Linear Covariance Analysis For Gimbaled Pointing Systems

Randall S. Christensen
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/etd

Part of the Aerospace Engineering Commons, and the Mechanical Engineering Commons

Recommended Citation
Christensen, Randall S., "Linear Covariance Analysis For Gimbaled Pointing Systems" (2013). All Graduate Theses and Dissertations. 1766.
https://digitalcommons.usu.edu/etd/1766

This Dissertation is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.
LINEAR COVARIANCE ANALYSIS FOR GIMBALED POINTING SYSTEMS

by

Randall S Christensen

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Mechanical Engineering

Approved:

Dr. David K. Geller
Major Professor

Dr. R. Rees Fullmer
Committee Member

Dr. Steven R. Wassom
Committee Member

Dr. Jake Gunther
Committee Member

Dr. Stephen A. Whitmore
Committee Member

Dr. Mark McLellan
Vice President for Research and
Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2013
Abstract

Linear Covariance Analysis for Gimbaled Pointing Systems

by

Randall S Christensen, Doctor of Philosophy
Utah State University, 2013

Major Professor: Dr. David K. Geller
Department: Mechanical and Aerospace Engineering

Linear covariance analysis has been utilized in a wide variety of applications. Historically, the theory has made significant contributions to navigation system design and analysis. More recently, the theory has been extended to capture the combined effect of navigation errors and closed-loop control on the performance of the system. These advancements have made possible rapid analysis and comprehensive trade studies of complicated systems ranging from autonomous rendezvous to vehicle ascent trajectory analysis. Comprehensive trade studies are also needed in the area of gimbaled pointing systems where the information needs are different from previous applications. It is therefore the objective of this research to extend the capabilities of linear covariance theory to analyze the closed-loop navigation and control of a gimbaled pointing system.

The extensions developed in this research include modifying the linear covariance equations to accommodate a wider variety of controllers. This enables the analysis of controllers common to gimbaled pointing systems, with internal states and associated dynamics as well as actuator command filtering and auxiliary controller measurements. The second extension is the extraction of power spectral density estimates from information available in linear covariance analysis. This information is especially important to gimbaled pointing systems where not just the variance but also the spectrum of the pointing error impacts the performance. The extended theory is applied to a model of a gimbaled pointing system which includes both flexible and rigid body elements as well as input disturbances, sensor errors, and actuator errors. The results of the analysis are validated by direct comparison to a Monte Carlo-based analysis approach. Once the developed linear covariance theory is validated, analysis techniques that are often prohibitory with Monte Carlo analysis are
used to gain further insight into the system. These include the creation of conventional error budgets through sensitivity analysis and a new analysis approach that combines sensitivity analysis with power spectral density estimation. This new approach resolves not only the contribution of a particular error source, but also the spectrum of its contribution to the total error. In summary, the objective of this dissertation is to increase the utility of linear covariance analysis for systems with a wide variety of controllers and for whom the spectrum of the errors is critical to performance.
Linear covariance analysis has been utilized in a wide variety of applications. Historically, the theory has made significant contributions to navigation system design and analysis. More recently, the theory has been extended to capture the combined effect of navigation errors and closed-loop control on the performance of the system. These advancements have made possible rapid analysis and comprehensive trade studies of complicated systems ranging from autonomous rendezvous to vehicle ascent trajectory analysis. Comprehensive trade studies are also needed in the area of gimbaled pointing systems where the information needs are different from previous applications. It is therefore the objective of this research to extend the capabilities of linear covariance theory to analyze the closed-loop navigation and control of a gimbaled pointing system.

The extensions developed in this research comprise two areas. The first is related to generalizing the linear covariance controller models. Previous controller models have been somewhat limited in their applicability to controllers with internal states, a common feature in gimbaled pointing systems. This research extends the controller model to allow for accurate modeling of such controllers. The second extension is related to characterizing the frequency content of the pointing errors. In previous applications of linear covariance, the focus has been on computing the total error at some critical mission time. No consideration was given to the bandwidth of the error, i.e. whether the error varied slowly or quickly with time. The analysis method developed in this research enables the designer to identify not only the magnitude of the error sources, but the portion of the spectrum to which they contribute. This knowledge is extremely valuable in areas such as satellite imagery, weapon stabilization, and remote sensing. In summary, the objective of this dissertation is to increase the utility of linear covariance analysis for systems with a wide variety of controllers and for whom the spectrum of the errors is critical to performance. The extended theory is applied to a model of a gimbaled pointing system and validated by direct comparison to conventional analysis techniques. The results show excellent correlation with conventional methods, but at the drastically lower computational load typical of linear covariance analysis. This efficiency enables comprehensive trade studies, allowing the system designer to span the entire design space and select the system configuration with the lowest cost and complexity that still meets mission requirements.

Randall S Christensen
Acknowledgments

I’m sure it is the case for most people that the older they get, the more they realize that all they have has been made possible by the goodness of others. It is with this in mind that I would like to give sincere thanks to the following organizations and people.

First, I would like to thank the Space Dynamics Laboratory and the Utah State University Research Foundation for their support in both the finances and time required to complete this degree. Without their support, this would have been a much rougher road.

Second, I wish to thank all members of my committee for their support and guidance throughout my undergraduate, graduate, and professional years. Many of the skills I use on a daily basis originate from a course, project, or office discussion that I had with these good people. Special thanks go to Dr. Jacob Gunther for helping navigate what to me was the relatively new area of signal processing. Special thanks also go to Dr. David Geller for his professional expertise, dedication to academics, and contagious enthusiasm. It is rare to have the privilege of working with somebody so supportive and genuinely kind as Dr. Geller.

Third is to my parents, Zan and Dena Christensen. A wise man once said, “All that I am, or hope to be, I owe to my angel mother,” and I might add (literally) to my angel father. It is from them that I learned a strong work ethic and a desire to strive for perfection. They both continue to be a strong influence for good in my life and cannot be thanked enough.

Fourth is to my wife, Talea, and my four children: Kyle, Britnie, Abbey, and Emily. So much of what I do is motivated by my love for them, and they should be thanked for the hours of course-work and dissertation work that they endured. While trying to explain what I have been doing these last 6 years, I showed Talea some plots and talked about pointing mirrors and the effects of jitter. After enduring my explanation, she joked that she would tell others that I basically “played with pointy things and learned the jitter bug!” While I hope these years have been a little more productive than that, I will be forever grateful for her sense of humor and support.

Finally, thanks be to God from whom all blessings flow and to country for the opportunities of education and success.

Randall S Christensen
# Contents

Abstract ................................................................. iii
Public Abstract ......................................................... v
Acknowledgments ........................................................ vi
List of Tables ........................................................... ix
List of Figures ........................................................... x
Acronyms ................................................................. xiv

1 Introduction ............................................................ 1
  1.1 Monte Carlo Analysis ............................................. 2
  1.2 Linear Covariance Analysis ...................................... 4

2 Thesis ................................................................. 6

3 Related Literature .................................................... 8
  3.1 Navigation and Pointing Performance Analysis of Gimbaled Pointing Systems ........ 8
  3.2 Classical vs. Modern Linear Covariance Analysis ......................... 9
  3.3 Frequency Domain Analysis of Random Processes ....................... 10

4 Theoretical Development ............................................. 12
  4.1 Notation ............................................................ 12
  4.2 Nonlinear Truth Model ............................................ 12
  4.3 Mean Reference Trajectory ....................................... 15
  4.4 Linear Modeling .................................................. 17
    4.4.1 Truth State Dynamics ....................................... 17
    4.4.2 Navigation State Dynamics ................................ 19
    4.4.3 Controller State Dynamics ................................. 21
    4.4.4 True Actuator Response ................................... 22
    4.4.5 True, Navigation, and Controller State Dispersion Update ... 22
    4.4.6 True Navigation Errors ..................................... 23
    4.4.7 Summary of Linearized Equations .......................... 24
  4.5 Covariance Models ............................................... 24
    4.5.1 Covariance Equations ....................................... 25
    4.5.2 Performance Evaluation ................................... 27
    4.5.3 Covariance Initialization .................................. 28
  4.6 Frequency Domain Analysis ..................................... 29
    4.6.1 Monte Carlo Approach ....................................... 29
    4.6.2 LinCov Autocorrelation Approach .......................... 31
    4.6.3 LinCov Transfer Matrix Approach ........................... 37
    4.6.4 Frequency Domain Frequency Analysis Summary ............... 41
    4.6.5 LinCov/Monte Carlo Comparison ............................ 42

5 Precision Pointing Simulation ..................................... 52
  5.1 Structural Dynamics Model ...................................... 53
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.1</td>
<td>Disturbance Sources</td>
<td>55</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Free Body Diagrams</td>
<td>55</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Kinematic Constraints</td>
<td>56</td>
</tr>
<tr>
<td>5.1.4</td>
<td>Equations of Motion</td>
<td>56</td>
</tr>
<tr>
<td>5.1.5</td>
<td>State Space Representation</td>
<td>58</td>
</tr>
<tr>
<td>5.1.6</td>
<td>Model validation</td>
<td>61</td>
</tr>
<tr>
<td>5.2</td>
<td>Navigation Algorithm Development</td>
<td>62</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Truth Model</td>
<td>66</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Design Model</td>
<td>67</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Navigation Filter</td>
<td>68</td>
</tr>
<tr>
<td>5.3</td>
<td>Control Algorithm Development</td>
<td>70</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Truth Model</td>
<td>70</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Controller and Anti-Resonance Filter Design</td>
<td>72</td>
</tr>
<tr>
<td>5.3.3</td>
<td>State-Space Realization</td>
<td>79</td>
</tr>
<tr>
<td>6</td>
<td>Simulation Overview</td>
<td>86</td>
</tr>
<tr>
<td>6.1</td>
<td>Monte Carlo</td>
<td>86</td>
</tr>
<tr>
<td>6.2</td>
<td>Linear Covariance</td>
<td>94</td>
</tr>
<tr>
<td>6.3</td>
<td>Simulation Parameters Summary</td>
<td>97</td>
</tr>
<tr>
<td>6.4</td>
<td>Initial Conditions</td>
<td>101</td>
</tr>
<tr>
<td>7</td>
<td>Pointing Analysis</td>
<td>106</td>
</tr>
<tr>
<td>7.1</td>
<td>Mean Reference Trajectory</td>
<td>106</td>
</tr>
<tr>
<td>7.2</td>
<td>Monte Carlo Vs. Linear Covariance</td>
<td>109</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Hair plots</td>
<td>109</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Autocorrelation Function Estimation</td>
<td>133</td>
</tr>
<tr>
<td>7.2.3</td>
<td>PSD computation</td>
<td>139</td>
</tr>
<tr>
<td>7.3</td>
<td>Navigation and Control Analysis</td>
<td>150</td>
</tr>
<tr>
<td>7.4</td>
<td>Spectral Sensitivity Analysis</td>
<td>157</td>
</tr>
<tr>
<td>8</td>
<td>Conclusions</td>
<td>167</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>170</td>
</tr>
<tr>
<td>Curriculum Vitae</td>
<td></td>
<td>175</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Method comparison, assumptions</td>
<td>43</td>
</tr>
<tr>
<td>4.2</td>
<td>Method comparison, system features</td>
<td>43</td>
</tr>
<tr>
<td>4.3</td>
<td>Stationary simulation parameters</td>
<td>44</td>
</tr>
<tr>
<td>4.4</td>
<td>Stationary simulation parameters</td>
<td>47</td>
</tr>
<tr>
<td>4.5</td>
<td>Stationary simulation parameters</td>
<td>49</td>
</tr>
<tr>
<td>5.1</td>
<td>Model parameter descriptions</td>
<td>54</td>
</tr>
<tr>
<td>5.2</td>
<td>Physical system parameters used for validating the model.</td>
<td>63</td>
</tr>
<tr>
<td>5.3</td>
<td>Impulse values</td>
<td>63</td>
</tr>
<tr>
<td>6.1</td>
<td>Structural dynamics model parameters</td>
<td>99</td>
</tr>
<tr>
<td>6.2</td>
<td>Disturbance state time constants</td>
<td>99</td>
</tr>
<tr>
<td>6.3</td>
<td>Parameter state time constants</td>
<td>100</td>
</tr>
<tr>
<td>6.4</td>
<td>Disturbance steady state standard deviation $1\sigma$</td>
<td>100</td>
</tr>
<tr>
<td>6.5</td>
<td>Parameter steady-state standard deviation $1\sigma$</td>
<td>100</td>
</tr>
<tr>
<td>6.6</td>
<td>Measurement noise $1\sigma$</td>
<td>101</td>
</tr>
<tr>
<td>6.7</td>
<td>Simulations times</td>
<td>102</td>
</tr>
<tr>
<td>7.1</td>
<td>Subsystem definition</td>
<td>158</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Generic Monte Carlo simulation for GN&amp;C analysis</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Confidence intervals</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>Proposed extensions of Linear Covariance theory</td>
<td>7</td>
</tr>
<tr>
<td>4.1</td>
<td>Time diagram</td>
<td>32</td>
</tr>
<tr>
<td>4.2</td>
<td>Covariance for the stationary simulation</td>
<td>45</td>
</tr>
<tr>
<td>4.3</td>
<td>Autocorrelation function for the stationary simulation</td>
<td>46</td>
</tr>
<tr>
<td>4.4</td>
<td>PSD for the stationary simulation</td>
<td>46</td>
</tr>
<tr>
<td>4.5</td>
<td>Covariance for the non-stationary simulation</td>
<td>48</td>
</tr>
<tr>
<td>4.6</td>
<td>Autocorrelation function for the non-stationary simulation</td>
<td>48</td>
</tr>
<tr>
<td>4.7</td>
<td>PSD for the non-stationary simulation</td>
<td>49</td>
</tr>
<tr>
<td>4.8</td>
<td>Covariance for highly non-stationary simulation</td>
<td>50</td>
</tr>
<tr>
<td>4.9</td>
<td>Autocorrelation for the highly non-stationary simulation</td>
<td>50</td>
</tr>
<tr>
<td>4.10</td>
<td>PSD for the highly non-stationary simulation</td>
<td>51</td>
</tr>
<tr>
<td>5.1</td>
<td>Conceptual diagram of a gimbaled pointing system</td>
<td>53</td>
</tr>
<tr>
<td>5.2</td>
<td>Free body diagrams</td>
<td>56</td>
</tr>
<tr>
<td>5.3</td>
<td>ADAMS model</td>
<td>61</td>
</tr>
<tr>
<td>5.4</td>
<td>SM inertial angle impulse response due to the force ($F$) acting on the spacecraft. Left-SM joint angle for the derived and ADAMS models. Right-Difference between derived and ADAMS models.</td>
<td>62</td>
</tr>
<tr>
<td>5.5</td>
<td>SM inertial angle impulse response due to the torque ($T$) acting on the spacecraft. Left-SM joint angle for the derived and ADAMS models. Right-Difference between derived and ADAMS models.</td>
<td>64</td>
</tr>
<tr>
<td>5.6</td>
<td>SM inertial angle impulse response due to the SM motor torque ($T_m$). Left-SM joint angle for the derived and ADAMS models. Right-Difference between derived and ADAMS models.</td>
<td>64</td>
</tr>
<tr>
<td>5.7</td>
<td>SM inertial angle impulse response due to the disturbance torque ($T_D$). Left-SM joint angle for the derived and ADAMS models. Right-Difference between derived and ADAMS models.</td>
<td>65</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>5.8</td>
<td>SM inertial angle impulse response due to the disturbance torque ($T_D$). Left-SM joint angle for the derived and ADAMS models. Right-Difference between derived and ADAMS models.</td>
<td></td>
</tr>
<tr>
<td>5.9</td>
<td>SM attitude control block diagram</td>
<td></td>
</tr>
<tr>
<td>5.10</td>
<td>SM transfer function.</td>
<td></td>
</tr>
<tr>
<td>5.11</td>
<td>Frequency response of the plant $G_{SM}(s)$ in blue, anti-resonance filter $H_f(s)$ in green, and modified plant $H_f(s) \cdot G_{SM}(s)$ in red.</td>
<td></td>
</tr>
<tr>
<td>5.12</td>
<td>Inner OLT TF frequency response. Phase margin = 77 deg at 8Hz. Gain margin = 14.8dB at 43Hz.</td>
<td></td>
</tr>
<tr>
<td>5.13</td>
<td>Inner CL TF frequency response with a -3dB bandwidth = 11Hz.</td>
<td></td>
</tr>
<tr>
<td>5.14</td>
<td>Frequency response of the plant for the outer control loop.</td>
<td></td>
</tr>
<tr>
<td>5.15</td>
<td>Frequency response of the outer loop controller.</td>
<td></td>
</tr>
<tr>
<td>5.16</td>
<td>Frequency response of the outer OLT TF. Phase margin = 63 deg at 3 Hz. Gain margin = 20.1dB at 17 Hz.</td>
<td></td>
</tr>
<tr>
<td>5.17</td>
<td>Frequency response of the outer CL TF with a -3dB bandwidth = 5Hz.</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Key simulation times</td>
<td></td>
</tr>
<tr>
<td>7.1</td>
<td>Nominal attitude (top) and angular rates (bottom) of system components</td>
<td></td>
</tr>
<tr>
<td>7.2</td>
<td>Nominal inertial position (top) and velocity (bottom) of system components</td>
<td></td>
</tr>
<tr>
<td>7.3</td>
<td>Nominal relative attitude (top) and angular rates (bottom) of system components</td>
<td></td>
</tr>
<tr>
<td>7.4</td>
<td>Zoomed-in nominal relative attitude (top) and angular rate (bottom) of system components</td>
<td></td>
</tr>
<tr>
<td>7.5</td>
<td>Nominal relative position (top) and velocity (bottom) of system components</td>
<td></td>
</tr>
<tr>
<td>7.6</td>
<td>Zoomed-in nominal relative position (top) and velocity (bottom) of system components</td>
<td></td>
</tr>
<tr>
<td>7.7</td>
<td>Nominal actuator torque</td>
<td></td>
</tr>
<tr>
<td>7.8</td>
<td>SC attitude (top) and angular rate (bottom) 3σ dispersion hair plots</td>
<td></td>
</tr>
<tr>
<td>7.9</td>
<td>RM attitude (top) and angular rate (bottom) 3σ dispersion hair plots</td>
<td></td>
</tr>
<tr>
<td>7.10</td>
<td>SM attitude (top) and angular rate (bottom) 3σ dispersion hair plot</td>
<td></td>
</tr>
<tr>
<td>7.11</td>
<td>RM position (top) and velocity (bottom) 3σ dispersion hair plot</td>
<td></td>
</tr>
<tr>
<td>7.12</td>
<td>Translational acceleration disturbance 3σ dispersion hair plot</td>
<td></td>
</tr>
</tbody>
</table>
7.13 Rotational acceleration disturbance $3\sigma$ dispersion hair plot ........................................... 121
7.14 SM disturbance torque $3\sigma$ dispersion hair plot ................................................................. 122
7.15 Star camera bias $3\sigma$ dispersion hair plot ........................................................................ 123
7.16 $3\sigma$ RM attitude estimation error, initial (top) and steady-state (bottom) ...................... 125
7.17 $3\sigma$ Star camera bias estimation error .................................................................................. 126
7.18 $3\sigma$ Gyro 1 bias estimation error .......................................................................................... 127
7.19 $3\sigma$ Gyro 1 scale factor estimation error .............................................................................. 128
7.20 SC to RM relative position (top) and velocity (bottom) $3\sigma$ dispersion hair plots . 129
7.21 SM to BM relative position (top) and velocity (bottom) $3\sigma$ dispersion hair plots . 130
7.22 SC to RM relative attitude (top) and angular rate (bottom) $3\sigma$ dispersions plots . 131
7.23 SM to BM relative attitude (top) and angular rate (bottom) $3\sigma$ dispersion plots . 132
7.24 Actuator torque $3\sigma$ dispersion plots, initial (top) and steady-state (bottom) . . 134
7.25 Truth SM attitude autocorrelation estimates for short term (top) and long term (bottom) . ................................................................................................................ 135
7.26 Truth SM angular rate autocorrelation estimates for short term (top) and long term (bottom) . ................................................................................................................ 136
7.27 Translational acceleration disturbance autocorrelation estimates ................................. 137
7.28 Rotational acceleration disturbance autocorrelation estimates ........................................ 138
7.29 RM angular rate autocorrelation estimates for long term (top) and short term (bottom) . ................................................................................................................ 140
7.30 SM disturbance torque autocorrelation estimates ............................................................ 141
7.31 RM position autocorrelation estimate .............................................................................. 142
7.32 RM attitude autocorrelation estimate ................................................................................ 143
7.33 SM attitude (top) and angular rate (bottom) PSD ............................................................. 145
7.34 Translational acceleration disturbance PSD ................................................................. 146
7.35 Rotational acceleration disturbance PSD ........................................................................... 147
7.36 RM angular rate PSD ........................................................................................................ 148
7.37 SM disturbance torque PSD ............................................................................................ 149
7.38 Normalized Fourier transform of the Hamming window ........................................ 150
7.39 RM attitude estimation error 1σ standard deviation showing the initial uncertainty (top) and the steady-state uncertainty (bottom) ........................................ 152
7.40 Star camera bias estimation error 1σ standard deviation ........................................ 153
7.41 Gyro 1 bias estimation error 1σ standard deviation ............................................. 154
7.42 Gyro 1 scale factor estimation error 1σ standard deviation ..................................... 155
7.43 RM attitude estimation error budget showing initial uncertainty (top) and steady-state uncertainty (bottom) ................................................................. 156
7.44 SM attitude error budget showing initial uncertainty (top) and steady-state uncertainty (bottom) ................................................................. 159
7.45 SM angular rate error budget showing initial uncertainty (top) and steady-state uncertainty (bottom) ................................................................. 160
7.46 SM attitude error budget (bar plot) at $t_{image}$ ...................................................... 161
7.47 SM angular rate error budget (bar plot) at $t_{image}$ .................................................. 161
7.48 SM attitude dispersion spectral sensitivity (PSD) .................................................. 164
7.49 SM angular rate dispersion spectral sensitivity (PSD) ............................................. 165
7.50 SM attitude spectral sensitivity (bar plot) at $t_{image}$ from 8 to 13Hz ....................... 166
7.51 SM angular rate spectral sensitivity (bar plot) at $t_{image}$ from 8 to 13Hz ............... 166
**Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARSS</td>
<td>Autonomous Rotorcraft Sniper System</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>INS</td>
<td>Inertial Navigation System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Position System</td>
</tr>
<tr>
<td>VNIR</td>
<td>Visible and Near Infrared</td>
</tr>
<tr>
<td>LWIR</td>
<td>Long-Wave Infrared</td>
</tr>
<tr>
<td>EO</td>
<td>Electro-Optical</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>GN&amp;C</td>
<td>Guidance Navigation and Control</td>
</tr>
<tr>
<td>PSD</td>
<td>Power Spectral Density</td>
</tr>
<tr>
<td>ACS</td>
<td>Attitude Control System</td>
</tr>
<tr>
<td>ISS</td>
<td>International Space Station</td>
</tr>
<tr>
<td>FEA</td>
<td>Finite Element Analysis</td>
</tr>
<tr>
<td>LTI</td>
<td>Linear Time-Invariant</td>
</tr>
<tr>
<td>ADS</td>
<td>Attitude Determination System</td>
</tr>
<tr>
<td>CG</td>
<td>Center of Gravity</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current</td>
</tr>
<tr>
<td>DOF</td>
<td>Degree of Freedom</td>
</tr>
<tr>
<td>ECRV</td>
<td>Exponentially Correlated Random Variable</td>
</tr>
<tr>
<td>OLTF</td>
<td>Open-Loop Transfer Function</td>
</tr>
<tr>
<td>CLTF</td>
<td>Closed-Loop Transfer Function</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>EMF</td>
<td>Electro-Motive Force</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Squared</td>
</tr>
<tr>
<td>LinCov</td>
<td>Linear Covariance</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Recent hardware developed at Utah State University’s Space Dynamic Laboratory has shown the need to analyze and conduct rapid trade studies of the closed-loop pointing performance of gimbaled pointing systems. Among this hardware is the Autonomous Rotorcraft Sniper System (ARSS), EyePod, and the Solar Occultation for Ice Experiment (SOFIE) instrument. ARSS is a light-weight, inertially stabilized, precision pointing platform that hosts a .338 Lapua Magnum sniper rifle. The precision pointing is accomplished via a two-axis gimbal with a tactical-grade IMU coupled to a navigation grade INS/GPS [1]. EyePod is a stabilized, geo-referenced imaging system with VNIR, LWIR, and EO sensors. In addition to a two axis gimbal for coarse stabilization and slewing, the imaging sensors are also equipped with a custom jitter-reduction stage to attenuate high frequency jitter and improve image quality [2]. Finally, SOFIE is 16 channel radiometer launched into a polar orbit. The SOFIE instrument was designed with a pointing control system consisting of two-axis steering mirror and associated sensors and driving electronics. The purpose of the pointing control system was to lock onto and precisely track the top edge of the sun in the presence of spacecraft disturbance motions [3].

The systems mentioned have several aspects in common. First is the requirement of accurate pointing and low jitter.\(^1\) Second is the gimbaled pointing mechanism to enable slewing and attenuate jitter. Third is the interaction between the gimbaled pointing mechanism and the adjacent structure. Fourth and final is the coupling of the pointing accuracy and control to the estimation errors in the navigation filter and errors in the feedback sensors.

Due to the cost of such systems and their sensitivity to jitter, it is important to accurately predict the closed-loop pointing performance before hardware is built and deployed. Hence the need for a trade study involving all relevant components of these pointing systems. A comprehensive trade study of these systems must analyze:

\(^1\) Jitter is defined as angular displacement or angular rate as a function of frequency, commonly represented in a PSD plot
• The effects of the environmental conditions wherein the system will operate, including structural dynamics.

• The effects of imperfections in actuators and sensors.

• The effects of estimation errors.

• The coupling of environmental conditions, imperfect actuators and sensors, and estimation errors on the close-loop pointing performance.

• The frequency content of the pointing errors to assess the amount of jitter in a given frequency range.

• The effect of the feedback control law with actuator command filtering.

1.1 Monte Carlo Analysis

Two analysis tools that are commonly used to analyze the closed-loop performance of a GN&C system are Monte Carlo analysis and Linear Covariance analysis. Figure 1.1 describes the overall setup of a Monte Carlo analysis for a GN&C system as developed in [4]. The variables \( \mathbf{w} \), \( \Delta \mathbf{w}_j \), \( \eta \), \( \Delta \eta_j \), and \( v_k \) are white random processes that, along with the actuator commands, \( \hat{\mathbf{u}} \) and \( \Delta \hat{\mathbf{u}}_j \), drive the dynamics of the truth model. The truth models as developed in [4] account for the environmental conditions, sensors error models, and actuator error models. Output from the truth model is the true state \( \mathbf{x} \), and the simulated sensor data \( \tilde{\mathbf{z}}_k \), \( \tilde{\mathbf{y}} \), and \( \Delta \tilde{\mathbf{y}}_j \), which contain errors as specified by the truth models. The navigation algorithm processes the measurements and produces an estimate of the state \( \hat{\mathbf{x}} \), upon which the control algorithm operates and produces the actuator command, thus closing the loop. Quantities important to the GN&C analysis are \( \delta \mathbf{x} \), \( \delta \hat{\mathbf{x}} \), and \( \delta \mathbf{e} \). As illustrated in Figure 1.1 the true state dispersion, \( \delta \mathbf{x} \), is the deviation of the true state from the reference trajectory. The navigation dispersion \( \delta \hat{\mathbf{x}} \) is the deviation of the navigation state from the same reference trajectory. Finally, the true navigation error \( \delta \mathbf{e} \) is the difference between the estimated navigation state, and the true navigation state.

A Monte Carlo analysis computes the covariance of dispersion and error states by generating \( N \) samples from the distribution of each state, then estimating the covariances as

\[
D_{\text{true}} \approx \frac{1}{N-1} \sum_{i=1}^{N} \delta \mathbf{x}_i \delta \mathbf{x}_i^T
\] (1.1)
When estimating covariances using equations 1.1-1.3, one must consider the uncertainty in that estimate. This is done using confidence intervals given a chosen probability. Figure 1.2 illustrates that given a desired confidence interval of ±10% of the computed standard deviation, the number of Monte Carlo runs required for 90%, 95%, and 99% confidence, is approximately 600, 850, and 1500, respectively. In large simulations, even with the computational power of modern computers, the time required to perform the runs quickly becomes prohibitory, making large trade studies costly and/or impractical. The next section describes an alternative algorithm that produces the same statistical results as Monte Carlo but in a small fraction of the time.
1.2 Linear Covariance Analysis

Linear Covariance analysis is basically a linearized version of the nonlinear Monte Carlo analysis. The nonlinear functions describing the truth models, navigation, and control blocks of Figure 1.1 must be linearized about a reference trajectory, which is often a laborious and time-consuming process. The return on investment however is large when performing large trade studies or sensitivity analyses. Given the validity of the linearization, Linear Covariance analysis produces the same statistical information as Monte Carlo analysis, but in a single run, by analytically propagating, updating, and correcting the covariance of the dispersion and error states.

For a linear system, the Linear Covariance approach is analytical and exact. Care must be taken, however, when applying a linear analysis to a nonlinear system. Many nonlinearities are sufficiently smooth, such that in a localized region about the reference trajectory, the state dynamics can be approximated as linear. This is often true for the case of orbital dynamics, structural dynamics, and control algorithms. When this condition is not true, however, tools such as statistical linearization [5] or a conservative Gaussian distribution can be used to appropriately account for hard nonlinearities.
Recent publications have clearly shown that highly nonlinear problems can be accurately analyzed within the assumptions of Linear Covariance analysis [4, 6, 7]. The focus of these publications, however, has been on computing the covariance of the dispersion and error states of systems involving only rigid body dynamics and full state feedback control laws. In addition, covariances as computed in current Linear Covariance theory are, by definition, the integral of the power spectral density across all frequencies. Thus information regarding the frequency content of the dispersion is not retained.
Chapter 2

Thesis

The thesis of the proposed research is that the current state-of-the-art in Linear Covariance theory can be extended to enable rapid trade studies of gimbaled pointing systems. Required extensions to the theory are illustrated in red in Figure 2.1, namely:

• The combined analysis of the flexible and rigid body elements of the structure

• Controllers with internal states and associated state dynamics, actuator command filtering, and direct sensor feedback,

• Frequency domain analysis of dispersion states

The extended theory will be applied to the performance analysis of a realistic, SOFIE-like, gimbaled pointing system.
Figure 2.1: Proposed extensions of Linear Covariance theory
Chapter 3

Related Literature

The techniques required to sufficiently analyze the performance of a closed-loop pointing control system span a wide range of fields, including structural dynamics, controls, and navigation. Due to the random nature of the disturbances and errors in the system, it is also important to consider fields related to statistics and random processes. To provide a foundation for the research detailed in this dissertation, several papers from each field are summarized in the following sections.

3.1 Navigation and Pointing Performance Analysis of Gimbaled Pointing Systems

Past and modern pointing systems have relied on a variety of approaches to satisfy the pointing requirement of the various missions [8]. Low frequency disturbances (up to \( \sim 3 \text{Hz} \)) have typically been attenuated by the ACS of the spacecraft. When the frequency content of the disturbances on the spacecraft exceeds the bandwidth of the ACS, finer pointing mechanisms, such as gimbaled mirrors or faster steering mirrors are used. Important to the pointing analysis of such satellites is the inclusion of a variety of technologies and analyses including attitude sensors and actuators, avionics, structural dynamics, optics, mechanical gimbals, and others. A powerful and fundamental tool to analyze the performance of this system of systems is Monte Carlo analysis [9]. The models involved in Monte Carlo analysis are typically refined throughout the design process and used in an iterative fashion such that when new, more accurate data is available, or a new calibration procedure is developed, the Monte Carlo analysis is repeated. Thus a large amount of time is spent running Monte Carlo simulations.

Studies more focused on the gimbal subsystem have also been carried out. In [10], Kennedy and Kennedy developed a rigid-body model of a dual axis gimbal that incorporates measurement noise models for the gyro, joint angle, and tachometer measurements. They also consider the effects of external disturbances such as linear accelerations and platform motion. Linear accelerations are coupled to the dynamics of the gimbal via mass imbalance and gimbal geometry. Platform motion is coupled to the gimbal via friction and cable restraint torques. The dynamics model developed
is a coupled, nonlinear model of medium to high complexity. The paper focuses on comparing the implementation of two different control laws, with their associated available measurements. Due to the focus on comparing control laws, the model developed neglects several important error sources in the gyros and also neglects the effects of a navigation filter on the overall pointing performance. In addition, the model does not account for flexible modes of the structure, which can also adversely affect pointing performance.

Carpenter et. al. focused on calculating the pointing jitter of a flexible structure with linear and rotational base excitations [11]. They give a general description of how to characterize an actuator in terms of force and torques induced on the structure (via a waterfall plot for example). They also outline how to couple the PSD of the input forces and torques to the structural transfer function and predict the jitter PSD. The analysis however, neglects the effects of closed-loop control of a gimbaled pointing system and its associated navigation filter.

Jacobson performed an error budget analysis for placing an imaging sensor onboard the ISS [12]. Many relevant error sources were considered including position, velocity, and timing uncertainties, as well as jitter, drift, and atmospheric effects. Also a FEA model was used to define the disturbance environment that the imaging sensor would operate in. While the scope and number of error sources considered is very complete, the focus is an error budget analysis and as such neglects the dynamics of closed-loop control of the pointing mechanism and the effects of a navigation filter.

3.2 Classical vs. Modern Linear Covariance Analysis

Linear covariance analysis is one solution to avoiding the computational requirements of a Monte Carlo analysis, and producing the same statistical information. Historically, linear covariance theory has been applied to general estimation theory problems [13–15], as well as in the design and analysis of orbit determination algorithms [16–19], inertial navigation systems [20–23], and attitude determination systems [24–26]. These analysis approaches are more commonly known as consider analysis [16, 27, 28], true covariance analysis [24], or generalized covariance analysis [25, 29]. In all of these examples, the effects of closed-loop guidance and control on the overall performance of the system is not considered.

Recent developments in linear covariance theory have combined the work of Battin [30] and
Maybeck [15] with continuous feedback control and model replacement [31] (i.e., state propagation using gyro or accelerometer measurements) to produce linear covariance tools that can be applied to many different types of closed-loop GN&C problems [4]. Specific applications include autonomous rendezvous [4, 6], powered lunar descent [32], and launch vehicle ascent trajectory analysis [7]. The emphasis in these applications has been on rigid body dynamics which is not accurate enough for the analysis of gimbaled pointing systems. Also, the validity of the assumption of full-state feedback control laws implicit in equation 20 of [4] is not typical of gimbaled pointing systems. Finally, the desired metric for the cited applications has been the covariance of quantities such as arrival dispersion, dispersion of the ascent trajectory, and estimation errors in the relative position and orientation of two spacecraft. The frequency content of such quantities has not been previously considered. Given the desire to apply modern closed-loop linear covariance theory to the analysis of pointing systems, it is concluded that extensions to the theory are needed.

3.3 Frequency Domain Analysis of Random Processes

The PSD is by definition the Fourier transform of the autocorrelation function. Analytic approaches to computing the PSD of a random process assume a LTI system and stationary statistics. Given these assumptions, Maybeck [15] and Moon and Stirling [33] outline similar approaches to computing the PSD of the output of a LTI system given the PSD of the input and the transfer function of the linear system.

When the system is time varying or the statistics of the input are not stationary, the available theory is minimal. One existing approach for non-stationary systems is outlined in [34] wherein the PSD is defined as the Fourier transform of the integral the autocorrelation function across a characteristic time period. This can be viewed as the average autocorrelation function across that time period. One application of this theory is to cyclostationary processes, in which some of the assumptions regarding constant statistics are relaxed to include statistics that are periodic in time.

Another approach to analyzing non-stationary systems is mentioned in [35]. In essence the approach entails selecting a time period across which the statistics are approximately stationary, then applying the well-developed theory regarding stationary random processes. In the context of estimating the PSD of time-series data, Proakis and Manolakis state that the length of the time period selected is a balance between the time variations in the statistics, and the spectral resolution
of the computed PSD.
Chapter 4

Theoretical Development

This chapter documents the general theory behind linear covariance analysis. General expressions for the nonlinear truth models will be developed and linearized about a reference trajectory. In addition, general expressions for the true state dispersions, navigation state dispersions, and true navigation state errors will be defined. Finally the linear covariance propagation and update equations will be defined.

4.1 Notation

Before starting, it is necessary to outline the notation used throughout this chapter. The variable, \( x \), the state of an object, will be used to illustrate. When written without any extra symbols, and in plain font, \( x \), indicates the true, one dimensional state, and contains no error. When written in bold font, \( \mathbf{x} \) represents the true vector state of the object with a specified dimension. When written with a hat, \( \hat{x} \) represents the navigation estimate of the truth state. When written with a check, \( \check{x} \) represents a controller-related state. When written with a bar, \( \bar{x} \), represents the nominal state or reference state of the object. A subscript, \( k \), indicates a variable evaluated at the discrete time, \( t_k \). As an example of putting it all together the variable, \( \hat{x}_k \in \mathbb{R}^n \), indicates the \( n \times 1 \) navigation state vector, \( \mathbf{x} \), evaluated at time, \( t_k \), along the nominal trajectory. Subscripts other than the letter \( k \), are used to for description purposes only. Another example is the variable, \( \check{x} \in \mathbb{R}^\check{n} \), which represents the \( \check{n} \times 1 \) controller state vector.

4.2 Nonlinear Truth Model

The nonlinear model of the system comprises several pieces, including dynamics models, measurement models, control laws, and Kalman filter update equations. The dynamics models include nonlinear differential equations to propagate the truth state vector, navigation state vector, and controller state vector.

The vector \( \mathbf{x} \in \mathbb{R}^n \) represents the true state of the dynamic system. The elements of the
truth state vector must be sufficient to provide a complete and accurate description of the system and may include flight, sensor, or actuator dynamics, as well as shaping filters to provide colored noise inputs to various system elements. The dynamics of the truth states are described as a function of the truth states and the output of the control law \( \hat{u} \in \mathbb{R}^{n\hat{u}} \) with additive noise.

\[
\dot{x} = f(x, \hat{u}) + Bw
\]  

(4.1)

where \( w \in \mathbb{R}^{n_w} \) is the process noise in the truth state dynamics to account for small modeling uncertainties.

\[
E[w(t) = 0]
\]  

(4.2)

\[
E[w(t)w^T(t')] = S_w(t)\delta(t-t')
\]  

(4.3)

where \( E[\cdot] \) corresponds to the expected value, or mean of the quantity. Note that explicit dependencies on time can be included, but have been omitted here to simplify the notation.

The measurements available to the navigation and control system are separated into three categories. \textit{Continuous inertial measurements} (e.g. gyro measurements) are used in model replacement mode to propagate the inertial navigation state vector. \textit{Continuous controller feedback} measurements are used to provide direct input to feedback control laws (e.g. joint angle feedback, such as a resolver or encoder). Finally \textit{discrete Kalman filter measurements} are used to update the navigation filter state and state covariance in the update step of the on-board Kalman filter.

The continuous inertial measurements, \( \tilde{y} \in \mathbb{R}^{n_{\tilde{y}}} \), are expressed as a function of the truth states with additive noise \( \eta \in \mathbb{R}^{n_{\eta}} \)

\[
\tilde{y} = c(x) + \eta
\]  

(4.4)

where

\[
E[\eta(t)] = 0, \quad E[\eta(t)\eta^T(t')] = S_\eta\delta(t-t')
\]  

(4.5)

The continuous controller feedback measurements, \( \tilde{y}_c \in \mathbb{R}^{n_{\tilde{y}_c}} \), are defined similarly as a function of the truth state with additive noise

\[
\tilde{y}_c = q(x) + \eta_c
\]  

(4.6)
where $\eta_c \in \mathbb{R}^{n_{yc}}$ is the associated noise vector with the following properties.

\[
E[\eta_c(t)] = 0 \quad E[\eta_c(t)\eta_c^T(t')] = S_{\eta_c}(t)\delta(t - t') \tag{4.7}
\]

Finally the discrete Kalman filter measurements $\tilde{z}_k \in \mathbb{R}^{n_z}$ are expressed as a function of truth states with additive noise

\[
\tilde{z}_k = h(x_k) + \nu_k \tag{4.8}
\]

where $\nu_k \in \mathbb{R}^{n_z}$ is the discrete measurement noise with the following properties

\[
E[\nu_k] = 0, \quad E[\nu_k\nu_k^T] = R_{\nu}\delta_{kk'} \tag{4.9}
\]

This completes the truth models for the dynamics, sensors, and actuators.

Next, the navigation algorithms and guidance and control algorithms are defined. Note that all elements of the navigation-related algorithms are decorated with a ‘$\hat{\cdot}$’. The navigation state vector, $\hat{x} \in \mathbb{R}^n$, is propagated and updated according to

\[
\dot{\hat{x}} = \hat{f} (\hat{x}, \tilde{y}) \tag{4.10}
\]

\[
\hat{x}_k^+ = \hat{x}_k^- + \hat{K}_k \left[ \tilde{z}_k - \hat{\tilde{z}}_k \right] \tag{4.11}
\]

where $\hat{\tilde{z}}_k \in \mathbb{R}^{n_z}$ is the estimated value of the measurement,

\[
\hat{\tilde{z}}_k = \hat{h}(\hat{x}_k) \tag{4.12}
\]

The Kalman gain is determined by

\[
\hat{K}_k = \hat{P}_k^- \hat{H}_k^T \left( \hat{H}_k \hat{P}_k^- \hat{H}_k^T + \hat{R}_{\nu} \right)^{-1} \tag{4.13}
\]

where the filter state covariance matrix, $\hat{P}_k$, is propagated and updated according to

\[
\hat{P} = \hat{F}\hat{P} + \hat{P}\hat{F}^T + \hat{B}Q\hat{B}^T \tag{4.14}
\]
\[ \dot{P}_k^+ = \left( I - \hat{K}_k \hat{H}_k \right) \dot{P}_k^- \left( I - \hat{K}_k \hat{H}_k \right)^T + \hat{K}_k \hat{R}_\nu \hat{K}_k^T \] (4.15)

Note that \( \hat{f} \) and \( \hat{h}, \hat{F}, \hat{H}, \hat{Q}, \) and \( \hat{R}_\nu \) are all determined by the navigation design model.

The true values of the navigation state vector can be derived from the true state vector via the mapping
\[ \mathbf{x}_n = \mathbf{m}(\mathbf{x}) \] (4.16)
where \( \mathbf{x}_n \in \mathbb{R}^{\hat{n}} \) are the true value of the navigation states.

The dynamics of the controller state \( \mathbf{x} \in \mathbb{R}^\hat{n} \) are propagated using the current controller state, navigation state, and continuous controller feedback measurements. Note that the controller dynamics and states are decorated with a ’
\[ \dot{\mathbf{x}} = \hat{\mathbf{f}}(\mathbf{x}, \hat{\mathbf{x}}, \hat{\mathbf{y}}_c) \] (4.17)

Finally, coupling between the three sets of states \( \mathbf{x}, \hat{\mathbf{x}}, \check{\mathbf{x}} \) is accomplished via the output of the control law which utilizes the navigation states, controller states and the controller feedback measurements to produce the actuator command.
\[ \hat{\mathbf{u}} = \hat{\mathbf{g}}(\check{\mathbf{x}}, \hat{\mathbf{x}}, \hat{\mathbf{y}}_c) \] (4.18)

In many cases it is also important to analyze the true response of the actuator, as applied to the system. This is expressed as a function of the commanded actuator value and the truth states
\[ \mathbf{u} = \mathbf{g}(\hat{\mathbf{u}}, \mathbf{x}) \] (4.19)

### 4.3 Mean Reference Trajectory

The mean reference trajectory is used in the linearization of the nonlinear truth model. It is defined as the trajectory of all quantities of interest when all error sources are removed from the nonlinear Monte Carlo simulation. It is worth noting that even though all error sources are zeroed out, there will still be transients/tracking errors due to the controller. Alternatively, the nominal trajectory can be defined as the mean trajectory of the Monte Carlo simulation.
With the definition for the reference trajectory given above, the following notation is adopted:

\[ \overline{x} = \text{nominal truth state} \quad (4.20) \]

\[ \overline{u} = \text{nominal actuator command} \quad (4.21) \]

\[ \overline{y} = \text{nominal continuous inertial measurements} \quad (4.22) \]

\[ \overline{y}_c = \text{nominal continuous controller feedback measurements} \quad (4.23) \]

\[ \overline{x}_n = \text{nominal true navigation state} \quad (4.24) \]

\[ \hat{\overline{x}} = \text{nominal navigation states as computed by the flight computer} \quad (4.25) \]

\[ \check{\overline{x}} = \text{nominal controller states} \quad (4.26) \]

\[ \overline{z} = \text{discrete Kalman filter measurements} \quad (4.27) \]

By definition, the nominal navigation states are equal to the nominal true navigation states. That is

\[ \hat{\overline{x}} = \overline{x}_n \quad (4.28) \]

Along the nominal trajectory, the nominal dynamics, measurements, and control are defined as follows:

\[ \{ \dot{\overline{x}} = f(\overline{x}, \hat{\overline{u}}) + Bw \}_{\text{nominal}} \iff \hat{\overline{x}} = f(\overline{x}, \overline{u}) \quad (4.29) \]

\[ \{ \hat{\overline{y}} = c(x) + \eta \}_{\text{nominal}} \iff \overline{y} = c(\overline{x}) \quad (4.30) \]

\[ \{ \hat{\overline{y}}_c = q(x) + \eta_c \}_{\text{nominal}} \iff \overline{y}_c = q(\overline{x}) \quad (4.31) \]

\[ \{ \dot{\hat{\overline{x}}} = \hat{f}(\hat{\overline{x}}, \hat{\overline{y}}) \}_{\text{nominal}} \iff \dot{\overline{x}}_n = \hat{f}(\overline{x}_n, \overline{y}) \quad (4.32) \]

\[ \{ \dot{\check{\overline{x}}} = \check{f}(\check{\overline{x}}, \hat{\overline{y}}_c) \}_{\text{nominal}} \iff \check{\overline{x}} = \check{f}(\check{\overline{x}}, \overline{x}_n, \overline{y}_c) \quad (4.33) \]

\[ \{ x_n = m(x) \}_{\text{nominal}} \iff \overline{x}_n = m(\overline{x}) \quad (4.34) \]

\[ \{ \hat{\overline{u}} = \check{g}(\check{\overline{x}}, \hat{\overline{y}}_c) \}_{\text{nominal}} \iff \overline{u} = \check{g}(\check{\overline{x}}, \overline{x}_n, \overline{y}_c) \quad (4.35) \]

\[ \{ \bar{\overline{z}}_k = h(x_k) + \nu_k \}_{\text{nominal}} \iff \overline{z}_k = h(\overline{x}_k) \quad (4.36) \]
\[
\left\{ \hat{z}_k = \hat{h}(\hat{x}_k) \right\}_{\text{nominal}} \iff \hat{z}_k = \hat{h}(\bar{x}_{n,k}) \tag{4.37}
\]

\[
\left\{ \hat{x}^+_k = \hat{x}^-_k + \hat{K}_k [\hat{z}_k - \hat{z}_k] \right\}_{\text{nominal}} \iff \hat{x}^+_k = \hat{x}^-_k + \hat{K}_k [\bar{z}_k - \bar{z}_k] \tag{4.38}
\]

### 4.4 Linear Modeling

In this section, the general nonlinear models in section 4.2 are linearized about the mean reference trajectory. The main goal of linearization is to derive the linear dynamic equations for the true state dispersion, \(\delta \hat{x}\), the navigation state dispersion, \(\delta \hat{\hat{x}}\), and the controller state dispersion, \(\delta \hat{\hat{\hat{x}}}\) as a set of coupled linear differential equations. The linearization is accomplished via three steps: two expansions and one subtraction. The first expansion is accomplished by expressing all variables as the nominal value plus a small dispersion (e.g. \(x = \bar{x} + \delta x\)). The second step involves expanding to first order all nonlinear functions about the nominal using a Taylor series expansion. The nominal trajectories defined in equations 4.29 to 4.38 are then subtracted from the expanded terms, yielding a first-order approximation of the dispersion of the state about the nominal. Occasionally, after linearization, the resulting equation is not an explicit function of the true, navigation, or controller states dispersions. In these cases, further substitutions and/or linearization are needed to achieve the desired final form.

#### 4.4.1 Truth State Dynamics

The truth state dynamics are linearized by substituting \(x = \bar{x} + \delta x\) and \(\hat{u} = \bar{u} + \delta \hat{u}\) into equation 4.1 to yield

\[
\dot{\bar{x}} + \delta \dot{x} = f(\bar{x} + \delta x, \bar{u} + \delta \hat{u}) + Bw \tag{4.39}
\]

where the function \(f(\cdot, \cdot)\) is expanded about the nominal using a multivariate Taylor series to become

\[
\dot{\bar{x}} + \delta \dot{x} = f(\bar{x}, \bar{u}) + F_x \delta x + F_{\bar{u}} \delta \hat{u} + Bw \tag{4.40}
\]

Subtracting the nominal trajectory in equation 4.29 results in the linear differential equation for the dispersion of the truth state

\[
\delta \dot{x} = F_x \delta x + F_{\bar{u}} \delta \hat{u} + Bw \tag{4.41}
\]
where the Jacobians are defined as

\[ F_x = \frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \hat{\mathbf{u}}) \bigg|_{\mathbf{x}, \mathbf{u}} \]  
(4.42)

\[ F_{\hat{u}} = \frac{\partial}{\partial \hat{\mathbf{u}}} \mathbf{f}(\mathbf{x}, \hat{\mathbf{u}}) \bigg|_{\mathbf{x}, \mathbf{u}} \]  
(4.43)

Note that equation 4.41 still is not an explicit function of the true, navigation, and controller state dispersion. Thus it remains to expand the expression for the actuator command dispersion, \( \delta \hat{\mathbf{u}} \). This is done in similar fashion by substituting \( \ddot{x} = \ddot{x} + \delta \ddot{x}, \dot{x} = \dot{x} + \delta \dot{x}, \) and \( \ddot{y}_c = \ddot{y}_c + \delta \ddot{y}_c \) into 4.18

\[ \mathbf{u} + \delta \hat{u} = \hat{\mathbf{g}}(\ddot{x} + \delta \ddot{x}, \dot{x} + \delta \dot{x}, \ddot{y}_c + \delta \ddot{y}_c, t) \]  
(4.44)

expanding the nonlinear output of the control law to first order with a Taylor series,

\[ \mathbf{u} + \delta \hat{u} = \hat{\mathbf{g}}(\ddot{x}, \dot{x}, \ddot{y}_c, t) + \hat{G}_x \delta \ddot{x} + \hat{G}_\dot{x} \delta \dot{x} + \hat{G}_{\ddot{y}_c} \delta \ddot{y}_c \]  
(4.45)

and subtracting the nominal in equation 4.35 to yield

\[ \delta \hat{u} = \hat{G}_x \delta \ddot{x} + \hat{G}_\dot{x} \delta \dot{x} + \hat{G}_{\ddot{y}_c} \delta \ddot{y}_c \]  
(4.46)

where

\[ \hat{G}_x = \frac{\partial}{\partial \ddot{x}} \hat{\mathbf{g}}(\ddot{x}, \dot{x}, \ddot{y}_c) \bigg|_{\ddot{x}, \dot{x}, \ddot{y}_c} \]  
(4.47)

\[ \hat{G}_\dot{x} = \frac{\partial}{\partial \dot{x}} \hat{\mathbf{g}}(\ddot{x}, \dot{x}, \ddot{y}_c) \bigg|_{\ddot{x}, \dot{x}, \ddot{y}_c} \]  
(4.48)

\[ \hat{G}_{\ddot{y}_c} = \frac{\partial}{\partial \ddot{y}_c} \hat{\mathbf{g}}(\ddot{x}, \dot{x}, \ddot{y}_c) \bigg|_{\ddot{x}, \dot{x}, \ddot{y}_c} \]  
(4.49)

Once again, the controller output dispersion is not an explicit function of the desired state dispersions. Therefore, equation 4.46 must be further expanded to eliminate the dependence on the continuous feedback controller measurement dispersion \( \delta \ddot{y}_c \). Following the same procedure as
Before, $\tilde{y}_c = \tilde{y}_c + \delta \tilde{y}_c$ and $x = \bar{x} + \delta x$ are substituted into equation 4.46

$$\bar{y}_c + \delta \tilde{y}_c = q (\bar{x} + \delta x) + \eta_c \quad (4.50)$$

and expanded in a Taylor series to become

$$\bar{y}_c + \delta \tilde{y}_c = q (x, t) + Q_x \delta x + \eta_c \quad (4.51)$$

The nominal controller feedback in equation 4.31 is then subtracted to yield

$$\delta \tilde{y}_c = Q_x \delta x + \eta_c \quad (4.52)$$

where

$$Q_x = \frac{\partial}{\partial x} q (x) \bigg|_{\bar{x}} \quad (4.53)$$

Substituting equations 4.52 into 4.46 yields

$$\delta \hat{u} = \hat{G}_x \delta \bar{x} + \hat{G}_{\tilde{y}_c} Q_x \delta x + \hat{G}_{\tilde{y}_c} \eta_c \quad (4.54)$$

Finally substituting equation 4.54 into equation 4.41 yields the final form for the linearized truth state dispersion dynamics.

$$\delta \dot{x} = \left( F_x + F_u \hat{G}_{\tilde{y}_c} Q_x \right) \delta x + F_u \hat{G}_x \delta \bar{x} + F_u \hat{G}_{\tilde{y}_c} \delta \bar{x} + F_u \hat{G}_{\tilde{y}_c} \eta_c + B w \quad (4.55)$$

### 4.4.2 Navigation State Dynamics

The navigation state propagation is linearized in like manner. Substituting $\dot{x} = \bar{x}_n + \delta \bar{x}$, and $\tilde{y} = \bar{y} + \delta \tilde{y}$ into equation 4.10 produces

$$\dot{\bar{x}}_n + \delta \dot{\bar{x}} = \hat{f} (\bar{x}_n + \delta \bar{x}, \bar{y} + \delta \tilde{y}) \quad (4.56)$$
And expanding to first order using a Taylor series results in

\[ \dot{x}_n + \delta \dot{x} = \hat{f}(\mathbf{x}_n, y) + \hat{F}_x \delta \dot{x} + \hat{F}_y \delta \tilde{y} \]  

(4.57)

The nominal in equation 4.32 is subtracted to become

\[ \delta \dot{x} = \hat{F}_x \delta \dot{x} + \hat{F}_y \delta \tilde{y} \]  

(4.58)

where

\[ \hat{F}_x = \frac{\partial}{\partial \dot{x}} \hat{f}(\dot{x}, \tilde{y}) \bigg|_{\mathbf{x}_n, \tilde{y}} \]  

(4.59)

\[ \hat{F}_y = \frac{\partial}{\partial \tilde{y}} \hat{f}(\dot{x}, \tilde{y}) \bigg|_{\mathbf{x}_n, \tilde{y}} \]  

(4.60)

The continuous inertial measurement dispersion is then derived by substituting \( x = \mathbf{x} + \delta \dot{x} \) and \( \tilde{y} = \tilde{y} + \delta \tilde{y} \) into equation 4.58

\[ \tilde{y} + \delta \tilde{y} = c(\mathbf{x} + \delta \dot{x}) + \eta \]  

(4.61)

and expanding in a Taylor series to yield

\[ \tilde{y} + \delta \tilde{y} = c(\mathbf{x}) + C_x \delta \dot{x} + \eta \]  

(4.62)

The nominal inertial measurement in equation 4.32 is subtracted to give

\[ \delta \tilde{y} = C_x \delta \dot{x} + \eta \]  

(4.63)

where

\[ C_x = \frac{\partial}{\partial \dot{x}} c(\dot{x}) \bigg|_{\tilde{x}} \]  

(4.64)
Finally, substituting 4.63 into 4.58 yields the final form of the navigation state dispersion dynamics.

\[ \delta \dot{x} = \hat{F}_x \delta \dot{x} + \hat{F}_y C_x \delta x + \hat{F}_y \eta \]  

(4.65)

### 4.4.3 Controller State Dynamics

The controller state differential equation is also linearized following the same three steps. Expansion about the nominals yields

\[ \dot{\bar{x}} + \delta \dot{x} = \bar{f} (\bar{x}, \bar{x}_n, \bar{y}_c, t) \]  

(4.66)

which after Taylor series expansion becomes

\[ \dot{\bar{x}} + \delta \dot{x} = \bar{f} (\bar{x}, \bar{x}_n, \bar{y}_c, t) + \bar{F}_x \delta \dot{x} + \bar{F}_y \delta y_c \]  

(4.67)

from which the nominal in 4.33 is subtracted to yield

\[ \delta \dot{x} = \bar{F}_x \delta \dot{x} + \bar{F}_y \delta y_c \]  

(4.68)

where

\[ \bar{F}_x = \frac{\partial}{\partial \bar{x}} \bar{f} (\bar{x}, \bar{x}_n, \bar{y}_c) \]  

(4.69)

\[ \bar{F}_x = \frac{\partial}{\partial \bar{x}} \bar{f} (\bar{x}, \bar{x}_n, \bar{y}_c) \]  

(4.70)

\[ \bar{F}_y = \frac{\partial}{\partial \bar{y}_c} \bar{f} (\bar{x}, \bar{x}_n, \bar{y}_c) \]  

(4.71)

Now substituting equation 4.52 into 4.68 yields the final form for the linearized controller state dispersions

\[ \delta \dot{x} = \bar{F}_x \delta \dot{x} + \bar{F}_x \delta \dot{x} + \bar{F}_y Q_x \delta x + \bar{F}_y \eta_c \]  

(4.72)
4.4.4 True Actuator Response

Expansion of the true actuator value (equation 4.19) about the nominal state and actuator command yields

$$u + \delta u = g (\hat{u} + \delta \hat{u}, x + \delta x)$$

(4.73)

Expansion in a Taylor series yields

$$u + \delta u = g (\hat{u}, x) + G_{\hat{u}} \delta \hat{u} + G_{x} \delta x$$

(4.74)

from which the nominal can be subtracted to yield

$$\delta u = G_{\hat{u}} \delta \hat{u} + G_{x} \delta x$$

(4.75)

Equation 4.54 can then be substituted to find the actuator response dispersion as an explicit function of the truth, navigation, and controller dispersion states

$$\delta u = G_{\hat{u}} \hat{G}_{\hat{x}} \delta \hat{x} + G_{\hat{u}} \hat{G}_{x} \delta \hat{x} + \left( G_{\hat{u}} \hat{G}_{\hat{y}, c} Q_{x} + G_{x} \right) \delta x + G_{\hat{u}} \hat{G}_{\hat{y}, c} \eta_c$$

(4.76)

4.4.5 True, Navigation, and Controller State Dispersion Update

It is observed that the truth state and controller state vectors remain unchanged after a navigation state update. Thus we have

$$\delta x^+ = \delta x^-$$

(4.77)

$$\delta \hat{x}^+ = \delta \hat{x}^-$$

(4.78)

The navigation state dispersion, however, is dependent on the navigation filter state updates. Substituting $\hat{\delta} = \hat{x} + \delta \hat{x}$ and $x = \hat{x} + \delta \hat{x}$ into equation 4.11 and noting that the update does not alter the nominal navigation state (i.e. $x_{n,k}^+ = x_{n,k}^- = x_{n,k}$) produces

$$x_{n,k} + \delta \hat{x}^+_k = x_{n,k} + \delta \hat{x}^-_k + K_k \left[ h (x_k + \delta x_k) + \nu_k - \hat{h} (x_{n,k} + \delta \hat{x}^-_k) \right]$$

(4.79)
Expanding the nonlinear functions using a Taylor series yields

\[ \mathbf{x}_{n,k} + \delta \mathbf{x}^+_k = \mathbf{x}_{n,k} + \delta \mathbf{x}^-_k + K_k \left[ h(\mathbf{x}_k) + H_k \delta \mathbf{x}_k + \nu_k - \hat{h}(\mathbf{x}_n) - \hat{H}_k \delta \hat{x}_k \right] \]  

(4.80)

Subtracting the nominal values in equation 4.38 yields the final form

\[ \delta \mathbf{x}^+_k = \left[ I - K_k \hat{H}_k \right] \delta \mathbf{x}^-_k + K_k H_k \delta \mathbf{x}^- + K_k \nu_k \]  

(4.81)

where

\[ H_k = \frac{\partial}{\partial \mathbf{x}} h(\mathbf{x}) \bigg|_{\mathbf{x}_n} \]  

(4.82)

\[ \hat{H}_k = \frac{\partial}{\partial \mathbf{x}} \hat{h}(\mathbf{x}) \bigg|_{\mathbf{x}_n} \]  

(4.83)

### 4.4.6 True Navigation Errors

Recall that the true navigation errors, \( \delta \mathbf{e} \), are defined as

\[ \mathbf{e} = \dot{\mathbf{x}} - \mathbf{m}(\mathbf{x}) \]  

(4.84)

This can be expanded about the nominal to yield

\[ \mathbf{e} + \delta \mathbf{e} = \mathbf{x}_n + \delta \dot{\mathbf{x}} - \mathbf{m}(\mathbf{x} + \delta \mathbf{x}) \]  

(4.85)

Expansion in a Taylor series produces

\[ \mathbf{e} + \delta \mathbf{e} = \mathbf{x}_n + \delta \dot{\mathbf{x}} - \mathbf{m}(\mathbf{x}) - M_{\mathbf{x}} \delta \mathbf{x} \]  

(4.86)

Subtracting the nominal in equation 4.34 yields the final form for the linearized true navigation errors

\[ \delta \mathbf{e} = \delta \dot{\mathbf{x}} - M_{\mathbf{x}} \delta \mathbf{x} \]  

(4.87)
where
\[ M_x = \frac{\partial}{\partial x} m(x) \bigg|_x \quad (4.88) \]

### 4.4.7 Summary of Linearized Equations

Equations 4.55, 4.65, 4.72, repeated here, form a complete set of linearized equations of motion.

\[
\delta \dot{x} = \left( F_x + F_u \hat{G}_{\delta_y} Q_x \right) \delta x + F_u \hat{G}_{\delta_x} \delta \dot{x} + F_u \hat{G}_{\delta_y} \hat{G}_{\delta_x} \eta_c + Bw \quad (4.89)
\]

\[
\hat{\delta} \dot{x} = \hat{F}_x \delta \dot{x} + \hat{F}_y C_x \delta x + \hat{F}_y \eta \quad (4.90)
\]

\[
\check{\delta} \dot{x} = \check{F}_x \delta \dot{x} + \check{F}_x \delta x + \check{F}_y Q_x \delta x + \check{F}_y \eta_c \quad (4.91)
\]

with equations 4.77, 4.78, and 4.81 used for state updates.

\[
\delta x_k^+ = \delta x_k^- \quad (4.92)
\]

\[
\delta \check{x}_k^+ = \delta \check{x}_k^- \quad (4.93)
\]

\[
\delta \hat{x}_k^+ = \left[ I - K_k \hat{H}_k \right] \delta \hat{x}_k^- + K_k H_k \delta x_k^- + K_k \nu_k \quad (4.94)
\]

where the Kalman gain is computed along the reference trajectory using equations 4.13 through 4.14.

### 4.5 Covariance Models

Using the results of the previous section, an augmented state vector \( \mathbf{X} \) can now be defined as

\[
\mathbf{X} = \begin{bmatrix} \delta x \\ \delta \check{x} \\ \delta \hat{x} \end{bmatrix} \in \mathbb{R}^{n + \delta + \eta_c} \quad (4.95)
\]
where the dynamics and update equations can be combined to form a single set of propagation and update equations

\[
\dot{\mathbf{X}} = \mathbf{F} \mathbf{X} + \mathbf{R} \eta_c + \mathbf{G} \eta + \mathbf{W} \mathbf{w}
\]  
\[
\mathbf{X}_k^+ = \mathbf{A}_k \mathbf{X}_k^- + \mathbf{B}_k \nu_k
\]

where

\[
\mathbf{F} = \begin{bmatrix}
F_x + F_u \mathbf{G} \mathbf{y}_c \mathbf{Q}_x & F_u \mathbf{G} \dot{\mathbf{x}} & F_u \mathbf{G} \mathbf{x} \\
\dot{\mathbf{F}} \mathbf{C}_x & \dot{\mathbf{F}} \dot{\mathbf{x}} & 0_{\mathbf{n} \times \mathbf{n}} \\
\dot{\mathbf{F}} \mathbf{y}_c \mathbf{Q}_x & \dot{\mathbf{F}} \dot{\mathbf{x}} & \mathbf{F} \mathbf{x}
\end{bmatrix}
\]

\[
\mathbf{W} = \begin{bmatrix}
\mathbf{B} \\
0_{\mathbf{n} \times \mathbf{n}_w} \\
0_{\mathbf{n} \times \mathbf{n}_w}
\end{bmatrix}
\]

\[
\mathbf{G} = \begin{bmatrix}
0_{\mathbf{n} \times \mathbf{n}_\delta} \\
\dot{\mathbf{F}} \mathbf{y}_c \\
0_{\mathbf{n} \times \mathbf{n}_\delta}
\end{bmatrix}
\]

\[
\mathbf{R} = \begin{bmatrix}
F_u \mathbf{G} \mathbf{y}_c \\
0_{\mathbf{n} \times \mathbf{n}_{yc}} \\
\mathbf{F} \mathbf{y}_c
\end{bmatrix}
\]

\[
\mathbf{A}_k = \begin{bmatrix}
\mathbf{I}_{\mathbf{n} \times \mathbf{n}} & 0_{\mathbf{n} \times \mathbf{n}} & 0_{\mathbf{n} \times \mathbf{n}} \\
K_k H_k & \mathbf{I}_{\mathbf{n} \times \mathbf{n}} - K_k \dot{\mathbf{H}}_k & 0_{\mathbf{n} \times \mathbf{n}} \\
0_{\mathbf{n} \times \mathbf{n}} & 0_{\mathbf{n} \times \mathbf{n}} & \mathbf{I}_{\mathbf{n} \times \mathbf{n}}
\end{bmatrix}
\]

\[
\mathbf{B}_k = \begin{bmatrix}
0_{\mathbf{n} \times \mathbf{n}_z} \\
K_k \\
0_{\mathbf{n} \times \mathbf{n}_z}
\end{bmatrix}
\]

### 4.5.1 Covariance Equations

The covariance equations for the augmented linear system in equations 4.96 and 4.97 can now be developed. This is done by first showing that the mean of the dispersion state are zero. Given the definition of the nominal trajectories for the truth, navigation, and controller states, the mean of the dispersions can be written as

\[
E[\delta \mathbf{x}] = E[\mathbf{x} - \mathbf{x}]
\]

\[
E[\delta \dot{\mathbf{x}}] = E[\dot{\mathbf{x}} - \dot{\mathbf{x}}]
\]
\[ E[\delta \hat{x}] = E[\hat{x} - \bar{x}] \quad (4.103) \]

where it is noted that the nominal state is not a random quantity and can therefore be taken out of the expectation operator to yield

\[ E[\delta x] = E[x] - \bar{x} \quad (4.104) \]
\[ E[\delta \hat{x}] = E[\hat{x}] - \bar{x} \quad (4.105) \]
\[ E[\delta \tilde{x}] = E[\tilde{x}] - \bar{x} \quad (4.106) \]

Note that for the models developed, the mean of the truth, navigation, and controller states is defined to be equal to the nominal. Substituting \( E[x] = \bar{x}, \ E[\hat{x}] = \hat{x}, \) and \( E[\tilde{x}] = \tilde{x} \) into equations 4.104 through 4.106 yields the desired result

\[ E[\delta x] = \bar{x} - \bar{x} = 0 \quad (4.107) \]
\[ E[\delta \hat{x}] = \hat{x} - \hat{x} = 0 \quad (4.108) \]
\[ E[\delta \tilde{x}] = \tilde{x} - \tilde{x} = 0 \quad (4.109) \]

This implies that the mean of the augmented system is zero for all time

\[ E[X] = 0 \quad (4.110) \]

and the covariance of the augmented system can be calculated directly from the augmented state

\[ C_A = E[X(t)X^T(t)] \quad (4.111) \]

with the following propagation equation where it is assumed that \( \eta_c, \eta, \) and \( w \) are mutually uncorrelated.

\[ \dot{C}_A = FC_A + C_A \mathcal{F} + \mathcal{R}S_{\eta_c} \mathcal{R}^T + \mathcal{G}S_{\eta} \mathcal{G}^T + \mathcal{W}S_w \mathcal{W}^T \quad (4.112) \]

The augmented system covariance update equation can be derived as

\[ C_A(t^+_k) = E[X_k^+X_k^{+T}] \quad (4.113) \]
which, with the assumption that the current augmented state is not correlated with the current
discrete measurement noise becomes

\[ C_A(t^+_k) = A_k C_A(t^-_k) A^T_k + B_k R_k (t_k) B^T_k \] (4.114)

### 4.5.2 Performance Evaluation

The overall closed-loop performance of the GN&C system is often characterized by the
covariance of the truth state dispersions, navigation dispersions, controller state dispersions, and
the true filter errors. The covariance of the truth state dispersions are given by

\[
D_{true} = E[\delta \mathbf{x}(t) \delta \mathbf{x}^T(t)] = \begin{pmatrix}
I_{n \times n} & 0_{n \times \hat{n}} & 0_{n \times \hat{n}} \\
0_{\hat{n} \times n} & I_{\hat{n} \times \hat{n}} & 0_{\hat{n} \times \hat{n}} \\
0_{\hat{n} \times n} & 0_{\hat{n} \times \hat{n}} & 0_{\hat{n} \times \hat{n}}
\end{pmatrix} C_A \begin{pmatrix}
I_{n \times n} & 0_{n \times \hat{n}} & 0_{n \times \hat{n}} \\
0_{\hat{n} \times n} & I_{\hat{n} \times \hat{n}} & 0_{\hat{n} \times \hat{n}} \\
0_{\hat{n} \times n} & 0_{\hat{n} \times \hat{n}} & 0_{\hat{n} \times \hat{n}}
\end{pmatrix}^T
\] (4.115)

The truth state dispersion covariance characterizes the effects of the navigation error, guidance and
control errors, and environmental uncertainties.

The covariance of the navigation state dispersion is also calculated from the augmented
covariance matrix

\[
D_{nav} = E[\delta \mathbf{x}(t) \delta \mathbf{x}^T(t)] = \begin{pmatrix}
0_{\hat{n} \times n} & I_{\hat{n} \times \hat{n}} & 0_{\hat{n} \times \hat{n}} \\
0_{\hat{n} \times n} & I_{\hat{n} \times \hat{n}} & 0_{\hat{n} \times \hat{n}}
\end{pmatrix} C_A \begin{pmatrix}
0_{\hat{n} \times n} & I_{\hat{n} \times \hat{n}} & 0_{\hat{n} \times \hat{n}} \\
0_{\hat{n} \times n} & I_{\hat{n} \times \hat{n}} & 0_{\hat{n} \times \hat{n}}
\end{pmatrix}^T
\] (4.116)

The covariance of the true navigation error is given by.

\[
P_{true} = E[\{\delta \mathbf{x}(t) - \mathbf{M}_x \delta \mathbf{x}(t)\} \{\delta \mathbf{x}(t) - \mathbf{M}_x \delta \mathbf{x}(t)(t)\}^T]
\] (4.117)

which, can be written as

\[
P_{true} = \begin{pmatrix}
-M_x & 0_{\hat{n} \times \hat{n}} \\
0_{\hat{n} \times n} & 0_{\hat{n} \times \hat{n}}
\end{pmatrix} C_A \begin{pmatrix}
-M_x & 0_{\hat{n} \times \hat{n}} \\
0_{\hat{n} \times n} & 0_{\hat{n} \times \hat{n}}
\end{pmatrix}^T
\] (4.118)

Finally, the covariance of the controller state dispersions is

\[
D_{cont} = E[\delta \mathbf{x}(t) \delta \mathbf{x}^T(t)] = \begin{pmatrix}
0_{\hat{n} \times n} & 0_{\hat{n} \times \hat{n}} & I_{\hat{n} \times \hat{n}} \\
0_{\hat{n} \times n} & 0_{\hat{n} \times \hat{n}} & I_{\hat{n} \times \hat{n}}
\end{pmatrix} C_A \begin{pmatrix}
0_{\hat{n} \times n} & 0_{\hat{n} \times \hat{n}} & I_{\hat{n} \times \hat{n}} \\
0_{\hat{n} \times n} & 0_{\hat{n} \times \hat{n}} & I_{\hat{n} \times \hat{n}}
\end{pmatrix}^T
\] (4.119)
4.5.3 Covariance Initialization

An important part of the LinCov simulation is the initialization of the augmented covariance matrix and the navigation state covariance matrix. While the details of the initialization are very problem specific, some general considerations are presented here. The initial covariance of the augmented system is calculated using the following equations.

\[ C_A(t_0) = E \left[ X(t_0) X(t_0)^T \right] \tag{4.120} \]

\[ = E \left\{ \begin{bmatrix} \delta x_0 \\ \delta \hat{x}_0 \\ \delta \tilde{x}_0 \end{bmatrix} \begin{bmatrix} \delta x_0^T & \delta \hat{x}_0^T & \delta \tilde{x}_0^T \end{bmatrix} \right\} \]

\[ = \begin{bmatrix} E[\delta x_0 \delta x_0^T] & E[\delta x_0 \delta \hat{x}_0^T] & E[\delta x_0 \delta \tilde{x}_0^T] \\ E[\delta \hat{x}_0 \delta x_0^T] & E[\delta \hat{x}_0 \delta \hat{x}_0^T] & E[\delta \hat{x}_0 \delta \tilde{x}_0^T] \\ E[\delta \tilde{x}_0 \delta x_0^T] & E[\delta \tilde{x}_0 \delta \hat{x}_0^T] & E[\delta \tilde{x}_0 \delta \tilde{x}_0^T] \end{bmatrix} \tag{4.121} \]

Note that in general, there exist many important correlations between the truth, navigation, and controller state dispersion, which must be considered. For the case of this research, however, the initial value of the truth, navigation, and controller state dispersions are not correlated and equation 4.121 reduces to the block diagonal matrix

\[ C_A(t_0) = \begin{bmatrix} E[\delta x_0 \delta x_0^T] & 0_{n \times \hat{n}} & 0_{n \times \tilde{n}} \\ 0_{\hat{n} \times n} & E[\delta \hat{x}_0 \delta \hat{x}_0^T] & 0_{\hat{n} \times \tilde{n}} \\ 0_{\tilde{n} \times n} & 0_{\tilde{n} \times \hat{n}} & E[\delta \tilde{x}_0 \delta \tilde{x}_0^T] \end{bmatrix} \tag{4.122} \]

The initial navigation covariance matrix is derived in similar manner as

\[ P_0 = E \left[ \hat{x}_0 \hat{x}_0^T \right] \tag{4.123} \]
where in general the correlation between navigation states must be considered. The details specific to the initialization of these matrices for the problem analyzed in this research are discussed in detail in section 6.4.

4.6 Frequency Domain Analysis

As stated in Chapter 2, a key element of this research is to extend the capabilities of LinCov analysis to produce estimates of the frequency content of the dispersions, and to compare these estimates to classical Monte Carlo approaches. Many methods already exist in the Monte Carlo realm, ranging from non-parametric methods that operate on the raw data, to parametric methods that fit the data to a model. The pros and cons of these different methods are discussed in chapter 14 of [35].

For the purpose of this research, a classical non-parametric was chosen to determine the frequency content of the data generated by the Monte Carlo simulation. Two LinCov-based methods were developed to compare to the conventional Monte Carlo approach. The first LinCov method is a non-parametric method based on the autocorrelation function. The second method is based on the transfer function of the augmented system. The details of all methods are discussed in the following subsections. An illustration of each approach based on a simple first-order, time-varying system is presented.

4.6.1 Monte Carlo Approach

The classical approach for estimating the power spectral density (PSD) of a random process consist of several non-parametric approaches discussed in section 14.2 of [35]. The method for handling non-stationary statistics is to select a time window across which the statistics can be assumed stationary. Since the frequency resolution of the PSD estimate is proportional to the length of the data set, the analyst must balance the desire for a long data set to ensure good frequency resolution and a short data set to ensure validity of the stationarity assumption. The non-parametric method chosen for this research is the Welch method.

The Welch method [35, 36] consists of partitioning a single data set of length $N$ into $K$ possibly overlapping sections each of length $L$. A window function, $W(j)$, is then applied to each
section and the Fourier transform computed for each of the $K$ sections.

$$A_k(n) = \frac{1}{L} \sum_{j=0}^{L-1} x_k(j) W(j) e^{-2kijn/L}, \ k = 1, 2, \cdots K$$  \hspace{1cm} (4.124)

where $i = \sqrt{-1}$.

The modified periodogram for each segment is obtained by computing the magnitude squared of the Fourier transform

$$I_k(f_n) = \frac{L}{U} |A_k(n)|^2$$  \hspace{1cm} (4.125)

where $f_n$ is the normalized frequency

$$f_n = \frac{n}{L} \hspace{1cm} n = 0, \cdots, L/2$$  \hspace{1cm} (4.126)

and $U$ is function of the window function $W$

$$U = \frac{1}{L} \sum_{j=0}^{L-1} W^2(j)$$  \hspace{1cm} (4.127)

Finally, the PSD is determined by taking the average of the $K$ modified periodograms

$$\bar{\Psi}_{xx}(f_n) = \frac{1}{K} \sum_{k=1}^{K} I_k(f_n)$$  \hspace{1cm} (4.128)

The method as outlined in [35, 36] is tailored to the case where only one realization of a random process is available. In the case of a Monte Carlo simulation however, many realizations of the random process are available. The Welch method is therefore modified slightly such that the $K$ segments do not overlap and are obtained from each Monte Carlo run.

In addition to estimating the PSD from the raw data, the autocorrelation function will also be estimated from the raw Monte Carlo data using the following equation

$$\Psi_{xx}(m) = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-m-1} x_{n+m} x_n & m \geq 0 \\ \Psi_{xx}(-m) & m < 0 \end{cases}$$  \hspace{1cm} (4.129)

The autocorrelation function based on the Monte Carlo simulation will be compared to the autocorrelation function based on the LinCov simulation.
4.6.2 LinCov Autocorrelation Approach

In the Monte Carlo simulation environment, one has access to several realizations of the random process. As discussed previously, these realizations can be windowed, Fourier transformed, and averaged to produce an estimate of the PSD. In the LinCov simulation environment, however, only the statistics of the random process are available. Hence the Welch method cannot be used. The purpose of this section is to develop a method for estimating the PSD using the information available from the LinCov simulation.

Given the following continuous-time linear system

$$\dot{x}(t) = F(t)x(t) + w(t)$$

where

$$E[w(t)] = 0$$

$$E[w(t)w(t')] = Q(t)\delta(t - t')$$

An equivalent discrete model can be formed

$$x(t_{i+1}) = \Phi(t_{i+1}, t_i)x(t_i) + w_d(t_i)$$

where

$$\Phi(t_{i+1}, t_i) = \text{the state transition matrix from } t_i \text{ to } t_{i+1}$$

$$E[w_d(t_i)] = 0$$

$$E[w_d(t_i)w_d(t_i)] = Q_d(t_i) = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1}, t_i)Q(t)\Phi(t_{i+1}, t_i)^T \, d\tau$$

$$E[w_d(t_i)w_d(t_j)] = 0, \quad i \neq j$$

To simplify the following the derivation, a nominal time $t_0$ will be considered with a time in the future $t_n$ and a time in the past $t_m$ as illustrated in Figure 4.1.
The discrete state and covariance forward propagation equations relative to the nominal time \( t_0 \) are

\[
\mathbf{m}_x (t_n) = \Phi (t_n, t_0) \mathbf{m}_x (t_0)
\]

\[
P_{xx} (t_n) = \Phi (t_n, t_0) P_{xx} (t_0) \Phi (t_n, t_0)^T + Q_d (t_0)
\]

Recall that the definition for the autocorrelation kernel is

\[
\Psi_{xx} (t_p, t_q) = E \left[ \mathbf{x} (t_p) \mathbf{x} (t_q)^T \right]
\]

which is typically plotted as a function of \( t_p - t_q \) such that autocorrelation for a positive lag implies \( t_p > t_q \), and the autocorrelation for a negative lag corresponds to \( t_p < t_q \). The covariance kernel is related to the autocorrelation kernel via

\[
\Psi_{xx} (t_p, t_q) = P_{xx} (t_p, t_q) + \mathbf{m}_x (t_p) \mathbf{m}_x (t_q)^T
\]

For linear covariance analysis, however, the mean of the state is zero \((\mathbf{m}_x = 0)\). Thus equation 4.141 can be simplified such that the autocorrelation kernel is equal to the covariance kernel

\[
\Psi_{xx} (t_p, t_q) = P_{xx} (t_p, t_q)
\]

The output of a standard LinCov simulation is the covariance kernel, evaluated at zero lag

\[
P_{xx} (t_p, t_q) \text{ where } t_p = t_q
\]

or in simpler notation

\[
P_{xx} (t_p, t_p) = P_{xx} (t_p)
\]
Thus, it is desired to develop a method that utilizes the output of linear covariance analysis, equation 4.143, to produce the autocorrelation matrix $\Psi_{xx}(t_p, t_q)$.

To develop an equation for the positive lag portion of the autocorrelation matrix relative to the nominal time $t_0$, equation 4.133 can be substituted into equation 4.140 with the appropriate time variable substitutions to yield

$$\Psi_{xx}(t_n, t_0) = E \left[ \{\Phi(t_n, t_0) x(t_0) + w_d(t_0)\} x(t_0)^T \right] \quad (4.145)$$

which can be rearranged to

$$\Psi_{xx}(t_n, t_0) = E \left[ \Phi(t_n, t_0) x(t_0) x(t_0)^T \right] + E \left[ w_d(t_0) x(t_0)^T \right] \quad (4.146)$$

where the second expectation goes to zero since $x(t_0)$ is uncorrelated with $w_d(t_0)$ yielding

$$\Psi_{xx}(t_n, t_0) = E \left[ \Phi(t_n, t_0) x(t_0) x(t_0)^T \right] \quad (4.147)$$

or

$$\Psi_{xx}(t_n, t_0) = \Phi(t_n, t_0) P_{xx}(t_0) \quad (4.148)$$

Thus the autocorrelation matrix for positive lags can be generated by simply left multiplying the zero-lag covariance kernel with the state transition matrix, very similar in form to transitioning the state as in equation 4.133.

It remains to develop an expression for negative lags of the autocorrelation matrix. Substituting $t_{i+1} = t_0$ and $t_i = t_m$, equation 4.133 can be rearranged as

$$x(t_m) = \Phi(t_0, t_m)^{-1} \{x(t_0) - w_d(t_m)\} \quad (4.149)$$

Substituting equation 4.149 into equation 4.140 with $t_p = t_m$ and $t_q = t_0$ yields

$$\Psi_{xx}(t_m, t_0) = E \left[ \Phi(t_0, t_m)^{-1} \{x(t_0) - w_d(t_m)\} x(t_0)^T \right] \quad (4.150)$$
which is rearranged to

\[ \Psi_{xx}(t_m, t_0) = \Phi(t_0, t_m)^{-1} E \left[ x(t_0) x(t_0)^T \right] - \Phi(t_0, t_m)^{-1} E \left[ w_d(t_m) x(t_0)^T \right] \]  

(4.151)

which can be further simplified by recognizing that \( E \left[ x(t_0) x(t_0)^T \right] = P_{xx}(t_0) \)

\[ \Psi_{xx}(t_m, t_0) = \Phi(t_0, t_m)^{-1} P_{xx}(t_0) - \Phi(t_0, t_m)^{-1} E \left[ w_d(t_m) x(t_0)^T \right] \]  

(4.152)

Note that, in contrast to the positive lag autocorrelation, the second expectation does not go to zero because the state \( x(t_0) \) is correlated to the noise in the past, \( w_d(t_m) \). To aid further development, equation 4.133 can be written in terms of \( t_0 \) and \( t_m \) as

\[ x(t_0) = \Phi(t_0, t_m) x(t_m) + w_d(t_m) \]  

(4.153)

and substituted into equation 4.152 to yield

\[ \Psi_{xx}(t_m, t_0) = \Phi(t_0, t_m)^{-1} P_{xx}(t_0) - \Phi(t_0, t_m)^{-1} E \left[ w_d(t_m) \{ \Phi(t_0, t_m) x(t_m) + w_d(t_m) \}^T \right] \]  

(4.154)

which simplifies to

\[ \Psi_{xx}(t_m, t_0) = \Phi(t_0, t_m)^{-1} \left\{ P_{xx}(t_0) - E \left[ w_d(t_m) w_d(t_m)^T \right] \right\} \]  

(4.155)

since the term \( E \left[ w_d(t_m) x(t_m)^T \right] = 0 \). Using equation 4.136, this simplifies to

\[ \Psi_{xx}(t_m, t_0) = \Phi(t_0, t_m)^{-1} \left[ P_{xx}(t_0) - Q_d(t_m) \right] \]  

(4.156)

where

\[ Q_d(t_m) = \int_{t_m}^{t_0} \Phi(t_0, t_m) Q(\tau) \Phi(t_0, t_m)^T d\tau \]  

(4.157)

Equation 4.156 is now analyzed for the case of a wide-sense stationary process. One property of a wide sense stationary process is that the zero-lag covariance kernel has reached steady state.
This can be expressed using equation 4.139 which can be written in terms of \( t_0 \) and \( t_m \) as

\[
P_{xx}(t_0) = \Phi(t_0, t_m) P_{xx}(t_m) \Phi(t_0, t_m)^T + Q_d(t_m)
\]  

(4.158)

For steady-state conditions

\[
P_{xx}(t_0) = P_{xx}(t_m)
\]  

(4.159)

which can be substituted into equation 4.158 to yield

\[
P_{xx}(t_0) = \Phi(t_0, t_m) P_{xx}(t_0) \Phi(t_0, t_m)^T + Q_d(t_m)
\]  

(4.160)

Equation 4.160 can then be substituted into equation 4.156 to yield

\[
\Psi_{xx}(t_m, t_0) = \Phi(t_0, t_m)^{-1} \left[ \Phi(t_0, t_m) P_{xx}(t_0) \Phi(t_0, t_m)^T + Q_d(t_m) - Q_d(t_m) \right]
\]  

(4.161)

which is rearranged to give

\[
\Psi_{xx}(t_m, t_0) = \Phi(t_0, t_m)^{-1} \Phi(t_0, t_m) P_{xx}(t_0) \Phi(t_0, t_m)^T
\]  

(4.162)

and simplifies to

\[
\Psi_{xx}(t_m, t_0) = P_{xx}(t_0) \Phi(t_0, t_m)^T
\]  

(4.163)

Since both the autocorrelation matrix and the zero-lag covariance kernel are symmetric, the following is true

\[
\Psi_{xx}(t_m, t_0) = \Psi_{xx}(t_0, t_0)^T = \Phi(t_0, t_m) P_{xx}(t_0)
\]  

(4.164)

And for a time-invariant system

\[
\Phi(t_0, t_m) = \Phi(t_n, t_0)
\]  

(4.165)

provided that \( t_n - t_0 = t_0 - t_m \), which yields the important and final result

\[
\Psi_{xx}(t_m, t_0) = \Phi(t_n, t_0) P_{xx}(t_0) = \Psi_{xx}(t_n, t_0) = \Psi(t_0, t_0 + \tau)
\]  

(4.166)

This equation shows that under the assumptions of a time-invariant (equation 4.165) and
wide-sense stationary system (equation 4.159), the autocorrelation kernel is an even function about
the nominal time \( t_0 \) that can be computed using the covariance at time \( t_0 \), and the transition matrix,
both of which are available from the LinCov simulation. Once samples of the autocorrelation kernel
are calculated, the power spectral density (PSD) can be estimated in a variety of ways. Mirroring
the approach taken for estimating the PSD in the Monte Carlo environment, a non-parametric
approach is chosen for estimating the PSD in the LinCov environment.

The chosen method is based on the definition that the PSD is the Fourier transform of the
autocorrelation function (see section 4.3 of [15]). To simplify the notation, let \( \Psi_{xx}^j (t_0, t_0 + \tau) \) be
the \( j \)th diagonal component of the autocorrelation kernel \( \Psi_{xx} (t_0, t_0 + \tau) \). Thus \( \Psi_{xx}^j (t_0, t_0 + \tau) \) is
the scalar autocorrelation function of the \( j \)th component of the random process vector \( \mathbf{x} \). The PSD
and the autocorrelation function are related by the continuous time Fourier transform

\[
\Psi_{xx}^j (f, t_0) = \int_{-\infty}^{\infty} \Psi_{xx}^j (t_0, t_0 + \tau) e^{-i2\pi f \tau} d\tau
\]  \hspace{1cm} (4.167)

\[
\Psi_{xx} (t_0, t_0 + \tau) = \int_{-\infty}^{\infty} \Psi_{xx}^j (f, t_0) e^{i2\pi f \tau} df
\]  \hspace{1cm} (4.168)

Since the simulation is implemented in a computer, the discrete Fourier transform (DFT)
must be used to evaluate equation 4.167. For the discrete Fourier transform to equal the continuous
time Fourier transform, the following criteria must be met

1. The autocorrelation function \( \Psi_{xx}^j (t_0, t_0 + \tau) \) must be finite. In other words, there must be
some time, \( \tau = \tau_f \), beyond which the value of the function is zero.

2. The PSD is band-limited such that \( \Psi_{xx}^j (f, t_0) = 0 \) when \( f \) is greater than the bandwidth, \( B \).

3. To prevent aliasing, the autocorrelation function is sampled at a rate greater than \( 2 \times \) the
bandwidth of the signal.

While few physical systems meet these three criteria, in most cases, they can be sufficiently satisfied
such that the results are meaningful. For example, the Hamming window can be applied to the
autocorrelation function to ensure that Criterion 1 is adequately met. This comes at the cost of
“windowing effects” which tend to smooth the peaks and raise the noise floor of the PSD estimate.
Regarding Criterion 2, even though most physical systems have infinite bandwidth, the magnitude
of the frequency content beyond some effective bandwidth \( B_{eff} \) is negligible, where it can be safely
assumed 0. Finally, knowledge of the effective bandwidth, $B_{eff}$, of the system can be used to set the sampling rate such that the aliasing discussed in Criterion 3 is also negligible.

In practice Criteria 2 and 3 are easily solved by appropriately setting the sampling rate of the simulation, often a very flexible parameter. Criterion 1 can however be difficult to satisfy especially in the case of a time varying system where the desire for a long data set to reduce the effects of the Hamming (or other) window competes with the need for a short data set such that the assumption of stationarity is valid. There is no general procedure that perfectly balances these two factors. Thus it is left to the developer of the simulation to use good engineering judgment.

In summary the equation that will be used for estimating the PSD from the LinCov simulation is

$$\bar{\Psi}_{xx}^j (f, t_0) = \int_{-\infty}^{\infty} W(\tau) \Psi_{xx}^j (t_0, t_0 + \tau) e^{-i2\pi f \tau} d\tau$$

where $W(\tau)$ is the same Hamming window used in the Monte Carlo simulation. The autocorrelation function $\Psi_{xx}^j (t_0, t_0 + \tau)$, is computed using the covariance and state transition matrix as stated in equation 4.166, and the Fourier transform integral is computed using the DFT.

4.6.3 LinCov Transfer Matrix Approach

Section 4.6.2 derived an approach that uses information available from the LinCov simulation to compute the autocorrelation function, under the assumptions of a time-invariant system with steady-state statistics. Once obtained, the autocorrelation function is windowed and discrete Fourier transformed to obtain the PSD of the dispersion states. In this section, a method is derived that directly computes the PSD of the dispersion states from the PSD of the input, without the need for windowing or the DFT. Maybeck [15] outlines a method for computing the output PSD of an scalar LTI system based on the impulse response and PSD of the input. Following the derivation template given by Maybeck, this approach is generalized to the case of a multiple-input, multiple-output (MIMO) LTI system.

The output $z(t) \in \mathbb{R}^{n \times 1}$ of a MIMO linear system can be computed by convolving the impulse response $G_t (t, t') \in \mathbb{R}^{n \times m}$ with the input $n(t) \in \mathbb{R}^{m \times 1}$

$$z(t) = n(t) \otimes G_t (t, t') = \int_{-\infty}^{t} G_t (t, t') n(t') dt'$$

(4.170)
where the impulse response describes the system output at time $t$ due to an impulse applied at time $t'$. For a time invariant system, the impulse response is a function only of the time difference $t - t'$, resulting in the following input-output relationship for a linear, time-invariant system.

$$z(t) = n(t) \otimes G_t(t - t') = \int_{-\infty}^{t} G_t(t - t') n(t') dt' \quad (4.171)$$

The change of variables $(t - t') = \tau$ yields a more convenient form.

$$z(t) = G_t(t) \otimes n(t) = \int_{0}^{\infty} G_t(\tau) n(t - \tau) d\tau \quad (4.172)$$

It is important to note that up to this point, two assumptions have been made. First is that the system is LTI. This enables expression of the impulse response as a function of the time difference $\tau$ and not the absolute time. The second assumption is that the input $n(t)$ has been applied since $t = -\infty$, implying that the steady-state response of the system has been reached. The autocorrelation kernel of the output is defined using the expected value operator

$$\Psi(t, t + \tau) = E\left\{z(t) z(t + \tau)^T\right\} \quad (4.173)$$

Substituting equation 4.172 into equation 4.173 yields

$$\Psi(t, t + \tau) = E\left\{\int_{0}^{\infty} G_t(\tau_1) n(t - \tau_1) d\tau_1 \int_{0}^{\infty} n(t + \tau - \tau_2)^T G_t(\tau_2)^T d\tau_2\right\} \quad (4.174)$$

which can be rearranged to

$$\Psi(t, t + \tau) = E\left\{\int_{0}^{\infty} \int_{0}^{\infty} G_t(\tau_1) n(t - \tau_1) n(t + \tau - \tau_2)^T G_t(\tau_2)^T d\tau_1 d\tau_2\right\} \quad (4.175)$$

Since the impulse response $G_t(t)$ is not a random quantity, the expectation operator can be moved inside the integral

$$\Psi(t, t + \tau) = \int_{0}^{\infty} \int_{0}^{\infty} G_t(\tau_1) E\left\{n(t - \tau_1) n(t + \tau - \tau_2)^T\right\} G_t(\tau_2)^T d\tau_1 d\tau_2 \quad (4.176)$$
Assuming the input \( n_t \) is wide-sense stationary, the central expectation term can be expressed as the autocorrelation kernel of the input evaluated at a lag of \( \tau + \tau_1 - \tau_2 \).

\[
E \left\{ n(t - \tau_1) n(t + \tau - \tau_2)^T \right\} G_t = \Psi_{nn}(\tau + \tau_1 - \tau_2) \tag{4.177}
\]

which results in the output autocorrelation kernel being a function of the time-difference \( \tau \), the input autocorrelation kernel, and the system impulse response

\[
\Psi_{zz}(\tau) = \int_{0}^{\infty} \int_{0}^{\infty} G_t(\tau_1) \Psi_{nn}(\tau + \tau_1 - \tau_2) G_t(\tau_2)^T d\tau_1 d\tau_2 \tag{4.178}
\]

It is important to note that for equation 4.178 to be finite, the system described by the impulse response \( G_t(\tau) \) must be stable, such that the steady-state variance (i.e. \( \Psi_{zz}(0) \)) is finite. For the purposes of this research, it is assumed that the system is stable.

The PSD of the output can then be computed by taking the Fourier transform of equation 4.178.

\[
\tilde{\Psi}_{zz}(\omega) = \int_{-\infty}^{\infty} \int_{0}^{\infty} G_t(\tau_1) \Psi_{nn}(\tau + \tau_1 - \tau_2) G_t(\tau_2)^T e^{-i\omega\tau} d\tau_1 d\tau_2 \tag{4.179}
\]

Changing the order of integration yields

\[
\tilde{\Psi}_{zz}(\omega) = \int_{0}^{\infty} \int_{0}^{\infty} G_t(\tau_1) \left\{ \Psi_{nn}(\tau + \tau_1 - \tau_2) e^{-i\omega\tau} d\tau_2 \right\} e^{-i\omega\tau_1} d\tau_1 d\tau_2 \tag{4.180}
\]

Using the shifting theorem of the Fourier transform, term A in equation 4.180 becomes

\[
\int_{-\infty}^{\infty} \Psi_{nn}(\tau + \tau_1 - \tau_2) e^{-i\omega\tau} d\tau = \tilde{\Psi}_{nn}(\omega) e^{-i\omega(\tau_2 - \tau_1)} \tag{4.181}
\]

Substituting equation 4.181 into 4.180 and rearranging yields

\[
\tilde{\Psi}_{zz}(\omega) = \int_{0}^{\infty} G_t(\tau_1) e^{i\omega\tau_1} d\tau_1 \tilde{\Psi}_{nn}(\omega) \left\{ \int_{0}^{\infty} G_t(\tau_2)^T e^{-i\omega\tau_2} d\tau_2 \right\} \tag{4.182}
\]

Terms \( B \) and \( C \) are the one-sided Fourier transform of the impulse response function, evaluated at \(-\omega\) and \(\omega\), respectively. This is equivalent to the Laplace transform with \( s = j\omega \). Due to physical
realizability, the one-sided transform is equal to the two-sided transform, yielding the following relationship.

\[ G(\omega) = \int_{-\infty}^{\infty} G_t(\tau) e^{-i\omega \tau} d\tau = \int_{0}^{\infty} G_t(\tau) e^{-i\omega \tau} d\tau \]  \hspace{1cm} (4.183)

Substituting equations 4.183 into 4.182 yields the final relationship between the PSD matrix of the input and the PSD matrix of the output.

\[ \bar{\Psi}_{zz}(\omega) = G(-\omega) \bar{\Psi}_{nn}(\omega) G(\omega)^T \]  \hspace{1cm} (4.184)

where \( G(\cdot) \) is the \( n \times m \) transfer matrix which maps each element of the \( m \times m \) input PSD matrix \( \bar{\Psi}_{nn}(\omega) \) to the appropriate element of the \( n \times n \) output PSD matrix. In the LinCov system summarized in section 4.4.7, the input noise sources are white and uncorrelated. The associated PSD matrix is therefore a diagonal matrix of noise strengths

\[ \bar{\Psi}_{nn}(\omega) = \text{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_m^2) \]  \hspace{1cm} (4.185)

The transfer matrix can be computed from the coefficient matrices of the augmented system in section 4.5. Rearranging equation 4.96 to group all process noise sources yields

\[ \dot{X} = FX + Dn \]  \hspace{1cm} (4.186)

where

\[ D = \begin{bmatrix} R & G & W \end{bmatrix} \]  \hspace{1cm} (4.187)

\[ n = \begin{bmatrix} \eta_c \\ \eta \\ w \end{bmatrix} \]  \hspace{1cm} (4.188)

Another matrix \( C \) is included to extract the states or combination of states being analyzed.

\[ z = CX \]  \hspace{1cm} (4.189)
Note that for the purposes of this research the $C$ matrix is limited to a row vector with a 1 corresponding to the state being analyzed. The transfer matrix can then be computed by taking the Laplace transform of equations 4.186 and 4.189

\[ sX(s) = FX(s) + Dn(s) \]  \hspace{1cm} (4.190)

\[ z(s) = CX(s) \]  \hspace{1cm} (4.191)

which can be combined to yield the well-known expression for the transfer matrix

\[ z(s) = C(sI - F)^{-1}Dn(s) \]  \hspace{1cm} (4.192)

As mentioned previously, the transfer matrix can be converted from the Laplace form to the Fourier form consistent with equation 4.184 by substituting $s = i\omega$.

In summary, under the assumptions of an LTI system, steady-state conditions, and wide-sense stationary input, the augmented system coefficient matrices are used in conjunction with the input noise strengths to compute the PSD of the chosen dispersion state. Like the LinCov autocorrelation approach, this method computes the PSD using the augmented system differential equation only (equation 4.96), and thus does not incorporate the full effects of the Kalman filter described in equation 4.97. In addition this approach computes the steady-state PSD and therefore neglects the effects of initial conditions.

This method, however, has two notable advantages over the LinCov autocorrelation approach. First the PSD can be computed without running a LinCov simulation. The information needed for the Transfer Matrix method resides in the coefficient matrices of equation 4.96, which are computed along the reference trajectory. This method also has the advantage that the Fourier transform is computed analytically and therefore does not suffer from the effects of data windowing and the DFT.

### 4.6.4 Frequency Domain Frequency Analysis Summary

Subsections 4.6.1, 4.6.2, and 4.6.3 outlined three methods for computing the power spectral density of the augmented system. The assumptions related to each method as well as the system
features captured by each method differ slightly. Tables 4.1 and 4.2 summarize the similarities and differences. The Monte Carlo-based approach is the most general. The only assumption made regarding the random process is that it is sufficiently stationary across the time window of the data. It makes no assumptions regarding a LTI system or steady-state statistics. The Monte Carlo method also accounts for all system features. Both the navigation propagation and Kalman update are accounted for in this method, as well as the effects of the controller. The LinCov Acorr and Transfer Matrix methods are similar in the assumptions made and features accounted for, with one subtle difference. In both cases, assumptions were made regarding stationarity, LTI, and steady-state. Both methods account for the closed-loop control and the propagation portion of the navigation filter. The difference in these two methods arises from the update portion of the Kalman filter. The LinCov Transfer Matrix method does not account for any of the effects of the Kalman filter. The LinCov Acorr method, however, partially accounts for these effects, by using the covariance from the LinCov simulation as the zero-lag autocorrelation value. This covariance value includes the effects of the Kalman updates up to that time instant, but when computing the autocorrelation values around that time instant, the Kalman updates are neglected.

Although differences exist amongst the three methods, the resulting PSDs are very similar for many systems of practical engineering importance. In the case of gimbaled pointing systems, the LTI assumption is often valid for angles close to the commanded orientation. In addition, the steady-state pointing performance is commonly the desired metric. Depending on the portion of the PSD considered, the relevant system features can also be very similar. For example, the updates of the Kalman filter are typically much lower in frequency than the update rate of the controller feedback. Thus, for the higher frequencies commonly ascribed to jitter, the effects of the Kalman filter are small, and the resulting PSDs in this region are similar. The following subsection demonstrates the use of the different methods and analyzes the effects of the assumptions listed in Table 4.1 for a system that does not contain a controller of Kalman filter.

4.6.5 LinCov/Monte Carlo Comparison

To illustrate the use of the Monte Carlo and LinCov based PSD estimation methods and compare the results, the analysis of a simple example is presented in this section. The random process to be examined is similar to a first-order Markov process driven by white noise, but with a
Table 4.1: Method comparison, assumptions

<table>
<thead>
<tr>
<th>Assumption</th>
<th>Monte Carlo</th>
<th>LinCov Acorr</th>
<th>LinCov Transfer Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stationary</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>LTI</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Steady-State</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 4.2: Method comparison, system features

<table>
<thead>
<tr>
<th>System Feature</th>
<th>Monte Carlo</th>
<th>LinCov Acorr</th>
<th>LinCov Transfer Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closed-loop control</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Navigation Propagation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Kalman Update</td>
<td>Yes</td>
<td>Yes/No</td>
<td>No</td>
</tr>
</tbody>
</table>

Varying time constant.

\[
\dot{x}(t) = \frac{1}{T(t)} x(t) + w(t) \tag{4.193}
\]

Where

\[
E[w(t)] = 0 \tag{4.194}
\]

\[
E[w(t)w(t')] = Q\delta(t-t') \tag{4.195}
\]

\[
T(t) = [2 + \cos(2\pi ft)] T_{\text{nominal}} \tag{4.196}
\]

The analytical autocorrelation function and PSD at some time \(t_0\) are, respectively (see [15], pages 184-185)

\[
\Psi_{xx}(t_0, t_0 + \Delta t) = \sigma^2(t_0) e^{-|\Delta t|/T(t_0)} \tag{4.197}
\]

\[
\bar{\Psi}_{xx}(\omega) = \frac{2\sigma^2(t_0)/T(t_0)}{\omega^2 + (1/T(t_0))^2} \tag{4.198}
\]

Where

\[
\sigma^2(t_0) = \frac{QT(t_0)}{2} \tag{4.199}
\]

It is important to note that in Maybeck’s development of the autocorrelation function and PSD, it is assumed that the system is wide-sense stationary and LTI. As will be shown in the following analysis, when the time constant \(T(t)\) changes slowly, these assumptions are valid. When
Table 4.3: Stationary simulation parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal time constant</td>
<td>$T_{\text{nominal}}$</td>
<td>0.1</td>
<td>s</td>
</tr>
<tr>
<td>Driving noise power spectral density</td>
<td>$Q$</td>
<td>1</td>
<td>m²/s²</td>
</tr>
<tr>
<td>Time constant frequency</td>
<td>$f_T$</td>
<td>0</td>
<td>Hz</td>
</tr>
<tr>
<td>Nominal time</td>
<td>$t_0$</td>
<td>49.99</td>
<td>s</td>
</tr>
<tr>
<td>Time constant at $t_0$</td>
<td>$T(t_0)$</td>
<td>0.3</td>
<td>s</td>
</tr>
<tr>
<td>Window length about $t_0$</td>
<td>$T_{\text{window}}$</td>
<td>1.2</td>
<td>s</td>
</tr>
<tr>
<td>Monte Carlo Runs</td>
<td>$N_{\text{mc}}$</td>
<td>300</td>
<td>unit-less</td>
</tr>
</tbody>
</table>

the time constant changes quickly, however, the time constant (and therefore the variance) of the random process changes too quickly to be approximated as stationary and LTI.

To establish a baseline, the frequency of the time constant is set to zero, such that the process is truly stationary and LTI, and the different PSD estimation techniques are compared. Table 4.3 lists the simulation parameter values used for this case, where the window length, $T_{\text{window}}$, corresponds to the time window of data used in both the Monte Carlo and LinCov simulations to compute the autocorrelation function and PSD.

Figure 4.2 shows the covariance as a function of time, as calculated from the LinCov simulation. The black vertical line denotes the nominal time, $t_0$, where the PSD is computed. Note that after the initial transient, the covariance is constant, and hence the system is truly stationary at the nominal time, $t_0$.

Figure 4.3 compares different ways of calculating the autocorrelation function to the analytical or true autocorrelation function in equation 4.197. Note that the analytical and non-windowed LinCov curves are identical in this case. The LinCov curve, however, approaches zero quicker which is to be expected given the Hamming window function. The Monte Carlo calculated autocorrelation function also differs from the analytical case but does not approach zero as fast as the windowed case. Thus, it is expected that the PSD estimated from the windowed case will contain power at frequencies higher than the other methods.

Figure 4.4 compares the three methods of computing the PSD to the analytical PSD. Note that for the case of a truly stationary/LTI system, the Monte Carlo-based method matches the
LinCov Acorr method. Both methods however exhibit a lower peak at 0Hz and a higher tail from 1 to 2Hz. This behavior is expected and is due to the effects of the Hamming window used by both methods. The LinCov Transfer Matrix method, however, matches the analytical PSD exactly.

Now that a baseline regarding PSD estimation has been established on a truly stationary/LTI system, the frequency corresponding to the changing time constant is set to 0.05Hz (see Table 4.4), with the remaining simulation parameters identical to the stationary case.

Figure 4.5 shows the covariance as a function of time, as calculated from the LinCov simulation. As before, the black vertical line denotes the nominal time, \( t_0 \), where the PSD is computed. Note that due to the time varying nature of the covariance, the system is not stationary. The statistics across the time window \( T_{\text{window}} \), are approximately constant. Specifically, the covariance varies by no more than 2% across the time window. Thus the system can be approximated as stationary across the chosen time window. Figure 4.6 compares the different methods for computing the auto-correlation function for the non-stationary case. The trends are very similar to the stationary case, showing near perfect agreement between the analytical and non-windowed LinCov curves.

Figure 4.7 illustrates the results of estimating the PSD via the different methods discussed
Figure 4.3: Autocorrelation function for the stationary simulation.

Figure 4.4: PSD for the stationary simulation.
<table>
<thead>
<tr>
<th>Description</th>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal time constant</td>
<td>$T_{\text{nominal}}$</td>
<td>0.1</td>
<td>s</td>
</tr>
<tr>
<td>Driving noise power spectral density</td>
<td>$Q$</td>
<td>1</td>
<td>$m^2/s^2$</td>
</tr>
<tr>
<td>Time constant frequency</td>
<td>$f_T$</td>
<td>0.05</td>
<td>Hz</td>
</tr>
<tr>
<td>Nominal time</td>
<td>$t_0$</td>
<td>49.99</td>
<td>s</td>
</tr>
<tr>
<td>Time constant at $t_0$</td>
<td>$T(t_0)$</td>
<td>0.3</td>
<td>s</td>
</tr>
<tr>
<td>Window length about $t_0$</td>
<td>$T_{\text{window}}$</td>
<td>1.2</td>
<td>s</td>
</tr>
<tr>
<td>Monte Carlo Runs</td>
<td>$N_{\text{mc}}$</td>
<td>300</td>
<td>unit-less</td>
</tr>
</tbody>
</table>

earlier. As in the case of a truly stationary system, the LinCov Acorr and Monte Carlo methods have good agreement, but again have a lower spectral peak and longer tails due to windowing effects. Nevertheless, the PSD estimate is adequate for most purposes. Also as before, the LinCov Transfer Matrix method matches the analytical PSD exactly.

To illustrate the behavior of the methods on a highly non-stationary/time varying system, the frequency of the time constant is set to 3Hz, with the other parameters identical to the previous two simulations, as shown in Table 4.5. Figure 4.8 illustrates the covariance of this system with the nominal time again indicated by the dashed black line. The covariance during the the time window varies rapidly, varying by a factor of 0.7 to 1.2 throughout the time window. Figure 4.9 illustrates the different autocorrelation functions. Note that the analytical curve for the autocorrelation function and PSD is not displayed, because the system is not truly stationary. Figure 4.10 illustrates the different PSD estimates. As in both previous cases, it is observed that the Monte Carlo and the LinCov Acorr methods match closely, even though the system is non-stationary. Figure 4.10 also illustrates an important aspect of the Transfer Matrix method which matches the “Analytical” curve. It is important to note that the analytical curve is computed using the time constant at the nominal time $t_0$ and input noise strength. In other words, the analytical curve, and therefore the Transfer Matrix method, represent the response of the system if it were truly LTI. This emphasizes the fact that Transfer Matrix method must only be used if the system is close to LTI.

The examples presented in this section illustrate the performance of the PSD estimation techniques. Good agreement is observed between the Monte Carlo method and the LinCov Acorr
Figure 4.5: Covariance for the non-stationary simulation.

Figure 4.6: Autocorrelation function for the non-stationary simulation.
Figure 4.7: PSD for the non-stationary simulation.

Table 4.5: Stationary simulation parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal time constant</td>
<td>$T_{nominal}$</td>
<td>0.1</td>
<td>s</td>
</tr>
<tr>
<td>Driving noise power spectral density</td>
<td>$Q$</td>
<td>1</td>
<td>$m^2/s^2$</td>
</tr>
<tr>
<td>Time constant frequency</td>
<td>$f_T$</td>
<td>3</td>
<td>Hz</td>
</tr>
<tr>
<td>Nominal time</td>
<td>$t_0$</td>
<td>49.99</td>
<td>s</td>
</tr>
<tr>
<td>Time constant at $t_0$</td>
<td>$T(t_0)$</td>
<td>0.3</td>
<td>s</td>
</tr>
<tr>
<td>Window length about $t_0$</td>
<td>$T_{window}$</td>
<td>1.2</td>
<td>s</td>
</tr>
<tr>
<td>Monte Carlo Runs</td>
<td>$N_{mc}$</td>
<td>300</td>
<td>unit-less</td>
</tr>
</tbody>
</table>
Figure 4.8: Covariance for highly non-stationary simulation.

Figure 4.9: Autocorrelation for the highly non-stationary simulation.
Figure 4.10: PSD for the highly non-stationary simulation.

method. This result is somewhat expected given the similarities between the two methods, most notably the use of the Hamming window. When the system is approximately LTI, however, the LinCov Transfer Matrix method matches the analytical PSD and is therefore the preferred method when this assumption is valid.
Chapter 5

Precision Pointing Simulation

To illustrate the utility of the theory developed in chapter 4, a system similar to the SOFIE instrument is analyzed [3]. To analyze the pointing and stabilization performance of the hosted, gimbaled pointing system, a closed-loop navigation and control simulation is created. Figure 5.1 illustrates the overall setup of the hosted gimbaled pointing system, which comprises a set of four rigid bodies connected by rotational and translational spring/damper elements. It is important to note that while the rigid body names are representative of a space-based, optical imaging system with a steering mirror, the model is generic and capable of representing a wide variety of gimbaled pointing systems, such as a turreted gun system on a helicopter [1] or a ball-gimbal imaging system on an aircraft [2].

The rigid body labeled SC (for spacecraft) represents the host vehicle and has specified rotational and translational acceleration PSDs. The acceleration profiles drive the dynamics of the rest of the system. The rigid body labeled RM (for reaction mass) represents the portion of the gimbaled pointing system that is attached to the SC via a set of translational and rotational spring/damper elements. The spring/damper elements model the compliance between the spacecraft and the gimbaled pointing system. The amount of compliance can range from soft vibration isolators to stiff structural members. The rigid body labeled BM (for base mass) represents the portion of the gimbal structure that is attached to the controlled rigid body, the SM (for steering mirror). Compliance between the RM and BM and the BM and SM is included to model structural modes internal to the gimbal structure. To control the orientation of the SM, motor torques \( T_m \) are applied to the SM and produce reaction torques on the BM. In addition, disturbance torques can be applied to the SM, modeling the effects of external disturbance sources such as wind loading, weapon recoil, etc. Finally, the rotational and translational dynamics of the system are coupled via the CG offset of the the SM. For a large CG offset, the coupling is strong. For a small/negligible offset, the translational dynamics do not affect the rotational dynamics, hence the importance of dynamically balancing the SM.
The system is equipped with many sensors to enable estimation and control of the orientation of the SM. A full attitude solution is produced on-board the RM via gyros aided by a star tracker.\(^1\) The inertial orientation of the SM is controlled by combining the RM attitude solution with resolvers attached between the SM and BM. The jitter of the SM is controlled by using feedback from gyros attached to the SM. The full derivation and validation of the system model shown in Figure 5.1 is documented in following sections.

The rest of this chapter is organized as follows. Section 5.1 details the truth dynamics and disturbance states. Section 5.2 describes the navigation algorithms and the noise models of the sensors used by the Kalman filter. Section 5.3 describes the control law, anti-resonance filter, and all the measurements used by the controller with their associated error models.

5.1 Structural Dynamics Model

The model incorporates the effects of structural dynamics, actuator torques, translation and rotational spacecraft disturbances, and center of mass offset of the gimbaled mirror. The model consists of 4 rigid bodies attached via linear springs and viscous dampers. This setup is sufficient to model up to 3 modes in the structure. Figure 5.1 details the setup of the system dynamics model, and Table 5.1 lists the associated parameters.

\(^1\)A more typical arrangement for a turreted gun or ball-gimbal imaging system is to have the attitude solution produced on the host vehicle. However, for a space application where the imaging sensor is very sensitive to vibration (and thus the amount of compliance between the the SC and RM is large) it is more common for the attitude solution be produced on-board the sensor.
Table 5.1: Model parameter descriptions

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1, K_2, K_3$</td>
<td>Torsional stiffness between the SC/RM, RM/BM, and BM/SM, respectively</td>
</tr>
<tr>
<td>$C_1, C_2, C_3$</td>
<td>Torsional damping coef. between the SC/RM, RM/BM, and BM/SM, respectively</td>
</tr>
<tr>
<td>$K_4, K_5, K_6$</td>
<td>Linear stiffness between the SC/RM, RM/BM, and BM/SM, respectively</td>
</tr>
<tr>
<td>$C_4, C_5, C_6$</td>
<td>Linear damping coefficients between the SC/RM, RM/BM, and BM/SM, respectively</td>
</tr>
<tr>
<td>$\theta, \lambda, \phi, \psi$</td>
<td>Inertial orientation of the SC, RM, BM, and SM, respectively</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Location of the SM center of mass from the SM attachment point</td>
</tr>
<tr>
<td>$T_m$</td>
<td>Torque applied by an actuator between the SM and CM</td>
</tr>
<tr>
<td>$T_D$</td>
<td>External disturbance torques applied to the SM</td>
</tr>
<tr>
<td>$M, m_1, m_2, m_3$</td>
<td>Mass of the SC, RM, CM, and SM, respectively</td>
</tr>
<tr>
<td>$J, J_1, J_2, J_3$</td>
<td>Mass moment of inertia of the SC, RM, CM, and SM, respectively</td>
</tr>
<tr>
<td>$F = Ma_{trans}$</td>
<td>Force acting on the SC</td>
</tr>
<tr>
<td>$T = Ja_{rot}$</td>
<td>Torque acting on the SC</td>
</tr>
</tbody>
</table>
5.1.1 Disturbance Sources

Three disturbances act on the structure of the gimbal. A translational and rotational acceleration act on the SC while a disturbance torque acts on the SM. All of the disturbances are modeled as colored noise with the following differential equations

\[ \dot{a}_{\text{trans}} = -\frac{a_{\text{trans}}}{\tau_{\text{trans}}} + w_{\text{trans}} \]  
\[ \dot{a}_{\text{rot}} = -\frac{a_{\text{rot}}}{\tau_{\text{rot}}} + w_{\text{rot}} \]  
\[ \dot{T}_D = -\frac{T_D}{\tau_D} + w_D \]

where the driving noise terms are

\[ E[w_{\text{trans}} (t)] = 0, \quad E[w_{\text{trans}} (t) w_{\text{trans}} (t')] = \frac{2\sigma_{\text{trans,ss}}^2}{\tau_{\text{trans}}} \delta(t - t') \]  
\[ E[w_{\text{rot}} (t)] = 0, \quad E[w_{\text{rot}} (t) w_{\text{rot}} (t')] = \frac{2\sigma_{\text{rot,ss}}^2}{\tau_{\text{rot}}} \delta(t - t') \]  
\[ E[w_D (t)] = 0, \quad E[w_D (t) w_D (t')] = \frac{2\sigma_{D,ss}^2}{\tau_D} \delta(t - t') \]

where \( \sigma_{\text{trans,ss}}^2, \sigma_{\text{rot,ss}}^2, \) and \( \sigma_{D,ss}^2 \) are the steady-state covariances for the translational acceleration, rotational acceleration, and SM disturbance torques, respectively. The parameters \( \tau_{\text{trans}}, \tau_{\text{rot}}, \) and \( \tau_D \) are the corresponding time constants.

5.1.2 Free Body Diagrams

Figure 5.2 shows the free body diagrams of each rigid body and center of mass offset of the SM. Equations 5.7 to 5.12 define the values of the intermediate torques and forces.

\[ T_1 = K_1 (\lambda - \theta) + C_1 (\dot{\lambda} - \dot{\theta}) \]  
\[ T_2 = K_2 (\dot{\phi} - \lambda) + C_2 (\dot{\phi} - \dot{\lambda}) \]  
\[ T_3 = K_3 (\psi - \phi) + C_3 (\dot{\psi} - \dot{\phi}) \]  
\[ F_1 = K_4 (x_1 - x) + C_4 (\dot{x}_1 - \dot{x}) \]
5.1.3 Kinematic Constraints

Assuming small $\psi$, the following kinematic relationship holds

$$x_4 = x_3 - \psi \epsilon$$  \hspace{1cm} (5.13)

where $x_4$ is the displacement of the SM center of mass. Taking the first and second time derivative of equation 5.13 yields the following two kinematic constraints

$$\dot{x}_4 = \dot{x}_3 - \psi \epsilon$$  \hspace{1cm} (5.14)

$$\ddot{x}_4 = \ddot{x}_3 - \psi \epsilon$$  \hspace{1cm} (5.15)

5.1.4 Equations of Motion

Given the free body diagrams and force/torque definitions of section 5.1.2, the following four equations of motion can be written for the rotational dynamics.

$$J \ddot{\theta} = T + T_1 = J_{\text{rot}} + K_1 (\lambda - \theta) + C_1 \left( \dot{\lambda} - \dot{\theta} \right)$$  \hspace{1cm} (5.16)

$$J_1 \ddot{\lambda} = T_2 - T_1 = K_2 (\phi - \lambda) + C_2 \left( \dot{\phi} - \dot{\lambda} \right) - K_1 (\lambda - \theta) - C_1 \left( \dot{\lambda} - \dot{\theta} \right)$$  \hspace{1cm} (5.17)
\[ J_2 \ddot{\phi} = T_3 - T_2 - T_m \]  
\[ = K_3 (\psi - \phi) + C_3 (\dot{\psi} - \dot{\phi}) - K_2 (\phi - \lambda) - C_2 (\dot{\phi} - \dot{\lambda}) - T_m \]  
\[ (5.18) \]

\[ J_3 \ddot{\psi} = T_m + T_D - T_3 - F_3 \epsilon \]
\[ = T_m + T_D - K_3 (\psi - \phi) - C_3 (\dot{\psi} - \dot{\phi}) - \epsilon K_6 (x_3 - x_2) - C_6 (\dot{x}_3 - \dot{x}_2) \]  
\[ (5.19) \]

where \( J_3 \) can be computed using the parallel axis theorem and the inertia of the SM about the attachment point \( J_{SM} \)
\[ J_3 = J_{SM} - m_3 \epsilon^2 \]  
\[ (5.20) \]

Likewise, the following four translational equations of motion can be written
\[ M \ddot{x} = F + F_1 = Ma_{trans} + K_4 (x_1 - x) + C_4 (\dot{x}_1 - \dot{x}) \]  
\[ (5.21) \]

\[ m_1 \ddot{x}_1 = F_2 - F_1 = K_5 (x_2 - x_1) + C_5 (\dot{x}_2 - \dot{x}_1) - K_4 (x_1 - x) - C_4 (\dot{x}_1 - \dot{x}) \]  
\[ (5.22) \]

\[ m_2 \ddot{x}_2 = F_3 - F_2 = K_6 (x_3 - x_2) + C_6 (\dot{x}_3 - \dot{x}_2) - K_5 (x_2 - x_1) - C_5 (\dot{x}_2 - \dot{x}_1) \]  
\[ (5.23) \]

\[ m_3 \ddot{x}_4 = -F_3 = -K_6 (x_3 - x_2) - C_6 (\dot{x}_3 - \dot{x}_2) \]  
\[ (5.24) \]

Note that the dependence of equation 5.25 on \( \ddot{x}_4 \) can be eliminated by substituting equation 5.15 into equation 5.25 to yield
\[ m_3 (\dot{x}_3 - \dot{\psi} \epsilon) = -K_6 (x_3 - x_2) - C_6 (\dot{x}_3 - \dot{x}_2) \]  
\[ (5.25) \]

Finally note that equation 5.26 has two terms with double time derivatives, namely \( \dddot{x}_3 \) and \( \dddot{x}_4 \). This causes problems when converting to state space form. Therefore, it remains to convert equation 5.26 into an equation that depends only on one double derivative term. Rearranging equation 5.26 yields
\[ m_3 \dddot{x}_3 = -K_6 (x_3 - x_2) - C_6 (\dot{x}_3 - \dot{x}_2) + m_3 \dddot{\psi} \epsilon \]  
\[ (5.26) \]
Substituting equation 5.20 into equation 5.27 yields

\[ m_3 \ddot{x}_3 = -K_6 (x_3 - x_2) - C_6 (\dot{x}_3 - \dot{x}_2) \]
\[ + \frac{m_3 \epsilon}{J_3} \left[ T_m + T_D - K_3 (\dot{\psi} - \dot{\phi}) - C_3 (\dot{\psi} - \dot{\phi}) \right] \]
\[ - \frac{m_3 \epsilon}{J_3} \left[ \epsilon K_6 (x_3 - x_2) + \epsilon C_6 (\dot{x}_3 - \dot{x}_2) \right] \]

5.28

5.1.5 State Space Representation

Equations 5.16-5.20, 5.22-5.24, and 5.28 with the definition in equation 5.21 form a complete set of differential equations that can be put into state space form. The chosen state space form is shown below. The subscript \( g \) denotes parameters belonging to the physical model of the gimbal pointing system and is included to avoid confusion with other states in this dissertation. The subscript \( d \) denotes disturbance states.

\[ \dot{x}_g = A_g x_g + B_{g1} x_d + B_{g2} T_m \]

5.29

\[ x_g \equiv \begin{bmatrix} x & x_1 & x_2 & x_3 & \theta & \lambda & \phi & \psi & \dot{x}_1 & \dot{x}_2 & \dot{x}_3 & \dot{\theta} & \dot{\lambda} & \dot{\phi} & \dot{\psi} \end{bmatrix}^T \]

5.30

\[ x_d \equiv \begin{bmatrix} a_{\text{trans}} & a_{\text{rot}} & T_D \end{bmatrix}^T \]

5.31

For conciseness, the \( A_g, B_{g1}, \) and \( B_{g2} \) matrices can be partitioned

\[ A_g = \begin{bmatrix} 0_{8 \times 8} & I_{8 \times 8} \\ F_{8 \times 8} & G_{8 \times 8} \end{bmatrix} \]

5.32

\[ B_{g1} = \begin{bmatrix} 0_{8 \times 3} \\ K_{g1} \end{bmatrix} \]

5.33
\[ B_{y2} = \begin{bmatrix} 0_{8 \times 1} \\ K_{y2} \end{bmatrix} \]  

(5.34)

where

\[ F = \begin{bmatrix} \frac{-K_4}{M} & \frac{K_4}{M} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{K_4}{m_1} & -\frac{(K_3 + K_4)}{m_1} & \frac{K_5}{m_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{K_5}{m_2} & -\frac{(K_6 + K_5)}{m_2} & \frac{K_6}{m_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \left( \frac{K_6}{m_3} + \frac{K_6 \epsilon^2}{J_3} \right) & -\left( \frac{K_6}{m_3} + \frac{K_6 \epsilon^2}{J_3} \right) & 0 & 0 & -\frac{K_3}{J_3} & -\frac{-K_3}{J_3} \\ 0 & 0 & 0 & 0 & -\frac{K_1}{J_1} & \frac{K_1}{J_1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{K_1}{J_1} & -\frac{(K_2 + K_1)}{J_1} & \frac{K_2}{J_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{K_2}{J_2} & -\frac{(K_3 + K_2)}{J_2} & \frac{K_3}{J_2} \\ 0 & 0 & \frac{\epsilon K_6}{J_3} & -\frac{-\epsilon K_6}{J_3} & 0 & 0 & \frac{K_3}{J_3} & -\frac{-K_3}{J_3} \end{bmatrix} \]  

(5.35)

\[ G = \begin{bmatrix} \frac{-C_4}{M} & \frac{C_4}{M} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{C_4}{m_1} & -\frac{(C_5 + C_4)}{m_1} & \frac{C_5}{m_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{C_5}{m_2} & -\frac{(C_6 + C_5)}{m_2} & \frac{C_6}{m_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \left( \frac{C_6}{m_3} + \frac{C_6 \epsilon^2}{J_3} \right) & -\left( \frac{C_6}{m_3} + \frac{C_6 \epsilon^2}{J_3} \right) & 0 & 0 & \frac{C_3 \epsilon}{J_3} & -\frac{-C_3 \epsilon}{J_3} \\ 0 & 0 & 0 & 0 & -\frac{C_1}{J_1} & \frac{C_1}{J_1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{C_1}{J_1} & -\frac{(C_2 + C_1)}{J_1} & \frac{C_2}{J_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{C_2}{J_2} & -\frac{(C_3 + C_2)}{J_2} & \frac{C_3}{J_2} \\ 0 & 0 & \frac{\epsilon C_6}{J_3} & -\frac{-\epsilon C_6}{J_3} & 0 & 0 & \frac{C_3}{J_3} & -\frac{-C_3}{J_3} \end{bmatrix} \]  

(5.36)
and

\[
K_{g1} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & \epsilon/J_3 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1/J_3
\end{bmatrix}
\]  \hspace{1cm} (5.37)

\[
K_{g2} = \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
\epsilon/J_3 \\
0 \\
0 \\
-1/J_2 \\
1/J_3
\end{bmatrix}
\]  \hspace{1cm} (5.38)

Though it is not required for the LinCov or Monte Carlo simulations it is often convenient to define an output equation

\[
y_g = C_g x_g
\]  \hspace{1cm} (5.39)

where the matrix \(C_g\) can be chosen to satisfy the goals of the analysis. For example, if one were
designing a SM rate damping controller using angular rate feedback, the appropriate choice for the matrix \( C_g \) would be

\[
C_g = \begin{bmatrix} 0_{1 \times 15} & 1 \end{bmatrix}
\]

such that the output is equal to the inertial angular rate of the SM.

\[
y_g = \dot{\psi}
\]

5.1.6 Model validation

To ensure correctness of the structural dynamics model, it was desired to validate the model by comparing to ADAMS, a commercial dynamics software package. Figure 5.3 illustrates the model that was developed in ADAMS. From left to right, the rigid bodies correspond to the SC, RM, BM, and SM, with each body connected by a translational and rotational spring/damper element. In ADAMS, all of the rigid bodies were constrained to translate only in the x-axis and rotate only in the z-axis, thus matching the model developed in the preceding subsections.

Recall that the dynamics model comprises four inputs: three disturbances and one actuator. The method of validation chosen is to compare the impulse response of the ADAMS model to the model derived in the previous subsections. The chosen output is the inertial angle of the SM, or \( \psi \). Thus, the \( C \) matrix for the simulation is

\[
C_g = \begin{bmatrix} 0_{1 \times 7} & 1 & 0_{1 \times 8} \end{bmatrix}^T
\]
The simulation parameters are listed in Table 5.2. Note that these values were not chosen as representative of any real system. They were chosen such that the dynamics of all the rigid bodies would be highly coupled. In other words, by applying forces or torques to one rigid body, all rigid bodies would move significantly.

The responses analyzed are the impulse response to each of the input torques or forces (4 total) plus the impulse response of all inputs, simultaneously. For each impulse response, two plots are shown: one of the SM inertial angle $\psi$ for the ADAMS model and the derived model, and one of the difference between the angle, or the error angle. The impulses are applied at $t = 1$ sec with the values shown in Table 5.3.

Figures 5.4-5.7 show the impulse responses of the dynamics model due to the force acting on the spacecraft ($F$), torque acting on the spacecraft ($T$), SM motor torque ($T_m$), and the disturbance torque ($T_D$), respectively. Figure 5.8 shows the impulse response due to all of the inputs simultaneously. In all cases, large scale agreement is excellent. Small-scale discrepancies grow very slowly in some cases, typical of numerical integration.

5.2 Navigation Algorithm Development

This section documents the development of the navigation algorithm. Referring again to Figure 5.1, the RM is equipped with a star tracker and gyro. A Kalman filter onboard the RM uses the gyro to propagate the attitude with updates from the star tracker. It is important to note that
Table 5.2: Physical system parameters used for validating the model.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>10</td>
<td>Nm/rad</td>
</tr>
<tr>
<td>$K_2$</td>
<td>20</td>
<td>Nm/rad</td>
</tr>
<tr>
<td>$K_3$</td>
<td>22</td>
<td>Nm/rad</td>
</tr>
<tr>
<td>$C_1$</td>
<td>0.15</td>
<td>Nm/(rad/s)</td>
</tr>
<tr>
<td>$C_2$</td>
<td>0.32</td>
<td>Nm/(rad/s)</td>
</tr>
<tr>
<td>$C_3$</td>
<td>$5 \times 10^{-2}$</td>
<td>Nm/(rad/s)</td>
</tr>
<tr>
<td>$K_4$</td>
<td>1</td>
<td>N/m</td>
</tr>
<tr>
<td>$K_5$</td>
<td>1.2</td>
<td>N/m</td>
</tr>
<tr>
<td>$K_6$</td>
<td>1.5</td>
<td>N/m</td>
</tr>
<tr>
<td>$C_4$</td>
<td>0.1</td>
<td>N/(m/s)</td>
</tr>
<tr>
<td>$C_5$</td>
<td>0.2</td>
<td>N/(m/s)</td>
</tr>
<tr>
<td>$C_6$</td>
<td>0.3</td>
<td>N/(m/s)</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>0.03</td>
<td>m</td>
</tr>
<tr>
<td>$M$</td>
<td>$1 \times 10^{12}$</td>
<td>kg</td>
</tr>
<tr>
<td>$m_1$</td>
<td>4</td>
<td>kg</td>
</tr>
<tr>
<td>$m_2$</td>
<td>5</td>
<td>kg</td>
</tr>
<tr>
<td>$m_3$</td>
<td>6</td>
<td>kg</td>
</tr>
<tr>
<td>$J$</td>
<td>$1 \times 10^{12}$</td>
<td>kg $\cdot$ m$^2$</td>
</tr>
<tr>
<td>$J_1$</td>
<td>1</td>
<td>kg $\cdot$ m$^2$</td>
</tr>
<tr>
<td>$J_2$</td>
<td>2</td>
<td>kg $\cdot$ m$^2$</td>
</tr>
<tr>
<td>$J_3$</td>
<td>3</td>
<td>kg $\cdot$ m$^2$</td>
</tr>
</tbody>
</table>

Table 5.3: Impulse values.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F = m \ddot{x}_e$</td>
<td>$1 \times 10^{12}$</td>
<td>N</td>
</tr>
<tr>
<td>$T = J \ddot{\theta}_e$</td>
<td>$1 \times 10^{12}$</td>
<td>N $\cdot$ m</td>
</tr>
<tr>
<td>$T_m$</td>
<td>1</td>
<td>N $\cdot$ m</td>
</tr>
<tr>
<td>$T_D$</td>
<td>1</td>
<td>N $\cdot$ m</td>
</tr>
</tbody>
</table>
Figure 5.5: SM inertial angle impulse response due to the torque ($T$) acting on the spacecraft. Left-SM joint angle for the derived and ADAMS models. Right-Difference between derived and ADAMS models.

Figure 5.6: SM inertial angle impulse response due to the SM motor torque ($T_m$). Left-SM joint angle for the derived and ADAMS models. Right-Difference between derived and ADAMS models.
Figure 5.7: SM inertial angle impulse response due to the disturbance torque ($T_D$). Left-SM joint angle for the derived and ADAMS models. Right-Difference between derived and ADAMS models.

Figure 5.8: SM inertial angle impulse response due to the disturbance torque ($T_D$). Left-SM joint angle for the derived and ADAMS models. Right-Difference between derived and ADAMS models.
the navigation filter only attempts to estimate that state of the RM. Due to the compliance between structural members, the orientation of the SC, BM, and SM can differ from the value estimated by the Kalman filter. This section defines the true error models for each measurement available to the Kalman filter followed by the definition of the dynamics of the sensor parameters. A design model is built based on the true models, with gyro measurements replacing the dynamics of the RM. The design model is then used to develop the estimation algorithms for the Kalman filter.

5.2.1 Truth Model

The portion of the truth model corresponding to the dynamics of the inertial angle $\lambda$ was developed in the previous section. In this section it remains to define the measurements available to the Kalman filter along with the dynamics of the measurement error sources. The star tracker attached to the RM measures the true value of the inertial angle $\lambda$, corrupted by bias and noise.

$$\tilde{\lambda}_k = \lambda_k + b_{\text{star},k} + \eta_{\text{star},k}$$ (5.43)

$$E[\eta_{\text{star},k}] = 0, \quad E[\eta_{\text{star},k}\eta_{\text{star},k}'] = \sigma_{\text{star}}^2 \delta_{kk'}$$ (5.44)

The gyro measures the inertial angular rate of the RM corrupted by scale factor, bias, and noise.

$$\tilde{\omega}_{\text{gyro}} = (1 + s_{\text{gyro}}) \left( \dot{\lambda} + b_{\text{gyro}} + \eta_{\text{gyro}} \right)$$ (5.45)

$$E[\eta_{\text{gyro}}(t)] = 0, \quad E[\eta_{\text{gyro}}(t)\eta_{\text{gyro}}(t')] = \sigma_{\text{gyro}}^2 \delta(t-t')$$ (5.46)

The truth models for the dynamics of the parameter states are modeled as first-order Markov processes

$$\dot{b}_{\text{star}} = -\frac{b_{\text{star}}}{\tau_{\text{star}}} + w_{\text{star}}$$ (5.47)

$$\dot{b}_{\text{gyro}} = -\frac{b_{\text{gyro}}}{\tau_{\text{gyro}}} + w_{\text{gyro}}$$ (5.48)

$$\dot{s}_{\text{gyro}} = -\frac{s_{\text{gyro}}}{\tau_{\text{gyro}}} + w_{\text{gyro}}$$ (5.49)
with Gaussian noise inputs

$$E[w_{\text{star}}(t)] = 0, \quad E[w_{\text{star}}(t)w_{\text{star}}(t')] = \frac{2\sigma^2_{\text{star},ss}}{\tau_{\text{star}}} \delta(t - t')$$ (5.50)

$$E[w_{\text{gyro1}}(t)] = 0, \quad E[w_{\text{gyro1}}(t)w_{\text{gyro1}}(t')] = \frac{2\sigma^2_{\text{gyro1},ss}}{\tau_{\text{gyro1}}} \delta(t - t')$$ (5.51)

$$E[w_{\text{sgyro1}}(t)] = 0, \quad E[w_{\text{sgyro1}}(t)w_{\text{sgyro1}}(t')] = \frac{2\sigma^2_{\text{sgyro1},ss}}{\tau_{\text{sgyro1}}} \delta(t - t')$$ (5.52)

where $\sigma^2_{\text{star},ss}$, $\sigma^2_{\text{gyro1},ss}$, and $\sigma^2_{\text{sgyro1},ss}$ are the steady-state variance of the star camera bias, gyro 1 bias, and gyro 1 scale factor, respectively. The parameters $\tau_{\text{star}}$, $\tau_{\text{gyro1}}$, and $\tau_{\text{sgyro1}}$ are the corresponding time constants.

### 5.2.2 Design Model

The definition of a design model is a bit of a formalism but is important in understanding the derivation of the Kalman filter. The lifetime of the design model is very brief. It is built, marginalized, linearized, then discarded, but the result of this process is an equation to propagate estimates of the mean and covariance of the state. The design model in this problem is defined consistent with the truth models outlined previously, with the exception of model replacement for the dynamics of the RM inertial angle $\lambda$. The design model state vector consists of the RM inertial angle, star camera bias, gyro bias, and scale factor.

$$\mathbf{x}^{dm} = \begin{bmatrix} \lambda^{dm} & b_{\text{star}}^{dm} & b_{\text{gyro1}}^{dm} & s_{\text{gyro1}}^{dm} \end{bmatrix}^T$$ (5.53)

The dynamics of the design model states are defined by the following set of differential equations

$$\dot{\lambda}^{dm} = \omega^{dm}_{\text{gyro1}} = \frac{\ddot{\omega}_{\text{gyro1}}}{1 + s_{\text{gyro1}}^{dm}} - b_{\text{gyro1}}^{dm} - \tau_{\text{gyro1}}^{dm}$$ (5.54)

$$\dot{b}_{\text{star}}^{dm} = -\frac{b_{\text{star}}^{dm}}{\tau_{\text{star}}} + w_{\text{star}}^{dm}$$ (5.55)

$$\dot{b}_{\text{gyro1}}^{dm} = -\frac{b_{\text{gyro1}}^{dm}}{\tau_{\text{gyro1}}} + w_{\text{gyro1}}^{dm}$$ (5.56)

$$\dot{s}_{\text{gyro1}}^{dm} = -\frac{s_{\text{gyro1}}^{dm}}{\tau_{\text{sgyro1}}} + w_{\text{sgyro1}}^{dm}$$ (5.57)
and the process noise is defined identical to the truth models discussed earlier

\[
E\left[\eta_{\text{gyro}}(t)\right] = 0, \quad E\left[\eta_{\text{gyro}}(t)\eta_{\text{gyro}}(t')\right] = \sigma_{\text{gyro}}^2 \delta(t - t') \tag{5.58}
\]

\[
E\left[w_{\text{star}}(t)\right] = 0, \quad E\left[w_{\text{star}}(t)w_{\text{star}}(t')\right] = \frac{2\sigma_{\text{star}, \text{ss}}^2}{\tau_{\text{star}}} \delta(t - t') \tag{5.59}
\]

\[
E\left[w_{\text{bgyro}}(t)\right] = 0, \quad E\left[w_{\text{bgyro}}(t)w_{\text{bgyro}}(t')\right] = \frac{2\sigma_{\text{bgyro}, \text{ss}}^2}{\tau_{\text{bgyro}}} \delta(t - t') \tag{5.60}
\]

\[
E\left[w_{\text{sgyro}}(t)\right] = 0, \quad E\left[w_{\text{sgyro}}(t)w_{\text{sgyro}}(t')\right] = \frac{2\sigma_{\text{sgyro}, \text{ss}}^2}{\tau_{\text{sgyro}}} \delta(t - t') \tag{5.61}
\]

The set of design model differential equations can be combined into a single, nonlinear vector differential equation

\[
\dot{x}^{dm} = \hat{f} \left( x^{dm}, \tilde{\omega}_{\text{gyro}} \right) + \hat{B} w^{dm} \tag{5.62}
\]

where

\[
\hat{f} \left( x^{dm}, \tilde{\omega}_{\text{gyro}} \right) = \begin{bmatrix}
\frac{\tilde{\omega}_{\text{gyro}}}{(1 + s_{\text{gyro}}^{dm})} - \eta_{\text{gyro}}^{dm} \\
-\frac{\eta_{\text{star}}^{dm}}{\tau_{\text{star}}} \\
-\frac{\eta_{\text{bgyro}}^{dm}}{\tau_{\text{bgyro}}} \\
-\frac{\eta_{\text{sgyro}}^{dm}}{\tau_{\text{sgyro}}}
\end{bmatrix}, \quad \hat{B} = \begin{bmatrix}
-1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}, \quad w^{dm} = \begin{bmatrix}
\eta_{\text{gyro}}^{dm} \\
\eta_{\text{star}}^{dm} \\
\eta_{\text{bgyro}}^{dm} \\
\eta_{\text{sgyro}}^{dm}
\end{bmatrix}
\]

Finally, the design model measurement is defined as the true inertial angle of RM corrupted by bias and noise

\[
z_k^{dm} = \hat{h} \left( x_k^{dm} \right) + \eta_{\text{star}, k}^{dm} = \lambda_k^{dm} + \nu_{\text{star}, k}^{dm} + \eta_{\text{star}, k}^{dm} \tag{5.63}
\]

where the noise has the following properties.

\[
E\left[\eta_{\text{star}, k}^{dm}\right] = 0, \quad E\left[\eta_{\text{star}, k}^{dm}\eta_{\text{star}, k'}^{dm}\right] = \sigma_{\text{star}}^2 \delta_{kk'} \tag{5.64}
\]

5.2.3 Navigation Filter

The navigation state vector is defined identical to the design model
\[
\dot{\hat{x}} = \begin{bmatrix} \dot{\hat{\lambda}} & \dot{\hat{b}_{\text{star}}} & \dot{\hat{b}_{\text{gyro}}} & \dot{\hat{s}_{\text{gyro}}} \end{bmatrix}^T
\]

(5.65)

The propagation of the navigation state vector and covariance matrix involves linearizing equation 5.62 about the current estimate. The derivation is a lengthy one, so only the end results are included here

\[
\dot{\hat{x}} = \hat{\mathbf{f}} (\hat{x}, \hat{\omega}_{\text{gyro}}), \quad \hat{x}_0 = \text{given} \tag{5.66}
\]

\[
\dot{\hat{P}} = \hat{\mathbf{F}} \hat{P} + \hat{\mathbf{F}}^T \hat{\mathbf{B}} \hat{\mathbf{Q}} \hat{\mathbf{B}}^T, \quad \hat{P}_0 = \text{given} \tag{5.67}
\]

where

\[
\hat{\mathbf{Q}} = \mathbb{E} \left[ \mathbf{w}_{\text{dm}} (\mathbf{w}_{\text{dm}})^T \right] = \begin{bmatrix}
\sigma_{\text{gyro}}^2 & 0 & 0 & 0 \\
0 & 2\sigma_{\text{star},ss}^2 / \tau_{\text{star}} & 0 & 0 \\
0 & 0 & 2\sigma_{\text{gyro1},ss}^2 / \tau_{\text{gyro1}} & 0 \\
0 & 0 & 0 & 2\sigma_{\text{sgyro1},ss}^2 / \tau_{\text{sgyro1}}
\end{bmatrix} \tag{5.68}
\]

\[
\hat{\mathbf{F}} = \left. \frac{\partial \hat{\mathbf{f}} (\mathbf{x}_{\text{dm}}, \hat{\omega}_{\text{gyro}})}{\partial \mathbf{x}_{\text{dm}}} \right|_{\hat{x}} = \begin{bmatrix}
0 & 0 & -1 & \frac{-\hat{\omega}_{\text{gyro1}}}{(1+\hat{s}_{\text{gyro1}})^2} \\
0 & -1 / \tau_{\text{star}} & 0 & 0 \\
0 & 0 & -1 / \tau_{\text{gyro1}} & 0 \\
0 & 0 & 0 & -1 / \tau_{\text{gyro1}}
\end{bmatrix} \tag{5.69}
\]

To complete the navigation algorithm, it remains to define the navigation state and navigation state covariance update equations given the measurement. The update equations are detailed below.

\[
\hat{x}^+_k = \hat{x}_k + \hat{K}_k \left[ \hat{\lambda}_k - \hat{\mathbf{h}} (\hat{x}^-_k) \right] \tag{5.70}
\]

\[
\hat{P}^+_k = (I - \hat{K}_k \hat{H}_k) \hat{P}^-_k (I - \hat{K}_k \hat{H}_k)^T + \hat{K}_k \sigma_{\text{star}}^2 \hat{K}_k^T \tag{5.71}
\]

with

\[
\hat{\mathbf{H}}_k = \left. \frac{\partial \hat{\mathbf{h}} (\mathbf{x}_{\text{dm}})}{\partial \mathbf{x}_{\text{dm}}} \right|_{\hat{x}} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}
\]
\[
\hat{K}_k = \hat{P}_k^{-1} \hat{H}_k^T \left( \hat{H} \hat{P}_k^{-1} \hat{H}_k^T + \sigma_{\text{star}}^2 \right)^{-1}
\]

Where the superscripts - and + denote the state or state covariance before and after the update, respectively.

5.3 Control Algorithm Development

The purpose of this section is to document the development of the control algorithm, command filtering, and the associated control system stability and performance specifications. Figure 5.1 illustrates the overall setup of the control model. The SM is equipped with a gyro for rate damping and a resolver for command angle tracking. A torque actuator is located between the SM and the BM and applies control torques to move the SM to the commanded position. It is important to note that when a control torque is applied to the SM, an equal and opposite torque is applied to the BM, which in turn excites translational and rotational modes throughout the structure. Thus it is important to filter the commanded control torque to avoid exciting the modes of the structure. This section will focus first on outlining the error models associated with the measurements and actuator. This is followed by a detailed controller design and a state-space realization of the controller.

5.3.1 Truth Model

The feedback used in the control law consists of two parts. The first is the inertial angular rate of the SM (\(\dot{\psi}\)) as sensed by a gyro attached to the SM. The output of this gyro is corrupted with unknown and unestimated bias, scale factor, and noise.

\[
\tilde{\omega}_{\text{gyro}2} = (1 + s_{\text{gyro}2}) \left( \dot{\psi} + b_{\text{gyro}2} + \eta_{\text{gyro}2} \right)
\]

(5.72)

\[
E[\eta_{\text{gyro}2}(t)] = 0, \quad E[\eta_{\text{gyro}2}(t) \eta_{\text{gyro}2}(t')] = \sigma_{\text{gyro}2}^2 \delta(t - t')
\]

(5.73)

The second controller feedback measurement is a combination of the inertial angle of the SM (\(\dot{\psi}\)) as estimated by the navigation filter (\(\hat{\lambda}\)) and a resolver which measures the relative angle between the SM and BM (\(\tilde{\rho}\)).

\[
\hat{\psi} = \hat{\lambda} + \tilde{\rho} = \hat{\lambda} + \rho + b_{\text{resolver}} + \eta_{\text{resolver}}
\]

(5.74)
where the measurement model incorrectly assumes that the compliance between the RM and BM is zero ($\phi - \lambda = 0$) and where the true relative angle between the SM and CM is

$$\rho = \psi - \phi$$  \hfill (5.76)

The actuator that drives the SM is a brushless DC motor which applies actuation torques to the SM and reaction torques to the BM. The DC motor is modeled simply as a torque actuator, corrupted by scale factor and bias errors, with no back electro-motive force (EMF).

$$T_m = (1 + s_{\text{torque}}) (T_{mc} + b_{\text{torque}})$$  \hfill (5.77)

The lack of back EMF is justified for two reasons: 1) the objective of the control law is to maintain the inertial angular velocity of the SM very near zero, and 2) the angular rate of the host vehicle is also very near zero. Thus the relative angular rate between the SM and BM will be very small and results in negligible back EMF.

The dynamics of the parameter states are modeled as first-order Markov processes

$$\dot{b}_{\text{resolver}} = -\frac{b_{\text{resolver}}}{\tau_{\text{resolver}}} + w_{\text{resolver}}$$  \hfill (5.78)

$$\dot{b}_{\text{gyro}} = -\frac{b_{\text{gyro}}}{\tau_{\text{gyro}}} + w_{\text{gyro}}$$  \hfill (5.79)

$$\dot{s}_{\text{gyro}} = -\frac{s_{\text{gyro}}}{\tau_{\text{gyro}}} + w_{\text{gyro}}$$  \hfill (5.80)

$$\dot{b}_{\text{torque}} = -\frac{b_{\text{torque}}}{\tau_{\text{torque}}} + w_{\text{torque}}$$  \hfill (5.81)

$$\dot{s}_{\text{torque}} = -\frac{s_{\text{torque}}}{\tau_{\text{torque}}} + w_{\text{torque}}$$  \hfill (5.82)

$$E[w_{\text{resolver}}(t)] = 0, \quad E[w_{\text{resolver}}(t)w_{\text{resolver}}(t')] = \frac{2\sigma^2_{\text{b resolver, ss}}}{\tau_{\text{resolver}}} \delta(t - t')$$  \hfill (5.83)

$$E[w_{\text{gyro}}(t)] = 0, \quad E[w_{\text{gyro}}(t)w_{\text{gyro}}(t')] = \frac{2\sigma^2_{\text{b gyro, ss}}}{\tau_{\text{gyro}}} \delta(t - t')$$  \hfill (5.84)
\[ E[w_{\text{sgyro}}(t)] = 0, \quad E[w_{\text{sgyro}}(t)w_{\text{sgyro}}(t')] = \frac{2\sigma_{\text{sgyro},ss}^2}{\tau_{\text{sgyro}}} \delta(t - t') \quad (5.85) \]

\[ E[w_{\text{btorque}}(t)] = 0, \quad E[w_{\text{btorque}}(t)w_{\text{btorque}}(t')] = \frac{2\sigma_{\text{btorque},ss}^2}{\tau_{\text{btorque}}} \delta(t - t') \quad (5.86) \]

\[ E[w_{\text{storque}}(t)] = 0, \quad E[w_{\text{storque}}(t)w_{\text{storque}}(t')] = \frac{2\sigma_{\text{storque},ss}^2}{\tau_{\text{storque}}} \delta(t - t') \quad (5.87) \]

where \( \sigma_{\text{resovler},ss}^2, \sigma_{\text{bgyro},ss}^2, \sigma_{\text{sgyro},ss}^2, \sigma_{\text{btorque},ss}^2, \) and \( \sigma_{\text{storque},ss}^2 \) are the steady-state variance of the resolver bias, gyro 2 bias, gyro 2 scale factor, actuator bias, and actuator scale factor, respectively. The parameters \( \tau_{\text{resolvler}}, \tau_{\text{bgyro}}, \tau_{\text{sgyro}}, \tau_{\text{btorque}}, \) and \( \tau_{\text{storque}} \) are the corresponding ECRV time constants.

Note that there are several unknown and unestimated error sources that influence the output of the controller. Tracking the effect of these error sources on the pointing performance of the control algorithm is the purpose behind the Monte Carlo and LinCov analyses. The purpose of this section, however, is to derive the equations and calculate the gain values for the SM pointing controller. The error sources therefore will be neglected throughout the remainder of this section.

### 5.3.2 Controller and Anti-Resonance Filter Design

Figure 5.9 shows a block diagram of the SM control law and comprises several components. Moving from left to right through the diagram, \( \psi^* \) is the desired orientation of the SM. The next block is the outer-loop attitude controller which has two gains, a proportional gain \( K_P \) and an integral gain \( K_I \). The output of this controller is the angular rate command to the inner-loop angular rate controller which has a single proportional gain \( K_V \). The inner-loop controller produces the output \( u_{\text{prefilt}} \), which is then filtered by the anti-resonance filter whose transfer function is expressed as \( H_f(s) \). The output of the filter is the command torque to the SM actuator, \( T_{mc} \). The response of the SM is modeled with the transfer function \( G_{SM}(s) \), resulting in the quantities used for control, namely, the inertial angle and angular rate, \( \psi \) and \( \dot{\psi} \), respectively.

The SM transfer function, \( G_{sm}(s) \), is used for designing the gains of the inner-loop angular rate controller and the anti-resonance filter. Figure 5.10 shows the frequency response of \( G_{SM}(s) \), whose parameters are listed later in this dissertation (Table 6.1).

Figure 5.10 shows structural resonances at approximately 3Hz, 70Hz, and 160Hz. For the inner loop controller, it is desired to place the gain cross-over frequency around 8 Hz. However,
Figure 5.9: SM attitude control block diagram

Figure 5.10: SM transfer function.
before designing the inner loop controller, and to prevent the controller from exciting the upper two structural resonances, an anti-resonance filter, $H_f(s)$, is designed. The state-space representation of $H_f(s)$ is specified as

$$\dot{x}_f = A_f x_f + B_f u_{prefilt} \quad (5.88)$$

$$T_{mc} = u = C_f x_f + D_f u_{prefilt} \quad (5.89)$$

where the numerical values for the coefficients were chosen for a fourth-order, Chebyshev Type II low-pass filter, with a cut-off frequency of 61Hz and greater than 15 dB suppression

$$A_f = \begin{bmatrix} -814.98 & -513.85 & 0 & 0 \\ 513.85 & 0 & 0 & 0 \\ -814.98 & 1438.26 & -148.65 & -340.98 \\ 0 & 0 & 340.98 & 0 \end{bmatrix} \quad (5.90)$$

$$B_f = \begin{bmatrix} 383.27 \\ 0 \\ 383.27 \\ 0 \end{bmatrix} \quad (5.91)$$

$$C_f = \begin{bmatrix} -0.37813 & 0.66731 & -0.06896 & 0.07597 \end{bmatrix} \quad (5.92)$$

$$D_f = [0.17783] \quad (5.93)$$

The effect of the anti-resonance filter is illustrated in Figure 5.11. Note that the upper two resonances have been eliminated from the perspective of the inner loop controller.
Figure 5.11: Frequency response of the plant $G_{SM}(s)$ in blue, anti-resonance filter $H_f(s)$ in green, and modified plant $H_f(s) \cdot G_{SM}(s)$ in red.
Recall that the transfer function of the inner loop controller is simply

\[ C_{\text{inner}}(s) = K_V \quad (5.94) \]

whose gain is chosen to be

\[ K_V = 40.74 \frac{Nm}{\text{rad/sec}} \quad (5.95) \]

Combining the controller transfer function \( C_{\text{inner}}(s) \) with the modified plant \( H_f(s) \cdot G_{SM}(s) \) yields the open-loop transfer function (OLTF) for the inner control loop.

\[ OLTF_{\text{inner}}(s) = C_{\text{inner}}(s) \cdot H_f(s) \cdot G_{SM}(s) \quad (5.96) \]

The frequency response in Figure 5.12 shows a phase margin of 77 deg at 8Hz and a gain margin of 14.8dB at 43Hz. The corresponding CLTF is defined with the following equation.

\[ CLTF_{\text{inner}}(s) = \frac{OLTF_{\text{inner}}(s)}{1 + OLTF_{\text{inner}}(s)} \quad (5.97) \]

Figure 5.13 illustrates the frequency response of the closed-loop transfer function (CLTF) with a -3dB bandwidth of 11Hz.

The plant for the outer control loop is defined as the integral of the inner loop CLTF. The frequency response is illustrated in Figure 5.14.

\[ G_{\text{outer}}(s) = CLTF_{\text{inner}}(s) \cdot \frac{1}{s} \quad (5.98) \]

Recall that the transfer function of the outer controller is

\[ C_{\text{outer}}(s) = K_P + \frac{K_I}{s} \quad (5.99) \]

where the gains are chosen to be

\[ K_P = 19.50 \frac{1}{\text{sec}} \quad (5.100) \]

\[ K_I = 36.75 \frac{1}{\text{sec}^2} \quad (5.101) \]

The frequency response of the outer controller is illustrated in Figure 5.15.
Figure 5.12: Inner OLT frequency response. Phase margin = 77 deg at 8Hz. Gain margin = 14.8dB at 43Hz.
Figure 5.13: Inner CLTF frequency response with a -3dB bandwidth = 11Hz.
As in the case of the inner loop, the OLTF of the outer control loop is created by combining the controller transfer function with the plant

\[
OLTF_{outer} = C_{outer}(s) \cdot G_{outer}(s)
\] (5.102)

Figure 5.16 illustrates the frequency response of the outer-loop OLTF, showing a phase margin of 63 degrees at 3 Hz and a gain margin of 20.1 dB at 17 Hz. The outer CLTF is defined in the same manner as the inner loop

\[
CLTF_{outer} = \frac{OLTF_{outer}}{1 + OLT_{outer}}
\] (5.103)

The frequency response is illustrated in Figure 5.17 and shows a -3dB bandwidth of 5Hz.

5.3.3 State-Space Realization

In order to numerically integrate the controller dynamics in the Monte Carlo and LinCov simulations, the control law must be formulated as a state-space model. A procedure described in [37] on pages 713 to 715 converts a transfer function in the Laplace domain to the controllable canonical state-space form.

The inner loop controller transfer function is expressed as the output over the input

\[
G_{inner}(s) = \frac{U_{prefilt}(s)}{E_v(s)} = K_v
\] (5.104)

Once in the form of equation 5.104, the procedure yields the following for the inner loop controller transfer function.

\[
u_{prefilt} = K_v e_v
\] (5.105)

Finally, expanding the error term \(e_v\) yields

\[
u_{prefilt} = K_v \left[ \dot{\psi}^* - \dot{\omega}_{gyro2} \right]
\] (5.106)
Figure 5.14: Frequency response of the plant for the outer control loop.
Figure 5.15: Frequency response of the outer loop controller.
Figure 5.16: Frequency response of the outer OLT. Phase margin = 63 deg at 3 Hz. Gain margin = 20.1 dB at 17 Hz.
Figure 5.17: Frequency response of the outer CLTF with a -3dB bandwidth = 5Hz.
Following the same procedure as for the inner loop, the outer loop controller transfer function is also expressed as the output over the input

\[
G_{outer}(s) = \frac{\dot{\Psi}^*(s)}{E_p(s)} = K_P + \frac{K_I}{s} = \frac{K_PS + K_I}{s} \tag{5.107}
\]

And the corresponding state space system in controllable canonical form is

\[
\dot{x}_{outer} = e_p \tag{5.108}
\]

\[
\dot{\Psi}^* = (K_I)x_{outer} + (K_P)e_p \tag{5.109}
\]

As before, expanding the error term yields

\[
\dot{x}_{outer} = \psi^* - \hat{\psi} = \psi^* - \hat{\lambda} - \tilde{\rho}_k \tag{5.110}
\]

\[
\dot{\Psi}^* = (K_I)x_{outer} + (K_P)\left(\psi^* - \hat{\lambda} - \tilde{\rho}_k\right) \tag{5.111}
\]

where \(\hat{\psi}\) has been replaced with \(\hat{\lambda} + \tilde{\rho}_k\), i.e. the estimate of the RM attitude and the resolver measurement.

The rate and output equations for the inner and outer loop can be combined by eliminating the variable \(\dot{\psi}^*\). This is done by substituting equation 5.111 into equation 5.106, yielding the following system.

\[
\dot{x}_{outer} = \psi^* - \hat{\psi} \tag{5.112}
\]

\[
u_{prefilt} = K_v \left[K_I x_{outer} + K_P \left(\psi^* - \hat{\lambda} - \tilde{\rho}_k\right) - \tilde{\omega}_{gyro2}\right] \tag{5.113}
\]

The final piece of the controller to include is the anti-resonance filter. This is accomplished by substituting equation 5.113 into equations 5.88 and 5.89. This yields the final form of the control law which consists of five states with corresponding propagation and output equations.

\[
\dot{x}_{outer} = \psi^* - \hat{\psi} \tag{5.114}
\]
\[ \dot{x}_f = A_f x_f + B_f K_v \left[ K_I x_{\text{outer}} + K_P \left( \psi^* - \hat{\lambda} - \hat{\rho}_k \right) - \tilde{\omega}_{\text{gyro}2} \right] \]

\[ T_{\text{mc}} = C_f x_f + D_f K_v \left[ K_I x_{\text{outer}} + K_P \left( \psi^* - \hat{\lambda} - \hat{\rho}_k \right) - \tilde{\omega}_{\text{gyro}2} \right] \]

In summary, the SM pointing controller consists of an inner rate-damping control loop and an outer positioning loop, both of which were designed with adequate gain and phase margins. The inner loop uses the uncompensated measurements from gyro 2 for rate-damping. Because it neglects the bias and scale factor, these error states will directly impact the stabilization of the SM. The information available to the outer loop for pointing control are the estimated RM orientation and the resolver. Due to the unmeasured compliance between the BM and RM, this makes the pointing controller susceptible to rotational vibrations in the system. To prevent the actuator from exciting these structural resonances, the output of the inner loop is low-pass filtered. The Laplace transform of the controllers were converted to a state space formulation consistent with the form derived in equations 4.17 and 4.18.
Chapter 6

Simulation Overview

Chapter 5 outlined in detail the elements of the structural dynamics model, navigation algorithm, and pointing control algorithm. It is the purpose of this chapter to consolidate all the equations into a single location and present them in a form that is compatible with the equations developed in Chapter 4. Section 6.1 focuses on defining all equations needed for the Monte Carlo simulation. Section 6.2 presents the matrices corresponding to the linearization of the Monte Carlo equations, thus defining the elements of the Linear Covariance simulation. Once the layout of the two simulations is constructed, numerical values for all simulation parameters will be summarized in section 6.3, followed by the initial conditions of the states in section 6.4.

6.1 Monte Carlo

This section mirrors as much as possible the structure of section 4.2, systematically defining all variables needed for the Monte Carlo simulation. Equation 4.1 defines the propagation of the truth states as a function of controller output and process noise. The truth state vector of equation 4.1 for this problem is

\[
x = \begin{bmatrix} x_g^T & x_d^T & x_p^T \end{bmatrix}^T
\]  

(6.1)

where the gimbal states, \(x_g\), are defined in equation 5.30, the disturbance states, \(x_d\), are defined in equation 5.31 and the parameter states \(x_p\) are defined by

\[
x_p = \begin{bmatrix} b_{star} & b_{resolver} & b_{gyro1} & s_{gyro1} & b_{gyro2} & s_{gyro2} & b_{torque} & s_{torque} \end{bmatrix}^T
\]  

(6.2)
The dynamics of the truth state vector in equation 4.1 are defined with the following

\[
\begin{bmatrix}
A_g x_g + B_{g1} x_d + B_{g2} T_m \\
A_d x_d \\
A_p x_p
\end{bmatrix}
\]

\[
f(\mathbf{x}, \hat{u}) = \begin{bmatrix}
A_g x_g + B_{g1} x_d + B_{g2} T_m \\
A_d x_d \\
A_p x_p
\end{bmatrix}
\]  \hspace{1cm} (6.3)

\[
\begin{bmatrix}
0_{16 \times 1} \\
w_d \\
w_p
\end{bmatrix}
\]

\[
w = \begin{bmatrix}
0_{16 \times 1} \\
w_d \\
w_p
\end{bmatrix}
\]  \hspace{1cm} (6.4)

\[
B = \begin{bmatrix}
0_{16 \times 3} & 0_{16 \times 8} \\
I_{3 \times 3} & 0_{3 \times 8} \\
0_{8 \times 3} & I_{8 \times 8}
\end{bmatrix}
\]  \hspace{1cm} (6.5)

where the disturbance and parameter coefficient and noise vectors are defined as

\[
A_p = \text{diag}
\begin{bmatrix}
-1/T_{\text{bstar}} \\
-1/T_{\text{brresolver}} \\
-1/T_{\text{bgyro1}} \\
-1/T_{\text{sgyro1}} \\
-1/T_{\text{bgyro2}} \\
-1/T_{\text{sgyro2}} \\
-1/T_{\text{btorque}} \\
-1/T_{\text{storage}}
\end{bmatrix}
\]  \hspace{1cm} (6.6)
The continuous inertial measurements of equation 4.4 for this problem consist of the gyro 1 measurements

\[ \tilde{y} = \tilde{\omega}_{gyro1} \quad (6.10) \]

with the following definitions

\[ c(x) = \dot{\lambda} \quad (6.11) \]
\[ \eta = \eta_{gyro1} \quad (6.12) \]

The continuous controller feedback measurements of equation 4.6 are

\[ \tilde{y}_c = \begin{bmatrix} \tilde{\omega}_{gyro2} \\ \tilde{\rho} \end{bmatrix} \quad (6.13) \]

with

\[ q(x) = \begin{bmatrix} (1 + s_{gyro2}) \left( \dot{\psi} + b_{gyro2} \right) \\ \psi - \phi + b_{resolver} \end{bmatrix} \quad (6.14) \]

\[ \eta_c = \begin{bmatrix} \eta_{gyro2} \\ \eta_{resolver} \end{bmatrix} \quad (6.15) \]

The discrete Kalman filter measurements of equation 4.8 are

\[ \tilde{z}_k = \tilde{\lambda}_k \quad (6.16) \]

\[ h(x_k) = \lambda_k + b_{star,k} \quad (6.17) \]
\[ \nu_k = \eta_{\text{star}, k} \quad (6.18) \]

The navigation algorithm of equation 4.10 is defined as

\[
\dot{x} = \begin{bmatrix} \dot{\lambda} & \dot{b}_{\text{star}} & \dot{b}_{\text{gyro}} & \dot{s}_{\text{gyro}} \end{bmatrix}^T \quad (6.19)
\]

\[
\hat{f} (\hat{x}, \hat{y}) = \begin{bmatrix}
\frac{\ddot{\omega}_{\text{gyro}}}{(1 + s_{\text{gyro}})} - \dot{b}_{\text{gyro}} \\
-\frac{\dot{b}_{\text{star}}}{\tau_{\text{star}}} \\
-\frac{\dot{b}_{\text{gyro}}}{\tau_{\text{gyro}}} \\
-\frac{s_{\text{gyro}}}{\tau_{\text{gyro}}}
\end{bmatrix} \quad (6.20)
\]

The estimated discrete Kalman filter measurement of equation 4.12 is

\[ \hat{z}_k = \hat{h} (\hat{x}_k) = \lambda_k + \dot{b}_{\text{star}, k} \quad (6.21) \]

The Kalman gain and covariance propagation of equations 4.13-4.15 are

\[
\hat{K}_k = \hat{P}_k^{-1} \hat{H}_k \left( \hat{H}_k \hat{P}_k^{-1} \hat{H}_k^T + \hat{R}_\nu \right)^{-1} \quad (6.22)
\]

\[
\hat{P} = \hat{F} \hat{P} + \hat{P} \hat{F}^T + \hat{B} \hat{Q} \hat{B}^T \quad (6.23)
\]

\[
\hat{P}_k^- = \left( I - \hat{K}_k \hat{H}_k \right) \hat{P}_k^- \left( I - \hat{K}_k \hat{H}_k \right)^T + \hat{K}_k \hat{R}_\nu \hat{K}_k^T \quad (6.24)
\]

where

\[
\hat{H}_k = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \quad (6.25)
\]

\[
\hat{F} = \begin{bmatrix} 0 & 0 & -1 & -\frac{\ddot{\omega}_{\text{gyro}}}{(1 + s_{\text{gyro}})^2} \\
0 & -\frac{1}{\tau_{\text{star}}} & 0 & 0 \\
0 & 0 & -\frac{1}{\tau_{\text{gyro}}} & 0 \\
0 & 0 & 0 & -\frac{1}{\tau_{\text{gyro}}}
\end{bmatrix} \quad (6.26)
\]
\[
\hat{Q} = \begin{bmatrix}
\sigma_{gyrol}^2 & 0 & 0 & 0 \\
0 & \frac{2\sigma_{star,ss}^2}{\tau_{star}} & 0 & 0 \\
0 & 0 & \frac{2\sigma_{bgyrol,ss}^2}{\tau_{bgyrol}} & 0 \\
0 & 0 & 0 & \frac{2\sigma_{sgyro,ss}^2}{\tau_{sgyro}} \\
\end{bmatrix}
\] (6.27)

\[
\hat{R}_\nu = \sigma_{star}^2
\] (6.28)

\[
\hat{B} = \begin{bmatrix}
-1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\] (6.29)

The mapping function from the truth states to the navigation states of equation 4.16 is

\[
m(x) = \begin{bmatrix}
\lambda \\
b_{star} \\
b_{gyrol} \\
s_{gyrol} \\
\end{bmatrix}^T
\] (6.30)

The controller states with their corresponding dynamics and output equation defined in equations 4.17 and 4.18 are

\[
\mathbf{x} = \begin{bmatrix}
x_{outer} \\
x_f \\
\end{bmatrix}
\] (6.31)

\[
\mathbf{f} (\mathbf{x}, \hat{x}, \tilde{y}_c) = \begin{bmatrix}
\psi^* - \hat{\psi} \\
A_f x_f + B_f K_v \left\{ K_f x_{outer} + K_P \left( \psi^* - \hat{\lambda} - \tilde{\rho}_k \right) - \tilde{\omega}_{gyro2} \right\} \\
\end{bmatrix}
\] (6.32)

\[
\mathbf{u} = T_{mc}
\] (6.33)

\[
\mathbf{g} (\mathbf{x}, \hat{x}, \tilde{y}_c) = C_f x_f + D_f K_v \left\{ K_f x_{outer} + K_P \left( \psi^* - \hat{\lambda} - \tilde{\rho}_k \right) - \tilde{\omega}_{gyro2} \right\}
\] (6.34)

And finally, the true actuator value of equation 4.19 is
\[ T_m = (1 + s_{torque}) (T_{mc} + b_{torque}) \] (6.35)

To give further insight into the system, it is often times useful to define particularly important elements or combinations of the state vector, named here as the *performance state vector*. In this analysis, the performance state vector includes the relative position of the different structural members as well as the applied actuator torque. The relative positions and orientations are defined with the central/reference body being the RM. Thus the orientation of the SC and BM are computed relative to the RM. And the position and orientation of the SM is computed relative to BM. The applied actuator torque is computed using the definition in equation 6.35. These definitions correspond to the following performance dispersion vector.
\[ \Pi = \begin{bmatrix} 
\delta x - \delta x_1 \\
\delta x_1 \\
\delta x_2 - \delta x_1 \\
\delta x_3 - \delta x_2 \\
\delta \theta - \delta \lambda \\
\delta \lambda \\
\delta \phi - \delta \lambda \\
\delta \psi - \delta \phi \\
\delta \dot{x} - \delta \dot{x}_1 \\
\delta \dot{x}_1 \\
\delta \dot{x}_2 - \delta \dot{x}_1 \\
\delta \dot{x}_3 - \delta \dot{x}_2 \\
\delta \dot{\theta} - \delta \dot{\lambda} \\
\delta \dot{\lambda} \\
\delta \dot{\phi} - \delta \dot{\lambda} \\
\delta \dot{\psi} - \delta \dot{\phi} \\
\delta \dot{\psi} \\
\delta T_m 
\end{bmatrix} \]
which can be expressed in matrix form as function of the augmented state vector and continuous controller feedback measurements

$$\Pi = A_\Pi X + B_\Pi \eta_c$$  \hspace{1cm} (6.37)$$

where $A_\Pi$ and $B_\Pi$ are defined as

$$B_\Pi = \begin{bmatrix} 0_{18 \times 2} \\ G_u \hat{G}_y \end{bmatrix}$$  \hspace{1cm} (6.38)$$

$$A_\Pi = \begin{bmatrix} a_\Pi & 0_{18 \times 4} & 0_{18 \times 5} \\ G_u \hat{G}_y \hat{Q}_x + G_x & G_u \hat{G}_x & G_u \hat{G}_x \hat{Q}_c \\ a_\Pi & b_\Pi & 0_{18 \times 11} \end{bmatrix}$$  \hspace{1cm} (6.39)$$

$$a_\Pi = \begin{bmatrix} b_\Pi & 0_{18 \times 11} \end{bmatrix}$$  \hspace{1cm} (6.40)$$
In section 4.5, an augmented state vector was formed and the covariance propagation and update equations were outlined. The coefficients of the equations consisted of several partial deriv
tives of the nonlinear Monte Carlo equations. The purpose of this section is to define all of the partial derivatives needed for the specific problem outlined in the previous section.

The partials related to the truth state dynamics are

\[
F_u = \begin{bmatrix}
B_g2 (1 + s_{torque}) \\
0_{11 \times 1}
\end{bmatrix}_{\text{nominal}}
\]  
(6.42)

\[
F_x = \begin{bmatrix}
A_g & B_g1 & B_g2 \frac{\partial T_m}{\partial x_p} \\
0_{3 \times 16} & A_d & 0_{3 \times 8} \\
0_{8 \times 16} & 0_{8 \times 3} & A_p
\end{bmatrix}_{\text{nominal}}
\]  
(6.43)

where

\[
\frac{\partial T_m}{\partial x_p} = \begin{bmatrix}
0_{1 \times 6} & (1 + s_{torque}) (T_{mc} + b_{torque})
\end{bmatrix}
\]  
(6.44)

The partials related to the output of the control law are

\[
\hat{G}_y = \left[ -D_f K_v -D_f K_v K_p \right]_{\text{nominal}}
\]  
(6.45)

\[
\hat{G}_x = \left[ -D_f K_v K_P 0_{1 \times 3} \right]_{\text{nominal}}
\]  
(6.46)

\[
\hat{G}_x = \left[ D_f K_v K_f C_f \right]_{\text{nominal}}
\]  
(6.47)

The partial of the continuous controller feedback measurement is

\[
Q_x = \begin{bmatrix}
\frac{\partial \hat{\omega}_{gyro2}}{\partial x} \\
\frac{\partial \hat{\psi}}{\partial x}
\end{bmatrix}_{\text{nominal}}
\]  
(6.48)

where

\[
\frac{\partial \hat{\omega}_{gyro2}}{\partial x} = \begin{bmatrix}
0_{1 \times 15} & (1 + s_{gyro2}) & 0_{1 \times 7} & (1 + s_{gyro2}) & \left( \dot{\psi} + b_{gyro2} \right) & 0_{1 \times 2}
\end{bmatrix}
\]  
(6.49)
\[
\frac{\partial \hat{\rho}}{\partial x} = \begin{bmatrix}
0_{1 \times 6} & -1 & 1 & 0_{1 \times 12} & 1 & 0_{1 \times 6}
\end{bmatrix}
\tag{6.50}
\]

The partial of the continuous inertial measurements is

\[
C_x = \begin{bmatrix}
0_{1 \times 13} & (1 + s_{gyro1}) & 0_{1 \times 7} & (1 + s_{gyro1}) & \left( \dot{\lambda} + b_{gyro1} \right) & 0_{1 \times 4}
\end{bmatrix}
\bigg|_{\text{nominal}}
\tag{6.51}
\]

The partials related to the control law dynamics are

\[
\hat{F}_c^y = \begin{bmatrix}
0 & -1 \\
-B_f K_v & -B_f K_v K_P
\end{bmatrix}
\bigg|_{\text{nominal}}
\tag{6.52}
\]

\[
\hat{F}_x = \begin{bmatrix}
-1 & 0_{1 \times 3} \\
-B_f K_v K_P & 0_{4 \times 3}
\end{bmatrix}
\bigg|_{\text{nominal}}
\tag{6.53}
\]

\[
\hat{F}_x = \begin{bmatrix}
0 & 0_{1 \times 4} \\
B_f K_v K_I & A_f
\end{bmatrix}
\bigg|_{\text{nominal}}
\tag{6.54}
\]

The partials related to the navigation state propagation are

\[
\hat{F}_y = \begin{bmatrix}
\frac{1}{1 + s_{gyro1}} \\
0_{3 \times 1}
\end{bmatrix}
\bigg|_{\text{nominal}}
\tag{6.55}
\]

\[
\hat{F}_x = \begin{bmatrix}
0 & 0 & -1 & -\frac{\dot{\omega}_{gyro1}}{(1 + s_{gyro1})^2} \\
0 & \frac{1}{\tau_{star}} & 0 & 0 \\
0 & 0 & -\frac{1}{\tau_{gyro1}} & 0 \\
0 & 0 & 0 & -\frac{1}{\tau_{gyro1}}
\end{bmatrix}
\bigg|_{\text{nominal}}
\tag{6.56}
\]
The partials related to the true actuator value are

$$G_u = (1 + s_{torque})|_{nominal}$$  \hfill (6.57)

$$G_x = \begin{bmatrix} 0_{1 \times 25} & (1 + s_{torque}) & (T_{mc} + b_{torque}) \end{bmatrix}|_{nominal}$$  \hfill (6.58)

The partial of the discrete Kalman filter measurement is

$$H_k = \begin{bmatrix} 0_{1 \times 5} & 1 & 0_{1 \times 13} & 1 & 0_{1 \times 7} \end{bmatrix}$$  \hfill (6.59)

To facilitate computing the performance state covariance matrix, the augmented state covariance can be transformed according to

$$C_\Pi = A_\Pi C_A A_\Pi^T + B_\Pi S_\eta B_\Pi^T$$  \hfill (6.60)

### 6.3 Simulation Parameters Summary

With the exception of the anti-resonance filter design and control gain selection, the equations up to this point have been purely symbolic with no numerical values. This section defines all numerical values for the structural dynamics model, noise strengths, and time constants. The reader is referred to section 5.3 for the anti-resonance filter coefficients and controller gains. The simulation parameters listed are organized into two types, state dynamics coefficients and noise strengths. The first set of coefficients are those that are related to the gimbal dynamics discussed in section 5.1, namely $A_g$, $B_{g1}$, and $B_{g2}$. The values needed for these coefficient matrices are listed in Table 6.1. Several parameters are worthy of discussion. First is the relatively large values for the spacecraft mass and inertia, $M$ and $J$. Though much bigger than real systems, these values enable enforcement of angular and translational acceleration profiles, without being affected by the motion of the pointing system bodies. The ability to enforce an angular and translational acceleration profile is important when the disturbance environment is specified by a host vehicle PSD, which is the case for this research. Also noteworthy are the values defining the rotational stiffness and damping between the SM and the BM, $K_3$ and $C_3$, which are consistent with a bearing or flexure interface. In practice, the offset of the SM center of mass can be controlled to tighter tolerances than 10 cm, but this relatively large value was chosen to increase the coupling between linear and
rotational dynamics. Finally the remaining mass, inertia, stiffness, and damping values were chosen to be representative of a physical system and to yield structural damping ratios near 1%, typical of real structures.

The remaining coefficient matrices are those related to the disturbance states, \( x_d \), and the parameter states, \( x_p \). The parameters needed to define the corresponding coefficient matrices, \( A_d \) and \( A_p \), are shown in tables 6.2 and 6.3. The parameter state time constants were chosen such that all of the parameters were very stable. The disturbance state time constants however were chosen to be representative of a noisy satellite environment, with sufficient bandwidth to excite the modes of the structure and cause significant pointing dispersions.

Table 6.4 provides the noise strengths for the disturbances that drive the state differential equations. All of the noise values listed are the steady-state values that the states will reach after a few time constants. As was the case for the time constants, the steady-state values for the disturbance states were chosen to be representative of a noisy satellite environment. Next, Table 6.5 lists the values that define the performance of the sensors and actuators. This table lists the steady-state standard deviation of the star tracker, gyros, resolver, and actuator parameters. Finally, Table 6.6 lists the continuous measurement noise of the gyros and resolver and the discrete measurement noise of the star tracker. The parameter and measurement noise values were chosen to be representative of a current navigation-grade star tracker, resolver, and gyro.

The simulation contains five key times instants, \( t_{on} \), \( t_{star} \), \( t_{step} \), \( t_{image} \), and \( t_{end} \) which are illustrated in Figure 6.1. Prior to the time \( t_{on} \), the satellite is sitting with all power systems turned off and at some random orientation. Thus no disturbance sources are present and there is no relative motion between structural components. The significance of this time period will be discussed more in the following section on initialization. At time \( t_{on} \), the power system of the satellite turns on, and gyro 1 starts measuring changes in the orientation, assuming zero angle as the initial condition. During the time period from \( t_{on} \) to \( t_{star} \), the controller is inactive, but all disturbance sources are on. When \( t_{star} \) is reached, the star tracker produces its first measurement, which is processed by the Kalman filter, and the SM pointing controller is turned on, with a command angle of zero. Once \( t_{step} \) is reached, the satellite receives a command to slew to a target, in preparation for imaging. The command angle used for this simulation is 0.02 radians. Adequate time is given following \( t_{step} \) before the time for imaging the target, \( t_{image} \). The time \( t_{image} \) is a critical time for the satellite,
Table 6.1: Structural dynamics model parameters.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>7200</td>
<td>Nm/rad</td>
</tr>
<tr>
<td>$K_2$</td>
<td>$3.2 \times 10^6$</td>
<td>Nm/rad</td>
</tr>
<tr>
<td>$K_3$</td>
<td>0</td>
<td>Nm/rad</td>
</tr>
<tr>
<td>$C_1$</td>
<td>22</td>
<td>Nm/(rad/s)</td>
</tr>
<tr>
<td>$C_2$</td>
<td>18.3</td>
<td>Nm/(rad/s)</td>
</tr>
<tr>
<td>$C_3$</td>
<td>0.09</td>
<td>Nm/(rad/s)</td>
</tr>
<tr>
<td>$K_4$</td>
<td>$2.9 \times 10^4$</td>
<td>N/m</td>
</tr>
<tr>
<td>$K_5$</td>
<td>$2.8 \times 10^6$</td>
<td>N/m</td>
</tr>
<tr>
<td>$K_6$</td>
<td>$2.5 \times 10^6$</td>
<td>N/m</td>
</tr>
<tr>
<td>$C_4$</td>
<td>80</td>
<td>N/(m/s)</td>
</tr>
<tr>
<td>$C_5$</td>
<td>173</td>
<td>N/(m/s)</td>
</tr>
<tr>
<td>$C_6$</td>
<td>50</td>
<td>N/(m/s)</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>0.1</td>
<td>m</td>
</tr>
<tr>
<td>$M$</td>
<td>$1.0 \times 10^{12}$</td>
<td>kg</td>
</tr>
<tr>
<td>$m_1$</td>
<td>35</td>
<td>kg</td>
</tr>
<tr>
<td>$m_2$</td>
<td>23</td>
<td>kg</td>
</tr>
<tr>
<td>$m_3$</td>
<td>3</td>
<td>kg</td>
</tr>
<tr>
<td>$J$</td>
<td>$1.0 \times 10^{12}$</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$J_1$</td>
<td>15.6</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$J_2$</td>
<td>2.6</td>
<td>kg·m²</td>
</tr>
<tr>
<td>$J_3$</td>
<td>0.8</td>
<td>kg·m²</td>
</tr>
</tbody>
</table>

Table 6.2: Disturbance state time constants

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_{\text{trans}}$</td>
<td>14.5</td>
<td>ms</td>
</tr>
<tr>
<td>$\tau_{\text{rot}}$</td>
<td>0.5</td>
<td>s</td>
</tr>
<tr>
<td>$\tau_D$</td>
<td>20</td>
<td>s</td>
</tr>
</tbody>
</table>
Table 6.3: Parameter state time constants

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_{b\text{star}}$</td>
<td>28</td>
<td>hr</td>
</tr>
<tr>
<td>$\tau_{b\text{resolver}}$</td>
<td>28</td>
<td>hr</td>
</tr>
<tr>
<td>$\tau_{b\text{gyro}}$</td>
<td>28</td>
<td>hr</td>
</tr>
<tr>
<td>$\tau_{s\text{gyro}}$</td>
<td>28</td>
<td>hr</td>
</tr>
<tr>
<td>$\tau_{b\text{gyro}}$</td>
<td>28</td>
<td>hr</td>
</tr>
<tr>
<td>$\tau_{s\text{gyro}}$</td>
<td>28</td>
<td>hr</td>
</tr>
<tr>
<td>$\tau_{b\text{torque}}$</td>
<td>17</td>
<td>min</td>
</tr>
<tr>
<td>$\tau_{s\text{torque}}$</td>
<td>17</td>
<td>min</td>
</tr>
</tbody>
</table>

Table 6.4: Disturbance steady state standard deviation 1$\sigma$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{\text{tran,ss}}$</td>
<td>1.35</td>
<td>m/sec$^2$</td>
</tr>
<tr>
<td>$\sigma_{\text{rot,ss}}$</td>
<td>0.5</td>
<td>mrad/sec$^2$</td>
</tr>
<tr>
<td>$\sigma_{D,ss}$</td>
<td>0.17</td>
<td>mN-m</td>
</tr>
</tbody>
</table>

Table 6.5: Parameter steady-state standard deviation 1$\sigma$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{b\text{star,ss}}$</td>
<td>0.33</td>
<td>$\mu$rad</td>
</tr>
<tr>
<td>$\sigma_{b\text{resolver,ss}}$</td>
<td>0.67</td>
<td>$\mu$rad</td>
</tr>
<tr>
<td>$\sigma_{b\text{gyro1,ss}}$</td>
<td>0.005</td>
<td>deg/hr</td>
</tr>
<tr>
<td>$\sigma_{s\text{gyro1,ss}}$</td>
<td>1</td>
<td>ppm</td>
</tr>
<tr>
<td>$\sigma_{b\text{gyro2,ss}}$</td>
<td>0.005</td>
<td>deg/hr</td>
</tr>
<tr>
<td>$\sigma_{s\text{gyro2,ss}}$</td>
<td>1</td>
<td>ppm</td>
</tr>
<tr>
<td>$\sigma_{b\text{torque,ss}}$</td>
<td>1</td>
<td>mN-m</td>
</tr>
<tr>
<td>$\sigma_{s\text{torque,ss}}$</td>
<td>0.05</td>
<td>unit-less</td>
</tr>
</tbody>
</table>
as any motion during image acquisition blurs and otherwise degrades of the image. Thus for the
analysis in chapter 7, much of the focus will be on the error budget and spectrum of the data at
the image acquisition time. When $t_{sim}$ is reached, the simulation ends.

There are also three important time increments. The increment $dt$ is the step time of the
simulation and represents the rate at which the gyros measurements are available and the control law
executed. The increment $dt_{kalman}$ represents the output rate of the star camera and consequently
the update rate of the Kalman filter. The time increment $dt_{psd}$ represents the integration time of
the imager and specifies the time across which the PSD is calculated. The numerical values for the
times discussed are listed in Table 6.7, where the number of Monte Carlo runs and the confidence
interval for variance estimation are also listed.

### 6.4 Initial Conditions

It is important to properly initialize both the Monte Carlo and LinCov simulations. As
part of this initialization, one must consider the correlation between state uncertainties. It is also
important to consistently initialize the two simulations. Both of these points will be discussed as they
relate to the problem being analyzed. Since it is often more intuitive to think in terms of a Monte
Carlo simulation, the initial conditions for the Monte Carlo will be defined first. Subsequently, the
initial conditions for the LinCov simulation will be defined such that they are consistent with the
Monte Carlo.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{gyro1}$</td>
<td>0.005</td>
<td>deg/$\sqrt{\text{hr}}$</td>
</tr>
<tr>
<td>$\sigma_{gyro2}$</td>
<td>0.005</td>
<td>deg/$\sqrt{\text{hr}}$</td>
</tr>
<tr>
<td>$\sigma_{star}$</td>
<td>4.8</td>
<td>$\mu$rad</td>
</tr>
<tr>
<td>$\sigma_{resolver}$</td>
<td>5.0</td>
<td>$\mu$rad/$\sqrt{s}$</td>
</tr>
</tbody>
</table>

Table 6.6: Measurement noise 1$\sigma$
Table 6.7: Simulations times

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{on}$</td>
<td>0</td>
<td>s</td>
</tr>
<tr>
<td>$t_{star}$</td>
<td>2</td>
<td>s</td>
</tr>
<tr>
<td>$t_{step}$</td>
<td>4</td>
<td>s</td>
</tr>
<tr>
<td>$t_{image}$</td>
<td>30</td>
<td>s</td>
</tr>
<tr>
<td>$t_{sim}$</td>
<td>50</td>
<td>s</td>
</tr>
<tr>
<td>$dt$</td>
<td>0.25</td>
<td>ms</td>
</tr>
<tr>
<td>$dt_{kalman}$</td>
<td>0.1</td>
<td>s</td>
</tr>
<tr>
<td>$dt_{psd}$</td>
<td>19.8</td>
<td>s</td>
</tr>
<tr>
<td>$N_{mc}$</td>
<td>100</td>
<td>unit-less</td>
</tr>
<tr>
<td>$CI$</td>
<td>0.95</td>
<td>unit-less</td>
</tr>
</tbody>
</table>

Recall that before time $t_{on}$, the satellite is in a state of rest. Thus, the typical sources of vibration, such as cryo coolers, reaction wheels, or other mechanical systems were not producing significant disturbances. Hence the chosen initial conditions for the gimbal truth states is to be at complete rest with zero position, velocity, and angular velocity uncertainty, but with each member of the structure at the same initial random orientation. Thus the initial orientation is random, but the correlation between each component is perfect, or unity. This results in the following initial condition for the gimbal truth states $t_{on}$

$$E[\mathbf{x}_g(t_{on})] = \mathbf{0}$$ (6.61)

$$P_{g0} = E[\mathbf{x}_g(t_{on}) \mathbf{x}_g(t_{on})^T] = 
\begin{bmatrix}
0_{4\times4} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} \\
0_{4\times4} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4} \\
0_{4\times4} & 0_{4\times4} & P_{angle} & 0_{4\times4} \\
0_{4\times4} & 0_{4\times4} & 0_{4\times4} & 0_{4\times4}
\end{bmatrix}$$ (6.62)
where

\[
P_{\text{angle}} = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
\end{bmatrix} \sigma_{\text{ang}}^2
\]  

(6.63)

and the initial angular uncertainty is chosen to be \( \sigma_{\text{ang}}^2 = 1 \) deg

Because the disturbance sources were also not active prior to time \( t_{\text{on}} \), the corresponding initial condition is

\[
E[x_d(t_{\text{on}})] = 0
\]  

(6.64)

\[
P_{d0} = E[x_{d}(t_{\text{on}})x_{d}(t_{\text{on}})^T] = 0_{3 \times 3}
\]  

(6.65)

Note that due to the long time constant of the SM disturbance torques, steady state will not have been reached at the time of image acquisition. Thus the system is not truly at steady state, and the covariance vs. time plot will need to be analyzed to see if the statistics are sufficiently stationary for PSD estimation.

Since the parameter states are modeled as constants, they are all initialized with an uncertainty equal to their steady-state standard deviation

\[
E[x_p(t_{\text{on}})] = 0
\]  

(6.66)
As in many Kalman filters, the initial state estimate is zero for all states. This is accompanied by an initial estimate for the uncertainty that is guaranteed to be equal or larger than the true uncertainty

\[ P_{p0} = E \left[ x_p (t_{on}) x_p (t_{on})^T \right] = diag \begin{bmatrix} \sigma_{bstar,ss}^2 \\ \sigma_{bresolver,ss}^2 \\ \sigma_{bgyro1,ss}^2 \\ \sigma_{sgyro1,ss}^2 \\ \sigma_{bgyro2,ss}^2 \\ \sigma_{sgyro2,ss}^2 \\ \sigma_{btorque,ss}^2 \\ \sigma_{storque,ss}^2 \end{bmatrix} \] (6.67)

Finally the controller states are initialized at zero with zero uncertainty, recalling that the controller is not activated until \( t_{star} \)

\[ \dot{x} (t_{on}) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \ ] (6.68)

\[ \dot{P} (t_{on}) = diag \begin{bmatrix} \sigma_{ang}^2 & \sigma_{bstar,ss}^2 & \sigma_{bgyro1,ss}^2 & \sigma_{sgyro1,ss}^2 \end{bmatrix} \] (6.69)

Next, the LinCov simulation must be initialized in a manner consistent with the Monte Carlo simulations. Since for this problem, there exists no cross-correlation between the truth, navigation, and controller states, the initial augmented covariance matrix can be partitioned into mutually
uncorrelated truth, navigation, and controller state dispersion covariances

\[
C_A(t_{on}) = \begin{bmatrix}
C_{\delta \delta x} (t_{on}) & 0 & 0 \\
0 & C_{\delta \delta x} (t_{on}) & 0 \\
0 & 0 & C_{\delta \delta x} (t_{on})
\end{bmatrix}
\]  \hspace{1cm} (6.72)

where

\[
C_{\delta \delta x} = \begin{bmatrix}
P_{g0} & 0 & 0 \\
0 & P_{d0} & 0 \\
0 & 0 & P_{p0}
\end{bmatrix}
\]  \hspace{1cm} (6.73)

\[
C_{\delta \delta x} = 0_{4 \times 4}
\]  \hspace{1cm} (6.74)

\[
C_{\delta \delta x} (t_{on}) = C_{\delta \delta x} (t_{star}) = \hat{P} (t_0)
\]  \hspace{1cm} (6.75)

where it is again pointed out that the controller is not activated until the star camera is activated, so the dispersions remain zero until \(t_{star}\).
Chapter 7
Pointing Analysis

This chapter documents the analysis of the system described in chapter 6 and comprises four sections. Section 7.1 analyzes the trajectory that the system undergoes when no error sources are present. It describes both the inertial and relative motion of several structural members as well as the nominal torque required of the actuator. Section 7.2 compares the results of the LinCov simulation with the results of the Monte Carlo simulation including state dispersions, autocorrelation estimates, and PSD estimates. The autocorrelation and PSD estimates are analyzed for states whose statistics range from stationary, to approximately stationary, to non-stationary. Section 7.3 analyzes the performance of the navigation and control system. This analysis includes assessing observability of the navigation states and the optimality (or tuning) of the Kalman filter. This is followed by the sensitivity analysis of the navigation and pointing control system. The final section 7.4 describes a new analysis method which combines the benefits of sensitivity analysis and PSD estimation to identify not only the driving error sources, but the frequency range to which they contribute.

7.1 Mean Reference Trajectory

As defined in chapter 4, the nominal or mean reference trajectory is the response of the system when all error sources are zero. This section presents the trajectory of several states to give insight into the mean response of system and the meaning of the dispersions about the nominal. Figures 7.1 and 7.2 illustrate the nominal response of the inertial angle, angular rate, position, and velocity states. Several aspects are noteworthy. First is the period of no motion before $t_{step}$ observed in all states. This is followed by significant motion of all components except for the SC. The rotational motion of the BM and RM is very tightly coupled and dominated by a 3Hz component. The attitude of the SM however is driven to the set-point of 20 mrad and remains stable following the slew. The translational motions of the SM, RM, and BM are very tightly coupled and are also dominated by a 3Hz component. Finally it is observed that the attitude motion damps out before image acquisition time, $t_{image}$. 
Figure 7.1: Nominal attitude (top) and angular rates (bottom) of system components
Figure 7.2: Nominal inertial position (top) and velocity (bottom) of system components
In addition to looking at the inertial angle and position states, much intuition regarding the problem can be obtained from analyzing the relative motion of the gimbal components. This information is illustrated in figures 7.3 through 7.6. Figure 7.3 contains many features similar to those contained in the inertial plots, specifically a 3Hz ringing following $t_{\text{step}}$. Upon closer inspection of Figure 7.3 (shown in Figure 7.4), there exists a 60Hz component in the relative angular motion of the BM and RM, some of which is transferred to the relative motion of the SM and BM. The 3Hz, 60Hz, and even higher translational frequencies are present in the relative motion of the SM and BM as seen figures 7.5 and 7.6. All of the high frequencies, however, damp out quickly and are not expected to be present at significant levels during image acquisition.

Figure 7.7 shows the nominal torque profile, predicting a maximum torque required at the time of $t_{\text{step}}$ of about 15 N-m. When combined with the true torque dispersion discussed in section 7.2.1, this nominal torque profile is useful in predicting the probability of saturating the actuator.

7.2 Monte Carlo Vs. Linear Covariance

This section serves two purposes. First is to validate the accuracy of the LinCov simulation in calculating the covariance, autocorrelation function, and PSD of the dispersions from the nominal trajectory discussed in the previous section. The second purpose is to analyze trends in time history of the dispersions and the frequency content of the dispersion during the image acquisition.

7.2.1 Hair plots

An effective way of comparing the results of the Monte Carlo and LinCov simulation is the formation of “hair plots.” The hair plot visualizes the variance of the dispersion by showing each Monte Carlo sample of the random process. Superimposed on the Monte Carlo samples are the $3\sigma$ standard deviation estimates from the Monte Carlo simulation and the LinCov simulation. Because the Monte Carlo simulation estimates the variance from samples of the random process, it is also important to plot the confidence intervals of the Monte Carlo-estimated standard deviation. If the assumptions regarding linearization of the system are valid, then the LinCov estimate of the standard deviation will fall within the confidence intervals of the Monte Carlo-computed standard deviation, which is the case for all of the figures in this section.

Figures 7.8 through 7.10 contain the hair plots for the inertial angle and angular rate of the SC, RM, and SM, respectively. An important feature common to all the attitude dispersions is
Figure 7.3: Nominal relative attitude (top) and angular rates (bottom) of system components
Figure 7.4: Zoomed-in nominal relative attitude (top) and angular rate (bottom) of system components
Figure 7.5: Nominal relative position (top) and velocity (bottom) of system components
Figure 7.6: Zoomed-in nominal relative position (top) and velocity (bottom) of system components
Figure 7.7: Nominal actuator torque
the initial condition. As discussed in section 6.4, the initial attitude for all members is the same, and perfectly correlated. Since the attitude of the SC and RM are not controlled, the attitude dispersion uncertainty increases with time, as illustrated in Figure 7.8 and 7.9. The dispersion of the RM angular rate, however is significant affected by the activation of the SM control law. The control law has the effect of initially amplifying the angular rate uncertainty, then decreasing it to a quasi-steady state value. Note however that closer inspection of the RM angular rate dispersion shows that it is still increasing slowly even after the controller has settled for the same reason that the SC rate uncertainty is increasing, i.e. lack of attitude control.

Figure 7.10 shows the behavior of the SM inertial angle and angular rate dispersions. At $t_{on}$, the attitude dispersion is at an uncertainty specified by the initial conditions. Both the attitude and angular rate uncertainties grow slowly from $t_{on}$ to $t_{star}$, while the SM controller is inactive. At $t_{star}$, the SM controller begins to decrease the dispersions and reaches a steady-state value by the time the step command is received, $t_{step}$. The controller has the benefit of significantly reducing both the attitude and angular rate dispersions of the SM.

Figure 7.11 shows the inertial position and velocity dispersion of the RM and is representative of the position and velocity dispersions of all structural members. The initial uncertainty starts at zero for both position and velocity and, because of the lack of control, increases with time.

Figures 7.12 to 7.14 show the dispersions of the three disturbance sources, the translational and rotational acceleration on the SC followed by the disturbance torques on the SM. Recall that as discussed in section 6.4, all disturbance sources start at an initial condition of zero variance. The acceleration disturbance time constants are fast enough that both reach steady state well before $t_{image}$, the image acquisition time. The disturbances torques on the mirror, however, are slower and are still in the transient period during image acquisition.

Figure 7.15 shows the dispersion of one of the parameter states, the star camera bias. This plot is representative of all of the parameter states in that it is modeled as a bias and is initialized with a variance equation to the steady-state variance.

Figures 7.16 to 7.19 illustrate the variance of the estimation error of the Kalman filter on-board the RM. These plots are useful for assessing the observability of the four navigation states. The top plot of Figure 7.16 shows the initial large uncertainty of the RM attitude estimate before $t_{star}$, the time that the first star tracker measurement is processed. The bottom plot shows that
Figure 7.8: SC attitude (top) and angular rate (bottom) $3\sigma$ dispersion hair plots
Figure 7.9: RM attitude (top) and angular rate (bottom) 3σ dispersion hair plots
Figure 7.10: SM attitude (top) and angular rate (bottom) $3\sigma$ dispersion hair plot
Figure 7.11: RM position (top) and velocity (bottom) $3\sigma$ dispersion hair plot
Figure 7.12: Translational acceleration disturbance $3\sigma$ dispersion hair plot
Figure 7.13: Rotational acceleration disturbance $3\sigma$ dispersion hair plot
Figure 7.14: SM disturbance torque $3\sigma$ dispersion hair plot
Figure 7.15: Star camera bias $3\sigma$ dispersion hair plot
upon processing the star camera measurements, the Kalman filter rapidly converges to a steady-state variance. Figures 7.17 through 7.19 show that given the dynamics of the RM, which contain very low angular rates, the gyro bias and scale factor and the star camera bias are not observable.

As was the case for the mean reference trajectory analysis in the previous section, insight into the problem is gained by analyzing the relative states and their associated dispersions. Figure 7.20 illustrates the behavior of the relative position and velocity of the RM relative to the SC. Since the system starts at rest, the dispersions start at zero, followed by a short transient to the steady state value. The activation of the controller, navigation system, and step command to the controller do not significantly impact these dispersions. The relative position and velocity dispersions of structural members closer to the SM, however are more affected by the controller. Figure 7.21 illustrates the behavior of the position and velocity dispersions of the SM relative to the BM. Some trends are similar to Figure 7.20, specifically the zero initial condition and quick rise to steady state. One difference is the high frequency and shorted-lived dispersions caused by the onset of the controller at $t_{\text{star}}$. At $t = 4$ seconds, the velocity dispersions in Figure 7.21 start to develop numerical problems, causing the LinCov estimates to go beyond the Monte Carlo confidence intervals. This is due to round-off error caused by the very the small relative covariance which is computed by differenting the comparatively large inertial covariances shown in Figure 7.11. A square-root or UD factorization of the covariance could be used to avoid this problem.

Figures 7.22 and 7.23 show the behavior of two of the relative angle states. Figure 7.22 shows the attitude and angular rate of the SC with respect to the RM and is representative of the attitude/angular rate of the RM with respect to the BM. The dispersions are low prior to $t_{\text{star}}$, followed by a sharp increase in the covariance and eventual convergence to the steady state variance value. In contrast to the truth state angular dispersions shown in figures 7.8 and 7.9, the relative attitude and rate dispersions of the uncontrolled structural members do not grow with time. The relative angular dispersions of the SM, however, do grow with time as shown in Figure 7.23. This is because the inertial angle of the SM is being controlled, while the BM dispersions are increasing with time.

The final dispersion analyzed in this section is that of the actuator torque, shown in Figure 7.24. The behavior of this dispersion comprises two parts. First is a very high dispersion induced by the onset of the controller at $t_{\text{star}}$ (top plot), followed by convergence to a much lower steady-
Figure 7.16: 3σ RM attitude estimation error, initial (top) and steady-state (bottom)
Figure 7.17: 3σ Star camera bias estimation error
Figure 7.18: 3σ Gyro 1 bias estimation error
Figure 7.19: $3\sigma$ Gyro 1 scale factor estimation error
Figure 7.20: SC to RM relative position (top) and velocity (bottom) $3\sigma$ dispersion hair plots
Figure 7.21: SM to BM relative position (top) and velocity (bottom) $3\sigma$ dispersion hair plots
Figure 7.22: SC to RM relative attitude (top) and angular rate (bottom) $3\sigma$ dispersions plots
Figure 7.23: SM to BM relative attitude (top) and angular rate (bottom) 3σ dispersion plots
state variance (bottom plot). This information combined with knowledge of the nominal torque trajectory in Figure 7.7 provides valuable insight into the design of the actuator. It allows the designer to know the range of the torques that will be encountered which helps in specifying the required voltages and currents. It also provides motivation for a more intelligent initialization of the controller. If the current start-up procedure is used, the design must handle extremely large torques due to initialization, but will be significantly over designed for steady-state operating conditions. Modification of the start-up procedure could lower the initial torque dispersions and allow for a more appropriate actuator design. An example modification would be gain scheduling, where the controller gains are lowered when a large error is measured. This is a common approach used to avoid actuator saturation.

### 7.2.2 Autocorrelation Function Estimation

Two approaches to estimating the autocorrelation function of the random process were outlined in section 4.6. The Monte Carlo approach estimates the autocorrelation function from samples of the random process. The LinCov approach directly computes the function using the state transition matrix and the auto-covariance kernel. Given that the assumptions of stationary statistics (equation 4.159) and a time-invariant system (4.165) are valid, the two methods are equivalent. The following plots will illustrate examples of computing the autocorrelation function when the statistics are truly stationary, approximately stationary, and clearly not stationary.\(^1\) From figures 7.10, 7.12, and 7.13 it is apparent that the SM attitude/angular rate and the translational/rotational acceleration disturbances are truly stationary at the image acquisition time, \(t_{image}\). The autocorrelation estimates for these states are shown in figures 7.25 through 7.28. For all of these stationary states, the match between Monte Carlo-calculated and LinCov-calculated autocorrelation functions is excellent, especially for the short term.\(^2\)

When the states are slowly changing but have not yet reached steady-state, the match between the two methods begins to degrade. As shown in figures 7.9 and 7.14 of the previous section, the covariance of the RM angular rate and SM disturbance torque have not yet reached steady-state. The covariance of both states is very slowly changing, however, so the statistics of these

---

\(^1\)Note that for all of the plots in this section, the autocorrelation is not windowed, which aids in identifying some distinguishing features.

\(^2\)In all cases, the autocorrelation function is computed at the simulation time of \(dt = 0.25\text{ms}\) and across the window \([dt_{pad}, dt_{pad}]\) centered at the image acquisition time \(t_{image}\).
Figure 7.24: Actuator torque $3\sigma$ dispersion plots, initial (top) and steady-state (bottom)
Figure 7.25: Truth SM attitude autocorrelation estimates for short term (top) and long term (bottom)
Figure 7.26: Truth SM angular rate autocorrelation estimates for short term (top) and long term (bottom)
Figure 7.27: Translational acceleration disturbance autocorrelation estimates
Figure 7.28: Rotational acceleration disturbance autocorrelation estimates
states are approximately stationary. Figures 7.29 and 7.30 show the corresponding autocorrelation estimates. Two important features are noteworthy. First is the significant difference between the Monte Carlo and LinCov-calculated autocorrelations present in Figure 7.30. This difference is easy to spot. Figure 7.29, however, contains a more subtle feature. Although the match is very good for the short term, the LinCov-calculated autocorrelation contains a small bias, evident in the long-term plot. As discussed in [15] section 4.13, a bias in the autocorrelation function suggests a bias in the random process, which is not consistent with the definition of the dispersion states. Although there exists no formal way of determining if a random process is “stationary enough,” if the bias is small, then the process can be assumed stationary, and the resultant PSD assumed accurate.

When a random process is far from stationary, it is expected that the calculated autocorrelation function will have serious inaccuracies. Two examples of non-stationary states are the inertial position and angle of the RM. The variances of these states are clearly growing with time as illustrated in figures 7.11 and 7.9. The resulting LinCov-calculated autocorrelation functions in figures 7.31 and 7.32 are obviously incorrect as they show increasing correlations with respect to the lag. Thus the PSD estimated from these autocorrelations would be incorrect.

### 7.2.3 PSD computation

A key element of this research deals with computing estimates of the power spectral density of the dispersions. As described in section 4.6, the PSD can be computed via three different methods. The Monte Carlo-based method computes the PSD by averaging modified periodograms of samples of the random process. This is done under the assumptions of steady-state and stationary conditions. Under the same assumptions, the LinCov Acorr method estimates the PSD using the DFT of the windowed autocorrelation function. Under the assumption of a LTI system, the LinCov Transfer Matrix method directly computes the PSD from the input noise strengths and the dispersion transfer functions. In this section, PSD estimates for the stationary and approximately stationary states discussed in the previous section are presented. A non-stationary example is also included to illustrate an advantage to the LinCov Transfer Matrix method. In all the spectral plots of this section, a vertical black dashed line represents the lowest resolvable frequency, which corresponds to the total time period considered for PSD estimation, specifically $f_{\text{lower}} = 1/2dt_{\text{psd}} = 0.025\text{Hz}$.

Figure 7.33 compares the three methods of PSD estimation for the SM inertial attitude and
Figure 7.29: RM angular rate autocorrelation estimates for long term (top) and short term (bottom)
Figure 7.30: SM disturbance torque autocorrelation estimates
Figure 7.31: RM position autocorrelation estimate
Figure 7.32: RM attitude autocorrelation estimate
angular rate. The match is excellent for all frequencies below 30Hz. All three methods are able to resolve the strong 3Hz component. Above 30 Hz, the LinCov Acorr method resolves spectral peaks at 68 Hz and 160 Hz, the existence of which is supported by the OLTF discussed in section 5.3.2 and displayed in Figure 5.10. While the LinCov Transfer Matrix method also resolves the higher resonant peaks, the magnitude of the peaks are suppressed, consistent with the transfer function of the anti-resonance filter in Figure 5.11.

Figure 7.34 and 7.35 compare the PSD estimates of the translational and rotational acceleration disturbances, both of which are modeled as first-order Markov processes. Excellent agreement between the Monte Carlo and LinCov methods is observed across the entire spectrum.

Figures 7.36 and 7.37 contain the PSD estimates of the states that are approximately stationary, namely the inertial angular rate of the RM and the SM disturbance torques. In the case of the SM disturbance torques, the match is excellent across all frequencies. Even though this process had not reached the steady-state covariance value, it was “close enough” for the Monte Carlo and LinCov methods to agree. The PSD estimates for the RM angular rate in Figure 7.36 agree well for the 3Hz and 68Hz resonant peaks with a slight disagreement in the 10-40 Hz range. Above the 68Hz peak, the Transfer Matrix method falls off significantly faster than the other methods. This feature is also observed in Figure 7.33. One probable explanation for the divergence of the methods at higher frequencies is related to the window function, which is applied in the Monte Carlo and LinCov Acorr methods. In these methods, the Hamming window is multiplied, element by element, to the time-domain data. In the case of the Monte Carlo method, the window is applied to each realization of the random process. In the case of the LinCov Acorr method, the window is applied to the computed autocorrelation function. Multiplication in the time-domain is equivalent to convolution in the frequency domain. Figure 7.38 shows the Fourier transform of Hamming window with the magnitude normalized to one. Note that the peak of the curve is 13-15 orders of magnitude above the sides of the window. In many applications, this level of suppression is sufficient. In this analysis however, many of the PSDs exhibit a range of values similar to the Hamming window. During the convolution process, this results in an increase in the noise floor of the PSD. To illustrate the effect, consider the point in the convolution of the PSD in Figure 7.36 where the Fourier transform of the Hamming window in centered at 100Hz. At this point, the peak of Figure 7.38 is multiplied by a number on the order of $10^{-17}$. The left side of the curve, however, is multiplied by numbers ranging
Figure 7.33: SM attitude (top) and angular rate (bottom) PSD
Figure 7.34: Translational acceleration disturbance PSD
Figure 7.35: Rotational acceleration disturbance PSD
Figure 7.36: RM angular rate PSD
Figure 7.37: SM disturbance torque PSD
from $10^{-17}$ to $10^{-6}$, or up 11 orders of magnitude greater. To complete the convolution, all the multiplications are summed, and the result is an artificial increase in the PSD estimate. Although more in-depth investigation is needed to verify these results, it can be safely concluded that the effects of the window are not negligible in the analysis of this problem. The window effects however, are avoided in the Transfer Matrix method, because the PSD is analytically computed from the transfer functions of the system.

### 7.3 Navigation and Control Analysis

Three performance metrics were discussed in section 4.5.2, namely the truth state dispersions covariance, the navigation state dispersions covariance, and the covariance of the navigation errors. Often, the navigation state dispersion are not useful for performance analysis and will not be discussed in this work. The navigation error covariance, however, is very useful in determining the observability of the navigation states and optimality/tuning of the Kalman filter. The truth state dispersions are also useful in the context of this analysis to determine the amount of jitter present
in the pointing of the SM. An analysis technique applicable to both performance metrics is called sensitivity analysis. This technique involves calculating the contribution of each error source to the overall performance metric and is extremely useful in identifying the driving error sources in the problem. This section will start with a discussion regarding the navigation errors in the context of Kalman filter tuning. This is followed by a sensitivity analysis of some key performance-related states, namely, the RM attitude estimation error and the SM attitude and angular rate dispersions.

Figures 7.39 through 7.42 show the navigation error covariance as compared to the Kalman filter’s estimate of the same quantity. Consistent with the findings of section 7.2.1, the observability of the star tracker and gyro 1 parameters is poor. Given the y-axis scale of figures 7.41 and 7.42, however, there is a very slow downward trend in the covariance of the gyro 1 bias, suggesting that eventually, this state will become observable. In all examples of the navigation errors, the Kalman filter correctly estimates the covariance of the estimation errors. This is expected since the system is close to linear and the process and measurement noise matrices used in the Kalman filter are equal to the true values.

Figure 7.43 separates the contributors to the estimation error of the RM attitude. This is done by running several LinCov simulations, one for each error source. During each simulation, a single error source is activated, while the remaining error sources are set to zero. This has the effect of calculating the response of the system to a single error source. Once all of the individual simulations are run, a final simulation is run with all error sources activated. In the end, the RSS of all of the individual simulations is equal to the response of the final simulation.

Prior to the time of the first star camera measurement, $t_{\text{star}}$, the estimation error is dominated by the initial uncertainty in the attitude of the entire system (see top plot). Following the first star camera measurement, the Kalman filter gradually balances the contribution of the star camera and gyro 1 measurement noise. After $t = 4$ seconds, the main contributors to the estimation error are the star camera and gyro noise, with the star camera bias being the next significant contributor. Note that, consistent with the observability findings discussed earlier, the error contribution of the star camera bias does not decrease with time, which would be the case if this state was observable.

For the remaining sensitivity analyses in this section, the error sources are divided by subsystem as detailed in Table 7.1. Figures 7.44 and 7.45 show the error budget for the attitude and angular rate dispersions, respectively. In both cases, the initial dispersions are dominated almost
Figure 7.39: RM attitude estimation error 1σ standard deviation showing the initial uncertainty (top) and the steady-state uncertainty (bottom)
Figure 7.40: Star camera bias estimation error $1\sigma$ standard deviation
Figure 7.41: Gyro 1 bias estimation error $1\sigma$ standard deviation
Figure 7.42: Gyro 1 scale factor estimation error 1σ standard deviation
Figure 7.43: RM attitude estimation error budget showing initial uncertainty (top) and steady-state uncertainty (bottom)
entirely by the translational disturbances. This is due the large center of mass offset present in the SM. In the case of the attitude dispersions, the contribution of the initial conditions briefly exceeds that of the disturbance. Once the controller is activated at $t_{star}$, both dispersions shrink rapidly. In the case of the angular rate dispersions, the contribution of the initial conditions temporarily increase following $t_{star}$, but return to zero shortly thereafter. After the transient following $t_{star}$, it is clear that the major contributor to both the attitude and angular rate dispersions is the translational disturbance, which lies nearly directly beneath the total line.

An important time for calculating an error budget is the image acquisition time $t_{image}$. This is typically done with a bar plot. Figures 7.46 and 7.47 are a snapshot of the error budget at this critical time. Both confirm the conclusion that the translational disturbances are by far the largest source of error in the system, with the mirror system being the only other significant contributor. From a designers standpoint, there is no benefit to improving any of the subsystems until the sensitivity to translational accelerations is reduced. On the other hand, the requirements of other subsystem could be relaxed without significantly impacting the performance.

7.4 Spectral Sensitivity Analysis

In the previous section, sensitivity analysis was used to create an error budget, which aids the designer in identifying the main error contributors in the system. In section 7.2.3, the PSD was useful in identifying the spectrum of the dispersions. Suppose that for a particular system, it is desired to understand the portion of the spectrum caused by each of the individual error sources. This can be obtained by combining the two methods of sensitivity analysis and spectrum estimation into a single method, named Spectral Sensitivity Analysis. This method is illustrated for this problem in figures 7.48 and 7.49 using the LinCoV Transfer Matrix method. From these figures, it is clear that for frequencies below 10Hz, the translational disturbances are the largest contributor by several orders of magnitude. Interestingly, above 10Hz, the contribution of the mirror system errors exceeds that of the translational disturbances, except for in the vicinity of the 68 Hz and 160 Hz peaks, where the translational disturbance couple with the structural resonances. This same trade-off between dominant error sources is also observed in the PSD of the SM angular rate shown in Figure 7.49.

Another important feature shown in 7.49 is the low-frequency behavior of the error sources
Table 7.1: Subsystem definition

<table>
<thead>
<tr>
<th>Error Source</th>
<th>Initial Conditions</th>
<th>Star Tracker Navigation System</th>
<th>Mirror System</th>
<th>Actuator System</th>
<th>Rotational Disturbances</th>
<th>Translational Disturbances</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{\text{ang}}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{bstar},ss}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{star}}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{bgyro1},ss}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{sgyro1},ss}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{gyro1}}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{bresolver},ss}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{resolver}}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{bgyro2},ss}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{sgyro2},ss}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{gyro2}}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{btorque},ss}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{storque},ss}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{D},ss}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{rot},ss}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_{\text{trans},ss}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 7.44: SM attitude error budget showing initial uncertainty (top) and steady-state uncertainty (bottom)
Figure 7.45: SM angular rate error budget showing initial uncertainty (top) and steady-state uncertainty (bottom)
Figure 7.46: SM attitude error budget (bar plot) at $t_{\text{image}}$

Figure 7.47: SM angular rate error budget (bar plot) at $t_{\text{image}}$
that induce disturbance torques on the SM. These include the rotational and translational disturbances as well as elements of the mirror and actuator systems. Recall that the bandwidth of the inner loop controller is 11Hz. Thus disturbance torques inside the bandwidth will be attenuated by the rate-damping controller. This is consistent with the PSD for each of these contributors. The controller however cannot attenuate error sources that are in the feedback path. A good example of this type of error source is the star tracker navigation system, which maintains a constant influence across the bandwidth of the rate damping controller.

It is also interesting to observe which error sources couple into the structural modes of the system at 3Hz, 68Hz, and 160Hz. The translational disturbances strongly couple into every mode. The coupling of the rotational disturbance also exists, but 10 orders of magnitude lower. The actuator system error sources couple very softly with the 68Hz and 160Hz modes. On the other hand, the mirror and star tracker navigation system errors exhibit nulls at the location of the upper two resonances, due to the effects of the anti-resonance filter.

Two final observations are made from figures 7.48 and 7.49. First is that the sum of the individual error sources (solid blue line) is equal to the simulation where all sources are active (dashed blue line). This is due to the principle of superposition which states that for a linear system, the combined response due to multiple inputs is equal to the sum of the response to each input. This characteristic is a useful validation of the sensitivity analysis. Finally, when computing the spectral sensitivity using the LinCoV Transfer Matrix method, the sensitivity to initial conditions is zero. This is consistent with the fact that the Transfer Matrix method computes the steady-state response to the driving inputs, and does not account for initial conditions.

Another useful application of the spectral sensitivity plot is to determine the error budget for a certain frequency band. This is particularly useful when a sensor on-board the system is sensitive to a specific band of frequencies. The PSDs in the spectral sensitivity analysis can be integrated across a specified frequency range to calculate the RMS dispersion in the band of interest. Figures 7.50 and 7.51 illustrate an example of this, where the spectral band considered is from 8Hz to 13Hz. Note that when compared to the sensitivity bar plots of the entire spectrum (figures 7.46 and 7.47), the mirror system is a much more significant contributor, comparable to the translational accelerations. Thus to reduce the sensitivity of the system in this frequency band both error sources must be considered. This type of analysis provides the designer with another powerful tool, created
from LinCov simulation data, to better understand not only the spectrum of the system, but the contributors to different parts of the spectrum.
Figure 7.48: SM attitude dispersion spectral sensitivity (PSD)
Figure 7.49: SM angular rate dispersion spectral sensitivity (PSD)
Figure 7.50: SM attitude spectral sensitivity (bar plot) at $t_{\text{image}}$ from 8 to 13Hz

Figure 7.51: SM angular rate spectral sensitivity (bar plot) at $t_{\text{image}}$ from 8 to 13Hz
Chapter 8

Conclusions

Prior to the time of this research, the focus in LinCov analysis has been the calculation of the dispersions from the mean reference trajectory. In all cases, the most important metric has been the covariance of the dispersion. In a gimbaled pointing system however, the frequency spectrum of the dispersion is equally important, as it relates to the amount of jitter observed by the sensor. In addition, flexible structures have not been considered in previous LinCov work, the presence of which requires control laws that support internal states and actuator command filtering. The purpose of this research was to extend linear covariance analysis to support these three advancements—dispersion frequency spectrum, flexible structures, and internal control law states—and apply these extensions to the analysis of a gimbaled pointing system.

In chapter 4 of this dissertation, a nonlinear truth model is presented that is capable of modeling structures with rigid and flexible bodies. This model also defines a more flexible form of the control law, with internal states and associated state dynamics, actuator command filtering, and direct sensor feedback. In addition to these new elements, many of previously existing LinCov elements were included, namely actuator, sensor, and navigation models. The nonlinear truth model was then linearized about a nominal trajectory, which results in a set of linear differential and update equations describing the deviations from the nominal, from which the covariance of the dispersions states is derived. Using information available from the LinCov simulation (namely the state transition matrix and covariance matrix), a method for computing the autocorrelation function under the assumptions of a stationary random process and LTI system was developed. The PSD was then computed as the DFT of the windowed autocorrelation function. The LinCov-based PSD estimate was empirically shown to be equivalent to the Monte Carlo-based Welch method in the case of a first-order Markov process. An additional LinCov-based method was also developed to analytically compute the PSD from the coefficient matrices of the augmented system and the input noise strengths. Under the conditions of a LTI system, this latter method was shown to more accurately compute the PSD of the dispersion states and did not suffer from window effects.
Chapter 5 focused on developing the structural dynamics model and the navigation and control algorithms for a gimbaled pointing system. The dynamics model is tailored to a SOFIE-like instrument but is generic enough to accurately model many types of gimbaled systems. The structural dynamics model is capable of modeling up to 6 structural modes, 3 translational and 3 rotational, and includes the effects of disturbance torques and center of mass offsets, which couple the translational and rotational dynamics. This model also supports colored rotational and translational acceleration noise in the system, the bandwidth of which can be obtained from the host platform PSD. The developed model was validated via the modern software dynamics package ADAMS. The navigation algorithm was developed to incorporate measurements from a star tracker and gyro, corrupted by biases, scale factor, and noise. Propagation of the attitude was done via model replacement using the gyro measurements. Updates to the attitude were performed using star tracker measurements. Finally, the controller and anti-resonance filter were designed with adequate stability margins using resolver feedback for position control and a second gyro for rate damping. The product of this chapter is a realistic, closed-loop navigation and control model of a gimbaled pointing system.

Chapter 6 was included to define, in one location, all of the equations necessary for the simulation with the corresponding numerical values for system parameters and initial conditions. It is intended to be a one-stop-shop for all equations and parameters necessary for the simulation.

Chapter 7 contains the analysis of the system dispersions, navigation errors, and dispersion PSDs. To give meaning to the dispersion analysis results, the mean reference trajectory is analyzed. This is followed by analysis of the system using conventional Monte-Carlo-based approaches for dispersion analysis and PSD estimation and compares the results to a LinCov-based analysis of the same quantities. For all dispersion states, the match between Monte Carlo and LinCov variance estimates is excellent. The three PSD estimation techniques match perfectly for simple ECRV states. The match is also good in the lower frequencies for states with more spectral content. At higher frequencies, however, the different PSDs diverge, presumably due to the effects of the data windowing present in the Monte Carlo and LinCov Acorr methods. The LinCov Transfer Matrix method does not suffer from window effects.

Though not the focus of this research, analysis is done on the observability of several parameter states. The conclusion is that, given the low level of dynamics typical of the problem analyzed,
the star tracker and gyro parameters are not observable. After validating the accuracy of the Lin-Cov model, it was used to perform sensitivity analysis which creates error budgets for both the navigation errors and pointing dispersions. These error budgets show a well-balanced navigation system dominated by star tracker and gyro measurement noise. The pointing errors however are not well balanced and are dominated almost entirely by translational disturbance accelerations. It was therefore concluded that, prior to any other subsystem improvement, the sensitivity of the gimbal system to translational accelerations needs to be reduced. At the end of this chapter, the methods of sensitivity analysis and PSD estimation are combined to create a new analysis method, named Spectral Sensitivity Analysis. This new method shows the frequency distribution of the errors caused by individual error sources. The translational acceleration disturbances dominate over the lower end of the spectrum. However, when focusing on a specific portion of the spectrum, 8 to 13Hz in this example, the contribution of the mirror system becomes equally important. This becomes an important tool when considering sensors on-board the gimbal that are sensitive to certain regions of the spectrum.

In summary, this dissertation has shown that the capabilities of linear covariance analysis can be extended to analyze a gimbaled pointing system. It was shown that with moderate changes to the linear covariance equations, controllers with internal states and command filtering can be incorporated and their effects analyzed. It was also shown that, under similar assumptions used in conventional Monte Carlo approaches, knowledge of the state transition matrix and covariance matrix permits the calculation of the autocorrelation function, which can be used to estimate the PSD. An alternative method to compute the PSD was also presented, based on the transfer matrix of the LinCov system, which analytically computes the PSD of the dispersion states. These two methods provide a missing capability to the LinCov tool set and further extends the applicability of this powerful analysis approach.

During the course of this work, several new areas were identified as potential areas of advancement. The first area of future work relates to developing alternate methods of PSD estimation from the LinCov simulation. One promising candidate is the use of parametric PSD estimation algorithms. The LinCov Acorr method used in this work is to window the autocorrelation function, then estimate the PSD using the DFT, categorized as a non-parametric method. To reduce the effects of the window function, a large number of autocorrelation samples is desired. An alternate
approach that works well when there is a limited number of samples is to fit the coefficients of an assumed model to the samples of the autocorrelation function. Since this method is model-based, a window need not be used, thus avoiding the associated window-effects.

The second area involves incorporating the frequency content of the mean reference trajectory to compute the total PSD of the truth state. As applied to the system in this work, the image smear that will be produced is a function of the frequency content of attitude dispersion and the nominal attitude trajectory. In this work, this problem was minimized by allowing the controller step response to settle before the image acquisition time. But in a rapidly scanning system, this may not be possible and so the frequency content of the nominal must be considered.

The third area of future work relates to incorporating the frequency content of the Kalman update. This research focused on the frequency content of the dispersion state dynamics. In some systems, the Kalman update causes significant frequency content in the performance-related states. Coa et al. [38] outline a promising approach where the transfer function of a multi-rate system is computed. Once obtained, the transfer function could be used in the framework of the LinCov Transfer Matrix method to compute the PSD of the combined propagation and update system.

The fourth and final area of future work concerns the friction model used in the gimbaled point system. The model used in this research assumed viscous friction in the joint between the steering mirror and the base mass. While this is a very good approximation for some systems, a higher fidelity model that includes the combination of stiction, viscous, and Coulombic friction would result in a more accurate understanding of the performance of the system.
References


Curriculum Vitae

Randall S Christensen

CAREER OBJECTIVE

Make significant contributions in the fields of guidance, navigation, control, and dynamics.

EDUCATION

Ph.D Degree: Mechanical and Aerospace Engineering
Utah State University, Logan, UT. 2008-2013

- GPA: 3.96
- Emphasis on linear covariance analysis, guidance, navigation, and control
- Dissertation: Linear Covariance Analysis for Gimbaled Pointing Systems

Master of Science/ Bachelor of Science, Concurrent: Mechanical Engineering
Utah State University, Logan, UT. 2001-2006

- GPA: 3.97
- Emphasis on 3D model generation, guidance, navigation, and control
- Thesis Project: Design and Implementation of a Turntable-Based Texel Camera for Rapid 3D Model Generation

PUBLICATIONS

Journal Publications


Conference Proceedings

**EXPERIENCE**

**Mechanical Engineering Group, The Space Dynamics Laboratory**, North Logan, UT. January 2007-Present

- Aided inertial navigation (GPS, vision, Doppler, etc.)
- Navigation and control for satellite proximity operations (Angles-only relative navigation, Lidar-based relative navigation)
- Spacecraft attitude determination and control
- Stabilization and geo-pointing of gimbaled pointing systems
- Statistical modeling and simulation
- Electro-mechanical system design and analysis
- Dynamic structural analysis

**Research Assistant, CAIL (Center for Advanced Imaging Lidar)**, Logan, Utah, February 2005 – December 2006

- Responsible for prototype development of combined laser rangefinder, digital camera apparatus for automated 3D model generation.
- Experience in optical fixture design, fabrication of mechanical components, PCB design, firmware and mid-scale software development.


- Designed, drafted, and documented thrill ride components. Maintained project parts books.

**Mechanical Engineering Intern, Autoliv ASP**, Promontory, Utah, Summer 2004

- Performed large-scale design of experiment (DOE) studies on various prototype pyrotechnics.
- Experience in statistical models and quality assurance.

**Research Assistant, IBM Almaden Research Center**, San Jose, California, Summer 2003

- Investigated the mechanical, electrical, and surface properties of various bi-stable memory device structures on the nanometer scale.
- Aided in development a novel, less-harsh manufacturing method for memory devices.

---

**AWARDS**

Space Dynamics Laboratory Tomorrow Fellowship 2007-2013, Space Dynamics Laboratory Outstanding Performance Achievement Award 2010, USU Department of Mechanical Engineering Outstanding Senior 2006 and Outstanding Junior 2005, Eagle Scout.
RELEVANT COURSES

Advanced Strapdown Inertial Navigation Analytics (Roscoe), Spacecraft Navigation (Geller), Spacecraft Attitude Control (Fullmer), Stochastic Estimation with Aerospace Applications (Geller), Robotics (Cripps), Linear Multivariable Control (Wen), Mechatronics (Chen), Dynamic System and Controls (Fullmer), Optimal Spacecraft Guidance (Geller), Advanced Astrodynamics (Geller), Compressible Fluid Flow (Whitmore), Continuum Mechanics and Elasticity (Yu), Fluid Dynamics (Spall), Dynamics and Vibration (Folkman), Stochastic Processes (Lo), Math Methods for Signals and Systems (Bose).