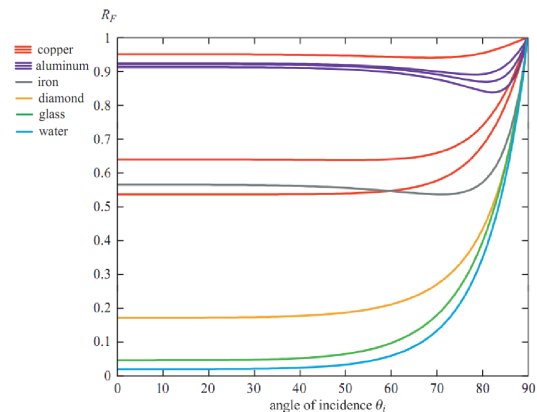


Figure 3.10 Fresnel Reflectance

[59]

materials as a function of  $\theta_i$  is shown in Figure 3.10. Copper and aluminum have colored reflectance and the three lines represent the channels of red, green, and blue, while the other items have uncolored reflectance. The Schlick approximation to the Fresnel is used in many applications due to it being both computationally cheap and its high accuracy [33] and is shown in (Eq 3.108).  $R_s$  is a color, with range:  $R_s = \{x \in \mathbb{R} : 0 \leq x \leq 1\}$ , that specifies the specular reflectance at normal (e.g.  $90^\circ$ ) incidence.

$$F_s(R_s, \hat{l}, \hat{h}) = R_s + (1 - R_s) \left(1 - (\hat{l} \cdot \hat{h})\right)^5 \quad (3.108)$$

$D_{\hat{h}}$  is the normal distribution function which gives the amount of microfacet normals pointing in the half angle direction,  $\hat{h}$ , relative to the surface area.  $D_{\hat{h}}$  determines the size and shape of the reflected highlight.

$G(\hat{l}, \hat{v}, \hat{h})$  is the geometry or shadow masking functions that gives the probability that a given microfacet is visible to the observer based on the BRDF geometry. This attempts to take into account things like one microfacet bump masking another behind it, shadowing

effects from different microfacets, etc.

The simulation uses the Ashikhmin-Shirley BRDF model which uses the Schlick approximation to the Fresnel and defines its normal distribution function and geometry function for the BRDF. Some useful properties of the Ashikhmin-Shirley model are its relatively low computational costs, the use of an anisotropic microfacet model that uses simple and intuitive parameters, the use of a diffuse term that guarantees reciprocity and energy conservation laws, and thus is a plausible model [33, 60]. This model uses a weighted sum of the diffuse and specular reflection. The Ashikhmin-Shirley BRDF model components of specular and diffuse reflectance as well as the combined components are shown in (Eq 3.109), (Eq 3.110), (Eq 3.111) respectively. All of the vectors in the equations are from Figure 3.7 with  $R_s$  being a color that specifies specular reflectance at normal incidence and  $R_d$  being a color that specifies the diffuse reflectance of the substrate under specular coating as shown in Figure 3.9.

$$f_s(\hat{l}, \hat{v}) = \frac{\sqrt{(n_u + 1)(n_t + 1)}}{8\pi} \frac{(\hat{n} \cdot \hat{h})^{\frac{n_u(\hat{h} \cdot \hat{u})^2 + n_t(\hat{h} \cdot \hat{t})^2}{1 - (\hat{h} \cdot \hat{n})^2}}}{(\hat{h} \cdot \hat{l}) \max((\hat{n} \cdot \hat{l}), (\hat{n} \cdot \hat{v}))} F_s(R_s, \hat{l}, \hat{h}) \quad (3.109)$$

$$f_d(\hat{l}, \hat{v}) = \frac{28R_d}{23\pi} (1 - R_s) \left( 1 - \left( 1 - \frac{\hat{n} \cdot \hat{l}}{2} \right)^5 \right) \left( 1 - \left( 1 - \frac{\hat{n} \cdot \hat{v}}{2} \right)^5 \right) \quad (3.110)$$

$$f(\hat{l}, \hat{v}) = f_s(\hat{l}, \hat{v}) + f_d(\hat{l}, \hat{v}) \quad (3.111)$$

### 3.4.2 Charge Coupled Device Noise Model

The simulation models a medium professional telescope with a mid-range to high-end professional charge coupled device (CCD) camera. Thus, the telescope is in the 1.0 - 0.5 meter main mirror diameter range and the class of CCD modeled is the SBIG STT-8300M

[61]. This section very briefly discusses the signal to noise ratio for the selected telescope and CCD camera. A more detailed analysis of this noise model can be found in references [62, 63] among others.

Primary sources of noise for a CCD sensor are:

- Shot (photon) Source Noise
- Shot Sky Noise
- Readout Noise (RN)
- Dark Current Noise

There are many other sources of potential CCD sensor noise. The list of potential noise is virtually infinite. However, these are the primary sources typically considered [64–66] and the only ones considered for the simulation developed for this thesis.

Shot or photon source noise is a combination of the inherent statistical rate of photons reaching the detector and the percentage of those photons that get converted to electrons. This is a countable statistic and thus is represented by Poisson statistics, thus the square root of the signal is equal to the noise:  $\sigma_{SN} = \sqrt{signal}$ . Shot noise is measured in number of electrons,  $e^-$ . The signal is computed by normalizing the expected photon flux for a magnitude 20 object along with the camera's quantum efficiency ( $Q_e$ ) of converting those photons to electrons times the integration (e.g. dwell) time of the sensor as seen in (Eq 3.112) - (Eq 3.116) .

$$E_p = \frac{\lambda}{hc} \quad (3.112)$$

$$A = \pi \left( \frac{d_{tel}}{2} \right)^2 \quad (3.113)$$

$$V_{20} = I_{20} E_p A^2 Q_e \quad (3.114)$$

$$V_m = V_{20} 10^{0.4(20-m)} \quad (3.115)$$

$$N = V_m t \quad (3.116)$$

$E_p$  is the energy per photon measured in Joules.  $\lambda$  is wavelength of light in meters, the simulation modeled light with a wavelength in the visible light area of the EM spectrum centered at 551 nm.  $h$  is Planck's constant, which is  $6.626070040 \times 10^{-34} m^2 kg/s$ .  $c$  is the speed of light in a vacuum which is  $2.99792458 \times 10^8 m/s$ .

$A$  is the area of the primary mirror of the telescope in meters. The simulation in this work modeled a telescope with a one meter diameter main mirror.

$V_{20}$  is the electron flux expected from a magnitude 20 object with the selected instruments.  $A$  is the telescope's main mirror area.  $I_{20}$  is the irradiance of a magnitude 20 object. This model used a value of  $I_{20} = 2.518021001999993 \times 10^{-16} W/m^2$ .  $E_p$  is the photon energy computed in (Eq 3.112).  $Q_e$  is the CCD camera's quantum efficiency.  $Q_e$  is a non-dimensional statistical probability from  $[0 \dots 1]$  of photons exciting electrons in the array. Based on the selected CCD camera and the expected wavelength to be detected, this model used a  $Q_e = 0.56$  [61]. There exists high-end CCD cameras with  $Q_e$  above 0.9, which would improve detections dramatically [64, 67].  $V_{20}$  is measured in  $e^-/s$  or number of electrons per second.

$V_m$  is the expected electron flux from an object with magnitude  $m$ .

$N$  is the number of expected electrons detected at the CCD sensor in number of elec-

trons,  $e^-$ , and represents the pure signal.

There are many factors which can affect the brightness of the night sky including human generated sources such as artificial lights and natural sources such as weather, phase of the moon and its location in the sky, altitude, etc. Shot or photon sky noise is the noise generated due to the glow of the night sky. This factor represents the uncertainty in sky brightness per pixel and is measured in  $e^-/s$ . The simulation assumed a remote mountain top near Keck observatory atop Mauna Kea, HI and thus the apparent magnitude of a moonless sky is about 22.6 [67]. The night sky atop Mt. Lemmon, Mt Wrightson, and other remote locations near Tucson, AZ can also be this dark [68]. Thus, the shot sky noise is  $V_{sky} = V_{20}10^{0.4(20-22.6)}$ .

Readout noise is a combination of converting CCD charge carriers into voltage signal for quantification and the conversion from analog to digital. Manufacturers of a CCD camera typically list the root mean square (RMS) of the readout noise. The major source of readout noise is from the on-chip preamplifier. The experiments run in the simulation modeled a readout noise with a RMS of  $9.3e^-/px$  [61]. The readout noise is typically Gaussian in nature [69].

Dark current noise is the statistical variation of thermally generated electrons within the silicon constituting the CCD. Dark current noise describes the rate of generation of thermal electrons at a given CCD temperature. The dark current noise is Poisson in nature and is typically very low for cooled CCD cameras. A value of  $0.02e^-/s/px$  was used in the simulation for dark current noise [61].

Thus, the total noise of this model is shown in (Eq 3.117). Recall that  $N$  is the source signal and thus  $\sqrt{N}$  is the shot source noise since it is Poisson distributed,  $n$  is the number of pixels,  $t$  is the integration or dwell time in seconds,  $D$  is the dark current noise,  $V_{sky}$  is the shot sky noise, and  $R$  is the readout noise.  $S_n$  in (Eq 3.118) represents the signal

over noise and is the signal,  $N$ , over the uncertainty noise. (Eq 3.119) shows the one-sigma uncertainty of the magnitude which is then used for a random draw from a Gaussian distribution every time step in the experiments. The apparent magnitude uncertainty ends up being a function of the intensity of the object as viewed from the observer and thus changes every time step.

$$\sigma = \sqrt{N + n(V_{sky}t + Dt + R)} \quad (3.117)$$

$$S_n = \frac{N}{\sigma} \quad (3.118)$$

$$\sigma_{mag} = 2.5 \log_{10} \left( 1 + \frac{1}{S_n} \right) \quad (3.119)$$

### 3.4.3 Apparent Magnitude Model

The apparent magnitude system is used to repeatedly transform photometric measurements into radiative luminosities and irradiances. Initially the apparent magnitude system was a shift from the Sun's apparent magnitude. However, the Sun's luminosity is not constant. The newer bolometric apparent magnitude system performs the goal of the traditional apparent magnitude system independent of the Sun.

The International Astronomical Union (IAU) defines the standard of the bolometric apparent magnitude system. The bolometric apparent magnitude defines the zero point of the absolute bolometric apparent magnitude scale,  $M_{Bol} = 0$ , by specifying that this radiation source has a radiative luminosity of exactly  $L_0 = 3.0128 \times 10^{28} W$  [52]. Next, it specifies the absolute bolometric apparent magnitude of a radiation source with radiative

luminosity of  $L$  as

$$M_{bol} = -2.5 \log_{10} \left( \frac{L}{L_0} \right) \quad (3.120)$$

$$= -2.5 \log_{10} L + 71.197425757 \quad (3.121)$$

This value corresponds to a nominal solar luminosity of  $L_{\odot} = 3.828 \times 10^{26} W$ , which corresponds to an absolute bolometric magnitude of  $M_{bol\odot} = 4.74$  [52].

The IAU also defines the zero point of the apparent bolometric magnitude scale by specifying that  $m_{bol} = 0$  corresponds to an irradiance of  $f_0 = 2.518021002 \times 10^{-8} W/m^2$ . Thus the apparent bolometric magnitude  $m_{bol}$  for an irradiance of  $f$  ( $W/m^2$ ) is:

$$m_{bol} = -2.5 \log_{10} \left( \frac{f}{f_0} \right) \quad (3.122)$$

$$= -2.5 \log_{10} f - 18.99735162976 \quad (3.123)$$

This zero point irradiance corresponds to an isotropically emitting radiation source with bolometric absolute magnitude of  $M_{bol} = 0$  and luminosity  $L_0$  at the standard apparent magnitude distance of 10 parsecs or approximately 32.6156 light years.

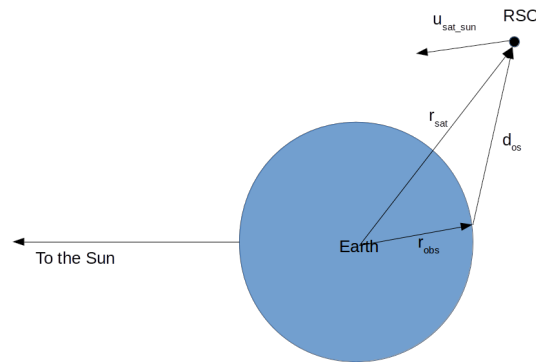


Figure 3.11 Basic Earth RSO Sun Geometry

Thus, this model first computes the irradiance of the sun at the RSO's current position:  $l_{\odot} = L_{\odot}/4\pi|\hat{r}_{sun}|^2$ , with a nominal solar luminosity of  $L_{\odot} = 3.828 \times 10^{26} W$ .  $\hat{r}_{sun}$  is the vector from the RSO to the Sun. Next, it determines the total reflectance of all sides facing the radiation source of the Sun by running the BRDF model and computing  $f(\hat{l}, \hat{v})$ . Next, the irradiance of the RSO as viewed from the observer is thus shown in (Eq 3.124).

$$l_{sat} = l_{\odot} \sum_{i=1}^S \left( \frac{f(\hat{l}, \hat{v})_i \theta_{I_i} A_i \theta_{obs-n_i}}{d_{os}^T d_{os}} \right) \quad (3.124)$$

Figure 3.11 shows the basic Earth-RSO-Sun geometry and shows the  $\vec{d}_{os}$  observer to RSO vector.  $S$  is the total number of outward faces of the RSO. Thus for a rectangular cuboid,  $S = 6$ . The BRDF  $f$  reflectance is computed for each side.  $\theta_{I_i}$  is the incident angle, or the angle between the  $\hat{u}_{sat-sun}$  as shown in Figure 3.11 and surface normal. This incident angle is shown in Figure 3.7 as the angle from the incident light ray and the outside surface normal vector. If the  $\theta_{I_i} < 0$ , then the BRDF for that face is set to exactly zero as it contributes nothing to the total irradiance of the RSO.  $A_i$  is the surface area of each face of the RSO.  $\theta_{obs-n_i}$  is the dot product (or cosine of the angle) between the surface outside normal vector and the unit vector from the observer to the RSO,  $\hat{d}_{os}$ . Thus,  $\theta_{obs-n_i} = \hat{n}_i \cdot \hat{d}_{os}$ . All of the vectors in this computation are rotated to the ECI frame for consistency and simplicity.

The irradiance of the RSO as viewed from the observer,  $l_{sat}$ , is then put through (Eq 3.123) to get the truth bolometric apparent magnitude. Then the  $1-\sigma$  apparent magnitude is computed from (Eq 3.119), a random draw is computed, and the error added to the truth apparent magnitude for the measured apparent magnitude which now includes the noise term.

### 3.4.4 Angle Measurement Model

The angle measurements start with the geometry shown in Figure 3.11. The truth ECI position is converted to topocentric azimuth and elevation via a transformation as shown in (Eq 3.125) - (Eq 3.129).

$$\vec{d}_{os} = \vec{r}_{sat} - \vec{r}_{obs} \quad (3.125)$$

$$\theta_{site} = \theta_G + \lambda \quad (3.126)$$

$$\rho_{uen} = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta_{site} & \sin \theta_{site} & 0 \\ -\sin \theta_{site} & \cos \theta_{site} & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{d}_{os} \quad (3.127)$$

$$az = \arctan \frac{\rho_{uen}(2)}{\rho_{uen}(3)} \quad (3.128)$$

$$el = \arcsin \frac{\rho_{uen}(1)}{|\rho_{uen}|} \quad (3.129)$$

Again,  $\vec{d}_{os}$  is the full vector from the observer to the RSO in ECI, while  $\hat{d}_{os}$  is the unit vector.  $\theta_G$  is the sidereal time and its formula can be found in many sources including Algorithm 15 in reference [70].  $\lambda$  and  $\phi$  are the geodetic longitude and latitude respectively. The azimuth equation is shown in (Eq 3.128). When using computer systems to compute the arctan function, it is preferable to use the **atan2** function to get the proper quadrant of the resultant angle. The elevation equation is shown in (Eq 3.129).

The truth values then have noise drawn from a Gaussian distribution with the selected 1- $\sigma$  user selected noise values for the angles to give the measurement angles with noise

added. The noise values selected for the simulation are listed below:

$$\sigma_{az} = 0.1'' \text{ (arcseconds)} \quad (3.130)$$

$$\sigma_{el} = 0.1'' [15] \quad (3.131)$$

### 3.5 Use of HME with UKF Experts

This section goes into some detail into how the HME and UKF were setup and used.

#### 3.5.1 UKF Details

The experiments used in this work used the UKF described in Section 3.1.2 and the options of where both the process and measurement noise sources are additive and independent. Thus, the simulation uses the algorithm with the unaugmented state listed in Table 3.9. Tuning lead to the UKF parameters of:

- $\alpha = \sqrt{3}$
- $\beta = 2$
- $\kappa = 1$
- $\lambda = 1$

Tuning found that just setting  $\lambda = 1$  as opposed to the formula in (Eq 3.16). This means that the  $\kappa$  parameter is not used. This work followed the  $\lambda$  tuning used in reference [24]. More details as to how the tuning values were selected are discussed in Section 4.1. As mentioned in Section 3.3.3, the state consists of attitude quaternion, attitude rates, position, and velocity. This state vector has 13 components while the error state covariance has dimension  $12 \times 12$ . The difference is due to the state containing 4-vector quaternions,

while the state error covariance contains the 3-vector Euler angle error states. The number of states is  $L = 12$  for the number of states in the UKF as this number represents the number of fundamental state components and this is the dimension of the state error covariance. The quaternion is used during propagation due to its favorable properties. The equations of state propagation are described in Section 3.3.3.

### 3.5.2 UKF Attitude Details

The quaternion sigma points are constructed using quaternion multiplication with the sigma points.

Euler angles are a popular representation of how the spacecraft is oriented or rotated relative to another outside frame such as ECI for objects in Earth orbit. The three angles are typically represented as  $[\psi, \theta, \phi]$  or yaw, pitch, roll and represent rotations to the axis,  $[x, y, z]$ , of this outside frame. The use of pure Euler angles for attitude propagation is discouraged due to issues such as discontinuities and a phenomenon known as a ‘gimbal-lock’ singularity when the middle angle ( $\theta$  or pitch) is  $90^\circ$  [71]. To avoid this and other potential singularities, one solution is to use a quaternion while another is to use the DCM (or attitude matrix) as shown in [21]. The DCM is a rotation matrix that represents the transformation from RSO body to another outside coordinate system. This work used quaternions as it is typically simpler to restore quaternion normalization than it is to restore DCM orthogonality [21].

Quaternions are generalizations of Euler angles without suffering from discontinuities nor singularities. A quaternion is defined as:  $\bar{q} = q_4 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ , where  $\mathbf{i}, \mathbf{j}, \mathbf{k}$  are imaginary coordinates in 4D space. Let  $\mathbf{q} = [q_1, q_2, q_3]^T = [\mathbf{e} \sin \theta/2]^T$ , where  $\mathbf{e}$  is the Euler axis and  $\theta$  is the rotation angle about that axis. Now,  $q_4 = \sqrt{1 - |\mathbf{q}|^2}$  satisfying the quaternion normalization requirement that the magnitude of the quaternion is exactly one,

$|\bar{q}| = 1$ . Quaternions describe their own algebra which is described in nice detail in reference [57]. One draw back of quaternion algebra is that it is not closed in addition. However, it is closed in multiplication and thus does not need to be renormalized. The simulation makes use of that property.

The sigma points are constructed a little bit differently for the quaternion components as shown in (Eq 3.132) and (Eq 3.133). The other components of the sigma points (e.g. angular rates, position, and velocity) are constructed as in (Eq 3.24) and (Eq 3.25).

$$\chi_i(1:4) = \hat{x}(1:4) \otimes \left( \sqrt{(L+\lambda)P_{xx}(1:4)} \right)_i \quad i = \{1, \dots, L\} \quad (3.132)$$

$$\chi_{i+L}(1:4) = \hat{x}(1:4) \otimes \left( -\sqrt{(L+\lambda)P_{xx}(1:4)} \right)_i \quad i = \{1, \dots, L\} \quad (3.133)$$

The symbol  $\otimes$  is meant here to represent quaternion multiplication. After the quaternion components are constructed in (Eq 3.132) and (Eq 3.133), they are normalized. The mean point,  $\chi_0$ , and  $2L$  sigma points are propagated through the state time update portion of the UKF as shown in (Eq 3.45). After the state propagation update, the quaternion components of the mean and sigma points are renormalized again.

The next step in the UKF algorithm is computed slightly differently for the quaternion components of the mean and sigma points as shown in (Eq 3.135).

$$\frac{\delta\theta_i}{2} = (\hat{\chi}_0(1:4) \otimes \hat{\chi}_i(1:4))(2:4) \quad (3.134)$$

$$\hat{x}_{x+1}^-(1:4) = \frac{\hat{\chi}_0(1:4) \otimes \left[ 1 \quad \sum_{i=1}^{2L} w_i \frac{\delta\theta_i}{2} \right]}{\left| \hat{\chi}_0(1:4) \otimes \left[ 1 \quad \sum_{i=1}^{2L} w_i \frac{\delta\theta_i}{2} \right] \right|} \quad (3.135)$$

(Eq 3.134) only take the last three components of the quaternion multiplication between the mean point and the sigma points. Note that a one is placed in the  $q_4$  (thus first) component

of the quaternion in quaternion multiplication terms of the numerator and denominator in (Eq 3.135). This is due to the small angle assumption of  $\delta\theta_i/2$  as described in much more detail in [24]. In addition, (Eq 3.135) shows explicit quaternion normalization. The rest of the states are updated normally as shown in (Eq 3.46).

The state error covariance matrix is also setup a bit differently for quaternions. First let the state differences be defined as:

$$\delta x_{i,k+1} = \begin{bmatrix} 2 \left[ \left\{ \hat{x}_{k+1}^- (1:4) \otimes \hat{\chi}_{i,k+1} (1:4) \right\} (2:4) \right] \\ \hat{\chi}_{i,k+1} (5:13) - \hat{x}_{k+1}^- (5:13) \end{bmatrix} \quad (3.136)$$

Note that  $\delta x_{i,k+1}$  has only 12 components which is consistent with the dimension of the error state covariance and  $L$ . Then the state error covariance is updated in a similar manner to that in (Eq 3.47):

$$P_{xx,k+1}^- = \sum_{i=0}^{2L} W_i [\delta x_{i,k+1}] [\delta x_{i,k+1}]^T + Q_k \quad (3.137)$$

The predicted measurement sigma points, the expected measurement from the propagated state, the measurement residual, and the innovation covariance matrix are all computed as per the standard UKF as shown in (Eq 3.48) - (Eq 3.51).

The cross correlation covariance matrix is computed using a combination of the delta states shown in (Eq 3.136) and the delta expected measurements:

$$P_{xy} = \sum_{i=0}^{2L} W_i [\delta x_{i,k+1}] \left[ \hat{\Psi}_{i,k+1} - \hat{y}_{k+1}^- \right]^T \quad (3.138)$$

The Kalman gain is computed exactly as in (Eq 3.53).

The state correction update is also a bit different due to the quaternion:

$$\frac{\delta\theta^*}{2} = K_k \tilde{y}_k \quad (3.139)$$

$$\delta q^* = \begin{bmatrix} 1 & \frac{\delta\theta^*}{2} \end{bmatrix} \quad (3.140)$$

$$\hat{x}_k(1:4) = \frac{\begin{bmatrix} \hat{x}_k^-(1:4) \end{bmatrix} \otimes \delta q^*}{\begin{bmatrix} \hat{x}_k^-(1:4) \end{bmatrix} \otimes \delta q^*} \quad (3.141)$$

$$\hat{x}_k(5:13) = \left\{ \hat{x}_k^- + K_k \tilde{y}_k \right\} (5:13) \quad (3.142)$$

Finally the state error covariance is corrected as per the standard UKF algorithm as shown in (Eq 3.55).

Many more details of using the Challa, Moore, and Rogers method of attitude quaternion with a UKF can be found in reference [24].

### 3.5.3 MMAE & Single-Layer ME Details

For both the MMAE and the single layer ME as described in sections Sections 3.2.1 and 3.2.2 respectively, the inputs are the measurement residuals,  $\tilde{y}_k^i$ , and measurement innovation covariance matrices,  $S_k^i$ . The outputs of these learning functions are not the state vector as described in the general MMAE and ME.

Rather, the simulation uses the output from the likelihood probability function for the MMAE,  $W_i$ . The model hypothesis with the highest likelihood probability is considered to be the most probable model and thus is selected,  $n = \text{Idx}[\max(W_i)]$ .

For the single-layer ME, the simulation uses the a-priori probability gating weight,  $g_i$ . Thus, the model hypothesis with the highest a-priori probability gating weight is considered to be the most probable model and is selected,  $n = \text{Idx}[\max(g_i)]$ .

The simulation then compares the selected models of the MMAE and single layer ME

and determines if it matches the best-match model of the experiment as is shown in chapter 4.

### 3.5.4 Dual-Layer HME Details

Recall from Section 3.2.3 that the dual-layer HME's first layer consists of a collection of macromodes. Each macromode may contain up to  $[x \in \mathbb{N} : 1 \dots M_l]$  UKF expert nodes or neurons, where  $M_l$  is the number of nodes in each macromode. In addition, recall that measurements are received from the sensor, which consists of a CCD camera attached to an optical telescope, and are the inputs to the UKF. The UKF processes these measurements along with propagating a state model as explained in sections Sections 3.1.2, 3.3, 3.4, 3.5.1 and 3.5.2. The  $\tilde{y}_k^i$  and  $S_k^i$  are then sent to the macromode's gating network and processed as in a single layer ME. The outputs of the ME, including the  $\tilde{y}_k^i$  and  $S_k^i$  for the selected node,  $n_l = \text{Idx}[\max(g_{l,i})]$ , are sent to the TLGN. Then the TLGN selects the macromode with the highest gating network,  $n = \text{Idx}[\max(g_l)]$ , to determine the most probable macromode. The HME details are described in Section 3.2.3.

One of the ways the simulation utilized the HME is to determine multiple characteristics of an RSO simultaneously. For example, to determine both the size and reflectivity three macromodes (MM) are setup:

- MM 1 - Nominal Model - this macromode contains only a single node - the nominal model. It starts with the initial guess and then if another characteristic is determined to have a higher probability of being the best-match and meets the threshold criteria described below, then its state is updated accordingly. The goal is to have the HME converge to the nominal model being the most probable model.
- MM 2 - Size Model - This macromode contains a node for every size hypothesis. It also contains the nominal model for reflectivity.

- MM 3 - Reflectivity Model - This macromode contains a node for every reflectivity hypothesis. It also contains the nominal model for size.

For MM 2 and 3, the nodes should cover the known or hypothesized parameter space. The density of the parameter space, or number of nodes, is determined based on the ability to grab the signal from the noise.

The threshold criteria where a different size or reflectivity model replaces the nominal model is user selected. Unless otherwise stated in the results, TLGN a-prior probability gating weight in the simulation is set to  $g_l > 0.95$  for 4 consecutive measurement update frames. This value is based on numerous simulation results.

For example, let the  $l = 2$  where  $g_{l=2} > 0.95$  for 4 consecutive frames. Then the system selects the index from macromode 2, the size model, that has the max gating weight,  $n_{l=2} = \text{Idx}(\max(g_{l=2,i}))$ , and replaces the size model component of the nominal model in MM 1 and the size model components of all the nodes in MM 3. It then resets all of the UKF filters with initialized state and covariance from node  $n_{l=2}$  in MM  $l = 2$ . In addition, the HME is reset, which includes resetting the time of the learning parameter. This resets the decay function and each node in a macromode has a balanced learning rate and weights again. Then the process continues to run. If no macromode reaches the threshold criteria or MM 1 reaches the threshold criteria, then the system does not react. It only reacts as shown if a macromode reaches or exceeds the threshold criteria.

Every time a new node within a specific macromode exceeds the  $g_l$  max threshold, 0.95 unless otherwise stated in the results, the TLGN resets. This has the effect of re-initializing all of the weights back to the balanced  $1/L$  value. This is to erase the history which contained a different node value. As will be seen in chapter 4, this method shows some promising results.

The goal of dual-layer HME is to have the the states converge to the the best fit states

and then have the TLGN gating weight for MM 1 be the most probable macromode.

### 3.6 Simulation

The work of this thesis began with MATLAB<sup>®</sup> codes from Dr. Linares [15]. Dr. Linares' codes was originally created to determine which rectangular cuboid model had the correct size parameters [15] using MMAE with a bank of UKFs. These codes were modified to support estimating reflectivity along with determining the time, magnitude, and direction of a maneuver in addition to size determination. The following additional changes were made to the original codes by Dr. Linares:

- Completely revamped the driver program which separated the driver called from the main algorithm thus allowing for the selection of experiments at runtime, the ability to run Monte Carlo results, and the ability to collect the output of the runs and call analysis tools to plot and grade the results.
- Added an input structure to easily allow multiple scenarios to be setup independently and easily; this allows for the driver program to select which experiment the user wants to run
- Added an output structure to allow for saving of simulation data for later analysis
- Added ability to set the random seed to ensure repeatable results
- Added ability to run Monte-Carlo sets
- Implemented the Challa, Moore, and Rogers quaternion propagation algorithm within the UKF [24] as opposed to the GRP method
- Modified the Ashikhmin-Shirley BRDF model to match [60]

- 
- Implemented a slightly different solar radiation pressure model based on [51]
  - Implemented an apparent magnitude model based on the bolometric apparent magnitude from the IAU
  - Implemented a dynamic apparent magnitude error model which varies the uncertainty based on CCD camera noise characteristics and the intensity of the RSO
  - Added the HME
  - Added a grader to grade the performance of the MMAE and HME models and many additional plots

The purpose of modifying the codes was to develop a simulation capable of modeling a RSO and its truth states, modeling the measurements, and using a MMAE, single layer ME, and a dual-layer HME along with banks of UKFs to estimate certain RSO characteristics.

## Chapter 4

# Results

### 4.1 Tuning

The tuning of the UKF  $\alpha$ ,  $\beta$ ,  $\kappa$ , and  $\lambda$  along with the HME initial learning rate and learning rate decay factor used in the simulation were derived using Monte Carlo methods and experimentation. The tuning factors selected gave the best results over the simulation experiments that were run. The best values were determined from a combination of analyzing the measurement residuals and their  $3\sigma$  innovation uncertainties, the state error plots along with their  $3\sigma$  uncertainties, and the ability of the HME gating weights or normalized MMAE weights to select the actual best fit to the truth data. A grading factor used were the true model state errors whose tuning values had the smallest errors within the  $3\sigma$  uncertainties or the most accurate estimates of the states. When analyzing the HME and MMAE weights converging to the closest match to truth, the tuning grading factor used was the earliest time of selecting the nominal nodes and the accuracy of selecting the closest to truth node. The tuning values which gave the best results were selected for the experiments.

Unless otherwise specified, the UKF tuning values listed in Table 4.1 were used in the experiments.

UKF Parameter	Value
$\alpha$	$\sqrt{3}$
$\beta$	2
$\kappa$	1
$\lambda$	1

Table 4.1 UKF Tuning Values

The HME tuning values are listed in Table 4.2.

HME Parameter	Value
$\eta_{i,0}$	5
$m_\eta$	0
$\lambda$	$\frac{1}{3600}$
$\gamma$	$\sqrt{L + \lambda}$

Table 4.2 HME Tuning Values

Dual layer HME used the threshold criteria for selecting a new nominal model parameter listed in Table 4.3.

HME Parameter	Value	Description
$g_{min}$	0.95	gating weight minimum threshold
num_meas	4	number of measurements gating weight needs to be above minimum threshold to be declared best fit / most probable

Table 4.3 Dual Layer HME Threshold Values

## 4.2 Experiment Setup

Unless otherwise noted, the experiments used a 1,000 kg RSO with constant density in near geosynchronous orbit with parameters listed in Table 4.4.

Orbital Element	Value	Description
a	42,164.7900 km	semi-major axis
e	0.000242	eccentricity
i	35.0000 <sup>o</sup>	inclination
$\Omega$	340 <sup>o</sup>	longitude of the ascending node
$\omega$	0 <sup>o</sup>	argument of periapsis
$M_0$	100 <sup>o</sup>	mean anomaly at epoch

Table 4.4 Orbital Elements of RSO

The observer is positioned atop Muana Kea, HI located at longitude 155.474659<sup>o</sup> W, latitude 19.825949<sup>o</sup> N, and altitude 4.160 km. The epoch is 15 Dec 2016 06:00:00 UTC. Integration time of the CCD camera is set to 10 s and measurements arrive at a frequency of once every 15 s and for a duration ranging from 1 to 4 hours. The initial attitude is set to [90.44, 5.63, 8.92] degrees and represent the Euler angles (ECI to RSO body). The attitude rate is [206, 103, 540] deg/hr, thus this RSO is modeled as tumbling.

The UKF initial state  $1\sigma$  error values are listed in Table 4.5:

State	Value	Description
$\theta$	$\frac{20^o}{3}$	Euler angle
$\omega$	24 $\frac{\text{deg}}{\text{hr}}$	Angular rate
$\vec{r}$	100 km	Position
$\dot{\vec{r}}$	1 $\frac{\text{km}}{\text{s}}$	Velocity

Table 4.5 UKF Initial State  $1\sigma$  Errors

The azimuth and elevation  $1\sigma$  measurement noise values are 0.1 seconds of arc. The UKFs did not use any process noise. Thus, the truth dynamics model is the same as the UKF dynamics model.

The shape of the RSO is a rectangular cuboid.

The simulation begins by reading in an experimental input file. It then seeds the randomization object. Next, it performs draws from a normal distribution with the mean

state values listed above and state error  $1\sigma$  values listed in Table 4.5 and adds this error to the initial state. All of the UKFs in the filter banks are initialized with the same state, UKF tuning values, and initial state error covariances. Then, the models are initialized with values depending on the experiment and are different in some aspect between each other.

Next a loop begins to generate the measurements. The true model is propagated at each measurement time step and the measurement is collected. A random draw from a normal distribution with mean zero and the  $1\sigma$  measurement noise value is generated. This noise is added to the true measurement to create the measurement with noise vectors. After all of the measurements with noise have been collected, the measurement processing loop begins.

In the measurement processing loop, each UKF in the filter bank(s) will have the exact same measurement feeding it. After the UKFs run, the measurement residual and measurement innovation covariance are sent to the MMAE and either single layer ME or dual-layer HME adaptive estimation technique. The adaptive estimation techniques then output either the likelihood probability or a-priori probability (gating weight) of each model. The MMAE then computes the normalized weight after each measurement processing loop iteration, the HME gating weight is already normalized. The loop then runs to completion after all of the measurements have been processed.

The output of both the UKFs and adaptive estimation techniques are then sent to analysis graders and plotters. The graders will determine which model the MMAE and either single layer ME or dual-layer HME believes is the most probable model. The MMAE uses the MMAE normalized weight of each model and the HME uses the a-priori probability or gating weight of each model. The model with highest probability is considered the ‘winner’. The grader compares this ‘winning’ probability to a probability threshold which

is set to 0.95, unless otherwise specified. This value was determined experimentally. If the model is the true model as selected in the experimental input file and the weights are above the probability threshold, then the experiment is considered a success as the experiment converged to selecting the true model. The plotting functions plot the outputs over time.

### 4.3 Experiment 1 - Size Detection

The purpose of this experiment is to detect the size of a RSO. This experiment assumes we already know the reflectivity properties of the surface materials and the true size model is contained within the filter bank. This test contains a MMAE and single layer ME each using the same nodes of 10 UKFs. The purpose of this experiment is to compare the results of the MMAE and single layer ME in selecting the proper size model. The size models are listed in Table 4.6. The units are in meters. The true model is in the bank and is represented by an asterisk. Thus, the true model is node #5 or [2.25, 1.00, 0.50].

Node#	Size
1	[1.00, 1.00, 1.00]
2	[1.00, 1.50, 2.00]
3	[1.00, 2.00, 3.00]
4	[0.50, 2.50, 1.00]
5*	[2.25, 1.00, 0.50]
6	[1.00, 1.50, 0.75]
7	[2.00, 2.00, 2.00]
8	[0.25, 2.50, 3.00]
9	[3.00, 0.50, 0.25]
10	[0.25, 4.00, 6.00]

Table 4.6 Experiment 1 - Size Options

This experiment used the following reflectivity properties:

- specular reflectance = 0.70

- diffuse reflectance = 0.50
- fraction of total light reflected = 0.583
- $n_u = 1000$
- $n_v = 1000$

Recall that  $n_u$  and  $n_v$  are two Phong like exponents that control the shape of the specular lobe in the horizontal and vertical directions respectively.

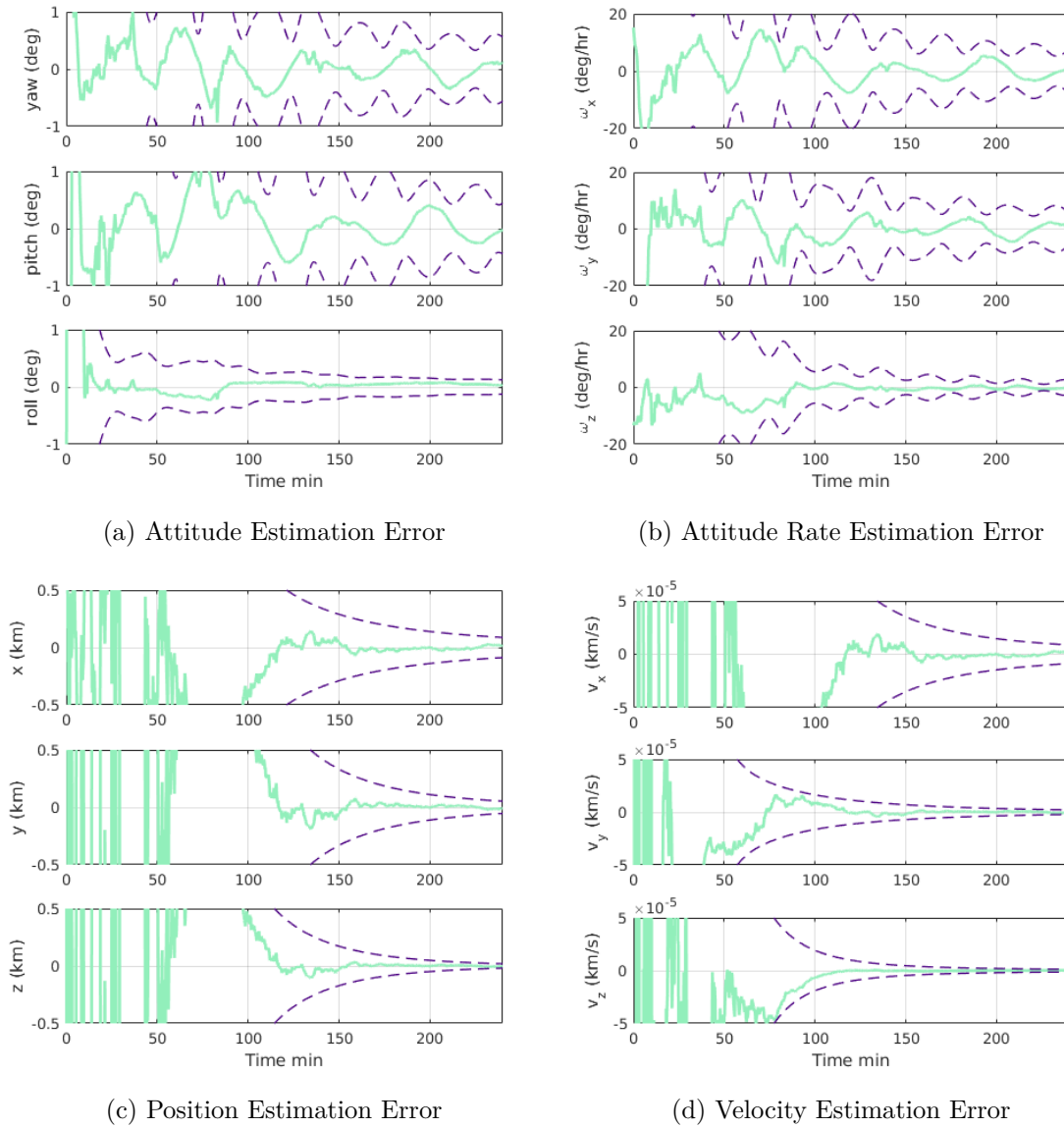
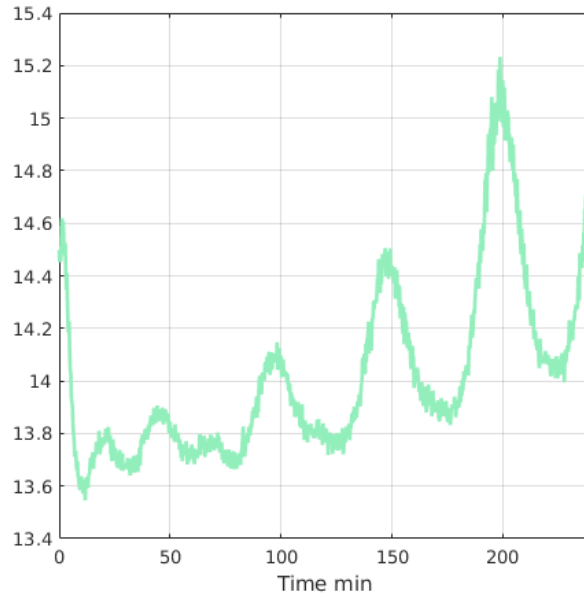


Figure 4.1 Experiment 1 - RSO State Estimation Errors

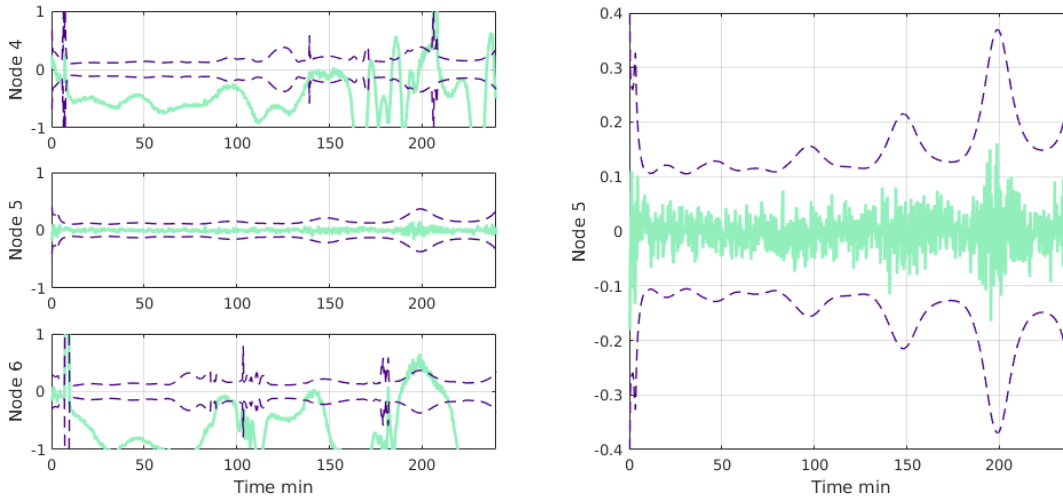
The state estimation errors of the true model along with their  $3\sigma$  uncertainty bounds calculated from the state error covariances for the true size model is shown in Figure 4.1. The attitude is estimated to within  $0.5^\circ$  of the truth in the  $x$ -axis and  $y$ -axis and  $0.1^\circ$  in

the  $z$ -axis. The attitude rate is estimated to be within  $5^{deg/hr}$  for the  $x$ -axis and  $y$ -axis and within  $1^{deg/hr}$  for the  $z$ -axis. The position is estimated to within 0.03 km in the  $x$ -axis and  $y$ -axis and to within 0.006 km in the  $z$ -axis. The velocity is estimated to within  $4 \times 10^{-6}$  km/s in the  $x$ -axis and  $y$ -axis and within  $3 \times 10^{-7}$  km/s in the  $z$ -axis.

The bolometric apparent magnitude measurements and measurement residuals along with the  $3\sigma$  uncertainties from the innovation covariance for the true model, node 5, and two non-true models, nodes 4 & 6, are shown in Figure 4.2. Next to that is a zoomed in version of the true model, node 5, bolometric apparent magnitude residuals and its  $3\sigma$  uncertainty.



(a) Bolometric Apparent Magnitude Measurements



(b) Models 4,5,&6 Apparent Magnitude Residuals (c) True Model Apparent Magnitude Residuals

Figure 4.2 Experiment 1 - Apparent Magnitude Measurement & Residuals

The MMAE weights and single layer ME gating weights are shown in Figure 4.3. The figures in the bottom row show the MMAE and ME weights zoomed in on the  $x$ -axis to

show the progression of the weights and when they both selected the true nodes as most probable.

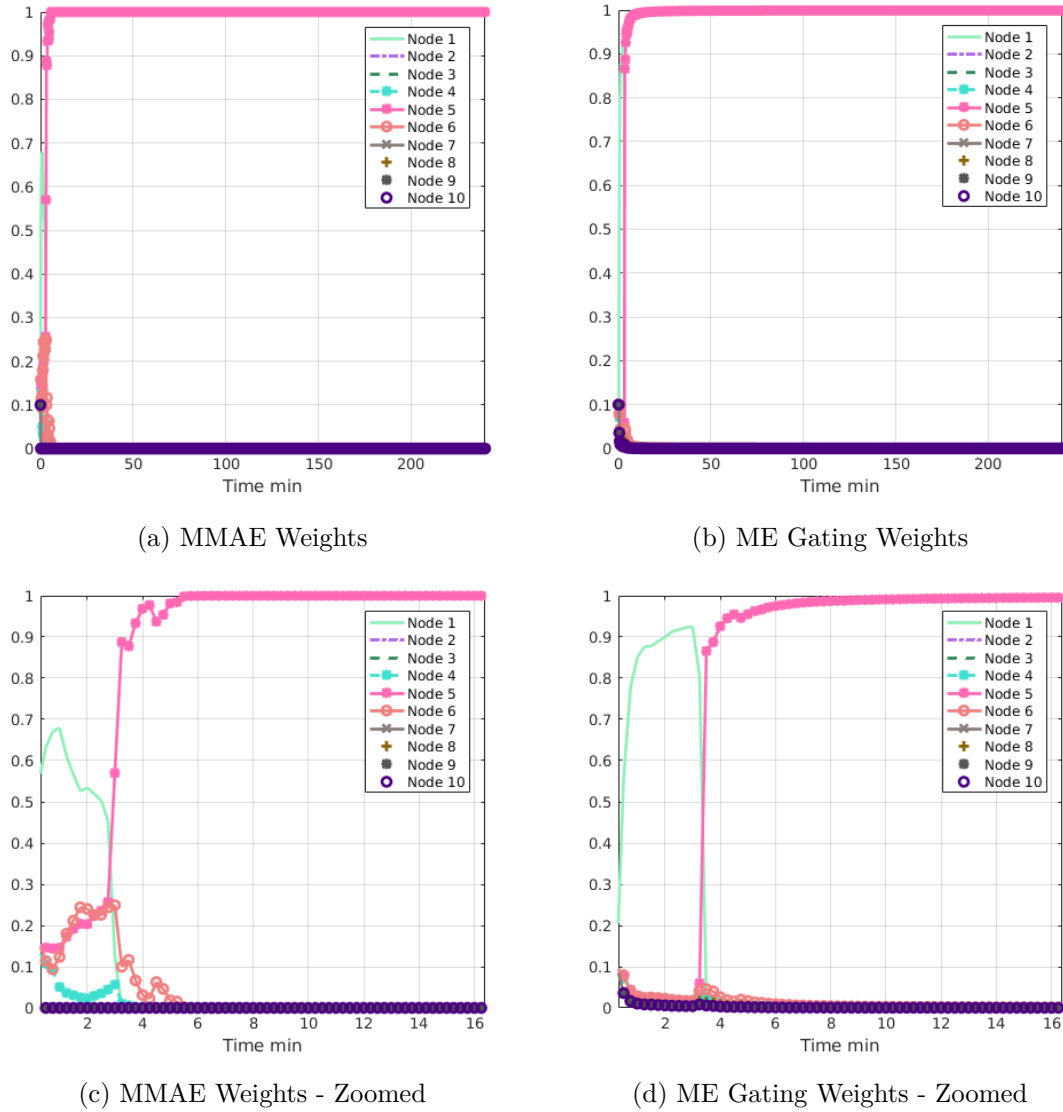


Figure 4.3 Experiment 1 - MMAE & ME Weights

It is clear that when the true size is in the filter bank both the MMAE and single layer ME converge to the true value after just 18 measurements.

## 4.4 Experiment 2 - Surface Material Reflectance Detection

The purpose of this experiment is to detect the surface material reflectance of a RSO. This experiment assumes we already know the size and shape and the true surface material reflectance model is contained within the filter bank. This test contains a MMAE and single layer ME each using the same nodes of 10 UKFs. The purpose of this experiment is to compare the results of the MMAE and single layer ME in selecting the proper surface material reflectance model. The surface material reflectance models are listed in Table 4.7.

Node#	Surface Material Reflectance
1	[0.25, 0.15, 0.625, 100, 100]
2	[0.29, 0.25, 0.525, 500, 500]
3	[0.35, 0.35, 0.425, 1000, 1000]
4	[0.45, 0.55, 0.425, 1200, 1200]
5*	[0.70, 0.50, 0.583, 1000, 1000]
6	[0.72, 0.53, 0.482, 800, 800]
7	[0.67, 0.43, 0.382, 500, 500]
8	[0.50, 0.50, 0.700, 500, 500]
9	[0.90, 0.80, 0.529, 800, 900]
10	[0.95, 0.78, 0.501, 1200, 1200]

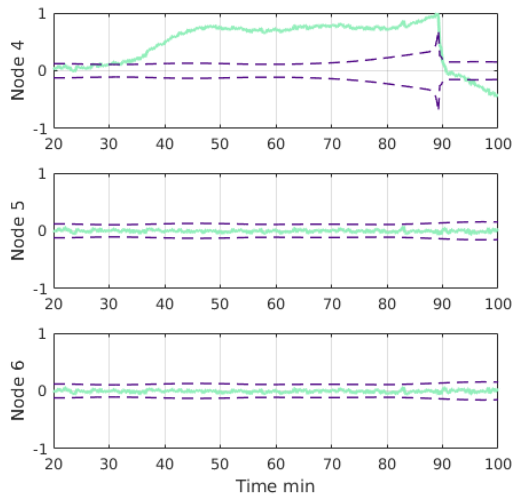
Table 4.7 Experiment 2 - Surface Material Reflectance Options

The columns are the specular reflectance, diffuse reflectance, fraction of total light reflected,  $n_u$ , and  $n_v$  respectively. The true model is in node #5 or [0.70, 0.50, 0.583, 1000, 1000]. This experiment used a RSO with size of [2.25, 1.00, 0.50].

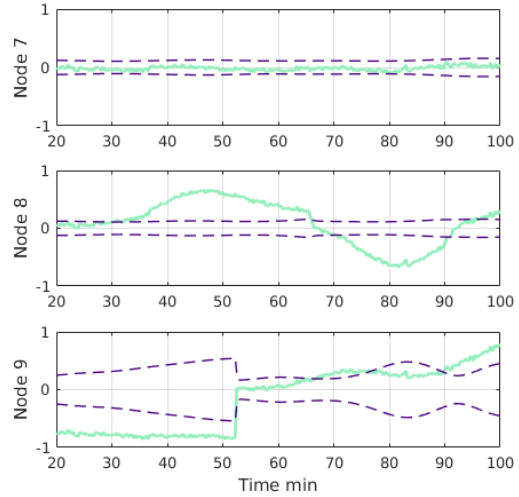
Experiment 2 ran the same scenario as experiment 1, just with different model parameters. Thus, the state errors and measurements are the same.

The bolometric apparent magnitude measurement residuals along with the  $3\sigma$  uncertainties from the innovation covariance for nodes 4-9, are shown in Figure 4.4. Below them are the zoomed in versions which demonstrate how similar nodes 5, 6, and 7 are to each

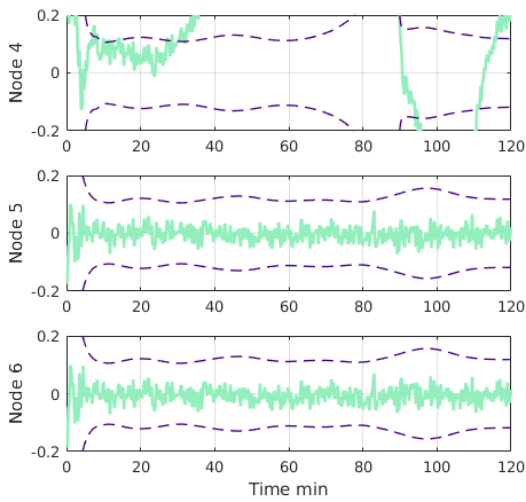
other.



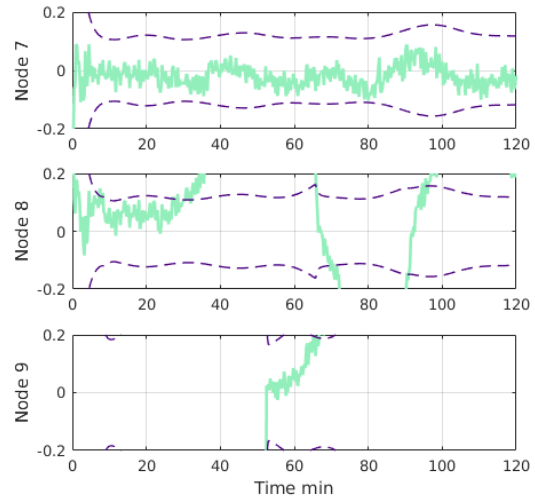
(a) Models 4,5,&6 AM Residuals



(b) Models 7,8,&9 AM Residuals



(c) Models 4,5,&6 AM Residuals - Zoomed

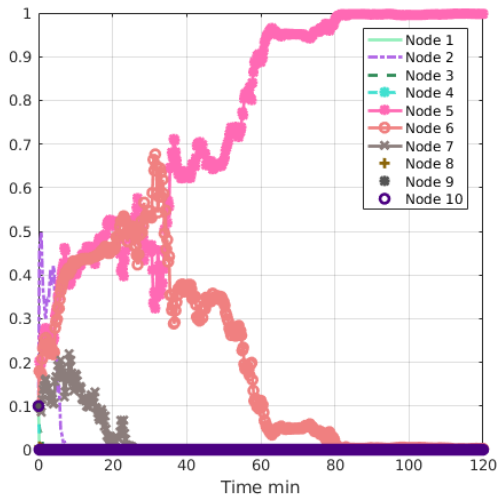


(d) Models 7,8,&9 AM Residuals - Zoomed

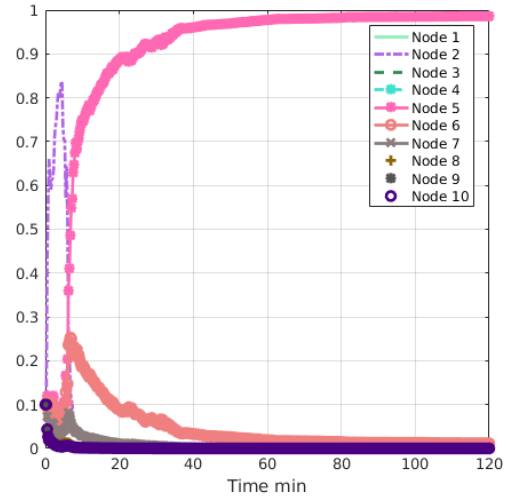
Figure 4.4 Experiment 2 - Apparent Magnitude Residuals

The MMAE weights and single layer ME gating weights are shown in Figure 4.5. The figures in the bottom row show the MMAE and ME weights zoomed in on the  $x$ -axis to

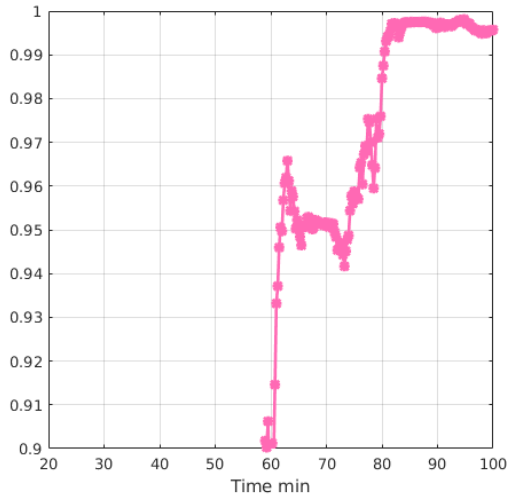
show the progression of the weights and when they both selected the true nodes as most probable.



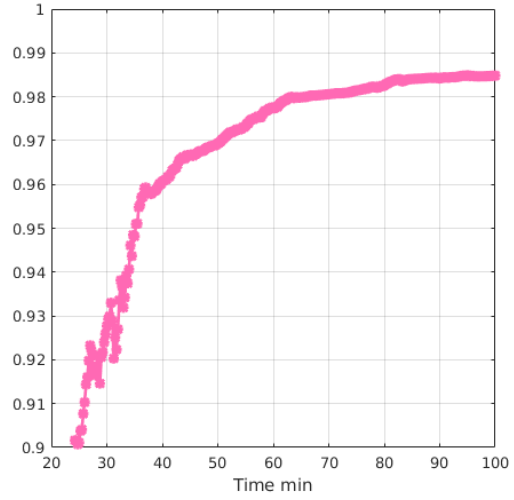
(a) MMAE Weights



(b) ME Gating Weights



(c) MMAE Weights - Zoomed



(d) ME Gating Weights - Zoomed

Figure 4.5 Experiment 2 - MMAE & ME Weights

Due to the similarities in nodes 5, 6, and 7, it takes a bit more time for the MMAE

and HME to converge to the true value. However, both do end up converging to the true value. The single layer ME selects Node 5 as the most probable with a gating weight above 0.95 at 35.25 minutes. While the MMAE selects Node 5 as the most probable with a MMAE weight above 0.95 at 62.65 minutes, then it dips below 0.95 to rise again for the final time at 74.25 minutes. The ME appeared to be more confident in the true value than the MMAE from about 35.25 - 74.25 minutes.

### 4.5 Experiment 3 - Dual Layer HME - Size & Reflectance

This experiment is to test a dual-layer HME with combined Size and Reflectance models from the previous two experiments. The primary purpose of this experiment is to demonstrate the dual-layer HME and compare it to a permutation of states for the MMAE. There are a total of 21 HME nodes and 101 MMAE nodes. The HME macromodes are listed in Tables 4.8 to 4.10.

Node#	Reflectance	Size
1	[0.25, 0.15, 0.625, 100, 100]	[0.25, 0.50, 0.25]

Table 4.8 Experiment 3 MM1 - Nominal Model

Node#	Surface Material Reflectance	Size
1	[0.25, 0.15, 0.625, 100, 100]	[0.25, 0.50, 0.25]
2	[0.29, 0.25, 0.525, 500, 500]	[0.25, 0.50, 0.25]
3	[0.35, 0.35, 0.425, 1000, 1000]	[0.25, 0.50, 0.25]
4	[0.45, 0.55, 0.425, 1200, 1200]	[0.25, 0.50, 0.25]
5*	[0.70, 0.50, 0.583, 1000, 1000]	[0.25, 0.50, 0.25]
6	[0.72, 0.53, 0.482, 800, 800]	[0.25, 0.50, 0.25]
7	[0.67, 0.43, 0.382, 500, 500]	[0.25, 0.50, 0.25]
8	[0.50, 0.50, 0.700, 500, 500]	[0.25, 0.50, 0.25]
9	[0.90, 0.80, 0.529, 800, 900]	[0.25, 0.50, 0.25]
10	[0.95, 0.78, 0.501, 1200, 1200]	[0.25, 0.50, 0.25]

Table 4.9 Experiment 3 MM2 - Reflectance Model

Node#	Surface Material Reflectance	Size
1	[0.25, 0.15, 0.625, 100, 100]	[1.00, 1.00, 1.00]
2	[0.25, 0.15, 0.625, 100, 100]	[1.00, 1.50, 2.00]
3	[0.25, 0.15, 0.625, 100, 100]	[1.00, 2.00, 3.00]
4	[0.25, 0.15, 0.625, 100, 100]	[0.50, 2.50, 1.00]
5*	[0.25, 0.15, 0.625, 100, 100]	[2.25, 1.00, 0.50]
6	[0.25, 0.15, 0.625, 100, 100]	[1.00, 1.50, 0.75]
7	[0.25, 0.15, 0.625, 100, 100]	[2.00, 2.00, 2.00]
8	[0.25, 0.15, 0.625, 100, 100]	[0.25, 2.50, 3.00]
9	[0.25, 0.15, 0.625, 100, 100]	[3.00, 0.50, 0.25]
10	[0.25, 0.15, 0.625, 100, 100]	[0.25, 4.00, 6.00]

Table 4.10 Experiment 3 MM3 - Size Model

Recall from Section 3.5.4 that MM2 is initialized with the nominal size and MM3 is initialized with the nominal reflectance. As the simulation runs and different size and reflectance nodes cross the high probability threshold, then all of the nominal states are updated.

The true model is again in node 5 of both MM2 and MM3.

The dual-layer HME selected index results are shown in Figure 4.6. This shows that the HME initially selects the incorrect node in MM3 at 1.75 minutes. It quickly recovers and selects the true node in MM3 at 7.25 minutes. Then, the HME selects the true node in MM2 at 25.75 minutes. At this point, the HME has converged to the true size and surface material reflectance models. It stays with this selection for the rest of the run. The HME TLGN and individual macromode GN gating weights are shown in Figures 4.7 and 4.8 respectively along with zoomed in versions to show the weights prior to convergence. Note that the TLGN weights go into an approximate balanced state after convergence. In other words, after convergence at 25.75 minutes, the TLGN gating weights are approximately equivalent. This is because macromodes 1, 2, and 3 all contain the true model due to the convergence of the nominal state.

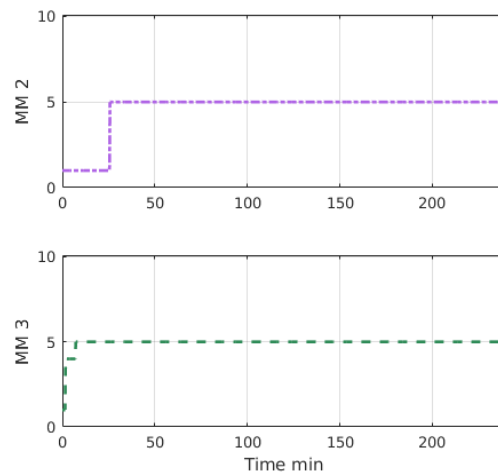
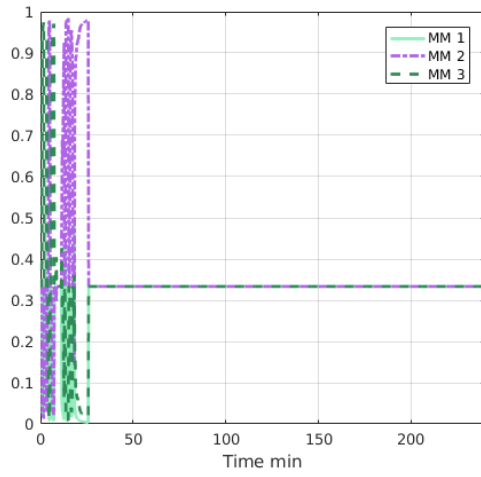
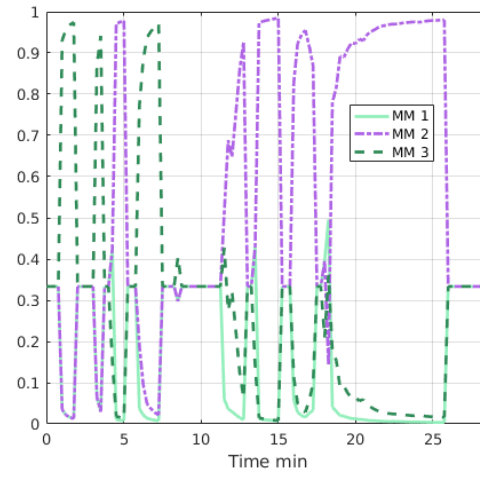


Figure 4.6 Experiment 3 - Selected Nominal Indices

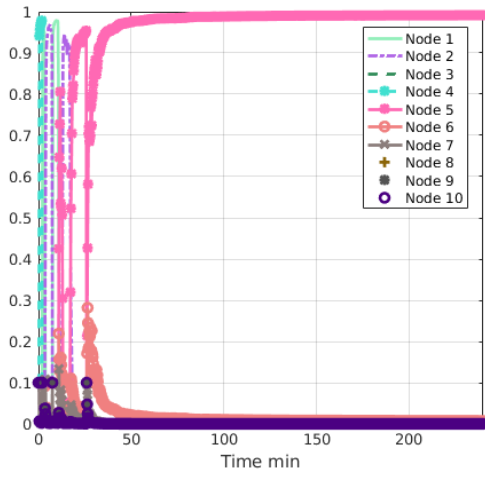


(a) HME TLGN Gating Weights

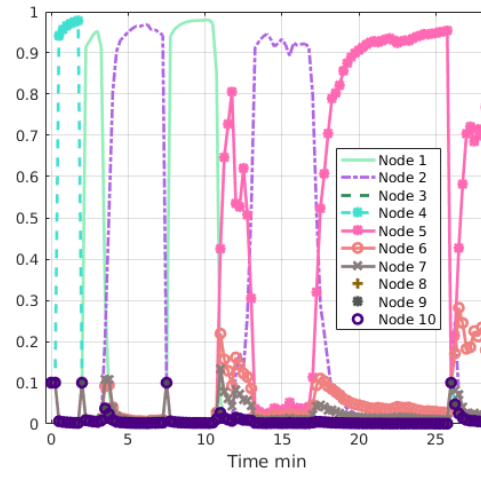


(b) HME TLGN Gating Weights - Zoomed

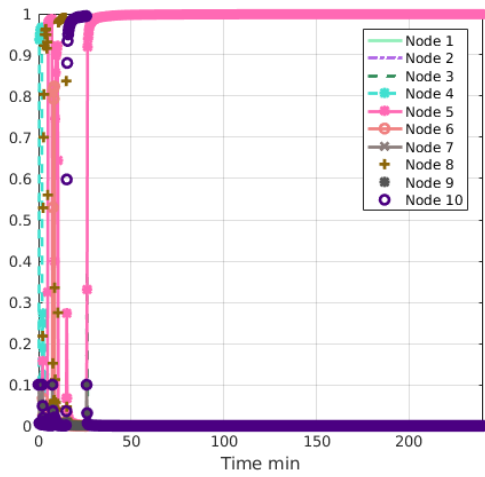
Figure 4.7 Experiment 3 - HME TLGN Gating Weights



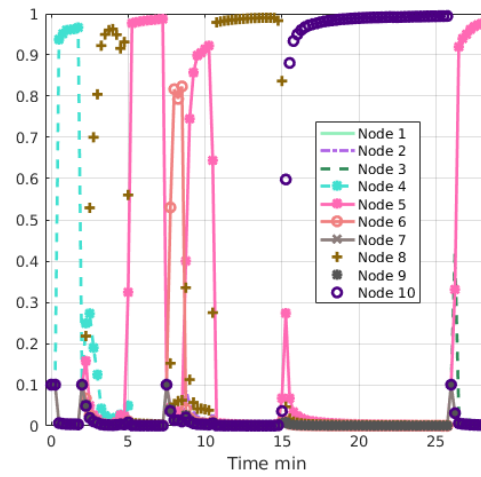
(a) HME MM2 Gating Weights



(b) HME MM2 Gating Weights - Zoomed



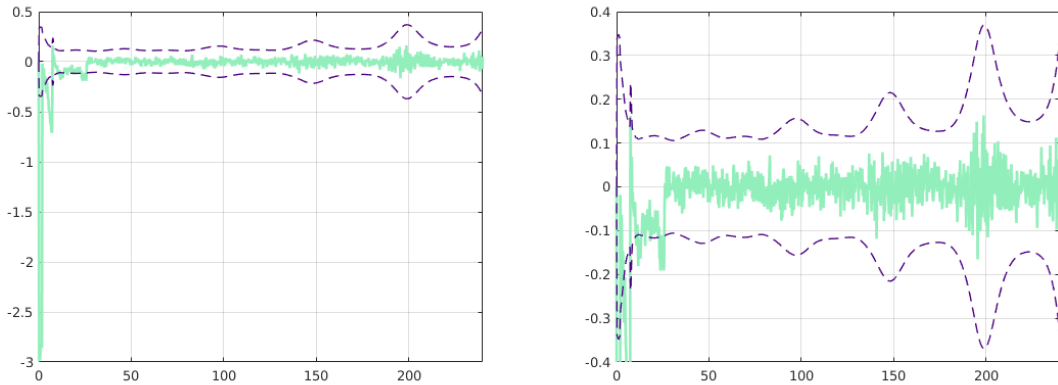
(c) HME MM3 Gating Weights



(d) HME MM3 Gating Weights - Zoomed

Figure 4.8 Experiment 3 - HME Macromode Gating Weights

The nominal model residuals start off with wild errors as seen in Figure 4.9. However, after the HME converges to the true models and updates the nominal model with the true models, then the residuals become well behaved again.

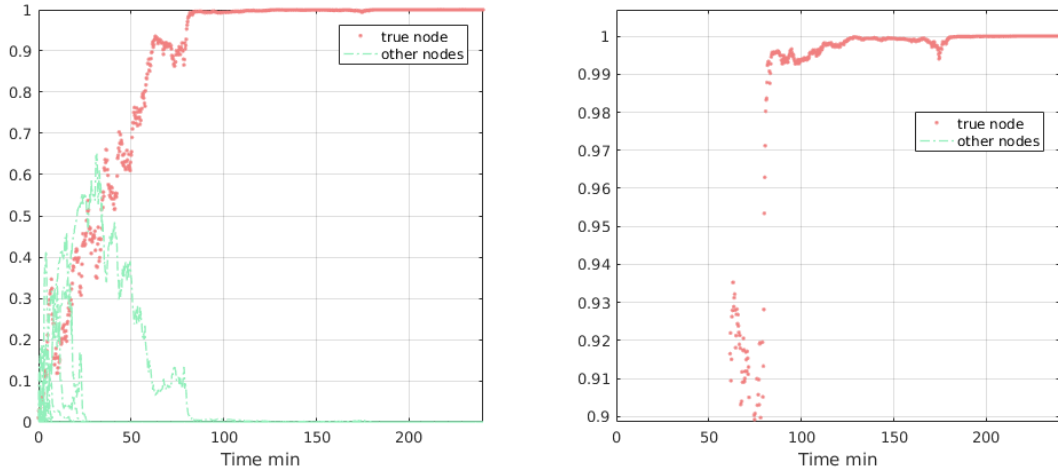


(a) Nominal Model AM Residuals

(b) Nominal Model AM Residuals - Zoomed

Figure 4.9 Experiment 3 - Nominal Model Am Residuals

The MMAE weights are shown in Figure 4.10. The MMAE selects the true node above the threshold criteria initially at 80 minutes and maintains it through the rest of the simulation.



(a) MMAE Weights

(b) MMAE Weights - Zoomed

Figure 4.10 Experiment 3 - MMAE Weights

The state errors of the nominal node along with their  $3\sigma$  values are listed in Figures 4.11

and 4.12. This shows that the states start off with large errors due to the poor initial estimate. However, as the models converge to the true models, the states settle down to become well behaved.

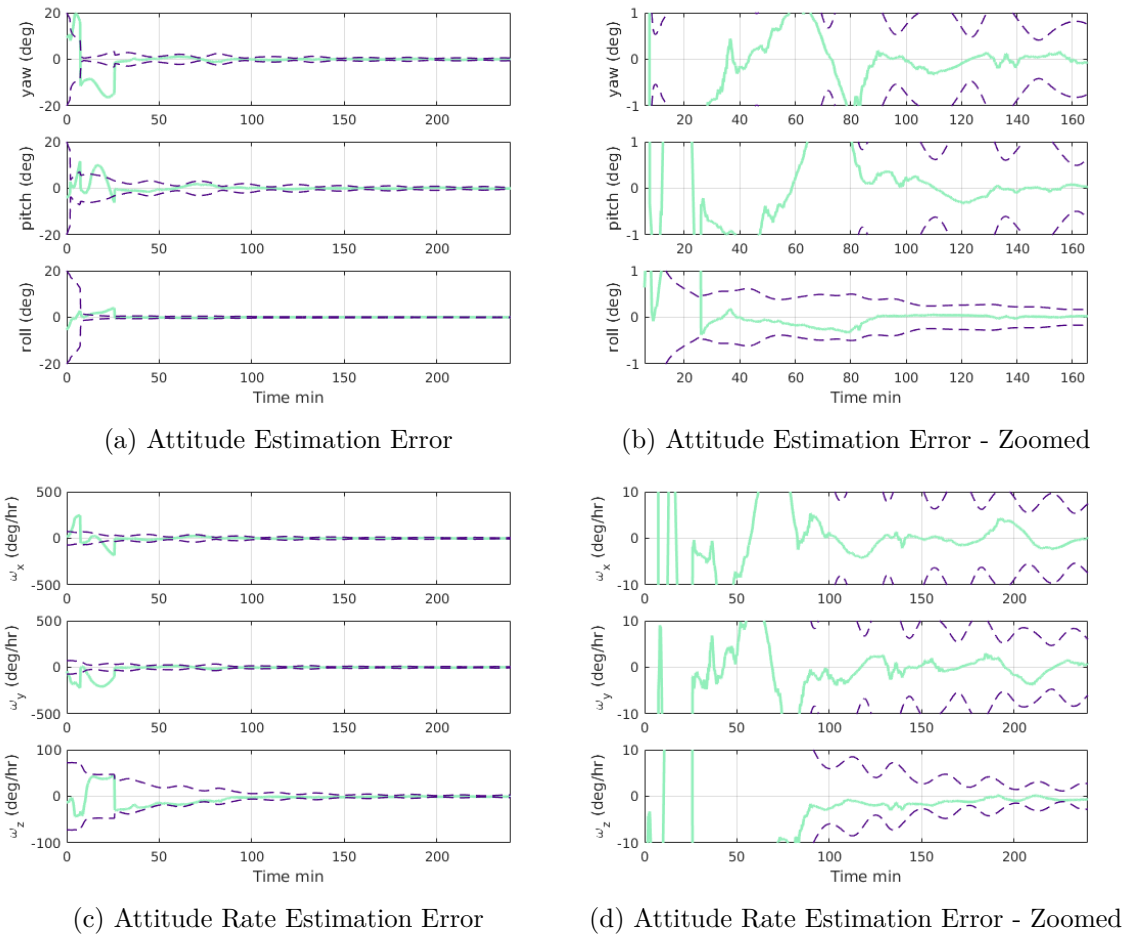


Figure 4.11 Experiment 3 - RSO Attitude & Rate State Estimation Errors

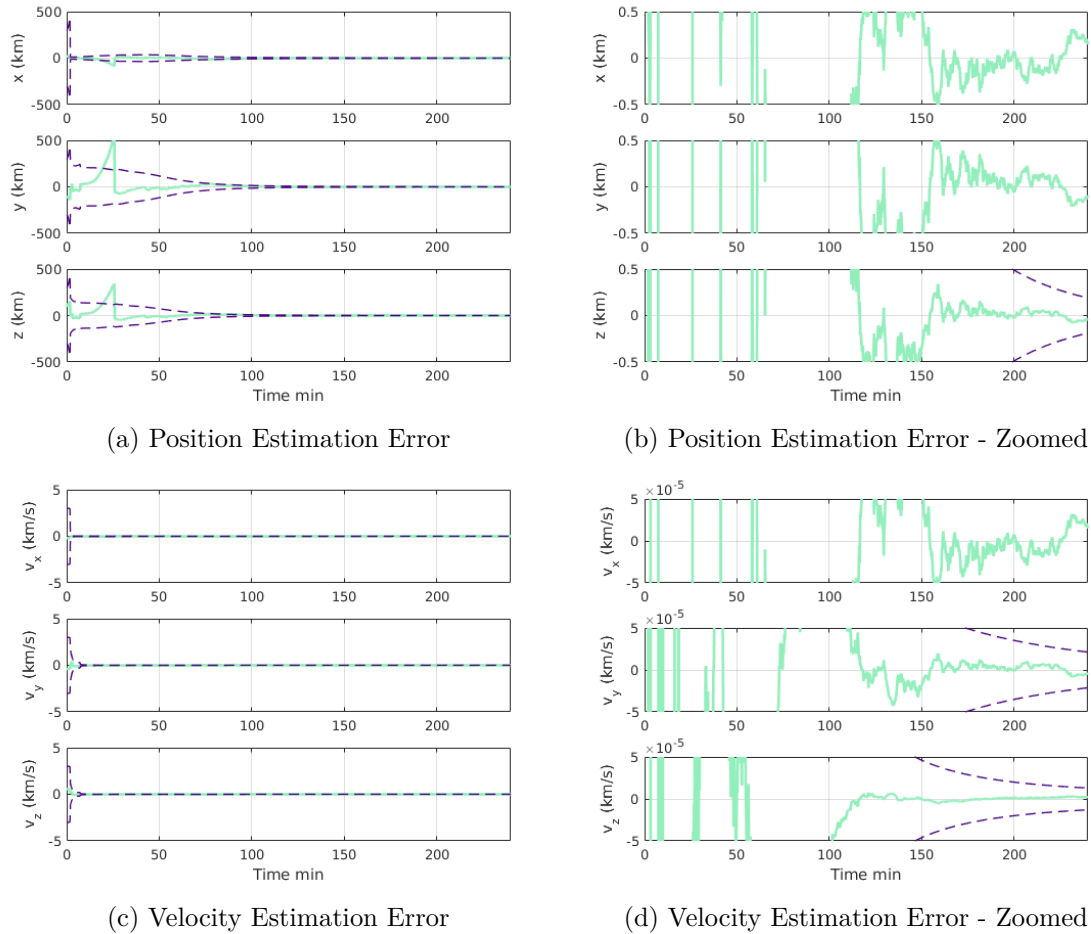


Figure 4.12 Experiment 3 - RSO Position &amp; Velocity State Estimation Errors

The HME converged to the true value at 25.75 minutes, while the MMAE converged to the true value at 80 minutes. The HME started with a fairly large initial learning rate of 5 and macromodes 2 and 3 only contained 10 nodes each, while the MMAE contained 101 nodes. In addition, the MMAE is multiplicatively recursive and thus removing past knowledge can be a bit slower than the HME with a large learning rate. This is probably why the HME converged quicker than the MMAE in this case. The HME took approximately 7 minutes and 33.2112 seconds to run, while the MMAE took approximately 36 minutes and

4.1514 seconds to run. The code for this experiment was run in MATLAB<sup>®</sup> 2016a with the test computer specifications listed in Table 4.11. In this experiment and with the software and computer tested, the HME converged approximately 68% faster than the MMAE. In addition, the HME ran about 80% quicker on the CPU due to running with about 80% fewer nodes than the MMAE needed.

Item	Value
CPU	Intel <sup>®</sup> vPro 8-core i7-4860HQ @ 2.40 GHz
RAM	DDR3L MHz SDRAM 32 GB
Video	NVIDIA <sup>®</sup> GeForce <sup>®</sup> GTX980M with 4GB GDDR5
HDD	256 GB SDD
OS	Linux Kernel - 4.4.0-45 - Ubuntu 16.04 SMP x86-64

Table 4.11 Simulation Computer Specs

## 4.6 Experiment 4 - Size Detection - Truth not in bank

These experiments are similar to that run in Section 4.3. This time a random selection of sizes were drawn from a uniform distribution between  $[0.01, 10]$  meters and the true value does not exist in the bank. The first experiment in this section will be setup so that the true value is very close to and in between two node members. The second experiment will have the true value also drawn from the uniform distribution, but still contained within the filter bank. The third and final experiment will show what happens if the size falls outside of the parameter space covered by the filter bank. These experiments chose a starting HME learning parameter of 0.75 to make it less aggressive in selecting the most probable model.

### 4.6.1 4a - Truth Bounded By Filter Bank - Close

The size values in the filter bank are shown in Table 4.12. The experiment was specifically chosen so that the true value is very close in size and in between the sizes of nodes 1 and 2. The true value's dimensions are [2.148221, 1.528310, 0.383102] meters. The root mean square (RMS) of each sides surface area between the true value and node 1 is 0.0462 meters. The RMS between the true value and node 2 is 0.0787 meters. Thus, the node that is closest to the truth is represented by an asterisk and is node #1.

Node#	Size	Node#	Size
1*	[2.1931, 1.5312, 0.3876]	14	[5.2123, 0.0703, 2.5080]
2	[2.1000, 1.5000, 0.3800]	15	[1.1487, 3.8697, 3.6752]
3	[8.0184, 8.3192, 8.6192]	16	[3.6698, 6.7922, 1.3556]
4	[6.0741, 6.2948, 0.3629]	17	[6.9358, 3.1119, 7.1133]
5	[8.4041, 2.5050, 1.6847]	18	[0.6574, 3.7356, 0.1331]
6	[2.4956, 2.4221, 9.5364]	19	[8.1981, 1.6046, 8.3224]
7	[1.1457, 6.5768, 6.6392]	20	[4.8095, 6.6329, 5.3119]
8	[1.6560, 1.5997, 8.1073]	21	[8.5630, 9.9092, 4.0989]
9	[8.0394, 1.9792, 7.4537]	22	[1.4670, 1.0379, 3.0749]
10	[5.9828, 6.1228, 0.5668]	23	[9.5632, 3.1219, 8.0460]
11	[8.0750, 1.8253, 5.1686]	24	[1.0359, 2.1514, 0.8791]
12	[3.0411, 8.8775, 8.1177]	25	[3.7595, 6.5309, 3.3503]
13	[1.8489, 1.5086, 6.4279]		

Table 4.12 Experiment 4a - Size Options

The MMAE weights and HME gating weights are shown in Figure 4.13. They both converge to node 1 as the most probable size as this size was the closest to the true value. Once again, the HME converged to the 95% threshold quicker than the MMAE, 13 minutes for the HME and 26 minutes, 29.5 minutes, and finally 32 minutes before the MMAE finally settled in on node 1 as the most probable model. The HME converged quicker than the MMAE even with a lower initial learning parameter than used in other experiments.

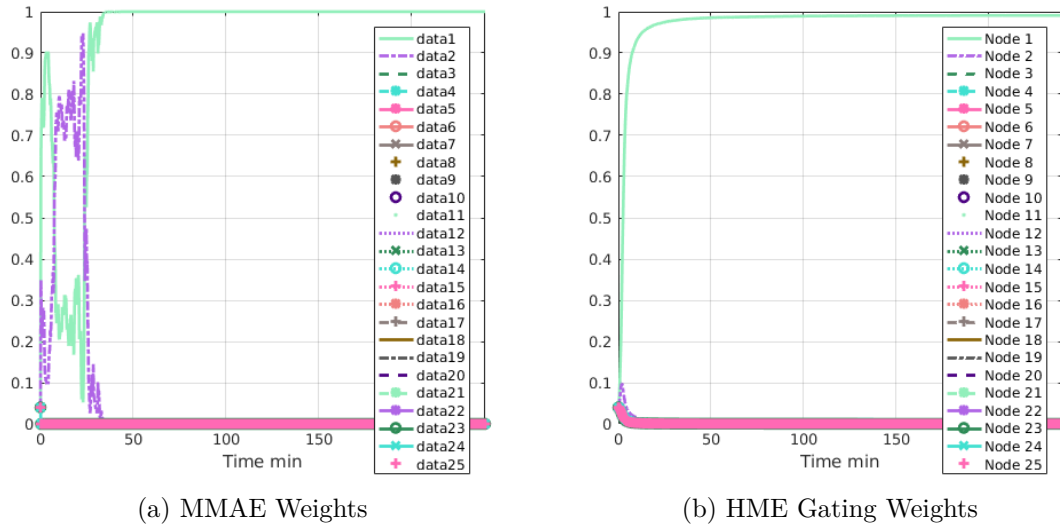


Figure 4.13 Experiment 4a - MMAE &amp; HME Gating Weights

The bolometric apparent magnitude measurement residuals along with the  $3\sigma$  uncertainty from the innovation covariances for nodes 1, 2, and 3 are shown in Figure 4.14. Notice how nodes 1 and 2 both have well behaved residuals while node 3 is completely off.

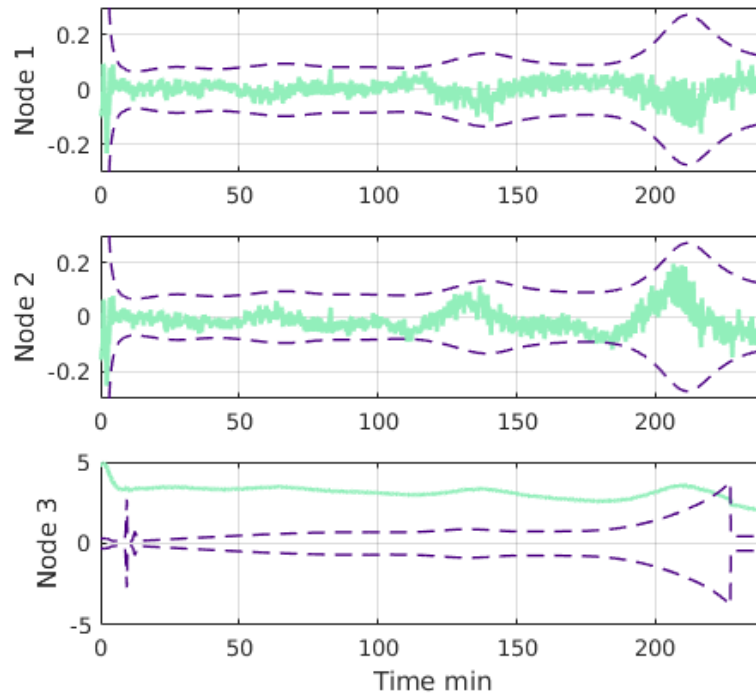


Figure 4.14 Experiment 4a - Measurement Residuals

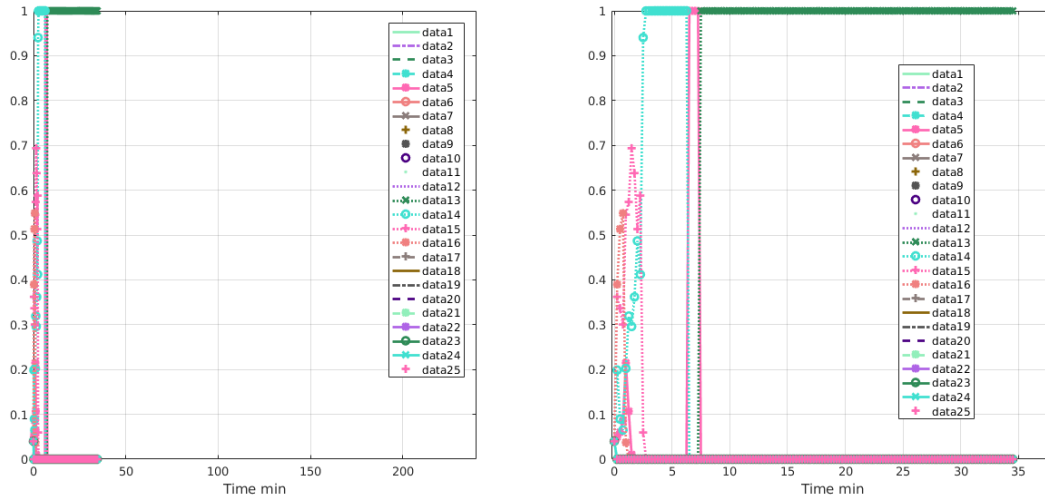
#### 4.6.2 4b - Truth Bounded By Bank - Further Away

The size values in the filter bank are shown in Table 4.13. This experiment was chosen such that the nodes properly spanned the expected true value parameter space. The values were chosen at random with 25 items again. The true value was also selected at random in this experiment and is  $[5.3682, 0.6662, 1.74952]$  meters. Using the RMS of the surface area of each size of the RSO as compared to the true value, then the closest item in the filter bank is node 14 with a RMS value of 2.8769 meters. Notice that this RMS value is much larger than that in the previous experiment. However, as shown in Table 4.14 that while node 14 has the overall lowest RMS score, there is no one node that best matches the true size.

Node#	Size	Node#	Size
1	[1.5332, 0.5759, 1.0271]	14*	[5.2123, 0.0703, 2.5080]
2	[7.8923, 4.9225, 5.2796]	15	[1.1487, 3.8697, 3.6752]
3	[8.0184, 8.3192, 8.6192]	16	[3.6698, 6.7922, 1.3556]
4	[6.0741, 6.2948, 0.3629]	17	[6.9358, 3.1119, 7.1133]
5	[8.4041, 2.5050, 1.6847]	18	[0.6574, 3.7356, 0.1331]
6	[2.4956, 2.4221, 9.5364]	19	[8.1981, 1.6046, 8.3224]
7	[1.1457, 6.5768, 6.6392]	20	[4.8095, 6.6329, 5.3119]
8	[1.6560, 1.5997, 8.1073]	21	[8.5630, 9.9092, 4.0989]
9	[8.0394, 1.9792, 7.4537]	22	[1.4670, 1.0379, 3.0749]
10	[5.9828, 6.1228, 0.5668]	23	[9.5632, 3.1219, 8.0460]
11	[8.0750, 1.8253, 5.1686]	24	[1.0359, 2.1514, 0.8791]
12	[3.0411, 8.8775, 8.1177]	25	[3.7595, 6.5309, 3.3503]
13	[1.8489, 1.5086, 6.4279]		

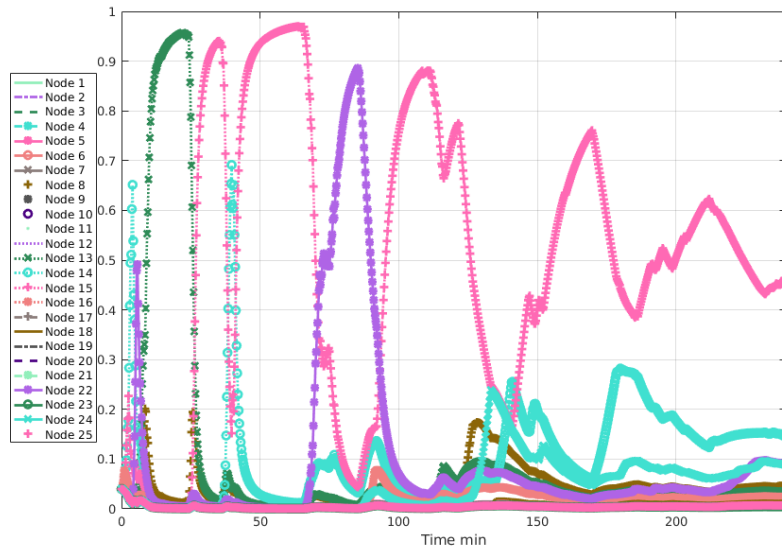
Table 4.13 Experiment 4b - Size Options

The MMAE weights and HME gating weights are shown in Figure 4.15. The MMAE fails due to underflow with the last non-NAN (not-a-number) value reported at 34.5 minutes. The HME is able to continue on until the end of the experiment. The experiment ends with node 15 clearly being the most likely. The final gating weight value is a relatively small value of 46.24%.



(a) MMAE Weights

(b) MMAE Weights - Zoomed



(c) HME Gating Weights

Figure 4.15 Experiment 4b - MMAE & HME Gating Weights

The UKFs are designed to shutdown if the values of the state error covariance are no longer valid. For instance, if the matrix is no longer positive definite, then this signifies

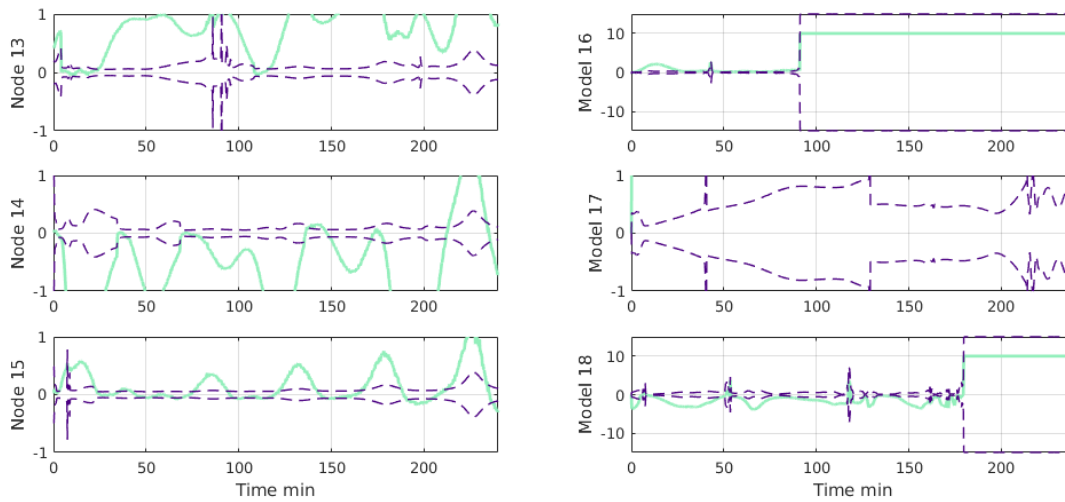
a significant numerical issue with the filter and it is shutdown for the rest of the run. In addition, if any NAN values are detected, then the filter is again shutdown. This signifies numerical difficulties of numbers getting too large (numerical overflow) or too small (numerical underflow). This typically occurs due to the filter diverging.

The bolometric apparent magnitude (BAM) measurement residuals along with the  $3\sigma$  uncertainty from the innovation covariances for nodes 13-18 are shown in Figure 4.16. It is clear from these plots that node 15 while not that well behaved is clearly better behaved than the other nodes shown here. In addition, notice how nodes 16, 18, the innovation covariance and residuals get very large and constant eventually. This is due to their UKFs being shutdown due to numerical difficulties.

Table 4.14 shows the areas of the 6 sided rectangular cuboid along with the RMS of the areas with the true value of 8 nodes with the lowest overall RMS values along with the true node (index 0) values. It is clear that node 15 matches much better than node 14 for sides 1 and 3. Node 13 is better than node 14 for sides 2 and 4. However, nodes 15 and 13 are not a very close match for sides 5 and 6. The best match is very much dependent on the orientation of the RSO and changes based on the orientation along with the position of the sun and observer. This can be seen from the results of this experiment.

Node#	Area						RMS
0*	3.5763	9.3917	3.5763	9.3917	1.1655	1.1655	0
14	0.3664	13.0724	0.3664	13.0724	0.1763	0.1763	2.8769
22	1.5226	4.5109	1.5226	4.5109	3.1914	3.1914	3.2733
1	0.8830	1.5747	0.8830	1.5747	0.5915	0.5915	4.7850
24	2.2286	0.9107	2.2286	0.9107	1.8913	1.8913	4.9756
13	2.7893	11.8845	2.7893	11.8845	9.6971	9.6971	5.1518
18	2.4558	0.0875	2.4558	0.0875	0.4972	0.4972	5.4243
8	2.6491	13.4257	2.6491	13.4257	12.9692	12.9692	7.2218
15	4.4451	4.2217	4.4451	4.2217	14.2219	14.2219	8.1231

Table 4.14 Experiment 4b - Select Areas &amp; RMS values



(a) BAM Measurement Residuals 13, 14, 15

(b) BAM Measurement Residuals 16,17,18

Figure 4.16 Experiment 4b - BAM Measurement Residuals

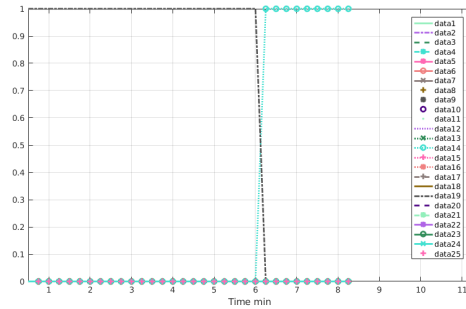
This experiment showed that both adaptive estimators struggled when the true value, while being contained within the values of the filter bank, was not very close to any value in the filter bank. This also showed how the HME was able to get through some of the difficulties and still give results that represented the closest match in the filter bank to the

true value, even if its confidence in that value was not very high as demonstrated from a lower gating weight value, while the MMAE suffered an underflow condition and stopped reporting results.

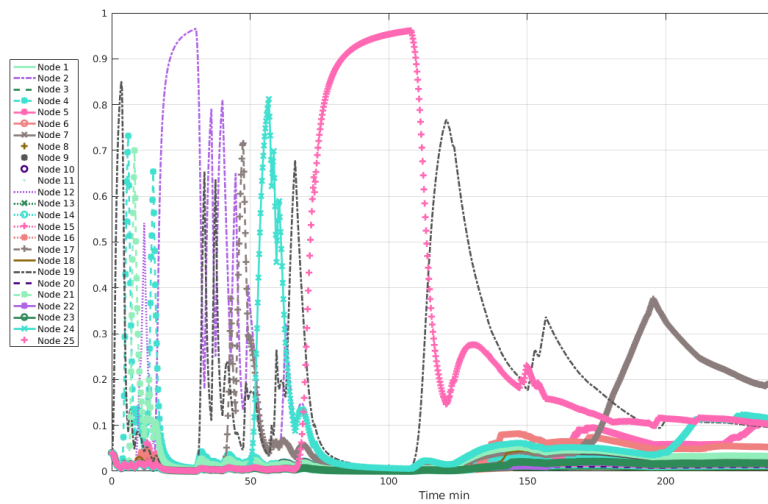
### 4.6.3 4c - Truth Outside Bank

This experiment looks at the case where the filter bank does not properly span the parameter space of an expected RSO size. The filter bank is the same as that used in the previous section and is listed in Table 4.13. The true value was selected at random from a uniform distribution in the range of  $[10.01, 12]$ . Thus this case looks at a true value just outside of the range spanned by the filter bank. The true value used is  $[10.8283, 11.1636, 11.9493]$ . The closest node in the filter bank is node 21 with a RMS of the surface area of each size of the RSO as compared to the true value of 43.3529 meters.

Based on extrapolation of the previous experiment, one would expect the adaptive estimators to struggle to converge to a solution for the size. This was indeed the case as can be seen from the MMAE and HME weights in Figure 4.17. The MMAE weights again failed due to underflow and the HME again went through a series of different selections before not having high confidence in any node by the end of the experiment.



(a) MMAE Weights - Zoomed



(b) HME Gating Weights

Figure 4.17 Experiment 4c - MMAE & HME Gating Weights

## 4.7 Experiment 5 - Dual Layer HME - Maneuver Time Detection

The purpose of this experiment is to test the detection of the time and magnitude of an out of plane maneuver using a dual-layer HME and MMAE. There are a total of 19 HME nodes and 19 MMAE nodes. The HME macromodes are listed in Tables 4.15, 4.17 and 4.18

and  $??$ . Where time is in seconds and the velocity vectors are in  $km/s$ . The velocity vector components are in the local RSO centered velocity, orbital normal, bi-normal (VNB) frame, which is similar to the Frenet frame as shown in Figure 4.18.  $V$  is along the instantaneous velocity vector,  $N$  is normal to the orbital plane, and  $B$  completes the right hand rule and typically points away from the Earth for an orthogonal coordinate system centered on the RSO and relative to the Earth. The true maneuver vector is  $[6000, 0, 0.00004835, 0]$ . Thus the true maneuver occurred at 100 minutes and contained a  $\Delta v$  in the normal direction of  $0.04835m/s$ . This value was selected as a possible value for a station-keeping maneuver of a RSO at GEO. Thus, the true value is not in the filter bank and the truth time and closest velocity to the true value is MM3 node 3 and again represented with an asterisk in the table.

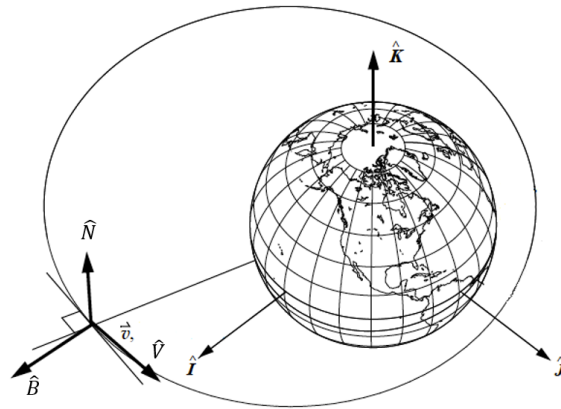


Figure 4.18 Local VNB Frame

[72]

This experiment changed a few input values. This experiment did not use a decaying learning rate. Instead, it used a constant learning rate of 5. This value was experimentally determined and it was also determined that a decaying learning rate decaying to a value below 5 cannot detect small  $\Delta v$  values with the selected noise values used in this experiment. In addition, the weight threshold of a macromode or node's gating weight to

represent the best fit to the true model was changed to 0.75. This value was determined experimentally.

Node#	time	$\Delta v_v$	$\Delta v_n$	$\Delta v_b$
1	0	0	0	0

Table 4.15 Experiment 5 MM1 - Nominal Model

Node#	time	$\Delta v_v$	$\Delta v_n$	$\Delta v_b$
1	3000	0	0	0
2	3000	0	0.0000078	0
3	3000	0	0.0000429	0
4	3000	0	0.0000780	0
5	3000	0	0.0004290	0
6	3000	0	0.0007800	0

Table 4.16 Experiment 5 MM2 - Maneuver at 50 minutes

Node#	time	$\Delta v_v$	$\Delta v_n$	$\Delta v_b$
1	6000	0	0	0
2	6000	0	0.0000078	0
3*	6000	0	0.0000429	0
4	6000	0	0.0000780	0
5	6000	0	0.0004290	0
6	6000	0	0.0007800	0

Table 4.17 Experiment 5 MM3 - Maneuver at 100 minutes

Node#	time	$\Delta v_v$	$\Delta v_n$	$\Delta v_b$
1	9000	0	0	0
2	9000	0	0.0000078	0
3	9000	0	0.0000429	0
4	9000	0	0.0000780	0
5	9000	0	0.0004290	0
6	9000	0	0.0007800	0

Table 4.18 Experiment 5 MM4 - Maneuver at 150 minutes

The  $\Delta v$  is only applied to the non-nominal model macromode at the time listed in the macromode node. Thus, prior to 50 minutes (or 3000 seconds), all of the nodes in all of the macromodes are equivalent. This is shown in the gating weights. Only after a  $\Delta v$  is applied to the nodes, do the weights start separating and a winner is selected for each macromode, and an overall macromode winner is selected.

At 117 minutes, the incorrect node is selected. The node with the highest probability of being the true node was node 4 with a  $\Delta v$  in the normal direction of  $0.0780^m/s$ . Eventually, at 174 minutes, the actual closest node to the truth is selected, node 3 with a  $\Delta v$  in the normal direction of  $0.0429^m/s$ .

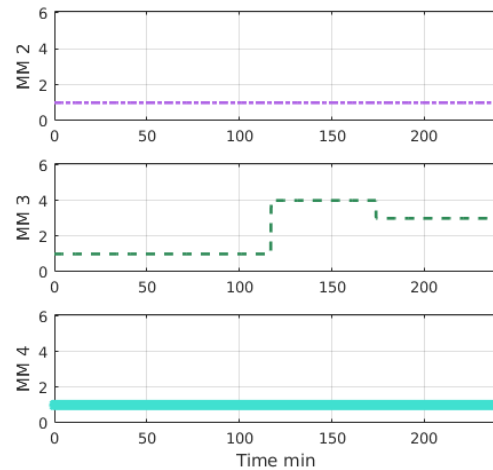
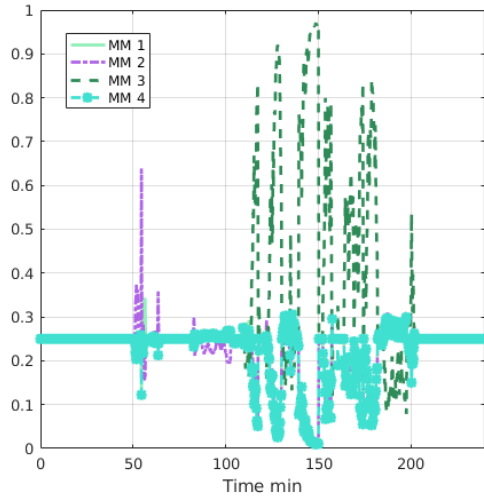
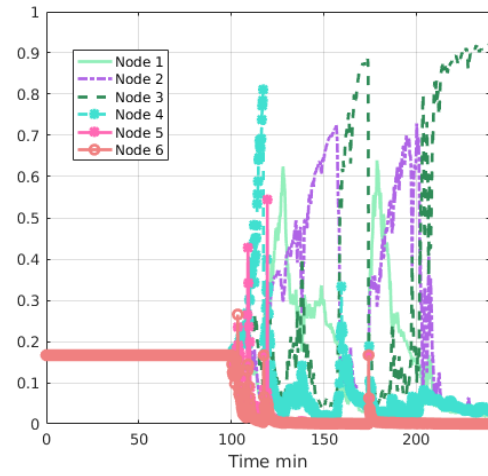


Figure 4.19 Experiment 5 - Selected MM Nodes

The dual-layer HME node selections are shown in Figure 4.19. This shows the macromode selection, and the node selection within macromode 3 ( $\Delta v$  occurring at 6000 seconds or 100 minutes).



(a) HME Gating Weights - Final



(b) HME Weights - MM3

Figure 4.20 Experiment 5 - HME Gating Weights

The MMAE normalized weights are shown in Figure 4.21. The MMAE behaved similarly to the HME in selecting other nodes prior to settling in on node 10. Node 10 is the same as MM3, node 3 in the HME and represents the actual closest  $\Delta v$  and time in the filter banks to the true value.

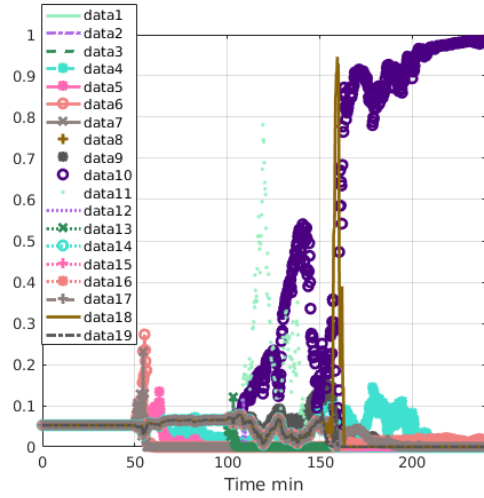


Figure 4.21 Experiment 5 - MMAE Weights

#### 4.7.1 Experiment 5b - In-Track Maneuver Detection

This experiment is to test the time and magnitude of an in plane maneuver using a dual-layer HME. There are a total of 19 HME nodes in 4 macromodes and 19 MMAE nodes. The HME macromodes are listed in Tables 4.19, 4.20 and 4.22 and ???. Time is in seconds and velocity vectors are in  $km/s$  and in the VNB coordinate system. The true maneuver is  $\Delta v_v = 0.0439m/s$  at 100 minutes. This value was selected to be approximately the same magnitude as that in experiment 5 but in the in-track direction instead of out of plane. The true maneuver node is HME MM3 node 3 and MMAE node 10.

Node#	time	$\Delta v_v$	$\Delta v_n$	$\Delta v_b$
1	0	0	0	0

Table 4.19 Experiment 5b MM1 - Nominal Model

Node#	time	$\Delta v_v$	$\Delta v_n$	$\Delta v_b$
1	3000	0	0	0
2	3000	0.00000780	0	0
3	3000	0.00004290	0	0
4	3000	0.00007800	0	0
5	3000	0.00042900	0	0
6	3000	0.00078000	0	0

Table 4.20 Experiment 5b MM2 - maneuver at 50 minutes

Node#	time	$\Delta v_v$	$\Delta v_n$	$\Delta v_b$
1	6000	0	0	0
2	6000	0.00000780	0	0
3*	6000	0.00004290	0	0
4	6000	0.00007800	0	0
5	6000	0.00042900	0	0
6	6000	0.00078000	0	0

Table 4.21 Experiment 5b MM3 - maneuver at 100 minutes

Node#	time	$\Delta v_v$	$\Delta v_n$	$\Delta v_b$
1	9000	0	0	0
2	9000	0.00000780	0	0
3	9000	0.00004290	0	0
4	9000	0.00007800	0	0
5	9000	0.00042900	0	0
6	9000	0.00078000	0	0

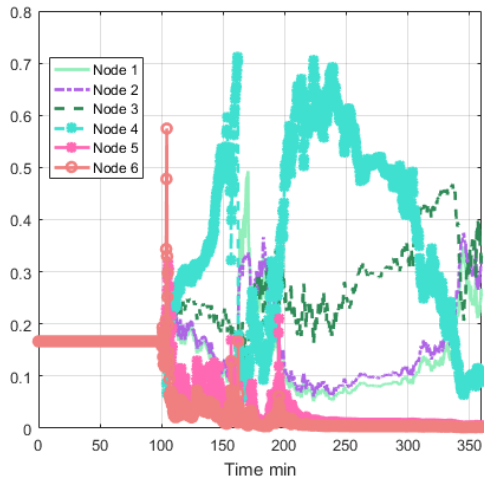
Table 4.22 Experiment 5b MM4 - maneuver at 150 minutes

The gating weight for macromode 3 and the TLGN along with the MMAE normalized weights are shown in Figure 4.22. The measurement residuals along with the  $3\sigma$  innovation covariances for MM3 nodes 3 & 4 (MMAE nodes 10 & 11 respectively) are shown in Figure 4.23. The position and velocity errors along with the  $3\sigma$  error estimate uncertainty for MM3 nodes 3& 4 are shown in Figures 4.24 and 4.25. The apparent magnitude

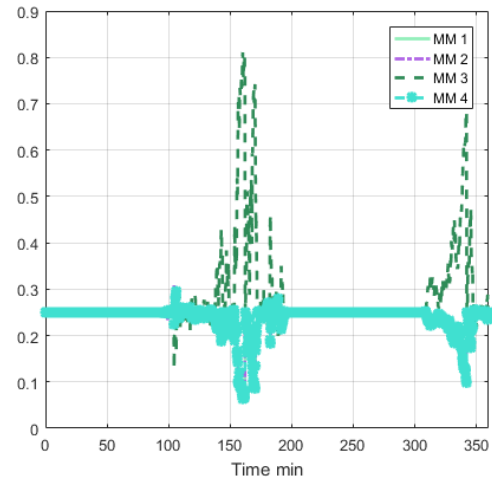
measurement input plots are shown in Figure 4.26.

It is quite clear that in this scenario, the simulation has a more difficult time detecting the in-track maneuver with the same magnitude as the out of plane maneuver. Both the HME and MMAE select HME MM3 Node 4, which is MMAE node 11, first before slowly starting to select the correct node, HME MM3 Node 3, which is MMAE node 10.

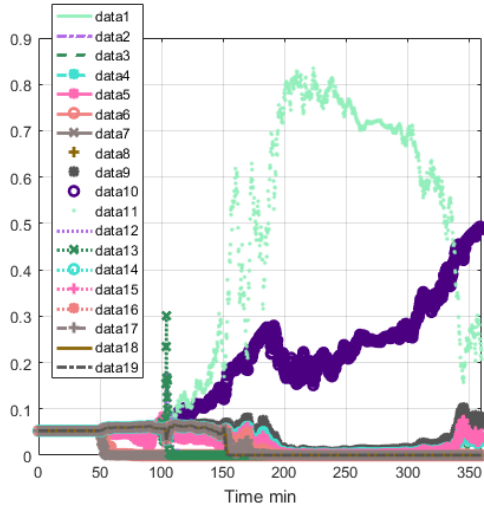
This result is a bit counter-intuitive as it should be easier to detect an in-track maneuver than an of plane maneuver. The most likely explanation is the result of the specific geometry of this simulation and how much velocity change is detectable via the maneuver from the observer's vantage point. Future work should be done to verify this result.



(a) HME Gating Weights - MM3

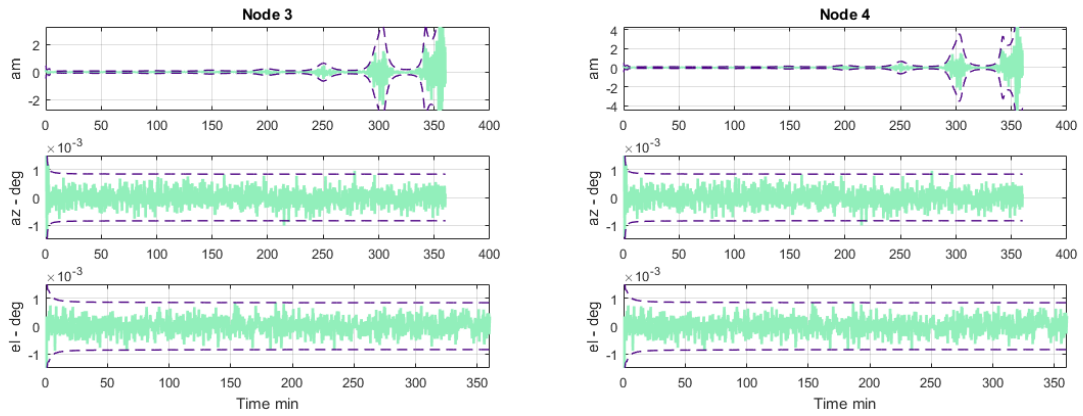


(b) HME Gating Weights - TLGN



(c) MMAE Weights

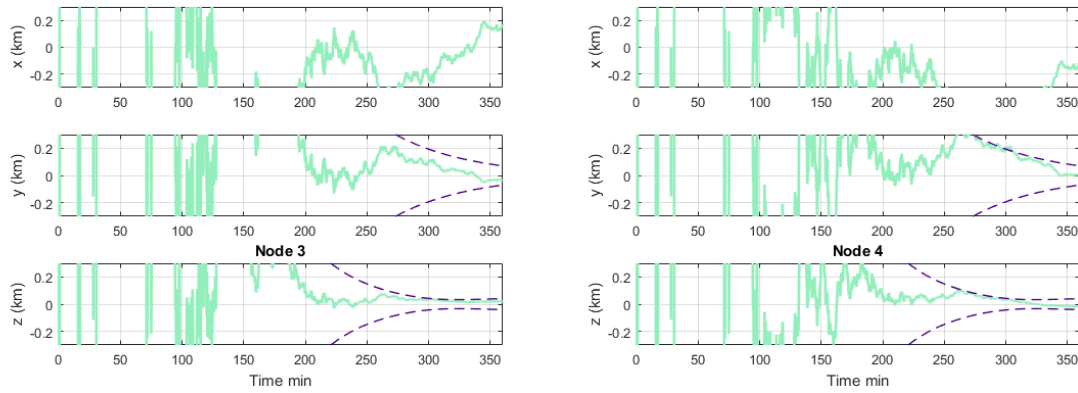
Figure 4.22 Experiment 5b - Weights



(a) MM3 - Node 3 Measurement Residuals

(b) MM3 - Node 4 Measurement Residuals

Figure 4.23 Experiment 5b - Measurement Residuals



(a) MM3 - Node 3 Position Error

(b) MM3 - Node 4 Position Error

Figure 4.24 Experiment 5b - Position Errors

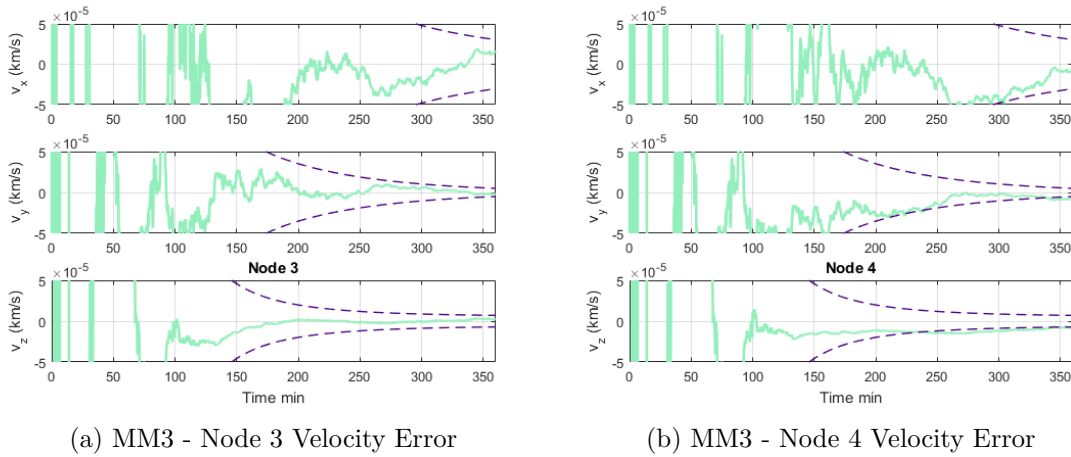


Figure 4.25 Experiment 5b - Velocity Errors

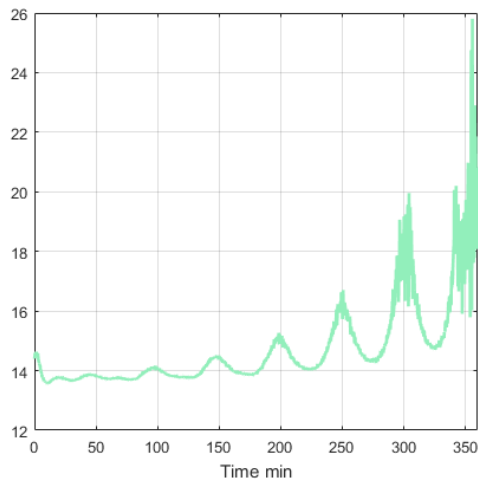


Figure 4.26 RSO Apparent Magnitude Measurement

## 4.8 Experiment 6 - Dual Layer HME - Compare Methods

This experiment is a reformulation of the experiment in Section 4.5. The purpose of this experiment is to compare the enhanced HME as described in this thesis with the

**Initialization:**

$$a_0 = \frac{1}{L} \quad (4.1)$$

**HME Loop:**

$$u_l = a_{l,k-1}, \quad (4.2)$$

$$g_l = \frac{e^{u_l}}{\sum_{j=1}^L e^{u_j}} \quad (4.3)$$

$$f(z_k | \alpha_l) = \frac{1}{\sqrt{|2\pi S_{l,k}|}} \exp \left\{ \frac{-\tilde{y}_{l,k}^T S_{l,k}^{-1} \tilde{y}_{l,k}}{2} \right\} \quad (4.4)$$

$$h_l = \frac{g_l \sum_{j=1}^M f(z_k | \alpha_{l,j}) g_{l,j}}{\sum_{n=1}^L g_n \sum_{j=1}^M f(z_k | \alpha_{n,j}) g_{n,j}} \quad (4.5)$$

$$a_{l,k} = a_{l,k-1} + \eta (h_l - g_l) \quad (4.6)$$

Table 4.23 Original HME TLGN algorithm

HME described by Crain [13] and Jordan and Jacobs [50, 73], which will be referred to in this section as the original HME. The differences between the original and enhanced HME are that the original HME uses a constant learning parameter, a slightly different TLGN a-posteriori probability algorithm, and not using any threshold in selecting the most probable node. The TLGN algorithm for the original HME is shown in Table 4.23. The TLGN algorithm for the enhanced HME is shown in Table 3.11.

The differences between the two simulation runs were the HME algorithm, enhanced HME learning rate parameters, and winner selection thresholds. The enhanced HME used

the learning rate parameters listed in Table 4.24.

item	value
HME Base Learning Rate	3
HME Decay Factor	$1/900$
HME Learning Rate Minimum Value	0.75
HME Weight Threshold	0.93

Table 4.24 Experiment 6 - Enhanced HME Learning Rate Parameters

The original HME used a constant learning parameter of 3 and no weight threshold. Both experiments used the same simulation time of 1 hr.

This experiment did not include the truth model parameters in the nodes. There are a total of 21 HME nodes. The HME macromodes are listed in Tables 4.25 to 4.27. The nominal portion of the macromode for MM2 and MM3 were not shown for brevity here. Recall that MM2 will contain the nominal size and MM3 will contain the nominal reflectance. The truth reflectance is  $[0.7098312, 0.512837, 0.5923, 993.832, 993.832]$ , while the truth size is  $[2.148221, 1.528310, 0.383102]$ . Thus, the node closest to the true reflectance macromode is node 5 in MM2, while the node closest to the true size macromode is node 1 in MM3.

Node#	Reflectance	Size
1	$[0.55, 0.475, 0.541, 850, 850]$	$[3.5, 3.0, 2.4]$

Table 4.25 Experiment 6 MM1 - Nominal Model

Node#	Surface Material Reflectance
1	[0.50, 0.25, 0.312, 1000, 1000]
2	[0.55, 0.35, 0.358, 1000, 1000]
3	[0.60, 0.43, 0.391, 1000, 1000]
4	[0.65, 0.45, 0.425, 1000, 1000]
5*	[0.70, 0.50, 0.583, 1000, 1000]
6	[0.74, 0.83, 0.595, 950, 950]
7	[0.76, 0.85, 0.596, 850, 950]
8	[0.78, 0.88, 0.598, 900, 900]
9	[0.95, 0.95, 0.621, 1000, 1000]

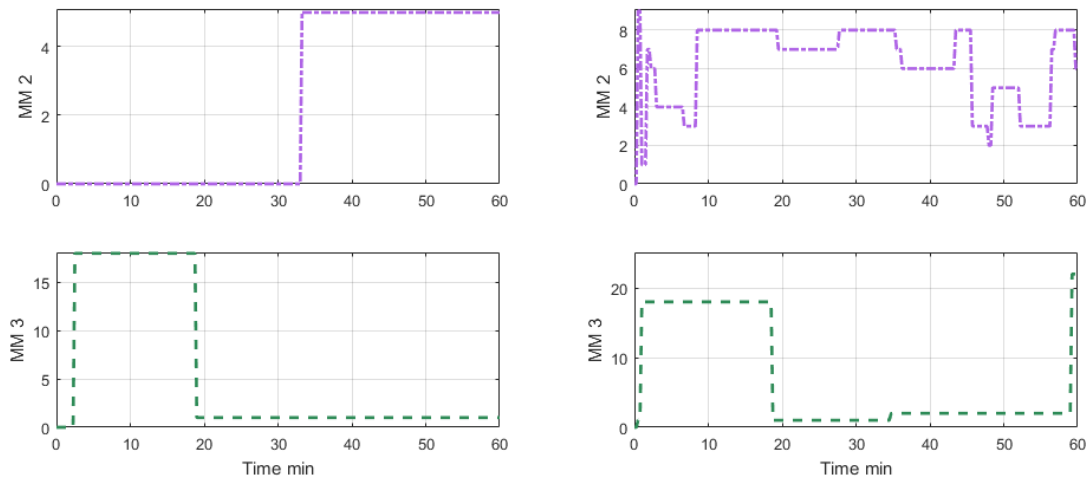
Table 4.26 Experiment 6 MM2 - Reflectance Model

Node#	Size	Node#	Size
1*	[2.1931, 1.5312, 0.3876]	14	[5.2123, 0.0703, 2.5080]
2	[2.5283, 1.9839, 0.9783]	15	[1.1487, 3.8697, 3.6752]
3	[8.0184, 8.3192, 8.6192]	16	[3.6698, 6.7922, 1.3556]
4	[6.0741, 6.2948, 0.3629]	17	[6.9358, 3.1119, 7.1133]
5	[8.4041, 2.5050, 1.6847]	18	[0.6574, 3.7356, 0.1331]
6	[2.4956, 2.4221, 9.5364]	19	[8.1981, 1.6046, 8.3224]
7	[1.1457, 6.5768, 6.6392]	20	[4.8095, 6.6329, 5.3119]
8	[1.6560, 1.5997, 8.1073]	21	[8.5630, 9.9092, 4.0989]
9	[8.0394, 1.9792, 7.4537]	22	[1.4670, 1.0379, 3.0749]
10	[5.9828, 6.1228, 0.5668]	23	[9.5632, 3.1219, 8.0460]
11	[8.0750, 1.8253, 5.1686]	24	[1.0359, 2.1514, 0.8791]
12	[3.0411, 8.8775, 8.1177]	25	[3.7595, 6.5309, 3.3503]
13	[1.8489, 1.5086, 6.4279]		

Table 4.27 Experiment 6 MM3 - Size Model

The selected macromode nodes are shown in Figure 4.27. The gating weights for the macromodes are shown in Figure 4.28 while the gating weights for the TLGN for both the enhanced and original HME are shown in Figure 4.29. The enhanced HME selected index 5 for MM2 at about 32 minutes and selected index 18 first at about 2 minutes and index

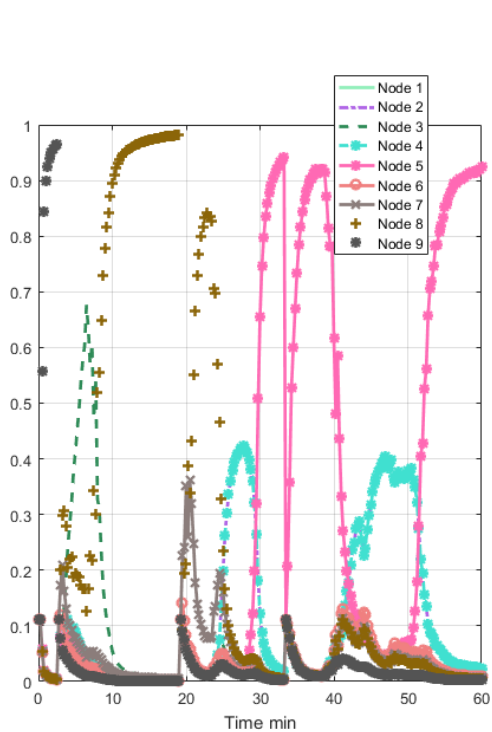
1 at about 19 minutes for MM1. The original HME selected every index at MM2 at least once and never settled in on one, while briefly selecting index 1 before switching to index 2 and finally index 22 near the end of the simulation.



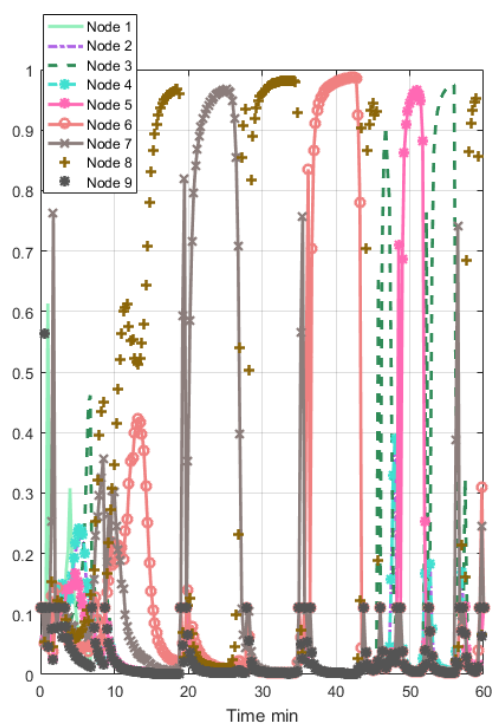
(a) Enhanced HME Selected Nominal Index

(b) Original HME Selected Nominal Index

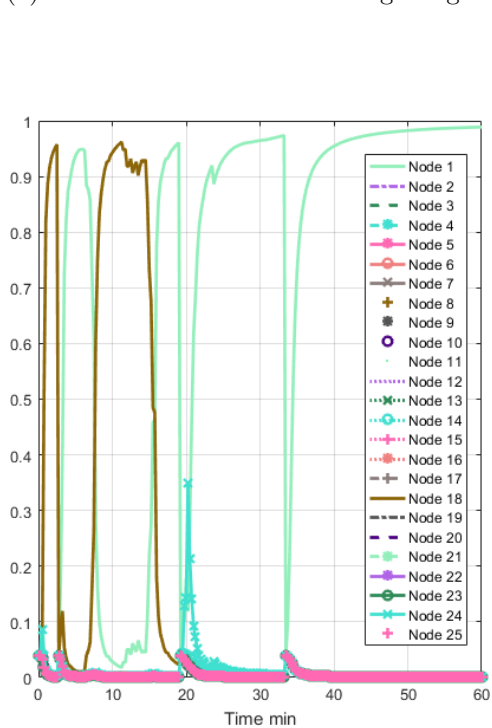
Figure 4.27 Experiment 6 - HME Selected Nominal Index



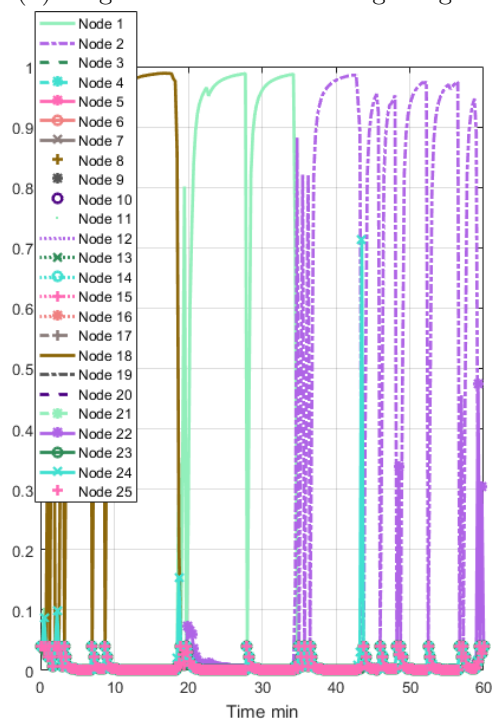
(a) Enhanced HME MM2 Gating Weights



(b) Original HME MM2 Gating Weights

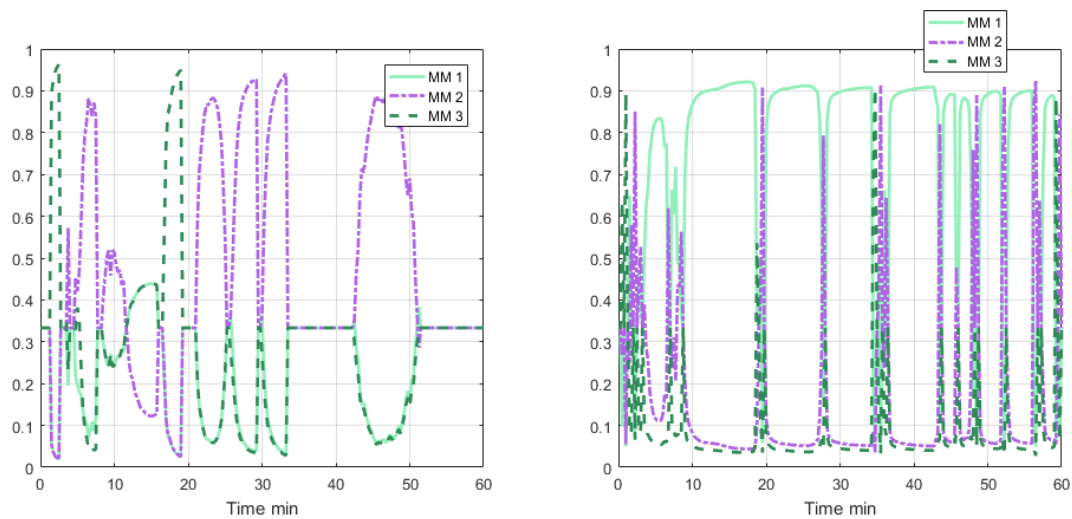


(c) Enhanced HME MM3 Gating Weights



(d) Original HME MM3 Gating Weights

Figure 4.28 Experiment 6 - HME MM Gating Weights



(a) Enhanced HME TLGN Gating Weights

(b) Original HME TLGN Gating Weights

Figure 4.29 Experiment 6 - HME TLGN Gating Weights

In this experiment, the enhanced HME was able to converge to the solution closest to the true value, while the original HME could not. This experiment demonstrates the value of the enhanced HME explained in this thesis and used in the simulation experiments in certain situations. This experiment also demonstrated the ability of the enhanced HME to select model characteristics where the true value is not in the bank of filters (or HME nodes).

## 4.9 Experiment 7 - Dual Layer HME Monte Carlo

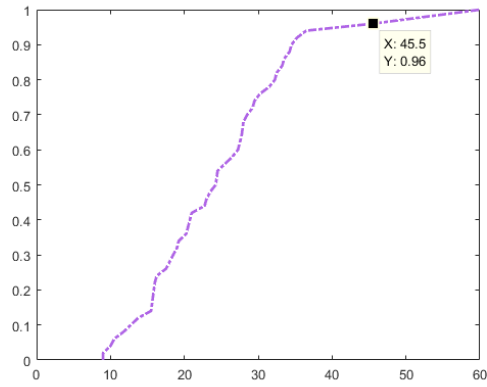
The purpose of this experiment is to perform a Monte Carlo set of 50 experiments varying the random seed. Varying the random seed causes a different initial estimate to be drawn from a normal distribution with zero-mean and the  $1\sigma$  values shown in Table 4.5. In addition, changing the random seed causes different measurement noise values to be drawn from a normal distribution with zero mean and the  $1\sigma$  values drawn from the CCD error

model shown in Sections 3.4.2 and 3.4.3 and the angle errors mentioned in equation (3.130) and equation (3.131). The Monte Carlo runs show how the scenario performs with different initial state estimates and measurement noise values drawn from the same  $1\sigma$  uncertainties. This experiment used most of the settings in Section 4.8. The initial nominal model is shown in Table 4.28, while the two macromodes of reflectance and size models are shown in Tables 4.26 and 4.27 respectively.

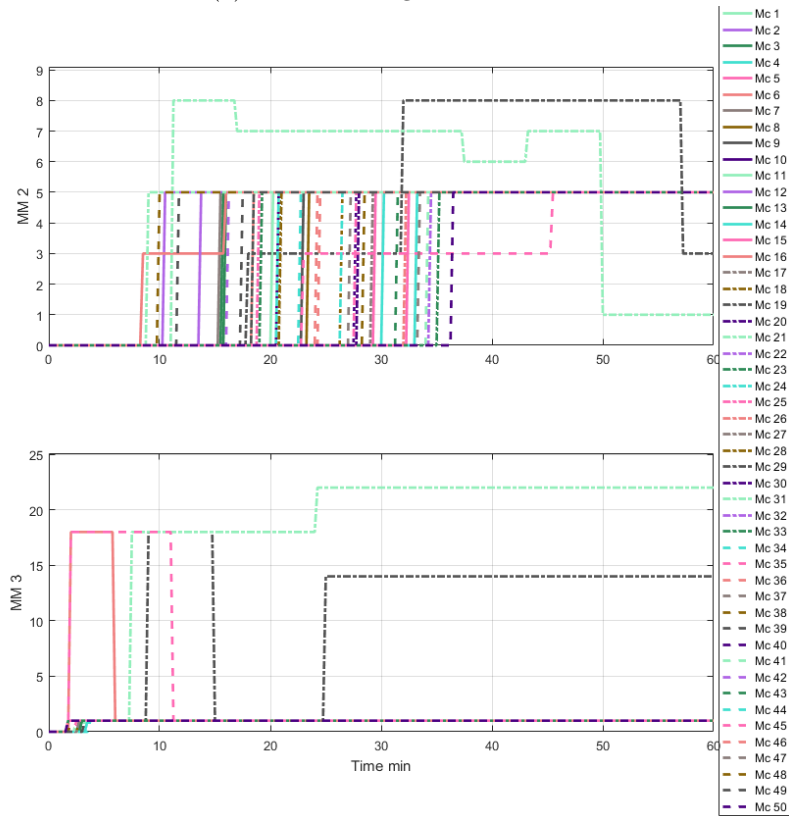
<b>Node#</b>	<b>Reflectance</b>	<b>Size</b>
1	[0.60, 0.475, 0.541, 850, 850]	[3.5, 3.0, 2.4]

Table 4.28 Experiment 7 MM1 - Nominal Model

The resultant plot of the cumulative distribution function (CDF) of the convergence times as well as the plot of all the selected macromodes are shown in Figure 4.30. These results show that 96% of the macromodes converged to the closest macromode nodes to the true value. The closest macromode nodes to the true value in this experiment are node 5 for MM2 and node 1 for MM3. Thus, Monte Carlo sets 19 and 21 did not converge to the proper nodes.



(a) CDF Convergence Plot



(b) Selected Nominal Index

Figure 4.30 Experiment 7 - HME Selected Nominal Index

## 4.10 Summary Results

A summary of the results of the experiments are shown in Table 4.29.

#	Success	Conclusions
1	Yes	Both MMAE and single layer ME converged to the correct size node very quickly
2	Yes	Both MMAE and single layer ME converged to correct nodes; although the ME converges sooner
3	Yes	Both the MMAE and HME converged to the correct size and reflectance nodes; however, HME converged about 68% and ran about 80% quicker on the CPU than the MMAE
4a	Yes	Both the MMAE and HME converged to the correct size nodes; HME converged quicker than the MMAE
4b	No	MMAE underflowed; HME did not converge to closest overall size to truth; HME ended with about a 46.25% probability on the node which was not the closest to the overall size to truth
4c	No	MMAE underflowed; HME did not converge to closest overall size to truth; the HME results showed very low probability (< 20%) for a few choices combined
5	Yes	Both the MMAE and HME eventually detect the closest node to the true out of plane maneuver as the most probable
5b	No	Both the MMAE and HME cannot detect with high probability the closest node to the true in-track maneuver as the most probable; although, the closest node to true is technically the most probable, just with much lower probability ( 40%) and just barely more probable than other nodes, than in experiment 5 - out of plane maneuver detection;
6	Yes	Enhanced HME converged to the closest nodes and macromodes to the truth while the original HME could not
7	Yes	96% of nodes and macromodes converged to the closest nodes and macromodes to the truth

Table 4.29 Experiment Conclusion Summary

## Chapter 5

# Conclusions

In this thesis, an improvement to dual-layer hierarchical mixture of experts which can assist in resident space object model characterization was demonstrated. This work focused on the problem of a resident space object in geosynchronous orbit using optical telescopes with charge coupled device cameras collecting non-resolved images. The extended Kalman filter and unscented Kalman filter were discussed along with different methods of attitude estimation. The dynamic processes modeled in this work were then discussed along with the different measurement models. Then the adaptive estimation techniques were discussed along with a new method of computing the top level gating network and introducing a decaying learning parameter for the dual-layer hierarchical mixture of experts. The experiments showed that a single layer mixture of experts behaves as well or better than a multiple model adaptive estimation. The hierarchical mixture of experts showed that it can converge more quickly and is more sensitive to changes than the multiple model adaptive estimation. In addition, in cases of poor coverage of the parameters space in the bank of filters, the hierarchical mixture of experts is more resilient to numerical underflow than the multiple mode adaptive estimation. This work demonstrated the potential usefulness and

viability of either a single layer mixture of experts and dual-layer hierarchical mixture of experts. The experimental results showed that both the MMAE and HME struggled when the true size was not close to any value in the filter bank. However, the HME gave results which indicated that no model was very probable. Thus, this knowledge can allow one to know that the true value is not close to any value contained within the filter bank. Overall, the HME is a useful tool in the adaptive estimation techniques toolbox that warrants future study.

## 5.1 Future Work

This thesis provides a foundation for using a single layer ME and dual-layer HME for estimating RSO characteristics. However, more work is needed to expand on this. Some future work ideas are listed here:

1. Expand the estimation over several days with multiple observers with long delays in between measurements
2. Measurements should be taken from multiple spectra instead of the single 551 nm centered visible light measurement modeled in this work
3. More work should be done to investigate the proper usage and tuning of the learning rate function
4. Work should be done to investigate the effect of different CCD and telescope parameters have on the ability to estimate RSO models and states
5. Use much more sophisticated and realistic shape models along with different BRDF models

6. Validate the use of BRDF models with real light curve data from a RSO in geosynchronous orbit
7. Use different BRDF models and use HME macromodes to determine which BRDF model is closest to the truth
8. Investigate integrating this work with ESA's very sophisticated Orekit framework and test on real data pending validation of the light curve models; in addition, investigate using much higher fidelity models for the truth models and adding appropriate process noise to the filters; the higher fidelity models should contain uncertainty in the physics of solar luminosity, gravity models, etc. to be as realistic as possible in a simulation prior to testing with the real data
9. Investigate either running different models for the truth and process models or add 'realistic' random variations to the true models such as biases and noise as the real physics of the dynamics can never be perfectly understood. Thus, the process model should never use the exact same model as the truth unless realistic noise and bias terms are added to the truth to account for the real uncertainty in gravity, SRP, and other models. The filters can account for this uncertainty in the process noise model
10. Investigate running Monte Carlo simulations where the truth initial conditions move around the uncertainty ellipsoid while the noise values remain the same
11. Test a vast range of sizes and surface material reflectance properties where some of the size and reflectance choices couple into each as related to apparent magnitude; then, see if the filters over a long period of time can use SRP to select the proper model
12. Investigate the use of other estimation filters such as the divided difference filter

- (DDF) and see how it compares to the UKF in estimating states, especially for cases where the UKF suffers either numerical or divergence problems
13. Investigate the use of HME macromodes with different learning rate parameters so that some banks can be used for change detection and others for noise rejection
  14. Investigate the mathematical rationale for using the function  $Idx(max(g_{l,i}))$  and attempt to provide a more solid theoretical backing as to why this ‘appears’ to work better
  15. Investigate trying to estimate the moment of inertia ratios of RSOs
  16. Using real data from different telescopes and CCD cameras determine realistic angle measurement noise models from these different devices
  17. Run more scenarios with different in-track and out of plane station-keeping maneuvers to determine which are easier to detect when in GEO.
  18. Investigate the use of artificial intelligence (AI) techniques which are solving big data problems in the field of computer science

# Bibliography

- [1] NASA Lyndon B. Johnson Space Center. Top ten satellite breakups reevaluated. *Orbital Debris Quarterly News*, 20(1 and 2):5–6, April 2016.
- [2] Akhter Nafi, Arun Bernard, and Kohei Fujimoto. A unified approach for optical survey strategy design of resident space objects. In *AIAA/AAS Astrodynamics Specialist Conference*, page 5500, 2016.
- [3] Adam C Snow, Johnny L Worthy III, Angela den Boer, Luke J Alexander, Marcus J Holzinger, and David Spencer. Optimization of cubesat constellations for uncued electrooptical space object detection and tracking. *Journal of Spacecraft and Rockets*, pages 401–419, 2016.
- [4] Tyler J Hardy and Stephen C Cain. Investigating prior probabilities in a multiple hypothesis test for use in space domain awareness. In *SPIE Defense+ Security*, pages 983804–983804. International Society for Optics and Photonics, 2016.
- [5] Piyush M Mehta, Gonzalo Blanco Arnao, Davide Bonetti, Edmondo Minisci, and Massimiliano Vasile. Computer graphics for space debris. In *Debris, Safety and Awareness*. 6th International Conference on Astrodynamics Tools and Techniques, 2016.

- 
- [6] JC Liou. Usa space debris environment, operations, and measurement updates. *52nd Session of the Scientific and Technical Subcommittee Committee on the Peaceful Uses of Outer Space*, 2015.
- [7] NASA Lyndon B. Johnson Space Center. Monthly number of objects in earth orbit by object type. *Orbital Debris Quarterly News*, 20(1 and 2):14, April 2016.
- [8] SM Lederer, EG Stansbery, HM Cowardin, P Hickson, LF Pace, KJ Abercromby, PW Kervin, and RJ Alliss. The nasa meter class autonomous telescope: Ascension island. Technical report, DTIC Document, 2013.
- [9] Oliver Montenbruck and Eberhard Gill. *Satellite Orbits*. Springer, Weßling, Deutschland, 2005.
- [10] Charles J Wetterer, C Channing Chow, John L Crassidis, Richard Linares, and Moriba K Jah. Simultaneous position, velocity, attitude, angular rates, and surface parameter estimation using astrometric and photometric observations. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 997–1004. IEEE, 2013.
- [11] D. T. Magill. Optimal adaptive estimation of sampled stochastic processes. *IEEE Transactions on Automatic Control*, AC-10(4):434–439, October 1965.
- [12] Wassim S Chaer, Robert H Bishop, and Joydeep Ghosh. A mixture-of-experts framework for adaptive kalman filtering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(3):452–464, 1997.
- [13] Timothy Price Crain. *Adaptive Interplanetary Orbit Determination*. PhD thesis, The University of Texas at Austin, 2000.

- 
- [14] Richard Linares, Moriba K Jah, John L Crassidis, and Christopher K Nebelecky. Space object shape characterization and tracking using light curve and angles data. *Journal of Guidance, Control, and Dynamics*, 37(1):13–25, 2013.
- [15] Richard Linares. *Probabilistic identification and discrimination of deep space objects via astrometric and photometric data fusion*. PhD thesis, State University of New York at Buffalo, 2013.
- [16] M Jah and Ronald A Madler. Satellite characterization: angles and light curve data fusion for spacecraft state and parameter estimation. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, volume 49, 2007.
- [17] Richard Linares, Moriba K Jah, and John L Crassidis. Space object area-to-mass ratio estimation using multiple model approaches. *Advances in the Astronautical Sciences*, 144:55–72, 2012.
- [18] David Gaylor and Jessica Anderson. Use of hierarchical mixtures of experts to detect resident space object attitude. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, volume 1, page 70, 2014.
- [19] David E Gaylor and Jessica T. Anderson. Resident space object feature identification and attitude detection using hierarchical mixtures of experts. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2013.
- [20] Liansheng Wang and Yuanqing Xia. Uncertain parameters identification based on hierarchical adaptive filters during mars entry. In *Variable Structure Systems (VSS), 2016 14th International Workshop on*, pages 373–378. IEEE, 2016.
- [21] John L Crassidis, F Landis Markley, and Yang Cheng. Survey of nonlinear attitude estimation methods. *Journal of guidance, control, and dynamics*, 30(1):12–28, 2007.

- 
- [22] Hanspeter Schaub and John L Junkins. Stereographic orientation parameters for attitude dynamics: A generalization of the rodrigues parameters. *Journal of the Astronautical Sciences*, 44(1):1–19, 1996.
- [23] John L Crassidis and F Landis Markley. Unscented filtering for spacecraft attitude estimation. *Journal of guidance, control, and dynamics*, 26(4):536–542, 2003.
- [24] Murty S Challa, Jay G Moore, and Daniel J Rogers. A simple attitude unscented kalman filter: Theory and evaluation in a magnetometer-only spacecraft scenario. *IEEE Access*, 4:1845–1858, 2016.
- [25] Mikko Kaasalainen and Johanna Torppa. Optimization methods for asteroid lightcurve inversion: I. shape determination. *Icarus*, 153(1):24–36, 2001.
- [26] Mikko Kaasalainen, Johanna Torppa, and Karri Muinonen. Optimization methods for asteroid lightcurve inversion: Ii. the complete inverse problem. *Icarus*, 153(1):37–51, 2001.
- [27] R Szabó, A Pál, K Sárneczky, Gy M Szabó, L Molnár, LL Kiss, O Hanyecz, E Plachy, Cs Kiss, K Vida, et al. Uninterrupted optical light curves of main-belt asteroids from the k2 mission. *arXiv preprint arXiv:1609.02759*, 2016.
- [28] Andrii Kokorev and Oleksiy Golubov. Modeling of asteroid shapes. *arXiv preprint arXiv:1601.06967*, 2016.
- [29] Dilini Subasinghe, Margaret D Campbell-Brown, and Edward Stokan. Physical characteristics of faint meteors by light curve and high-resolution observations, and the implications for parent bodies. *Monthly Notices of the Royal Astronomical Society*, 457(2):1289–1298, 2016.

- 
- [30] Marcus J Holzinger, Kyle T Alfriend, Charles J Wetterer, K Kim Luu, Chris Sabol, and Kris Hamada. Photometric attitude estimation for agile space objects with shape uncertainty. *Journal of Guidance, Control, and Dynamics*, 37(3):921–932, 2014.
- [31] Charles J Wetterer, Bobby Hunt, Paul Kervin, and Moriba Jah. Comparison of unscented kalman filter and unscented schmidt kalman filter in predicting attitude and associated uncertainty of a geosynchronous satellite. Technical report, DTIC Document, 2014.
- [32] Charles J Wetterer, Richard Linares, John L Crassidis, Thomas M Kelecy, Marek K Ziebart, Moriba K Jah, and Paul J Cefola. Refining space object radiation pressure modeling with bidirectional reflectance distribution functions. *Journal of Guidance, Control, and Dynamics*, 37(1):185–196, 2013.
- [33] Rosana Montes Soldado and Carlos Ureña Almagro. An overview of brdf models. Technical report, Dept. Lenguajes y Sistemas Informáticos, University of Granada, Granada, Spain, 2012.
- [34] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. 1979.
- [35] Arthur Gelb. *Applied optimal estimation*. MIT press, 1974.
- [36] Byron D. Tapley, Bob E. Shutz, and George H. Born. *Statistical Orbit Determination*. Elsevier Academic Press, London, UK, 2010.
- [37] Robert Grover Brown and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*. John Wiley & Sons, Danvers, Massachusetts, 4th edition, 2012.

- 
- [38] John L. Crassidis and John L. Junkins. *Optimal Estimation of Dynamic Systems*. CRC Press, Boca Raton, FL, 2nd edition, 2012.
- [39] JFG De Freitas, Mahesan Niranjan, AH Gee, and Arnaud Doucet. *Sequential Monte Carlo methods for optimisation of neural network models*. Cambridge University Engineering Department, Cambridge, England, technical report tr-328 edition, 1998. Citeseer.
- [40] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [41] Simon J. Julier and Jeffery K. Uhlmann. New extension of the kalman filter to nonlinear systems. In *International Society for Optics and Photonics, AeroSense'97*, 1997.
- [42] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium.*, pages 153–158. IEEE, AS-SPCC, 2000.
- [43] Nicholas J Higham. *Accuracy and stability of numerical algorithms*, chapter 10, pages 195–212. Siam, 2002.
- [44] Eric A Wan and Rudolph van der Merwe. The unscented kalman filter, chapter 7. In *Simon Haykin ed., Kalman Filtering and Neural Networks*. Wiley, New York, NY, 2001.
- [45] Robert Grover Brown and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*, chapter 6, pages 216–226. John Wiley & Sons, Danvers, Massachusetts, 4th edition, 2012.

- 
- [46] John L. Crassidis and John L. Junkins. *Optimal Estimation of Dynamic Systems*, chapter 4, pages 249–256. CRC Press, Boca Raton, FL, 2nd edition, 2012.
- [47] Robert F. Stengel. *Optimal Control and Estimation*, chapter 4, pages 402–407. Dover Publications, New York, NY, 1994.
- [48] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.
- [49] Wassim S. Chaer, Robert H. Bishop, and Joydeep Ghosh. A mixture-of-experts framework for adaptive kalman filtering. *IEEE Transactions on Systems, Man, And Cybernetics - Part B: Cybernetics*, 27(3):452–464, 1997.
- [50] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.
- [51] Dayne G. Cook. Solar radiation pressure modeling issues for high altitude satellites. Master’s thesis, US Air Force Institute of Technology, 2001.
- [52] *Resolution B2 - on recommended zero points for the absolute and apparent bolometric magnitude scales*, 2015. The XXIXth International Astronomical Union General Assembly.
- [53] *Physics & Math of Shading*, 2015. SIGGRAPH - The 42nd International Conference and Exhibition on Computer Graphics and Interactive Techniques.
- [54] David A. Vallado. *Fundamentals of Astrodynamics and Applications*, chapter 8, pages 538–550. Space Technology Library, 4th edition, 2013.
- [55] David A. Vallado. *Fundamentals of Astrodynamics and Applications*, chapter 9, pages 641–671. Space Technology Library, 4th edition, 2013.

- [56] *Department of Defense World Geodetic Systems 1984*. Unite States of America National Imagery and Mapping Agency, tr8350.2 - amendment 1 - 3rd edition, 2000.
- [57] Nikolas Trawny and Stergios I Roumeliotis. Indirect kalman filter for 3d attitude estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, 2, 2005.
- [58] Oliver Montenbruck and Eberhard Gill. *Satellite Orbits*, chapter 4, pages 118–132. Springer, Weßling, Deutschland, 2005.
- [59] Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. *Real-time rendering*. CRC Press, 3rd edition, 2008.
- [60] Michael Ashikhmin and Peter Shirley. An anisotropic phong brdf model. *Journal of graphics tools*, 5(2):25–32, 2000.
- [61] *SBIG STT-8300M*. Diffraction Limited, 2014. URL <https://www.sbig.com/products/cameras/stt/stt-8300m/>.
- [62] Gerald C Holst. *CCD arrays, cameras, and displays*, chapter 4, pages 102–144. Cite-seer, 1998.
- [63] Y Reibel, M Jung, M Bouhifd, B Cunin, and C Draman. Ccd or cmos camera noise characterisation. *The European Physical Journal Applied Physics*, 21(1):75–80, 2003.
- [64] Michael Bolte. Signal-to-noise in optical astronomy. Lecutre Notes - AY-257 - Modern Observational Techniques - University of California Observatories - Keck Observatory, 2006. URL [http://www.ucolick.org/~bolte/AY257/s\\_n.pdf](http://www.ucolick.org/~bolte/AY257/s_n.pdf).
- [65] Thomas J. Fellers, Kimberly M. Vogt, and Michael W. Davidson. Ccd signal-to-noise ratio. Technical report, National High Magnetic Field Laboratory, 2016. URL <https://www.microscopyu.com/tutorials/ccd-signal-to-noise-ratio>.

- 
- [66] Photometrics. Keep the noise down! low noise: An integral part of high-performance ccd (hccd) camera systems. Technical report, Photometrics, 2010. URL <http://www.photometrics.com/resources/technotes/pdfs/snr.pdf>.
- [67] *Keck Telescope and Facility Instrument Guide*. Keck Observatory - California Institute of Technology, 2002. URL <https://www2.keck.hawaii.edu/observing/kecktelgde/ktelinstupdate.pdf>.
- [68] Kathryn F Neugent and Philip Massey. The spectrum of the night sky over kitt peak: Changes over two decades. *Publications of the Astronomical Society of the Pacific*, 122(896):1246, 2010.
- [69] Steve B Howell. *Handbook of CCD astronomy*, volume 5. Cambridge University Press, 2006.
- [70] David A. Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm Press, Hawthorne, California, 2013.
- [71] Malcolm D Shuster. A survey of attitude representations. *Navigation*, 8(9):439–517, 1993.
- [72] David A. Vallado. *Fundamentals of Astrodynamics and Applications*, chapter 3, page 157. Space Technology Library, 4th edition, 2013.
- [73] Michael I Jordan and Robert A Jacobs. Hierarchies of adaptive experts. In *NIPS*, pages 985–992, 1991.