Modeling and Performance Analysis of GPS Vector Tracking Algorithms

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

Matthew Lashley

Certificate of Approval:

David M. Bevly, Co-Chair
Associate Professor
Mechanical Engineering

John Y. Hung, Co-Chair
Professor
Electrical and Computer Engineering

Jitendra K. Tugnait
Professor
Electrical and Computer Engineering

George T. Flowers
Dean
Graduate School

Modeling and Performance Analysis of GPS Vector Tracking Algorithms

Matthew Lashley

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
December 18, 2009

Modeling and Performance Analysis of GPS Vector Tracking Algorithms

Matthew Lashley

Permission is granted to Auburn University to make copies of this dissertation at its
discretion, upon the request of individuals or institutions and at
their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

VITA

Matthew Vernon Lashley was born in Roanoke, Alabama on November 12, 1981. He is the second child of Vernon and Vicky Lashley, and has an older brother, William. Matthew grew up in the small community of Malone, near the town of Wadley, Alabama. He graduated from Wadley High School in 2000. Matthew earned his bachelor's degree in electrical engineering in May 2004 from Auburn University. Matthew started graduate school in the electrical engineering department studying control systems under Dr. John Hung in the Fall of 2004. In the Fall of 2005 Matthew accepted a position in the GPS and Vehicle Dynamics Lab (GAVLAB) as a graduate research assistant, working for Dr. David Bevly. Matthew received his M.S. in electrical engineering in December of 2006. In January 2007, he started working toward his doctorate in electrical engineering at Auburn University under the advisement of Dr. Hung and Dr. Bevly. Matthew became a employee of Navigation Technology Associates (NTA) in January 2007. He plans on continuing to work for NTA after he finishes his doctorate.

DISSERTATION ABSTRACT

MODELING AND PERFORMANCE ANALYSIS OF GPS VECTOR TRACKING ALGORITHMS

Matthew Lashley

Doctor of Philosophy, December 18, 2009
(M.S., Auburn University, 2006)
(B.E.E., Auburn University, 2004)

417 Typed Pages

Directed by John Y. Hung and David M. Bevly

This dissertation provides a detailed analysis of GPS vector tracking algorithms and the advantages they have over traditional receiver architectures. Standard GPS receivers use a decentralized architecture that separates the tasks of signal tracking and position/velocity estimation. Vector tracking algorithms combine the two tasks into a single algorithm. The signals from the various satellites are processed collectively through a Kalman filter.

The advantages of vector tracking over traditional, scalar tracking methods are thoroughly investigated. A method for making a valid comparison between vector and scalar tracking loops is developed. This technique avoids the ambiguities encountered when attempting to make a valid comparison between tracking loops (which are characterized by noise bandwidths and loop order) and the Kalman filters (which are characterized by process and measurement noise covariance matrices) that are used by vector tracking algorithms. The improvement in performance offered by vector tracking is calculated in multiple different scenarios.

Rule of thumb analysis techniques for scalar Frequency Lock Loops (FLL) are extended to the vector tracking case. The analysis tools provide a simple method for analyzing the performance of vector tracking loops. The analysis tools are verified using Monte Carlo simulations. Monte Carlo simulations are also used to study the effects of carrier to noise power density (C/N_o) ratio estimation and the advantage offered by vector tracking over scalar tracking. The improvement from vector tracking ranges from 2.4 to 6.2 dB in various scenarios.

The difference in the performance of the three vector tracking architectures is analyzed. The effects of using a federated architecture with and without information sharing between the receiver's channels are studied. A combination of covariance analysis and Monte Carlo simulation is used to analyze the performance of the three algorithms. The federated algorithm without information sharing performs poorer than the other two architectures. However, at low C/N_0 ratios the difference in the performance of the three algorithms becomes virtually zero.

The analysis of different vector tracking architectures is then extended to an analysis of different Deeply Integrated (DI) architectures. The effects of using a federated filtering architecture on DI's performance are investigated. Covariance analysis and Monte Carlo simulation are also used to study the performance of the different DI algorithms. The results from the DI analysis mirror the results from the analysis of different vector tracking algorithms. The different DI architectures exhibit the same performance at low C/N_0 ratios.

The vector tracking algorithms are also implemented in MATLAB. The algorithms are tested using data collected from an environment with dense foliage (having widely fluctuating signal levels) and from an urban canyon type environment. The performance of the vector tracking algorithms is compared to that of a NovAtel ProPak-V3 receiver in the same scenarios. The vector tracking algorithms provide near continuous coverage through both environments while the NovAtel receiver exhibits periods of prolonged outages.

ACKNOWLEDGMENTS

I would like to thank all my family and friends for their support during my time in graduate school. I would not have been able to complete my doctorate without their help. I would also like to thank Dr. John Hung for advising me during my time in graduate school. Dr. Hung has always been willing set aside time to share his knowledge and wisdom with me. Dr. Hung has been a mentor and role model to me and many other students at Auburn University.

I also owe a debt of gratitude to Dr. David Bevly. Dr. Bevly originally hired me as a research assistant when I was a master's student. He provided me with an office and all the equipment used to perform the research in this dissertation. It has been a great experience working and interacting with someone like Dr. Bevly who is heavily involved in navigation research.

I would also like to thank all the members of the GPS and Vehicle Dynamics Lab (GAVLAB) that I have worked with. Without their help I could not have completed my research. I would specifically like to acknowledge the help of William Travis. He spent a great deal of time helping me collect the data used in this dissertation.

I would like to thank the owners of Navigation Technology Associates (NTA) for employing me and allowing me to continue my education at Auburn. Specifically, I would like to thank Shannon Fields and Jeff Rhea. I am also obliged to Bill Negast at NTA for helping collect the RF simulator data used in the dissertation and for his proof reading services. I would also like to thank Brian Baeder and Edwin Hogan at the U.S. Army Aviation and Missile Research, Development and Engineering Center (AMRDEC) for their support.

Style manual or journal used Journal of Approximation Theory (together with the style known as "aums"). Bibliography follows van Leunen's *A Handbook for Scholars.*

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file `aums.sty`.

TABLE OF CONTENTS

LIST OF FIGURES xiv

1 INTRODUCTION 1
  1.1 Motivation 1
  1.2 Prior Work 3
    1.2.1 Vector Tracking 3
    1.2.2 Aided Receivers 5
    1.2.3 Deficiencies in the Current Understanding of Deeply Integrated GPS/INS Systems 19
  1.3 Contributions 21

2 GPS SIGNAL STRUCTURE 24
  2.1 Introduction 24
  2.2 General Signal Structure 24
  2.3 GPS Carrier Signal 25
  2.4 Gold Code 27
  2.5 Data Message 29
  2.6 Conclusion 32

3 POSITION AND VELOCITY DETERMINATION 33
  3.1 Introduction 33
  3.2 Received Signal Model 33
  3.3 Position Determination 34
  3.4 Velocity Determination 36
  3.5 Conclusion 38

4 SIGNAL TRACKING 40
  4.1 Introduction 40
  4.2 Conventional Tracking Loops 40
  4.3 Receiver Noise Performance Metrics 45
  4.4 Conclusion 49

5 VECTOR TRACKING RECEIVER 50
  5.1 Introduction 50
  5.2 Traditional Receiver Architecture 50
  5.3 Vector Tracking Architecture 52
    5.3.1 Position-State Formulation 54

5.3.2 Pseudorange-State Formulation 58
5.4 Carrier Phase Vector Tracking 61
5.5 Conclusion 63

6 VECTOR TRACKING LOOP PERFORMANCE 65
6.1 Introduction 65
6.2 Weighted Least Squares Example 65
6.2.1 Scalar Tracking Method 67
6.2.2 Vector Tracking Method 68
6.2.3 Performance Gain from Collective Processing 71
6.3 Coupled and Decoupled Pseudorange State VDFLL 74
6.3.1 Pseudorange State VDFLL Covariance Analysis 79
6.4 Conclusion 84

7 VECTOR TRACKING LOSS OF LOCK PERFORMANCE ANALYSIS 86
7.1 Introduction 86
7.2 Rule of Thumb Analysis Techniques 87
7.2.1 Frequency Lock Loop Analysis 87
7.2.2 Vector Delay/Frequency Lock Loop Analysis 90
7.2.3 Vector Delay/Frequency Lock Thermal Noise Performance 92
7.2.4 Vector Delay/Frequency Lock Loop Dynamic Stress Steady State Performance 98
7.2.5 Scenario Dependent Vector Delay/Frequency Lock Loop Dynamic Stress Performance 104
7.3 Monte Carlo Simulation Results 106
7.3.1 Simulation Setup 106
7.3.2 Simulation Results 109
7.3.3 Effect of C/N_0 Estimation 117
7.3.4 Comparison of Scalar and Vector Tracking Algorithms 122
7.4 High Dynamics Simulation Results 133
7.4.1 Simulation Setup 134
7.4.2 Simulation Results 135
7.5 Conclusion 140

8 ANALYSIS OF DIFFERENT VECTOR TRACKING ARCHITECTURES 143
8.1 Introduction 143
8.2 Deep Integration Filter Architectures 143
8.3 Single Central Filter 146
8.4 Federated Filtering Architectures 150
8.4.1 Federated Architecture with Coupled Channel Residual Filter 151
8.4.2 Federated Architecture with Decoupled Channel Residual Filters . . 159
8.5 Covariance Analysis of Vector Tracking Architectures 167

8.5.1 Covariance Analysis Results for Eleven Satellite Scenario 173
8.5.2 Covariance Analysis Results for Five Satellite Scenario 183
8.6 Monte Carlo Test Results of Vector Tracking Architectures 186
8.6.1 Monte Carlo Simulation Results for Eleven Satellite Scenario 188
8.6.2 Monte Carlo Simulation Results for Five Satellite Scenario 195
8.7 Conclusion 199

9 ANALYSIS OF DIFFERENT DEEP INTEGRATION ARCHITECTURES 201
9.1 Introduction 201
9.2 Deep Integration Architectures 201
9.2.1 Deep Integration with Centralized Filter 204
9.2.2 Federated Filtering Architectures 209
9.2.3 Federated Deep Integration Filter with Coupled Channels 210
9.2.4 Federated Deep Integration Filter with Decoupled Channels 221
9.3 Covariance Analysis of Deeply Integrated Architectures 231
9.3.1 Tactical Grade IMU, Temperature Controlled Oscillator, and Eleven Satellite Scenario 235
9.3.2 Automotive Grade IMU, Temperature Controlled Oscillator, and Five Satellite Scenario 248
9.4 Monte Carlo Test Results of Deep Integration Architectures 251
9.4.1 Eleven Satellite and Tactical Grade IMU Monte Carlo Simulation Results 253
9.4.2 Five Satellite and Automotive Grade IMU Monte Carlo Simulation Results 256
9.5 Conclusion 259

10 LIVE SKY TEST RESULTS USING VECTOR TRACKING 261
10.1 Introduction 261
10.2 Vector Tracking Test Results in Dense Foliage 261
10.3 Vector Tracking Test Results in an Urban Canyon 273
10.4 Conclusion 286

11 CONCLUSION AND FUTURE WORK 287
11.1 Conclusion 287
11.2 Future Work 291

BIBLIOGRAPHY 294

APPENDICES 303

A EQUIVALENCE OF POSITION-STATE AND PSEUDORANGE-STATE VECTOR TRACKING FORMULATIONS 304
A.1 Introduction 304

A.2 Covariance Analysis 304
  A.2.1 Position-State Covariance Analysis 306
  A.2.2 Pseudorange-State Covariance Analysis 310
  A.2.3 Equivalence in Steady State Prediction Errors 313

B C/N_0 RATIO ESTIMATION 318
B.1 Introduction 318
B.2 Noise Power Estimation 318
B.3 Signal Amplitude Measurements 320
B.4 Amplitude Kalman Filter 322
B.5 Adaptive Measures 324

C ADDITIONAL PLOTS 328
C.1 Additional Plots From Chapter 7 Section 3.2 328
C.2 Additional Plots From Chapter 7 Section 3.4 335
C.3 Additional Plots From Chapter 8 Section 6.1 342
C.4 Additional Plots From Chapter 8 Section 6.2 350
C.5 Additional Plots From Chapter 9 Section 3.1 362
C.6 Additional Plots From Chapter 9 Section 3.2 365
C.7 Additional Plots From Chapter 9 Section 4.1 368
C.8 Additional Plots From Chapter 9 Section 4.2 371

D ERROR STATE RECURSIONS 374
D.1 Introduction 374
D.2 Motivation 374
D.3 Error State Recursions 379

E INERTIAL SENSOR INTEGRATION AND ERROR MODELING 382
E.1 Introduction 382
E.2 Strap down Inertial Measurement Units 382
E.3 Inertial Measurement Unit Error Modeling 385
E.4 Extended Kalman Filter for Tight and Deeply Integrated Architectures . . 387

List of Figures

1.1 Loosely Coupled Architecture 7

1.2 Tightly Coupled Architecture 8

1.3 Ultra-Tightly Coupled Architecture 9

1.4 Raytheon Method of Ultra-Tight Coupling [Horslund and Hooker, 1999] . . 12

1.5 Anthony Abbott Method of Ultra-Tight Coupling [Abbott and Lillo, 2003] . 13

1.6 Draper Method of Deep Integration [Gustafson et al., 2001] 15

2.1 Autocorrelation Function of Gold Codes 29

2.2 Ideal Autocorrelation Function of Gold Codes 30

2.3 Cross-Correlation of Different Gold Codes 30

2.4 Arrangement of the GPS Navigation Message, [Hamm, 2005] 31

4.1 Receiver Correlator Structure 41

4.2 Delay Lock Loop 43

4.3 Costas Loop 44

4.4 Frequency Lock Loop 45

4.5 J/S Performance of an Unaided GPS Receiver for Wideband Jammer . . . 48

5.1 Traditional Receiver Architecture 51

5.2 Vector Tracking Receiver Architecture 52

5.3 Cascaded Vector Phase Lock Loop Architecture 62

6.1 Position Dilution of Precision and Number of Satellites Throughout Data Collection Period 73

6.2 Maximum and Minimum Performance Gain in C/N_0 from Vector Tracking for Weighted Least Squares Example 73

6.3 Pseudorange-Rate Variance Versus C/N_0 for Coupled and Decoupled Filters 81

6.4 Maximum and Minimum Performance Gain in C/N_0 from Vector Tracking Assuming Equal Signal Powers. 82

6.5 Maximum and Minimum Performance Gain in C/N_0 for Vector Tracking Using Actual Signal Powers 83

7.1 VDFLL Thermal Noise Jitter Versus C/N_o When Tracking Five Satellites . 96

7.2 VDFLL Thermal Noise Jitter Versus C/N_o When Tracking Seven Satellites 97

7.3 VDFLL Thermal Noise Jitter Versus C/N_o When Tracking Eleven Satellites 97

7.4 Maximum Jerk Stress Versus C/N_o When Tracking Five Satellites 102

7.5 Maximum Jerk Stress Versus C/N_o When Tracking Five Satellites 102

7.6 Maximum Jerk Stress Versus C/N_o When Tracking Eleven Satellites 103

7.7 Maximum Jerk Stress Versus C/N_o When Tracking Eleven Satellites 104

7.8 VDFLL Thermal Noise Analysis with 11 Satellites and σ_x^2= .2 m^2/s^2 110

7.9 VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= .2 m^2/s^2 111

7.10 VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= 2 m^2/s^2 111

7.11 VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= 10 m^2/s^2 112

7.12 VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= 50 m^2/s^2 113

7.13 VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= 100 m^2/s^2 113

7.14 VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= 300 m^2/s^2 114

7.15 VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= .02 m^2/s^2 115

7.16 VDFLL Simulation Results for 11 Satellites and σ^2= 300 m^2/s^2 With and Without Prior Knowledge of C/N_o ratio 119

7.17 VDFLL Simulation Results for 11 Satellites and σ^2= 50 m^2/s^2 With and Without Prior Knowledge of C/N_o ratio 119

7.18 VDFLL Simulation Results for 11 Satellites and σ^2= 10 m^2/s^2 With and Without Prior Knowledge of C/N_o ratio 120

7.19 VDFLL Simulation Results for 11 Satellites and σ^2= 2 m^2/s^2 With and Without Prior Knowledge of C/N_o ratio 120

7.20 VDFLL Simulation Results for 11 Satellites and σ^2= .2 m^2/s^2 With and Without Prior Knowledge of C/N_o ratio 121

7.21 VDFLL Simulation Results for 11 Satellites and σ^2= .02 m^2/s^2 With and Without Prior Knowledge of C/N_o ratio 121

7.22 Thermal Noise Analysis Results for 11 Satellites and σ^2= 10 m^2/s^2 125

7.23 Monte Carlo Simulation Results for 11 Satellites and σ^2= 10 m^2/s^2 126

7.24 Thermal Noise Analysis Results for 11 Satellites and σ^2= 50 m^2/s^2 127

7.25 Monte Carlo Simulation Results for 11 Satellites and σ^2= 50 m^2/s^2 127

7.26 Thermal Noise Analysis Results for 11 Satellites and σ^2= 100 m^2/s^2 128

7.27 Monte Carlo Simulation Results for 11 Satellites and σ^2= 100 m^2/s^2 129

7.28 Monte Carlo Simulation Results for a Single Channel of the Decoupled Filter 132

7.29 ECEF Y and Z Coordinates During Simulation 135

7.30 VDFLL Thermal Noise Jitter and Dynamic Stress Error at 19 dB-Hz Using 5 Satellites 136

7.31 VDFLL Thermal Noise Jitter and Dynamic Stress Error at 16 dB-Hz Using 7 Satellites 137

7.32 VDFLL Velocity Estimate Errors Using 7 Satellites at 16 dB-Hz 138

7.33 VDFLL Y and Z-Axis Acceleration Data Using 11 Satellites at 16 dB-Hz . 138

7.34 2 Hz FLL and VDFLL Performance at 19 dB-Hz 140

8.1 Single Centralized Filter Architecture 146

8.2 Federated Filter Architecture with Coupled-Channel Residual Filters 151

8.3 Federated Filter Architecture with Decoupled-Channel Residual Filters . . . 159

8.4 Covariance Analysis Results, X-axis Position Error for the Different Filter Architectures 173

8.5 Covariance Analysis Results, X-axis Velocity Error for the Different Filter Architectures 175

8.6 Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters, σ^2=2 $\mathrm{m}^2/\mathrm{s}^2$ 175

8.7 Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters, σ^2=2 $\mathrm{m}^2/\mathrm{s}^2$ 176

8.8 Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters when Varying the Process and Measurement Noise 179

8.9 Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters when Varying the Process and Measurement Noise 179

8.10 Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters, σ^2=10 $\mathrm{m}^2/\mathrm{s}^2$ 180

8.11 Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters, σ^2=10 $\mathrm{m}^2/\mathrm{s}^2$ 181

8.12 Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters, σ^2=50 $\mathrm{m}^2/\mathrm{s}^2$ 182

8.13 Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters, σ^2=50 $\mathrm{m}^2/\mathrm{s}^2$ 182

8.14 Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters, σ^2=2 $\mathrm{m}^2/\mathrm{s}^2$ 183

8.15 Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters, σ^2=2 $\mathrm{m}^2/\mathrm{s}^2$ 184

8.16 Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters, σ^2=50 $\mathrm{m}^2/\mathrm{s}^2$ 185

8.17 Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters, σ^2=50 m^2/s^2 185

8.18 Monte Carlo Simulation Results for the X-Axis Position with Eleven Satellites and σ^2=2 m^2/s^2 190

8.19 Monte Carlo Simulation Results for the X-Axis Velocity with Eleven Satellites and σ^2=2 m^2/s^2 191

8.20 Monte Carlo Simulation Results for the Clock Bias with Eleven Satellites and σ^2=2 m^2/s^2 191

8.21 Monte Carlo Simulation Results for the Clock Drift with Eleven Satellites and σ^2=2 m^2/s^2 192

8.22 Monte Carlo Simulation Results for the X-Axis Position with Eleven Satellites and σ^2=50 m^2/s^2 193

8.23 Monte Carlo Simulation Results for the X-Axis Velocity with Eleven Satellites and σ^2=50 m^2/s^2 194

8.24 Monte Carlo Simulation Results for the Clock Bias with Eleven Satellites and σ^2=50 m^2/s^2 194

8.25 Monte Carlo Simulation Results for the Clock Drift with Eleven Satellites and σ^2=50 m^2/s^2 195

8.26 Monte Carlo Simulation Results for the X-Axis Position with Five Satellites and σ^2=2 m^2/s^2 196

8.27 Monte Carlo Simulation Results for the X-Axis Velocity with Five Satellites and σ^2=2 m^2/s^2 196

8.28 Monte Carlo Simulation Results for the X-Axis Position with Five Satellites and σ^2=50 m^2/s^2 197

8.29 Monte Carlo Simulation Results for the X-Axis Velocity with Five Satellites and σ^2=50 m^2/s^2 198

9.1 Centralized Deep Integration Filter 205

9.2 Federated Deep Integration Filter with Coupled Channels 211

9.3 Federated Deep Integration Filter with Decoupled Channels 222

9.4 Receiver Trajectory During Flight Profile 234

9.5 Covariance Analysis Results for the X-Axis Position State with Eleven Satellites and Tactical Grade IMU 236

9.6 Covariance Analysis Results for the X-Axis Velocity State with Eleven Satellites and Tactical Grade IMU 237

9.7 Covariance Analysis Results for the Clock Bias State with Eleven Satellites and Tactical Grade IMU 238

9.8 Covariance Analysis Results for the Clock Drift State with Eleven Satellites and Tactical Grade IMU 238

9.9 X-Axis Position Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 45 dB-Hz 241

9.10 X-Axis Velocity Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 45 dB-Hz 241

9.11 Channel One Pseudorange Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 45 dB-Hz 242

9.12 Channel One Pseudorange-Rate Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 45 dB-Hz 243

9.13 X-Axis Position Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 20 dB-Hz 244

9.14 X-Axis Velocity Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 20 dB-Hz 245

9.15 Covariance Analysis Results for the X-Axis Position State with Eleven Satellites and Tactical Grade IMU 246

9.16 Covariance Analysis Results for the X-Axis Velocity State with Eleven Satellites and Tactical Grade IMU 247

9.17 Covariance Analysis Results for the X-Axis Position State with Five Satellites and Automotive Grade IMU 248

9.18 Covariance Analysis Results for the X-Axis Velocity State with Five Satellites and Automotive Grade IMU 249

9.19 Covariance Analysis Results for the Clock Bias State with Five Satellites and Automotive Grade IMU 250

9.20 Covariance Analysis Results for the Clock Drift State with Five Satellites and Automotive Grade IMU 250

9.21 Monte Carlo Simulation Results for the X-Axis Position State with Eleven Satellites and Tactical Grade IMU 254

9.22 Monte Carlo Simulation Results for the X-Axis Velocity State with Eleven Satellites and Tactical Grade IMU 254

9.23 Monte Carlo Simulation Results for the Clock Bias State with Eleven Satellites and Tactical Grade IMU 255

9.24 Monte Carlo Simulation Results for the Clock Drift State with Eleven Satellites and Tactical Grade IMU 256

9.25 Monte Carlo Simulation Results for the X-Axis Position State with Five Satellites and Automotive Grade IMU 257

9.26 Monte Carlo Simulation Results for the X-Axis Velocity State with Five Satellites and Automotive Grade IMU 257

9.27 Monte Carlo Simulation Results for the Clock Bias State with Five Satellites and Automotive Grade IMU 258

9.28 Monte Carlo Simulation Results for the Clock Drift State with Five Satellites and Automotive Grade IMU 258

10.1 Mini Baja Course at NCAT (from Google Earth) 262

10.2 ATV and Hardware Used to Collect Data 263

10.3 Skyplot of Visible Satellites for the First Data Set (from NordNav Receiver) 264

10.4 Estimated C/N_o Ratios for the First Data Set 265

10.5 Estimated Velocities from the VDFLL for the First Data Set 265

10.6 Estimated Accelerations from the VDFLL for the First Data Set 266

10.7 Position Estimates from the VDFLL and NovAtel for the First Data Set . . 266

10.8 Position Estimates from the VDFLL and NovAtel for the First Data Set . . 267

10.9 Position Estimates from the VDFLL and NovAtel for the First Data Set . . 267

10.10 Skyplot of Visible Satellites for the Second Data Set (from NordNav Receiver) 269

10.11 Estimated C/N_o Ratios for the Second Data Set 269

10.12 Estimated Velocities from the VDFLL for the Second Data Set 270

10.13 Estimated Accelerations from the VDFLL for the Second Data Set 271

10.14 Position Estimates from the VDFLL and NovAtel for the Second Data Set . 271

10.15 Position Estimates from the VDFLL and NovAtel for the Second Data Set . 272

10.16 Position Estimates from the VDFLL and NovAtel for the Second Data Set . 272

10.17 Urban Canyon at the Intersection of Peachtree ST. NE and Andrew Young International Blvd. NE, Looking North (from Google Street View) 274

10.18 Skyplot of Visible Satellites for the Atlanta Data (from NordNav Receiver) 276

10.19 Estimated C/N_o Ratios for the Total Atlanta Data Set 276

10.20 Position Estimates of the VDFLL and NovAtel 277

10.21 Position Estimates of the VDFLL and NovAtel Overladed on a Map of Downtown Atlanta (from Google Earth and GPS Visualizer) 279

10.22 Position Estimates from the VDFLL and NovAtel for the Second Data Set (from Google Earth and GPS Visualizer) 281

10.23 Position Estimates from the VDFLL and NovAtel While Inside Extreme Urban Canyon 282

10.24 Estimated C/N_o Ratios While in Urban Canyon 283

10.25 Position Estimates from the VDFLL and NovAtel 283

10.26 Position Estimates of the VDFLL and NovAtel Overladed on a Map of Atlanta (from Google Earth and GPS Visualizer) 285

B.1 C/N_0 Estimated Using Adaptive Filtering Techniques 327
C.1 VDFLL Monte Carlo Simulation Results for 5 Satellites and σ^2= .2 m^2/s^2 . 328
C.2 VDFLL Monte Carlo Simulation Results for 5 Satellites and σ^2= 2 m^2/s^2 . 329
C.3 VDFLL Monte Carlo Simulation Results for 5 Satellites and σ^2= 10 m^2/s^2 329
C.4 VDFLL Monte Carlo Simulation Results for 5 Satellites and σ^2= 50 m^2/s^2 330
C.5 VDFLL Monte Carlo Simulation Results for 5 Satellites and σ^2= 100 m^2/s^2 330
C.6 VDFLL Monte Carlo Simulation Results for 5 Satellites and σ^2= 300 m^2/s^2 331
C.7 VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= .2 m^2/s^2 . 331
C.8 VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= 2 m^2/s^2 . 332
C.9 VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= 10 m^2/s^2 332
C.10 VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= 50 m^2/s^2 333
C.11 VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= 100 m^2/s^2 333
C.12 VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= 300 m^2/s^2 334
C.13 Thermal Noise Analysis Results for 7 Satellites and σ^2= 10 m^2/s^2 335
C.14 Monte Carlo Simulation Results for 7 Satellites and σ^2= 10 m^2/s^2 336
C.15 Thermal Noise Analysis Results for 7 Satellites and σ^2= 50 m^2/s^2 336
C.16 Monte Carlo Simulation Results for 7 Satellites and σ^2= 50 m^2/s^2 337
C.17 Thermal Noise Analysis Results for 7 Satellites and σ^2= 100 m^2/s^2 337
C.18 4.5Monte Carlo Simulation Results for 7 Satellites and σ^2= 100 m^2/s^2 . . . 338
C.19 Thermal Noise Analysis Results for 5 Satellites and σ^2= 10 m^2/s^2 338
C.20 Monte Carlo Simulation Results for 5 Satellites and σ^2= 10 m^2/s^2 339
C.21 Thermal Noise Analysis Results for 5 Satellites and σ^2= 50 m^2/s^2 339

C.22 Monte Carlo Simulation Results for 5 Satellites and σ^2= 50 m^2/s^2 340

C.23 Thermal Noise Analysis Results for 5 Satellites and σ^2= 100 m^2/s^2 340

C.24 Monte Carlo Simulation Results for 5 Satellites and σ^2= 100 m^2/s^2 341

C.25 Monte Carlo Simulation Results for the Y-Axis Position with Eleven Satellites and σ^2=2 $\mathrm{m}^2/\mathrm{s}^2$ 342

C.26 Monte Carlo Simulation Results for the Y-Axis Velocity with Eleven Satellites and σ^2=2 $\mathrm{m}^2/\mathrm{s}^2$ 343

C.27 Monte Carlo Simulation Results for the Z-Axis Position with Eleven Satellites and σ^2=2 $\mathrm{m}^2/\mathrm{s}^2$ 343

C.28 Monte Carlo Simulation Results for the Z-Axis Velocity with Eleven Satellites and σ^2=2 $\mathrm{m}^2/\mathrm{s}^2$ 344

C.29 Monte Carlo Simulation Results for the X-Axis Position with Eleven Satellites and σ^2=10 $\mathrm{m}^2/\mathrm{s}^2$ 344

C.30 Monte Carlo Simulation Results for the X-Axis Velocity with Eleven Satellites and σ^2=10 $\mathrm{m}^2/\mathrm{s}^2$ 345

C.31 Monte Carlo Simulation Results for the Y-Axis Position with Eleven Satellites and σ^2=10 $\mathrm{m}^2/\mathrm{s}^2$ 345

C.32 Monte Carlo Simulation Results for the Y-Axis Velocity with Eleven Satellites and σ^2=10 $\mathrm{m}^2/\mathrm{s}^2$ 346

C.33 Monte Carlo Simulation Results for the Z-Axis Position with Eleven Satellites and σ^2=10 $\mathrm{m}^2/\mathrm{s}^2$ 346

C.34 Monte Carlo Simulation Results for the Z-Axis Velocity with Eleven Satellites and σ^2=10 $\mathrm{m}^2/\mathrm{s}^2$ 347

C.35 Monte Carlo Simulation Results for the Clock Bias with Eleven Satellites and σ^2=10 $\mathrm{m}^2/\mathrm{s}^2$ 347

C.36 Monte Carlo Simulation Results for the Clock Drift with Eleven Satellites and σ^2=10 $\mathrm{m}^2/\mathrm{s}^2$ 348

C.37 Monte Carlo Simulation Results for the Y-Axis Position with Eleven Satellites and σ^2=50 $\mathrm{m}^2/\mathrm{s}^2$ 348

C.38 Monte Carlo Simulation Results for the Y-Axis Velocity with Eleven Satellites and σ^2=50 m^2/s^2 349

C.39 Monte Carlo Simulation Results for the Z-Axis Position with Eleven Satellites and σ^2=50 m^2/s^2 349

C.40 Monte Carlo Simulation Results for the Z-Axis Velocity with Eleven Satellites and σ^2=50 m^2/s^2 350

C.41 Monte Carlo Simulation Results for the Y-Axis Position with Five Satellites and σ^2=2 m^2/s^2 351

C.42 Monte Carlo Simulation Results for the Y-Axis Velocity with Five Satellites and σ^2=2 m^2/s^2 351

C.43 Monte Carlo Simulation Results for the Z-Axis Position with Five Satellites and σ^2=2 m^2/s^2 352

C.44 Monte Carlo Simulation Results for the Z-Axis Velocity with Five Satellites and σ^2=2 m^2/s^2 352

C.45 Monte Carlo Simulation Results for the Clock Bias with Five Satellites and σ^2=2 m^2/s^2 353

C.46 Monte Carlo Simulation Results for the Clock Drift with Five Satellites and σ^2=2 m^2/s^2 353

C.47 Monte Carlo Simulation Results for the X-Axis Position with Five Satellites and σ^2=10 m^2/s^2 354

C.48 Monte Carlo Simulation Results for the X-Axis Velocity with Five Satellites and σ^2=10 m^2/s^2 354

C.49 Monte Carlo Simulation Results for the Y-Axis Position with Five Satellites and σ^2=10 m^2/s^2 355

C.50 Monte Carlo Simulation Results for the Y-Axis Velocity with Five Satellites and σ^2=10 m^2/s^2 355

C.51 Monte Carlo Simulation Results for the Z-Axis Position with Five Satellites and σ^2=10 m^2/s^2 356

C.52 Monte Carlo Simulation Results for the Z-Axis Velocity with Five Satellites and σ^2=10 m^2/s^2 356

C.53 Monte Carlo Simulation Results for the Clock Bias with Five Satellites and σ^2=10 m^2/s^2 357

C.54 Monte Carlo Simulation Results for the Clock Drift with Five Satellites and σ^2=10 m^2/s^2 358

C.55 Monte Carlo Simulation Results for the Y-Axis Position with Five Satellites and σ^2=50 m^2/s^2 359

C.56 Monte Carlo Simulation Results for the Y-Axis Velocity with Five Satellites and σ^2=50 m^2/s^2 359

C.57 Monte Carlo Simulation Results for the Z-Axis Position with Five Satellites and σ^2=50 m^2/s^2 360

C.58 Monte Carlo Simulation Results for the Z-Axis Velocity with Five Satellites and σ^2=50 m^2/s^2 360

C.59 Monte Carlo Simulation Results for the Clock Bias with Five Satellites and σ^2=50 m^2/s^2 361

C.60 Monte Carlo Simulation Results for the Clock Drift with Five Satellites and σ^2=50 m^2/s^2 361

C.61 Covariance Analysis Results for the Y-Axis Position State with Eleven Satellites and Tactical Grade IMU 362

C.62 Covariance Analysis Results for the Y-Axis Velocity State with Eleven Satellites and Tactical Grade IMU 363

C.63 Covariance Analysis Results for the Z-Axis Position State with Eleven Satellites and Tactical Grade IMU 363

C.64 Covariance Analysis Results for the Z-Axis Velocity State with Eleven Satellites and Tactical Grade IMU 364

C.65 Covariance Analysis Results for the Y-Axis Position State with Five Satellites and Automotive Grade IMU 365

C.66 Covariance Analysis Results for the Y-Axis Velocity State with Five Satellites and Automotive Grade IMU 366

C.67 Covariance Analysis Results for the Z-Axis Position State with Five Satellites and Automotive Grade IMU 366

C.68 Covariance Analysis Results for the Z-Axis Velocity State with Five Satellites and Automotive Grade IMU 367

C.69 Monte Carlo Simulation Results for the Y-Axis Position State with Eleven Satellites and Tactical Grade IMU 368

C.70 Monte Carlo Simulation Results for the Y-Axis Velocity State with Eleven Satellites and Tactical Grade IMU 369

C.71 Monte Carlo Simulation Results for the Z-Axis Position State with Eleven Satellites and Tactical Grade IMU 369

C.72 Monte Carlo Simulation Results for the Z-Axis Velocity State with Eleven Satellites and Tactical Grade IMU 370

C.73 Monte Carlo Simulation Results for the Y-Axis Position State with Five Satellites and Automotive Grade IMU 371

C.74 Monte Carlo Simulation Results for the Y-Axis Velocity State with Five Satellites and Automotive Grade IMU 372

C.75 Monte Carlo Simulation Results for the Z-Axis Position State with Five Satellites and Automotive Grade IMU 372

C.76 Monte Carlo Simulation Results for the Z-Axis Velocity State with Five Satellites and Automotive Grade IMU 373

D.1 Federated Filter Architecture with Decoupled-Channel Residual Filters . . . 375

Chapter 1

Introduction

1.1 Motivation

The NAVSTAR Global Positioning System (GPS) was developed in the 1970's by the Department of Defense (DoD). The purpose of the GPS system was to provide the U.S. military with accurate measurements of position, velocity, and time (PVT). The DoD desired position estimates with a root-mean-square (rms) error of ten meters, velocity estimates with an rms error of .1 m/s, and time estimates with an rms error of 100 ns, [Misra and Enge, 2001]. The system was also equipped with a less accurate, civilian component that non-military user's could use. Today the U.S. military uses the GPS system extensively. It is used for a myriad of tasks from as simple as personal navigation, to the guidance of precision munitions. Civilian usage of the GPS system has also become widespread. Many automobiles come equipped with GPS receivers and hand held receivers have become commonplace among joggers, hunters, and hikers. Industrial applications of the GPS system range from land surveying, to precision agriculture, to its usage in the finance system as a timing standard.

Despite its widespread application and usage, the GPS system has several inherent shortcomings. The primary foible of the system is the power level of the received signals. For the civilian user, the power of the received signals is approximately 10^{-16} watts, [Misra and Enge, 2001]. The low power level of the GPS signal makes it very susceptible to intentional and unintentional jamming. In fact, only about 3 pW of wideband RF interference is required to disable a civilian GPS receiver, [Ward, 1996a]. A wideband jammer with an Effective Radiated Power (ERP) of only 5 Watts can incapacitate civilian receivers within about a 10 mile radius [Ward, 1996a]. Military receivers are more robust in terms of their jamming immunity, but are still relatively easy to jam. A 100 Watt jammer can disable military receivers within a 25 km radius [Gustafson et al., 2000b]. In addition to being highly effective, GPS jammers are inexpensive to make. A jammer with several hundred watts of power can be manufactured at a cost of 25 dollars or less [Gustafson et al., 2000b]. In hostile environments, jamming represents a major threat to the GPS system. Civilian receivers are much less likely to encounter intentional jamming or RF interference. However, civilian receivers will lose signal lock and cease to function in environments with dense overhead foliage or in urban canyons.

Another shortcoming of GPS, and satellite based navigation systems in general, is the need to maintain line of sight between the user's antenna and the satellites. Blockage of the satellite signal disables the GPS receiver. During periods of blockage, the tracking loops inside the receiver are in a state of random walk [Dierendonck, 1996]. After prolonged periods of signal unavailability, the receiver must re-enter the time consuming acquisition stage before it can reinitialized its tracking loops. The tracking loops inside the receiver are also sensitive to the dynamics of the receiver. High dynamics can cause the tracking loops to fail.

In an effort to improve GPS availability, various methods of increasing the robustness of receivers have been investigated. The two methods investigated in this dissertation are advanced signal tracking algorithms and the fusion of GPS with other sensors. More specifically, vector based tracking algorithms are studied and different methods of incorporating

Inertial Measurement Units (IMU's) with GPS receivers are investigated (known as Deep Integration or Ultra-Tight Coupling).

1.2 Prior Work

In this section, prior work in vector based signal processing and aided receivers is covered.

1.2.1 Vector Tracking

Vector tracking algorithms are an advanced method of processing the received GPS signals. They have the ability to function at lower carrier to noise power density (C/N_o) ratios and in higher dynamics than traditional GPS receivers. Receivers must track the signals broadcast from at least four GPS satellites in order to navigate. Traditional receivers track the signals from different satellites independent of one another using tracking loops. Instead of processing the signals separately, vector tracking algorithms use a single Kalman filter to simultaneously track the GPS signals and determine the user's position, velocity, etc. Vector based methods combine the tasks of signal tracking and navigation into a single algorithm.

Tracking the GPS signals together, rather than separately, gives the vector tracking algorithms several advantages. The task of tracking the user's dynamics is accomplished jointly, by all the channels together. Using standard tracking loops, each channel must track the line-of-sight dynamics of the receiver by itself. The received signal power may also be too low for the tracking loops in each channel to operate. However, there can still be sufficient aggregate signal power in the sum of the channels to allow the vector based methods to operate. Another advantage of the vector based methods is that the user's dynamics can be constrained to be in different dimensions [Spilker, Jr., 1996a]. For applications such as pedestrian, ship, and automotive navigation, the majority of the receiver dynamics occur in only two axes. Vector tracking algorithms can account for constrained user dynamics.

The concept of vector tracking originates in the early 1980's [Copps et al., 1980] [Sennott, 1984]. In the 1990's, researchers from Bradley University published several papers using a GPS receiver with coupled signal tracking channels to achieve improved performance [Sennott and Senffner, 1997], [Sennott and Senffner, 1995], [Sennott and Senffner, 1992b], [Sennott and Senffner, 1992a]. The most commonly cited reference for vector tracking appeared in 1996 [Spilker, Jr., 1996a], where the Vector Delay Lock Loop (VDLL) is introduced and many of the commonly cited advantages of vector tracking are listed. However, few practical details of implementing the VDLL are included. A host of papers on vector tracking have appeared since the turn of the century. Researchers from the University of Calgary investigated different methods of estimating the phases of the received carrier signals for a federated vector tracking architecture in [Petovello and Lachapelle, 2006]. The robustness of the a Vector Delay/Frequency Lock Loop (VDFLL) is demonstrated in [Pany et al., 2005]. The ability of the VDFLL to operate in heavy foliage and reacquire blocked signals is studied in [Lashley and Bevly, 2008], [Lashley and Bevly, 2007], and [Kiesel et al., 2007]. The VDFLL is used to estimated the power of weak signals in [Pany and Eissfeller, 2006]. A detailed covariance analysis of the VDLL is performed in [Benson, 2007]. In [Benson, 2007], the average interference benefit of the VDLL over the scalar Delay Lock Loop (DLL) is quoted as being from 3 to 5.9 dB for varying satellite geometries. In [Lashley et al., 2008], the authors demonstrated the ability of the VDFLL to operate through 2 G and 4 G turns at 16 dB-Hz.

1.2.2 Aided Receivers

Additional sensors are often fused with GPS to produce a more robust navigation system. The most common sensors used are inertial measurement units (IMU's). Other sensors include dopplerometers, altimeters, speedometers, and odometers, [Bullock et al., 2006]. Inertial sensors provide acceleration measurements and angular rate measurements in the reference frame that they are mounted. Integrating the IMU measurements produces estimates of the user's change in velocity and change in position. The strengths of IMU's are that they can provide updates at several hundred Hertz and are immune to jamming. However, IMU's have inherit flaws of their own. They suffer from biases and drifts that cause errors to accumulate over time. Additionally, IMU's contain scale factors that must be estimated in order to determine the true accelerations and angular rates experienced, [Flenniken, IV, 2005] [Pethel, 2006].

The strengths and weaknesses of IMU's and GPS complement each other. The measurements provided by an IMU serve to update the user's PVT between GPS updates. During times of satellite occlusion, the IMU can be used to dead reckon and can help in re-initializing the tracking loops when the satellites reappear. On the other hand, the GPS measurements can be used to correct the drift in the dead reckoned IMU position estimates. The bias's and scale factors in the IMU's can also be estimated at the GPS updates, [Godha, 2004].

The Kalman filter is the default method of integrating GPS and IMU measurements. Combining the two systems through a Kalman filter provides overall improved performance, [Levy, 1997]. However, there is always the question of how exactly to fuse the two systems. Currently, the various architectures used to meld GPS and IMU systems fall into three categories, [Kreye et al., 2004]. The three classes of coupling, in order of complexity, are loose coupling, tight coupling, and ultra-tight coupling (UTC), which is also referred to as deeply integrated (DI).

Loose Coupling

Most integrated navigation systems are loosely coupled. The loosely coupled architecture is the simplest coupling scheme and provides significantly better performance than stand alone GPS or INS systems. Figure 1.1 shows a block diagram of a loosely coupled system. The two different systems are used to separately produce estimates of the platform's navigation states. The GPS and INS estimates are then combined by a Kalman filter. The Kalman filter produces improved state estimates and estimates the error states of the INS system, [Kreye et al., 2004]. An advantage of the loose coupling method is that the GPS receiver itself does not need to be modeled extensively. However, the GPS receiver contains internal filters and therefore the noise in the provided GPS measurement is not white. The colored noise means the Kalman filter is no longer an optimal observer and requires additional tuning. In addition, a loosely coupled system only receives GPS measurements if there is enough information to complete a full, GPS alone, position solution, [Gautier, 2003]. In situations where fewer than four satellite signals are available, no measurements are provided by the GPS receiver to the Kalman filter and the navigation solution reverts to dead reckoning.

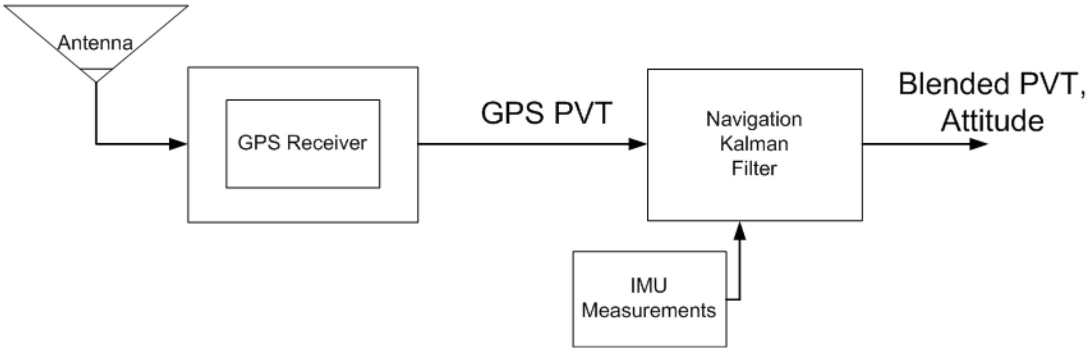

Figure 1.1: Loosely Coupled Architecture

Tight Coupling

Tightly coupled systems combine the inertial sensors and GPS at a deeper level. Figure 1.2 shows a diagram of a generic tightly coupled system. The navigation filter receives the raw pseudorange and pseudorange-rate measurements from the GPS receiver. This allows the filter to still receive information from the GPS receiver when there are less than four satellites available. Many tightly coupled systems also use the acceleration information provided by the INS to predict the Doppler shift caused by the user's dynamics. This information is then fed back to the tracking loops. This improves the performance of the GPS receiver's tracking loops and their ability to maintain lock in high dynamic environments, [Kreye et al., 2004]. The bandwidths of the tracking loops can also be narrowed because of the reduced need to track unknown user dynamics. The main drawback of the tightly coupled system is its complexity [Gautier, 2003]. Substantial improvements in navigation accuracy are obtained through tight coupling [Cox, 1978]. In the case of degraded satellite coverage, the performance of loosely coupled systems can be significantly worse than that of a tightly coupled system [Weiss and Kee, 1995].

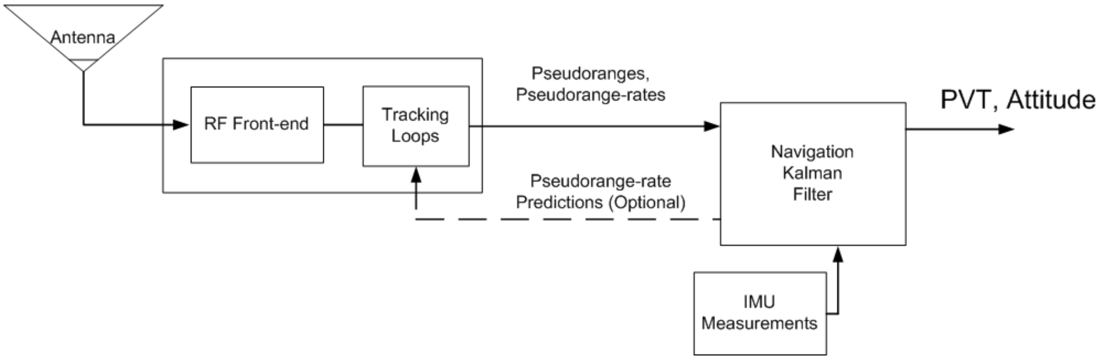

Figure 1.2: Tightly Coupled Architecture

The robustness of the tracking loops to interference and jamming can be improved if they are Doppler-aided. In [Gebre-Egziabher et al., 2005], the authors report an increase of 4 dB in carrier phase tracking threshold with Doppler-aiding. The analysis that was performed assumed a nominal grade receiver clock (TCXO) and a tactical grade IMU. The authors report that at low tracking loop bandwidths the dominant noise sources are oscillator instabilities and vibration-induced phase error. Other authors have reported a 14 dB improvement in phase noise suppression when using Doppler-aided, narrow bandwidth tracking loops versus non-aided loops [Alban et al., 2003] [Alban and Dietrich, 2004].

When the bandwidths of the tracking loops are reduced, their ability to reject noise improves. However, the decorrelation times associated with errors in the tracking loops increases. This fact has major consequences for the central navigation filter. The reason is that the noise present in the measurements provided to the Kalman filter is assumed to be white. Time correlated errors in the tracking loops can lead to filter divergence if not properly modeled.

Deep Integration and Ultra-tight Coupling

Ultra-tight coupling, or deep integration, is the most complicated and involved form of GPS/INS fusion. It takes the general architecture of vector tracking and adds to it an IMU system. Figure 1.3 shows a general block diagram of a DI/ultra-tight system. In a DI navigation system, the INS and the GPS receiver are tied together at the most basic level. The raw IMU measurements are fed directly into the navigation Kalman filter. The filter then predicts the pseudorange and pseudorange-rates. The filter's predictions are then used to control the inputs to the integrate and dump functions inside the GPS receiver. The signals produced by the integrate and dump functions are then processed by the filter to estimate the error's in the IMU related states. In a complete DI system, the task of tracking the GPS signals is accomplished by the navigation filter itself, [Kreye et al., 2004].

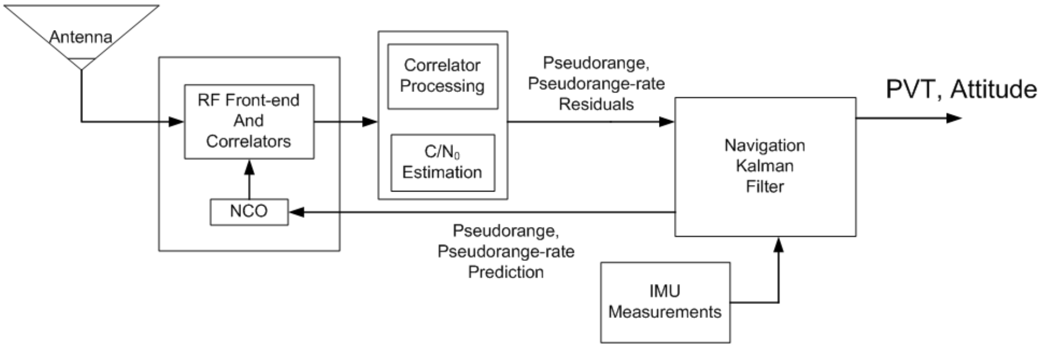
Antenna
RF Front-end And Correlators
NCO
Correlator Processing
C/N$_0$ Estimation
Pseudorange, Pseudorange-rate Residuals
PVT, Attitude
Navigation Kalman Filter
Pseudorange, Pseudorange-rate Prediction
IMU Measurements

Figure 1.3: Ultra-Tightly Coupled Architecture

The original concept of ultra-tight coupling/DI is found in [Copps et al., 1980]. In [Copps et al., 1980], an integrated signal tracker/navigation algorithm is formulated. The GPS satellite signals are not tracked individually in their own channels, but instead are processed collectively by a single algorithm. The DI algorithm relies on the fact that the pseudorange and pseudorange-rate observed by the user are defined by the user's position and velocity, and the satellite geometry. The satellite geometry is determined by using the broadcast satellite ephemeris. Standard GPS receiver's determine the user's position and velocity from the received signals. In a DI system, the GPS signals are predicted based on the user's position and velocity. Residuals in the correlator outputs are then used to update the user's estimated position and velocity.

The authors of [Copps et al., 1980] compare their integrated tracking algorithm to a more traditional partitioned system. Both systems rely on code tracking only because the carrier signal tracking is assumed to be lost. The partitioned system is essentially tight coupling with inertial aiding to the Delay Lock Loops (DLL's). The systems are compared through two dimensional simulations with three satellites available, with various levels of jamming present. Both algorithms function at 60 dB of jammer-to-signal power (J/S), but fail at 65 dB. A 2 dB improvement in J/S is observed in the integrated system, relative to the partitioned system. The integrated system is also found to be better suited to high dynamic maneuvers at high J/S levels. Researchers at Bradley University also published several early papers using coupled signal tracking/navigation algorithms [Sennott and Senffner, 1992b], [Sennott and Senffner, 1995], [Sennott and Senffner, 1992a]. Performance improvements of 8 to 15 dB are reported for the coupled signal tracking/navigation algorithm over traditional methods [Sennott and Senffner, 1992a].

The Raytheon company obtained a patent for their method of performing Ultra-Tight Coupling (UTC) in 1998, [Horslund and Hooker, 1999]. Figure 1.4 shows a the block diagram of their method included in the patent. The Raytheon method uses a NAV function to estimate the user's navigation states in the earth centered, earth fixed (ECEF) coordinate frame. The NAV function is driven by the output of a GPS residual Kalman filter and IMU. These states include user position, velocity, IMU alignment states, and initial attitude. The estimated state vector of the NAV function is used by a line of sight (LOS) geometry function to predict the pseudorange and pseudorange-rate for each individual satellite being tracked. The LOS function then sends the predictions to a signal processing chip (SPC) that performs the integrate and dump functions for each channel. The outputs of the integrate and dump functions are then used to form pseudorange and pseudorange-rate residuals. These residuals are then processed by a Kalman filter. The Kalman filter estimates corrections to the navigation states from the GPS residuals. These corrections are in turn used by the NAV function to close the loop.

The increase in jamming immunity offered by the Raytheon method is quoted with and without Data Wipe-Off (DWO) being used for P(Y) code tracking. Data wipe-off is a technique that effectively predicts the navigation data bits and allows for integrate and dump times greater than 20 milliseconds. Carrier and code tracking is reported to be possible with 15 dB more jamming when using DWO and 6 dB more jamming without DWO [Horslund and Hooker, 1999]. The increase in tolerable jamming is quoted relative to the state of the art. Using a low grade ring laser gyro, quality carrier measurements are quoted as being available for carrier phase and frequency measurements at J/S levels of 70 to 75 dB [Horslund and Hooker, 1999]. Code tracking is also reported possible in the range of

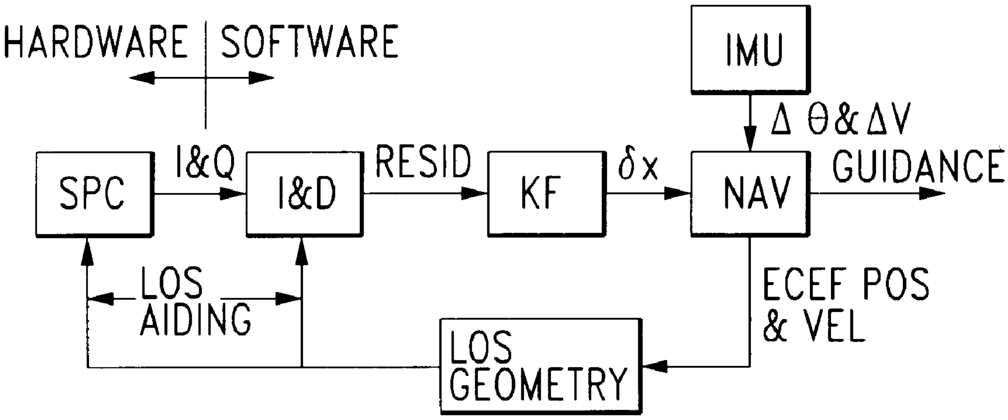

Figure 1.4: Raytheon Method of Ultra-Tight Coupling [Horslund and Hooker, 1999]

75 dB J/S [Horslund and Hooker, 1999]. In [Lewis, 2003], Raytheon's UTC algorithm was compared with a tightly coupled system in different jamming scenarios and with different grade IMU's. A 6 to 8 dB improvement in jamming performance was observed for the UTC algorithm relative to the tightly coupled system when using a poor quality Temperature Controlled Oscillator (TCXO) and IMU. A series of papers by Raytheon scientists have also studied the effects of antenna arm flexure on UTC performance, [Quinn et al., 2005] [Quinn et al., 2006].

Anthony Abbott developed a different ultra-tight method and obtained a patent for it in 1999, [Abbott and Lillo, 2003]. Figure 1.5 shows a block diagram of the his method. The Abbott method operates on similar principles as the Raytheon method. The main difference is that Abbott uses a federated Kalman filtering scheme, [Carlson, 1988]. A large integration Kalman filter updates the error state vector every second. The previous epoch's error state vector and the IMU samples are used between epochs of the integration filter by

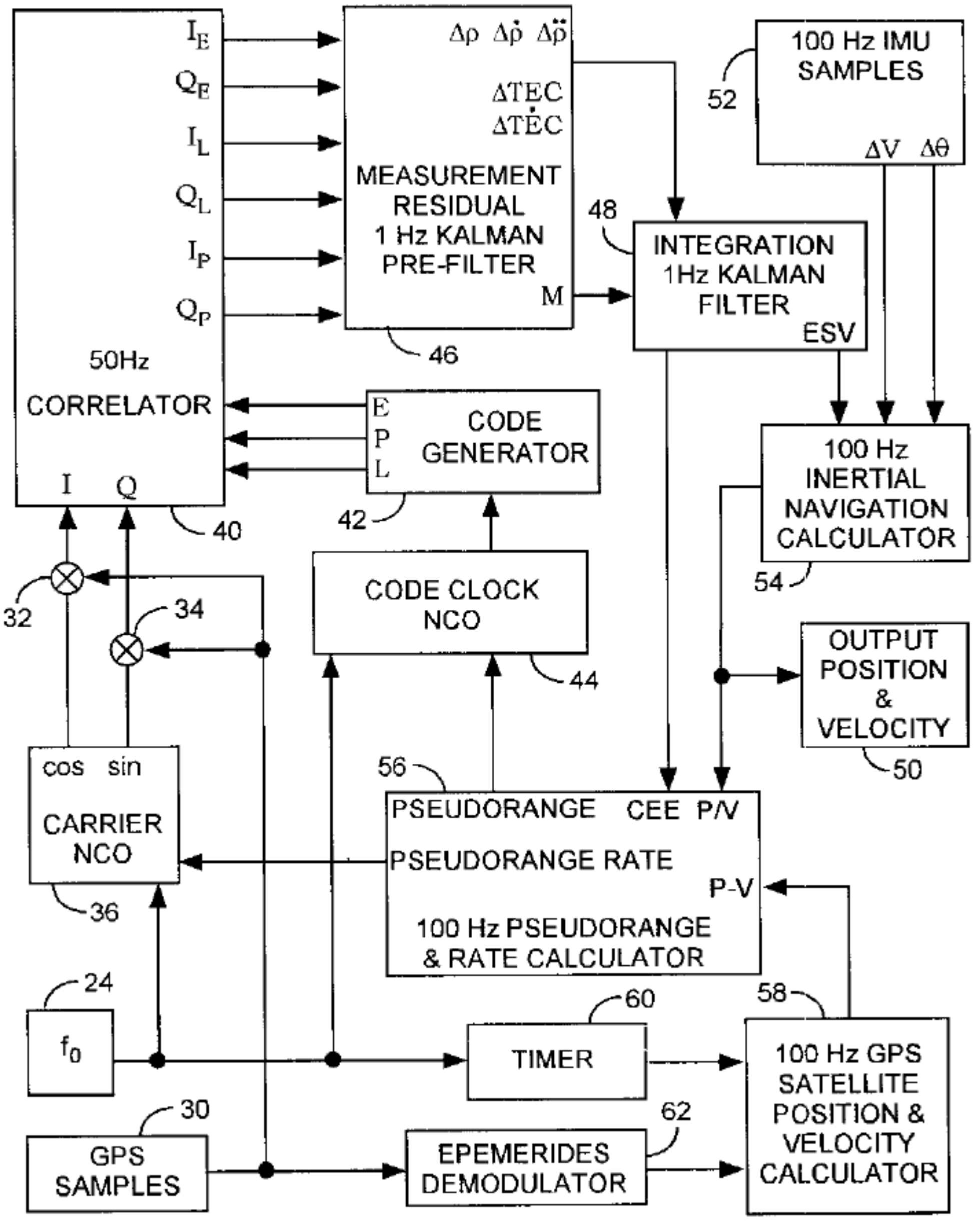

Figure 1.5: Anthony Abbott Method of Ultra-Tight Coupling [Abbott and Lillo, 2003]

the inertial navigation calculator. The inertial navigation calculator provides estimates of the user's navigation states. These estimates are used by the pseudorange and pseudorange-rate calculator to predict the inputs to the 50 Hz correlator bank. The output of each bank of correlators is processed by a Kalman prefilter. A Kalman prefilter exits for each satellite being tracked. The prefilters process the outputs of their respective correlators between updates of the integration filter. At measurement epochs of the integration filter, each prefilter provides the integration filter with measurements of pseudorange, pseudorange-rate, and pseudorange acceleration. The prefilters also provide an empirically determined three by three measurement covariance matrix. The integration filter processes the measurements of the smaller prefilters and updates the error state vector. The updated error state vector is then used until the next measurement epoch of the integration filter. In [Ohlmeyer, 2006], the jamming immunity of an ultra-tightly coupled system based on [Abbott and Lillo, 2003] is tested in various jamming scenarios. The test scenario has the receiver flying toward a jammer broadcasting in the P(Y) code 20-Mhz bandwidth. The power level of the jammer is varied from 10 to 150 W. The effective C/N_0 ratio varies as the receiver approaches the jammer. Using a medium grade Micro Electro-Mechanical (MEMS) IMU, the UTC algorithm begins to degrade at J/S levels around 75 dB for P(Y) code tracking. The fact that the C/N_0 ratio is not held constant makes ascertaining the maximum tolerable J/S of the system difficult. A side by side comparision with a tightly coupled system was also not performed.

The Charles Stark Draper Laboratory patented a DI algorithm in 2001, [Gustafson et al., 2001]. Figure 1.6 shows a block diagram of the Draper method. The Draper method is similar in basic concept to the Raytheon and Abbott methods. The

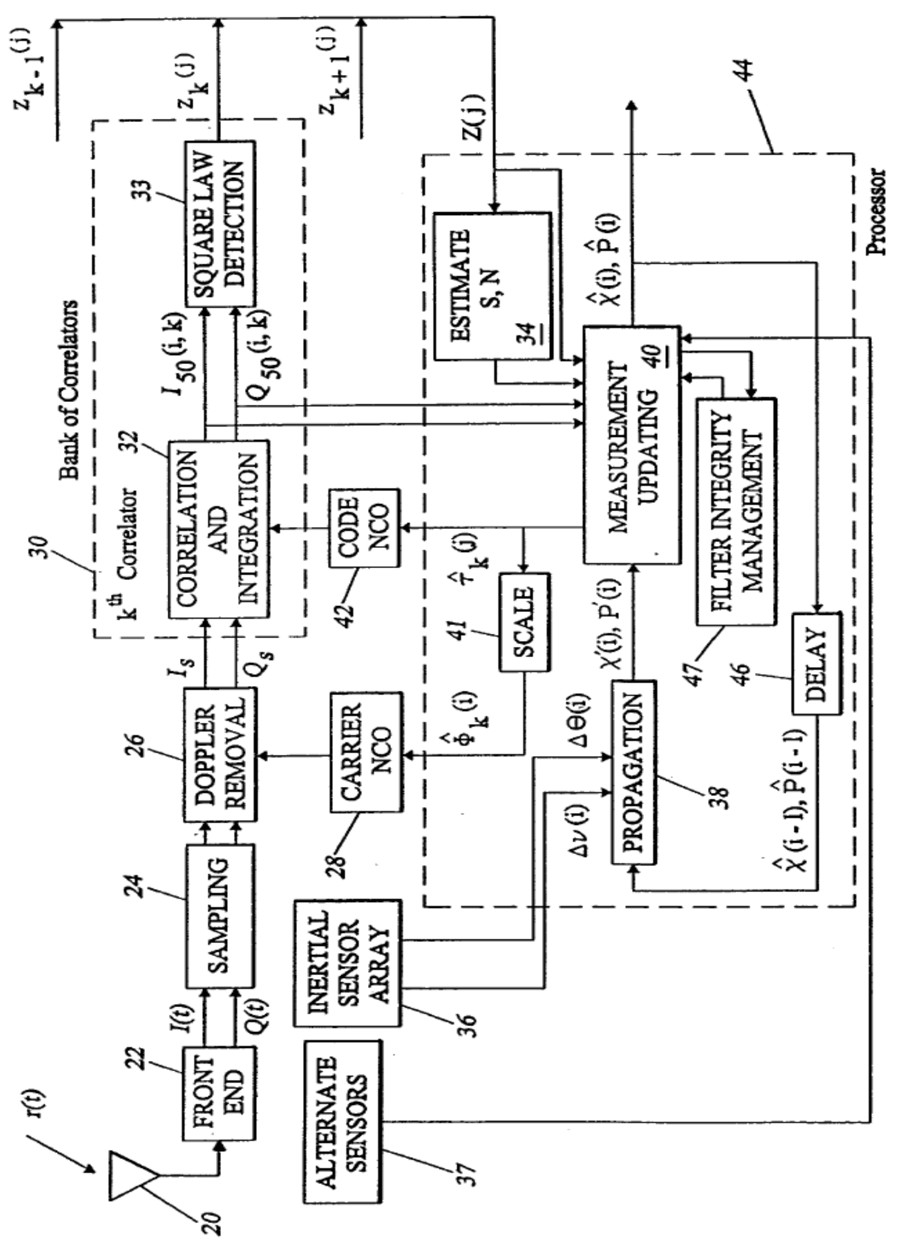

Figure 1.6: Draper Method of Deep Integration [Gustafson et al., 2001]

inputs to the correlation and integration block are derived from the user's navigation state. The outputs of the correlation and integration block are then used to update the navigation state vector. The major distinction in the Draper method is the filtering algorithm used to process the GPS and IMU measurements. The navigation filter techniques used by Draper "are significant departures from traditional Kalman and extended Kalman filter algorithms generally used for GPS-based navigation." The state and covariance propagation step of their filter is essentially the same as the Extended Kalman filter, however their filter differs from the EKF in how it performs the GPS measurement update. Draper Labs also makes use of an extended bank of range correlators [Gustafson et al., 2000c]. The majority of traditional code discriminators have a linear operating range of +/- one half chip of code phase error. The extended bank of range correlators is used to increase the detector range beyond one half chip. The disadvantage of using multiple correlators is that it increases the amount of noise present in the discriminator output [Bowles, 1980].

The performance enhancement offered by Draper's algorithm is studied in a series of papers authored by Draper scientists themselves. The benefits of their algorithm are quoted as being 15 to 20 dB improvement in code tracking for wideband jamming over traditional tight coupling [Gustafson et al., 2000b], [Gustafson et al., 2000a]. The performance of both algorithms was assessed through Monte Carlo simulations, over which the magnitude of the jamming was held constant. The hardware testbed used to analyze the DI algorithm is described in [Flueckiger et al., 2005]. A third paper compared the code tracking abilities of Draper's DI algorithm to a fixed-gain and adaptive gain code tracking loop

[Gustafson and Dowdle, 2003]. An overall lack of block diagrams and equations makes interpreting the results in [Gustafson and Dowdle, 2003] very difficult. The Draper DI algorithm has also been used for pedestrian navigation and indoor positioning [Landis et al., 2006].

Researchers at Ohio University have developed, implemented, and tested their own DI algorithms [Soloviev et al., 2004b], [Soloviev et al., 2008], [Gunawardena et al., 2004] [Soloviev et al., 2004a]. They have reported carrier phase tracking at 15 dB-Hz with an oven-controlled crystal oscillator (OCXO) and low cost inertial sensors. The algorithms' improvement in performance over [Gebre-Egziabher et al., 2005] stems mainly from extending the pre-detection integration time beyond the 20 ms data bit period to about 1 s. Prior knowledge of the data bits is not assumed and an energy-based bit estimation algorithm is used to detect bit transitions during the integration period. The Ohio University researchers have also applied their DI algorithms for tracking weak GPS signals in urban environments [Soloviev et al., 2007] and for constructively using multipath signals in urban canyons [Soloviev and van Graas, 2008].

Several other companies have developed their own DI algorithms, including L-3 communications [Besser et al., 2002] [Crane, 2007], KVH and NovAtel [Kennedy and Rossi, 2008], Honeywell/Rockwell Collins [Buck et al., 2006], the Center for Remote Sensing [Jovancevic and Ganguly, 2005] [Dyrud et al., 2007] [Woessner et al., 2006], and Tracking and Imaging Systems [Sennott and Senffner, 1997]. These references are primarily product announcements with limited analysis of the DI systems. Several other researchers have similarly published alternate formulations of DI algorithms [Kim et al., 2003], [Ziedan, 2006a], [Li and Wang, 2006].

Researchers at QinetiQ published an article on an Adaptive Tightly Coupled (ATC) algorithm in 2003 [Groves and Long, 2003]. The ATC algorithm uses inertial aiding of the tracking loops and a central Kalman filter to estimate errors in the inertial sensors. The distinction in their algorithm is that the bandwidths of the tracking loops are adapted for varying C/N_o ratios. Reducing the bandwidths causes the correlation times associated with errors in the tracking loops to increase, therefore the assumptions of the Kalman filter are violated when the noise present in the measurements is time correlated. The ATC algorithm uses several different methods for dealing with the correlated measurement noise that is encountered when the tracking loop bandwidths are significantly narrowed. The authors of the ATC algorithm also make comparisons between it and deep integration. According to the QinetiQ researchers, "deep integration brings an improvement in anti-jam margin of up to 2dB over that achievable with the ATC technique, ...".

In another paper, QinetiQ researchers demonstrated the weak signal tracking abilities of a noncoherent DI algorithm [Groves et al., 2007]. They made several relevant observations when testing their system. Their first observation is that DI has an inherent ability to bridge signal outages. This is due to changing the measurement noise covariance matrix in weak signal conditions. During periods of intense jamming, the GPS residuals are effectively ignored and the IMU is basically used to dead reckon through the signal outage. When the jamming abates, the GPS derived measurements are then used to correct the accumulated IMU errors (provided that the dead reckoned position has not drifted to far). This ability to bridge outages can distort the results of Monte Carlo simulations. For example, the QinetiQ researchers first determined the maximum sustained J/S their DI algorithm could tolerate. Then, they performed a series of simulations where the J/S level was increased in steps of about 1 dB every 25 s. The DI algorithm did not lose lock until the J/S was about 3 dB above the maximum sustainable J/S. The QinetiQ researchers concluded that simulations involving continually increasing J/S levels must be interpreted with caution [Groves et al., 2007].

A second observation made in [Groves et al., 2007] is the effect that C/N_0 ratio estimation has on DI's performance. The measurement noise covariance matrix of the DI Kalman filter is based on the estimated C/N_0 ratio. At low signal levels, measured C/N_0 ratio's must be averaged for several seconds to yield accurate estimates. For C/N_0's less than 10 dB-Hz, an averaging time of at least 5 s must be used [Groves et al., 2007]. The QinetiQ researchers performed a series of three trials. In one trial, the DI algorithm used an adaptive averaging time that was based on the current estimated C/N_0. This setup was used to determine the code tracking threshold of the DI algorithm. Another trial used a fixed 1 s averaging time, and yet another trial actually provided the DI algorithm with the true C/N_0 ratio used in the simulation. Using the fixed 1 s averaging time reduced the code tracking threshold by nearly 2 dB, and using the true value of C/N_0 improved the code tracking threshold by about 3 dB [Groves et al., 2007]. From their simulations, the QinetiQ researchers concluded that C/N_0 ratio estimation is a key limiting factor on the ability of DI algorithms to function at low C/N_0 ratios [Groves et al., 2007].

1.2.3 Deficiencies in the Current Understanding of Deeply Integrated GPS/INS Systems

The amount of jamming immunity offered by deep integration is a major unresolved question. The quoted benefits of DI relative to tight coupling vary greatly. Optimistic estimates range from 15 to 20 dB [Gustafson et al., 2000b], while pessimistic estimates put the increase at a maximum of 2 dB [Groves and Long, 2003]. The technical details of the algorithms and the circumstances under which they were tested are also absent from the published literature. The exact manner in which the algorithms are tested has a significant effect on their performance [Groves et al., 2007].

The majority of papers on DI focus solely on its performance at low C/N_0 ratios. A comparision in the performance of a DI algorithm to the performance of a tightly coupled algorithm in a degraded satellite geometry scenario has not been performed. It is unclear whether DI can perform better than a tightly coupled system in situations with three or fewer satellites.

A fundamental question that has yet to be resolved is the exact reason why DI performs better than tight coupling. It is often asserted that the lack of intermediate tracking loops is the reason for DI's improved performance. This assertion has not been proved mathematically in any of the published reports on DI. A heuristic argument, based on the decentralized nature of tracking loops, is generally offered. The idea is based on the fact that tracking loops use measurements from a single satellite to estimate the pseudorange and pseudorange-rate to that satellite. An overall improvement in signal tracking should occur when all the channels of the receiver are tied together through the navigation states of the user. Instead of twelve delay lock loops attempting to estimate twelve separate pseudoranges, four variables (three positions and a clock bias) are estimated using twelve measurements in a vector tracking architecture.

If the improvements offered by DI are from the elimination of tracking loops, it raises the question of whether federated DI architectures are superior to tight coupling. The Tony Abbott method of deep integration relies on a bank of small prefilters [Abbott and Lillo, 2003]. A prefilter exist for each satellite and processes the correlator output for that specific satellite. A central filter then uses the states of the smaller filters to estimate the IMU error states. These prefilters are essentially the same as tracking loops and should therefore incur the same loss of optimality.

The use of tracking loops also introduces correlated errors in the pseudorange and pseudorange-rate measurements. The measurement updates of the central Kalman filter are generally separated by enough time to guarantee that the noise present in the measurements is not time correlated. Between measurement updates of the central filter, multiple integrate and dump cycles occur. Therefore, maximum usage of the information present in the correlator outputs does not occur due to the delayed update times of the central filter. In other words, information is being wasted while waiting for the noise in the tracking loops to become uncorrelated. In a non-federated DI architecture, such as the Draper and Raytheon methods, the correlator outputs are used directly by the filter as they become available. No information is "thrown away" between update cycles of the central filter.

1.3 Contributions

The research presented in this dissertation studies the advantages offered by vector tracking algorithms over traditional architectures. A combination of analysis, Monte Carlo simulations, and RF data is used to study vector tracking algorithms. First, the origin of vector tracking's performance advantage relative to scalar tracking is explained using weighted least squares. Second, two distinct formulations of vector tracking are explored. The two incarnations are the position-state and pseudorange-state formulations of vector tracking. The equivalence in performance of the two algorithms in steady state is proven. By manipulating the structure of the pseudorange-state formulation, a method of making valid comparisons between scalar and vector tracking algorithms is introduced. The comparison method provides a simple way of determining the performance improvement solely from the coupling of the receiver channels together.

Analysis tools for studying the loss of lock performance of vector tracking algorithms are the third contribution of the dissertation. The analysis tools are derived by taking rule-of-thumb tracking thresholds for scalar tracking loops and expanding them to include the vector case. The analysis tools include the effects of thermal noise and receiver dynamics. The analysis tools are verified by using an array of Monte Carlo simulations and data collected from an RF simulator. The Monte Carlo simulations are also used to study the effect of requiring the algorithms to estimate the C/N_0 ratio, as opposed to assuming it is known *a priori*. The simulations are also used study the advantage vector tracking offers over scalar tracking in a variety of different scenarios. The pseudorange-state formulation of vector tracking is used to ensure a valid comparison is made.

The fourth contribution of this dissertation is an analysis of federated vector tracking architectures. This analysis of different architectures is undertaken to determine the performance loss when using a federated architecture, such as in [Abbott and Lillo, 2003]. Three different architectures are studied. The first uses a single central Kalman filter to perform the tasks of signal tracking and navigation state estimation. The second and third architectures both use a federated filtering architecture. In the federated architecture, a master filter and an array of residual filters are used to jointly accomplish the tasks of signal tracking and navigation state estimation. The distinction in the two federated architectures is whether or not the residual filters are coupled together. The pseudorange-state formulation is again used to make a valid comparison between using coupled and decoupled channel residual filters.

The fifth contribution of this dissertation is an analysis of different deeply integrated architectures. The deeply integrated algorithms are formed by taking the three different vector tracking architectures and including an inertial measurement unit. The effects of using a federated filtering architecture with and without the coupling of the channels of the receiver together on DI's performance are analyzed.

The sixth and final contribution of this dissertation is a series of live-sky tests of the vector tracking algorithms. A software GPS receiver is used to collect intermediate frequency (IF) data. The vector tracking algorithms are implemented in MATLAB and are used to process the live-sky data. Data is collected from environments with dense foliage and from urban canyon environments. The performance of the vector tracking algorithms is compared to that of a high-end commercial GPS receiver.

Chapter 2

GPS Signal Structure

2.1 Introduction

This chapter describes the signals within the GPS system and their unique properties that allow for global navigation. The modulation techniques and timing schemes employed by GPS are explained. All GPS receivers rely on the same basic properties of the GPS signals to function, regardless of their complexity or structure. The GPS system is composed of civilian and military components. Due to its availability, more attention is directed at the civilian component of the spectrum. Subsequent chapters will discuss how the signals are processed inside the receiver.

2.2 General Signal Structure

The GPS system is a code division, multiple access (CDMA) digital communication link. Each satellite's signal consists of a sinusoidal carrier, a digital navigation message, and a unique wide bandwidth pseudo random noise (PRN) sequence. The navigation message and PRN sequence (code) are encoded in the phase of the carrier signal by using binary phase shift keying (BPSK) modulation techniques [Sklar, 2001]. The system's multiple satellites all operate at the same frequency while not interfering with each other. This is made possible through the use of unique PRN codes. Additionally, the PRN codes are used to measure the range from each satellite to the user. The data message broadcast by each satellite provides sufficient information to calculate the position of the satellite. Using range measurements to four or more satellite's and using the data message to determine their positions, the user can solve for their position.

Satellite signals currently consist of civilian and military components. The civilian component of the signal is available to all users and can be used for position determination. The military portion of the satellite signal is only available to secure users and provides enhanced positioning capabilities. For much of the work described in this dissertation, only the civilian component of the GPS signal is considered.

The user must obtain the data message to determine the satellite positions. The user must also measure the phase of each of the PRN sequences being received from the satellites. Measuring the PRN phase of each received code is equivalent to measuring the relative range to each satellite. The receiver exploits the properties of the carrier and PRN signals to accomplish these tasks. The autocorrelation properties of the PRN sequences are used to determine the phases of the received PRN codes.

2.3 GPS Carrier Signal

The GPS satellites broadcast at two different center frequencies. The link 1 (L_1) frequency is approximately 1575.42 MHz and the link 2 (L_2) frequency is roughly 1227.6 MHz. The exact frequencies of L_1 and L_2 are shown in (2.1).

$$\begin{aligned} L_1 &= 154 \cdot 10.23 \text{ MHz} \\ L_2 &= 120 \cdot 10.23 \text{ MHz} \end{aligned} \tag{2.1}$$

The L_1 and L_2 frequencies are both multiples of a 10.23 MHz master clock. The L_1 and L_2 signals are partially offset from their nominal values when transmitted to help correct for relativistic effects. Dual frequency users can compensate for ionospheric effects by exploiting the frequency dependence in the ionospheric delay, [Spilker, Jr., 1996b].

The L_1 signal, broadcast by the ith satellite, is composed of an in-phase and quadrature component. Both components consist of a PRN sequence (also called Gold code), the navigation data message, and their sinusoidal carriers. The in-phase component is modulated by the Coarse/Acquisition (C/A) Gold code $G_i(t)$ and the data message $D_i(t)$ of the satellite. The quadrature component of the L_1 signal is modulated by Precision (P) Gold code $P_i(t)$ and the same data message as the in-phase component. The C/A code, P code, and data message are all unique to each satellite. The power P_c of the in-phase component is 3 dB stronger than the power of the quadraphase component, P_p, [Spilker, Jr., 1996b]. The L_1 signal is given by:

$$S_{L1_i} = \sqrt{2P_c} G_i(t) D_i(t) \cos(\omega_1 t + \phi) + \sqrt{2P_{p,L_1}} P_i(t) D_i(t) \sin(\omega_1 t + \phi) \qquad (2.2)$$

The L_2 signal is modulated by the same data message as the L_1 signal, (2.3).

$$S_{L2_i} = \sqrt{2P_{p,L_2}} P_i(t) D_i(t) \cos(\omega_2 t + \phi) \qquad (2.3)$$

The L_2 signal is modulated by either P or C/A code. Currently, the L_2 channel is modulated with P-code.

The minimum received signal powers are defined in [US Dept. of Defense, 2000]. The minimum power levels for L_1 C/A code, L_1 P code, and L_2 P code are -159.6, -162.6, and -165.2 dBW, respectively [Ward, 1996a]. However, its common to receive GPS signals at power levels several dB in excess of the minimum levels specified. The received GPS signals are well below the thermal noise floor in normal operating conditions. In a typical receiver, the C/A signal at its spectral peak is about 15 dB below the noise power spectral density and the P-code signal is about 28 dB below the thermal noise density [Spilker, Jr., 1996b].

2.4 Gold Code

The PRN sequences, $G_i(t)$ and $P_i(t)$, used by the satellites are called Gold codes. The Gold codes have a very high bandwidth and are used to spread the spectrum of the data message over a much wider bandwidth. In the receiver, the spreading effect of the PRN sequences is removed by using locally generated replicas of the broadcast Gold codes. Each satellite transmits its own unique Gold code and the user receives multiple satellite signals at nearly the same frequency. The user can track the individual codes received because the Gold codes are nearly orthogonal to each other. This means that if the receiver is using one Gold code to track a specific satellite in one channel of the receiver, the signals received from the other satellites are effectively nulled in that channel. This aspect of Gold codes makes the CDMA properties of the GPS system possible, [Misra and Enge, 2001].

Gold codes are formed by modulo-2 addition of two maximal length sequences. Maximal length sequences are formed by shift register generators (SRG) and repeat themselves every $2^N - 1$ bits, where N is the number of stages of the SRG. The C/A Gold codes used by the satellites are all formed by adding the same two maximal length sequences, labeled as $G1$ and $G2$. However, the Gold codes broadcast by each satellite are different. The different Gold codes are formed by shifting the $G1$ and $G2$ sequences in time with respect to each other before adding them. Each Gold code has a specific time shift associated with it. All that is needed to produce a specific satellite's PRN sequence is the delay time defined for that satellite and the $G1$ and G_2 sequences, [Spilker, Jr., 1996b].

The C/A Gold codes are transmitted at a chipping rate of 1.023 Mbps. The individual symbols of the codes are referred to as chips, as opposed to the bits of the navigation message. A code $g(t)$ is originally generated as a binary sequence and is used to modulate the phase of the carrier signal, (2.4).

$$S(t) = \cos(\omega_1 t + \pi g(t)) \tag{2.4}$$

$$S(t) = G(t)\cos(\omega_1 t) \tag{2.5}$$

Since the phase of the signal is being shifted by 180°, the Gold code can be equivalently expressed as (2.5). In Equation (2.5), the original binary sequence $g(t)$ has been mapped to a sequence of ±1's, $G(t)$. The form in (2.5) is the typical way the modulation of the Gold code is represented.

Gold codes have a very narrow time autocorrelation function, as can be seen in figures 2.1 and 2.2. The autocorrelation function is linear for $\tau < 1$ chip of misalignment. Beyond $\tau = 1$ chip, the autocorrelation function is approximately zero. The autocorrelation properties of Gold code are used to track the satellite signal and make pseudorange measurements. The cross-correlation properties of Gold code allow the satellites to all transmit on the same frequency. As was stated before, the different Gold codes are nearly orthogonal to each other, as shown in Figure 2.3. When a receiver collects RF data, multiple satellite signals are recorded in the same frequency band. The receiver "picks out" a specific satellite by multiplying the data by an in-phase replica of the satellite's Gold code. This

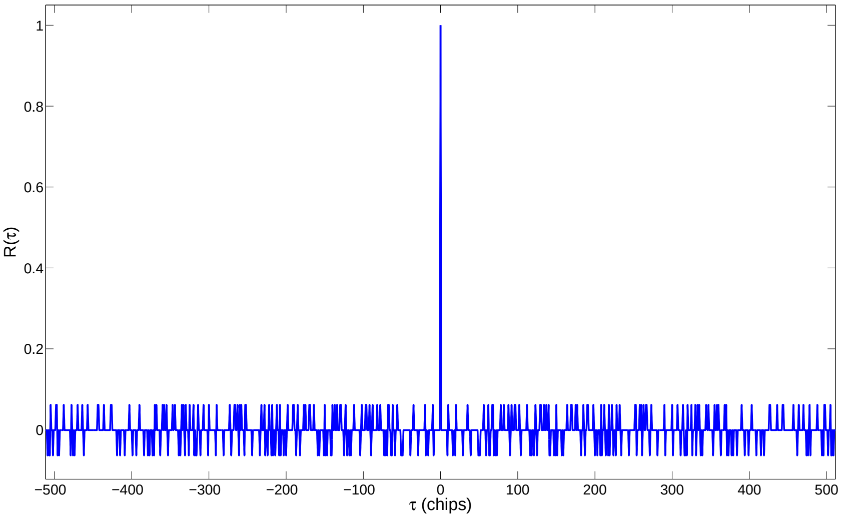

Figure 2.1: Autocorrelation Function of Gold Codes

multiplication effectively cancels out all the other satellite signals while removing the PRN modulation from the signal received from the selected satellite.

2.5 Data Message

The data message transmitted by each satellite contains information about the satellite's orbit, its health, GPS system time, and almanac data for the other satellites in the constellation. The data message contains the necessary information for users to synchronize their clock to system time and determine their position. The rising edge of a data bit is coincident with the rising edge of the first chip of the Gold code sequence. The rising edge of each data bit is also synchronized with the transmitting satellite's clock.

The data message is broadcast at 50 bits per second (bps). The message is arranged into thirty bit words. Ten words constitute a subframe and five subframes make a frame. Figure 2.4 shows how the navigation message is arranged. In each frame, the first three

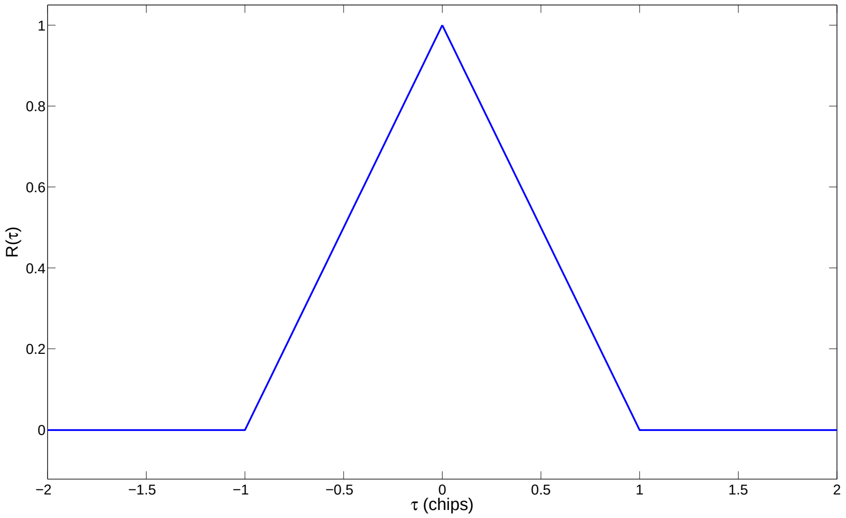

Figure 2.2: Ideal Autocorrelation Function of Gold Codes

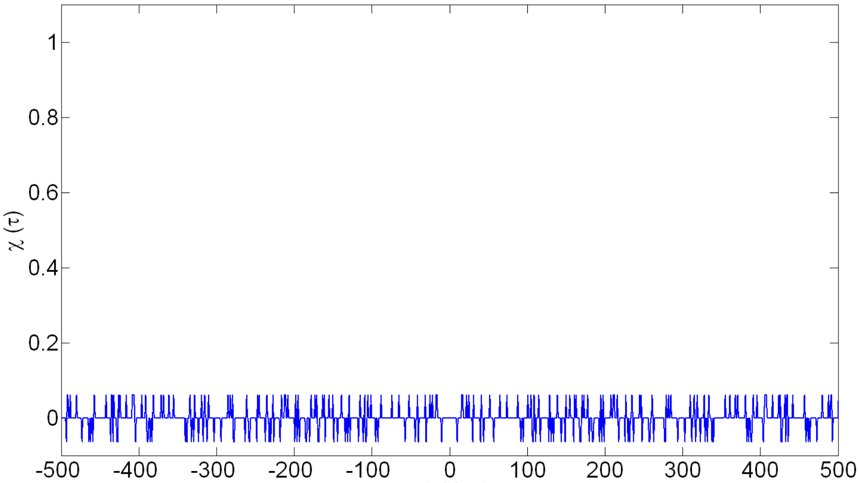

Figure 2.3: Cross-Correlation of Different Gold Codes

subframes contain clock and ephemeris information for the transmitting satellite. Subframes four and five alternately provide support information, such as almanac data and ionospheric correction data, [US Dept. of Defense, 2000]. The satellite ephemeris data is information about the satellite's orbit. Using the ephemeris data, the satellite's position is calculated. The almanac data is approximate ephemeris data for the other satellites in the constellation. It is not as accurate as the ephemeris data in subframes 2 and 3, but is valid for a longer period of time, [Tsui, 2000].

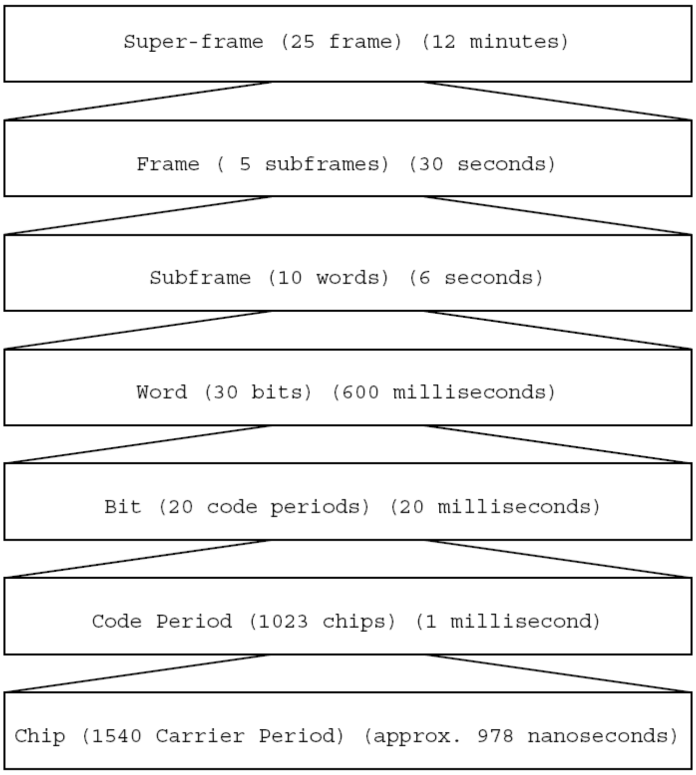
Super-frame (25 frame) (12 minutes)
Frame (5 subframes) (30 seconds)
Subframe (10 words) (6 seconds)
Word (30 bits) (600 milliseconds)
Bit (20 code periods) (20 milliseconds)
Code Period (1023 chips) (1 millisecond)
Chip (1540 Carrier Period) (approx. 978 nanoseconds)

Figure 2.4: Arrangement of the GPS Navigation Message, [Hamm, 2005]

2.6 Conclusion

The basic signals used by the GPS system are explained in this chapter. Currently, the GPS satellites broadcast civilian and military signals at the L_1 center frequency and military only signals at the L_2 frequency. Only the civilian signals at L_1 are considered in the remainder of this dissertation. The following chapters are dedicated to the algorithms that receivers use to process the received satellite signals. The traditional receiver architecture is explained and advanced algorithms that enable greater usability of the signals are analyzed.

CHAPTER 3

POSITION AND VELOCITY DETERMINATION

3.1 Introduction

This chapter discusses how GPS receivers use the received satellite signals to calculate their position and velocity. The system relies on the precise synchronization of the satellite's clocks and accurate determination of the satellite's orbital parameters. The satellites broadcast their data messages synchronously, however they are received at varying times at the receiver. Using the difference in arrival times and broadcast satellite orbital parameters, the receiver calculates its position. The receiver then uses the satellite orbit data and Doppler effect on the carrier signals to determine its velocity. The details of how these calculations are performed are given in this chapter.

3.2 Received Signal Model

The civilian GPS signal received by the GPS user, broadcast by the ith satellite, is modeled as [Sayre, 2003] [Psiaki and Jung, 2002]:

$$R_{L1_i} = \sqrt{2P_{r,i}} G_i(t - \tau_i) D_i(t - \tau_i) \cos((\omega_1 - \omega_{Dopp,i})t + \phi) \tag{3.1}$$

The power of the received signal is denoted by $P_{r,i}$. The frequency of the signal is offset from the broadcast frequency by a Doppler shift $\omega_{Dopp,i}$. A nominal phase ϕ is also added to the carrier. The Gold code and data message received by the user are delayed by the transit time τ_i from the satellite to the user.

The military components of the signal are omitted from (3.1). This is due to the fact that access to the P(Y) code was unavailable while performing this research. Henceforth, the military components are ignored all together.

3.3 Position Determination

GPS receivers determine their position by measuring the time delays τ_i in (3.1) from at least four satellites. To make these measurements, the GPS receiver first approximately synchronizes its internal clock to GPS time. Then, the receiver simply measures the amount of time between when all the satellite's transmit their signals and when the signals from the different satellite's are received. Recall that all the satellite's transmit their signals synchronously. Since the satellites are different distances away in general, the signals will all arrive at the user at different times. As shown in (3.2), the receiver generates *pseudorange* measurements by multiplying the time delays by the speed of light.

$$\begin{aligned} \rho_1 &= c \cdot \tau_1 \\ \vdots &= \vdots \\ \rho_N &= c \cdot \tau_N \end{aligned} \tag{3.2}$$

The pseudorange measurements are related to the position of the receiver by (3.3).

$$\begin{aligned} \rho_1 &= ||s_1 - u|| + ct_u + \eta_1 \\ \vdots &= \vdots \\ \rho_N &= ||s_N - u|| + ct_u + \eta_N \end{aligned} \tag{3.3}$$

$$
\begin{aligned}
||s_j - u|| &= \sqrt{(x_j - x_u)^2 + (y_j - y_u)^2 + (z_j - z_u)^2} \\
\eta &= c(\delta t_{iono} + \delta t_{tropo} + \delta t_{lump}) + e
\end{aligned}
$$

In (3.3), (x_u, y_u, z_u) and (x_j, y_j, z_j) are the coordinates of the receiver and the jth satellite in the Earth-Centered Earth-Fixed (ECEF) coordinate system. Other coordinate systems can be used, but the ECEF system is used throughout this dissertation. The satellite's position is determined from the ephemeris data contained in the data message. The user's clock error is represented by t_u. This is the error in the synchronization of the receiver's clock to GPS time. The terms δt_{iono} and δt_{tropo} represent the signal delay caused by the ionosphere and troposphere, respectively. The term δt_{lump} represents satellite clock errors and other various transmission delays. The effects of thermal noise are represent by the variable e.

Using the pseudorange measurements from at least four satellites, the receiver can solve for its position and clock error. This is typically done using iterative least squares or Kalman filtering [Kaplan et al., 1996]. For both methods, the pseudorange equations are linearized about some initial estimates of the receiver's position and clock bias. As shown in (3.4), the true position and clock bias are equal to the estimates plus corrections.

$$
\begin{aligned}
\rho_i &= f(x_u, y_u, z_u, t_u) \\
x_u &= \hat{x}_u + \delta x_u \\
y_u &= \hat{y}_u + \delta y_u \\
z_u &= \hat{z}_u + \delta z_u \\
t_u &= \hat{t}_u + \delta t_u
\end{aligned}
\tag{3.4}
$$

The pseudorange equations are then approximated by a first order Taylor series. The resulting linear system of equations is shown in (3.5).

$$
\begin{aligned}
\hat{\rho}_i &= \sqrt{(\hat{x}_i - x_u)^2 + (\hat{y}_i - y_u)^2 + (\hat{z}_i - z_u)^2} + c\hat{t}_u \\
\rho_i - \hat{\rho}_i &\approx a_{x,i}\delta x_u + a_{y,i}\delta y_u + a_{z,i}\hat{r}_i\delta z_u - c\delta t_u \\
\begin{bmatrix} \delta\rho_1 \\ \vdots \\ \delta\rho_i \end{bmatrix} &\approx \begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x,i} & a_{y,i} & a_{z,i} & -1 \end{bmatrix} \begin{bmatrix} \delta x_u \\ \delta y_u \\ \delta z_u \\ c\delta t_u \end{bmatrix} \\
\Delta\rho &= H \cdot [u \quad ct_u]'
\end{aligned}
\tag{3.5}
$$

When using iterative least squares, the vector of corrections is calculated and then added to the estimates $(\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{t}_u)$. In the case of four satellites, a matrix inversion can be used to solve for the vector of corrections in (3.5). Alternately, least squares is employed when the equations are overdetermined. The iterative least squares technique converges very rapidly to an estimated position and clock bias.

3.4 Velocity Determination

One method the receiver can use to determine it's velocity is based on measuring the Doppler frequency of the received carrier signals. The measured Doppler frequency of the signal is commonly referred to as the $pseudorange-rate$ measurement. This is because it is a function of the user's velocity relative to the satellites and the clock drift of the receiver's oscillator. The received carrier signal's measured frequency from satellite j given by:

$$f_j = \frac{f_T}{1+\dot{t}_u}\Big[1 - \frac{1}{c}[(v_j - \dot{u}) \cdot a_j]\Big](\mathrm{Hz}) \tag{3.6}$$

In (3.6), f_T is the frequency at which the signal is broadcast. The received signal's frequency is offset from the transmit frequency because of the relative motion of the user and the satellite and the user's clock drift. The terms v_j and a_j are the satellite's velocity vector and the line-of-sight unit vector from the user to the satellite, respectively. The satellite's velocity is calculated from the broadcast ephemeris parameters in the data message. The user's velocity vector is denoted as $\dot{u}$. The user's clock drift is represented as $\dot{t}_u$. The clock drift has units of seconds per second and is the rate at which the user's clock is diverging from GPS system time [Kaplan et al., 1996].

A variable d_j can be defined by expanding (3.6) [Kaplan et al., 1996]:

$$d_j = \frac{c(f_j - f_T)}{f_T} + v_{x,j}a_{x,j} + v_{y,j}a_{y,j} + v_{z,j}a_{z,j} \tag{3.7}$$

In (3.7), v_{xj}, v_{yj}, v_{zj} are the individual components of the satellite's velocity vector. The variables a_{xj}, a_{yj}, a_{zj} are the components of the line-of-sight unit vector from the user to the satellite. In (3.7), the variables are determined from the user's position and the satellite ephemeris data. The variable d_j is related to the user's velocity and clock drift by:

$$d_j = \dot{x_u}a_{x,j} + \dot{y_u}a_{y,j} + \dot{z_u}a_{z,j} - c\dot{t}_u \tag{3.8}$$

In (3.8), x_u, y_u, z_u are the user's velocity vector components. The equations for d_j for the different satellites can be arranged into a set of linear equations:

$$\begin{aligned}\begin{bmatrix} d_1 \\ \vdots \\ d_j \end{bmatrix} &\approx \begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x,j} & a_{y,j} & a_{z,j} & -1 \end{bmatrix} \begin{bmatrix} \dot{x}_u \\ \dot{y}_u \\ \dot{z}_u \\ c\dot{t}_u \end{bmatrix} \\ D &= H \cdot [\dot{u} \;\; c\dot{t}_u]' \end{aligned} \tag{3.9}$$

The receiver calculates the user's velocity and clock drift by first forming the vector D and matrix H. Least squares ($j \geq 5$) or a matrix inversion ($j = 4$) is then used to solve for the user's velocity and clock drift.

As shown in (3.5) and (3.9), the line-of-sight unit vectors are intimately involved in estimating the receiver's position, velocity, and clock states. The linearization involved (3.5) is a very good approximation of the nonlinear pseudorange equation. This is due in great part to the large distances that typically exist between the receiver and the satellites [Wendel et al., 2006].

3.5 Conclusion

GPS receivers use pseudorange measurements from at least four satellites to calculate their position. The pseudorange measurements are basically made by measuring the difference in the arrival times of the signals from different satellites. The different Doppler frequencies on the received carrier signals from four or more satellites are then used by the receiver to determine its velocity. When more than four satellite signals are available, least squares is used to solve the overdetermined system of equations. In general, the position and velocity estimates of the receiver become better as more satellite signals become available (provided that the measurements are properly weighted).

The receiver's ability to determine the time of arrival of the signals and the Doppler shift associated with them is made possible by tracking loops. The tracking loops use the properties of the GPS signals to determine their time of arrival and frequency. Tracking loops serve as an intermediate stage between signal reception and position/velocity estimation. When they cease to function, the receiver can no longer navigate. Because of this, they are a critical component of the receiver. Their ability to reliably function is a major limiting factor for GPS receivers. The next chapter discusses the various tracking loops found inside the receiver and their operation. Subsequent chapters study advanced receiver architectures that combine the two tasks of signal tracking and position/velocity estimation.

Chapter 4

Signal Tracking

4.1 Introduction

The previous chapter explained how a receiver uses the received satellite signals to determine its position and velocity. It was assumed that PRN code phase measurements and Doppler frequency measurements were available. It also assumed that the data messages from the satellite's had been decoded. This chapter explains how the measurements of the signals are generated and how the data message is recovered from the received signals. Tracking loops are the means by which the satellite signals are processed. This chapter describes how tracking loops operate and common metrics used to asses their performance.

4.2 Conventional Tracking Loops

A GPS receiver maintains replicas of the carrier signal and PRN code for each satellite being tracked. Two separate versions of the carrier are generated by the receiver's carrier Numerically Controlled Oscillator (NCO). They are referred to as the in-phase and quadrature signals. The in-phase signal is created using the receiver's best estimate of the phase and frequency of the received carrier. The quadrature signal is identical to the in-phase signal, but with an additional phase angle of 90^o. Three different replicas of the received PRN code are generally generated by the receiver's code NCO. The three versions are called the Early, Prompt, and Late replicas. The Prompt replica is created with the best estimate of the received PRN signal's phase. The Early replica is the same as the Prompt, but it has been advanced by a fraction of the chip period. The Late replica is also the same as the prompt, but it has been delayed by a fraction of the chip period. The fraction of a chip period that the Early and Late replicas are offset is called the correlator spacing.

The locally generated carrier and PRN signals are correlated with the received signal. The correlation process yields six correlator outputs. Fig. 4.1 shows the correlation process.

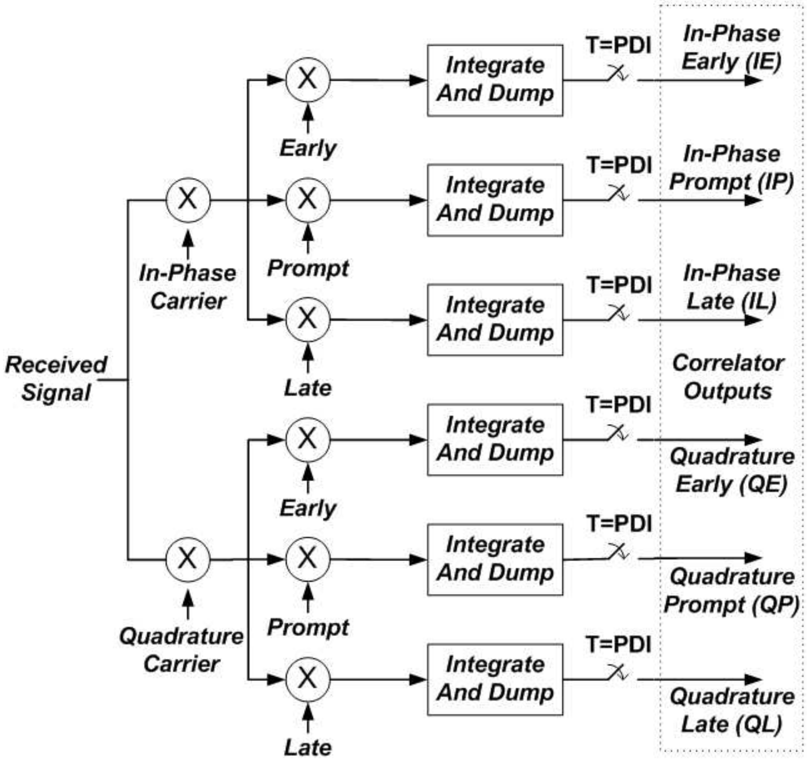

Figure 4.1: Receiver Correlator Structure

As can be seen in Fig. 4.1, the received signal is multiplied by the in-phase and quadrature carrier replicas in separate channels. The resulting signals in each channel are then multiplied by each replica of the PRN code. The resulting signals are summed over an interval of time, referred to as the Predetection Integration (PDI) time. At the end of each

PDI, a single value is produced. The process of integrating the signals and producing a single value is referred to as an integrate and dump operation. The value produced by each integrate and dump operation is referred to as a correlator output. The correlator outputs are labeled according to the signals used to produce them.

The correlator outputs in Fig. 4.1 are functions of several different variables [Sayre, 2003] [Dierendonck, 1996]:

$$\begin{aligned}
IE(k) &= AR(\epsilon+\tau)D(k)\cos(\pi f_{err}T+\theta_{err})+\eta_{IE}(k) \\
IP(k) &= AR(\epsilon)D(k)\cos(\pi f_{err}T+\theta_{err})+\eta_{IP}(k) \\
IL(k) &= AR(\epsilon-\tau)D(k)\cos(\pi f_{err}T+\theta_{err})+\eta_{IL}(k) \\
QE(k) &= AR(\epsilon+\tau)D(k)\sin(\pi f_{err}T+\theta_{err})+\eta_{QE}(k) \\
QP(k) &= AR(\epsilon)D(k)\sin(\pi f_{err}T+\theta_{err})+\eta_{QP}(k) \\
QL(k) &= AR(\epsilon-\tau)D(k)\sin(\pi f_{err}T+\theta_{err})+\eta_{QL}(k) \\
A &= \sqrt{\frac{2CT_{PDI}}{N_o}}\cdot\frac{\sin(\pi f_{err}T)}{(\pi f_{err}T)} \\
\eta &= N(0,1)
\end{aligned} \tag{4.1}$$

In (4.1), the correlator outputs have been normalized so that the noise is unit variance. The term C/N_o is the carrier-power to noise-power spectral density ratio. The term T_{PDI} is the predetection integration interval in seconds. The data bit during the integration period is assumed to be constant and is represented by the symbol $D(k)$. The variables f_{err} and θ_{err} are the frequency error (Hertz) and phase error (radians) between the received and locally generated carrier signals, respectively. The term ϵ is the phase error between the received and locally generated PRN codes, measured in fractions of a chip. The function $R(\epsilon)$ is the auto-correlation function of the PRN sequences. The symbol τ is the correlator spacing of the Early and Late PRN replicas, expressed in fractions of a chip. The autocorrelation as a function of the code phase error is:

$$R(\epsilon) = \begin{cases} 1 - |\epsilon| & \text{when } \epsilon < 1 \text{ chip} \\ 0 & \text{when } \epsilon > 1 \text{ chip.} \end{cases}$$

The receiver uses the correlator outputs to maintain its estimates of the received signal's PRN code phase, carrier frequency, and carrier phase. Discriminator functions use the various correlator outputs to generate measurements of the code phase error, carrier frequency error, and carrier phase error. A traditional receiver uses tracking loops to process the different error signals and maintain signal synchronization. A Delay Lock Loop (DLL) is used to maintain phase-lock between the locally generated and received PRN sequence. A separate DLL is used for each satellite being tracked. Figure 4.2 shows a generic block diagram of a delay lock loop.

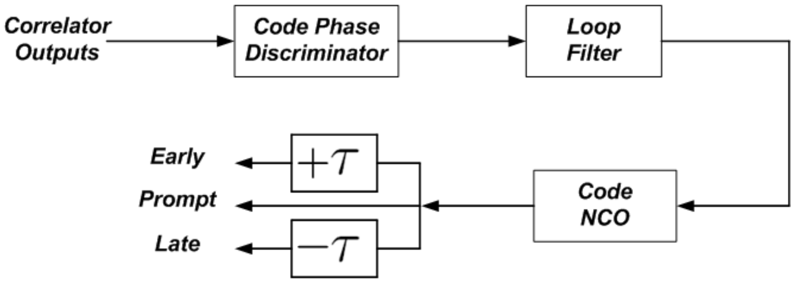

Figure 4.2: Delay Lock Loop

In Figure 4.2, a measurement of the code phase error is produced by the code phase discriminator. A discussion of different code phase discriminators is found in [Ward, 1996b].

The output of the code phase discriminator is filtered by the DLL's loop filter. The output of the loop filter is used to adjust the frequency of the receiver's code NCO. The Early, Prompt, and Late code replicas are generated based on the frequency of the code NCO. The predetection integration time and loop filter design determine the majority of the DLL's performance characteristics.

Most traditional receivers track both the phase and frequency of the received carrier signals. However, the receiver can track only the frequency of the received carrier. Figure 4.3 shows a block diagram of a generic carrier tracking loop for the GPS carrier signal.

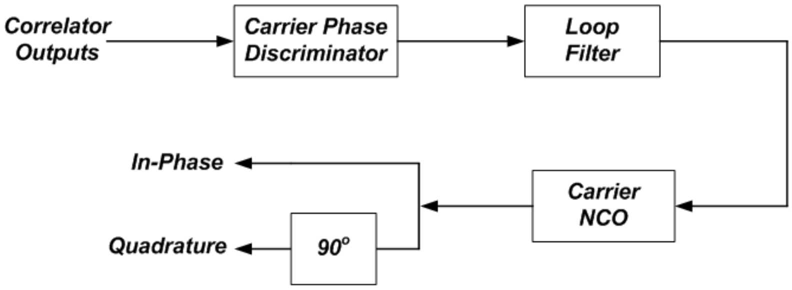

Figure 4.3: Costas Loop

This type of loop is referred to as a Costas loop, [Tsui, 2000]. In Figure 4.3, the carrier phase discriminator produces estimates of the phase error between the replica and received carrier signals. As with the delay lock loop, the frequencies of the replica carrier signals are controlled by the filtered phase error measurements. A discussion of carrier phase discriminators is also found in [Ward, 1996b].

If the receiver only tracks the carrier frequency, a Frequency Lock Loop (FLL) is employed. Figure 4.4 shows a block diagram of a frequency lock loop.

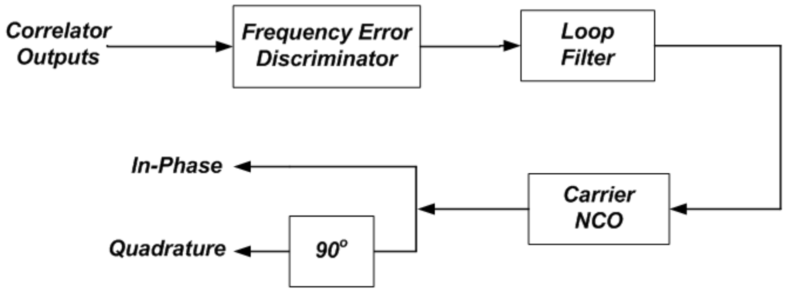

Figure 4.4: Frequency Lock Loop

A frequency lock loop provides a more coarse estimate of the carrier frequency than the Costas loop does. However, the FLL can operate in environments with lower C/N_o levels and higher dynamics than a Costas loop. Many receivers use a hybrid of Costas loops and FLLs to track the carrier signals. As with the DLL, a separate Costas loop or FLL is used to track each individual satellite carrier. The majority of the Costas loop's and FLL's performance characteristics are determined by the predetection integration time and loop filter design.

4.3 Receiver Noise Performance Metrics

The performance of the DLL, Costas loop, and FLL are dependent on the C/N_o ratio of the received signals, dynamic stress, and oscillator phase noise [Ward, 1996b]. The DLL is the most robust in terms of noise performance, followed by the FLL, then the Costas loop. For an unaided GPS receiver not using Data Wipe-Off (DWO), carrier tracking loops cannot reliability operate below about 28 dB-Hz [Soloviev et al., 2008], [Gebre-Egziabher et al., 2005]. Data wipe-off is a technique which effectively eliminates the navigation message modulation and thus allows for integrate and dump times beyond 20 milliseconds. The tracking threshold for an FLL is about 21 dB-Hz and a DLL cannot function below about 16 dB-Hz [Ward, 1996b].

RF interference (i.e. jamming) effects the tracking loops by reducing the effective C/N_o ratio of the received signals. The amount of jamming a receiver can tolerate is calculated by first determining the unjammed C/N_o ratio. The lowest effective C/N_o ratio that can be tolerated is then determined from the tracking threshold for the specific tracking loop being analyzed. The difference between the unjammed C/N_o ratio and the effective C/N_o ratio at which the receiver fails is the Jammer-to-Signal (J/S) ratio [Ward, 1996a]. The equivalent C/N_o ratio is related to unjammed C/N_o ratio and the J/S by [Ward, 1996a]:

$$\left[c/n_o\right]_{eq} = \frac{1}{\frac{1}{c/n_o} + \frac{j/s}{QR_c}} \tag{4.2}$$

where

c/n_o = unjammed carrier-to-noise power ratio

j/s = jammer-to-signal power ratio

R_c = PRN chipping rate (chips/s)

= 1.023 M chips/s for C/A code

= 10.23 M chips/s for P(Y) code

Q = spread spectrum processing gain adjustment factor

= 1 for narrowband jammer

= 1.5 for wideband spread spectrum jammer

= 2 for wideband Gaussian noise jammer

The above is expressed in dB-Hz as [Ward, 1996a]:

$$\left[C/N_o\right]_{eq} = -10\log\left[10^{-(C/N_o)/10} + \frac{10^{(J/S)/10}}{QR_c}\right] \tag{4.3}$$

where

$$C/N_o = \text{ unjammed carrier-to-noise power ratio (dB-Hz)}$$

$$= 10\log\left(c/n_o\right)$$

$$J/S = \text{ jammer-to-signal power ratio (dB)}$$

$$= 10\log\left(j/s\right)$$

The typical C/N_o ratio of the received GPS C/A signal is about 43 dB-Hz. Typical C/N_o ratio's of the L_1 P(Y) and L_2 P(Y) are 40 dB-Hz and 37 dB-Hz, respectively. Using this nominal C/N_o ratio and the tracking threshold's given before, the J/S performance of a stand alone receiver is calculated. Equation (4.3) is first arranged to solve for J/S [Ward, 1996a]:

$$J/S = 10\log\left[QR_c\left(\frac{1}{10^{[C/N_o]_{eq}/10}} - \frac{1}{10^{(C/N_o)/10}}\right)\right] \tag{4.4}$$

Using the nominal received C/N_o ratio's, the J/S ratio versus effective C/N_o ratio is shown in Fig. 4.5 for an unaided GPS receiver, assuming a wideband Gaussian noise jammer.

Using the 28 dB-Hz threshold for carrier phase tracking, the maximum sustainable J/S for an unaided C/A code receiver is about 35 dB. Similarly, the maximum J/S for a P(Y) code receiver is about 45 dB. The additional J/S for a military receiver stems from the higher chipping rate of the P(Y) code. If the 21 dB-Hz threshold for frequency tracking is

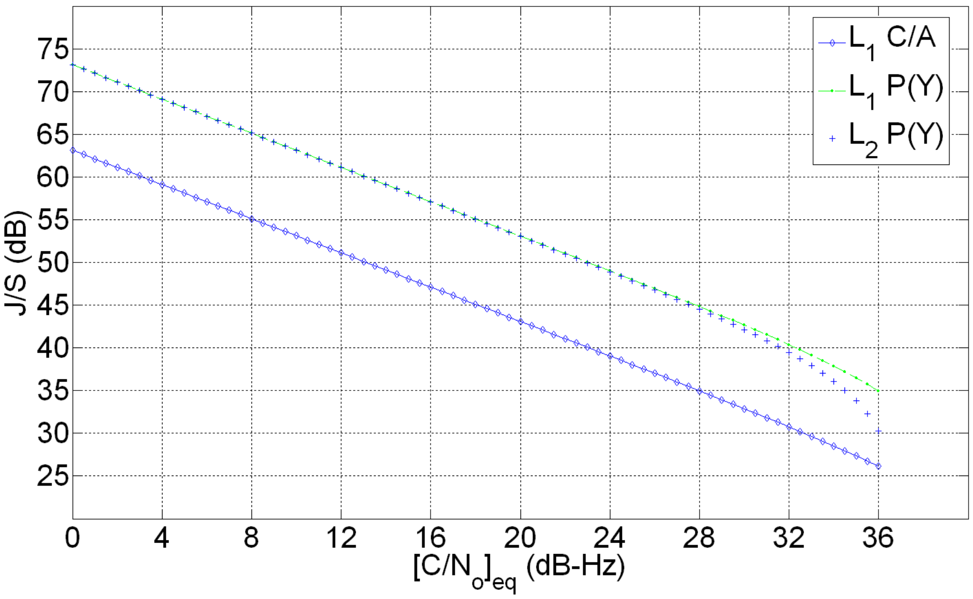

Figure 4.5: J/S Performance of an Unaided GPS Receiver for Wideband Jammer

used, the maximum J/S for a civilian receiver is about 42 dB and is around 52 dB for a military receiver.

The above J/S analysis assumed a carrier phase tracking threshold of 28 dB-Hz and a carrier frequency tracking threshold of 21 dB-Hz. Several researchers have presented advanced algorithms for tracking extremely weak GPS signals. In [Psiaki and Jung, 2002], the authors report carrier phase and code tracking down to 22 dB-Hz, and carrier frequency tracking down to 15 dB-Hz with a TCXO clock. The author of [Ziedan, 2005b], [Ziedan, 2005a], [Ziedan, 2006b], demonstrates the acquisition and tracking of signals in the 10-20 dB-Hz range with a TCXO and OXO clock. In [Lin and Tsui, 2002], the authors report C/A code tracking at 18 dB-Hz. These advanced algorithms replace the traditional tracking loops with extended Kalman filters and employ other optimal estimation techniques. They are significantly more computationally intense than tracking loops and their ability to reliably operate at such low C/N_o ratios with significant user dynamics has yet to be determined.

4.4 Conclusion

The tracking loops described in this chapter form the backbone of a traditional GPS receiver. The measurements used by the receiver to estimate its position and velocity are generated by the tracking loops. When the tracking loops fail, the receiver can no longer navigate. Therefore, maintaining tracking loop operation is critical. The presence of RF interference or weak signal strength compromise the ability of the loops to maintain lock. The next chapter describes an alternate receiver architecture, called vector tracking, that combines the tracking loops and the position/velocity estimation algorithm of the receiver. The combining of the two tasks results in a GPS receiver that can operate in more adverse environments than the traditional, tracking loop based architecture.

Chapter 5

Vector Tracking Receiver

5.1 Introduction

In this chapter, the differences between the traditional receiver architecture and vector based architecture are explained. The fundamental difference is the manner in which the received signals are tracked. The traditional architecture presented in the previous chapter uses a decentralized approach to signal tracking and position-velocity determination. In contrast, the tasks of signal tracking and position-velocity determination are handled by a single central Kalman filter in vector tracking algorithms. Two different formulations of vector tracking are presented in this chapter. The two manifestations of vector tracking are the position-state and pseudorange-state formulations. The formulations differ in their states and the manner in which the channels of the receiver are coupled. In the position-state formulation, the states of the algorithm are the receiver's position, position derivatives, and clock states. In the pseudorange-state formulation, the states of the algorithm are the pseudoranges and pseudorange derivatives of the various satellites being tracked.

5.2 Traditional Receiver Architecture

Figure 5.1 shows a block diagram of the traditional receiver architecture. Tracking loops in each channel are used to estimate the pseudorange and pseudorange-rate for each satellite in view of the receiver. The estimated pseudoranges and pseudorange-rates are fed forward to the navigation processor. The navigation processor uses the estimated pseudoranges and pseudorange-rates to estimate the receiver's position, velocity, and clock states.

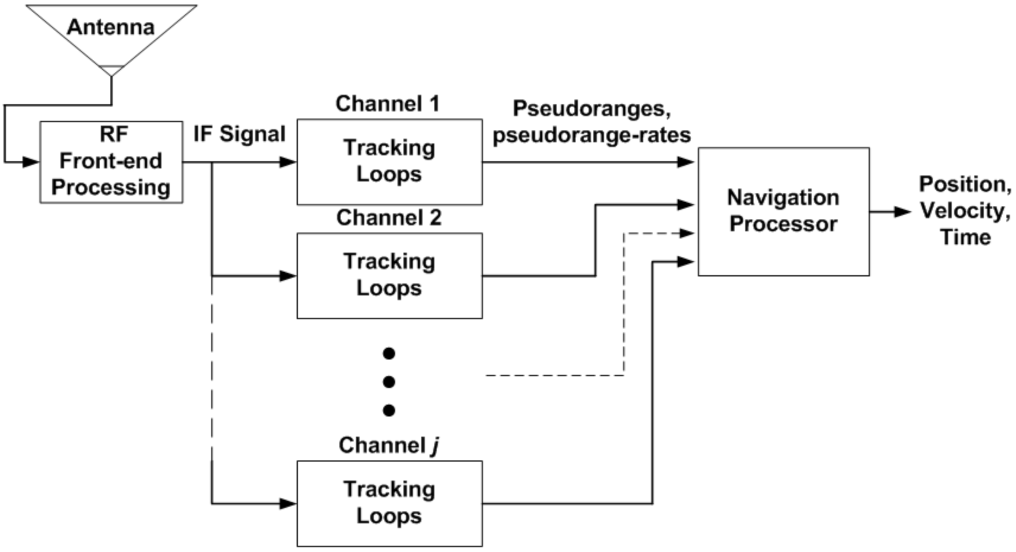

Figure 5.1: Traditional Receiver Architecture

In Figure 5.1, the flow of information is strictly left to right. No information from the navigation processor is fed back to the tracking loops. Additionally, the tracking loops in each channel operate independently of each other, therefore no information is exchanged between them. Each channel of the receiver tracks its respective signal independent of the other channels. Therefore, the traditional receiver architecture does not exploit the inherent coupling between the receiver's dynamics and the dynamics seen by the tracking loops. Equations (3.5) and (3.9) show that the dynamics in each channel are related to the dynamics of the receiver through the line-of-sight vectors. A change in the receiver's velocity will manifest itself in each of the received signals according to the line-of-sight vectors. This intimate relationship between signal tracking and position-velocity estimation are exploited by the vector tracking approach.

5.3 Vector Tracking Architecture

Figure 5.2 shows a block diagram of a vector tracking architecture. For the receiver shown in Figure 5.2, the tasks of signal tracking and navigation state estimation are no longer separate processes. The single Extended Kalman Filter (EKF) simultaneously tracks the received signals and estimates the receiver's position, velocity, etc.

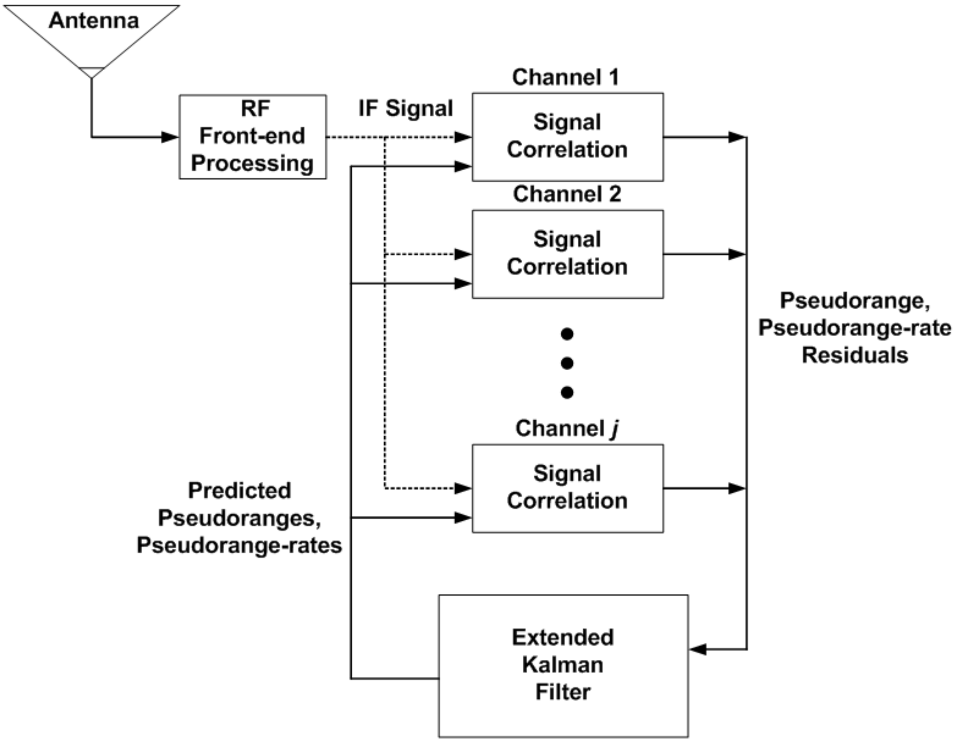

Figure 5.2: Vector Tracking Receiver Architecture

A Vector Delay Lock Loop (VDLL) only tracks the PRN code phases through the central filter. The task of carrier tracking is still handled by scalar tracking loops in each channel in the VDLL. A Vector Delay/Frequency Lock Loop (VDFLL) uses the central filter to track the PRN code phases and the carrier frequencies. The focus of the research in this dissertation is on the VDFLL.

The reason vector tracking is possible relies on the basic nature of the GPS network. The principle behind GPS is that the receiver's position and velocity can be determined based on the phase and frequency of the received signals. Vector tracking algorithms take this basic idea and reverse it. The phase and frequency of the received signals are predicted from the receiver's estimated position and velocity. Residuals are formed in each channel by taking the difference between the predicted and received signals. The residuals are then used to update the estimates of the receiver's position and velocity.

The vector tracking approach exploits the coupling between the receiver's dynamics and the dynamics seen by the tracking loops. Instead of every channel of the receiver tracking the dynamics of the individual signals, the user dynamics that are causing the change in the signals are tracked.

There are two different ways to design the central Extended Kalman filter in Figure 5.2. In the most common method, the states of the filter are the receiver's position, velocity, and clock states. Higher order position derivative states like acceleration and jerk can also be used. This configuration is referred to here as the position-state formulation. The position-state formulation of vector tracking is similar to the design presented in [Spilker, Jr., 1996a]. In the second method, the states of the central filter are the pseudoranges and pseudorange-rates of the available satellites. This type of filter design is referred to here as the pseudorange-state formulation and is similar to the filter used in [Sennott and Senffner, 1992b]. The details of the filters used in each method are described in the next two sections.

5.3.1 Position-State Formulation

The states of the central Kalman filter in the position-state formulation are the receiver's position and clock states. For the VDLL, only the position and clock bias states are required. The VDFLL requires additional velocity and clock drift states. For the algorithm's used in this dissertation, the extended Kalman filter used in the position state VDFLL algorithm tracks the errors in the filter's states, rather than the states themselves [Bullock et al., 2006]. This is referred to as an error-state filter. The error states of the EKF are:

$$\begin{bmatrix} \delta x \\ \delta \dot{x} \\ \delta y \\ \delta \dot{y} \\ \delta z \\ \delta \dot{z} \\ \delta ct \\ \delta c\dot{t} \end{bmatrix} = \begin{bmatrix} \text{Position Error, X-axis} \\ \text{Velocity Error, X-axis} \\ \vdots \\ \text{Clock Bias Error} \\ \text{Clock Drift Error} \end{bmatrix} \tag{5.1}$$

In (5.1), the position and velocity error states are all in the ECEF frame. The clock bias and drift error states are scaled by the speed of light c, giving them units of meters and meters/second, respectively. The actual estimates of the user's states are maintained outside of the EKF algorithm. At each measurement update, the errors in the states are estimated using the pseudorange and pseudorange-rate residuals supplied by each channel. The estimated state errors are then added to the state estimates outside the filter. The state error vector (5.1) is then reset to zeros [Bullock et al., 2006]. Additional states can be appended to the state vector to model the ionospheric and tropospheric errors present in the pseudorange equations. Both errors are typically modeled as first-order Gauss-Markov processes [Brown and Hwang, 1996]. The state dynamics of the filter are shown in (5.2).

$$\dot{x} = Ax + B^p_{dyn} w_{dyn} + B^p_{clk} w_{clk} \tag{5.2}$$

where

$$A = \begin{bmatrix} \kappa & 0_{2\times 2} & 0_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & \kappa & 0_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} & \kappa & 0_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} & 0_{2\times 2} & \kappa \end{bmatrix}$$

$$\kappa = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$B^p_{dyn} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$
\begin{aligned}
w_{dyn} &= \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \\
B^p_{clk} &= \begin{bmatrix} 0_{2\times 2} \\ 0_{2\times 2} \\ 0_{2\times 2} \\ I_{2\times 2} \end{bmatrix} \\
w_{clk} &= \begin{bmatrix} w_b \\ w_d \end{bmatrix}
\end{aligned}
$$

As shown in (5.2), the state dynamics of the EKF are kinematic. The process noise in the system comes from two sources, receiver dynamics and clock noise. The changing dynamics of the receiver are modeled by the noise vector w_{dyn}. The noise sources (w_x, w_y, w_z) drive their respective velocity states. This corresponds to a continuous Wiener process velocity model [Bar-Shalom et al., 2001]. The terms (w_b, w_d) in w_{clk} represent the clock phase and frequency error of the user's local oscillator, expressed in units of meters and meters per second [Brown and Hwang, 1996], respectively. The statistics for the noise vector w_{dyn} are:

$$
E\{w_{dyn}\} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.3}
$$

$$E\{w_{dyn}w_{dyn}^T\} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}$$

The values for σ_x^2, σ_y^2, and σ_z^2 are chosen based on the expected level of receiver dynamics. The statistics for the noise vector w_{dyn} are:

$$\begin{aligned} E\{w_{clk}\} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ E\{w_{clk}w_{clk}^T\} &= \begin{bmatrix} \sigma_b^2 & 0 \\ 0 & \sigma_d^2 \end{bmatrix} \end{aligned} \tag{5.4}$$

The values for σ_b^2 and σ_d^2 are usually based on rule of thumb numbers for the type of oscillator (compensated crystal, rubidium, etc.) the receiver uses [Brown and Hwang, 1996].

The states of the filter are updated with the residuals from each channel. The relationship between the states of the filter and the pseudorange and pseudorange-rate residuals is shown in (5.5).

$$y = Cx + v \tag{5.5}$$

$$C = \begin{bmatrix} a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 & 0 \\ 0 & a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 & 0 \\ 0 & a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 \end{bmatrix}$$

In (5.5), the variables $a_{x,N}, a_{y,N}, a_{z,N}$ are the components of the line-of-sight unit vector from the user to the N-th satellite.

The statistics for the measurement noise vector v are shown in (5.6).

$$\begin{aligned} E\{v\} &= 0_{N\times 1} \\ E\{vv^T\} &= R_v \end{aligned} \tag{5.6}$$

As shown in (5.5), the pseudoranges and pseudorange-rates from the different channels are related to the states of the filter through the line-of-sight vectors $(a_{x,N}, a_{y,N}, a_{z,N})$. After processing the residuals, the pseudoranges and pseudorange-rates are predicted for the next integration interval using the updated state estimates. Equation (5.5) shows that in the position-state VDFLL formulation the satellite signals are coupled together through the states of the filter. Errors in the predicted pseudoranges and pseudorange-rates are corrected by applying corrections to the states of the filter.

5.3.2 Pseudorange-State Formulation

The states of the extended Kalman filter in the pseudorange state formulation are the pseudoranges and pseudorange-rates for all the available satellites. The error states of the

EKF are shown below in (5.7).

$$\begin{bmatrix} \delta\rho_1 \\ \delta\dot{\rho}_1 \\ \vdots \\ \delta\rho_N \\ \delta\dot{\rho}_N \end{bmatrix} = \begin{bmatrix} \text{SV 1 Pseudorange Error} \\ \text{SV 1 Pseudorange-rate Error} \\ \vdots \\ \text{SV N Pseudorange Error} \\ \text{SV N Pseudorange-rate Error} \end{bmatrix} \tag{5.7}$$

The state dynamics of the filter are shown in (5.8).

$$\dot{x} = Ax + B^{\rho}_{dyn} w_{dyn} + B^{\rho}_{clk} w_{clk} \tag{5.8}$$

$$\begin{aligned} &where \\ A &= \begin{bmatrix} \kappa & 0_{2\times2} & \dots & 0_{2\times2} \\ 0_{2\times2} & \kappa & \dots & \vdots \\ \vdots & \vdots & \ddots & 0_{2\times2} \\ 0_{2\times2} & \dots & 0_{2\times2} & \kappa \end{bmatrix} \\ \kappa &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\ B^{\rho}_{dyn} &= \begin{bmatrix} 0 & 0 & 0 \\ a_{x,1} & a_{y,1} & a_{z,1} \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ a_{x,N} & a_{y,N} & a_{z,N} \end{bmatrix} \end{aligned}$$

$$
\begin{aligned}
w_{dyn} &= \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \\
B^{\rho}_{clk} &= \begin{bmatrix} I_{2\times 2} \\ \vdots \\ I_{2\times 2} \end{bmatrix} \\
w_{clk} &= \begin{bmatrix} w_b \\ w_d \end{bmatrix}
\end{aligned}
$$

The noise terms in (5.8) are the same as those shown in (5.2). The states of the filter are updated with the residuals from each channel. The relationship between the states of the filter and the pseudorange and pseudorange-rate residuals is shown in (5.9).

$$
y = I_{2N\times 2N}x + v \tag{5.9}
$$

As shown in (5.8) and (5.9), the signals from different satellites are not coupled through the state transition matrix or the measurement update equations. This would tend to imply that the Kalman gain computed by the filter is a diagonal matrix. If the Kalman gain were a diagonal matrix, then a measured pseudorange error for one satellite would not effect the pseudorange states for the other satellites. This would result in the states of the filter being completely unconnected. However, the Kalman gain is not a diagonal matrix in the pseudorange state formulation. This is because the noise processes driving all the states of the filter are shared. The common noise sources in (5.8) results in the Kalman gain computed by the filter having non-zero off-diagonal elements. Therefore, the signals are coupled together through the process noise in the pseudorange state formulation.

5.4 Carrier Phase Vector Tracking

The algorithms in the preceding sections are non-coherent, meaning that the carrier phase of the signals is not tracked. The phase of the PRN codes and the frequency of the carrier signals can be directly predicted from the receiver's estimated position, velocity, clock bias, and clock drift. The receiver's estimated position using stand alone GPS is not accurate enough to unambiguously predict the phase of the carrier signals. However, this does not preclude the possibility of using a Vector Phase Lock Loop (VPLL). A simple way of incorporating the carrier phase measurements is shown in Figure 5.3.

In Figure 5.3, the tasks of tracking the carrier signals and the PRN codes are handled separately. One filter tracks the carrier phases and frequencies, while the other only tracks the PRN code phases. This architecture is similar to that used in [Sennott and Senffner, 1992a]. The carrier tracking filter is analogous to the pseudorange-state filter formulation in that its states are the phases and frequencies of the various carrier signals. Phase and frequency error discriminators are used to transform the correlator outputs into usable residuals for the filter. The code phase filter only uses pseudorange error discriminators. The states of the carrier phase filter are used to remove almost all the dynamics in the PRN code phases (similar to Doppler aiding of a DLL).

It is possible to use a single filter to track the pseudoranges, pseudorange-rates, and the carrier phases. However, it is more complicated. If the pseudorange-state filter formulation is used, there would be at least two states per satellite, the two states being the phase and

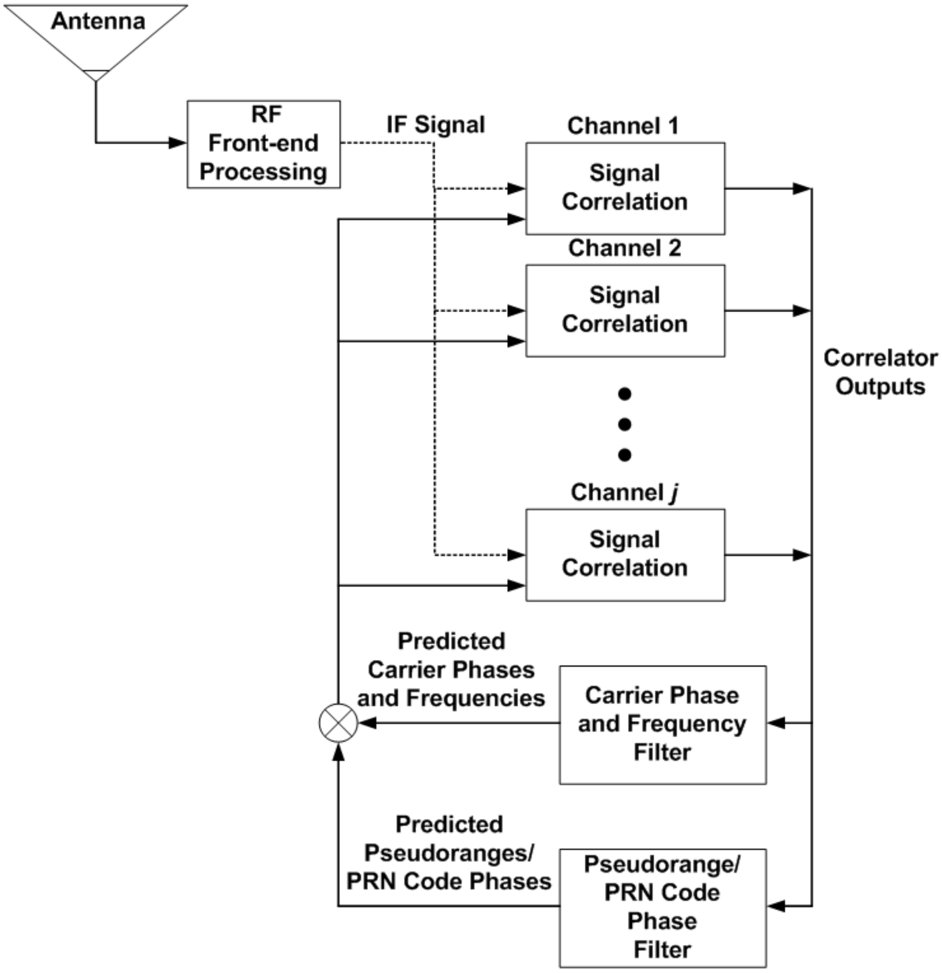
Antenna
RF Front-end Processing
IF Signal
Channel 1
Signal Correlation
Channel 2
Signal Correlation
Channel j
Signal Correlation
Correlator Outputs
Predicted Carrier Phases and Frequencies
Carrier Phase and Frequency Filter
Predicted Pseudoranges/ PRN Code Phases
Pseudorange/ PRN Code Phase Filter

Figure 5.3: Cascaded Vector Phase Lock Loop Architecture

frequency of its signals. Two measurements of the phase would be available, one a code phase measurement and the other a carrier frequency measurement. Issues such as code-carrier divergence and atmospheric effects would have to be addressed by in the design of the filter.

It is important to note that the carrier phase is more susceptible to disturbances caused by foliage and other obstructions than are the carrier frequencies and code phases. Consequently, care is needed in the design of a VPLL. Since the tracking of all the signals is done in unison, a disturbance in one carrier phase would affect all the other estimated phases and frequencies. Also, if the only process dynamics that are modeled are the receiver's motion and clock error dynamics, a disturbance in the phase of one carrier signal would cause the filter's estimate of its frequency to change. Perturbations in the carrier phases could cause the filter to diverge.

The algorithms investigated in this dissertation are all non-coherent. This is because of all the issues encountered when attempting to track the carrier phases in a vector-based architecture. In general, the frequency of the carrier signal is more robust to disturbances and can also be tracked at lower C/N_0 ratios than the carrier phase [Groves et al., 2007].

5.5 Conclusion

Two alternate formulations of vector tracking are presented in this chapter. The position-state formulation directly tracks the receiver's position, velocity, and clock states. The satellite signals are predicted using the estimated navigation states of the receiver. The pseudorange and pseudorange-rate residuals produced in each channel are then used to correct the filter's estimates. In contrast, the pseudorange-state formulation tracks the individual pseudoranges and pseudorange-rates of the satellites without using the intermediate position and clock states.

The performance of the two vector tracking algorithms is compared to that of scalar tracking loops in the following chapters. The pseudorange-state filter becomes particularly useful in making the comparison. This is because the filter can be operated with or without information sharing between the different channels. When information is not shared between the channels of the receiver, the filter reverts back to simply a collection of scalar tracking loops. The performance of the same filter in scalar and vector tracking modes can then be analyzed. The performance improvement offered by vector tracking can then be determined for a specific scenario.

CHAPTER 6

VECTOR TRACKING LOOP PERFORMANCE

6.1 Introduction

In this chapter, the performance of vector tracking algorithms is compared to scalar tracking loops. The reason for the improved performance of vector tracking over traditional methods is explained by using a simple weighted least squares example. A more detailed comparison of the two approaches is then performed by using a covariance analysis. The pseudorange state formulation is used as a convenient means of making a side by side comparison. It eliminates the need to try and equate the process noise tunings of a Kalman filter to the noise bandwidth used by a tracking loop. Therefore the pseudorange state formulation is used to make a valid comparison between vector and scalar tracking loops.

6.2 Weighted Least Squares Example

In this section, a simple example using instantaneous weighted least squares is used to explain the improved performance of vector tracking algorithms over scalar tracking loops. The variance in the pseudorange and pseudorange-rate estimates produced by the two architectures are used as the performance criteria. The general result is that vector tracking algorithms are able to exploit an overdetermined system of equations. The number of unknowns the vector based algorithm attempts to estimate remains constant as more measurements become available. In general, the estimates made by the vector tracking algorithm become better as the number of measurements increases. In contrast, a scalar tracking loop architecture does not exploit the overdetermined system of equations. For the scalar case, the number of unknowns being estimated is equal to the number of available measurements. No improvement is seen as the number of measurements increases.

The fact that the set of equations a vector based method uses to estimate position and clock bias is overdetermined does not necessarily mean it performs better. The manner in which the measurements are related to the receiver's position and clock bias is essential to determine how the vector based algorithm will perform. The line-of-sight unit vectors from the position of the receiver to the available satellite's position couple the measurements together. The weighted least squares example is used to demonstrate how the vector tracking and scalar tracking methods differ. The weighted least squares example uses a single, instantaneous set of measurements to illustrate the advantage in performance that vector tracking offers.

First, it is assumed that a receiver is tracking N satellites and that it is provided N pseudorange residual measurements, as shown in (6.1).

$$\begin{aligned}
\begin{bmatrix} \delta\tilde{\rho_1} \\ \vdots \\ \delta\tilde{\rho_N} \end{bmatrix} &= \begin{bmatrix} \delta\rho_1 \\ \vdots \\ \delta\rho_N \end{bmatrix} + \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} \\
\Delta\tilde{\rho} &= \Delta\rho + V \qquad (6.1) \\
V &\sim N(0, R_v) \\
R_v &= \sigma_v^2 \cdot I_{N\times N}
\end{aligned}$$

In (6.1), the measurements ($\delta\tilde{\rho}$) are the difference between the receiver's locally generated predictions of the pseudoranges ($\hat{\rho}$) and the actual pseudoranges (ρ) that are received from the satellites. The measured pseudorange residuals are equal to the true residuals plus zero-mean Gaussian noise. The variance of the measurement noise is assumed to be the same for all the satellites.

The first goal of the receiver is to estimate the true pseudorange residuals from the measured pseudorange residuals in (6.1). Then, the receiver corrects its locally generated pseudorange predictions with the estimated residuals. Two different approaches of updating the locally generated pseudorange predictions are considered. The first approach is analogous to using independent, scalar delay lock loops to process the measurements. The fact that the pseudoranges are all coupled through the receiver's position and clock bias is not exploited. The second approach uses the pseudorange residual measurements to estimate errors in the user's estimated position and clock bias. The locally generated pseudorange predictions are then corrected using the updated receiver position and clock bias estimates.

6.2.1 Scalar Tracking Method

The algorithm analogous to scalar tracking loops is considered first. It takes the N pseudorange residual measurements shown in (6.1) and uses them to estimate the N pseudorange errors independently. The measurement equation and weighted least squares solution for this method are shown in (6.2).

$$\begin{aligned}
\Delta\tilde{\rho} &= I_{N\times N}\cdot\Delta\rho + V \\
\Delta\hat{\rho} &= (I_{N\times N}^{T}\cdot R_v^{-1}\cdot I_{N\times N})^{-1}\cdot I_{N\times N}^{T}\cdot R_v^{-1}\cdot\Delta\tilde{\rho} \\
\Delta\hat{\rho} &= \Delta\tilde{\rho} \qquad (6.2)
\end{aligned}$$

As shown in (6.2), the weighted least squares solution reduces down to setting the estimated pseudorange errors equal to the measurements. The mean and covariance of the weighted least squares solution are shown in (6.3).

$$\begin{aligned} E\{\Delta\hat{\rho}\} &= \Delta\rho \\ E\{(\Delta\hat{\rho} - E\{\Delta\hat{\rho}\})(\Delta\hat{\rho} - E\{\Delta\hat{\rho}\})^T\} &= (I_{N\times N}^T \cdot R_v^{-1} \cdot I_{N\times N})^{-1} \\ E\{(\Delta\hat{\rho} - E\{\Delta\hat{\rho}\})(\Delta\hat{\rho} - E\{\Delta\hat{\rho}\})^T\} &= \sigma_v^2 \cdot I_{N\times N} \end{aligned} \tag{6.3}$$

The mean of the error in the estimated pseudorange residuals is zero and the variance of the error in the estimated pseudorange is equal to the variance of the noise. The locally generated pseudorange predictions are then corrected with the estimated residuals.

6.2.2 Vector Tracking Method

The second approach for correcting the estimated pseudoranges is based off the concept of vector tracking. Using the vector based method, the measurements are first related to errors in the receiver's estimated position and clock bias. The receiver's estimated position and clock bias are corrected using the set of N pseudorange residual measurements. The updated estimates are then used to correct the locally generated pseudorange predictions.

The measurements in (6.1) are related to errors in the user's position and clock bias by (6.4).

$$\begin{aligned} \Delta\tilde{\rho} &\approx \begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x,N} & a_{y,N} & a_{z,N} & -1 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ c\delta t_u \end{bmatrix} + V \\ \Delta\tilde{\rho} &\approx H_{N\times 4}\Delta X + V \end{aligned} \tag{6.4}$$

Equation (6.4) is obtained by linearizing the pseudorange equations. In (6.4), errors in the receiver's estimated position and clock bias are denoted by δx, δy, δz, and δt_u, respectively. The terms $a_{x,j}$, $a_{y,j}$, and $a_{z,j}$ are the line-of-sight unit vectors from the receiver's estimated position to the position of the jth satellite. The weighted least squares estimate of the vector ΔX is given in (6.5).

$$\Delta\hat{X} \quad = \quad (H_{N\times4}^T \cdot R_v^{-1} \cdot H_{N\times4})^{-1} \cdot H_{N\times4}^T \cdot R_v^{-1} \cdot \Delta\tilde{\rho} \tag{6.5}$$

The expected value and variance of the WLS estimate in (6.5) is given by (6.6).

$$\begin{aligned} E\{\Delta\hat{X}\} &= \Delta X \qquad (6.6) \\ E\{(\Delta\hat{X} - E\{\Delta\hat{X}\})(\Delta\hat{X} - E\{\Delta\hat{X}\})^T\} &= (H_{N\times4}^T \cdot R_v^{-1} \cdot H_{N\times4})^{-1} \\ E\{(\Delta\hat{X} - E\{\Delta\hat{X}\})(\Delta\hat{X} - E\{\Delta\hat{X}\})^T\} &= \sigma_v^2 \cdot (H_{N\times4}^T \cdot H_{N\times4})^{-1} \end{aligned}$$

The locally generated pseudorange predictions are then updated with the results from (6.6). Errors in the receiver's estimated position and clock bias are related to errors in the predicted pseudoranges by (6.7).

$$\Delta\rho \quad \approx \quad H_{N\times4}\Delta X \tag{6.7}$$

The variance of the corrections to the predicted pseudoranges is given by (6.8).

$$\begin{aligned} E\{(\Delta\hat{\rho} - E\{\Delta\hat{\rho}\})(\Delta\hat{\rho} - E\{\Delta\hat{\rho}\})^T\} &= \sigma_v^2 \cdot W_{N\times N} \qquad (6.8) \\ W_{N\times N} &= H_{N\times4} \cdot (H_{N\times4}^T \cdot H_{N\times4})^{-1} \cdot H_{N\times4}^T \end{aligned}$$

The variance of the corrections is found by multiplying the measurement noise variance (σ_v^2) by the diagonal elements of the matrix W. The scalar approach and vector based approach are compared by examining (6.3) and (6.8). For the scalar algorithm, the variance of the corrections is always equal to the variance of the measurement noise σ_v^2. In contrast, the variance of the corrections for the vector approach depends on the matrix W. The diagonal elements of W must be less than unity in order for the vector based approach to yield smaller variances.

If the number of satellites is equal to four, (6.3) and (6.8) both reduce to (6.9), assuming H has full rank.

$$E\{(\Delta\hat{\rho} - E\{\Delta\hat{\rho}\})(\Delta\hat{\rho} - E\{\Delta\hat{\rho}\})^T\} \quad = \quad \sigma_v^2 \cdot I_{4\times 4} \tag{6.9}$$

The two approaches yield the same variances in the case of four satellites. In general, if the number of satellites N exceeds four and the matrix H has four linearly independent rows, the diagonal elements of W will be less than one. In contrast, the diagonal elements from the identity matrix in (6.3) are always equal to one. The diagonal elements of W being less than unity constitutes a reduction in the variance of the estimated pseudoranges when processed collectively in the vector tracking approach. In general, the diagonal elements of the matrix W will also not be equal. Therefore, the decrease in the variance of the individual pseudoranges will be different. The improvement in tracking brought about by the vector based approach will depend on which specific diagonal element of the matrix W is examined.

The weighted least squares example demonstrates that the vector tracking approach works better than the traditional, scalar tracking method. The improvement offered by vector tracking is contingent on several different variables. Specifically, the number of satellite signals available and the geometry of the visible satellite constellation determines the relative advantage of the vector tracking approach. In the case of four satellites, the two approaches yield the same results. However, it is common for GPS receivers at low latitudes to have eight to ten visible satellites. In these situations, a significant advantage is offered by vector tracking.

6.2.3 Performance Gain from Collective Processing

To demonstrate the advantage offered by vector tracking for a typical GPS receiver, the visible satellite constellation was recorded over an 18 hour period at Auburn University. An 18 hour period provides a sufficiently diverse record of the satellite constellation available to the receiver. The positions of the visible satellites were recorded every minute during the data collection interval. The W and H matrices in (6.8) were then computed for all the various satellite geometries. Recall that the variances of the estimated pseudoranges from the scalar tracking method are equal to the variance of the measurement noise σ_v^2. For the vector tracking method, the variance of the estimated pseudoranges is equal to the variance of the measurement noise, scaled by corresponding diagonal element of the matrix W.

To determine the improvement offered by vector tracking in terms of the C/N_0 ratio, it is first assumed that the variance of the measurement noise is related to C/N_0 by the formula shown in (6.10) [Crane, 2007].

$$\begin{aligned} \sigma_v^2 &= \frac{\beta^2}{2(T \cdot 10^{(C/N_o)/10})^2} + \frac{\beta^2}{4T \cdot 10^{(C/N_o)/10}} \text{ (meters}^2\text{)} \\ \beta &= 293.3 \text{ (meters)} \end{aligned} \tag{6.10}$$

The formula in (6.10) comes from the code phase discriminator used elsewhere in this dissertation. In (6.10), the variable T is the predetection integration time used by the receiver. Using nominal values of 20 ms for the integration time and 45 dB-Hz for the C/N_0 ratio, the variance of the measurement noise is approximately 34 m^2. Therefore, the variance of the estimated pseudoranges from the scalar tracking algorithm is also 34 m^2. For the vector tracking method, it is assumed that all the satellite signals have the same C/N_0 ratio. The variance of the jth pseudorange is then equal to 34 m^2 scaled by the jth diagonal element of the matrix W from (6.8). The pseudorange with the maximum reduction in variance will have the smallest corresponding diagonal element. Similarly, the pseudorange with the minimum reduction in variance will have the largest corresponding diagonal element.

The performance gain from vector tracking is determined using the pseudoranges with the largest and smallest variances. The maximum performance gain from vector tracking is calculated using the pseudorange with the smallest variance. The maximum performance gain is defined as the reduction in C/N_0 ratio needed to make the smallest pseudorange variance equal to 34 m^2. Likewise, the minimum performance gain is defined as the reduction in C/N_0 ratio needed to make the largest pseudorange variance equal to 34 m^2. Using these definitions, the maximum and minimum performance gain were calculated for all the different satellite geometries that occurred over the 18 hour data collection interval. Figure. 6.1 shows the Position Dilution of Precision (PDOP) and number of available satellites throughout the 18 hour period. Figure 6.2 shows the maximum and minimum performance gains versus time for the recorded data.

As shown in Fig. 6.2, the performance gains from vector tracking vary throughout the data collection period. The average maximum performance gain is 5.2 dB and it varies from

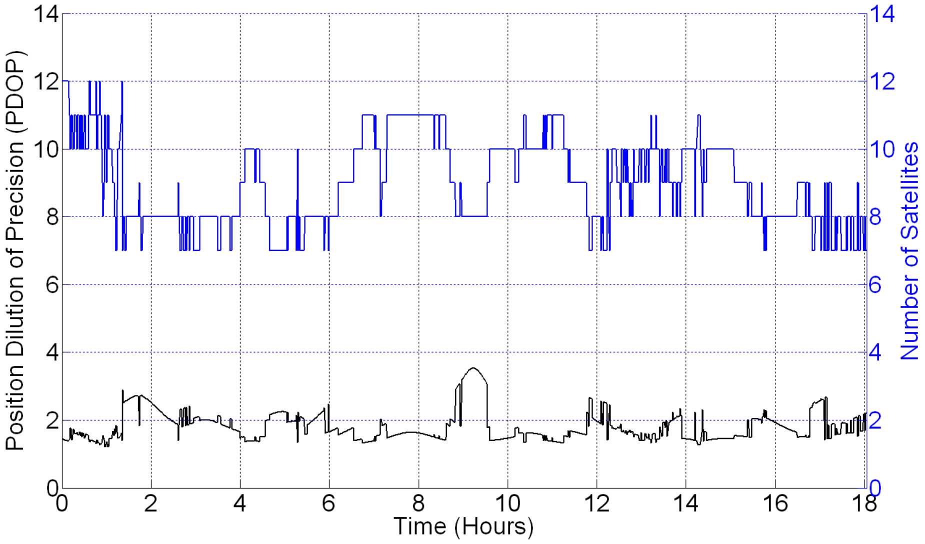

Figure 6.1: Position Dilution of Precision and Number of Satellites Throughout Data Collection Period

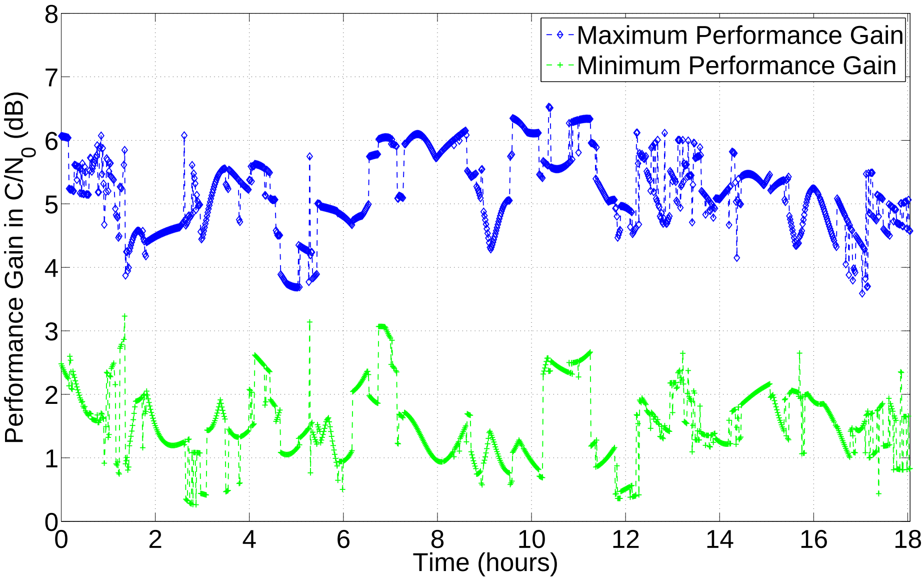

Figure 6.2: Maximum and Minimum Performance Gain in C/N_0 from Vector Tracking for Weighted Least Squares Example

3.56 to 6.53 dB. The average minimum performance gain is 1.54 dB and it varies from .26 to 3.23 dB over the data collection period.

The simple weighted least squares example used in this section explains the performance improvement offered by vector tracking over scalar tracking. The weighted least squares analysis also reveals several characteristics of vector tracking. For instance, it shows that the improvement from vector tracking is a function of the number of available satellites and their geometry. The live-sky GPS data demonstrated the improvement offered by vector tracking for a typical receiver. Over an 18 hour period, the average maximum and minimum performance improvements are 5.2 dB and 1.54 dB, respectively. The weighted least squares example also reveals that the performance improvement is different for the various satellite signals being tracked. In the next section, a more detailed analysis of scalar and vector tracking is performed. A covariance analysis of the Kalman filter used in the pseudorange-state VDFLL formulation is used to make a valid side by side comparison of the two architectures.

6.3 Coupled and Decoupled Pseudorange State VDFLL

A covariance analysis of the pseudorange-state formulation of vector tracking is performed in this section. The Kalman filter in the pseudorange-state formulation of the VDFLL is used to study the fundamental differences between scalar and vector tracking. The architecture of the pseudorange-state filter allows it to function with either the channels of the receiver coupled together in a vector tracking mode, or uncoupled in a scalar tracking mode. Using the same process and measurement noise tunings, the performance of the filter is studied in the coupled and uncoupled modes of operation. This allows the advantage offered solely by the vector tracking architecture to be isolated such that a valid comparison between scalar and vector tracking loops can be performed. Using the pseudorange-state VDFLL in the different modes of operation avoids the ambiguities encountered when attempting to make a valid comparison between tracking loops (which are characterized by noise bandwidths and loop order) and Kalman filters (which are characterized by process and measurement noise covariance matrices).

The performance improvement gained by going from an uncoupled to a coupled pseudorange-state filter is equivalent to the increased performance garnered by going from an uncoupled pseudorange-state filter to a position-state filter. This is because the performance of the position-state and coupled pseudorange-state filters is equivalent. This point is elucidated in Appendix A.

The continuous time dynamics of the the pseudorange-state VDFLL filter are shown in (6.11).

$$\dot{x} = Ax + B^{\rho}_{dyn} w_{dyn} + B^{\rho}_{clk} w_{clk} \tag{6.11}$$

The terms in (6.11) pertaining to the clock process noise, and the clock process noise statistics are given in (6.12).

$$\begin{aligned} B^{\rho}_{clk} &= \begin{bmatrix} I_{2\times 2} \\ \vdots \\ I_{2\times 2} \end{bmatrix} \\ w_{clk} &= \begin{bmatrix} w_b \\ w_d \end{bmatrix} \end{aligned} \tag{6.12}$$

$$E\{w_{clk}w_{clk}^T\} = \begin{bmatrix} \sigma_b^2 & 0 \\ 0 & \sigma_d^2 \end{bmatrix}$$

The receiver dynamics process noise terms and statistics are shown in (6.13).

$$\begin{aligned} B_{dyn}^{\rho} &= \begin{bmatrix} 0 & 0 & 0 \\ a_{x,1} & a_{y,1} & a_{z,1} \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ a_{x,N} & a_{y,N} & a_{z,N} \end{bmatrix} \\ w_{dyn} &= \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \\ E\{w_{dyn}w_{dyn}^T\} &= \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \end{aligned} \tag{6.13}$$

When the dynamics of the system are converted to discrete time, the following discrete-time state transition matrix is calculated.

$$A^{\rho} = \begin{bmatrix} \alpha & 0_{2\times 2} & \dots & 0_{2\times 2} \\ 0_{2\times 2} & \alpha & & \vdots \\ \vdots & & \ddots & \vdots \\ 0_{2\times 2} & \dots & \dots & \alpha \end{bmatrix}_{2N\times 2N} \tag{6.14}$$

$$\alpha = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$$

In (6.14), the time between measurement updates is represented by T. The discrete time process noise covariance matrix for the clock noise is shown in (6.15).

$$\begin{aligned} Q^{\rho}_{clk} &= \begin{bmatrix} Q^c & \dots & Q^c \\ \vdots & \ddots & \vdots \\ Q^c & \dots & Q^c \end{bmatrix} \\ Q^c &= \begin{bmatrix} \sigma_b^2 T + \sigma_d^2 \frac{T^3}{3} & \sigma_d^2 \frac{T^2}{2} \\ \sigma_d^2 \frac{T^2}{2} & \sigma_d^2 T \end{bmatrix} \end{aligned} \tag{6.15}$$

The discrete time covariance matrix for the receiver dynamics process noise is given in (6.16).

$$\begin{aligned} Q^{\rho}_{dyn} &= \begin{bmatrix} Q^d_{1,1} & Q^d_{1,2} & \dots & Q^d_{1,N} \\ Q^d_{2,1} & Q^d_{2,2} & \dots & Q^d_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ Q^d_{N,1} & \dots & \dots & Q^d_{N,N} \end{bmatrix} \\ Q^d_{i,j} &= \begin{bmatrix} \beta_{i,j} \frac{T^3}{3} & \beta_{i,j} \frac{T^2}{2} \\ \beta_{i,j} \frac{T^2}{2} & T\beta_{i,j} \end{bmatrix} \\ \beta_{i,j} &= \begin{bmatrix} \sigma_x^2 a_{x,i} a_{x,j} + \sigma_y^2 a_{y,i} a_{y,j} + \sigma_z^2 a_{z,i} a_{z,j} \end{bmatrix} \end{aligned} \tag{6.16}$$

For convenience, the measurement update equations of the filter are repeated below.

$$y = I_{2N\times 2N}x + v \tag{6.17}$$

As stated before, the channels of the receiver are not coupled through the state transition matrix or the measurement update equations. The coupling of the channels comes from the process noise covariance matrices. Mathematically, the coupling comes from the matrices in (6.15) and (6.16) having off-block diagonal elements. These off-block diagonal terms result in a Kalman gain with off-block diagonal terms as well. This means that a pseudorange or pseudorange-rate residual in one channel will affect the pseudorange or pseudorange-rate estimates in the other channels.

The physical reason for the coupling of the channels is understood by considering how a sudden change in position, velocity, clock bias or clock drift effects the pseudoranges and pseudorange-rates. An abrupt change in clock bias or drift will manifest itself equally in all the channels of the receiver. Similarly, a shift in the receiver's position or velocity will affect all the channels of the receiver. The magnitude of the affect is determined by the line-of-sight vectors for each satellite. The amount of coupling between channels due to the receiver dynamics is determined by the term $\beta_{i,j}$ in (6.16). The $\beta_{i,j}$ term is essentially the dot product of the line of sight vectors for ith and jth satellites.

The receiver channels can be easily decoupled in the pseudorange-state VDFLL. This is accomplished by replacing the process noise covariance matrices in (6.15) and (6.16) with those in (6.18).

$$
\begin{aligned}
\bar{Q}^{\rho}_{clk} &= \begin{bmatrix} Q^c & 0_{2\times2} & \dots & 0_{2\times2} \\ 0_{2\times2} & Q^c & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{2\times2} \\ 0_{2\times2} & \dots & 0_{2\times2} & Q^c \end{bmatrix} \\
\bar{Q}^{\rho}_{dyn} &= \begin{bmatrix} Q^d_{1,1} & 0_{2\times2} & \dots & 0_{2\times2} \\ 0_{2\times2} & Q^d_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{2\times2} \\ 0_{2\times2} & \dots & 0_{2\times2} & Q^d_{N,N} \end{bmatrix}
\end{aligned}
\tag{6.18}
$$

The matrices in (6.18) are block diagonal and using them in the VDFLL filter eliminates the coupling between the channels of the receiver. The Kalman gain matrix resulting from using the matrices in (6.18) is also block diagonal. Therefore, the pseudorange and pseudorange-rate residuals from one satellite will not effect the estimated pseudoranges and pseudorange-rates for the other satellites.

6.3.1 Pseudorange State VDFLL Covariance Analysis

Using the same process noise tunings, the performance of the coupled and uncoupled filters over a range of C/N_0's can be computed. The improvement in performance brought about by coupling the channels of the receiver together is ascertained by studying the performance of the two filters. The steady state, state covariance matrices of the two filters are used for the comparison. Just like in the weighted least squares analysis, the performance improvement from vector tracking is dependent on the number of satellites and their geometry. To determine the improvement available to a typical GPS receiver, the 18 hours of satellite constellation data is used again.

To perform the covariance analysis, the measurement noise statistics as a function of C/N_0 are needed. The variance of the pseudorange noise is again related to C/N_0 by the formula shown in (6.10). For the pseudorange-rate measurements, the variance of the noise is related to C/N_0 by (6.19) [Crane, 2007].

$$\begin{aligned} \sigma_{\dot{\rho}}^2 &= \left(\frac{2}{(T \cdot 10^{(C/N_o)/10})^2} + \frac{2}{T \cdot 10^{(C/N_o)/10}} \right) \alpha \ (\mathrm{m/s})^2 \\ \alpha &= \left(\frac{c}{\pi f_{L1} T} \right)^2 \end{aligned} \tag{6.19}$$

The formula in (6.19) comes from the frequency error discriminator used elsewhere in this dissertation. The term T in (6.19) is the predetection integration time used by the receiver. A value of 20 ms is again used for T.

The process noise terms σ_x, σ_y, and σ_z are all set equal to a value of 10 m^2/s^4. Varying the magnitude of the process noise from 2 to 50 m^2/s^4 had a negligible affect on the results of the covariance analysis.

Two covariance analyses were performed on the GPS live-sky data. The first assumed that the satellite signals are all received at the same C/N_0. Using the process and measurement noise statistics, the steady state, state covariance matrices for the coupled and decoupled pseudorange state filters are computed. The variances of the predicted pseudorange-rates are then compared. For the decoupled filter, the pseudorange-rate variances in the different channels are equal, given that the process noise terms are equal and the C/N_0 ratios are equal. However, as was the case in the weighted least squares analysis, in the coupled filter the pseudorange-rate variances will in general be different for the different channels. Therefore, the maximum and minimum pseudorange-rate variances are used to determine the maximum and minimum performance gain from the coupling.

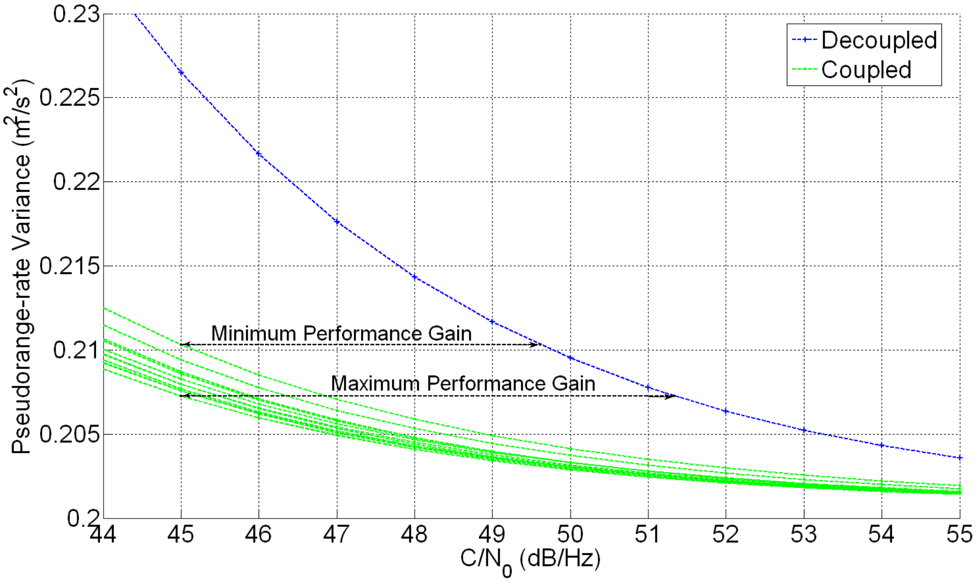

Figure 6.3: Pseudorange-Rate Variance Versus C/N_0 for Coupled and Decoupled Filters

Figure 6.3 shows the pseudorange-rate variances for the different satellites calculated from the coupled and decoupled filters. The performance curves for the different channels all lie on top of each other for the decoupled filter. For the coupled filter, the different channels of the receiver have varying pseudorange-rate variances. To determine the improvement in performance, the pseudorange-rates for the coupled filter are computed at 45 dB-Hz. The increase in C/N_0 needed to make the decoupled pseudorange-rate variances equal to the largest, coupled filter pseudorange-rate is defined as the minimum gain in performance. Similarly, the increase in C/N_0 needed to make the decoupled pseudorange-rate variances equal to the smallest, coupled filter pseudorange-rate is defined as the minimum gain in performance. Using these definitions, the maximum and minimum performance gains were calculated for the live-sky data. Figure 6.4 shows the results of the covariance analysis for the recorded data. In Figure 6.4, the maximum performance gain ranges from 3.88 to 6.63 dB and has a mean of 5.6 dB. The average minimum performance gain is 3.13 dB and it varies from 1.23 to 4.9 dB.

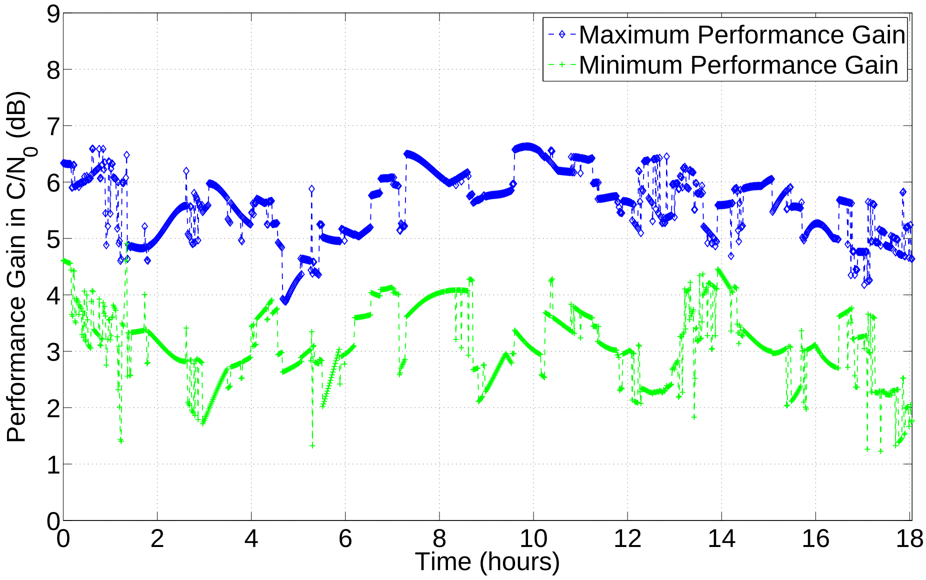

Figure 6.4: Maximum and Minimum Performance Gain in C/N_0 from Vector Tracking Assuming Equal Signal Powers.

The second covariance analysis did not assume that all the C/N_0 ratios are equal. Along with their positions, the C/N_0 ratios of the satellites were also recorded over the 18 hour period. With the C/N_0 ratios being different, the pseudorange-rate variances will no longer be equal for the decoupled filter. To determine the maximum and minimum performance gains from vector tracking, the pseudorange-rate variances for the coupled filter are again computed at 45 dB-Hz. The pseudorange-rate variances are then calculated for the decoupled filter. For each channel, the increase in C/N_0 needed to make the pseudorange-rate variance from the decoupled filter equal to the pseudorange-rate variance from the coupled filter is determined. The largest required increase in C/N_0 is defined as the maximum performance gain. The smallest required increase in C/N_0 is defined as the minimum performance gain. Figure 6.5 shows maximum and minimum performance increases from vector tracking with the varying C/N_0 ratios.

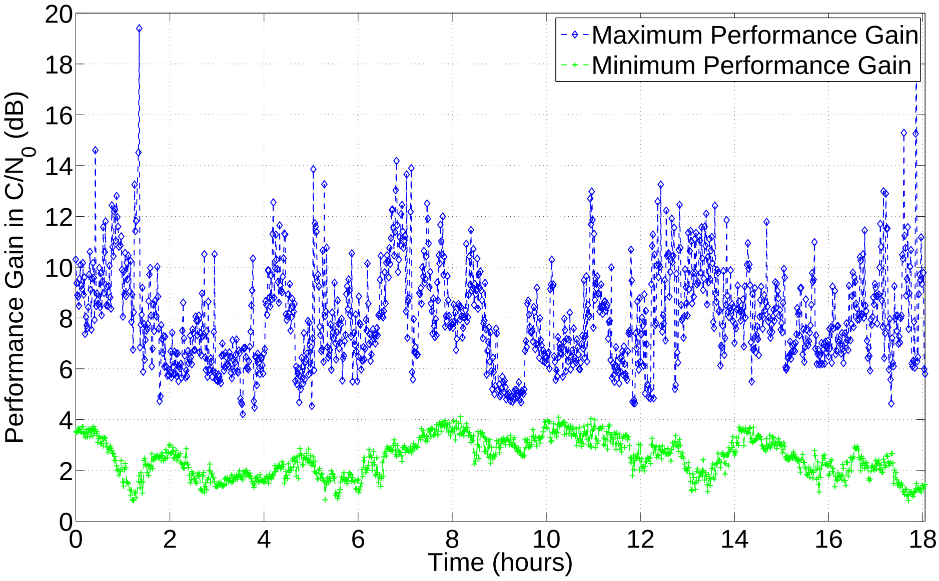

Figure 6.5: Maximum and Minimum Performance Gain in C/N_0 for Vector Tracking Using Actual Signal Powers

In Fig. 6.5 the maximum performance gain has a maximum value of 19.4 dB and a minimum of 4.21 dB. The average maximum gain is about 8 dB. The minimum gain ranges from .82 to 4.12 dB, and has a mean value of 2.5 dB. Comparing Fig. 6.4 and 6.5, the maximum performance gain is higher and the minimum gain is lower when the C/N_0 ratios of the satellite signals are not equal. The disparity in the minimum and maximum gains in

Fig. 6.5 comes from the wide range in the received C/N_0 ratios. The minimum performance gain generally corresponds to the signal with the highest C/N_0 ratio. Since the signal is already strong, it gains the least from the inclusion of the weaker signals. On the other hand, the maximum performance gain is generally associated with the channel with the lowest C/N_0 ratio. The weakest signal stands to gain the most from the inclusion of the other, stronger satellite signals in the coupled filter. In Fig. 6.5, the maximum performance gain of 19.4 dB occurs when a signal with a C/N_0 ratio of 28.8 dB-Hz is received in the presence of several other signals with C/N_0 ratios of in the neighborhood of 47 dB-Hz. In this situation, the variance of the pseudorange-rate predictions for the 28.8 dB-Hz significantly decreases when the channels of the receiver are coupled.

6.4 Conclusion

The advantages in performance offered by vector tracking are discussed in this chapter. A simple example using instantaneous weighted least squares is used to explain the reason vector tracking works better than scalar tracking loops in most situations. Vector tracking essentially exploits an overdetermined system of equations. As more satellite signals become available, the number of unknowns in the vector tracking formulation remains constant. In contrast, scalar tracking algorithms do not exploit the overdetermined system of equations. Scalar tracking algorithms do not benefit from increasing the number of available satellite signals.

The pseudorange-state formulation of vector is used to make a valid comparison between scalar and vector tracking loops. The filter in the pseudorange-state formulation can be operated in either the coupled or decoupled modes. Information is shared between the channels of the receiver in the coupled mode. In the decoupled mode the various channels of the receiver operate independently and do not share information. This property of the pseudorange-state filter allows the advantage offered by vector tracking to be isolated and computed for a given situation.

The performance improvement offered by vector tracking is studied by using 18 hours of satellite constellation data recorded at Auburn University. The performance gain offered by vector tracking is computed for each recorded satellite configuration in two different scenarios. In the first scenario, all the satellite signals have the same C/N_0 ratio. The second scenario uses the actual C/N_0 ratios that were recorded along with the satellite data. For the first scenario, the minimum gain from vector tracking has a mean of 3.13 dB and varies from 1.23 to 4.9 dB. The maximum gain from vector tracking in the first scenario ranges from 3.88 to 6.63 dB and has a mean of 5.6 dB. In the second scenario, there are periods where the maximum gain from vector tracking exceeds 19 dB. This occurs when weak signals are received in the presence of much stronger signals. The weak signals benefit greatly from the information sharing with the more powerful signals.

Chapter 7

Vector Tracking Loss of Lock Performance Analysis

7.1 Introduction

This chapter studies the performance of the Vector Delay/Frequency Lock Loop (VDFLL) as a function of C/N_0 ratio and receiver dynamics. The first section discusses analysis tools for the scalar frequency lock loop and extends them to the vector tracking case. Analysis tools are developed to evaluate the vector delay/frequency lock loop's frequency tracking capabilities. Most of the analysis will rely on the results of Kalman filtering. The results will be approximations, since the residuals supplied to the filter are nonlinear functions of its states. The noise in the residuals is also non-Gaussian. The analysis is also inherently scenario dependent since the pseudorange and pseudorange-rate residuals only contain information in the line-of-sight of the satellites. However, the linearization of the pseudorange equation is a very good approximation of the nonlinear function. This is due in great part to the large distances between the receiver and the satellites [Wendel et al., 2006]. Also, the noise sources both have quasi-Gaussian probability density functions. The minimum C/N_0 ratio at which the VDFLL can reliably operate is predicted analytically from rule of thumb tracking thresholds. The performance of the VDFLL is then compared to frequency locked loops of varying bandwidths. Intuitively, the vector tracking architecture should have advantages in performance over the traditional tracking loop approach. Since all the signals are tracked in unison, one would expect the signal tracking errors in all channels to be reduced.

The second section of this chapter verifies the results of the primary analysis with Monte Carlo simulations. The VDFLL is simulated with varying C/N_0 ratios, process noise tunings, and different satellite geometries. Its ability to function in the different scenarios is compared to the performance predicted by the rule of thumb analysis. The effects of requiring the algorithms to estimate the C/N_0 ratios, versus providing them *a priori* knowledge of the C/N_0 ratios, are studied with the Monte Carlo simulations. The position-state and pseudorange-state VDFLL filters are both simulated. Their performance is compared to that of the decoupled pseudorange-state filter. The comparison is done to study the gain offered solely by the coupling of the receiver's channels together.

Finally, the third section of this chapter uses data collected from a GPS constellation simulator to study the VDFLL's performance in high dynamics and at low C/N_0 ratios. The analysis tools developed in the second section are applied to the dynamic scenarios. The predictions from the rule of thumb analysis and the VDFLL's actual performance are then compared.

7.2 Rule of Thumb Analysis Techniques

7.2.1 Frequency Lock Loop Analysis

In a GPS receiver, the carrier tracking loops are the most susceptible to losing signal lock. This is due to the significantly shorter wavelength of the carrier signal [Jwo, 2001]. Since carrier tracking is the major limiting factor of vector tracking, the ability of the VDFLL to track the carrier signals is analyzed. However, the same type of analysis can be performed with respect to code tracking for a VDLL [Benson, 2007]. The frequency tracking performance of the VDFLL is compared to the performance of several scalar FLL's with different noise bandwidths. The scalar FLL's are based on the prescribed loop filter parameters given in [Ward, 1996b]. The metrics for assessing the performance of both the scalar and vector FLL's are drawn from [Ward, 1998].

The dominant sources of error in frequency locked loops are thermal noise and dynamic stress. Thermal noise is the source of the noise terms present in the correlator output formulas. Dynamic stress is caused by the platform dynamics of the receiver. The rule of thumb tracking threshold for a FLL is [Ward, 1998]:

$$3\sigma_w + f_e \leq \frac{.25}{T^*}\text{Hertz} \tag{7.1}$$

where

$$\sigma_w = \text{1-sigma frequency jitter from thermal noise}$$

$$f_e = \text{dynamic stress error}$$

In an FLL, the 1-sigma frequency jitter caused by thermal noise is [Ward, 1998]:

$$\sigma_w = \frac{1}{2\pi T^*}\sqrt{\frac{4FB_n}{C/N_o}\left(1+\frac{1}{T^*C/N_o}\right)}(Hz) \tag{7.2}$$

where

$$\begin{aligned} F &= 1 \text{ at high } C/N_o \\ &= 2 \text{ near threshold} \\ B_n &= \text{Noise Bandwidth (Hz)} \end{aligned}$$

In (7.1) and (7.2), T^* is the integrate and dump time used to generate the correlator outputs used by the frequency discriminators. For this dissertation, a integration time of

10 ms is used for the frequency discriminators and FLL analysis. The 10 ms integrate and dump period is the longest coherent integration time allowed by the 50 Hz data message.

The term f_e in (7.1) is the dynamic stress error in the FLL caused by the line of sight dynamics between the receiver and the satellite. Given the FLL noise bandwidth and using the loop filter characteristics in [Ward, 1996b], the term f_e is calculated based on the steady state error present in the FLL.

Using position, velocity, and acceleration states in the VDFLL's EKF provides it the ability to track step and ramp changes in velocity with zero steady state error. However, the VDFLL cannot track a ramp change in acceleration with zero steady state error. Among scalar FLL's, only a second-order FLL has comparable capabilities. A second-order FLL can track step and ramp changes in the line of sight velocity with zero steady state error. The second-order FLL cannot track a ramp change in line of sight acceleration with zero steady state error. For these reasons, the performance of the VDFLL algorithm is compared to that of several different scalar second-order FLL's. The noise bandwidths of the FLL's that are studied represent typical values used for low, moderate and high levels of user dynamics. Using the loop filter parameters given in [Ward, 1996b], the dynamic stress error for a second-order FLL with noise bandwidth B_n is calculated as:

$$f_e = F_M \frac{f_{L_1}}{c} \left(\frac{.53}{B_n} \right)^2 \tag{7.3}$$

where

$$F_M = \text{Jerk dynamics } (m/s^3)$$

For the VDFLL, the value of σ_w is calculated from the steady state covariance matrix of the EKF (P_{ss}^-). The superscript $-$ in P_{ss}^- denotes the state covariance prior to the measurement update. This notation will be used throughout this dissertation. A superscript + will denote a quantity after a measurement update. The dynamic stress error for the VDFLL is calculated using state space techniques, as described in the following sections.

7.2.2 Vector Delay/Frequency Lock Loop Analysis

In order to use the steady state covariance matrix of the EKF, it is assumed that the satellite geometry is fixed, the measurements from all satellites arrive concurrently, and that the mean of the errors in the states of the EKF are small. The steady state covariance matrix is a function of the process noise in the system (5.2). The magnitude of the values used for σ_x^2, σ_y^2, and σ_z^2 are determined from an expected level of user dynamics. In order to isolate the effects of the thermal noise, a two part procedure is used. First, the steady state Kalman gain (K_{ss}) is determined, using values of σ_x^2, σ_y^2, and σ_z^2 for a given level of dynamics. Second, P_{ss}^- is determined using K_{ss} with the only noise sources being the clock noise, and the thermal noise in the correlator outputs.

The values of K_{ss} and P_{ss}^- are calculated by converting the system dynamics to discrete time and iterating the discrete time Kalman filter equations. A 20 millisecond interval is assumed for the integrate and dump operations and for the interval between measurement epochs. Once K_{SS} is determined, the Kalman filter equations are iterated with the process noise terms σ_x^2, σ_y^2, and σ_z^2 set to zero. To do this, the Kalman filter equations must be augmented. The standard equation for the state covariance matrix after a measurement update is [Brown and Hwang, 1996]:

$$P^+(k) \quad = \quad (I - K(k)H(k))P^-(k) \tag{7.4}$$

However, this equation is only valid when using the optimal gain K. The general equation for the state covariance after a measurement update is [Brown and Hwang, 1996]:

$$P^{+}(k) \;=\; (I - K(k)H(k))P^{-}(k)(I - K(k)H(k))^{T} + K(k)R(k)R(k)^{T} \tag{7.5}$$

The diagonal elements of the steady state covariance matrix P_{ss}^{-} are the variances associated with each element of the error state vector. The variances of the error states are related to the frequency jitter using the observation matrix H. The error between the true frequency at time k and the predicted frequency at time k is approximately:

$$f_{received,k} - f_{predicted,k} \;\approx\; \frac{f_{L1}}{c} H\hat{x}^{-}(k) \tag{7.6}$$

The elements of the matrix H are:

$$H \;=\; \begin{bmatrix} 0_{1\times 3} & a_{xj} & a_{yj} & a_{zj} & 0_{1\times 4} & -1 \end{bmatrix} \tag{7.7}$$

The terms a_{xj}, a_{yj}, and a_{zj} are the elements of the line-of-sight unit vector from the receiver's position to the jth satellite. The errors between the received and predicted frequencies are represented by the vector ϵ. The covariance of ϵ is:

$$\begin{aligned} \epsilon \;&=\; (f_{received,k} - f_{predicted,k}) \\ E\{\epsilon\epsilon^{T}\} \;&\approx\; \frac{f_{L1}^{2}}{c^{2}} HP^{-}(k)H^{T} \end{aligned} \tag{7.8}$$

Once P_{ss}^{-} is calculated, the frequency jitter in each channel is found by inserting P_{ss}^{-} into (7.8), and examining the corresponding diagonal elements.

7.2.3 Vector Delay/Frequency Lock Thermal Noise Performance

The performance of the VDFLL is inherently scenario-dependent. The number of available satellites, the satellite geometry, and the type of clock used by the receiver vary widely and all impact the tracking abilities of the VDFLL. Therefore a specific scenario is used to illustrate the performance of the VDFLL relative to scalar FLL's. There are a total of eleven satellites in the scenario. The full eleven satellites are used and two subsets of the eleven satellites are also considered; one subset uses seven satellites, and the other uses five. The Dilution Of Precision (DOP) values for the different sets of satellites are shown in Table 7.1 [Misra and Enge, 2001].

For the receiver clock, a temperature compensated crystal oscillator (TCXO) is assumed. The power spectral density coefficients used for the oscillator are [Brown and Hwang, 1996]:

$$\begin{aligned} h_0 &= 2 \cdot 10^{-19} \\ h_{-1} &= 7 \cdot 10^{-21} \\ h_{-2} &= 2 \cdot 10^{-20} \end{aligned} \tag{7.9}$$

Table 7.1: Dilution of Precision Values for Different Satellite Configurations

Number of Satellites	PDOP	HDOP	VDOP
Eleven	1.4141	0.7527	1.1971
Seven	1.7503	1.0003	1.4363
Five	2.3602	1.1300	2.0721

Discriminator functions are used to process the correlator outputs and produce measurements of the code phase error and frequency error. Two separate discriminator functions are used in the algorithms in this paper. The first algorithm uses the correlator outputs accumulated over the entire integrate and dump interval [Crane, 2007]:

$$\begin{aligned}
Y_R(k) &= IE(k)^2 + QE(k)^2 - IL(k)^2 - QL(k)^2 \qquad (7.10)\\
Y_R(k) &= A(k)^2\Big(\frac{2\cdot\rho_e(k)}{\beta}\Big) + v_R(k)\\
\rho_e(k) &= \text{Line of sight range error (meters)}\\
\beta &= \text{PRN chip length, 293.3 meters for C/A code}\\
A(k) &= \text{Signal amplitude}\\
v_R(k) &= \text{Noise}
\end{aligned}$$

The mean and mean squared value of the random variable v_R are:

$$\begin{aligned}
E\{v_R\} &= 0\\
E\{v_R^2\} &= 8\sigma_F^4 + 4A^2\sigma_F^2 f(\rho_e) \qquad (7.11)\\
f(\rho_e) &= 2\frac{\rho_e^2}{b^2} + \frac{1}{2}
\end{aligned}$$

As a function of C/N_o :

$$E\{v_R^2\} = \frac{\beta^2}{2(TC/N_o)^2} + \frac{\beta^2}{TC/N_o}\left(\frac{\rho_e^2}{\beta^2} + \frac{1}{4}\right) \text{ (meters}^2\text{)}$$

A second discriminator is used to produce measurements of the frequency error $f_{err}(k)$ [Crane, 2007]:

$$
\begin{aligned}
Y_{RR}(k) &= IP_2(k)QP_1(k) - IP_1(k)QP_2(k) && (7.12)\\
Y_{RR}(k) &\approx -A_F(k)^2R^2(\rho_e(k))\frac{\pi f_{err}(k)T}{4} + v_{RR}\\
\rho_e(k) &= \text{Line of sight range error (meters)}\\
f_{err}(k) &= \text{Frequency error (Hz)}\\
v_{RR} &= \text{Noise}
\end{aligned}
$$

The mean and mean squared value of v_{RR} are:

$$
\begin{aligned}
E\{v_{RR}\} &= 0\\
E\{v_{RR}^2\}^\dagger &= \frac{\sigma_F^4}{2} + \frac{A^2}{4}R^2(\rho_e(k))\sigma_F^2 && (7.13)
\end{aligned}
$$

As a function of C/N_o :

$$
\begin{aligned}
E\{v_{RR}^2\}^\dagger &= \left(\frac{2}{(TC/N_o)^2} + \frac{2R^2(\rho_e(k))}{TC/N_o}\right)\left(\frac{1}{\pi T}\right)^2 (\text{Hz})^2\\
\dagger &= \text{assuming } |\rho_e(k)| \leq \frac{\beta}{2}
\end{aligned}
$$

In (7.11) and (7.13) the value of T is the predetection integration interval used to generated the correlator outputs in (7.10), which is 20 ms in this dissertation. The noise terms in the discriminator outputs in (7.11) and (7.13) are uncorrelated with each other [Crane, 2007].

Figure 7.1 shows the 1-σ frequency jitter for the VDFLL algorithm with the subset of five satellites being tracked. In Fig. 7.1 there is no dynamic stress, therefore only the effects of thermal noise are shown. The maximum diagonal entry of the matrix from (7.8) is plotted for each different C/N_o ratio. This is the channel of the VDFLL with the largest frequency jitter. The frequency jitter is calculated with varying magnitudes for the process noise terms σ_x^2, σ_y^2, and σ_z^2. In each scenario, the magnitudes of the three variables are identical. The magnitudes used for the process noise are 2, 12, 50, and 125 m^2/s^4. These values correspond to increasingly high levels of user dynamics. The 1-σ frequency jitter threshold is 8.333 Hz. The performance of the algorithm degrades with respect to rejecting thermal noise disturbances as the process noise level is increased. The thermal noise performance of three different scalar FLL's is also plotted in Fig. 7.1. The FLL's have noise bandwidths of 2 Hz, 10 Hz, and 18 Hz. The thermal noise jitter for the FLL's is calculated from (7.2). Using the lowest level of process noise, the VDFLL exceeds the 1-σ frequency jitter threshold at 13 dB-Hz and the 2 Hz noise bandwidth FLL exceeds the threshold at about 20.5 dB-Hz. With the highest level of process noise, the VDFLL exceeds the threshold at 16.5 dB-Hz, while the 18 Hz noise bandwidth FLL exceeds it at 27.8 dB-Hz. Depending upon its tuning, the VDFLL can function at anywhere from 4 dB-Hz to 14.8 dB-Hz lower than a scalar FLL for the five satellite scenario. Recall that this improvement in tracking is with respect to thermal noise only.

Figures 7.2 and 7.3 show the 1-σ frequency jitter for the VDFLL algorithm with both the subset of 7 satellites and the full 11 satellites being tracked, respectively. The performance of the scalar 2 Hz, 10 Hz, and 18 Hz noise bandwidth FLL's is also plotted. It is apparent from the figures that the ability of the VDFLL algorithm to operate in low C/N_o environments is dependent on the number of satellites being tracked. Comparing Fig. 7.1 and Fig. 7.2, the VDFLL's performance increases about 1 dB-Hz with the addition of 2

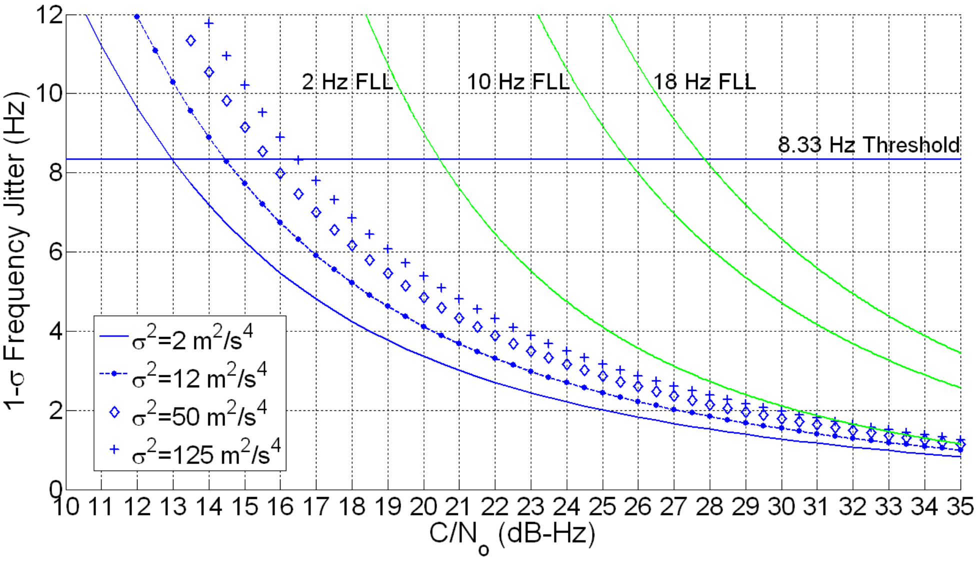

Figure 7.1: VDFLL Thermal Noise Jitter Versus C/N_o When Tracking Five Satellites

satellites. Another increase of about 1 dB-Hz in noise performance is seen in Fig. 7.3 with the further addition of 4 satellites. The VDFLL can function at anywhere from 5 dB-Hz to 15.8 dB-Hz lower than a scalar FLL for the seven satellite scenario, and 6 dB-Hz to 16.8 dB-Hz lower in the eleven satellite scenario.

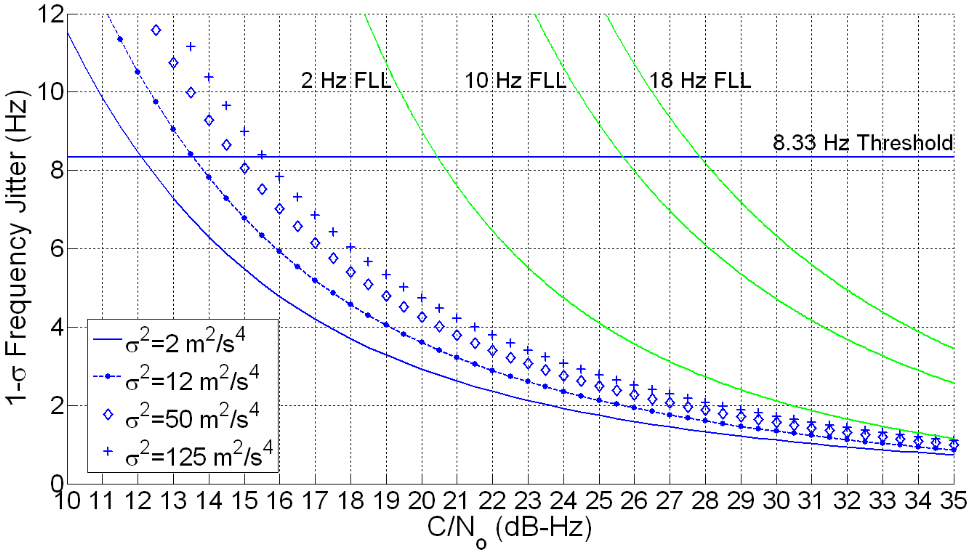

Figure 7.2: VDFLL Thermal Noise Jitter Versus C/N_o When Tracking Seven Satellites

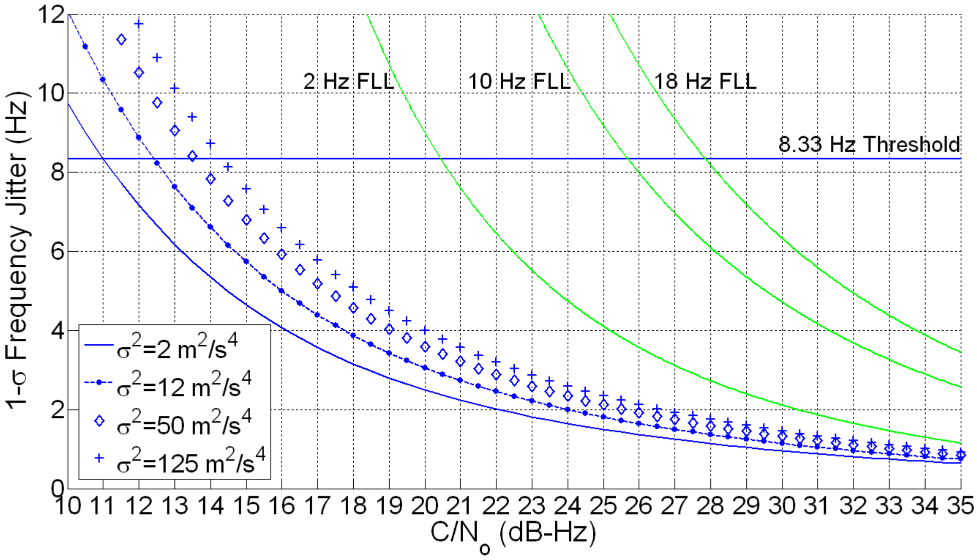

Figure 7.3: VDFLL Thermal Noise Jitter Versus C/N_o When Tracking Eleven Satellites

7.2.4 Vector Delay/Frequency Lock Loop Dynamic Stress Steady State Performance

Recall that the state vector of the VDFLL EKF contains position, velocity, and acceleration states (5.1). When a change occurs in the receiver's acceleration states, this is reflected by modifying the system model with an input vector:

$$\dot{x} = Ax + B_w w + Bu$$

where

$$B = \begin{bmatrix} 0_{3\times3} \\ 0_{3\times3} \\ I_{3\times3} \\ 0_{2\times3} \end{bmatrix} \tag{7.14}$$

when discretized:

$$B^d = \begin{bmatrix} (\frac{T^3}{6})I_{3\times3} \\ (\frac{T^2}{2})I_{3\times3} \\ (T)I_{3\times3} \\ 0_{2\times3} \end{bmatrix} \tag{7.15}$$

In order to study the effects of the input, it is useful to consider two separate Kalman filters. One is the simple Kalman filter which does not include the effects of a change in receiver acceleration. Its model is shown in (5.1) and its states are denoted as $\hat{x}$. The second Kalman filter uses the model in (7.14) with perfect knowledge of the input vector u at all times. Its states are denoted as $\tilde{x}$. First assume that up to time k, the input vector u has been all zeros, which means that the states of the two filters are equal. At time k, an input is applied to the system. The state vectors of the two filters prior to the measurement update at time $k+1$ are then [Chan et al., 1979]:

$$\begin{aligned}\tilde{x}^-(k+1) &= A^d\tilde{x}^+(k) - B^d u(k) \quad (7.16)\\ &\text{and}\\ \hat{x}^-(k+1) &= A^d\hat{x}^+(k)\end{aligned}$$

The difference between the two state vectors prior to and after the measurement update at time $k+1$ is:

$$\begin{aligned}\tilde{x}^-(k+1) - \hat{x}^-(k+1) &= M^-(k)u(k) \quad (7.17)\\ \tilde{x}^+(k+1) - \hat{x}^+(k+1) &= M^+(k)u(k)\\ &\text{where}\\ M^-(k) &= B^d \quad (7.18)\\ M^+(k) &= M^-(k) - K(k+1)HM^-(k)\end{aligned}$$

Assuming the input is constant, the subscript k can be omitted from u and the difference between the states of the two filters at time $k+1$ is:

$$\begin{aligned}\tilde{x}^-(k+2) - \hat{x}^-(k+2) &= M^-(k+1)u \quad (7.19)\\ \tilde{x}^+(k+2) - \hat{x}^+(k+2) &= M^+(k+1)u\\ &\text{where}\\ M^-(k+1) &= B^d + A^d M^+(k) \quad (7.20)\end{aligned}$$

$$M^{+}(k+1) = M^{-}(k+1) - K(k+2)HM^{-}(k+1)$$

The matrices M^{-} and M^{+} can be calculated at each time after the onset of the input by iterating (7.20). From (7.17) and (7.19) it is obvious that the states of the VDFLL's filter are biased if an input occurs to the acceleration states. A constant input causes a steady state error in the filter's estimates of acceleration, velocity, and position. The steady state error of the VDFLL's predicted velocity is calculated by iterating (7.20) until M^{-} converges to a steady state matrix. The effect of the steady state error on the VDFLL's frequency predictions is then computed from:

$$\begin{aligned}
\epsilon &= (f_{received} - f_{predicted}) \\
E\{\epsilon_{ss}\} &\approx \frac{f_{L1}}{c} E\{H\hat{x}_{ss}^{-}\} \\
\epsilon_{ss}^{-} &\approx \frac{f_{L1}}{c} HM_{ss}^{-}U \qquad (7.21)
\end{aligned}$$

The steady state error ϵ_{ss}^{-} in the VDFLL's frequency prediction is used as the dynamic stress error f_e used in (7.1). The effects of thermal noise are again calculated from (7.8). In general, the steady state frequency tracking error is different for all the visible satellites. This is because the steady state velocity errors caused by the acceleration inputs must be expressed in the line of sight of each satellite. Since the satellites have different line of sight vectors, the steady state frequency tracking error is also different for each of the satellites.

The maximum dynamic stress the VDFLL algorithm can tolerate at different C/N_o ratios is determined from using (7.1), (7.8), and (7.21). A step jerk input is applied to all three acceleration states. The magnitude of the step input to each axis is identical. For a given C/N_o ratio, the steady state frequency prediction error is calculated for a range of step input magnitudes. The thermal noise frequency jitter is also calculated for the specific C/N_o ratio. The largest step input magnitude that does not exceed the tracking threshold for all satellites is determined to be the maximum dynamics stress. As the magnitude of the step inputs increase, the tracking threshold is first exceeded by the satellite for which the dot product of its line of sight vector and the input vector is largest. For this reason, the jerk stress in the line of sight of the first satellite to cross the tracking threshold is plotted versus C/N_o ratio in the following figures. For all three sets of satellites, the dot product of the line of sight vector for the first satellite to cross the threshold and the input vector is approximately 1.6.

Figure 7.4 shows the maximum line of sight dynamic stress the VDFLL algorithm can tolerate at different C/N_o ratios when tracking the subset of 5 satellites. The maximum jerk stress is also calculated for varying values of σ_x^2, σ_y^2, and σ_z^2. For comparison, the maximum line of sight dynamic stress is also plotted for three different FLL's. The FLL's have noise bandwidths of 2 Hz, 10 Hz, and 18 Hz and use the loop filter parameters given in [Ward, 1996b]. The 2 Hz bandwidth FLL can operate down to about 20.5 dB-Hz but can only tolerate low dynamics. In contrast, the 18 Hz FLL can accommodate large dynamics, but cannot function below about 28 dB-Hz. Figure 7.5 gives a more detailed view of the performance of the VDFLL and FLL's at the lower C/N_o ratios. The VDFLL with σ^2 equal to 12 m^2/s^4 can tolerate higher dynamics and can operate at 6 dB-Hz lower than the 2 Hz FLL . The 18 Hz FLL outperforms the VDFLL with respect to the maximum tolerable dynamics for the various levels of process noise. However, the VDFLL can operate anywhere from 11.5 to 15 dB-Hz lower.

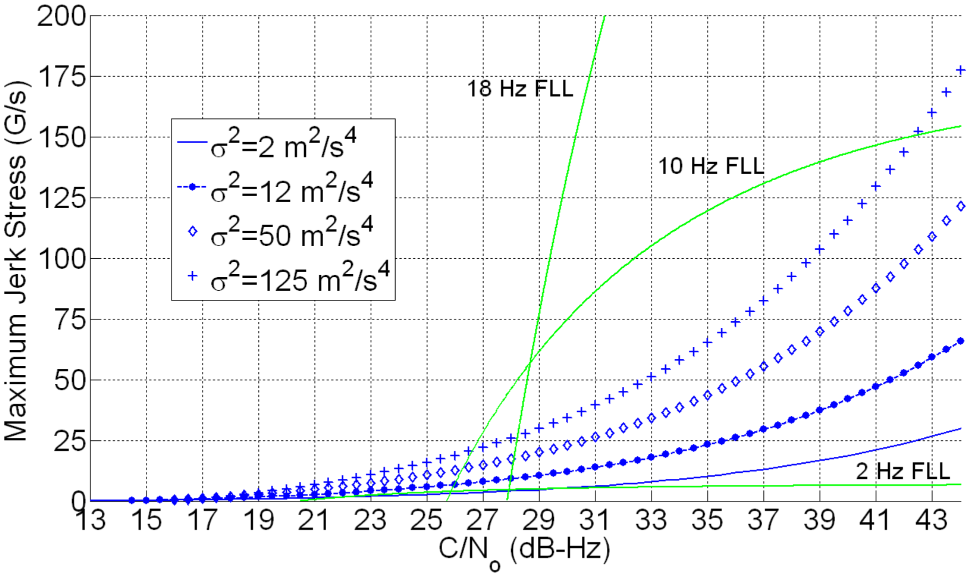

Figure 7.4: Maximum Jerk Stress Versus C/N_o When Tracking Five Satellites

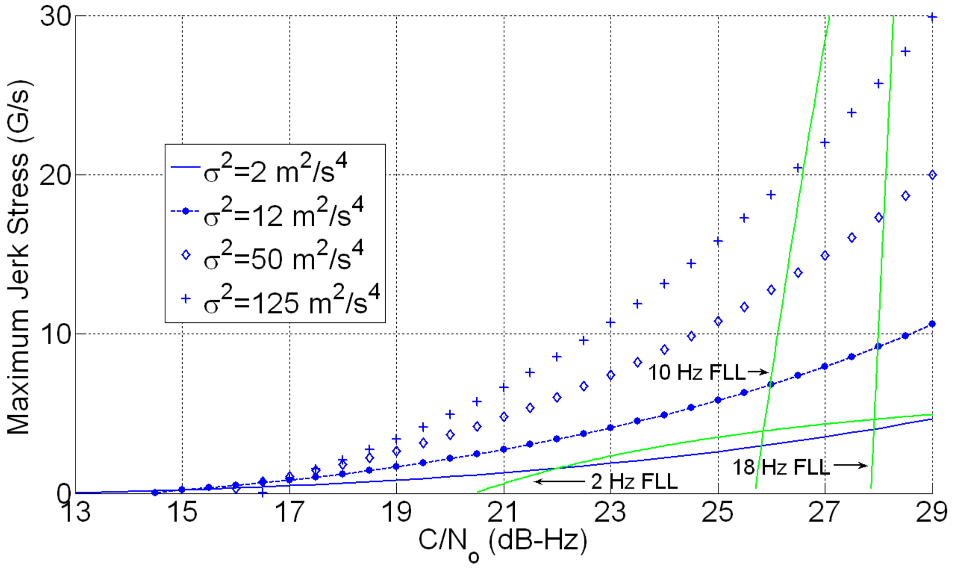

Figure 7.5: Maximum Jerk Stress Versus C/N_o When Tracking Five Satellites

Figures 7.6 and 7.7 show the improvements in maximum dynamic stress when the eleven satellites are tracked. Increasing the number of satellites from five to eleven effectively shifts the performance curves of the VDFLL only using the five satellites about 2 dB lower. This is the same result shown in Fig. 7.1 and Fig. 7.3.

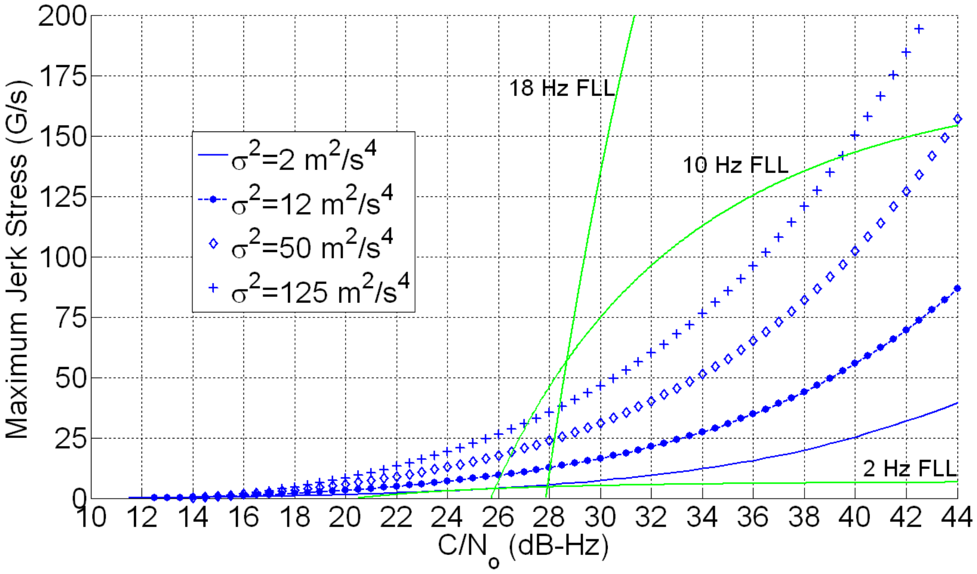

Figure 7.6: Maximum Jerk Stress Versus C/N_o When Tracking Eleven Satellites

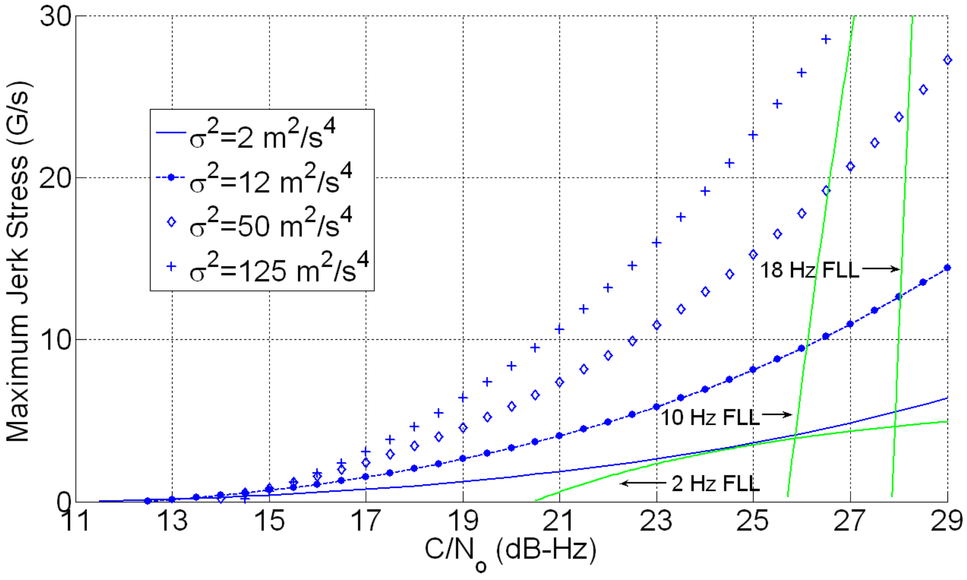

Figure 7.7: Maximum Jerk Stress Versus C/N_o When Tracking Eleven Satellites

7.2.5 Scenario Dependent Vector Delay/Frequency Lock Loop Dynamic Stress Performance

The analysis in the previous section assumed that the user dynamics were constant. Based on this assumption, the steady state errors caused by changes in the receiver's acceleration are calculated from the M_{ss}^{-} and M_{ss}^{+} matrices. The frequency tracking errors caused by time varying inputs can be calculated in a similar manner. Again, assume that up to time k the input vector u has been all zeros. Then the states of the two filters after the measurement update at time k are equal to each other. At time k an arbitrary input is applied to the system. The states of the two filters before and after the measurement update at time $k+1$ then differ by:

$$\begin{aligned}
\tilde{x}^-(k+1) - \hat{x}^-(k+1) &= M^-(k)u(k) && (7.22)\\
\tilde{x}^+(k+1) - \hat{x}^+(k+1) &= M^+(k)u(k) \\
\zeta(k+1) &= M^+(k)u(k) && (7.23)\\
\text{where} & \\
\zeta(k+1) &= \tilde{x}^+(k+1) - \hat{x}^+(k+1) \\
M^-(k) &= B^d && (7.24)\\
M^+(k) &= M^-(k) - K(k+1)HM^-(k)
\end{aligned}$$

The previous analysis assumed that the input at time k and all subsequent times were equal. Given another arbitrary input at time $k+1$, the states of the two filters after the next measurement update differ by:

$$\begin{aligned}
\zeta(k+2) &= M^+(k+1)u(k+1) + (A^d - K(k+2)H)\zeta(k+1) \\
\text{where} & \\
M^+(k+1) &= B^d - K(k+2)HB^d && (7.25)
\end{aligned}$$

Similarly, the difference in the two filter states after another arbitrary input at time $k+2$ is:

$$\begin{aligned}
\zeta(k+3) &= M^+(k+2)u(k+2) + (A^d - K(k+3)H)\zeta(k+2) \\
\text{where} &
\end{aligned}$$

$$M^+(k+2) \quad = \quad B^d - K(k+3)HB^d \tag{7.26}$$

These recursive equations can be carried out to find the errors in the states of the VDFLL caused by an arbitrary maneuver. They can also be easily rearranged to find the error between the filter states prior to the measurement update.

7.3 Monte Carlo Simulation Results

In this section Monte Carlo simulations are used to verify the thermal noise analysis presented in the previous section. Using the tracking threshold in [Ward, 1998] to analyze the VDFLL implicitly assumes that the scalar tracking loop rule of thumb threshold also applies to vector tracking loops. The efficacy of using the rule of thumb tracking thresholds to predict the performance of vector tracking algorithms is studied through Monte Carlo simulations. The algorithms were simulated over a range C/N_o ratios. At each C/N_o ratio, one hundred simulation runs of 60 second duration were carried out. The fraction of trials which the algorithms function for the full sixty seconds are plotted versus C/N_o ratio. The duration and number of the trials are deliberately chosen to mirror the values used in [Ward, 1997], where Monte Carlo simulations were similarly used to verify the rule of thumb tracking thresholds for the FLL.

7.3.1 Simulation Setup

For the Monte Carlo simulations, the received signals were simulated at the correlator output level. The various correlator outputs are modeled as shown in (7.27) [Sayre, 2003].

$$IE(k) \quad = \quad AR(\epsilon + \tau)D(k)\cos\left(\pi f_{err}T + \theta_{err}\right) + \eta_{IE}(k)$$

$$
\begin{aligned}
IP(k) &= AR(\epsilon)D(k)\cos\left(\pi f_{err}T + \theta_{err}\right) + \eta_{IP}(k) \\
IL(k) &= AR(\epsilon - \tau)D(k)\cos\left(\pi f_{err}T + \theta_{err}\right) + \eta_{IL}(k) \qquad (7.27) \\
QE(k) &= AR(\epsilon + \tau)D(k)\sin\left(\pi f_{err}T + \theta_{err}\right) + \eta_{QE}(k) \\
QP(k) &= AR(\epsilon)D(k)\sin\left(\pi f_{err}T + \theta_{err}\right) + \eta_{QP}(k) \\
QL(k) &= AR(\epsilon - \tau)D(k)\sin\left(\pi f_{err}T + \theta_{err}\right) + \eta_{QL}(k) \\
A &= \sqrt{\frac{2CT_{PDI}}{N_o}} \cdot \frac{\sin\left(\pi f_{err}T\right)}{\left(\pi f_{err}T\right)} \\
\eta &= N(0,1)
\end{aligned}
$$

In (7.27), the correlator outputs have been normalized so that the noise is unit variance. The variables f_{err} and θ_{err} are the frequency error (Hertz) and phase error (radians) between the received and locally generated carrier signals, respectively. The sinc (sin(x)/x) term in (7.27) is an effective loss in signal amplitude due to the frequency error, f_{err}. The term ϵ is the phase error between the received and locally generated PRN codes, measured in fractions of a chip. The function $R(\epsilon)$ is the auto-correlation function of the PRN sequences. The symbol τ is the correlator spacing of the Early and Late PRN replicas, expressed in fractions of a chip. A correlator spacing of one half chip is used for all the Monte Carlo simulation results. The autocorrelation of the Gold code as a function of the code phase error is:

$$
R(\epsilon) = \begin{cases} 1 - |\epsilon| & \text{when } \epsilon < 1 \text{ chip} \\ 0 & \text{when } \epsilon > 1 \text{ chip.} \end{cases}
$$

The noise terms in (7.27) are all zero-mean and unit variance. The in-phase and quadrature noise terms are uncorrelated. Using a half chip correlator spacing results in the early and late noise terms being uncorrelated. The early and prompt noise terms, and the prompt and late noise terms, all have a correlation coefficient of .5.

The pseudorange and pseudorange-rate discriminator functions in (7.11) and (7.13) are used in the simulations. An integrate and dump time of 20 milliseconds is used for the correlator outputs in (7.11). Two consecutive integrate and dump times of 10 ms are used for the correlator outputs used in (7.13).

The three satellite geometries from section 7.2.3 are used in the Monte Carlo simulations. A temperature compensated crystal oscillator with the power spectral density coefficients shown in (7.9) is used again.

A second order VDFLL algorithm is tested in the Monte Carlo simulations. Five levels of process noise are used by the VDFLL for each satellite constellation. The process noise terms σ_x^2, σ_y^2, and σ_z^2 are set equal to each other. The magnitudes of the process noise are .2, 2, 10, 50, 100, and 300 m^2/s^2. The VDFLL algorithm is simulated over a range of C/N_o ratios in increments of .5 dB-Hz for each satellite geometry and process noise tuning. The VDFLL does not use *a priori* information of the C/N_o ratios of the signals. Instead, the C/N_o ratios of the signals is estimated using the techniques in Appendix B.

Before the start of each simulation, the steady state, state covariance matrix is calculated. The state vector of the VDFLL filter is then initialized using a Gaussian random vector with a mean vector of zero and a covariance matrix equal to the steady state, state covariance matrix. These steps are taken to make sure the VDFLL filter's performance is being assessed in steady state.

Loss of lock is declared whenever any of the pseudorange errors exceeds a half chip. The reason for using this criteria is that the linear operating range of the pseudorange discriminator is one half chip. It was observed that a pseudorange error of over a half chip almost always immediately precedes a catastrophic failure of the algorithm. Using the pseudorange errors as the loss of lock criteria would seem to contradict the rule of thumb analysis. The rule of thumb threshold was a pseudorange-rate error of over $1/4T$, which might suggest that the pseudorange-rate errors should be monitored instead of the pseudorange errors. However, the VDFLL algorithm can tolerate occasionally exceeding the pseudorange-rate threshold and not lose lock. In the occasions when the algorithm does lose lock, errors in the position states tend to grow exponentially. The growing errors in the position states rapidly result in the pseudorange errors exceeding a half chip. In which case, loss of lock is declared using the pseudorange error threshold.

7.3.2 Simulation Results

Figure 7.8 shows the frequency jitter versus C/N_o ratio for the 11 satellite geometry and with a process noise tuning of .2 m^2/s^2. The channels of the VDFLL with the maximum and minimum frequency jitter are displayed. Depending on whether the maximum or minimum frequency jitters are used, the rule of thumb analysis predicts that the VDFLL should lose lock at about 1.7 or 3 dB-Hz.

Figure 7.9 shows the results of the Monte Carlo analysis for the 11 satellite geometry and .2 m^2/s^2 process noise tuning. The percent of trials that the VDFLL functions throughout the simulation is plotted versus C/N_o ratio. Vertical lines are plotted at the C/N_o ratios corresponding to where the channels with the minimum and maximum frequency jitters cross the rule of thumb tracking threshold. The Monte Carlo simulations reveal that there is a transition region where the probability of the VDFLL functioning

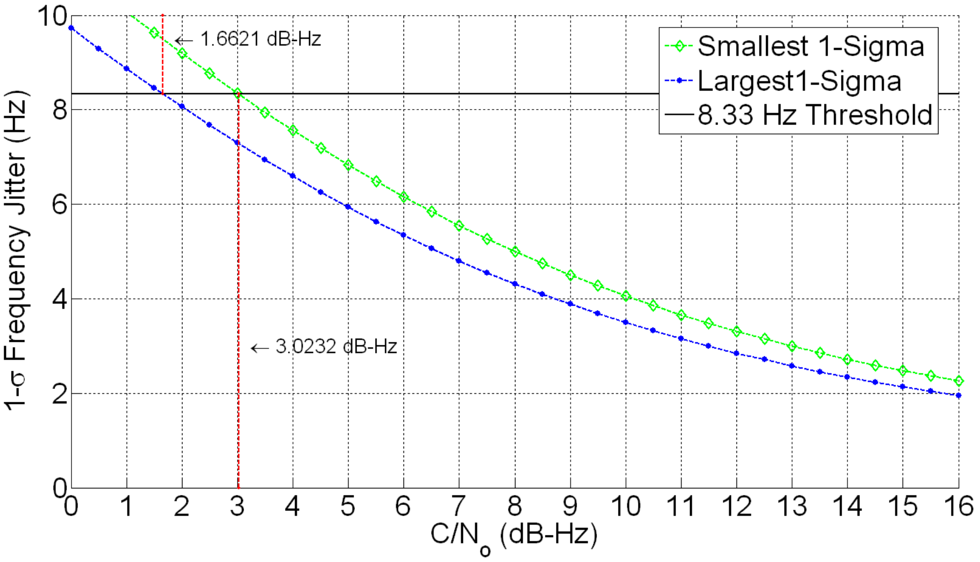

Figure 7.8: VDFLL Thermal Noise Analysis with 11 Satellites and σ_x^2= .2 m^2/s^2

goes from basically zero to one. For the simulation setup used to generate Figure 7.9, the transition region starts at approximately 2 dB-Hz and ends at about 7 dB-Hz. Ideally, the rule of thumb analysis would predict the point at which the probability of the algorithm to function is .5 [Ward, 1997]. The predictions from the rule of thumb analysis match the Monte Carlo results to within 3.3 (using the minimum frequency jitter) to 2 dB-Hz (using the maximum frequency jitter).

Figure 7.10 shows the results of the Monte Carlo analysis for the 11 satellite geometry and 2 m^2/s^2 process noise tuning. The vertical lines indicate the predicted thresholds from the rule of thumb analysis. The predicted thresholds are within .1 to 1 dB-Hz of the 50 percent probability of tracking crossing. Compared to Figure 7.9, the predicted thresholds match the results from the Monte Carlo simulations far better with the 2 m^2/s^2 process noise tuning.

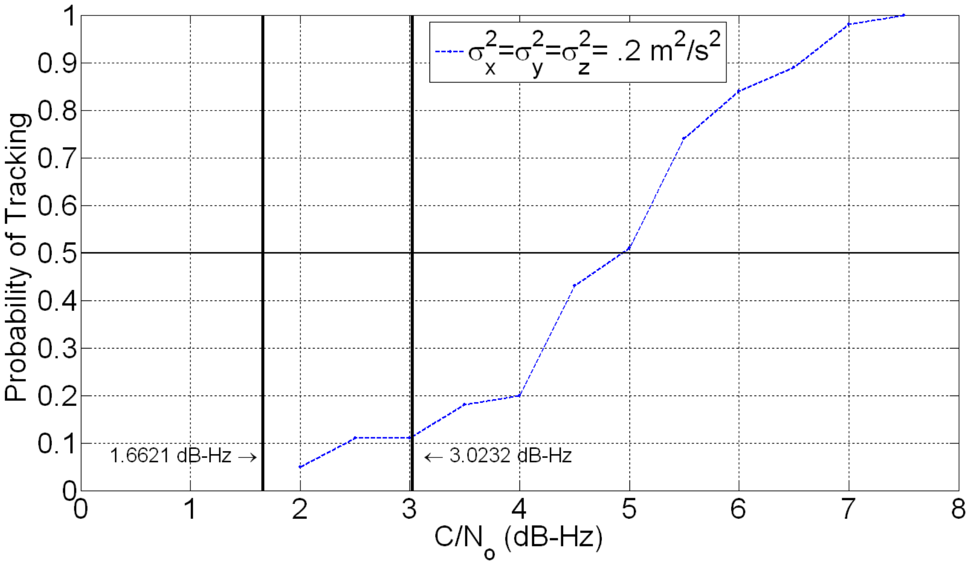

Figure 7.9: VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= .2 m^2/s^2

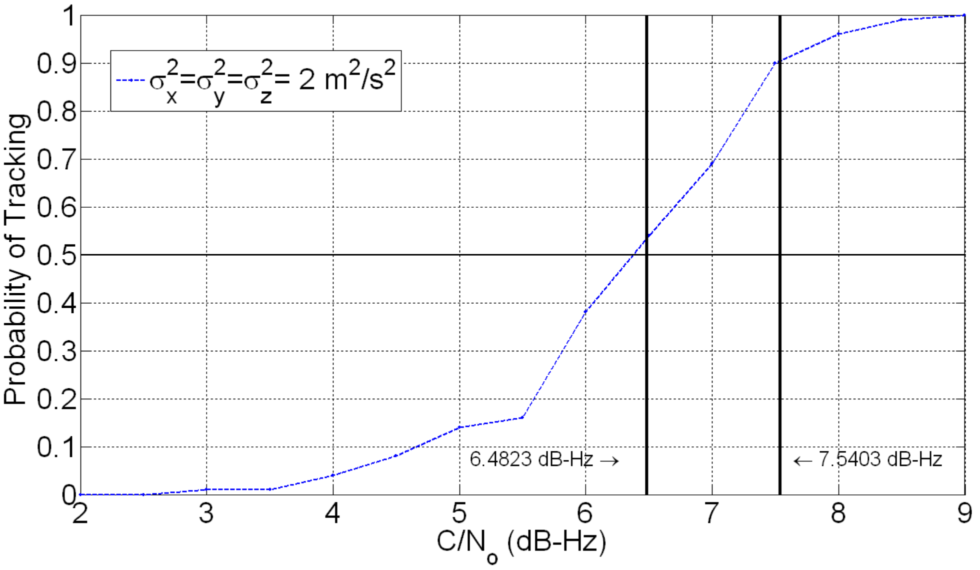

Figure 7.10: VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= 2 m^2/s^2

Figure 7.11 and Figure 7.12 shows the results of the Monte Carlo analysis for the 11 satellite geometry and the 10 and 50 m^2/s^2 process noise tunings, respectively. As the process noise tunings go up, the lowest C/N_o ratio that the VDFLL can reliably function at will also increase. When the minimum C/N_o ratio exceeds about 10 to 11 dB-Hz, the predictions from the rule of thumb analysis tend to be pessimistic by about 2 dB-Hz for the lower threshold, and 3.1 db-Hz for the upper threshold. This trend begins in Figure 7.11 and Figure 7.12. The trend continues for the 100 and 300 m^2/s^2 process noise tunings, shown in Figure 7.13 and Figure 7.14.

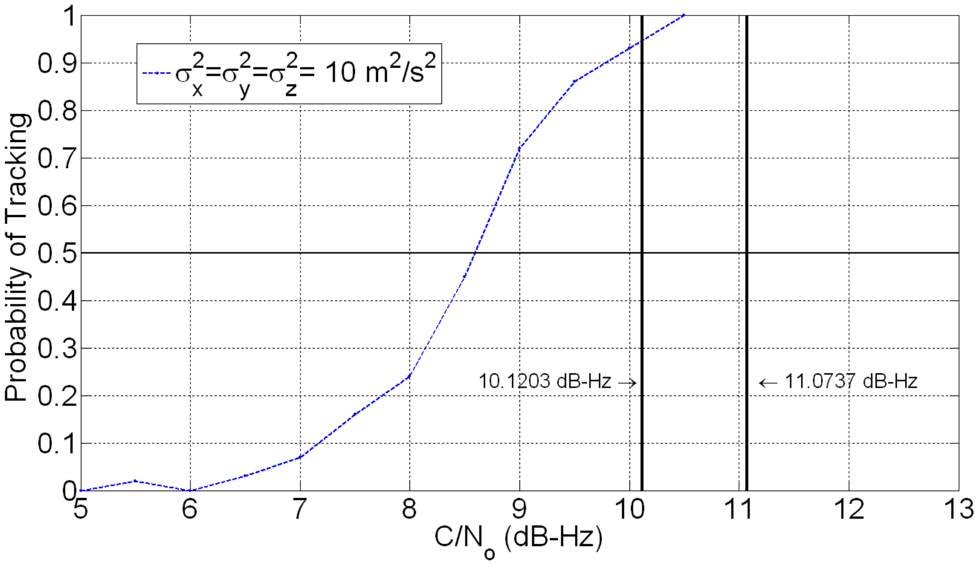

Figure 7.11: VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= 10 m^2/s^2

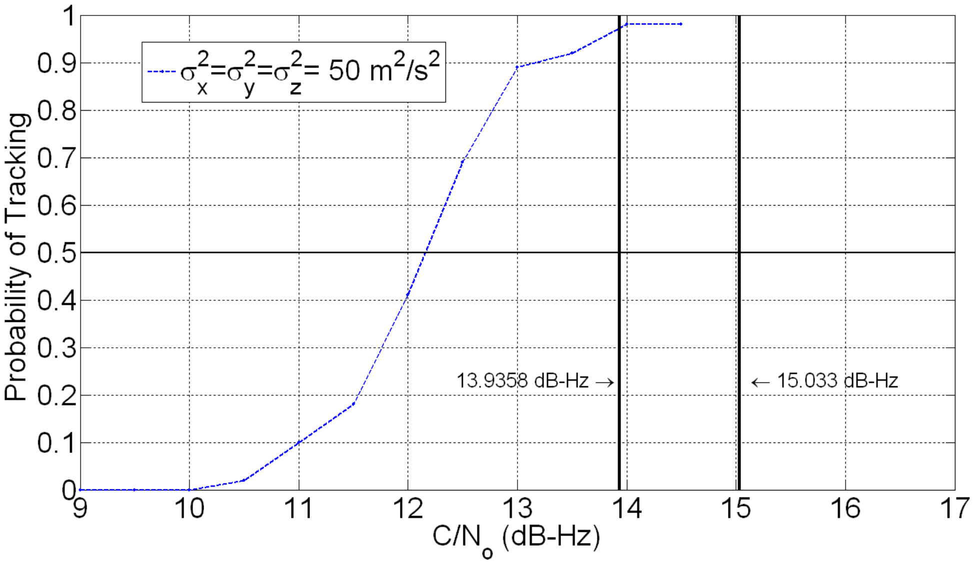

Figure 7.12: VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= 50 m^2/s^2

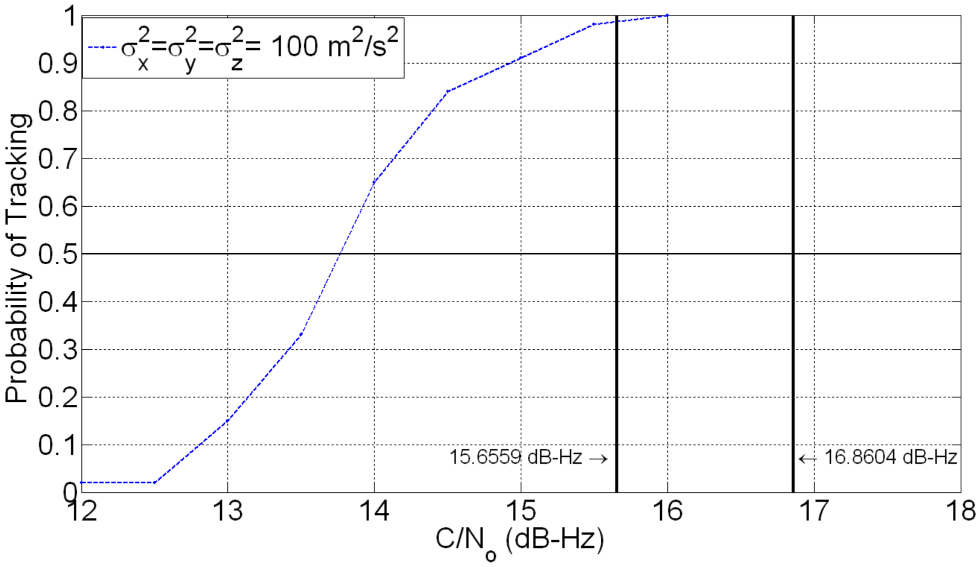

Figure 7.13: VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= 100 m^2/s^2

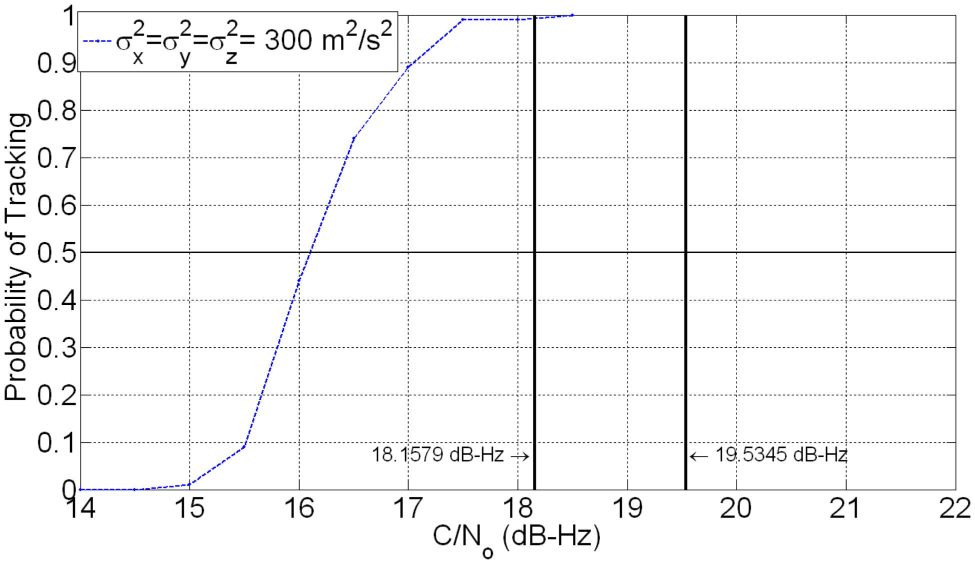

Figure 7.14: VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= 300 m^2/s^2

Examining figures 7.9 - 7.14 shows that lowering the process noise tuning increases the VDFLL's immunity to noise. When the process noise tuning is decreased from 100 to 10 m^2/s^2 the VDFLL gains about 5.3 dB in performance. However, when it is lowered from 2 to .2 m^2/s^2 only about 1.3 dB is gained. Figure 7.15 shows the results of Monte Carlo simulations when the process noise is further lowered to .02 m^2/s^2. Another 1.3 dB increase in performance is garnered by going from .2 to .02 m^2/s^2. Figures 7.9 through 7.15 demonstrate that below about 10 dB-Hz a point of diminishing returns is reached with respect to lowering the process noise tuning of the filter and increasing its noise immunity.

The Monte Carlo simulation results for the five and seven satellite constellations are similar to those for the eleven satellite scenario. When the minimum C/N_o ratio that the algorithms can function at exceeds around 10 or 11 dB-Hz, the rule of thumb analysis is pessimistic by about 2 dB-Hz for the lower threshold and 3.1 dB-Hz for the upper threshold.

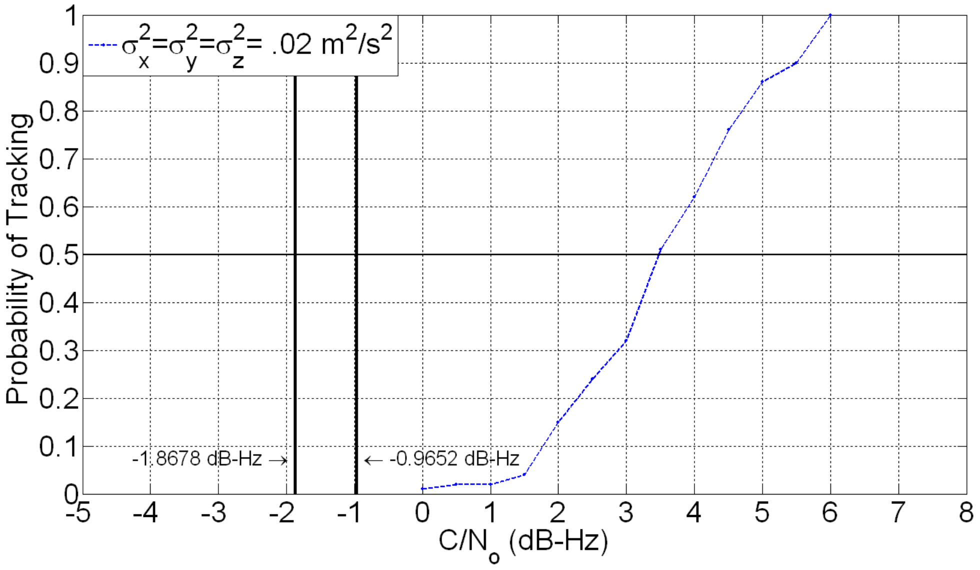

Figure 7.15: VDFLL Monte Carlo Simulation Results for 11 Satellites and σ^2= .02 m^2/s^2

Below about 10 dB-Hz, the disparity between the predictions from the rule of thumb analysis and the actual performance of the VDFLL starts to grow. As the number of satellites decreases, the performance of the VDFLL degrades. The plots of the Monte Carlo simulation results for the five and seven satellite constellations are included in Appendix C. The rule of thumb analysis and simulation results for all the different scenarios are summarized in Table 7.2.

The difference between the predicted thresholds and the results of the Monte Carlo simulations for the various scenarios are shown in Table 7.3. The differences between the lower predicted thresholds and the Monte Carlo results are comparable with the differences between the analysis and simulation results originally presented in [Ward, 1997]. The validity of the rule of thumb analysis begins to decline below about 10 dB-Hz. Below 10

Table 7.2: Summary of Monte Carlo Simulation Results

Number of Satellites	σ^2 (m^2/s^2)	Predicted Thresholds from Analysis Min. (dB-Hz)	Max. (dB-Hz)	Monte Carlo Results (dB-Hz)
Five	.2	3.9	4.5	6.1
	2	8.5	9	7.6
	10	11.8	12.9	10.2
	50	15.7	17.1	14
	100	17.5	19	15.6
	300	20.1	22	18.7
Seven	.2	2.5	4.3	5.5
	2	7.5	8.5	7.1
	10	11	12.1	9.5
	50	14.9	16.3	13.1
	100	16.7	18.2	15
	300	19.3	21	16.6
Eleven	.2	1.7	3	5
	2	6.5	7.5	6.4
	10	10.1	11	8.6
	50	13.9	15	12.1
	100	15.7	16.9	13.8
	300	18.2	19.5	16.1

Table 7.3: Summary of Monte Carlo Simulation Results

Number of Satellites	σ^2 (m^2/s^2)	Predicted Thresholds Minus Monte Carlo Result Min. (dB-Hz)	Max. (dB-Hz)
Five	.2	-2.2	-1.6
	2	.9	1.4
	10	1.6	2.7
	50	1.7	3.1
	100	1.9	3.4
	300	1.4	3.3
Seven	.2	-3	-1.2
	2	.4	1.4
	10	1.5	2.6
	50	1.8	3.2
	100	1.7	3.2
	300	2.7	4.4
Eleven	.2	-3.3	-2
	2	.1	1.1
	10	1.5	2.4
	50	1.8	2.9
	100	1.9	3.1
	300	2.1	3.4

dB-Hz, the actual performance of the VDFLL begins to degrade sharply with respect to its predicted performance.

7.3.3 Effect of C/N_0 Estimation

The C/N_0 ratio of the signals must be known for the vector tracking algorithms to function properly. The signal amplitudes are needed to use the pseudorange and pseudorange-rate discriminator functions in (7.11) and (7.13). In addition to the signal amplitudes, the noise power is needed by the filter to properly weight the measurements it is provided. In the previous section, the algorithms did not use any *a priori* information about the C/N_0 ratios of the signals. The signal amplitudes and noise power are estimated using the techniques described in Appendix B.

Researchers from QinetiQ have reported that the ability to estimate the C/N_0 ratio of the signals is possibly a major limiting factor of deeply integrated algorithms [Groves et al., 2007]. To test this hypothesis, the vector tracking algorithms used in the previous section are simulated with perfect knowledge of the signal amplitudes and noise power. The results of the Monte Carlo simulations are then compared for the same algorithms with and without knowledge of the C/N_0 ratios.

Figure 7.16 shows results of the Monte Carlo simulations for the eleven satellite constellation with the process noise set to 300 m^2/s^2. The two VDFLL algorithms are identical except one uses the true signal amplitudes and noise power and the other uses an estimate of the C/N_0 ratios from the correlator outputs. For the scenario in Figure 7.16 the *a priori* information of the C/N_0 ratios only increases the algorithms' performance by about .25 dB.

Figure 7.17 shows the results of the Monte Carlo simulations with a process noise tuning of 50 m^2/s^2. Similar to the 300 m^2/s^2 tuning scenario, perfect knowledge of the C/N_0 ratios only increases the algorithm's noise performance by about a .25 dB. Figures 7.18 through 7.21 display the simulation results with process noise tunings of 10, 2, .2, and .02 m^2/s^2, respectively. The minimum C/N_0 ratio that the VDFLL can function decreases with the decreasing process noise tunings. For the 10, 2, and .2 m^2/s^2 tunings, the performance of the VDFLL increases by about .6 to .8 dB when using the true C/N_0 ratios. The VDFLL simulation results for a process noise tuning of .02 m^2/s^2 are shown in Figure 7.21. Even at the lowest process noise tuning, the VDFLL only performs about .5 dB better when it has perfect knowledge of the C/N_0 ratio.

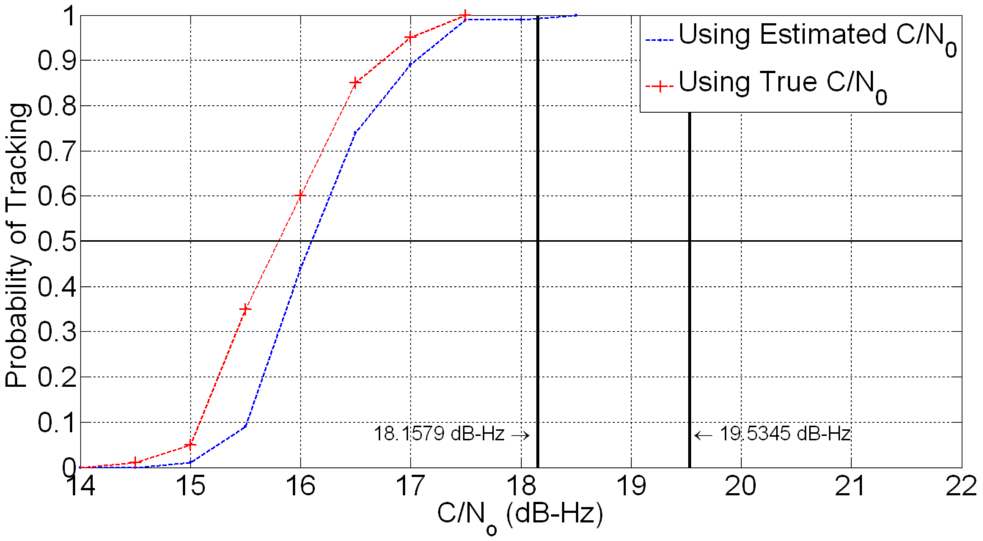

Figure 7.16: VDFLL Simulation Results for 11 Satellites and σ^2= 300 m^2/s^2 With and Without Prior Knowledge of C/N_o ratio

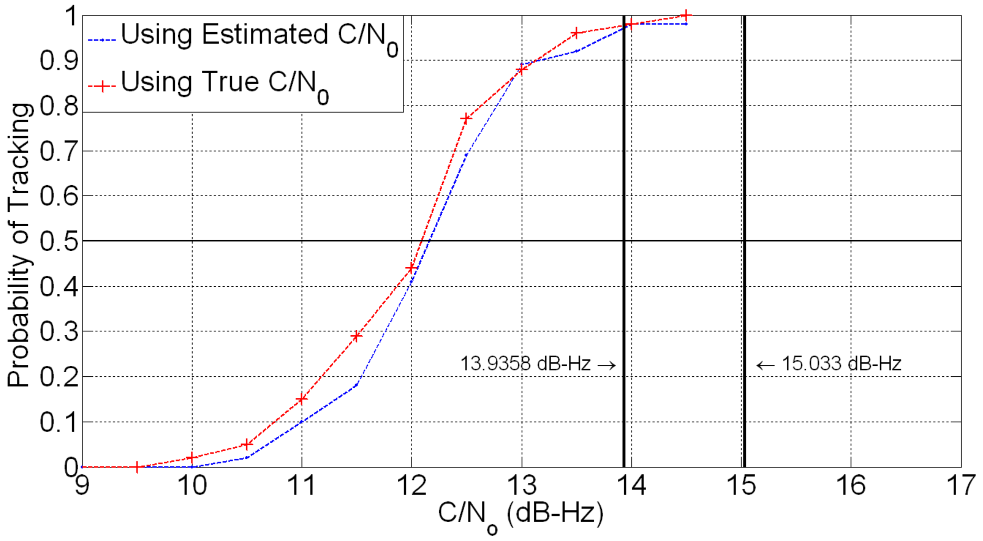

Figure 7.17: VDFLL Simulation Results for 11 Satellites and σ^2= 50 m^2/s^2 With and Without Prior Knowledge of C/N_o ratio

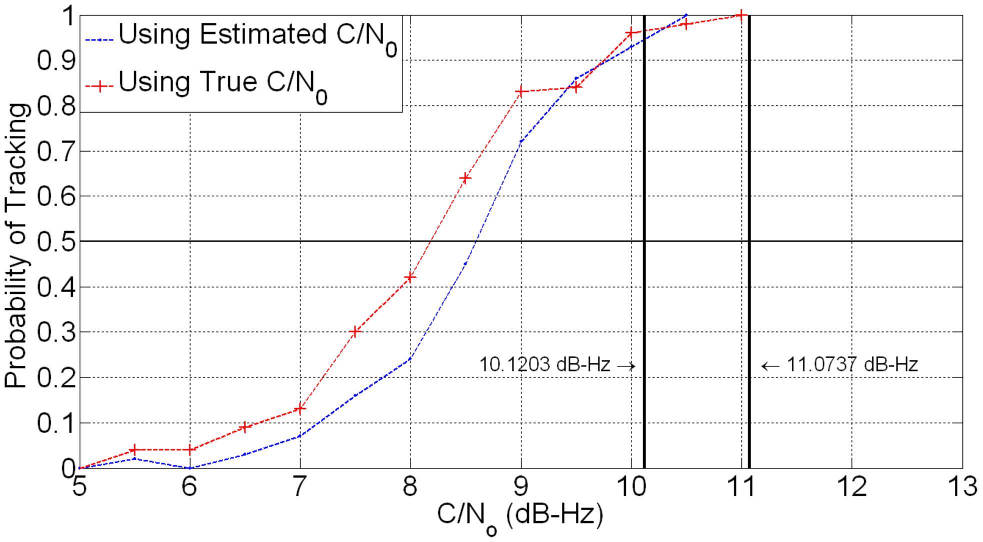

Figure 7.18: VDFLL Simulation Results for 11 Satellites and $\sigma^2 = 10\ m^2/s^2$ With and Without Prior Knowledge of C/N_o ratio

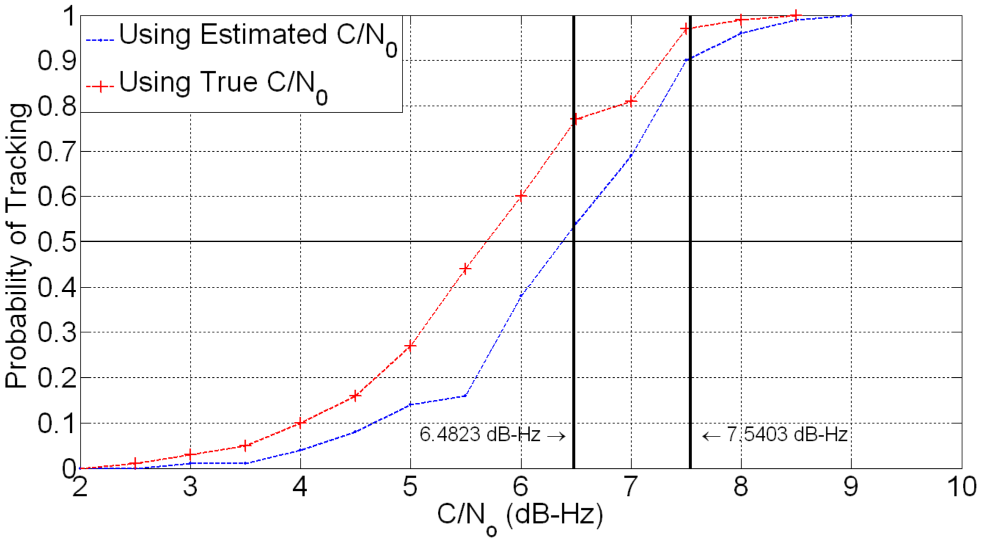

Figure 7.19: VDFLL Simulation Results for 11 Satellites and $\sigma^2 = 2\ m^2/s^2$ With and Without Prior Knowledge of C/N_o ratio

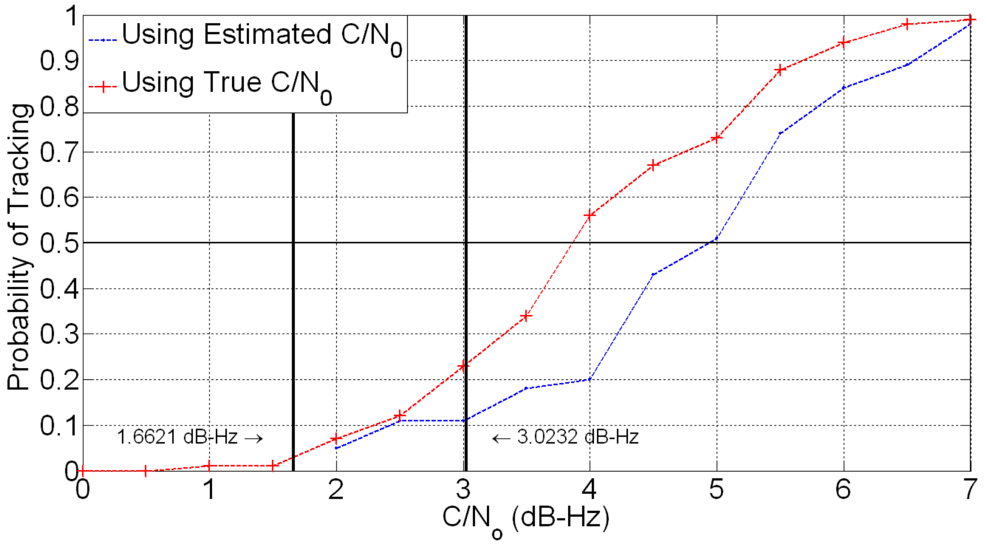

Figure 7.20: VDFLL Simulation Results for 11 Satellites and $\sigma^2 = .2\ m^2/s^2$ With and Without Prior Knowledge of C/N_o ratio

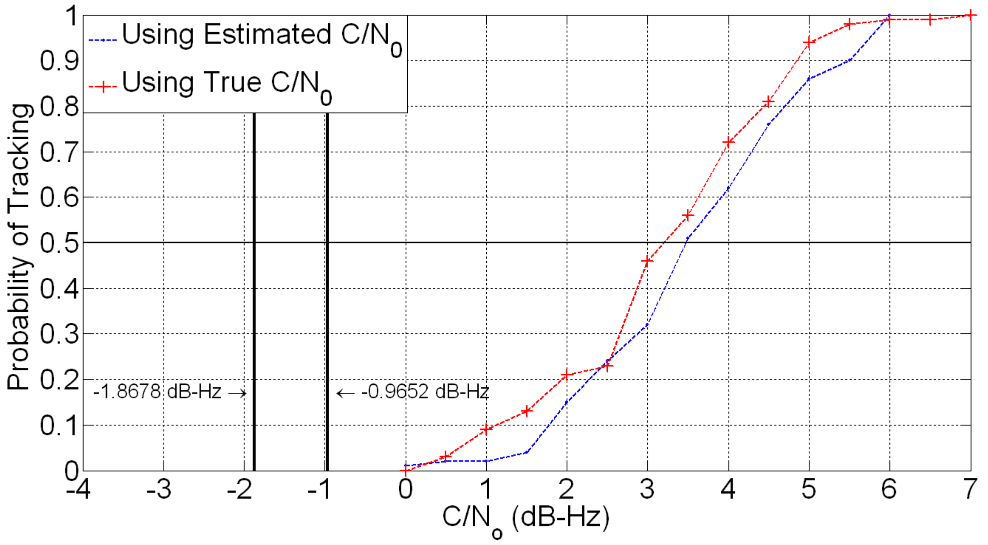

Figure 7.21: VDFLL Simulation Results for 11 Satellites and $\sigma^2 = .02\ m^2/s^2$ With and Without Prior Knowledge of C/N_o ratio

The Monte Carlo simulation results shown in Figures 7.16 through 7.21 indicate that accurately estimating the received signals' C/N_0 ratios does impact the performance of the VDFLL. However, the simulation results do not show a significant improvement when the algorithm knows the C/N_0 ratios perfectly. At C/N_0 ratios over 10 dB-Hz the VDFLL only performs about .25 dB better. At lower C/N_0 ratios, a maximum improvement of about .8 dB is experienced. This improvement is significantly less than that reported by the QinetiQ researchers [Groves et al., 2007], who reported an improvement in their DI algorithm's code tracking threshold of 3 dB when it used the true C/N_0 ratios instead of estimated C/N_0 ratios. Conversely, they reported a 2 dB decrease in the code tracking threshold when a sub par C/N_0 ratio estimation algorithm is used. Without having access to the C/N_0 ratio estimation techniques used by the QinetiQ researchers, it is not possible to explain the discrepancy between their results and the ones presented here.

7.3.4 Comparison of Scalar and Vector Tracking Algorithms

The previous Monte Carlo simulations all used the position-state formulation of the vector delay/frequency lock loop. In this section, the position and pseudorange-state formulations of the VDFLL are simulated in the same scenarios. These simulations are carried out for two reasons. First, the performance of the position-state and pseudorange-state vector tracking formulations can be shown to be equivalent. Mathematical proof of this claim is given in Appendix A. Therefore, the Monte Carlo simulations are carried out to show empirically that the formulations are equivalent. The second reason the simulations are performed is to ascertain the relative advantage offered by vector tracking over scalar tracking.

A valid comparison of scalar and vector tracking is allowed by using the pseudorange-state filter formulation.

Recall from Chapter 6 that the pseudorange-state filter can be operated with or without the channels of the receiver coupled together. With the receiver channels coupled together, the pseudorange-state filter is equivalent to the position-state filter. With the channels of the receiver uncoupled, the filter reverts back to scalar tracking. The coupling of the channels comes from the off-block diagonal elements in the discrete-time process noise covariance matrices. Setting the off-block diagonal elements to zero effectively decouples the channels. The signals from the various satellites are then tracked independent of one another. However, the decoupled filter still uses the same tunings as the coupled filter. The performance of the filter in the coupled and decoupled configurations can therefore be studied to ascertain the advantage provided solely by the coupling of the channels.

Three VDFLL algorithms are tested in each Monte Carlo simulation. The first is the position-state formulation. The second is the pseudorange-state formulation with coupled channels. The third algorithm is the pseudorange-state filter without coupled channels. The channels are decoupled by setting the off-block diagonal elements of the process noise covariance matrix to zero. The same process noise tunings are used for all three filters. The three satellite geometries from the previous section are used in the Monte Carlo simulations. A temperature compensated crystal oscillator with the power spectral density coefficients shown in (7.9) is used again.

Second-order VDFLL algorithms are tested in the Monte Carlo simulations. Three levels of process noise are used by the filters for each satellite constellation. The process noise terms σ_x^2, σ_y^2, and σ_z^2 are set equal to each other. The magnitudes of the process noise are 10, 50, and 100 m^2/s^2. The VDFLL algorithms are simulated over a range of C/N_o ratios in increments of .5 dB-Hz for each satellite geometry and process noise tuning. At each C/N_o ratio, one hundred Monte Carlo simulations are carried out.

The filters use the steady state Kalman gain corresponding to each scenario. The C/N_0 ratio is not estimated by the algorithms. In light of the previous section, the difference in the performance of the algorithms when having to estimate the C/N_0 ratio should not be significant. The C/N_0 ratio is assumed to be known and the steady state Kalman gains are used in order to reduce the computation times required to run the Monte Carlo simulations.

It is important to note that the Kalman filter equations for the coupled channel pseudorange-state filter have very poor numerical properties. To determine the steady state Kalman gain for the pseudorange-state filter, the steady state Kalman gain for the position-state filter is first determined. Then, relations from Appendix A are used to convert the steady state Kalman gain for the position-state filter to the steady state Kalman gain for the pseudorange-state filter. These extra steps are only necessary when the channels of the receiver are coupled together.

Figure 7.22 shows the results of a thermal noise analysis of the coupled and decoupled pseudorange-state filter for the eleven satellite scenario. A process noise tuning of 10 m^2/s^2 is used for the analysis. The same two part analysis from section 7.2.2 is carried out. The steady state Kalman gain for the specific clock and dynamic process noise levels is computed. The thermal noise jitter for every channel is then calculated with only measurement noise and clock noise. For the decoupled pseudorange-state filter, the thermal noise jitter is the same for all the channels. For the coupled channel filters, the channels of the receiver exhibit different thermal noise jitters. From Figure 7.22, the analysis shows that the channels have a reduction in frequency jitter ranging from 2.5 to 3.5 dB when coupled together.

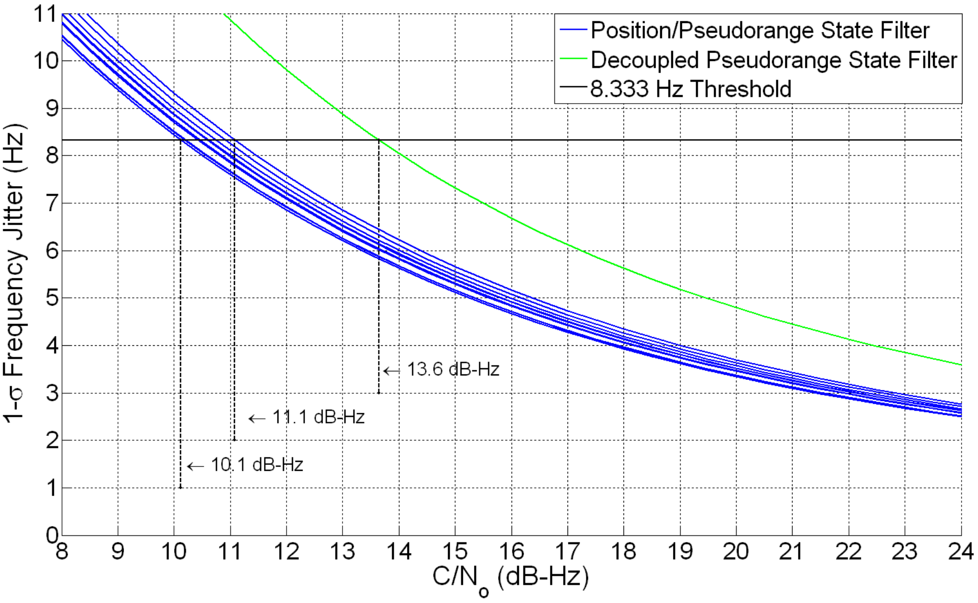

Figure 7.22: Thermal Noise Analysis Results for 11 Satellites and σ^2= 10 m^2/s^2

Figure 7.23 shows the results of the Monte Carlo simulations for the eleven satellite and 10 m^2/s^2 process noise tuning scenario. The criteria for loss of lock is whenever a pseudorange error exceeds half a chip of the Gold code. The results for the two different vector tracking algorithms essentially coincide. At about 8.3 dB-Hz, the probability of the vector tracking algorithms functioning is 50 percent. In comparison, the decoupled pseudorange-state filter has a 50 percent probability of tracking at around 13.35 dB-Hz. Therefore, for the specific scenario being simulated vector tracking yields an increase in performance of about 5.05 dB over scalar tracking.

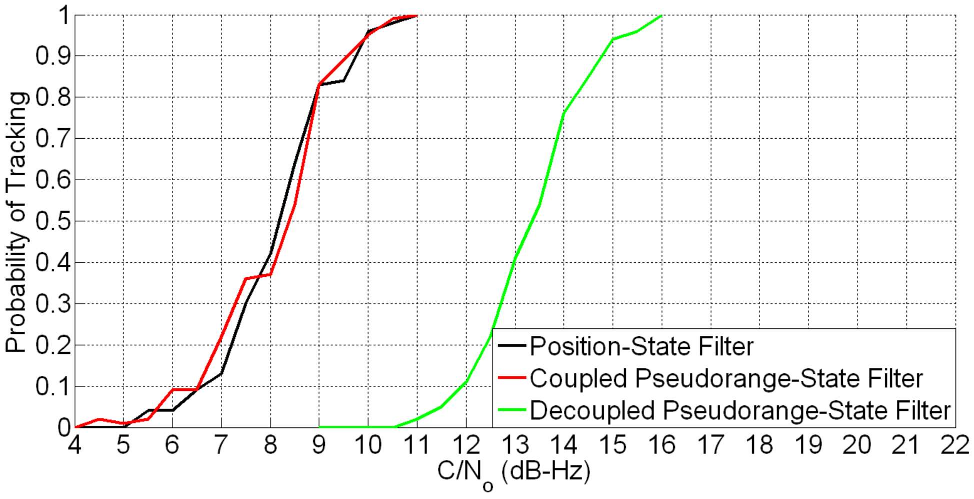

Figure 7.23: Monte Carlo Simulation Results for 11 Satellites and σ^2= 10 m^2/s^2

Figures 7.24 and 7.25 show the results of the thermal noise analysis and Monte Carlo simulations for the eleven satellite and 50 m^2/s^2 process noise tuning scenario, respectively. The analysis shows that the channels have a reduction in frequency jitter ranging from 3.1 to 4.2 dB when coupled together. At about 12 dB-Hz, the probability of the vector tracking algorithms functioning is 50 percent. In comparison, the decoupled pseudorange-state filter has a 50 percent probability of tracking at around 17.9 dB-Hz. Therefore, for the specific scenario being simulated vector tracking yields an increase in performance of about 5.9 dB over scalar tracking.

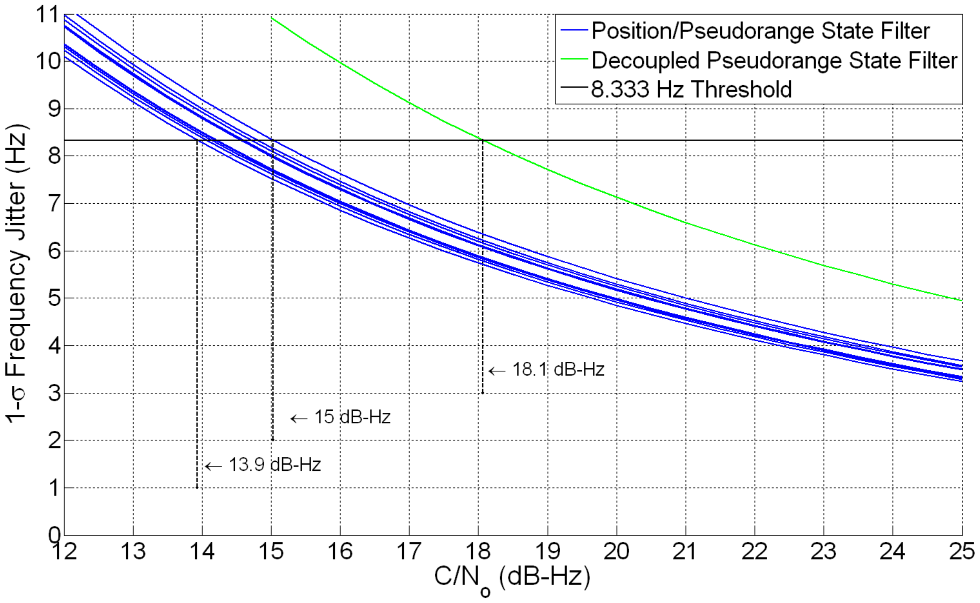

Figure 7.24: Thermal Noise Analysis Results for 11 Satellites and $\sigma^2 = 50\ m^2/s^2$

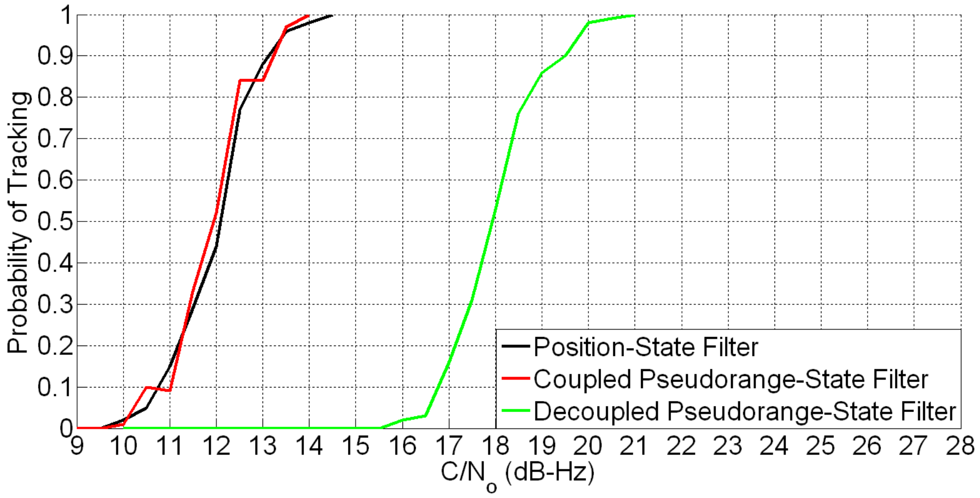

Figure 7.25: Monte Carlo Simulation Results for 11 Satellites and $\sigma^2 = 50\ m^2/s^2$

Figures 7.26 and 7.27 show the results of the thermal noise analysis and Monte Carlo simulations for the eleven satellite and 100 m^2/s^2 process noise tuning scenario, respectively. The analysis shows that the channels have a reduction in frequency jitter ranging from 3.3 to 4.5 dB when coupled together. At about 13.7 dB-Hz, the probability of the vector tracking algorithms functioning is 50 percent. In comparison, the decoupled pseudorange-state filter has a 50 percent probability of tracking at around 19.9 dB-Hz. Therefore, for the specific scenario being simulated vector tracking yields an increase in performance of about 6.2 dB over scalar tracking.

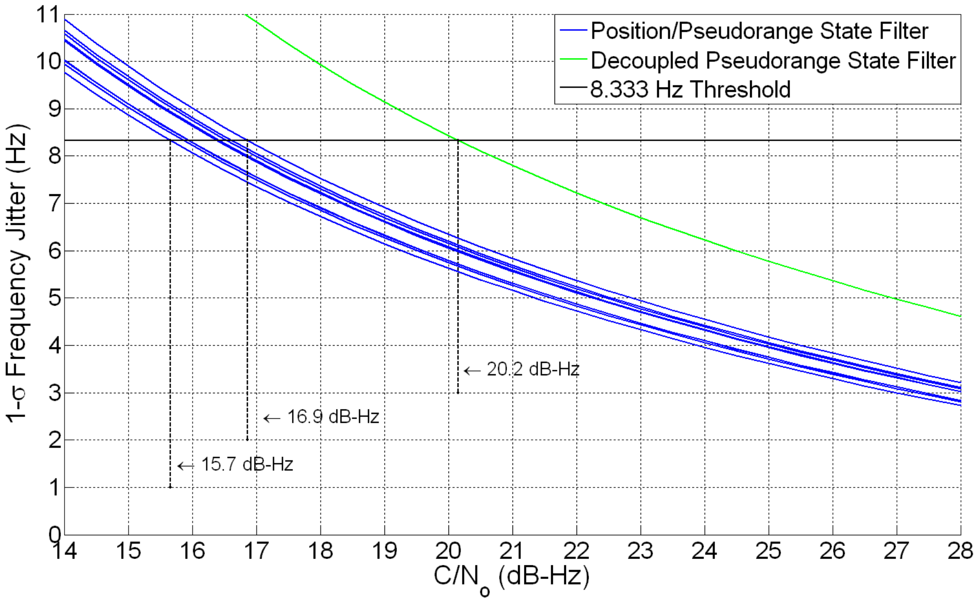

Figure 7.26: Thermal Noise Analysis Results for 11 Satellites and σ^2= 100 m^2/s^2

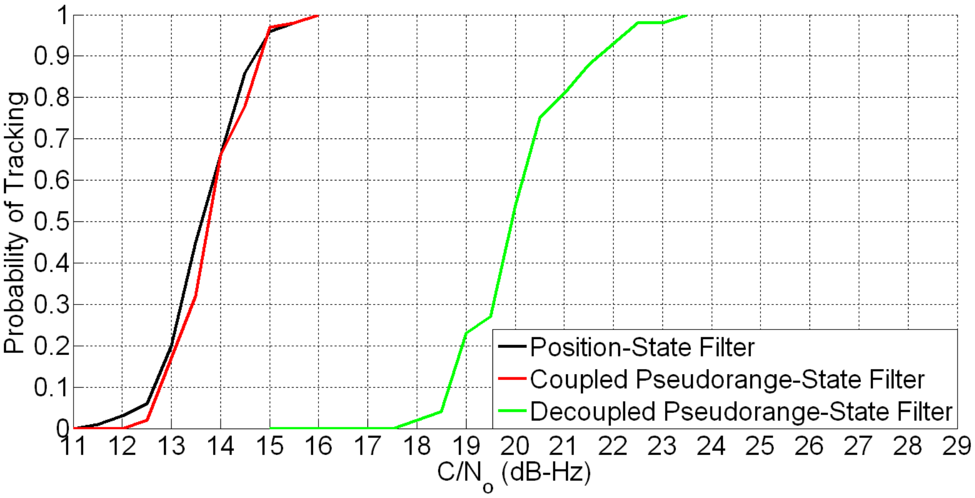

Figure 7.27: Monte Carlo Simulation Results for 11 Satellites and σ^2= 100 m^2/s^2

The results of the thermal noise analysis for all the satellite constellations and process noise tunings are summarized in Table 7.4. The plots of the analysis results for the five and seven satellite constellations are included in Appendix C. The right most column shows the thermal noise jitter for the decoupled pseudorange-state filter. The thermal noise jitter for a given process noise tuning does not change with the number of satellites. This is due to the fact the channels of the receiver are not coupled. The addition of satellites does not improve the performance of the filter. In contrast, the thermal noise jitter for the coupled filters decreases as more satellites are used. Consequently, the greatest reduction in thermal noise jitter comes for the eleven satellite constellation.

Table 7.5 summarizes the results of the Monte Carlo simulations for the various satellite constellations and process noise tunings. The plots of the Monte Carlo results for the five and seven satellite constellations are included in Appendix C. The approximate C/N_0 ratio at which the algorithms displayed a 50 percent chance of tracking are given for the

Table 7.4: Thermal Noise Analysis Results

Number of Satellites	σ^2 (m^2/s^2)	Position/Pseudorange-State Filter Maximum Jitter	Position/Pseudorange-State Filter Minimum Jitter	Decoupled Pseudorange-State Filter Jitter for All Channels
Eleven	10	10.1	11.1	13.6
	50	13.9	15	18.1
	100	15.7	16.9	20.2
Seven	10	11.1	12.1	13.6
	50	14.9	16.3	18.1
	100	16.7	18.2	20.2
Five	10	11.8	12.9	13.6
	50	15.7	17.1	18.1
	100	17.5	19.1	20.2

various scenarios. For a given process noise tuning, the performance of the decoupled filter is essentially the same for the different numbers of available satellites. The greatest disparities in performance occur for the eleven satellite scenario. The coupled filters exhibit an increase in performance of about 5 to 6.2 dB. The increase in performance for the other two constellations is more modest. For the seven satellite scenario, the increase varies from approximately 3.65 to 4.8 dB. The coupled filters have the least improvement for the five satellite constellation. This makes intuitive sense because fewer satellites results in less of an averaging effect. They show an improvement of around 2.4 to 3.3 dB.

Examining the results in Table 7.5, the 50 percent tracking threshold for the decoupled pseudorange-state filter increases as the number of satellites increases. This is counterintuitive because the channels of the filter are not coupled together. Changing the number of satellites should not affect the ability of the decoupled filter to operate. The reason the tracking threshold increases with the number of satellites is due to the way in which the performance of the algorithms is assessed. In order for the algorithms to pass a given Monte

Table 7.5: Summary of Monte Carlo Simulation Results

Number of Satellites	σ^2 (m^2/s^2)	C/N_0 Ratio Corresponding to 50 Percent Probability of Tracking Position/Pseudorange-State Filter	Decoupled Pseudorange-State Filter
Five	10	10.1	12.5
	50	14.1	17.1
	100	15.9	19.2
Seven	10	9.2	12.85
	50	13.1	17.45
	100	14.9	19.7
Eleven	10	8.3	13.35
	50	12	17.9
	100	13.7	19.9

Carlo run, the pseudorange errors in all the channels must not exceed half a chip of the Gold code (approximately 147 meters) for the entire simulation duration. This criteria becomes more restrictive as the number of satellites increases for the decoupled filter. Since the channels of the receiver are not coupled, no advantage is offered in going from five to eleven satellites. However, the number of channels that must successfully operate for each trial increases. This is why the tracking threshold increases for the decoupled pseudorange-state filter as the number of satellites increases.

It is important to note that the same criteria is used for all three algorithms. The criteria for success becomes more restrictive with the inclusion of more satellites for the coupled channel filters as well. However, the percent of trials passed for a given C/N_0 ratio still increases for the coupled channel filters when more satellites are available.

In the decoupled pseudorange-state filter, all the channels of the receiver operate independently. A fault in one channel will not effect the other channels of the receiver.

Consequently, it can be argued that requiring the pseudorange errors in all the channels not to exceed the half chip threshold is overly restrictive for the decoupled filter. In order to make a more fair comparison, the Monte Carlo simulations were performed again with just a single channel of the pseudorange-state filter. This is done to determine how a single channel will perform without the added constraint that the other channels perform throughout the simulation as well. Figure 7.28 shows the results of the Monte Carlo simulations with the 10, 50, and 100 m^2/s^2 process noise tunings.

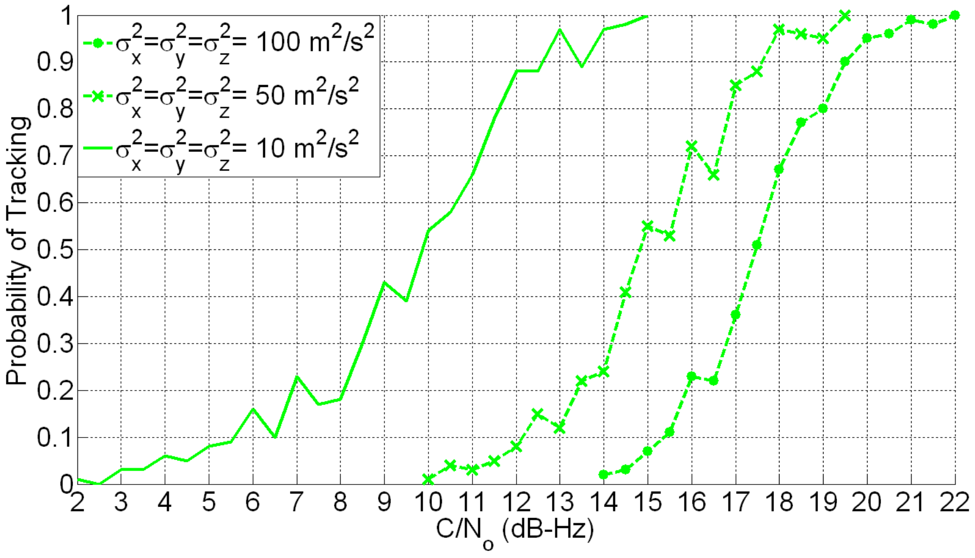

Figure 7.28: Monte Carlo Simulation Results for a Single Channel of the Decoupled Filter

In Figure 7.28, the C/N_0 ratios corresponding to a 50 percent probability of tracking for the 10, 50, and 100 m^2/s^2 process noise tunings are approximately 10, 15, and 17.5 dB-Hz, respectively. These C/N_0 ratios are about 1.7 to 2.5 dB-Hz lower than the corresponding thresholds for the decoupled pseudorange-state filter in the five satellite constellation. For the five satellite scenario, the vector tracking algorithms only perform about .1 to 1.6 dB-Hz better than the single channel filter. For the seven satellite constellation, the vector tracking algorithms perform about .8 to 2.6 dB-Hz better than the single channel filter. The vector tracking algorithms only perform about 1.7 to 3.8 dB-Hz better than the single channel filter for the eleven satellite scenario. Therefore, it is seen that the exact manner in which scalar and vector tracking are compared has a significant effect on the results of the comparison.

7.4 High Dynamics Simulation Results

In this section, simulation results are presented that demonstrate the ability of the Vector Delay/Frequency Lock Loop to operate in high dynamic environments and at low C/N_o ratios. The analysis tools presented earlier in the chapter are applied to study the VDFLL's performance as a function of receiver dynamics, C/N_0 ratio, and satellite constellation. Instead of using software simulations inside MATLAB, a hardware simulator is used to generate RF data. The RF data is recorded and then post processed in MATLAB. This data has also been used by the author in [Lashley et al., 2008] [Lashley et al., 2009]. An FLL is used to process the same data as a comparison for the VDFLL. As will be shown, the results from using the RF simulator data compare favorably with the results predicted by using the previous analysis tools and the VDFLL performs significantly better than the scalar FLL.

7.4.1 Simulation Setup

An L-3 Interstate Electronics GPS constellation simulator is used to generate high fidelity Radio Frequency (RF) data for a high dynamic flight profile. The RF data is collected with a NordNav RF front-end. The NordNav front-end downconverts RF signals to an Intermediate Frequency (IF) of 4.1304 MHz. The front-end samples the IF signal at 16.3676 MHz and saves the data for post-processing. The NordNav front-end is capable of recording 1, 2 and 4 bit samples. For the results presented here, 2 bit samples are used. The front-end only captures the L_1 GPS signal. The VDFLL algorithm only uses the civilian portion of the recorded L_1 signal. The simulator provides the receiver's true position, velocity, acceleration, attitude, etc. throughout the flight profile.

The flight profile used in the simulation starts at the coordinates (6378.2, 0, 0) km in the ECEF coordinate frame. The ECEF x-axis coordinate of the user remains essentially constant throughout the flight. Significant dynamics only occur in the ECEF y-axis and z-axis. Figure 7.29 shows the y-axis and z-axis coordinates during the flight profile. The simulation includes three coordinated turns. The first is a 2 G turn, followed by a 4 G turn. The last maneuver is an 8 G coordinated turn. The C/N_o starts at a nominal value of approximately 43 dB-Hz. After an initial stationary period (about 260 seconds), the user starts moving in the z-axis direction. After the user's velocity in the z-axis levels out, the C/N_o ratio is lowered and then held constant for the remaining duration of the simulation. The satellite constellation used in the scenario is identical to the eleven satellite scenario used in the previous sections. Subsets of five and seven satellites are also used which are identical to the five and seven satellite constellations used in the previous section.

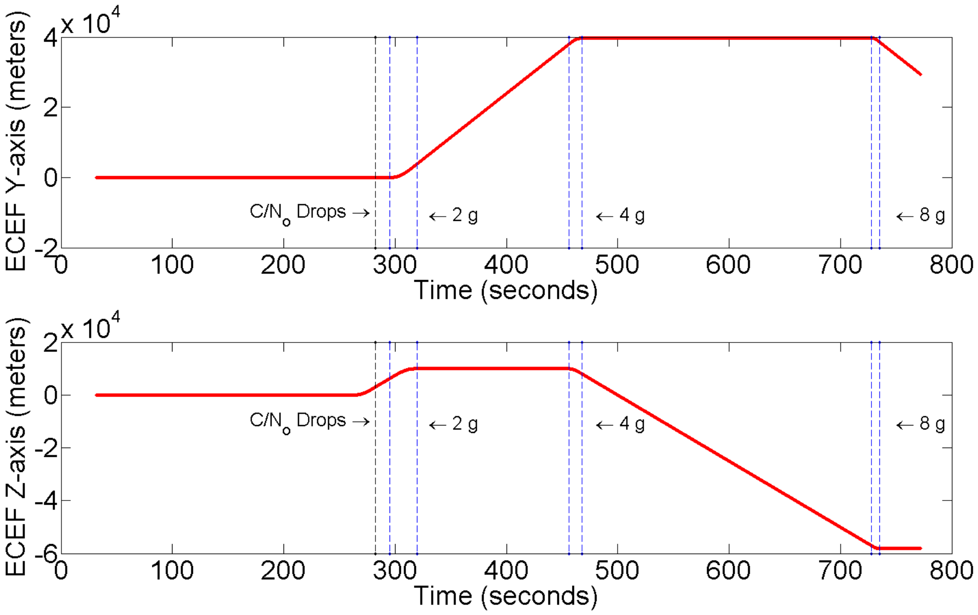

Figure 7.29: ECEF Y and Z Coordinates During Simulation

7.4.2 Simulation Results

Data was collected with the NordNav receiver at incrementally lower C/N_o ratios. The lowest C/N_o that the VDFLL algorithm operated for the whole flight profile is 19 dB-Hz. At 19 dB-Hz, the algorithm can function with five, seven, or eleven satellites through all three coordinated turns. The analysis tools presented earlier were applied to the VDFLL during the 4 G and 8 G turns when using five satellites. The C/N_o ratio and satellite geometry data were used to calculate the steady state frequency jitter due to thermal noise. A value of 125 m^2/s^4 was used for the process noise covariance terms σ_x^2, σ_y^2, and σ_z^2. The dynamic stress error was then calculated using the reference trajectory. Figure 7.30 shows the cumulative effects of the 3-σ thermal noise jitter and dynamic stress error for the VDFLL during the 4 G and 8 G turns at 19 dB-Hz. The five lines on the graphs correspond to the frequency jitter and dynamic stress error for each individual satellite being tracked. During the 4 G turn, the 25 Hz threshold is not exceeded by any of the satellites. However, during the 8 G turn the threshold is slightly exceeded by two satellites for about one second. The 25 Hz threshold tends to be conservative in this case.

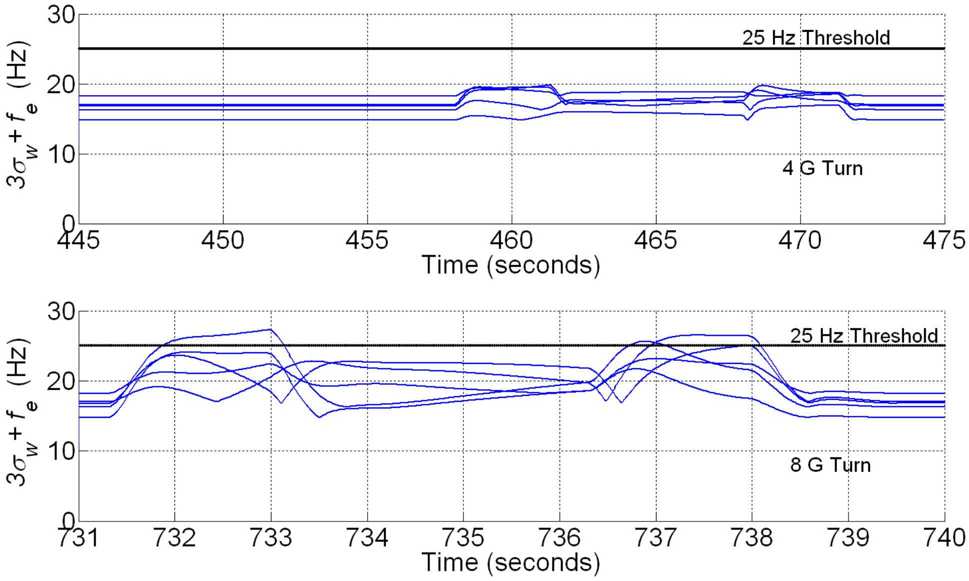

Figure 7.30: VDFLL Thermal Noise Jitter and Dynamic Stress Error at 19 dB-Hz Using 5 Satellites

Using a C/N_0 ratio of 16 dB-Hz data, the VDFLL algorithm can successfully operate through the first two turns with seven and eleven satellites being tracked. The algorithm however cannot maintain lock through the 8 G maneuver at this C/N_0 ratio. The analysis tools presented earlier were applied to the VDFLL during the 4 G and 8 G turns when using seven satellites. Figure 7.31 shows the cumulative effects of the 3-σ thermal noise jitter and dynamic stress error for the VDFLL during the 4 G and 8 G turns at 16 dB-Hz with seven satellites. As the figures show, the 25 Hz threshold is exceeded briefly during the 4 G turn by three of the satellites. However, during the 8 G turn the threshold is exceeded by the majority of the satellites for prolonged periods of time.

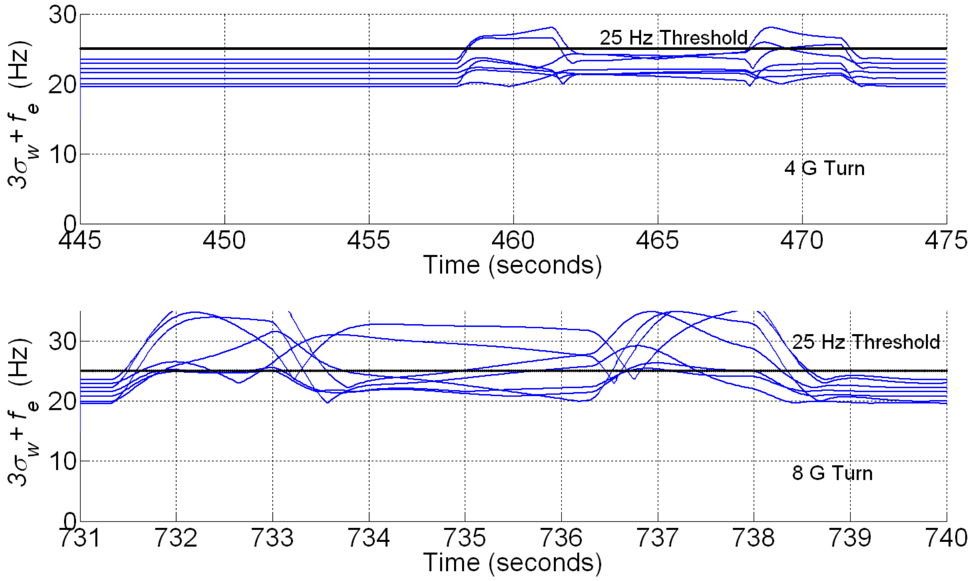

Figure 7.31: VDFLL Thermal Noise Jitter and Dynamic Stress Error at 16 dB-Hz Using 7 Satellites

The velocity errors in the EKF's estimates in the y- and z-axes are shown in Figure 7.32. The VDFLL's velocity estimates are very noisy and the errors diverge rapidly during the last 8 G turn.

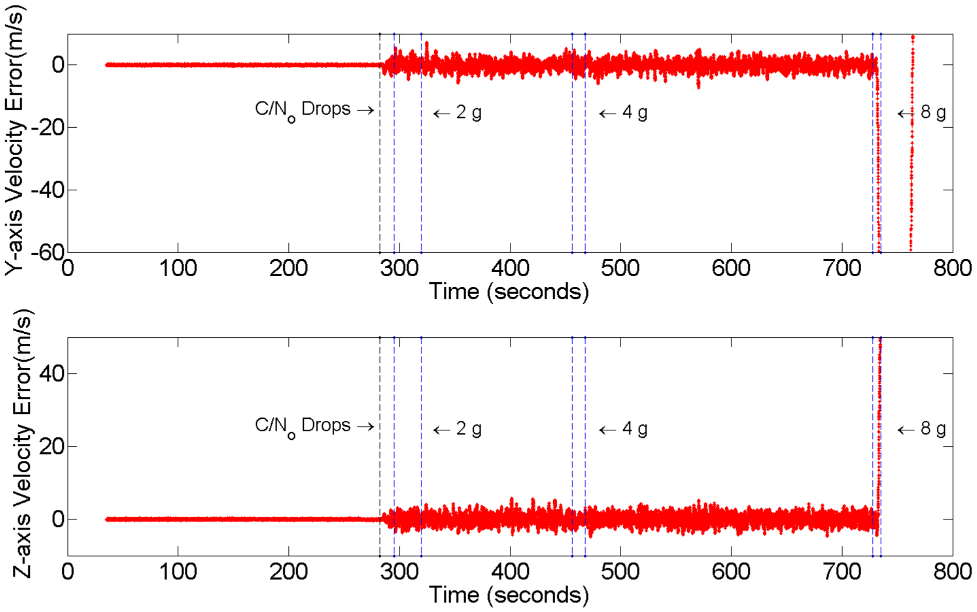

Figure 7.32: VDFLL Velocity Estimate Errors Using 7 Satellites at 16 dB-Hz

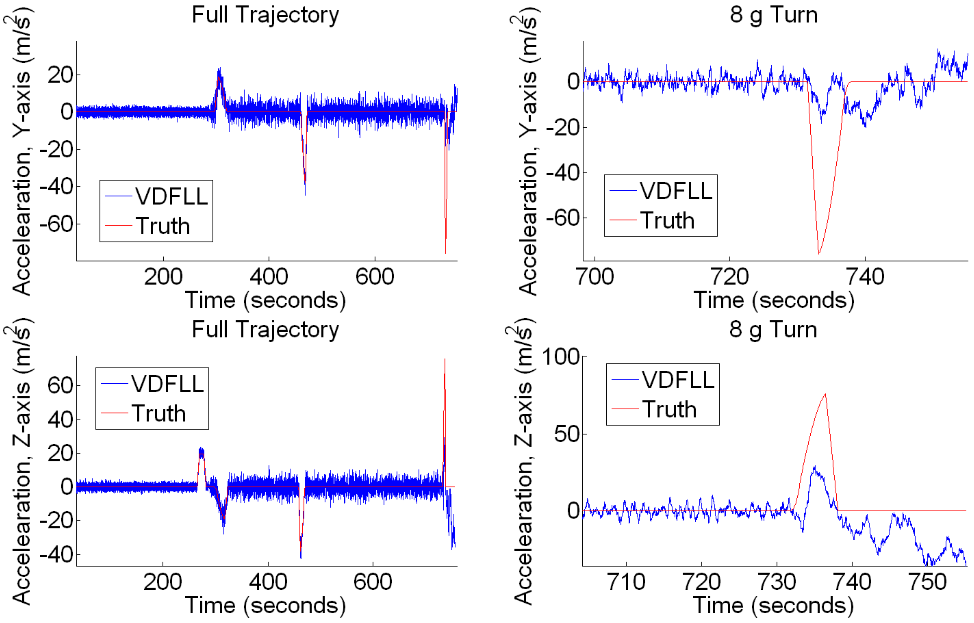

Figure 7.33: VDFLL Y and Z-Axis Acceleration Data Using 11 Satellites at 16 dB-Hz

Figure 7.33 shows the true y- and z-axis acceleration of the receiver during the simulation and the VDFLL's estimates. The VDFLL adequately tracks the changes in acceleration during the 2 G and 4 G turns. However, the combination of the low C/N_o ratio and the magnitude of the change in acceleration during the 8 G maneuver causes the VDFLL to lose track. As a comparison to the VDFLL, a 2 Hz FLL is also used to track the Doppler shift on the carrier signal during the flight profile. As before, the FLL uses the loop filter parameters given in [Ward, 1996b]. Figure 7.34 shows the FLL's estimates of the carrier Doppler frequency at 19 dB-Hz. The FLL's estimates diverge rapidly after the C/N_o ratio is lowered. Prior to the C/N_o being lowered, the user's velocity in all directions has settled out. The line of sight dynamics are essentially zero from the period when the C/N_o drops to when the FLL's estimates have diverged. The divergence of the FLL is caused by thermal noise and not dynamic stress. This is congruent with the analysis results shown in Fig. 7.1.

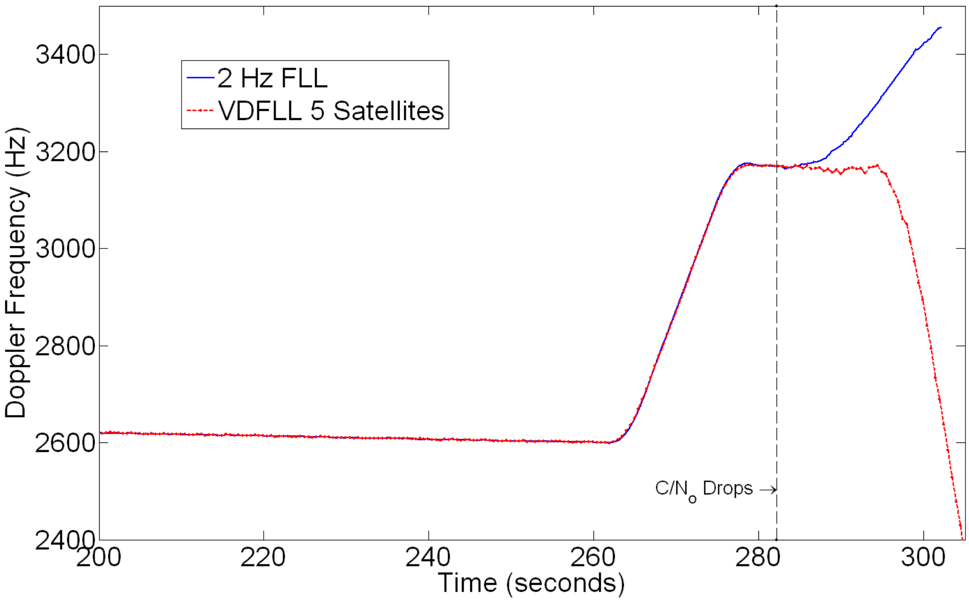

Figure 7.34: 2 Hz FLL and VDFLL Performance at 19 dB-Hz

The results of processing the RF simulator data show that the VDFLL performs significantly better than the scalar FLL. The ability of the VDFLL to maintain lock through the 2, 4, and 8 G turns also matches its predicted performance from the analysis tools.

7.5 Conclusion

Several analysis techniques for assessing the performance of vector tracking algorithms are introduced in this chapter. The various techniques are adapted from rule of thumb tracking thresholds originally used for studying scalar tracking loops. The analysis techniques are used to determine the lowest C/N_0 ratio at which the vector tracking loops will operate for a specific scenario. The techniques can also be used to determine the maximum allowable receiver dynamics for different situations.

The analysis tools used in this chapter implicitly assume that the rule of thumb tracking thresholds for scalar loops apply to vector tracking loops. To test this assumption, Monte Carlo simulations of different vector tracking loop configurations are carried out. For each configuration, the minimum C/N_0 ratio that the vector tracking algorithms can operate at is calculated by using the analysis techniques. The results of the Monte Carlo simulations are then compared to the analysis results. The results of the Monte Carlo simulations matched the predictions of the analysis to within about 2 dB for the majority of the scenarios. This level of accuracy for the vector tracking analysis techniques is comparable to the accuracy of the rule of thumb thresholds when applied to scalar tracking loops. The effect of requiring the algorithms to estimate the received signal amplitudes and noise power (as opposed to being supplied with the actual values) is also studied through Monte Carlo simulations. A maximum improvement of only about .8 dB is observed when the algorithms are supplied the true signal amplitudes and noise power.

The benefits of vector over scalar tracking architectures are also investigated using the analysis techniques and Monte Carlo simulations. The pseudorange-state filter is used in its coupled and decoupled channel modes to allow a valid comparison between the two architectures. For an eleven satellite constellation, vector tracking displays an improvement of 6.2 to 3.8 dB, depending upon how the comparison is made.

The analysis techniques are also tested using RF data recorded from a high fidelity GPS simulator. The recorded data is taken from a high dynamic flight profile containing 2, 4, and 8 G turns. The maximum receiver dynamics predicted by the analysis techniques are compared to the actual performance of the vector tracking algorithms throughout the flight profile. The maximum allowable dynamics predicted by the analysis tend to be conservative, but overall they match the results of the simulations reasonably well.

CHAPTER 8

ANALYSIS OF DIFFERENT VECTOR TRACKING ARCHITECTURES

8.1 Introduction

This chapter investigates three different vector tracking architectures. The goal of the chapter is to determine the loss in performance caused by using a federated filtering architecture instead of a single, central Kalman filter. The difference in the performance of the algorithms is studied through covariance analysis and Monte Carlo simulations. The performance of the different architectures is studied over a range of C/N_0 ratios, process noise intensities, and for two different satellite constellations. The covariance analysis is used to compare the variance of the estimates produced by each of the different algorithms under the same conditions. The results of the analysis allow the difference in performance of the algorithms to be expressed in decibels. Monte Carlo simulations are then used to verify the covariance analysis.

8.2 Deep Integration Filter Architectures

The majority of Deeply Integrated (DI) algorithms can be separated into two categories. The first category of algorithms rely on a single, centralized Kalman filter . The central filter processes the received GPS signals and Inertial Measurement Unit (IMU) measurements. The filter processes GPS measurements as they become available from the integrate and dump functions and it directly controls the Numerically Controlled Oscillators (NCO's). The second category of DI algorithms rely on a federated filtering architecture. In these algorithms, a large master filter processes the IMU measurements and also controls the NCO's for the GPS replica signals. Unlike the centralized filtering approach however, the master filter does not process the raw measurements produced by the integrate and dump operations. Instead, a set of smaller intermediate Kalman filters process the raw GPS measurements. In [Abbott and Lillo, 2003], these are referred to as prefilters. Each satellite has a prefilter dedicated to processing its signals. The prefilters take the raw GPS measurements and estimate the pseudorange and pseudorange-rate for their specific satellite. The state vectors of the prefilters are then periodically used by the master filter as measurements. The covariance matrices of the prefilters are also used by the master filter during its measurement update.

There are two main issues involved when using the federated filtering architecture. The first pertains to the information that is being fed into the master filter. Between measurement updates of the master filter, the prefilters process multiple batches of measurements from the integrate and dump functions. The master filter does not process the raw measurements, it only receives the states of the prefilters. This raises the question as to whether there is a loss in optimality when the master filter only uses the prefilters' estimates instead of the raw measurements themselves.

The second issue relates to the coupled and decoupled pseudorange-state filter discussed in Section 6.3. When the process noise covariance matrix of the pseudorange-state filter is block diagonal, no information is shared between the channels of the receiver. In that situation, the pseudorange-state filter can be broken down into multiple smaller filters that only process measurements from a single satellite. In other words, the pseudorange-state filter is no longer performing vector tracking and has reverted back to scalar tracking. The prefilters used in federated DI architectures do not share information between themselves.

The bank of prefilters essentially function as a collection of multiple scalar tracking loops. The loss in the performance exhibited by the decoupled pseudorange-state filter means that there should be a similar loss in performance when a bank of prefilters are used to preprocess the available GPS measurements.

Three different vector tracking architectures are used to investigate the issues related to using a federated filtering architecture. The first algorithm uses a single centralized Kalman filter to process the raw GPS measurements. No intermediate tracking loops or filters are used. This is representative of centralized DI algorithms. The second and third architectures use a federated filtering scheme. In both algorithms, a master filter tracks the receivers navigation states and controls its NCOs. A bank of prefilters process the raw GPS measurements and estimate the pseudorange and pseudorange-rate errors. At 1 Hertz, the master filter in both algorithms is updated using the states of the prefilters. However, the bank of prefilters differ in the two federated architectures. In one algorithm, the prefilters are coupled together and information is shared across the different channels. In the other algorithm, the prefilters are not coupled and no information is shared between the channels presented previously in Chapter 6. The two different prefilter configurations are accomplished by using the position-state and decoupled pseudorange-state formulations of vector tracking. The two formulations use identical process and measurement noise statistics. This ensures that the only difference between the two prefilter banks is the coupling of the channels.

The following sections describe in detail the different filtering architectures that are used followed by a covariance analysis of them. The covariance analysis of the three filter architectures is performed to study their ability to estimate the receiver's navigation states. The covariance analysis is performed under several different scenarios. Monte Carlo simulations of the algorithms are then performed to verify the covariance analysis.

8.3 Single Central Filter

The centralized filtering approach uses a single Kalman filter to process all the information provided by the different channels and track the receiver's position, velocity, and clock states. The central filter also predicts the pseudoranges and pseudorange-rates for all the channels. Figure 8.1 shows a block diagram of the centralized filtering approach.

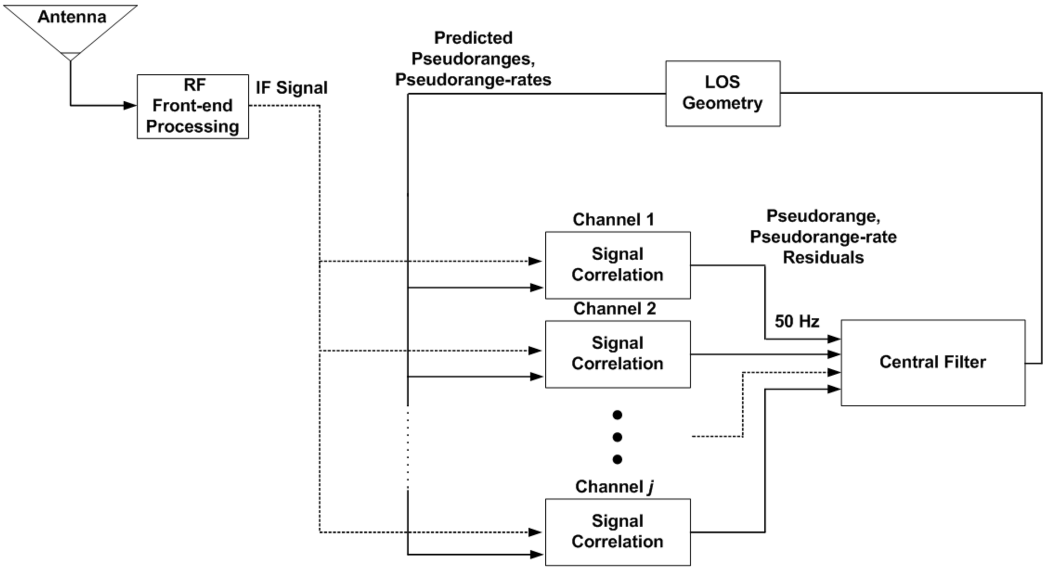

Figure 8.1: Single Centralized Filter Architecture

As shown in Figure 8.1, an integrate and dump time of 20 ms is used by the filter. Consequently, pseudoranges and pseudorange-rates are predicted by the central filter at 50 Hz. Pseudorange and pseudorange-rate residuals are supplied at the end of each integrate and dump period. These residuals are used by the filter to correct its state estimates. The updated states are then used to predict the next set of pseudoranges and pseudorange-rates.

The states of the central Kalman filter in the non-federated architecture are the receiver's position and clock states. The Earth-Centered Earth-Fixed (ECEF) coordinate frame is used to track the receiver's position and velocity. The state vector of the filter is shown in (8.1).

$$\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ ct \\ c\dot{t} \end{bmatrix} = \begin{bmatrix} \text{Position, X-axis} \\ \text{Velocity, X-axis} \\ \text{Position, Y-axis} \\ \text{Velocity, Y-axis} \\ \text{Position, Z-axis} \\ \text{Velocity, Z-axis} \\ \text{Clock Bias} \\ \text{Clock Drift} \end{bmatrix} \tag{8.1}$$

The clock bias and drift states are scaled by the speed of light c, giving them units of meters and meters/second, respectively. The discrete time state dynamics are shown in (8.2).

$$x(k+1) = A_d x(k) + w(k) \tag{8.2}$$

where

$$
\begin{aligned}
A_d &= \begin{bmatrix} \kappa & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & \kappa & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & \kappa & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & 0_{2\times2} & \kappa \end{bmatrix} \\
\kappa &= \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \\
w &= \begin{bmatrix} w_x \\ w_y \\ w_z \\ w_b \\ w_d \end{bmatrix}
\end{aligned}
$$

The variable ΔT represents the time between measurement updates of the filter. The process noise w is modeled as white and its statistics are shown in (8.3).

$$
\begin{aligned}
E\{w(k)w(k)^T\} &= \begin{bmatrix} Q_x & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & Q_y & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & Q_z & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & 0_{2\times2} & Q_c \end{bmatrix} \qquad (8.3) \\
Q_x &= \begin{bmatrix} \sigma_x^2 \frac{(\Delta T)^3}{3} & \sigma_x^2 \frac{(\Delta T)^2}{2} \\ \sigma_x^2 \frac{(\Delta T)^2}{2} & \sigma_x^2 (\Delta T) \end{bmatrix} \\
Q_y &= \begin{bmatrix} \sigma_y^2 \frac{(\Delta T)^3}{3} & \sigma_y^2 \frac{(\Delta T)^2}{2} \\ \sigma_y^2 \frac{(\Delta T)^2}{2} & \sigma_y^2 (\Delta T) \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
Q_z &= \begin{bmatrix} \sigma_z^2 \frac{(\Delta T)^3}{3} & \sigma_z^2 \frac{(\Delta T)^2}{2} \\ \sigma_z^2 \frac{(\Delta T)^2}{2} & \sigma_z^2 (\Delta T) \end{bmatrix} \\
Q_c &= \begin{bmatrix} \sigma_b^2 (\Delta T) + \sigma_d^2 \frac{(\Delta T)^3}{3} & \sigma_d^2 \frac{(\Delta T)^2}{2} \\ \sigma_d^2 \frac{(\Delta T)^2}{2} & \sigma_d^2 (\Delta T) \end{bmatrix}
\end{aligned}
$$

The process noise in the system w comes from two sources, receiver dynamics and clock noise. The noise sources (w_x, w_y, w_z) drive their respective velocity states. The terms (w_b, w_d) in w represent the clock phase and frequency error of the user's local oscillator, expressed in units of meters and meters per second [Brown and Hwang, 1996], respectively. The values for σ_b^2 and σ_d^2 are based on rule of thumb numbers for the type of oscillator (compensated crystal, rubidium, etc.) the receiver uses [Brown and Hwang, 1996].

The states of the filter are updated with the residuals from each channel. Two different discriminator functions are used to process the correlator outputs and produce measurements of the pseudorange and pseudorange-rate residuals. The relationship between the states of the filter and the pseudorange and pseudorange-rate residuals is shown in (8.4).

$$
\begin{aligned}
y &= Cx + v \qquad (8.4) \\
C &= \begin{bmatrix} a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 & 0 \\ 0 & a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 & 0 \\ 0 & a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 \end{bmatrix}
\end{aligned}
$$

In (8.4), the variables $a_{x,N}, a_{y,N}, a_{z,N}$ are the components of the line-of-sight unit vector from the user to the N-th satellite.

The statistics for the measurement noise vector v are shown in (8.5).

$$\begin{aligned} E\{v\} &= 0_{N\times 1} \\ E\{vv^T\} &= R_v \end{aligned} \tag{8.5}$$

The matrix R_v is a diagonal matrix because the noise in the different channels is uncorrelated and the noise in the pseudorange and pseudorange-rate measurements are uncorrelated. The individual entries in R_v are determined from the type of discriminators that is used and the current C/N_0.

8.4 Federated Filtering Architectures

The federated filtering architectures use a combination of a master filter and multiple smaller residual filters. The states of the master filter are used to predict the received pseudoranges and pseudorange-rates over a one second period. Pseudorange and pseudorange-rate residuals are produced in each channel of the receiver every 20 ms. The residual filter estimates the errors in the predictions of the master filter using the pseudorange and pseudorange-rate residuals. The estimates of the residual filters are fed back to correct the errors in the master filter's predictions. The states of the residual filter are then used as measurements by the master filter at the end of the one second period. After being used as measurements, the states of the residual filter are zeroed and the cycle is repeated.

Two different federated architectures are examined in this dissertation. The first uses a residual filter with coupled channels. This means that information is shared between the different channels of the receiver. The second architecture uses a residual filter without coupled channels. Without the channels being coupled, no information is shared between the different receiver channels. The details of the federated architectures are given in the following sections.

8.4.1 Federated Architecture with Coupled Channel Residual Filter

Figure 8.2 shows a block diagram of the federated architecture when using a coupled channel residual filter.

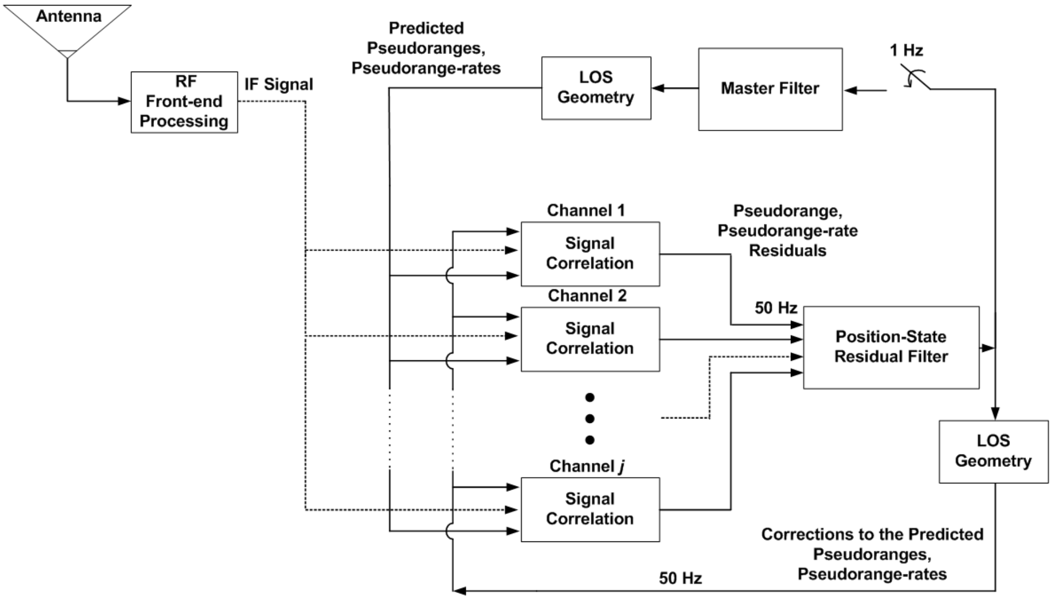

Figure 8.2: Federated Filter Architecture with Coupled-Channel Residual Filters

The master filter is updated at 1 Hz and its states are used to predict the pseudoranges and pseudorange-rates between measurement epochs. Between measurement epochs of the master filter, the residual filter processes the pseudorange and pseudorange-rate residuals produced by the integrate and dump operations at 50 Hz. The estimates of the residual filter (in accordance with the Line Of Sight (LOS) geometry) are routed back to the correct the master filter's predicted pseudoranges and pseudorange-rates. The feedback of the residual filter's estimates is optional, since the general filter architecture in Figure 8.2 can be operated without feedback. However, the discriminators used to transform the correlator outputs into pseudorange and pseudorange-rate residuals have a limited operating range. Therefore, the discriminators will produce erroneous measurements if the true residuals extend beyond their operating range. The operating point of the discriminators is maintained at approximately zero when feeding back the states of the residual filter.

The states of the residual filter used in the coupled channel architecture are shown in (8.6).

$$\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ ct \\ c\dot{t} \end{bmatrix} = \begin{bmatrix} \text{Position Error, X-axis} \\ \text{Velocity Error, X-axis} \\ \text{Position Error, Y-axis} \\ \text{Velocity Error, Y-axis} \\ \text{Position Error, Z-axis} \\ \text{Velocity Error, Z-axis} \\ \text{Clock Bias Error} \\ \text{Clock Drift Error} \end{bmatrix} \tag{8.6}$$

The residual filter tracks errors in the master filter's estimates of the receiver's position, velocity, and clock states. The position-state formulation of the vector tracking Kalman filter is used due to the numerical issues that are encountered when using the pseudorange-state formulation with coupled channels. The discrete time state dynamics of the filter are shown in (8.7).

$$x_R(k+1) = A_d x_R(k) + w_(k) \tag{8.7}$$

where

$$A_d = \begin{bmatrix} \kappa & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & \kappa & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & \kappa & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & 0_{2\times2} & \kappa \end{bmatrix}$$

$$\kappa = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}$$

In (8.7), the matrix A_d is the discrete time transition matrix for the system. The subscript R is used to distinguish the states of the residual filter from the states of the master filter. The vector w_k represents the disturbances caused by receiver dynamics and clock errors. The process noise vector w_k is modeled as white, zero mean, and has a covariance matrix given in (8.8).

$$E\{w(k)w(k)^T\} = \begin{bmatrix} Q_x & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & Q_y & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & Q_z & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & 0_{2\times2} & Q_c \end{bmatrix} \tag{8.8}$$

$$
\begin{aligned}
Q_x &= \begin{bmatrix} \sigma_x^2\frac{(\Delta T)^3}{3} & \sigma_x^2\frac{(\Delta T)^2}{2} \\ \sigma_x^2\frac{(\Delta T)^2}{2} & \sigma_x^2(\Delta T) \end{bmatrix} \\
Q_y &= \begin{bmatrix} \sigma_y^2\frac{(\Delta T)^3}{3} & \sigma_y^2\frac{(\Delta T)^2}{2} \\ \sigma_y^2\frac{(\Delta T)^2}{2} & \sigma_y^2(\Delta T) \end{bmatrix} \\
Q_z &= \begin{bmatrix} \sigma_z^2\frac{(\Delta T)^3}{3} & \sigma_z^2\frac{(\Delta T)^2}{2} \\ \sigma_z^2\frac{(\Delta T)^2}{2} & \sigma_z^2(\Delta T) \end{bmatrix} \\
Q_c &= \begin{bmatrix} \sigma_b^2(\Delta T) + \sigma_d^2\frac{(\Delta T)^3}{3} & \sigma_d^2\frac{(\Delta T)^2}{2} \\ \sigma_d^2\frac{(\Delta T)^2}{2} & \sigma_d^2(\Delta T) \end{bmatrix}
\end{aligned}
$$

The terms σ_x, σ_y, and σ_z are the variances of the noise processes driving the velocity states. The terms σ_b and σ_d are the variance of the clock phase and frequency jitter noise processes, respectively.

The relationship between the states of the residual filter and the pseudorange and pseudorange-rate residuals supplied at time k is shown in (8.9).

$$
\begin{aligned}
\epsilon_R(k) &= C_R(x_R(k) - \hat{x}_R^-(k)) + v(k) \qquad (8.9) \\
C_R &= \begin{bmatrix} a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 & 0 \\ 0 & a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 & 0 \\ 0 & a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 \end{bmatrix}
\end{aligned}
$$

The observation matrix C_R for the residual filter is identical to the observation matrix used by the centralized filter in the previous section. The state estimate of the residual filter after the measurement update is given in (8.10).

$$\hat{x}_R^+(k) = \hat{x}_R^-(k) + K_R(k)(\epsilon_R(k)) \tag{8.10}$$

The pseudorange and pseudorange-rate residuals ϵ_R generated by the discriminators is directly multiplied by the Kalman gain K_R. This is a consequence of the states of the residual filter being fed back to control the code and carrier NCOs. If the estimates of the residual filter were not fed back, the measurement update equation in (8.10) would be changed to (8.11).

$$\hat{x}_R^+(k) = \hat{x}_R^-(k) + K_R(k)(\epsilon_R(k) - C_R\hat{x}_R^-(k)) \tag{8.11}$$

The master filter is designed to accommodate the residual filter. As a consequence, the master filter's state vector is partioned into two segments. The first segment contains states that correspond to the receiver's position, velocity, and clock states. The second segment contains states that represent the errors in the residual filter's state estimates. Appendix D explains how the error states are derived. The full state vector is shown in (8.12).

$$\begin{bmatrix} x_M \\ \delta x_R \end{bmatrix} = \begin{bmatrix} \text{Master Filter's Navigation States} \\ \text{Residual Filter State Errors} \end{bmatrix}$$

where

$$
\begin{aligned}
x_M &= \begin{bmatrix} \text{Position, X-axis} \\ \text{Velocity, X-axis} \\ \text{Position, Y-axis} \\ \text{Velocity, Y-axis} \\ \text{Position, Z-axis} \\ \text{Velocity, Z-axis} \\ \text{Clock Bias} \\ \text{Clock Drift} \end{bmatrix} \\
\delta x_R &= \begin{bmatrix} \text{Residual Filter Error, State One} \\ \text{Residual Filter Error, State Two} \\ \vdots \\ \text{Residual Filter Error, State Eight} \end{bmatrix}
\end{aligned} \tag{8.12}
$$

The pseudoranges and pseudorange-rates are predicted based of the states corresponding to the receiver's position, velocity, clock bias, and clock drift. The extra states in the master filter are appended to model the errors in the states of the residual filter. As shown in (8.12), each state in the residual filter has a corresponding error state in the master filter.

The error states are defined in (8.13).

$$
\delta x_R(k) = x_R(k) - \hat{x}_R^+(k) \tag{8.13}
$$

The error states correspond to the difference between the true residuals and the residual filter's estimates. The dynamics of the error states are shown in (8.14).

$$\begin{aligned}
\delta x_R(k+1) &= (A_d - K_R(k)C_R A_d)\delta x_R(k) + (I - K_R(k)C_R)w(k) - K_R(k)v(k+1) \\
\delta x_R(k+1) &= A^{\delta x}\delta x_R(k) + \Gamma_w w(k) + \Gamma_v v(k+1) \\
&\text{where} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (8.14) \\
A^{\delta x} &= (A_d - K_R(k)C_R A_d) \\
\Gamma_w &= (I - K_R(k)C_R) \\
\Gamma_v &= -K_R(k)
\end{aligned}$$

In (8.14), the matrix A_d is the same as is given in (8.7). The noise terms w and v are the same as shown in (8.7) and (8.9). The matrices K_R and C_R are the Kalman gain and observation matrix of the residual state filter, respectively.

The state dynamics of the master filter are given in (8.15).

$$\begin{bmatrix} x_M(k+1) \\ \delta x_R(k+1) \end{bmatrix} = \begin{bmatrix} A_d & 0_{8\times 8} \\ 0_{8\times 8} & A^{\delta x} \end{bmatrix} \begin{bmatrix} x_M(k) \\ \delta x_R(k) \end{bmatrix} + \begin{bmatrix} I_{8\times 8} \\ \Gamma_w \end{bmatrix} w(k) + \begin{bmatrix} 0_{8\times 8} \\ \Gamma_v \end{bmatrix} v(k+1) \tag{8.15}$$

The statistics of the noise vector w are the same as given in (8.8).

The states of the residual filter are used as measurements by the master filter. The states of the residual filter are equal to the true residuals plus the errors in its estimates:

$$x_R^+(k) = x_M(k) - \hat{x}_M^-(k) - \delta x_R(k) \tag{8.16}$$

In general, the residuals of the filter are related to the states of the filter and the true states by:

$$\epsilon(k) = C(x(k) - \hat{x}^-(k)) + v(k) \tag{8.17}$$

Rearranging (8.16) into the form of (8.17) yields:

$$\begin{aligned} x_R^+(k) &= x_M(k) - \hat{x}_M^-(k) - \delta x_R(k) \\ \epsilon(k) &= x_R^+(k) + \delta x_R^-(k) \\ \epsilon(k) &= x_M(k) - \delta x_R(k) - \hat{x}_M^-(k) + \delta\hat{x}_R^-(k) \end{aligned} \tag{8.18}$$

The observation matrix for the master filter is therefore given by (8.19).

$$C_M = [I_{8\times8} \quad -I_{8\times8}] \tag{8.19}$$

After each measurement epoch of the master filter, the state vector of the residual filter is reset to zeros. This is because the state estimates of the master filter are unbiased and the residual filter is tracking the errors in the master filter's states. As a consequence of this, the errors in the master filter's states after a measurement update are correlated with the errors in the residual filter's states. The covariance matrix of the master filter after a measurement update is given in (8.20)

$$P^+(k) = \begin{bmatrix} E\{\hat{x}_M^+(k)(\hat{x}_M^+(k))^T\} & E\{\hat{x}_M^+(k)(\delta\hat{x}_R^+(k))^T\} \\ E\{\delta\hat{x}_R^+(k)(\hat{x}_M^+(k))^T\} & E\{\delta\hat{x}_R^+(k)(\delta\hat{x}_R^+(k))^T\} \end{bmatrix} \tag{8.20}$$

After the measurement update of the master filter and after the states of the residual filter are reset to zero, the errors in the residual filter's states are identical to the errors in the states of the master filter. Consequently, the covariance matrix of the master filter is therefore set equal to:

$$P^{+}(k) \quad = \quad \begin{bmatrix} E\{\hat{x}_M^{+}(k)(\hat{x}_M^{+}(k))^T\} & E\{\hat{x}_M^{+}(k)(\hat{x}_M^{+}(k))^T\} \\ E\{\hat{x}_M^{+}(k)(\hat{x}_M^{+}(k))^T\} & E\{\hat{x}_M^{+}(k)(\hat{x}_M^{+}(k))^T\} \end{bmatrix} \tag{8.21}$$

8.4.2 Federated Architecture with Decoupled Channel Residual Filters

Figure 8.3 shows a block diagram of the federated architecture when using a decoupled channel residual filter.

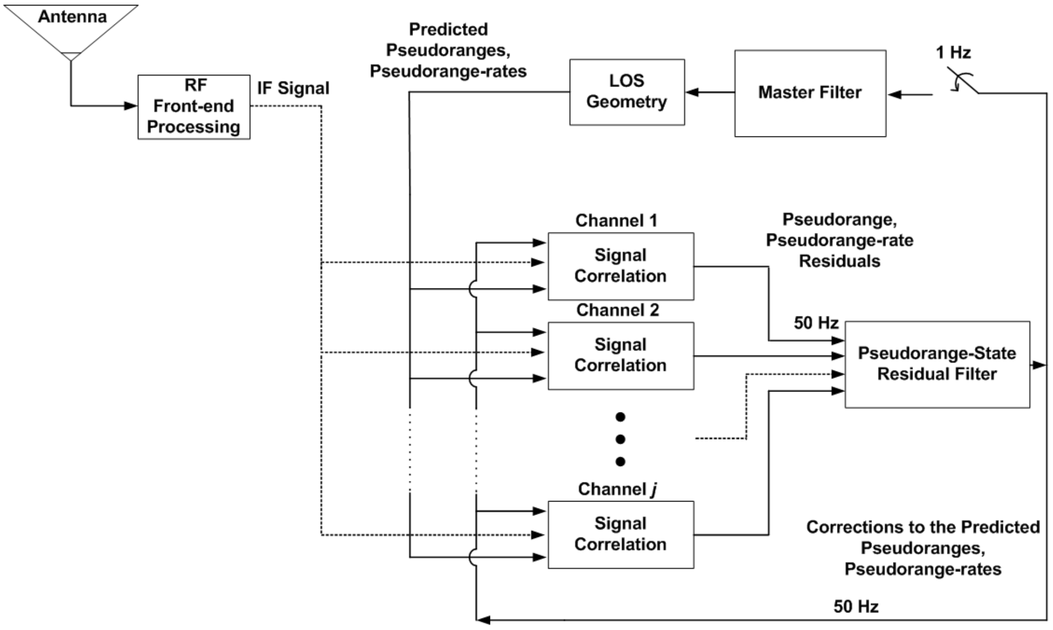
Antenna
RF Front-end Processing
IF Signal
Predicted Pseudoranges, Pseudorange-rates
LOS Geometry
Master Filter
1 Hz
Channel 1
Signal Correlation
Channel 2
Signal Correlation
Channel j
Signal Correlation
Pseudorange, Pseudorange-rate Residuals
50 Hz
Pseudorange-State Residual Filter
Corrections to the Predicted Pseudoranges, Pseudorange-rates
50 Hz

Figure 8.3: Federated Filter Architecture with Decoupled-Channel Residual Filters

The architectures shown in Figures 8.3 and 8.2 are very similar. The master filter is updated at 1 Hz and its states are used to predict the pseudoranges and pseudorange-rates between measurement epochs. The residual filter processes the pseudorange and pseudorange-rate residuals produced by the integrate and dump operations at 50 Hz and its estimates are routed back to correct the master filter's predictions. The main difference between Figures 8.3 and 8.2 is the design of the residual filter. The channels of the residual filter in Figure 8.3 are not coupled together. The pseudorange-state formulation of vector tracking is used to decouple the receiver channels, as described in Chapter 6. Since the residual filter tracks the pseudorange and pseudorange-rate errors, its estimates can be fed directly back to correct the master filter's predictions. The states of the residual filter are used as measurements by the master filter at 1 Hz. The residual filter states are then reset and the cycle is repeated.

The states of the residual filter used in the decoupled channel architecture are shown in (8.22).

$$\begin{bmatrix} \delta\rho_1 \\ \delta\dot{\rho}_1 \\ \delta\rho_2 \\ \delta\dot{\rho}_2 \\ \vdots \\ \delta\rho_N \\ \delta\dot{\rho}_N \end{bmatrix} = \begin{bmatrix} \text{Pseudorange Error, Channel One} \\ \text{Pseudorange-rate Error, Channel One} \\ \text{Pseudorange Error, Channel Two} \\ \text{Pseudorange-rate Error, Channel Two} \\ \vdots \\ \text{Pseudorange Error, Channel } N \\ \text{Pseudorange-rate Error, Channel } N \end{bmatrix} \tag{8.22}$$

The residual filter tracks errors in the master filter's predictions of the pseudoranges and pseudorange-rates. The pseudorange-state formulation of the vector tracking Kalman filter is used so that the channels can be decoupled. The pseudorange-state VDFLL does not exhibit numerical problems if the channels of the filter are decoupled. The state dynamics of the filter are shown in (8.23).

$$x_R(k+1) = A_d x_R(k) + w_\rho(k) \tag{8.23}$$

where

$$A_d = \begin{bmatrix} \kappa & 0_{2\times2} & \cdots & 0_{2\times2} \\ 0_{2\times2} & \kappa & & \vdots \\ \vdots & & \ddots & 0_{2\times2} \\ 0_{2\times2} & \cdots & 0_{2\times2} & \kappa \end{bmatrix}$$

$$\kappa = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}$$

In (8.23), the matrix A_d is the discrete time transition matrix for the system. The subscript R is again used to distinguish the states of the residual filter from the states of the master filter. The vector w_ρ represents the disturbances caused by receiver dynamics and clock errors. The process noise vector w_ρ is modeled as white, zero mean, and with the covariance matrix given in (8.24).

$$
\begin{aligned}
E\{w_\rho(k)(w_\rho(k))^T\} &= \begin{bmatrix} Q^c & 0_{2\times 2} & \dots & 0_{2\times 2} \\ 0_{2\times 2} & Q^c & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{2\times 2} \\ 0_{2\times 2} & \dots & 0_{2\times 2} & Q^c \end{bmatrix} + \begin{bmatrix} Q^d_{1,1} & 0_{2\times 2} & \dots & 0_{2\times 2} \\ 0_{2\times 2} & Q^d_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{2\times 2} \\ 0_{2\times 2} & \dots & 0_{2\times 2} & Q^d_{N,N} \end{bmatrix} \\
&\text{where} \qquad (8.24) \\
Q^d_{i,j} &= \begin{bmatrix} \beta_{i,j}\frac{T^3}{3} & \beta_{i,j}\frac{T^2}{2} \\ \beta_{i,j}\frac{T^2}{2} & T\beta_{i,j} \end{bmatrix} \\
\beta_{i,j} &= \begin{bmatrix} \sigma_x^2 a_{x,i}a_{x,j} + \sigma_y^2 a_{y,i}a_{y,j} + \sigma_z^2 a_{z,i}a_{z,j} \end{bmatrix} \\
Q^c &= \begin{bmatrix} \sigma_b^2 T + \sigma_d^2\frac{T^3}{3} & \sigma_d^2\frac{T^2}{2} \\ \sigma_d^2\frac{T^2}{2} & \sigma_d^2 T \end{bmatrix}
\end{aligned}
$$

The terms σ_x, σ_y, and σ_z are the variances of the noise processes driving the velocity states. The terms σ_b and σ_d are the variance of the clock phase and frequency jitter noise processes, respectively. The variables $a_{x,i}, a_{y,i}, a_{z,i}$ are the components of the line-of-sight unit vector from the receiver to the i-th satellite.

The relationship between the states of the residual filter and the pseudorange and pseudorange-rate residuals supplied at time k is shown in (8.25).

$$
\epsilon_R(k) = x_R(k) - \hat{x}_R^-(k) + v(k) \qquad (8.25)
$$

The observation matrix for the residual filter is an identity matrix, since the states and measurements are identical. The state estimate of the residual filter after the measurement update is given in (8.26).

$$\hat{x}_R^+(k) = \hat{x}_R^-(k) + K_R(k)(\epsilon_R(k)) \tag{8.26}$$

The master filter is again designed to accommodate the residual filter. The master filter's state vector is partioned into two segments. The first segment contains states that correspond to the receiver's position, velocity, and clock states. The second segment contains states that represent the error in the residual filter's state estimates. The full state vector is shown in (8.27).

$$\begin{aligned} \begin{bmatrix} x_M \\ \delta x_R \end{bmatrix} &= \begin{bmatrix} \text{Master Filter's Navigation States} \\ \text{Residual Filter State Errors} \end{bmatrix} \\ \text{where} & \\ x_M &= \begin{bmatrix} \text{Position, X-axis} \\ \text{Velocity, X-axis} \\ \text{Position, Y-axis} \\ \text{Velocity, Y-axis} \\ \text{Position, Z-axis} \\ \text{Velocity, Z-axis} \\ \text{Clock Bias} \\ \text{Clock Drift} \end{bmatrix} \end{aligned} \tag{8.27}$$

$$\delta x_R = \begin{bmatrix} \text{Residual Filter Error, State One} \\ \text{Residual Filter Error, State Two} \\ \vdots \\ \text{Residual Filter Error, State } N \end{bmatrix}$$

The pseudoranges and pseudorange-rates are predicted based off the states corresponding to the receiver's position, velocity, clock bias, and clock drift. The extra states in the master filter are appended to model the errors in the states of the residual filter. As shown in (8.27), each state in the residual filter has a corresponding error state in the master filter. The error states are defined in (8.28).

$$\delta x_R(k) = x_R(k) - \hat{x}_R^+(k) \tag{8.28}$$

The error states correspond to the difference between the true residuals and the residual filter's estimates. The dynamics of the error states are shown in (8.29).

$$\begin{aligned}
\delta x_R(k+1) &= (A_d - K_R(k)C_R A_d)\delta x_R(k) + (I - K_R(k)C_R)Hw(k) - K_R(k)v(k+1) \\
\delta x_R(k+1) &= A^{\delta x}\delta x_R(k) + \Gamma_w w(k) + \Gamma_v v(k+1) \\
&\text{where} \\
A^{\delta x} &= (A_d - K_R(k)C_R A_d) \\
\Gamma_w &= (I - K_R(k)C_R) \\
\Gamma_v &= -K_R(k)
\end{aligned} \tag{8.29}$$

$$
H = \begin{bmatrix}
a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 & 0 \\
0 & a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 & 0 \\
0 & a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1
\end{bmatrix}
$$

In (8.29), the matrix A_d is the same as is given in (8.23). The process noise vector w is identical to that shown in (8.2) and (8.14). The process noise corresponds to stochastic changes in the receiver's position, velocity, and clock states. The process noise is multiplied by the matrix H in order to relate it to the pseudorange and pseudorange-rate prediction errors. Since the observation matrix for the decoupled pseudorange-state filter is an identity matrix, equation (8.29) can be simplified to (8.30).

$$
\begin{aligned}
\delta x_R(k+1) &= A^{\delta x}\delta x_R(k) + \Gamma_w H w(k) + \Gamma_v v(k+1) \\
&\text{where} \\
A^{\delta x} &= (A_d - K_R(k)A_d) \\
\Gamma_w &= (I - K_R(k)) \\
\Gamma_v &= -K_R(k)
\end{aligned}
\tag{8.30}
$$

The noise terms w and v are the same shown in (8.7) and (8.9). The matrices K_R and C_R are the Kalman gain and observation matrix of the residual state filter, respectively.

The state dynamics of the master filter are given in (8.31).

$$\begin{bmatrix} x_M(k+1) \\ \delta x_R(k+1) \end{bmatrix} = \begin{bmatrix} A_d & 0_{8\times N} \\ 0_{N\times 8} & A^{\delta x} \end{bmatrix} \begin{bmatrix} x_M(k) \\ \delta x_R(k) \end{bmatrix} + \begin{bmatrix} I_{8\times 8} \\ \Gamma_w H \end{bmatrix} w(k) + \begin{bmatrix} 0_{8\times 8} \\ \Gamma_v \end{bmatrix} v(k+1) \tag{8.31}$$

The states of the residual filter are used as measurements by the master filter. The observation matrix for the master filter is given by (8.32).

$$C_M = [H \quad -I_{N\times N}] \tag{8.32}$$

The state vector of the residual filter is reset to zeros after each measurement epoch of the master filter. This is because the state estimates of the master filter are unbiased and the residual filter is tracking the errors in the master filter's states. As a consequence of this, the errors in the master filter's states after a measurement update are correlated with the errors in the residual filter's states. The covariance matrix of the master filter after a measurement update is given in (8.33)

$$P^+(k) = \begin{bmatrix} E\{\hat{x}_M^+(k)(\hat{x}_M^+(k))^T\} & E\{\hat{x}_M^+(k)(\delta\hat{x}_R^+(k))^T\} \\ E\{\delta\hat{x}_R^+(k)(\hat{x}_M^+(k))^T\} & E\{\delta\hat{x}_R^+(k)(\delta\hat{x}_R^+(k))^T\} \end{bmatrix} \tag{8.33}$$

After the measurement update of the master filter and after the states of the residual filter are reset to zeros, the errors in the residual filter's states are related to the errors in the states of the master filter by:

$$\delta\hat{x}_R^+(k) \quad = \quad H\hat{x}_M^+(k) \tag{8.34}$$

Consequently, the covariance matrix of the master filter is therefore set equal to:

$$P^+(k) \quad = \quad \begin{bmatrix} E\{\hat{x}_M^+(k)(\hat{x}_M^+(k))^T\} & E\{\hat{x}_M^+(k)(\hat{x}_M^+(k))^T\}\cdot H^T \\ H\cdot E\{\hat{x}_M^+(k)(\hat{x}_M^+(k))^T\} & H\cdot E\{\hat{x}_M^+(k)(\hat{x}_M^+(k))^T\}\cdot H^T \end{bmatrix} \tag{8.35}$$

8.5 Covariance Analysis of Vector Tracking Architectures

A covariance analysis of the three filter architectures is performed to study the differences in their performance. The covariance analysis is performed with two different satellite constellations. One satellite constellation contains eleven satellites and the other contains five satellites. The receiver's motion and clock are modeled using the formula shown in (8.36).

$$x(k+1) \quad = \quad A_d x(k) + w(k) \tag{8.36}$$

where

$$
\begin{aligned}
x_{8\times1} &= \begin{bmatrix} \text{Position, X-axis} \\ \text{Velocity, X-axis} \\ \text{Position, Y-axis} \\ \text{Velocity, Y-axis} \\ \text{Position, Z-axis} \\ \text{Velocity, Z-axis} \\ \text{Clock Bias} \\ \text{Clock Drift} \end{bmatrix} \\
A_d &= \begin{bmatrix} \kappa & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & \kappa & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & \kappa & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & 0_{2\times2} & \kappa \end{bmatrix} \\
\kappa &= \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \\
w &= \begin{bmatrix} w_x \\ w_y \\ w_z \\ w_b \\ w_d \end{bmatrix}
\end{aligned}
$$

The variable ΔT represents the time between measurement updates of the filter. A time of 20 ms is used for the covariance analysis and subsequent Monte Carlo simulations. The model in (8.36) matches the models used by the Kalman filters in the different receiver architectures. The velocity and clock states of the receiver are modeled as being driven by the noise process w. The process noise w is a Gaussian, white sequence and its statistics are shown in (8.37).

$$\begin{aligned}
E\{w(k)w(k)^T\} &= \begin{bmatrix} Q_x & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & Q_y & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & Q_z & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & 0_{2\times2} & Q_c \end{bmatrix} \qquad (8.37) \\
Q_x &= \begin{bmatrix} \sigma_x^2\frac{(\Delta T)^3}{3} & \sigma_x^2\frac{(\Delta T)^2}{2} \\ \sigma_x^2\frac{(\Delta T)^2}{2} & \sigma_x^2(\Delta T) \end{bmatrix} \\
Q_y &= \begin{bmatrix} \sigma_y^2\frac{(\Delta T)^3}{3} & \sigma_y^2\frac{(\Delta T)^2}{2} \\ \sigma_y^2\frac{(\Delta T)^2}{2} & \sigma_y^2(\Delta T) \end{bmatrix} \\
Q_z &= \begin{bmatrix} \sigma_z^2\frac{(\Delta T)^3}{3} & \sigma_z^2\frac{(\Delta T)^2}{2} \\ \sigma_z^2\frac{(\Delta T)^2}{2} & \sigma_z^2(\Delta T) \end{bmatrix} \\
Q_c &= \begin{bmatrix} \sigma_b^2(\Delta T) + \sigma_d^2\frac{(\Delta T)^3}{3} & \sigma_d^2\frac{(\Delta T)^2}{2} \\ \sigma_d^2\frac{(\Delta T)^2}{2} & \sigma_d^2(\Delta T) \end{bmatrix}
\end{aligned}$$

The individual terms of w represent the noise driving the velocity states (w_x, w_y, w_z) and the noise driving the clock states (w_b, w_d). The terms (w_b, w_d) in w represent the clock phase and frequency error of the user's local oscillator, expressed in units of meters and meters per second [Brown and Hwang, 1996], respectively. The values for σ_b^2 and σ_d^2 are based on rule of thumb numbers for a temperature compensated crystal oscillator (TCXO).

The power spectral density coefficients used for the oscillator are [Brown and Hwang, 1996]:

$$\begin{aligned} h_0 &= 2 \cdot 10^{-19} \\ h_{-1} &= 7 \cdot 10^{-21} \\ h_{-2} &= 2 \cdot 10^{-20} \end{aligned} \tag{8.38}$$

The covariance analysis is performed with differing values of σ_x^2, σ_y^2, and σ_z^2. The magnitudes of the three terms are always equal to each other in the individual scenarios. Values of 2, 10, and 50 m^2/s^2 are used for the process noise variances to represent varying levels of user dynamics.

The measurement noise statistics are defined by the type of discriminators that are used and the C/N_0 ratio. A pseudorange and pseudorange-rate discriminator are used to transform the correlator outputs into usable measurements for the filters. The pseudorange discriminator uses the correlator outputs accumulated over the entire 20 ms integrate and dump interval [Crane, 2007]:

$$\begin{aligned} Y_R(k) &= IE(k)^2 + QE(k)^2 - IL(k)^2 - QL(k)^2 \qquad (8.39) \\ Y_R(k) &= A(k)^2 \Big(\frac{2 \cdot \rho_e(k)}{\beta}\Big) + v_R(k) \\ \rho_e(k) &= \text{Pseudorange error (meters)} \\ \beta &= \text{PRN chip length, 293.3 meters for C/A code} \\ A(k) &= \text{Signal amplitude} \\ v_R(k) &= \text{Noise} \end{aligned}$$

The mean and mean squared value of the random variable v_R are:

$$\begin{aligned} E\{v_R\} &= 0 \\ E\{v_R^2\} &= \frac{\beta^2}{2(TC/N_o)^2} + \frac{\beta^2}{TC/N_o}\left(\frac{\rho_e^2}{\beta^2} + \frac{1}{4}\right) \text{ (meters}^2\text{)} \end{aligned} \tag{8.40}$$

The pseudorange-rate discriminator uses correlator outputs from two adjacent 10 ms integrate and dump operations to generate pseudorange-rate residuals $\dot{\rho}_{err}(k)$ [Crane, 2007]:

$$\begin{aligned} Y_{RR}(k) &= IP_2(k)QP_1(k) - IP_1(k)QP_2(k) \\ Y_{RR}(k) &\approx -A_F(k)^2R^2(\rho_e(k))\frac{\pi f_{err}(k)Tc}{4f_{L1}} + v_{RR} \\ \rho_e(k) &= \text{Pseudorange error error (meters)} \\ \dot{\rho}_{err}(k) &= \text{Pseudorange-rate error (m/s)} \\ v_{RR} &= \text{Noise} \end{aligned} \tag{8.41}$$

The mean and mean squared value of v_{RR} are:

$$\begin{aligned} E\{v_{RR}\} &= 0 \\ E\{v_{RR}^2\}^\dagger &= \left(\frac{2}{(TC/N_o)^2} + \frac{2R^2(\rho_e(k))}{TC/N_o}\right)\left(\frac{c}{\pi T f_{L1}}\right)^2 \text{ (m/s)}^2 \\ \dagger &= \text{assuming } |\rho_e(k)| \leq \frac{\beta}{2} \end{aligned} \tag{8.42}$$

In (8.40) and (8.42) the value of T is the predetection integration interval used to generated the correlator outputs in (8.39), which in this work is 20 ms. An early-to-late correlator spacing of one chip is assumed. The noise terms in the discriminator outputs in (8.40) and (8.42) are uncorrelated with each other [Crane, 2007].

Table 8.1 summarizes the statistical characteristics of the eleven and five satellite geometries.

Table 8.1: Dilution of Precision Values for Different Satellite Configurations

Number of Satellites	PDOP	HDOP	VDOP
Eleven	1.4141	0.7527	1.1971
Five	2.3602	1.1300	2.0721

The covariance analysis of the different architectures was performed by taking the process and measurement noise statistics and then iterating the Kalman filter recursions until the state covariance matrix reached steady state. For the federated filter architectures, the state covariance matrix of the master filter was monitored for steady state conditions. The covariance analysis was performed for each filter architecture with three different process noise intensities and over a range of C/N_0 ratios.

The architecture of the centralized and federated architectures makes performing a valid comparison complicated. The reason for this is due to the different update rates used by the centralized filter and the master filters in the two federated algorithms. The centralized filter is updated with pseudorange and pseudorange-rate residuals at 50 Hz, while the master filters in the federated architectures receive measurements at 1 Hz. Between measurement updates, the covariance matrix of the federated filters increases because no measurements are available. This is an unavoidable consequence of using the federated architectures. The state covariance matrix only decreases after the 1 Hz measurement update. In the interest of making a valid comparison, the state covariance matrices of the different algorithms are compared immediately after the measurement updates of the master filters.

8.5.1 Covariance Analysis Results for Eleven Satellite Scenario

The x-axis position results of the covariance analysis for the eleven satellite and 2 m^2/s^2 process noise intensity scenario are shown in Figure 8.4. The C/N_0 ratio ranges from 45 to 19 dB-Hz in 1 dB increments. For each of the three architectures, the diagonal element of the covariance matrix corresponding to the receiver's x-axis position is plotted. Recall that this is taken from the covariance matrix after the 1 Hz update for the federated architectures. The variance of the position estimate error is plotted in units of dB-m^2. This is done simply to make the results of the covariance analysis easier to visually inspect. The C/N_0 ratio varies over a wide range of values and so the magnitude of the position estimate error variance changes drastically.

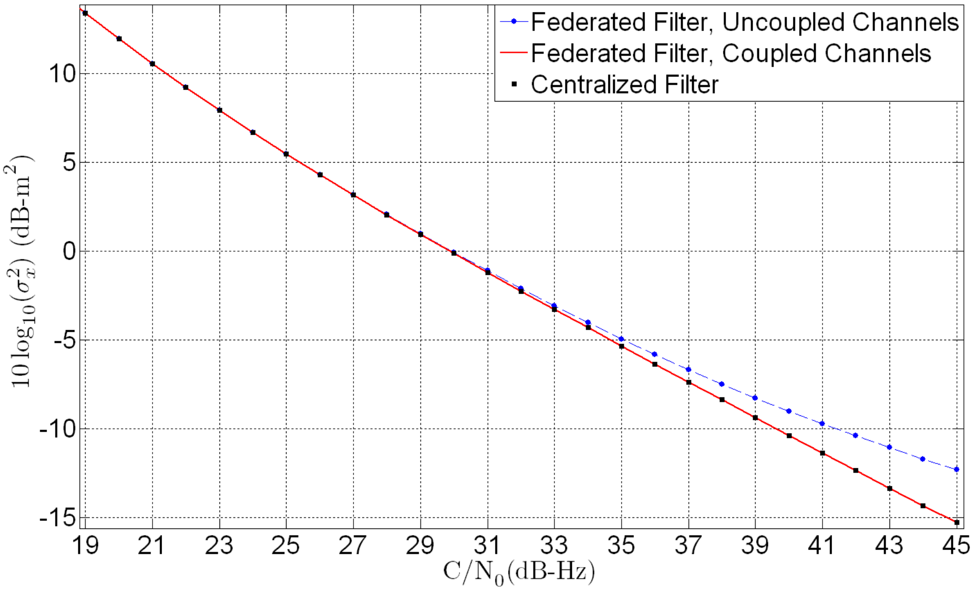

Figure 8.4: Covariance Analysis Results, X-axis Position Error for the Different Filter Architectures

Figure 8.4 shows that the federated filter with coupled channels and the centralized filtering architectures yield essentially identical results for every C/N_0 ratio. In contrast, the federated filter with uncoupled channels exhibits a higher variance at every C/N_0. The relative difference between the centralized filter and the federated filter with uncoupled channels decreases as the C/N_0 ratio decreases. The greatest disparity in performance is seen at 45 dB-Hz.

Figure 8.5 shows the results of the same covariance analysis for the x-axis velocity estimate. The variance of the x-axis velocity estimate is plotted in dB-m^2/s^2. Again, the centralized and coupled channel federated filter architectures produce virtually identical results. The federated architecture without coupled channels exhibits a higher variance for the velocity estimate at each C/N_0. The disparity also tends to decrease as the C/N_0 decreases.

The difference in the performance of the algorithms can be expressed in dB by examining the horizontal difference between the curves in Figures 8.4 and 8.5. This is done by taking the variance produced by the federated filter with decoupled channels at a given C/N_0 ratio and then determining the C/N_0 ratio at which the centralized and coupled channel federated filters yield the same variance. The difference in C/N_0 ratios is defined here as the difference in performance of the algorithms as measured in dB. Using this methodology, the performance difference between the decoupled channel federated filter, the centralized filter, and coupled channel federated filter for all three position and velocity states is shown in Figure 8.6. The corresponding differences in performance for the clock states is shown in Figure 8.7.

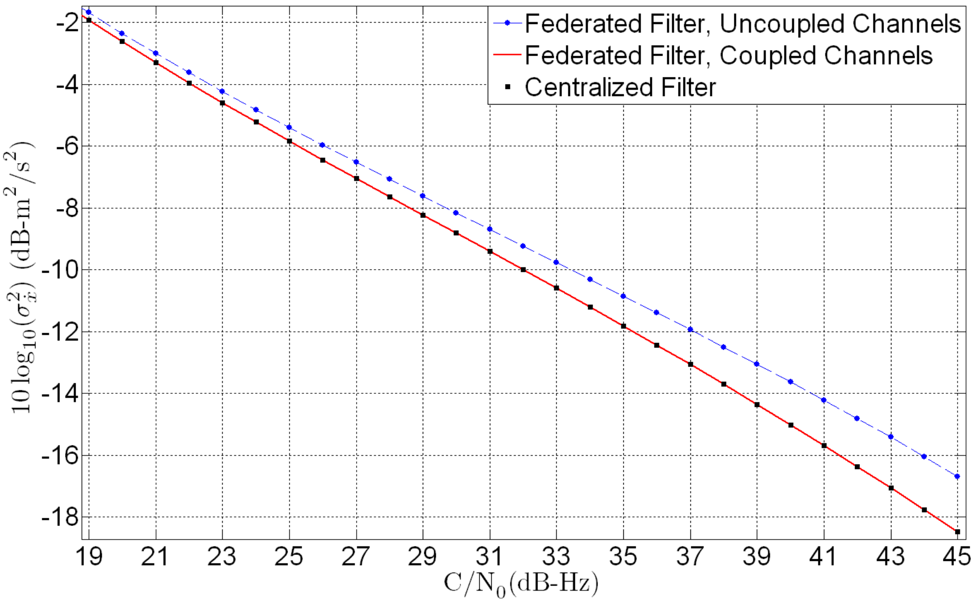

Figure 8.5: Covariance Analysis Results, X-axis Velocity Error for the Different Filter Architectures

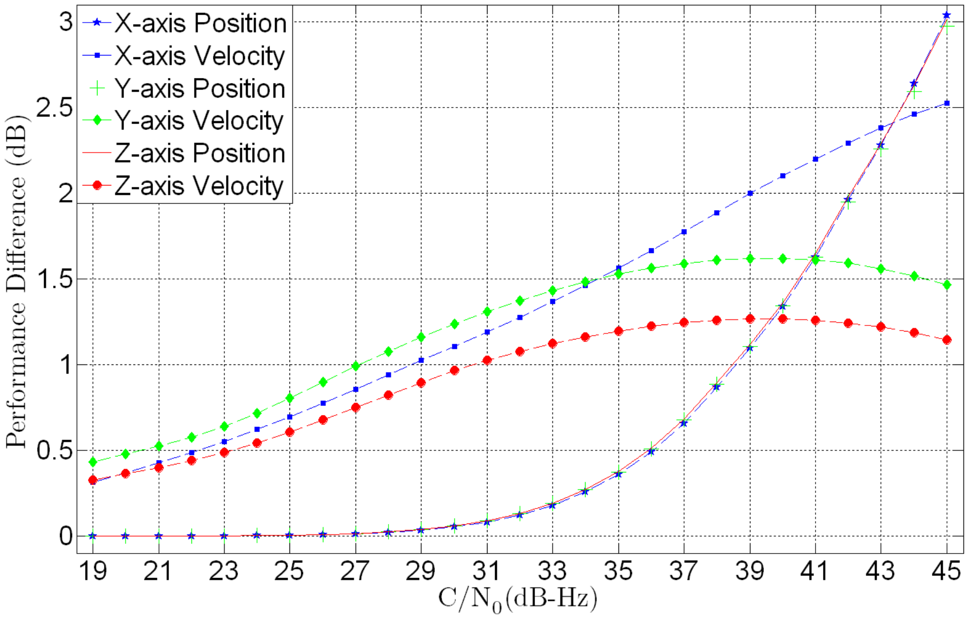

Figure 8.6: Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters, σ^2=2 m^2/s^2

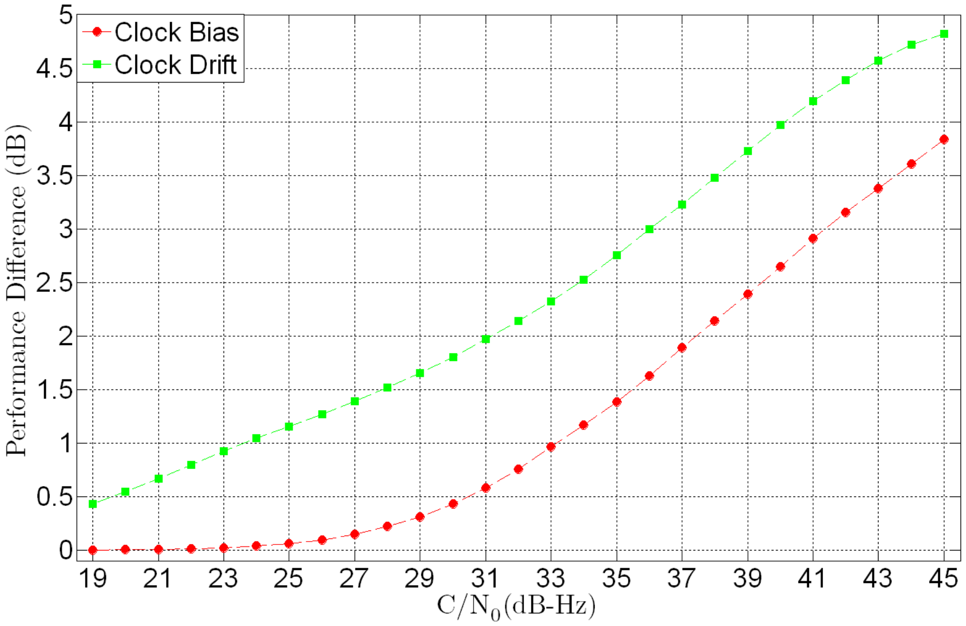

Figure 8.7: Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters, σ^2=2 m^2/s^2

As shown in Figures 8.6 and 8.7, the advantage offered by the centralized and the coupled channel federated architecture decreases as C/N_0 ratio decreases. At 45 dB-Hz, the federated algorithm with decoupled channels performs about 3 dB worse with respect to its position state estimates. At 19 dB-Hz, the difference in performance is virtually zero. The difference in the velocity state estimates has a maximum of about 1.6 dB and a minimum of less than .5 dB. The performance improvement in the clock state estimates also decreases as the C/N_0 ratio decreases. The difference in the clock drift variances has a maximum of about 4.8 dB at 45 dB-Hz and decreases to less than .5 dB at 19 dB-Hz. The improvement in the clock bias variance is more modest. The largest difference in performance is about 3.8 dB at 45 dB-Hz. At 19 dB-Hz, the difference in performance with respect to the clock bias variance is essentially zero.

The curves in Figures 8.6 and 8.7 show that the difference between the performance of the algorithms is itself a function of C/N_0. This is counter-intuitive and it would seem that one algorithm should have a constant performance advantage in decibels over the other algorithms. Furthermore, the difference in performance for the y and z velocity states is not strictly decreasing with the C/N_0. At about 39 dB-Hz, the difference in performance is greatest for the those two states. These observations raise questions as to the nature of the curves seen in Figures 8.6 and 8.7.

The shape of the curves in Figures 8.6 and 8.7 is explained by considering the nature of the Kalman filter. The filter essentially performs a trade-off between the measurement and process noise present in the system. The results in Figures 8.6 and 8.7 are generated by holding the process noise constant and varying the magnitude of the measurement noise. It should also be noted that the nature of the clock noise processes is different than that of the position and velocity states. This is shown in Equation (8.37), the clock bias is affected by clock phase and frequency noise. However, the position states are affected by the noise processes driving the velocity states. This is analogous to the clock frequency noise affecting the clock bias state. However, there is not a noise term that affects the position states like the clock phase affects the clock bias.

If the process noise covariance matrix is scaled along with the measurement noise covariance matrix, a constant difference in performance between the algorithms is obtained for each state. For example, consider the noise statistics of the pseudorange and pseudorange-rate discriminators in (8.40) and (8.42). At 45 dB-Hz the pseudorange and pseudorange-rate variances are approximately 34.1 m^2 and .0291 m^2/s^2, respectively. Assume that the measurement noise statistics are scaled according to (8.43).

$$\begin{aligned}
\sigma_\rho^2(C/N_0) &= 34.1 \cdot 2^{\frac{45-C/N_0}{3}}\ \mathrm{m}^2 \\
\sigma_{\dot\rho}^2(C/N_0) &= .0291 \cdot 2^{\frac{45-C/N_0}{3}}\ \mathrm{m}^2/\mathrm{s}^2 \qquad (8.43)
\end{aligned}$$

This amounts to doubling the measurement noise variances every time the C/N_0 ratio decreases by 3 dB. Furthermore, assume that the process noise covariance in (8.37) is similarly scaled according to (8.44).

$$Q(C/N_0) = \begin{bmatrix} Q_x & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & Q_y & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & Q_z & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & 0_{2\times2} & Q_c \end{bmatrix} \cdot 2^{\frac{45-C/N_0}{3}} \qquad (8.44)$$

Using the functions in (8.43) and (8.44), the covariance analysis of the three architectures was repeated. The centralized filter and the federated filter with coupled channels again yielded virtually identical results. The difference in their performance and the performance of the federated filter with uncoupled channels for the position and clock states is shown in Figures 8.8 and 8.9, respectively.

Figures 8.8 and 8.9 demonstrate that the performance improvements for all the different states are constant as the C/N_0 ratio changes. This explains that the varying performance improvements in Figures 8.6 and 8.7 are caused by increasing the measurement noise while holding the process noise constant.

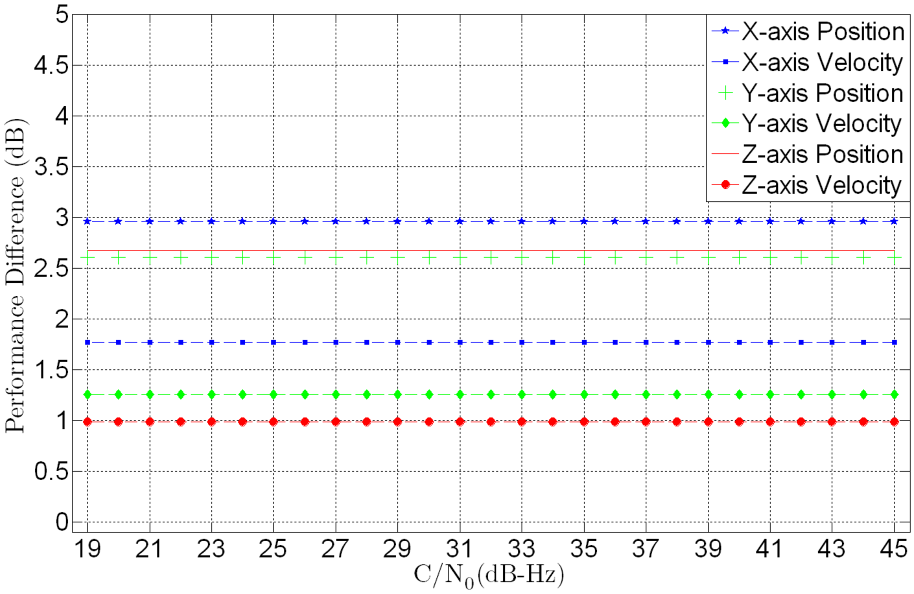

Figure 8.8: Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters when Varying the Process and Measurement Noise

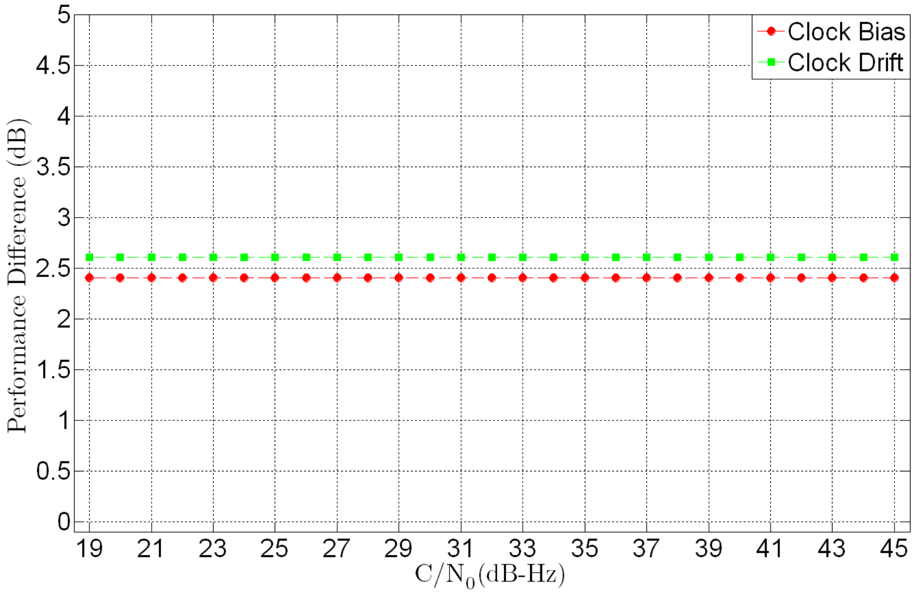

Figure 8.9: Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters when Varying the Process and Measurement Noise

Figures 8.10 and 8.11 show the difference in the performance of the federated filter with decoupled channels and the centralized filter for the process noise intensity of 10 m^2/s^2. Comparing Figures 8.6 and 8.10 shows that the difference in the performance for the position and velocity states is similar for the 2 and 10 m^2/s^2 process noise intensities. The difference in performance for the clock bias and drift states is larger with the 10 m^2/s^2 process noise variance.

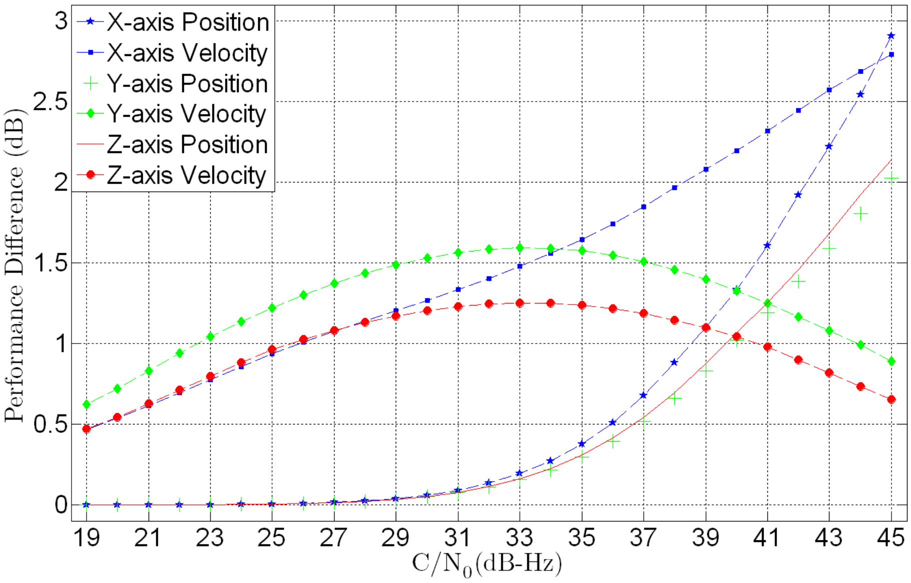

Figure 8.10: Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters, σ^2=10 m^2/s^2

Figures 8.12 and 8.13 show the difference in the performance of the federated filter with decoupled channels and the centralized filter for the process noise intensity of 50 m^2/s^2. The maximum difference in the performance of the algorithms occurs at about 28 dB-Hz for the y and z velocity states. The difference in the position state variances again becomes smaller as the C/N_0 decreases. At 19 dB-Hz, the difference is essentially nonextant. The

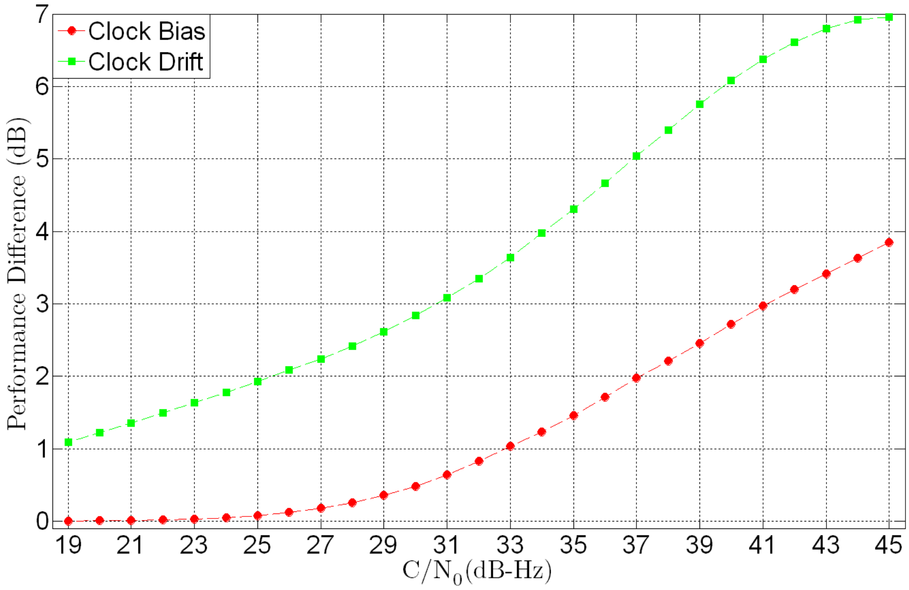

Figure 8.11: Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters, σ^2=10 m^2/s^2

difference in performance for the clock drift estimate is largest for the 50 m^2/s^2 process noise intensity. However, the difference in performance for the clock bias state is the smallest.

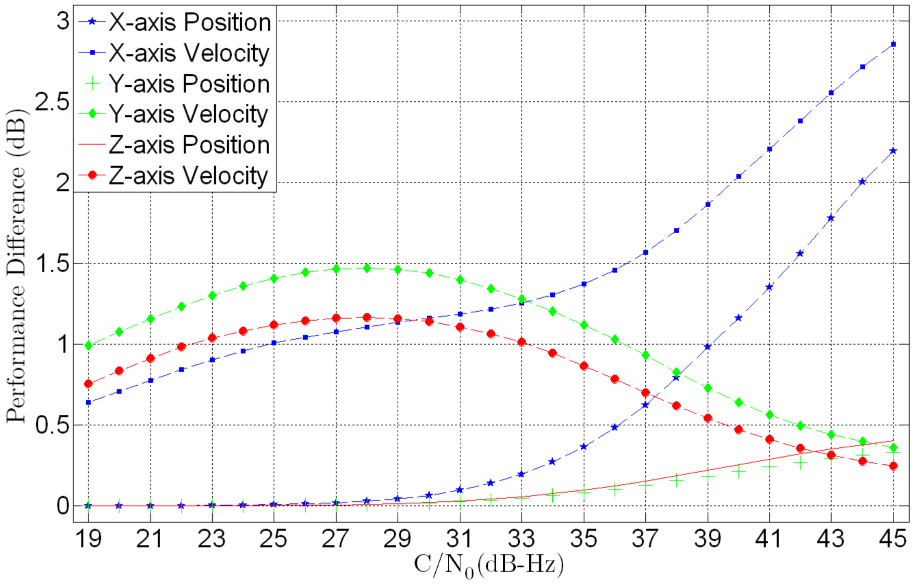

Figure 8.12: Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters, σ^2=50 m^2/s^2

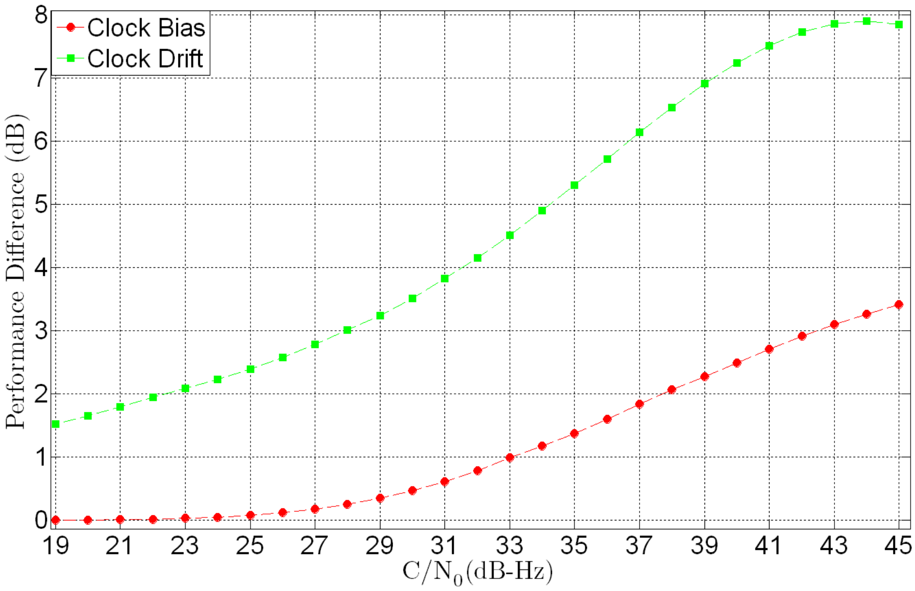

Figure 8.13: Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters, σ^2=50 m^2/s^2

8.5.2 Covariance Analysis Results for Five Satellite Scenario

The same covariance analysis presented in the previous section is carried out for the five satellite constellation. The same three process noise tunings are used for the alternate constellation. For the sake of brevity, only the results for the 2 and 50 m^2/s^2 process noise tuning are included here. Figures 8.14 and 8.15 show the difference in the performance of the federated filter with decoupled channels and the centralized filter for the process noise intensity of 2 m^2/s^2. Compared to Figure 8.6, the difference in performance with respect to the position states is similar for the five satellite constellation. The performance difference is less for the y and z velocities, but it is about the same for the x velocity. The difference in performance for the clock drift is slightly larger for the five satellite constellation and the performance difference for the clock bias is less.

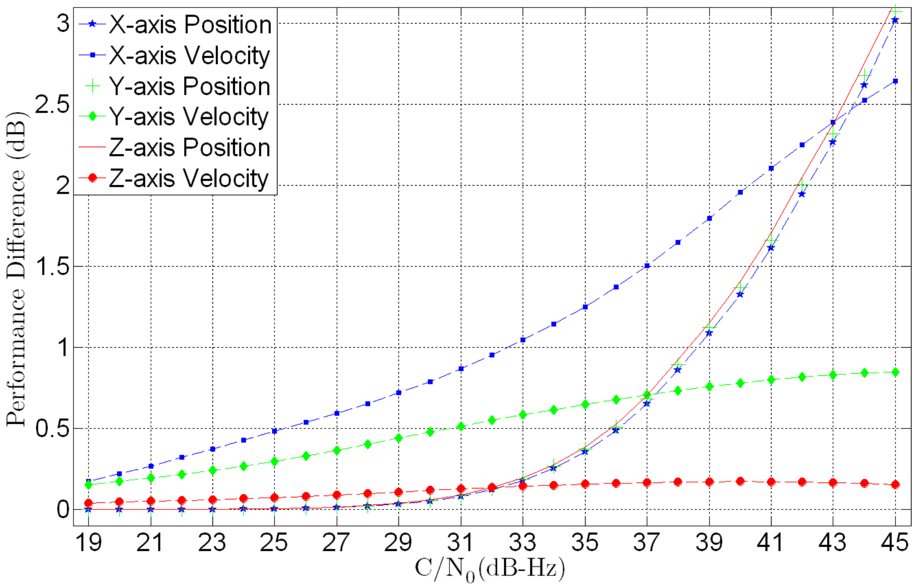

Figure 8.14: Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters, σ^2=2 m^2/s^2

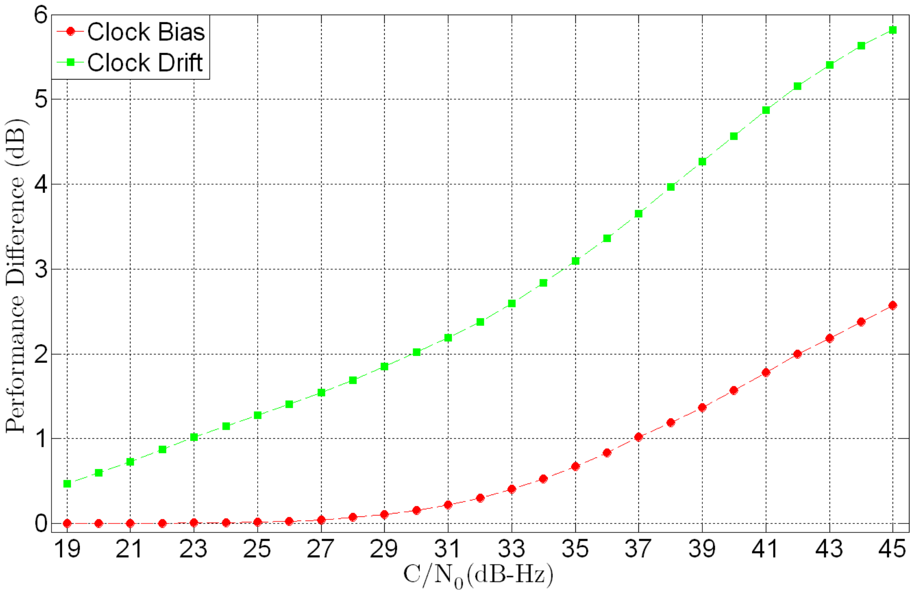

Figure 8.15: Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters, σ^2=2 m^2/s^2

Figures 8.16 and 8.17 show the difference in the performance of the federated filter with decoupled channels and the centralized filter for the process noise intensity of 50 m^2/s^2. The performance difference for the velocity states is generally lower than in the eleven satellite constellation. This is especially true at the lower C/N_0 ratios. In Figure 8.17, the performance difference for the clock drift is greater, while the performance for the clock bias is less relative to the eleven satellite constellation.

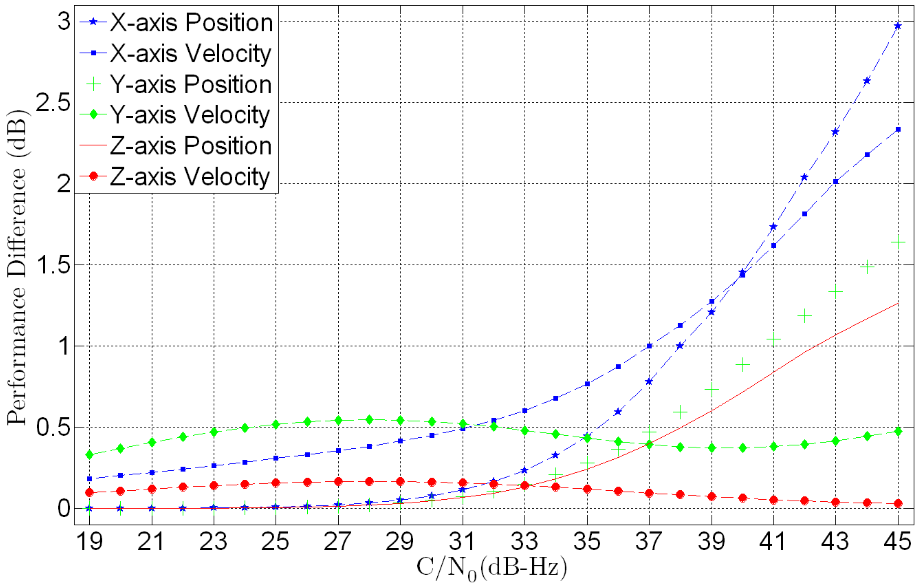

Figure 8.16: Differences in the Position and Velocity Performance of the Coupled and Decoupled Channel Filters, σ^2=50 m^2/s^2

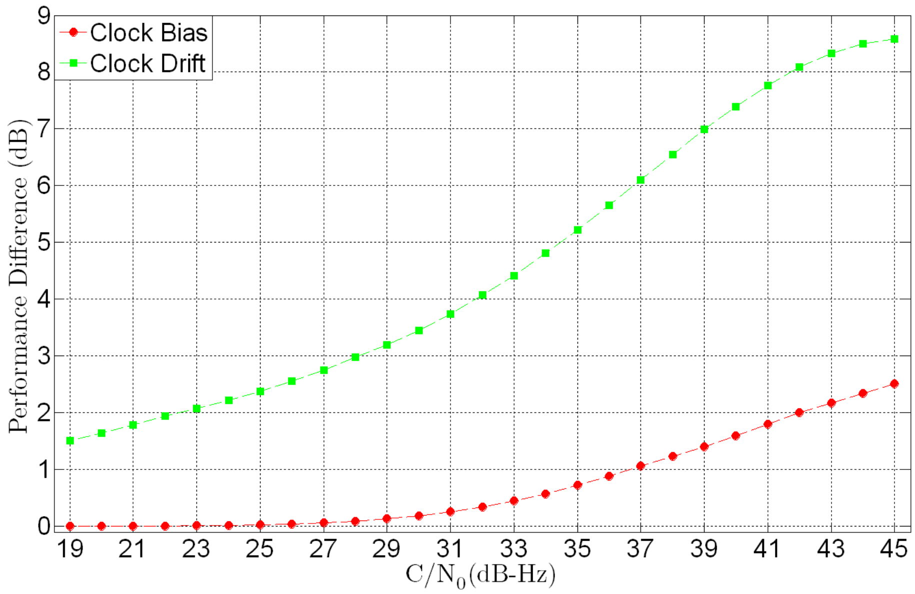

Figure 8.17: Differences in the Clock Bias and Drift Performance of the Coupled and Decoupled Channel Filters, σ^2=50 m^2/s^2

8.6 Monte Carlo Test Results of Vector Tracking Architectures

Monte Carlo simulations of the various scenarios in the previous section were carried out to verify the results of the covariance analysis. For each satellite constellation, simulations were performed with each of the three process noise variances. For each process noise variance, Monte Carlo simulations were performed over a range of C/N_0 ratios. The range of values for the C/N_0 ratio started at 19 dB-Hz and where performed at 2 dB intervals up to 45 dB-Hz. At each C/N_0 ratio, fifty Monte Carlo simulations were performed and all the simulations had a duration of one hour. For each individual Monte Carlo simulation number, all three algorithms used the same seed in the random number generator. Additionally, all three algorithms were initialized with the same errors at the start of a given Monte Carlo simulation.

The algorithms did not estimate the C/N_0 ratios of the simulated GPS signals. It was assumed that for all three algorithms the received signal amplitudes were known and that the variance of the thermal noise was also known. The runtime of the simulations was significantly decreased by not having the algorithm's estimate the C/N_0 ratios. In light of the findings presented in Chapter 7, Section 3.2 the difference in the performance of the algorithms should not be overly influenced by having to estimate the C/N_0 ratios.

The trajectory of the receiver in each Monte Carlo simulation was created by using the motion model in Equation (8.36). The statistics of the process noise terms in (8.36) were dictated by the receiver oscillator characteristics and the process noise intensity for the given scenario. As in the covariance analysis, noise statistics commensurate with TCXO oscillator were used (8.38).

The received GPS signals were simulated at the correlator output level. The various correlator outputs are modeled as shown in (8.45) [Sayre, 2003].

$$
\begin{aligned}
IE(k) &= AR(\epsilon+\tau)D(k)\cos(\pi f_{err}T+\theta_{err})+\eta_{IE}(k) \\
IP(k) &= AR(\epsilon)D(k)\cos(\pi f_{err}T+\theta_{err})+\eta_{IP}(k) \\
IL(k) &= AR(\epsilon-\tau)D(k)\cos(\pi f_{err}T+\theta_{err})+\eta_{IL}(k) \\
QE(k) &= AR(\epsilon+\tau)D(k)\sin(\pi f_{err}T+\theta_{err})+\eta_{QE}(k) \\
QP(k) &= AR(\epsilon)D(k)\sin(\pi f_{err}T+\theta_{err})+\eta_{QP}(k) \\
QL(k) &= AR(\epsilon-\tau)D(k)\sin(\pi f_{err}T+\theta_{err})+\eta_{QL}(k) \\
A &= \sqrt{\frac{2CT_{PDI}}{N_o}}\cdot\frac{\sin(\pi f_{err}T)}{(\pi f_{err}T)} \\
\eta &= N(0,1)
\end{aligned}
\tag{8.45}
$$

In (8.45), the correlator outputs have been normalized so that the noise is unit variance. The variables f_{err} and θ_{err} are the frequency error (Hertz) and phase error (radians) between the received and locally generated carrier signals, respectively. The sinc (sin(x)/x) term in (8.45) is an effective loss in signal amplitude due to the frequency error, f_{err}. The term ϵ is the phase error between the received and locally generated PRN codes, measured in fractions of a chip. The function $R(\epsilon)$ is the auto-correlation function of the PRN sequences. The symbol τ is the correlator spacing of the Early and Late PRN replicas, expressed in fractions of a chip. A correlator spacing of one half chip is used for all the Monte Carlo simulation results. The autocorrelation of the Gold code as a function of the code phase error is:

$$R(\epsilon) = \begin{cases} 1 - |\epsilon| & \text{when } \epsilon < 1 \text{ chip} \\ 0 & \text{when } \epsilon > 1 \text{ chip.} \end{cases}$$

The noise terms in (8.45) are all zero-mean and unit variance. The in-phase and quadrature noise terms are uncorrelated. Using a half chip correlator spacing results in the early and late noise terms being uncorrelated. The early and prompt noise terms, and the prompt and late noise terms, all have a correlation coefficient of 0.5 .

The pseudorange and pseudorange-rate discriminator functions in (8.40) and (8.42) are used in the simulations. An integrate and dump time of 20 milliseconds is used for the correlator outputs in (8.40). Two consecutive integrate and dump times of 10 ms are used for the correlator outputs used in (8.42).

At the start of each simulation, the steady state, state covariance matrix of the master filter is calculated. The state vector of the master filter for the different architectures is then initialized using a Gaussian random vector with a mean vector of zero and a covariance matrix equal to the steady state, state covariance matrix. Throughout the simulations, the algorithms were monitored for loss of lock. A pseudorange error of over 150 meters in any channel was used as the criteria for declaring loss of lock.

8.6.1 Monte Carlo Simulation Results for Eleven Satellite Scenario

The results of the Monte Carlo simulations for the eleven satellite geometry and 2 m^2/s^2 process noise levels are shown in Figures 8.18 through 8.21. A logarithmic scale is used because the Monte Carlo simulation results vary widely in their magnitudes. In Figure 8.18, the average of the variances from the Monte Carlo simulations for the x-axis position errors for all three architectures is shown. The average of the variances is computed from the formula shown in (8.46).

$$\bar{\sigma}^2_{x,C/N_0} = \frac{1}{50}\sum_{n=1}^{50}\tilde{\sigma}^2_{x,C/N_0}(n) \tag{8.46}$$

The x-axis position variance from each individual trial at a given C/N_0 ratio is denoted by $\tilde{\sigma}^2_{x,C/N_0}(n)$ and the average of the variances at that C/N_0 ratio is represented by $\bar{\sigma}^2_{x,C/N_0}$. Recall that the errors are calculated after the measurement update of the master filters for the federated algorithms. The errors are computed at the same times for the centralized filter.

In addition to the mean of the x-axis position errors, the variances predicted from the covariance analyses are plotted for the centralized filter and the federated filter with uncoupled channels. The results of the covariance analysis for the federated filter with coupled residual filters are not shown because they are virtually identical to the results for the centralized filter. The mean of the variances $\bar{\sigma}^2_{x,C/N_0}$ plus and minus three times the empirical standard deviation of the Monte Carlo simulation results is also plotted. This is done simply to relay a sense of how spread out the Monte Carlo simulation results are. The empirical standard deviation is calculated for each C/N_0 ratio from (8.47).

$$\sigma = \sqrt{\frac{1}{49}\sum_{n=1}^{50}(\tilde{\sigma}^2_{x,C/N_0}(n) - \bar{\sigma}^2_{x,C/N_0})^2} \tag{8.47}$$

The empirical standard deviation σ is added and subtracted from $\bar{\sigma}^2_{x,C/N_0}$ to yield the bounds shown in Figures 8.18 through 8.21. The addition and subtraction is performed prior to the conversion to the logarithmic scale.

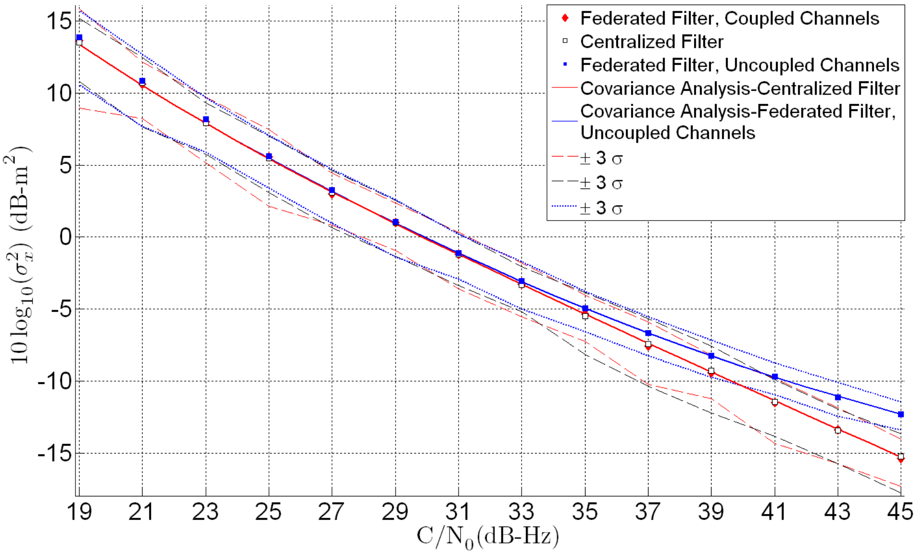

Figure 8.18: Monte Carlo Simulation Results for the X-Axis Position with Eleven Satellites and σ^2=2 m^2/s^2

Examining Figures 8.18 and 8.19, the results of the Monte Carlo simulations for x-axis position and velocity variances match closely with the predictions from the covariance analysis. As seen previously, the performance advantage of the centralized filter and coupled channel federated filter diminishes as the C/N_0 ratio declines.

The variance of the clock bias and drift states are shown in Figures 8.20 and 8.21, respectively. Again, the results of the Monte Carlo simulations closely match the predictions from the covariance analysis. The relative increase in performance of the centralized filter and federated filter with coupled channels decreases with the C/N_0 ratio.

The results of the Monte Carlo simulations shown in Figures 8.18 through 8.21 are representative of the simulation results for the other filter states. For the sake of brevity, they are not shown here. However, they are included in Appendix C. The Monte Carlo simulation results for process noise intensity of 10 m^2/s^2 are also included in Appendix C.

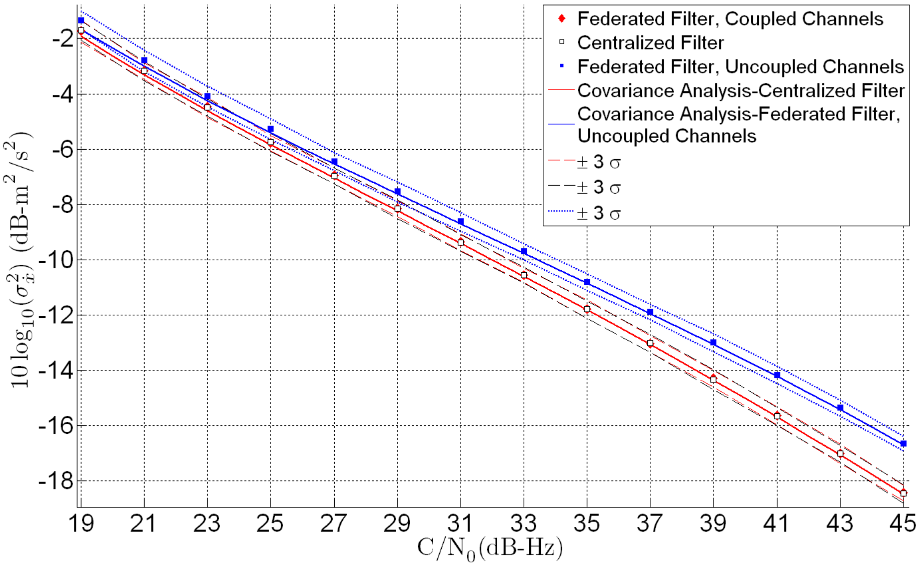

Figure 8.19: Monte Carlo Simulation Results for the X-Axis Velocity with Eleven Satellites and σ^2=2 m^2/s^2

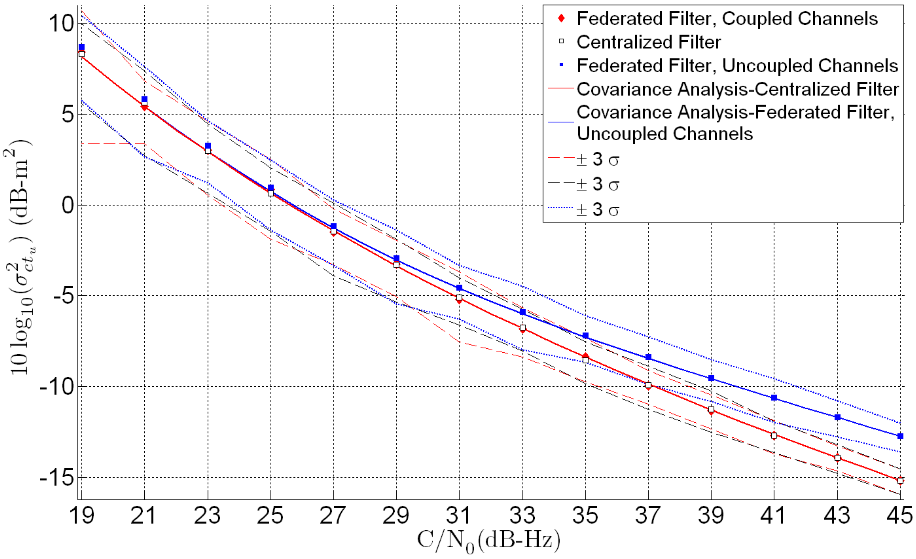

Figure 8.20: Monte Carlo Simulation Results for the Clock Bias with Eleven Satellites and σ^2=2 m^2/s^2

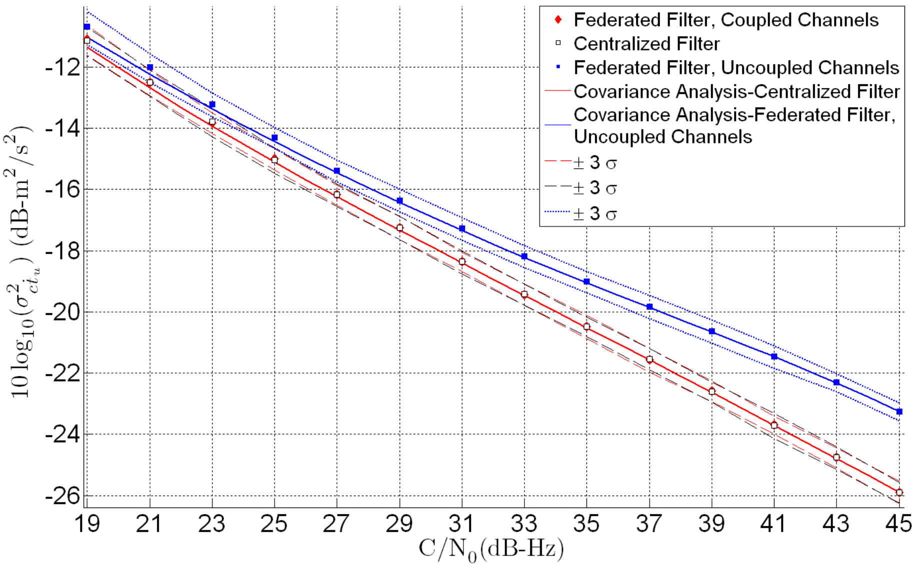

Figure 8.21: Monte Carlo Simulation Results for the Clock Drift with Eleven Satellites and σ^2=2 m^2/s^2

For the eleven satellite constellation, the simulation results for the 2 and 10 m^2/s^2 process noise levels are congruent with the predictions from the covariance analysis. However, the Monte Carlo simulation results for the 50 m^2/s^2 process noise level do not match the covariance analysis predictions nearly as well. The simulation results for the 50 m^2/s^2 process noise level are shown in Figures 8.22 through 8.25. Only the results for C/N_0 ratios above 25 dB-Hz are shown. This is because the algorithms began to fail at C/N_0 ratios below 27 dB-Hz.

The results for the x-axis velocity and clock drift display the largest deviations from the predictions of the covariance analysis. These deviations are caused predominantly by nonlinearities in the pseudorange-rate discriminators. The process noise driving the velocity states is large enough to induce pseudorange-rate errors that exceed the linear operating range of the pseudorange-rate discriminator. If the pseudorange-rate discriminators are bypassed, and the filter is provided the true pseudorange-rate error plus white noise with the statistics given in (8.42), the simulation results match the covariance analysis.

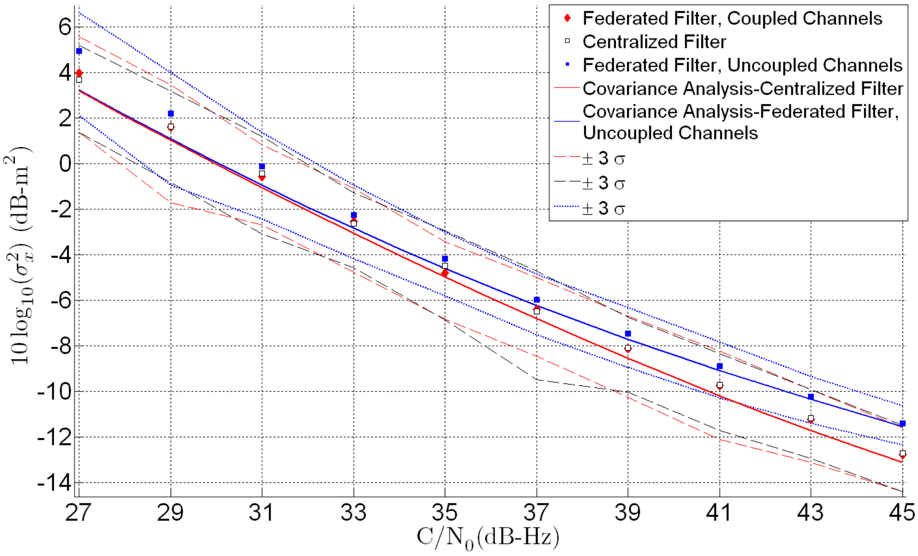

Figure 8.22: Monte Carlo Simulation Results for the X-Axis Position with Eleven Satellites and σ^2=50 m^2/s^2

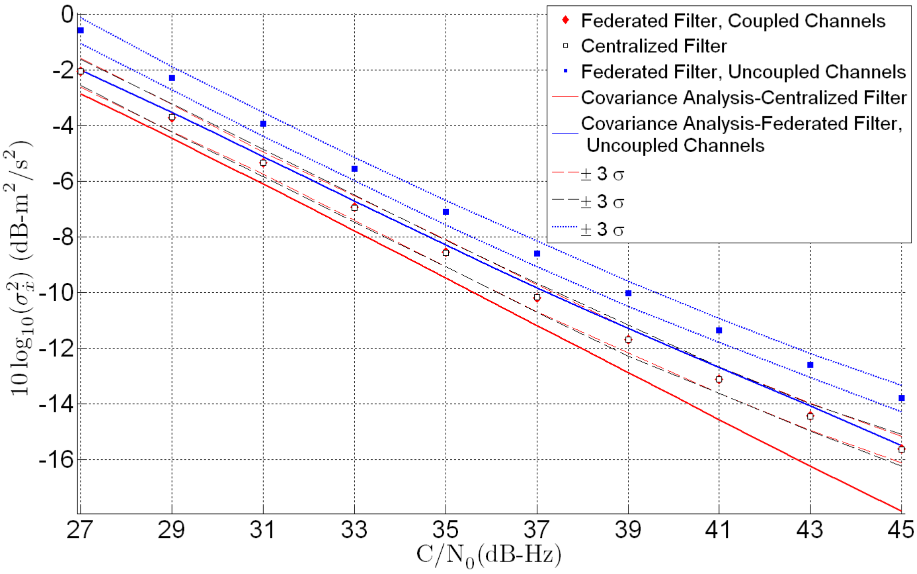

Figure 8.23: Monte Carlo Simulation Results for the X-Axis Velocity with Eleven Satellites and σ^2=50 m^2/s^2

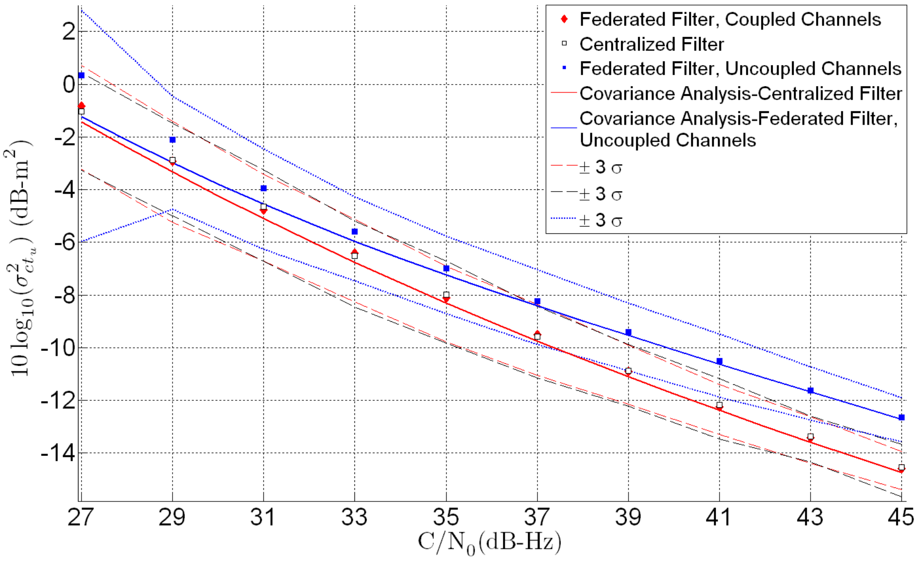

Figure 8.24: Monte Carlo Simulation Results for the Clock Bias with Eleven Satellites and σ^2=50 m^2/s^2

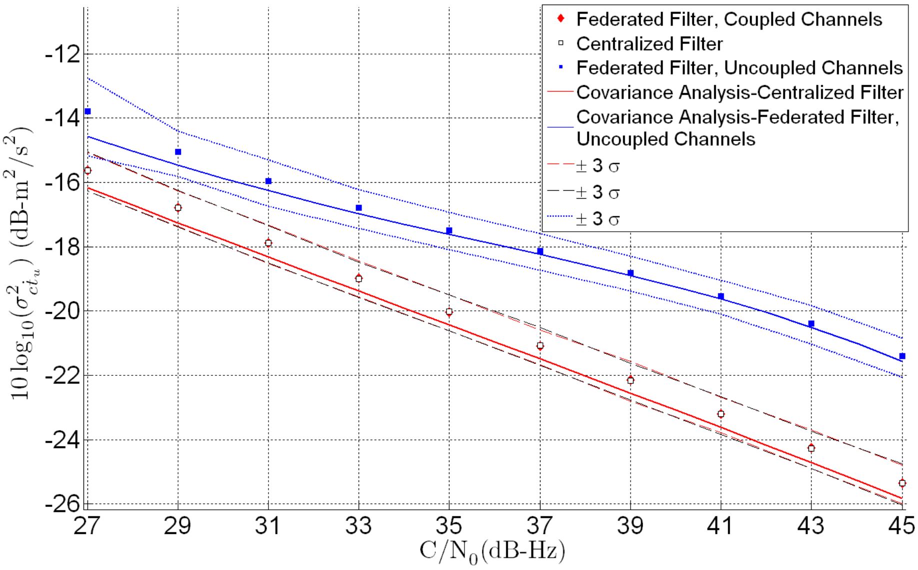

Figure 8.25: Monte Carlo Simulation Results for the Clock Drift with Eleven Satellites and σ^2=50 m^2/s^2

8.6.2 Monte Carlo Simulation Results for Five Satellite Scenario

The x-axis position and velocity Monte Carlo simulation results for the five satellite scenario with process noise levels of 2 m^2/s^2 are shown in Figures 8.26 and 8.27, respectively. The results of the simulations match the covariance analysis results. The results for the other states are included in Appendix C.

The x-axis position and velocity Monte Carlo simulation results for the five satellite scenario with process noise levels of 50 m^2/s^2 are shown in Figures 8.28 and 8.29, respectively. Only the results for C/N_0 ratios of 31 dB-Hz and above are shown. The algorithms began to lose lock below 31 dB-Hz. The position results match the covariance analysis far better than the velocity results. As in the eleven satellite constellation, the linear operating

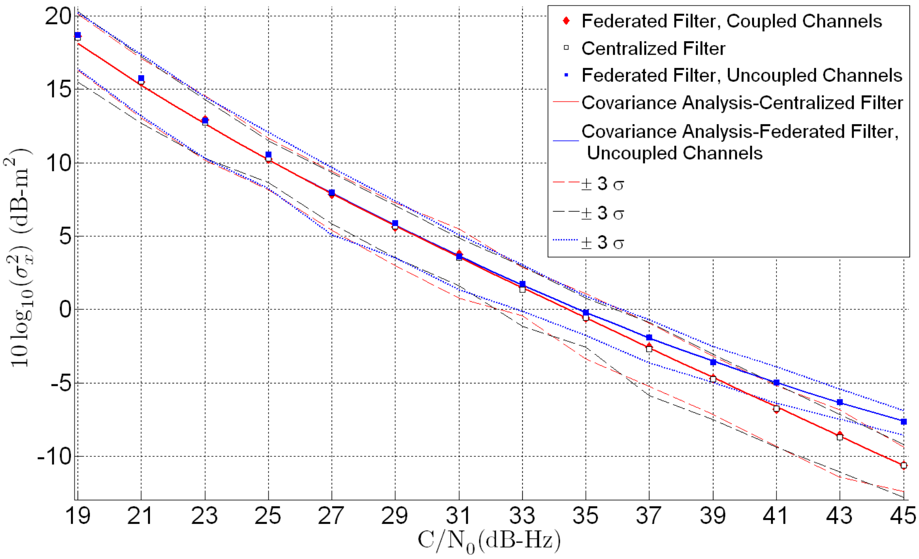

Figure 8.26: Monte Carlo Simulation Results for the X-Axis Position with Five Satellites and σ^2=2 m^2/s^2

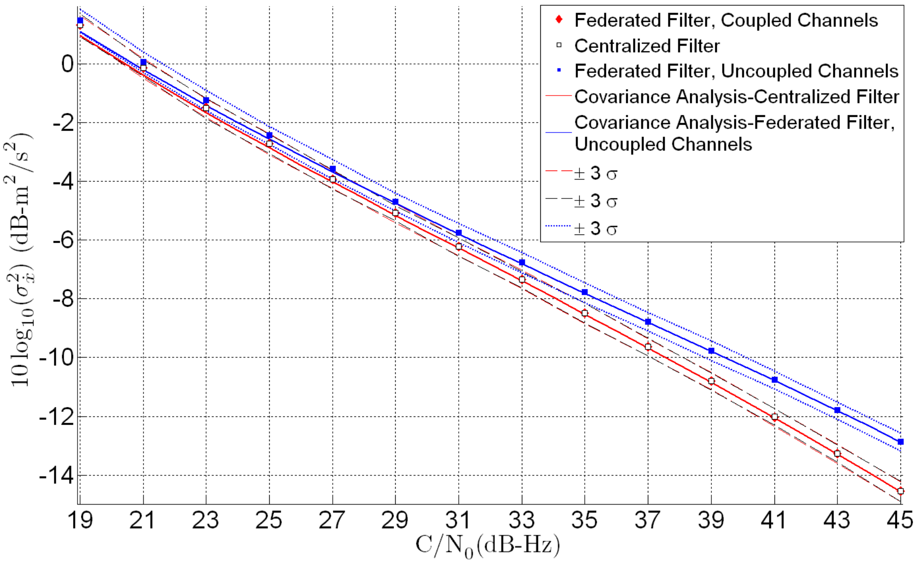

Figure 8.27: Monte Carlo Simulation Results for the X-Axis Velocity with Five Satellites and σ^2=2 m^2/s^2

range of the pseudorange-rate discriminators is exceeded with the 50 m^2/s^2 process noise levels. The simulation results for the other states are included in Appendix C.

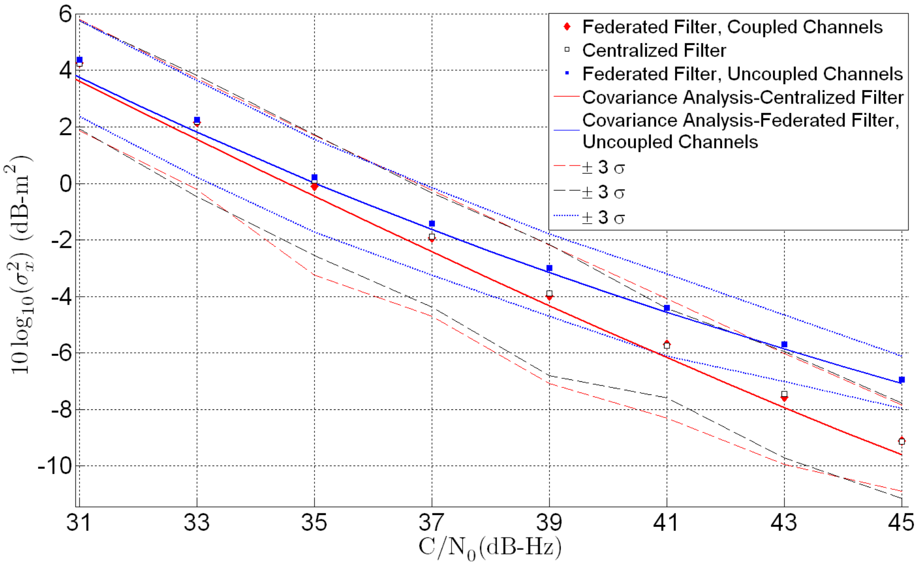

Figure 8.28: Monte Carlo Simulation Results for the X-Axis Position with Five Satellites and σ^2=50 m^2/s^2

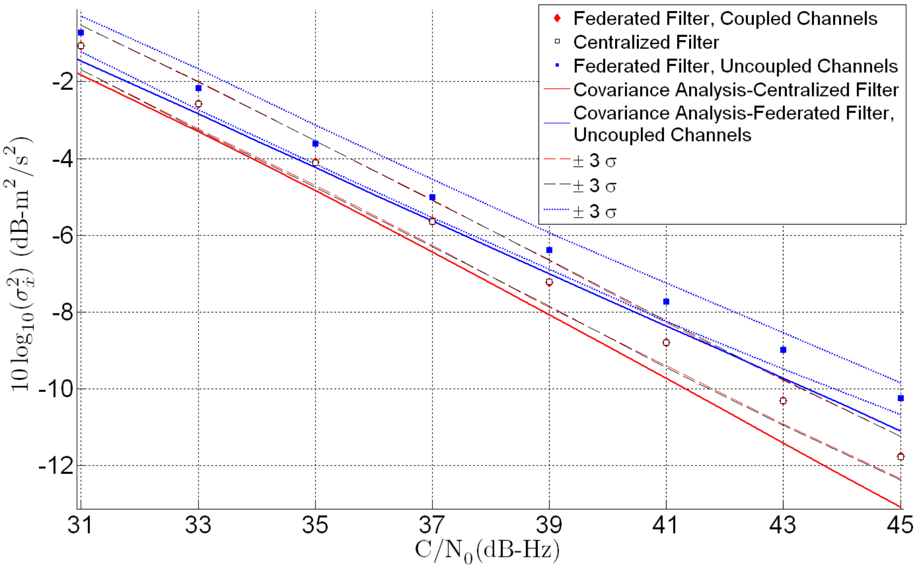

Figure 8.29: Monte Carlo Simulation Results for the X-Axis Velocity with Five Satellites and σ^2=50 m^2/s^2

8.7 Conclusion

The performance of three different vector tracking architectures is studied in this chapter. The differences in the architectures are in how the Kalman filters are designed in each of them. One algorithm uses a single, central filter to process the received satellite signals and estimate the receiver's navigation states. The other two algorithms use a federated filtering architecture. The two federated architectures use a master filter to estimate the receiver's navigation states and a series of residual filters to process the received GPS signals. The distinction between the two federated architectures is in the design of the residual filters. One algorithm allows the residual filters to share information between channels and the other does not. Otherwise, the residual filters are equivalent. The three algorithms are used to study the effects of using a federated architecture on vector tracking's performance. Furthermore, the effect of using independently operating residual filters is also studied.

The covariance of the position, velocity, and clock state estimates from the three algorithms are compared over a range of C/N_0 ratios. The covariance analysis reveals that the centralized filter and the federated filtering architecture with coupled channels exhibt the same performance. The federated filtering architecture with uncoupled channels performs worse by comparison. However, the disparity in the performance of the three algorithms decreases as the C/N_0 ratio decreases. At low C/N_0 ratios, the three algorithms provide essentially equivalent performance. Monte Carlo simulations of the algorithms are then used to verify the covariance analysis.

This chapter focuses on stand-alone GPS vector tracking algorithms. The next chapter takes the three vector tracking architectures and studies their performance when an IMU is added. The same combination of covariance analysis and Monte Carlo simulation is used to study the performance of the GPS/INS vector tracking algorithms.

Chapter 9

Analysis of Different Deep Integration Architectures

9.1 Introduction

This chapter examines the performance of several different Deeply Integrated (DI) architectures. The performance of three DI architectures is investigated. The three DI algorithms are differentiated by how the Kalman filter is implemented. One algorithm uses a single central filter to process all IMU and GPS measurements. The other two DI algorithms use a federated Kalman filter. The two federated filters are separated by how the GPS measurements are processed. One uses vector tracking and the other uses scalar tracking techniques.

The purpose of this chapter is to determine the difference in the performances of the three algorithms. The only difference between the algorithms is in the design of the Kalman filters. This allows the performance degradation caused by using federated filtering and scalar tracking to be evaluated. Covariance analyses are performed for varying grades of IMU's and with varying satellite constellations. Monte Carlo simulations of the different architectures are performed to verify the covariance analyses.

9.2 Deep Integration Architectures

In Chapter 1, several different deeply integrated architectures were discussed. The majority of DI algorithms can be divided into two separate categories. The delineation between the two categories is in the design of the Kalman filter. One group of algorithms use a single filter to process all the GPS and IMU measurements. No intermediate tracking loops or filtering is applied to the measurements prior to the central filter. In the other category are algorithms that use some sort of federated filtering architecture. In the federated architecture, multiple residual filters process the raw GPS measurements at high update rates (e.g. 50 Hz). A large master filter is then updated at a much lower rate (e.g. 1 Hz). The master filter uses the states of the residual filters as measurements and estimates the position, velocity, and attitude states of the receiver.

Two major issues arise when using the federated architecture. The first issue is the performance of the master filter in the federated architecture relative to that of the filter in the centralized approach. The use of a federated architecture might entail a loss in performance because the master filter does not use the raw measurements. Using the states of the residual filters as measurements, and not the actual raw data, could possibly result in a loss of information. Second, if the residual filters function independently then they are effectively operating as scalar tracking loops. Recall from Chapter 6 that the pseudorange-state formulation of vector tracking can be converted to scalar tracking by eliminating the information sharing between channels. When the pseudorange-state filter is reduced to scalar tracking, there is a loss in its performance. Therefore, when residual filters are used in a DI architecture and they are not coupled together, there must be some loss in performance of the overall algorithm. The extent of this loss in performance is studied in this chapter.

Chapter 1 also includes a discussion about the difference in performance of tight coupling and deep integration. When a federated DI architecture is used the residual filters function essentially as scalar tracking loops. The distinction between federated deep integration and tight coupling with Doppler-aiding becomes subjective and unclear.

In this chapter, the previously mentioned issues are investigated by studying the performance of three deeply integrated algorithms. The three DI algorithms are used to study the loss in performance caused by going from a centralized architecture to a federated architecture. One of the DI algorithm uses a single centralized filter while the other two use a federated architecture. Furthermore, the loss in performance in the federated architectures incurred by not sharing information between the residual filters is also investigated. The two federated DI architectures are separated by how the residual filters operate. One algorithm uses a residual filter design where information is shared between the different channels of the receiver. The other algorithm uses a residual filter design where information is not shared between channels. Other than the design of the residual filters, the two federated DI algorithms are identical. The residual filter designs are based off the position and pseudorange-state formulations of vector tracking described in Chapter 6.

Recall from Chapter 6 that the position and pseudorange-state formulations of vector tracking are equivalent. However, the pseudorange-state filter can be reverted back to scalar tracking by decoupling the channels (i.e. eliminating information sharing between channels). The position-state filter and the decoupled pseudorange-state filter are used to isolate the effects of information sharing between the channels. The performances of the algorithms are compared by using covariance analysis. Varying grades of IMU's and different satellite constellations are also used. The results of the covariance analyses are then verified by Monte Carlo simulations.

The rest of this chapter is organized as follows. The next section describes the different deeply integrated algorithms in detail. Then, the covariance analysis and its results are discussed in the subsequent section. Finally, the results of the Monte Carlo simulations are presented and compared to the covariance analyses.

9.2.1 Deep Integration with Centralized Filter

As its name implies, the centralized DI filter uses a single central filter to perform all the tasks of state estimation and signal tracking. Figure 9.1 shows a block diagram of the centralized DI architecture. As shown in Figure 9.1, an integrate and dump time of 20 ms is used by the filter. Consequently, pseudoranges and pseudorange-rates are predicted by the central filter at 50 Hz. Pseudorange and pseudorange-rate residuals are supplied at the end of each integrate and dump period. These residuals are used by the filter to correct its state estimates. The updated states are then used to predict the next set of pseudoranges and pseudorange-rates. Measurements from the IMU arrive at the filter at a rate of 100 Hz.

The states of the Extended Kalman filter (EKF) in the centralized architecture are the receiver's position, velocity, attitude, and clock states and the IMU biases. The Earth-Centered Earth-Fixed (ECEF) coordinate frame is used to track the receiver's position and velocity. The state vector of the filter is shown in (9.1).

$$\begin{bmatrix} P_{3\times 1} \\ V_{3\times 1} \\ A_{3\times 1} \\ B^{accel}_{3\times 1} \\ B^{gyro}_{3\times 1} \\ B^{clk}_{2\times 1} \end{bmatrix} = \begin{bmatrix} \text{Position States} \\ \text{Velocity States} \\ \text{Attitude States} \\ \text{Accelerometer Bias States} \\ \text{Gyro Bias States} \\ \text{Clock Bias and Drift States} \end{bmatrix} \tag{9.1}$$

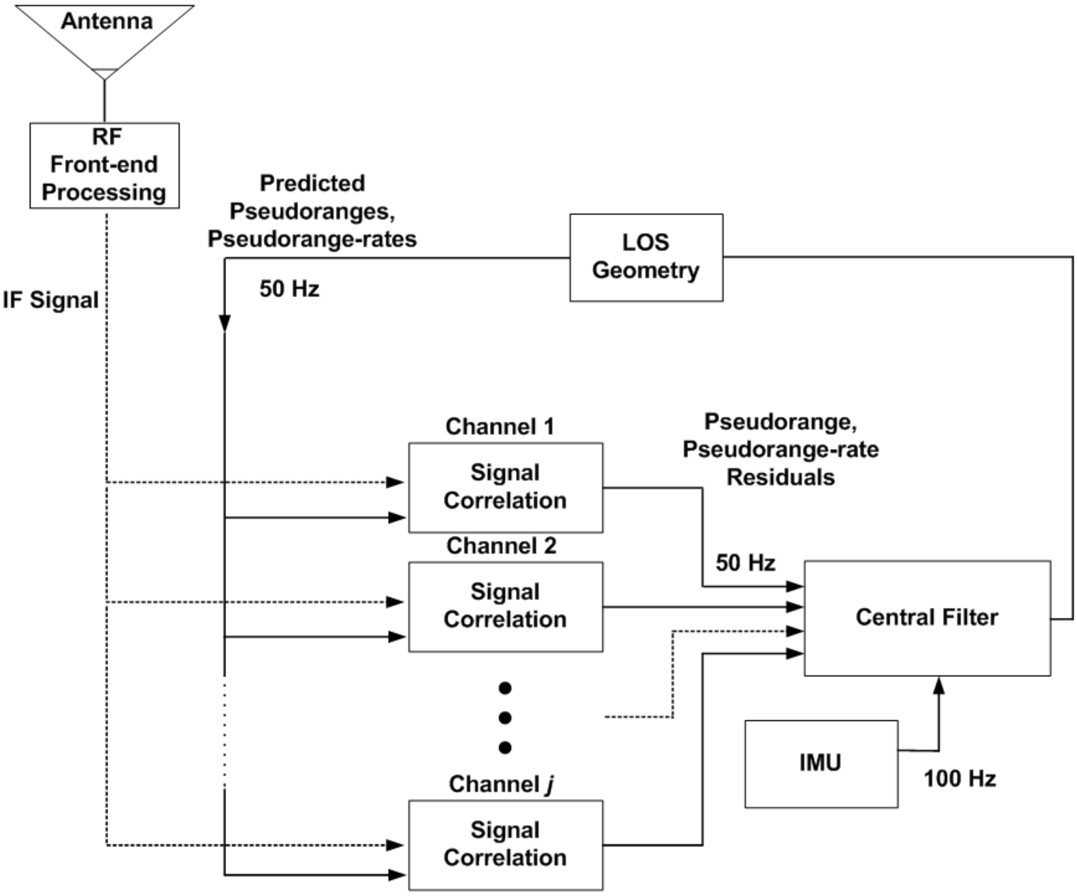

Figure 9.1: Centralized Deep Integration Filter

The stochastic errors in the IMU outputs are modeled as consisting of two terms. One is a white, Gaussian noise error term and the other is a 1^{st}-order Gauss-Markov process used to model a slowly moving bias. The intensities of the two processes and the time constant of the moving bias are determined by the grade of IMU in use. The details of how the IMU noise sources are modeled are described in detail in Appendix E.

The state dynamics and measurement update of the filter are nonlinear. The dynamics and measurement update equations of the filter are linearized about the current state estimates of the filter. The true state vector $x(k)$, the estimates of the filter $\hat{x}(k)$, and the errors in the states of the filter $\delta x(k)$ at time k are related to each other through (9.2).

$$x(k) \quad = \quad \hat{x}(k) + \delta x(k) \tag{9.2}$$

In general, the states of the system at time $k + 1$ are related to the state vector, IMU input vector $u(k)$, and IMU and clock noise terms $w(k)$ at time k by a nonlinear function:

$$x(k+1) \quad = \quad f(x(k), u(k), w(k)) \tag{9.3}$$

The details of the nonlinear function $f(x(k), u(k), w(k))$ are given in Appendix E. The input vector $u(k)$ represents the uncorrupted measurements from the IMU and the noise vector $w(k)$ models the noise processes in the IMU and receiver clock. The state estimates of the filter are propagated forward by using (9.4).

$$\hat{x}(k+1) \quad = \quad f(\hat{x}(k), u(k), 0) \tag{9.4}$$

The relationship between the errors in the filter's estimates at time k, the filter's estimates, and noise terms are approximated by (9.5).

$$\begin{aligned} x(k+1) &= f(\hat{x}(k) + \delta x(k), u(k), w(k)) \\ x(k+1) &\approx f(\hat{x}(k), u(k), 0) + A \cdot \delta x(k) + B \cdot w(k) \\ &\text{where} \\ A &= \left.\frac{\partial f(\cdot)}{\partial x}\right|_{x=\hat{x}(k)} \\ B &= \left.\frac{\partial f(\cdot)}{\partial w}\right|_{x=\hat{x}(k)} \end{aligned} \tag{9.5}$$

The errors in the filter's estimates at time $(k+1)$ are given in (9.6).

$$\begin{aligned} \delta x(k+1) &\equiv x(k+1) - \hat{x}(k+1) \\ \delta x(k+1) &\equiv x(k+1) - f(\hat{x}(k), u(k), 0) \end{aligned} \tag{9.6}$$

Using the linearization in (9.5), the mean and covariance of the state errors are approximated as shown in (9.7).

$$\begin{aligned} E\{\delta x(k+1)\} &= 0_{N\times 1} \\ E\{\delta x(k+1)\delta x^T(k+1)\} &\approx AP(k)A^T + BQ(k)B^T \end{aligned} \tag{9.7}$$

In (9.7), $P(k)$ is the covariance matrix of the state errors at time k and $Q(k)$ is the covariance matrix of the process noise vector $w(k)$. The relations in (9.7) assume that the errors $\delta x(k)$ are zero mean.

The states of the filter are updated with the residuals from each channel. Two different discriminator functions are used to process the correlator outputs and produce measurements of the pseudorange and pseudorange-rate residuals. The relationship between the states of the filter and the pseudorange and pseudorange-rate residuals is shown in (9.8).

$$y(k) = C\delta(x) + v \tag{9.8}$$

where

$$C = \begin{bmatrix} a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 0_{1\times 9} & 1 & 0 \\ 0 & a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0_{1\times 9} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 0_{1\times 9} & 1 & 0 \\ 0 & a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0_{1\times 9} & 0 & 1 \end{bmatrix}$$

In (9.8), the variables $a_{x,N}, a_{y,N}, a_{z,N}$ are the components of the line-of-sight unit vector from the receiver to the N-th satellite. The statistics for the measurement noise vector v are shown in (9.9).

$$\begin{aligned} E\{v\} &= 0_{N\times 1} \\ E\{vv^T\} &= R_v \end{aligned} \tag{9.9}$$

The matrix R_v is a diagonal matrix because the noise in the different channels is uncorrelated and the noise in the pseudorange and pseudorange-rate measurements are uncorrelated. The individual entries in R_v are determined from the type of discriminators that are used and the current C/N_0 ratio.

9.2.2 Federated Filtering Architectures

The federated filtering architectures use a combination of a master filter and multiple smaller residual filters. The general architecture is based on the description given in [Abbott and Lillo, 2003]. The states of the master filter are used to predict the received pseudoranges and pseudorange-rates over a one second period. Pseudorange and pseudorange-rate residuals are produced in each channel of the receiver every 20 ms. The residual filters estimate the errors in the predictions of the master filter using the pseudorange and pseudorange-rate residuals. The estimates of the residual filters are fed back to correct the errors in the master filter's predictions. The states of the residual filter are then used as measurements by the master filter at the end of the one second period. After being used as measurements, the states of the residual filter are zeroed and the cycle is repeated.

Two different federated architectures are examined in this dissertation. The two algorithms are separated by how the residual filters operate. The first uses a residual filter with coupled channels. This means that information is shared between the different channels of the receiver. The second architecture uses a residual filter without coupled channels. Without the channels being coupled, no information is shared between the different channels. The pseudorange and position-state formulations of vector tracking from Chapter 6 are used to design the residual filters. This is done to ensure that the only difference between the two federated architectures is the information sharing among the residual filters. The details of the federated architectures are given in the following sections.

9.2.3 Federated Deep Integration Filter with Coupled Channels

Figure 9.2 shows a block diagram of the federated architecture when using a coupled channel residual filter. The master filter is updated at 1 Hz and its states are used to predict the pseudoranges and pseudorange-rates between measurement epochs. Between measurement epochs of the master filter, the residual filter processes the pseudorange and pseudorange-rate residuals produced by the integrate and dump operations at 50 Hz. The IMU measurements are produced at a rate of 100 Hz. The master filter processes the IMU measurements at 100 Hz and uses its current best estimates of the IMU bias states to correct the measurements. The master filter effectively dead-reckons between measurement updates. The estimates of the residual filter (in accordance with the Line Of Sight (LOS) geometry) are routed back to correct the master filter's predicted pseudoranges and pseudorange-rates. The feedback of the residual filter's estimates is optional, since the general filter architecture in Figure 9.2 can be operated without feedback. However, the discriminators used to transform the correlator outputs into pseudorange and pseudorange-rate residuals have a limited operating range. Therefore, the discriminators will produce erroneous measurements if the true residuals extend beyond their operating range. The operating point of the discriminators is maintained at approximately zero by feeding back the states of the residual filter.

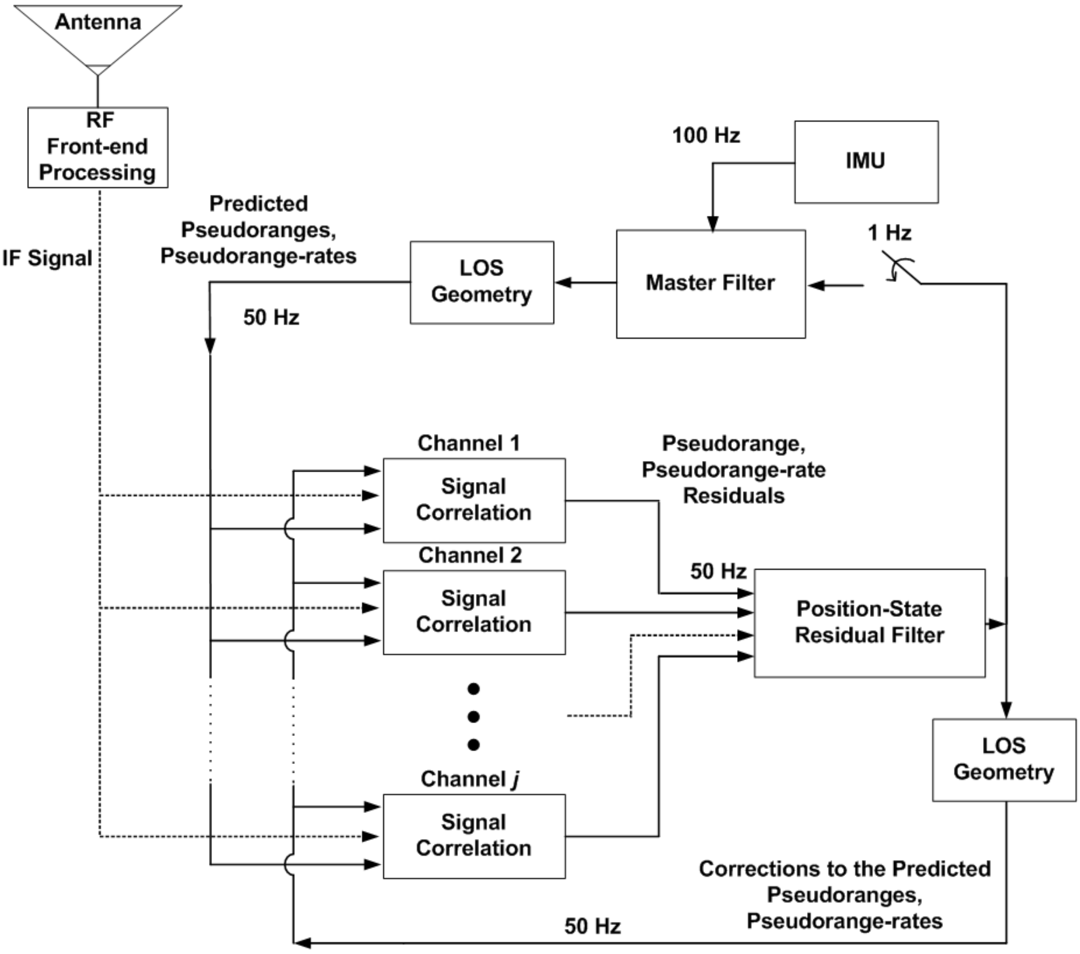

Figure 9.2: Federated Deep Integration Filter with Coupled Channels

The states of the residual filter used in the coupled channel architecture are shown in (9.10).

$$\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ ct \\ c\dot{t} \end{bmatrix} = \begin{bmatrix} \text{Position Error, X-axis} \\ \text{Velocity Error, X-axis} \\ \text{Position Error, Y-axis} \\ \text{Velocity Error, Y-axis} \\ \text{Position Error, Z-axis} \\ \text{Velocity Error, Z-axis} \\ \text{Clock Bias Error} \\ \text{Clock Drift Error} \end{bmatrix} \tag{9.10}$$

The residual filter tracks errors in the master filter's estimates of the receiver's position, velocity, and clock states. The IMU bias states and attitude states are not included in the residual filter state vector. The position-state formulation of the vector tracking Kalman filter is used due to the numerical issues that are encountered when using the pseudorange-state formulation with coupled channels. The discrete time state dynamics of the filter are given by the nonlinear function f_R:

$$x_R(k+1) \quad = \quad f_R(x_R(k), x^*(k), u(k), w(k)) \tag{9.11}$$

In (9.11), function f_R is the relation between the current states of the residual filter x_R, the errors in the states of the master filter that are not included in the residual filter x^*, the uncorrupted IMU measurements u, and the noise terms w present in the IMU and receiver clock. This function is very similar to that given in (9.3). The difference is that f_R

only includes the propagation of the states included in the residual filter state vector (i.e. the position, velocity, and clock states). The subscript R is used to distinguish the states of the residual filter from the states of the master filter. The vector w_k represents the noise sources in the IMU measurements and receiver clock. The vector $\delta x^*(k)$ represents errors in the states of the master filter, but not those states included in the state vector of the residual filter. The vector $\delta x^*(k)$ includes the errors in the master filters estimates of the receivers attitude and IMU biases. The statistics of the errors in the states of the filter are propagated by linearizing the nonlinear equation in (9.11):

$$\begin{aligned} x_R(k+1) &= f_R(x_R(k), x^*(k), u(k), w(k)) \\ x_R(k+1) &\approx f_R(\hat{x}_R(k), 0, u(k), 0) + A_R \cdot \delta x_R(k) + G_R \cdot x^*(k) + B_R \cdot w(k) \end{aligned} \tag{9.12}$$

where

$$\begin{aligned} A_R &= \left.\frac{\partial f_R(\cdot)}{\partial x_R}\right|_{x=\hat{x}(k)} \\ G_R &= \left.\frac{\partial f_R(\cdot)}{\partial x^*}\right|_{x=\hat{x}(k)} \\ B_R &= \left.\frac{\partial f_R(\cdot)}{\partial w}\right|_{x=\hat{x}(k)} \end{aligned}$$

The states of the residual filter are propagated forward in time using (9.13).

$$\begin{aligned} x_R(k+1) &= f_R(x_R(k), x^*(k), u(k), w(k)) \\ \hat{x}_R(k+1) &= f_R(\hat{x}_R(k), 0, u(k), 0) \end{aligned} \tag{9.13}$$

The errors in the states of the residual filter are defined in (9.14).

$$\begin{aligned} \delta x_R(k+1) &\equiv x_R(k+1) - \hat{x}_R^-(k+1) \\ \delta x_R(k+1) &\equiv x_R(k+1) - f_R(\hat{x}_R(k), 0, u(k), 0) \end{aligned} \tag{9.14}$$

Using (9.12) and (9.14), the statistics of the residual filter's estimates after the state propagation are approximated by (9.15).

$$\begin{aligned} E\{\delta x_R(k+1)\} &\approx 0_{8\times 1} \\ E\{\delta x_R(k+1)\delta x_R^T(k+1)\} &\approx A_R P_R(k) A_R^T + G_R E\{\delta x^*(k)\delta x_R^T(k)\} A_R^T \\ &\quad + B_R Q(k) B_R^T + A_R E\{\delta x_R(k)\delta x^{*T}(k)\} G_R^T \\ &\quad + G_R E\{\delta x^*(k)\delta x^{*T}(k)\} G_R^T \end{aligned} \tag{9.15}$$

Equation (9.15) requires knowledge of the covariance between the residual filter's state errors δx_R and the state errors in the master filter filters estimates δx^*. These values are maintained in the covariance matrix of the master filter and are used in the residual filter propagation.

The relationship between the states of the residual filter and the pseudorange and pseudorange-rate residuals supplied at time k is shown in (9.16).

$$\epsilon_R(k) = C_R(x_R(k) - \hat{x}_R^-(k)) + v(k) \tag{9.16}$$

where

$$C_R = \begin{bmatrix} a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 & 0 \\ 0 & a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 & 0 \\ 0 & a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 \end{bmatrix}$$

The superscripts $+$ and $-$ are used to denote the filter's estimates before and after the measurement update. The observation matrix C_R for the residual filter is similar to the observation matrix used by the centralized filter in the previous section. The state estimate of the residual filter after the measurement update is given in (9.17).

$$\hat{x}_R^+(k) = \hat{x}_R^-(k) + K_R(k)(\epsilon_R(k)) \tag{9.17}$$

The pseudorange and pseudorange-rate residuals ϵ_R generated by the discriminators are directly multiplied by the Kalman gain K_R. This is a consequence of the states of the residual filter being fed back to control the code and carrier NCOs. If the estimates of the residual filter were not fed back, the measurement update equation in (9.17) would be changed to (9.18).

$$\hat{x}_R^+(k) = \hat{x}_R^-(k) + K_R(k)(\epsilon_R(k) - C_R\hat{x}_R^-(k)) \tag{9.18}$$

The master filter in Figure 9.2 combines the IMU measurements and estimates of the residual filter in order to estimate the navigation states of the receiver. As a consequence, the master filter's state vector is partitioned into two segments. The first segment contains states that correspond to the receiver's position, velocity, attitude, IMU biases, and clock states. The second segment contains states that represent the errors in the residual filter's state estimates. Appendix D explains how the error states are derived. The full state vector is shown in (9.19).

$$\begin{bmatrix} x_M \\ \delta x_R \end{bmatrix} = \begin{bmatrix} \text{Master Filter's Navigation States} \\ \text{Residual Filter State Errors} \end{bmatrix} \tag{9.19}$$

where

$$x_M = \begin{bmatrix} \text{Position States} \\ \text{Velocity States} \\ \text{Attitude States} \\ \text{Accelerometer Bias States} \\ \text{Gyro Bias States} \\ \text{Clock Bias and Drift States} \end{bmatrix}$$

$$\delta x_R = \begin{bmatrix} \text{Residual Filter Error, State One} \\ \text{Residual Filter Error, State Two} \\ \vdots \\ \text{Residual Filter Error, State Eight} \end{bmatrix}$$

The pseudoranges and pseudorange-rates are predicted based of the states corresponding to the receiver's position, velocity, clock bias, and clock drift. The extra states in the master filter are appended to model the errors in the states of the residual filter. As shown in (9.19), each state in the residual filter has a corresponding error state in the master filter.

The error states are defined in (9.20).

$$\begin{aligned} \delta x_R^-(k) &= x_R(k) - \hat{x}_R^-(k) \\ \delta x_R^+(k) &= x_R(k) - \hat{x}_R^+(k) \end{aligned} \tag{9.20}$$

The error states correspond to the difference between the true residuals and the residual filter's estimates. Two update equations are used for the error states. The reason for this is that the residual filter is updated at the rate of the IMU. For this dissertation, a 100 Hz IMU measurement rate is used and a 50 Hz GPS measurement rate is used. One set of update equations are used for when an IMU measurement and a GPS measurement occur simultaneously. The other state update equations are used whenever only an IMU measurement is received. The dynamics of the error states for the simultaneous GPS and IMU measurements are shown in (9.21).

$$\delta x_R(k+1) = \Phi_{\delta x}\delta x_R(k) + \Gamma_w w(k) + \Gamma_* x^*(k-1) + \Gamma_v v(k+1) \tag{9.21}$$

where

$$\begin{aligned} \Phi_{\delta x} &= A_R - K_R(k)C_R A_R \\ \Gamma_w &= B_R - K_R(k)C_R B_R \\ \Gamma_* &= G_R - K_R C_R G_R \\ \Gamma_v &= -K_R(k) \end{aligned}$$

The state update equations for the error states when only an IMU measurement occurs are shown in (9.22).

$$\begin{aligned}
\delta x_R(k+1) &= \Phi_{\delta x}\delta x_R(k) + \Gamma_w w(k) + \Gamma_* x^*(k-1) \qquad (9.22)\\
&\text{where}\\
\Phi_{\delta x} &= A_R\\
\Gamma_w &= B_R\\
\Gamma_* &= G_R
\end{aligned}$$

In (9.21), the matrix A_R is the same as is given in (9.12). The noise terms w and v are the same as shown in (9.5) and (9.12). The matrices K_R and C_R are the Kalman gain and observation matrix of the residual state filter, respectively.

The master filter also uses two separate state update equations. The state dynamics of the master filter for simultaneous GPS and IMU measurements are given in (9.23).

$$\begin{bmatrix} x_M(k+1) \\ \delta x_R(k+1) \end{bmatrix} = \begin{bmatrix} A & 0_{17\times 8} \\ \Gamma_1 & \Phi_{\delta x} \end{bmatrix} \begin{bmatrix} x_M(k) \\ \delta x_R(k) \end{bmatrix} + \begin{bmatrix} B \\ \Gamma_2 \end{bmatrix} w(k) + \begin{bmatrix} 0_{17\times 2N} \\ \Gamma_v \end{bmatrix} v(k+1) \qquad (9.23)$$

where

$$\begin{aligned}
\Gamma_1 &= [0_{8\times 6}\ \Gamma_*\ 0_{8\times 2}]\\
\Gamma_2 &= [\Gamma_w(1:8,1:6)\ 0_{8\times 6}\ \Gamma_w(1:8,7:8)]
\end{aligned}$$

The state update equations of the master filter for an IMU only measurement are given in (9.24).

$$\begin{bmatrix} x_M(k+1) \\ \delta x_R(k+1) \end{bmatrix} = \begin{bmatrix} A & 0_{17\times 8} \\ \Gamma_1 & A_R \end{bmatrix} \begin{bmatrix} x_M(k) \\ \delta x_R(k) \end{bmatrix} + \begin{bmatrix} B \\ \Gamma_2 \end{bmatrix} w(k) \tag{9.24}$$

where

$$\begin{aligned} \Gamma_1 &= [0_{8\times 6}\ G_R\ 0_{8\times 2}] \\ \Gamma_2 &= [\Gamma_w(1:8,1:6)\ 0_{8\times 6}\ \Gamma_w(1:8,7:8)] \end{aligned}$$

The states of the residual filter are used as measurements by the master filter. The states of the residual filter are equal to the true residuals plus the errors in its estimates:

$$x_R^+(k) = x_M^*(k) - \hat{x}_M^{*-}(k) - \delta x_R(k) \tag{9.25}$$

The asterisk symbol is used to denote only the position velocity and clock states of the master filter.

In general, the residuals of the filter are related to the states of the filter and the true states by:

$$\epsilon(k) = C(x(k) - \hat{x}^-(k)) + v(k) \tag{9.26}$$

Rearranging (9.25) into the form of (9.26) yields:

$$\begin{aligned} x_R^+(k) &= x_M^*(k) - \hat{x}_M^{*-}(k) - \delta x_R(k) \\ \epsilon(k) &= x_R^+(k) + \delta x_R^-(k) \end{aligned}$$

$$\epsilon(k) \quad = \quad x_M^*(k) - \delta x_R(k) - \hat{x}_M^{*-}(k) + \delta\hat{x}_R^-(k) \tag{9.27}$$

The observation matrix for the master filter is therefore given by (9.28).

$$C_M \quad = \quad \begin{bmatrix} I_{6\times 6} & 0_{6\times 11} & -I_{6\times 6} & 0_{6\times 2} \\ 0_{2\times 15} & I_{2\times 2} & 0_{2\times 6} & -I_{2\times 2} \end{bmatrix} \tag{9.28}$$

After each measurement epoch of the master filter, the state vector of the residual filter is reset to zeros. This is because the state estimates of the master filter are unbiased and the residual filter is tracking the errors in the master filter's states. As a consequence of this, the errors in the master filter's states after a measurement update are correlated with the errors in the residual filter's states. The covariance matrix of the master filter after a measurement update is given in (9.29)

$$P^+(k) \quad = \quad \begin{bmatrix} E\{\hat{x}_M^+(k)(\hat{x}_M^+(k))^T\} & E\{\hat{x}_M^+(k)(\delta\hat{x}_R^+(k))^T\} \\ E\{\delta\hat{x}_R^+(k)(\hat{x}_M^+(k))^T\} & E\{\delta\hat{x}_R^+(k)(\delta\hat{x}_R^+(k))^T\} \end{bmatrix} \tag{9.29}$$

After the measurement update of the master filter and after the states of the residual filter are reset to zero, the errors in the residual filter's states are identical to the errors in the states of the master filter. Consequently, the covariance matrix of the master filter is therefore set equal to:

$$P^+(k) \quad = \quad \begin{bmatrix} E\{\hat{x}_M^+(k)(\hat{x}_M^+(k))^T\} & E\{\hat{x}_M^+(k)(\hat{x}_M^{*+}(k))^T\} \\ E\{\hat{x}_M^{*+}(k)(\hat{x}_M^+(k))^T\} & E\{\hat{x}_M^{*+}(k)(\hat{x}_M^{*+}(k))^T\} \end{bmatrix} \tag{9.30}$$

The covariance matrix of the residual filter is reinitialized to:

$$P_R^+(k) = \begin{bmatrix} E\{\hat{x}_M^+(k)(\hat{x}_M^+(k))^T\} \end{bmatrix} \tag{9.31}$$

9.2.4 Federated Deep Integration Filter with Decoupled Channels

Figure 9.3 shows a block diagram of the federated Deeply Integrated architecture when using a decoupled channel residual filter. The architectures shown in Figures 9.2 and 9.3 are very similar. The master filter is updated at 1 Hz and its states are used to predict the pseudoranges and pseudorange-rates between measurement epochs. The residual filter processes the pseudorange and pseudorange-rate residuals produced by the integrate and dump operations at 50 Hz and its estimates are routed back to correct the master filter's predictions. The main difference between Figures 9.2 and 9.3 is the design of the residual filter. The channels of the residual filter in Figure 9.3 are not coupled together. The pseudorange-state formulation of vector tracking is used to decouple the receiver channels, as described in Chapter 6. Since the residual filter tracks the pseudorange and pseudorange-rate errors, its estimates can be fed directly back to correct the master filter's predictions. The states of the residual filter are used as measurements by the master filter at 1 Hz. The residual filter states are then reset and the cycle is repeated.

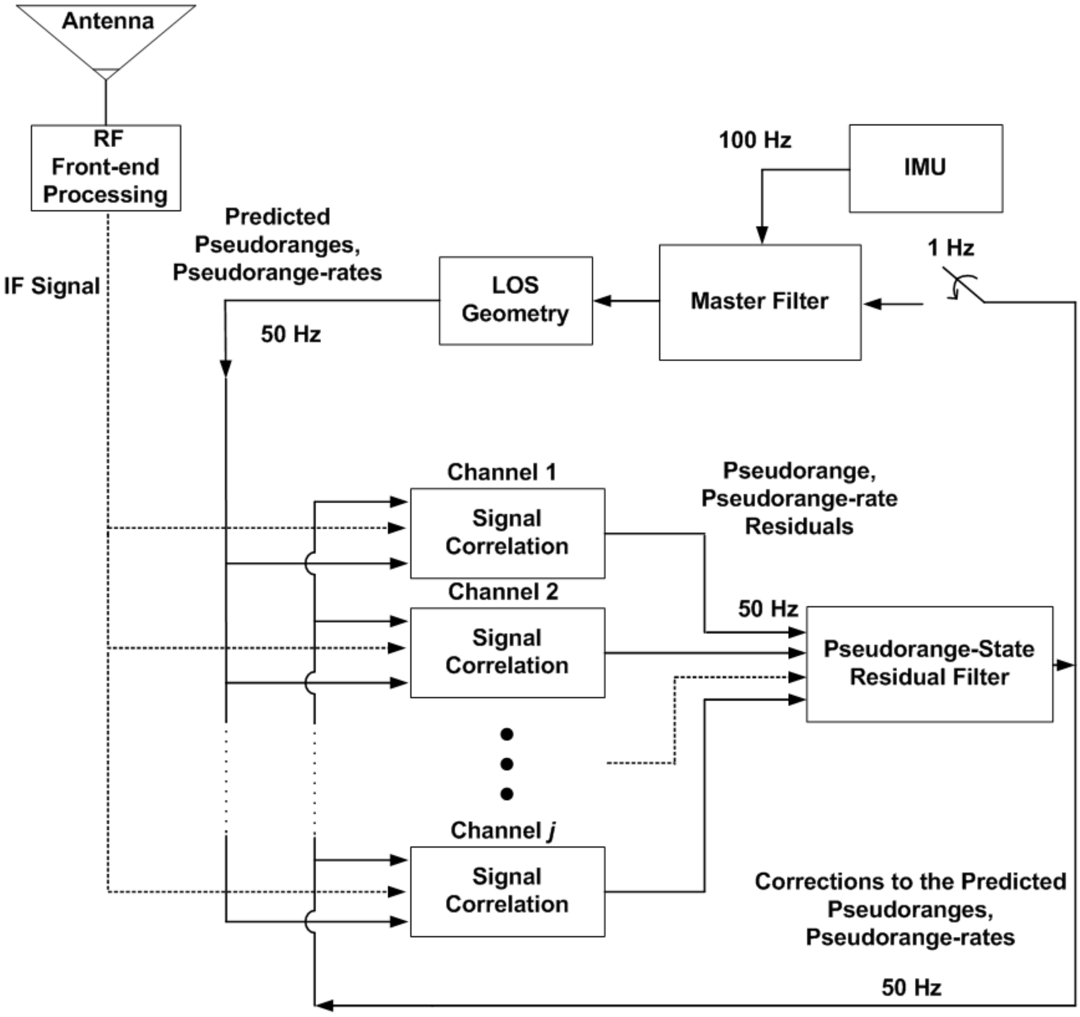

Figure 9.3: Federated Deep Integration Filter with Decoupled Channels

The states of the residual filter are shown in (9.32).

$$\begin{bmatrix} \delta\rho_1 \\ \delta\dot{\rho}_1 \\ \delta\rho_2 \\ \delta\dot{\rho}_2 \\ \vdots \\ \delta\rho_N \\ \delta\dot{\rho}_N \end{bmatrix} = \begin{bmatrix} \text{Pseudorange Error, Channel One} \\ \text{Pseudorange-rate Error, Channel One} \\ \text{Pseudorange Error, Channel Two} \\ \text{Pseudorange-rate Error, Channel Two} \\ \vdots \\ \text{Pseudorange Error, Channel } N \\ \text{Pseudorange-rate Error, Channel } N \end{bmatrix} \tag{9.32}$$

The residual filter tracks errors in the master filter's predictions of the pseudoranges and pseudorange-rates. The pseudorange-state formulation of the vector tracking Kalman filter is used so that the channels can be decoupled. The pseudorange-state VDFLL does not exhibit numerical problems if the channels of the filter are decoupled. The state dynamics of the filter are given by the nonlinear function f_R^ρ:

$$x_R(k+1) \;=\; f_R^\rho(x_R(k), x^*(k), u(k), w(k)) \tag{9.33}$$

In (9.33), function f_R^ρ is the relation between the current states of the residual filter x_R, the errors in the states of the master filter that are not included in the residual filter x^*, the uncorrupted IMU measurements u, and the noise terms w present in the IMU and receiver clock. The superscript ρ is used to delineate it from the function f_R used in the position-state residual filter. The subscript R is used to distinguish the states of the residual filter from the states of the master filter. The vector w_k represents the noise sources in the IMU measurements and receiver clock. The vector $\delta x^*(k)$ represents errors in the states of the master filter, but not those states included in the state vector of the residual filter. The vector $\delta x^*(k)$ includes the errors in the master filters estimates of the receivers attitude and IMU biases. The statistics of the residual filter's state vector are propagated forward by linearizing (9.33):

$$\begin{aligned}
x_R(k+1) &= f_R^\rho(x_R(k), x^*(k), u(k), w(k)) \\
x_R(k+1) &\approx f_R^\rho(\hat{x}_R(k), 0, u(k), 0) + A_R^\rho \cdot \delta x_R(k) + HG_R \cdot x^*(k) + HB_R \cdot w(k) \\
&\text{where} \\
H &= \begin{bmatrix} a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 & 0 \\ 0 & a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 & 0 \\ 0 & a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 \end{bmatrix} \\
A_R^\rho &= \left.\frac{\partial f_R^\rho(\cdot)}{\partial x_R}\right|_{x=\hat{x}(k)} \\
G_R &= \left.\frac{\partial f_R(\cdot)}{\partial x^*}\right|_{x=\hat{x}(k)} \\
B_R &= \left.\frac{\partial f_R(\cdot)}{\partial w}\right|_{x=\hat{x}(k)}
\end{aligned} \tag{9.34}$$

The states of the residual filter are propagated forward in time using (9.35).

$$\begin{aligned}
x_R(k+1) &= f_R^\rho(x_R(k), x^*(k), u(k), w(k)) \\
\hat{x}_R^-(k+1) &= f_R^\rho(\hat{x}_R(k), 0, u(k), 0)
\end{aligned} \tag{9.35}$$

The covariance of the errors in the estimates of the residual filter are propagated using the approximation in (9.34).

The errors in the states of the residual filter are defined in (9.36).

$$\begin{aligned} \delta x_R(k+1) &\equiv x_R(k+1) - \hat{x}_R(k+1) \\ \delta x_R(k+1) &\equiv x_R(k+1) - f_R^{\rho}(\hat{x}_R(k), 0, u(k), 0) \end{aligned} \tag{9.36}$$

Using (9.34) and (9.36), the covariance of the residual filter's estimates after the state propagation are approximated by (9.37).

$$\begin{aligned} E\{\delta x_R(k+1)\} &\approx 0_{2N\times 1} \\ E\{\delta x_R(k+1)\delta x_R^T(k+1)\} &\approx A_R^{\rho} P_R(k) A_R^{\rho T} + HG_R E\{\delta x^*(k)\delta x_R^T(k)\} A_R^{\rho T} \\ &\quad + HB_R Q(k) B_R^T H^T + A_R^{\rho} E\{\delta x_R(k)\delta x^{*T}(k)\} G_R^T H^T \\ &\quad + HG_R E\{\delta x^*(k)\delta x^{*T}(k)\} G_R^T H^T \end{aligned} \tag{9.37}$$

Equation (9.37) requires taking the covariance between the residual filter's state errors δx_R and the state errors in the master filter filters estimates δx^*. These values are maintained in the covariance matrix of the master filter and are extracted from there.

The relationship between the states of the residual filter and the pseudorange and pseudorange-rate residuals supplied at time k is shown in (9.38).

$$\epsilon_R(k) = x_R(k) - \hat{x}_R^{-}(k) + v(k) \tag{9.38}$$

Again, the $-$ and $+$ superscripts are used to represent the state vector before and after a measurement update, respectively.

The state estimate of the residual filter after the measurement update is given in (9.39).

$$\hat{x}_R^+(k) = \hat{x}_R^-(k) + K_R(k)(\epsilon_R(k)) \tag{9.39}$$

The pseudorange and pseudorange-rate residuals ϵ_R generated by the discriminators is directly multiplied by the Kalman gain K_R. This is a consequence of the states of the residual filter being fed back to control the code and carrier NCOs. If the estimates of the residual filter were not fed back, the measurement update equation in (9.39) would be changed to (9.40).

$$\hat{x}_R^+(k) = \hat{x}_R^-(k) + K_R(k)(\epsilon_R(k) - \hat{x}_R^-(k)) \tag{9.40}$$

The master filter in Figure 9.3 combines the IMU measurements and estimates of the residual filter in order to estimate the navigation states of the receiver. Consequently, the master filter's state vector is partitioned into two segments. The first segment contains states that correspond to the receiver's position, velocity, attitude, IMU biases, and clock states. The second segment contains states that represent the errors in the residual filter's state estimates. The full state vector is shown in (9.41).

$$\begin{bmatrix} x_M \\ \delta x_R \end{bmatrix} = \begin{bmatrix} \text{Master Filter's Navigation States} \\ \text{Residual Filter State Errors} \end{bmatrix} \tag{9.41}$$

where

$$
\begin{aligned}
x_M &= \begin{bmatrix} \text{Position States} \\ \text{Velocity States} \\ \text{Attitude States} \\ \text{Accelerometer Bias States} \\ \text{Gyro Bias States} \\ \text{Clock Bias and Drift States} \end{bmatrix} \\
\delta x_R &= \begin{bmatrix} \text{Residual Filter Pseudorange Error, State One} \\ \text{Residual Filter Pseudorange-rate Error, State One} \\ \vdots \\ \text{Residual Filter Pseudorange Error, State N} \\ \text{Residual Filter Pseudorange-rate Error, State N} \end{bmatrix}
\end{aligned}
$$

The pseudoranges and pseudorange-rates are predicted based of the states corresponding to the receiver's position, velocity, clock bias, and clock drift. The extra states in the master filter are appended to model the errors in the states of the residual filter. As shown in (9.41), each state in the residual filter has a corresponding error state in the master filter.

The error states are defined in (9.42).

$$
\begin{aligned}
\delta x_R^-(k) &= x_R(k) - \hat{x}_R^-(k) \\
\delta x_R^+(k) &= x_R(k) - \hat{x}_R^+(k)
\end{aligned}
\tag{9.42}
$$

The error states correspond to the difference between the true pseudorange residuals and the residual filter's estimates. Two update equations are used for the error states. The reason for this is that the residual filter is updated at the rate of the IMU. For this dissertation, a 100 Hz IMU measurement rate is used and a 50 Hz GPS measurement rate is used. One set of update equations are used for when an IMU measurement and a GPS measurement occur simultaneously. The other state update equations are used whenever only an IMU measurement is received. The dynamics of the error states for the simultaneous GPS and IMU measurements are shown in (9.43).

$$\delta x_R(k+1) = \Phi_{\delta x}\delta x_R(k) + \Gamma_w w(k) + \Gamma_* x^*(k-1) + \Gamma_v v(k+1) \tag{9.43}$$

where

$$\begin{aligned}
\Phi_{\delta x} &= A_R^\rho - K_R(k)A_R^\rho \\
\Gamma_w &= HB_R - K_R(k)HB_R \\
\Gamma_* &= HG_R - K_R HG_R \\
\Gamma_v &= -K_R(k)
\end{aligned}$$

The state update equations for the error states when only an IMU measurement occurs are shown in (9.44).

$$\delta x_R(k+1) = \Phi_{\delta x}\delta x_R(k) + \Gamma_w w(k) + \Gamma_* x^*(k-1) \tag{9.44}$$

where

$$\begin{aligned}
\Phi_{\delta x} &= A_R^\rho \\
\Gamma_w &= HB_R \\
\Gamma_* &= HG_R
\end{aligned}$$

$$\begin{bmatrix} x_M(k+1) \\ \delta x_R(k+1) \end{bmatrix} = \begin{bmatrix} A & 0_{17\times 2N} \\ \Gamma_1 & \Phi_{\delta x} \end{bmatrix} \begin{bmatrix} x_M(k) \\ \delta x_R(k) \end{bmatrix} + \begin{bmatrix} B \\ \Gamma_2 \end{bmatrix} w(k) + \begin{bmatrix} 0_{17\times 2N} \\ \Gamma_v \end{bmatrix} v(k+1) \tag{9.45}$$

where

$$\begin{aligned} \Gamma_1 &= [0_{2N\times 6}\ \Gamma_*\ 0_{2N\times 2}] \\ \Gamma_2 &= [\Gamma_w(1:2N,1:6)\ 0_{2N\times 6}\ \Gamma_w(1:2N,7:8)] \end{aligned}$$

The state update equations of the master filter for an IMU only measurement are given in (9.46).

$$\begin{bmatrix} x_M(k+1) \\ \delta x_R(k+1) \end{bmatrix} = \begin{bmatrix} A & 0_{17\times 2N} \\ \Gamma_1 & A_R \end{bmatrix} \begin{bmatrix} x_M(k) \\ \delta x_R(k) \end{bmatrix} + \begin{bmatrix} B \\ \Gamma_2 \end{bmatrix} w(k) \tag{9.46}$$

where

$$\begin{aligned} \Gamma_1 &= [0_{2N\times 6}\ HG_R\ 0_{2N\times 2}] \\ \Gamma_2 &= [\Gamma_w(1:2N,1:6)\ 0_{2N\times 6}\ \Gamma_w(1:2N,7:8)] \end{aligned}$$

The states of the residual filter are used as measurements by the master filter. The observation matrix for the master filter is given by (9.47).

$$C_M = [H(1:2N,1:6)\ 0_{2N\times 9}\ H(1:2N,7:8)\ -I_{2N\times 2N}] \tag{9.47}$$

The state vector of the residual filter is reset to zeros after each measurement epoch of the master filter. This is because the state estimates of the master filter are unbiased and the residual filter is tracking the errors in the master filter's states. As a consequence of this, the errors in the master filter's states after a measurement update are correlated with the errors in the residual filter's states. The covariance matrix of the master filter after a measurement update is given in (9.48)

$$P^{+}(k) = \begin{bmatrix} E\{\hat{x}_M^{+}(k)(\hat{x}_M^{+}(k))^T\} & E\{\hat{x}_M^{+}(k)(\delta\hat{x}_R^{+}(k))^T\} \\ E\{\delta\hat{x}_R^{+}(k)(\hat{x}_M^{+}(k))^T\} & E\{\delta\hat{x}_R^{+}(k)(\delta\hat{x}_R^{+}(k))^T\} \end{bmatrix} \tag{9.48}$$

After the measurement update of the master filter and after the states of the residual filter are reset to zeros, the errors in the residual filter's states are related to the errors in the states of the master filter by:

$$\delta\hat{x}_R^{+}(k) = H^{*}\hat{x}_M^{+}(k) \tag{9.49}$$

Consequently, the covariance matrix of the master filter is therefore set equal to:

$$P^{+}(k) = \begin{bmatrix} E\{\hat{x}_M^{+}(k)(\hat{x}_M^{+}(k))^T\} & E\{\hat{x}_M^{+}(k)(\hat{x}_M^{+}(k))^T\}\cdot H^{*T} \\ H^{*}\cdot E\{\hat{x}_M^{+}(k)(\hat{x}_M^{+}(k))^T\} & H^{*}\cdot E\{\hat{x}_M^{+}(k)(\hat{x}_M^{+}(k))^T\}\cdot H^{*T} \end{bmatrix} \tag{9.50}$$

where (9.51)

$$H^{*} = [H(1:2N,1:6)\ \ 0_{2N\times 9}\ \ H(1:2N,7:8)] \tag{9.52}$$

Similarly, after the measurement update the covariance matrix of the residual filter is reinitialized to:

$$P_R^{+}(k) = H^{*}\cdot E\{\hat{x}_M^{+}(k)(\hat{x}_M^{+}(k))^T\}\cdot H^{*T} \tag{9.53}$$

9.3 Covariance Analysis of Deeply Integrated Architectures

In this section, a covariance analysis of the three deeply integrated algorithms is performed using several different satellite constellations and IMU grades. The purpose of the covariance analysis is to determine the performance of the algorithms relative to each other. The loss in DI's performance caused by using a federated architecture and by using scalar residual filters is determined by comparing the three DI algorithms.

Noise specifications corresponding to two IMU grades are used in the covariance analysis. The noise specifications are for a tactical grade IMU and an automotive grade IMU. Table 9.1 contains the various values used for the IMU noise processes [Flenniken, IV, 2005]. Appendix E contains an explanation of how the different noise processes are modeled. A default IMU update time of 100 Hz is used in the covariance analysis.

Table 9.1: IMU Noise Parameters

Parameter	Random Walk	Bias Time Constant	Bias Variation
Units	$^\circ/\text{sec}/\sqrt{\text{Hz}}$	second	$^\circ/\text{hour}$
Tactical Grade Gyro	.0017	100	.35
Automotive Grade Gyro	.05	300	180
Parameter	Random Walk	Bias Time Constant	Bias Variation
Units	$\text{g}/\sqrt{\text{Hz}}$	second	g
Tactical Grade Accelerometer	.0005	60	$50 \cdot 10^{-5}$
Automotive Grade Accelerometer	.001	100	$1.2 \cdot 10^{-3}$

The measurement noise statistics are defined by the type of discriminators that are used and the C/N_0 ratio. A pseudorange and pseudorange-rate discriminator are used to transform the correlator outputs into usable measurements for the filters. The pseudorange discriminator uses the correlator outputs accumulated over the entire 20 ms integrate and dump interval [Crane, 2007]:

$$\begin{aligned} Y_R(k) &= IE(k)^2 + QE(k)^2 - IL(k)^2 - QL(k)^2 \\ Y_R(k) &= A(k)^2\Big(\frac{2\cdot\rho_e(k)}{\beta}\Big) + v_R(k) \\ \rho_e(k) &= \text{Pseudorange error (meters)} \\ \beta &= \text{PRN chip length, 293.3 meters for C/A code} \\ A(k) &= \text{Signal amplitude} \\ v_R(k) &= \text{Noise} \end{aligned} \tag{9.54}$$

The mean and mean squared value of the random variable v_R are:

$$\begin{aligned} E\{v_R\} &= 0 \\ E\{v_R^2\} &= \frac{\beta^2}{2(TC/N_o)^2} + \frac{\beta^2}{TC/N_o}\left(\frac{\rho_e^2}{\beta^2} + \frac{1}{4}\right) \text{ (meters}^2\text{)} \end{aligned} \tag{9.55}$$

The pseudorange-rate discriminator uses correlator outputs from two adjacent 10 ms integrate and dump operations to generate pseudorange-rate residuals $\dot{\rho}_{err}(k)$ [Crane, 2007]:

$$\begin{aligned} Y_{RR}(k) &= IP_2(k)QP_1(k) - IP_1(k)QP_2(k) \\ Y_{RR}(k) &\approx -A_F(k)^2R^2(\rho_e(k))\frac{\pi f_{err}(k)Tc}{4f_{L1}} + v_{RR} \\ \rho_e(k) &= \text{Pseudorange error error (meters)} \\ \dot{\rho}_{err}(k) &= \text{Pseudorange-rate error (m/s)} \\ v_{RR} &= \text{Noise} \end{aligned} \tag{9.56}$$

The mean and mean squared value of v_{RR} are:

$$\begin{aligned} E\{v_{RR}\} &= 0 \\ E\{v_{RR}^2\}^{\dagger} &= \left(\frac{2}{(TC/N_o)^2} + \frac{2R^2(\rho_e(k))}{TC/N_o}\right)\left(\frac{c}{\pi T f_{L1}}\right)^2 \text{ (m/s)}^2 \\ \dagger &= \text{assuming } |\rho_e(k)| \leq \frac{\beta}{2} \end{aligned} \tag{9.57}$$

In (9.55) and (9.57) the value of T is the predetection integration interval used to generated the correlator outputs in (9.54), which is 20 ms in this work. An early-to-late correlator spacing of one chip is assumed. The noise terms in the discriminator outputs in (9.55) and (9.57) are uncorrelated with each other [Crane, 2007].

The values for the clock phase and frequency noise are based on rule of thumb numbers for a temperature compensated crystal oscillator (TCXO). The power spectral density coefficients used for the oscillator are [Brown and Hwang, 1996]:

$$\begin{aligned} h_0 &= 2 \cdot 10^{-19} \\ h_{-1} &= 7 \cdot 10^{-21} \\ h_{-2} &= 2 \cdot 10^{-20} \end{aligned} \tag{9.58}$$

Two satellite constellations are used for the covariance analyses. One contains eleven satellites and the other contains five satellites. Table 9.3 summarizes the statistical characteristics of the eleven and five satellite geometries.

A dynamic flight profile is used for the covariance analyses. The flight profile starts at the coordinates (6378.2, 0, 0) km in the ECEF coordinate frame. The ECEF x-axis

Table 9.2: Dilution of Precision Values for Different Satellite Configurations

Number of Satellites	PDOP	HDOP	VDOP
Eleven	1.4141	0.7527	1.1971
Five	2.3602	1.1300	2.0721

coordinate of the user remains essentially constant throughout the flight. Significant dynamics only occur in the ECEF y-axis and z-axis. Figure 9.4 shows the y-axis and z-axis coordinates during the flight profile. The flight profile includes three coordinated turns. The first is a 2 G turn, followed by a 4 G turn. The last maneuver is an 8 G coordinated turn.

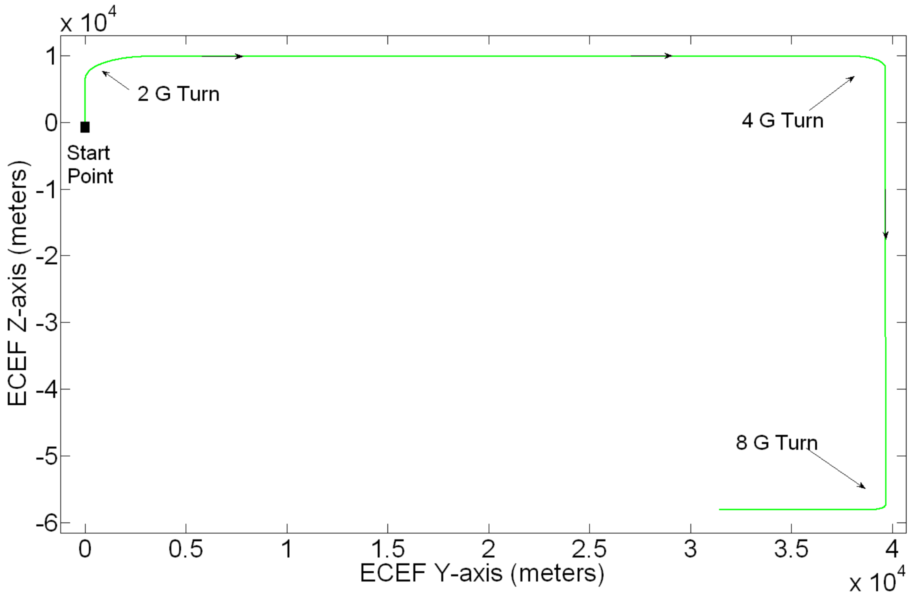

Figure 9.4: Receiver Trajectory During Flight Profile

The process and measurement noise data are specified by the IMU grade and C/N_0 ratio at which the simulations are being performed. Uncorrupted IMU data is used to propagate the states and covariance matrices of the extended Kalman filters used by the algorithms. The covariance matrices at the end of the flight profile are used for the comparison.

The architecture of the centralized and federated architectures makes performing a valid comparison complicated. The reason for this is due to the different update rates used by the centralized filter and the master filters in the two federated algorithms. The centralized filter is updated with pseudorange and pseudorange-rate residuals at 50 Hz, while the master filters in the federated architectures receive measurements at 1 Hz. Between measurement updates, the covariance matrix of the federated filters increases because no measurements are available. This is an unavoidable consequence of using the federated architectures. The state covariance matrix only decreases after the 1 Hz measurement update. In the interest of making a valid comparison, the state covariance matrices of the different algorithms are compared immediately after the measurement updates of the master filters.

9.3.1 Tactical Grade IMU, Temperature Controlled Oscillator, and Eleven Satellite Scenario

Figure 9.5 shows the results of the covariance analysis for the ECEF x-axis position state when using the eleven satellite constellation and a tactical grade IMU. The variance of the state is plotted logarithmically versus C/N_0 ratio. The log of the variance is plotted to make interpreting the analysis results easier. The performance of the centralized DI algorithm and the federated DI algorithm with coupled channels are identical. The federated DI algorithm with uncoupled channels performs more poorly than the other two DI algorithms. However, it is important to recognize that the performance of all three algorithms converge together as the C/N_0 ratio declines.

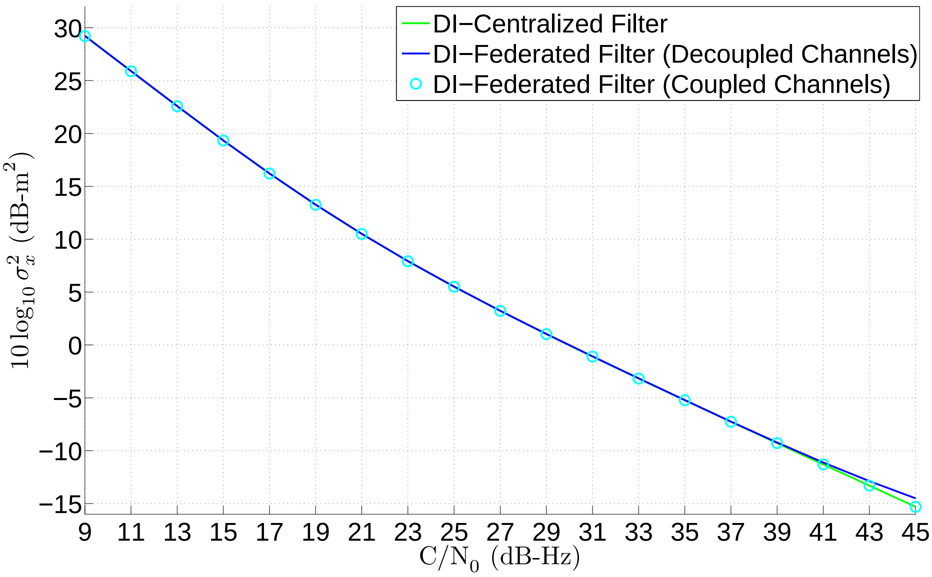

Figure 9.5: Covariance Analysis Results for the X-Axis Position State with Eleven Satellites and Tactical Grade IMU

The difference in performance of the algorithms can be expressed in dB by examining the horizontal difference between the curves. The horizontal difference is defined here as the effective gain in C/N_0 ratio. For example, at 45 dB-Hz the effective gain in C/N_0 ratio for the centralized DI algorithm over the decoupled channel DI algorithm is about .5 dB .

Figure 9.6 shows the variance of the x-axis velocity state versus C/N_0 ratio for the eleven satellite and tactical grade IMU scenario. As in Figure 9.5, the performance of all the architectures that employ vector tracking are identical. Also, the performance of the DI architecture with uncoupled channels is noticeably poorer by comparison. The difference in the performance of the algorithms is more distinct in the case of the velocity state. However, it is again important to note that the performance of all three architectures converge as the C/N_0 ratio decreases.

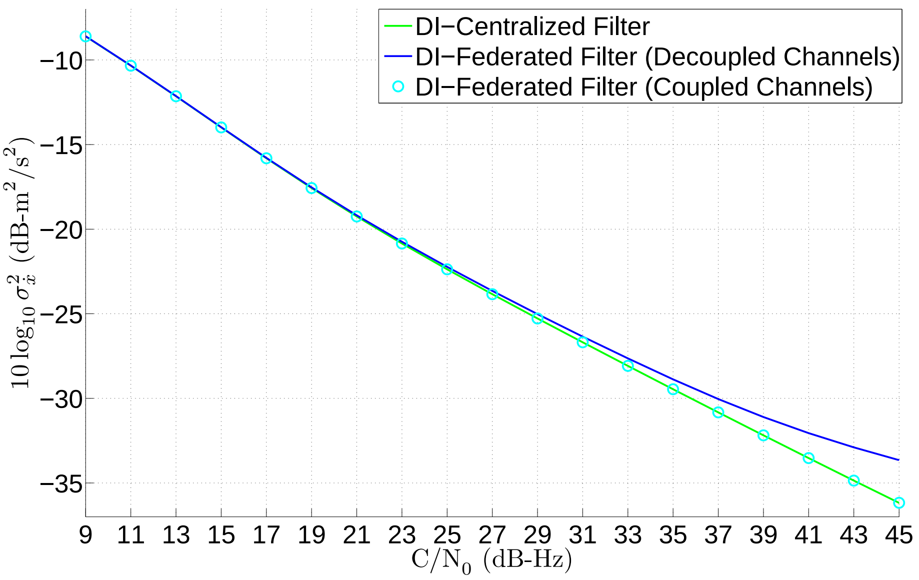

Figure 9.6: Covariance Analysis Results for the X-Axis Velocity State with Eleven Satellites and Tactical Grade IMU

The covariance analysis results for the clock bias and drift states for the eleven satellite and tactical grade IMU are shown in Figures 9.7 and 9.8, respectively. The performance of the algorithms for the clock states is similar to that previously shown for the position and velocity states. The difference in the performances of the algorithms is most noticeable in the clock drift state.

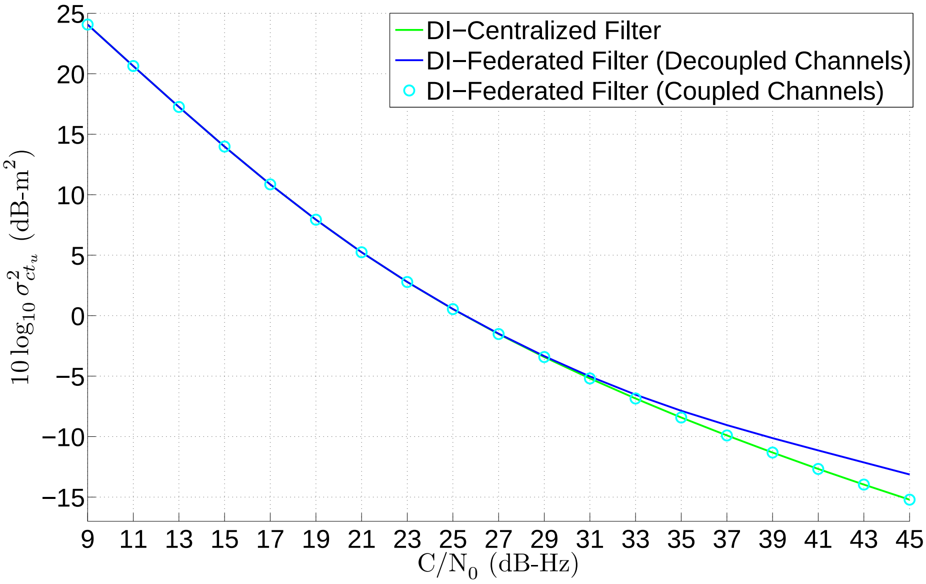

Figure 9.7: Covariance Analysis Results for the Clock Bias State with Eleven Satellites and Tactical Grade IMU

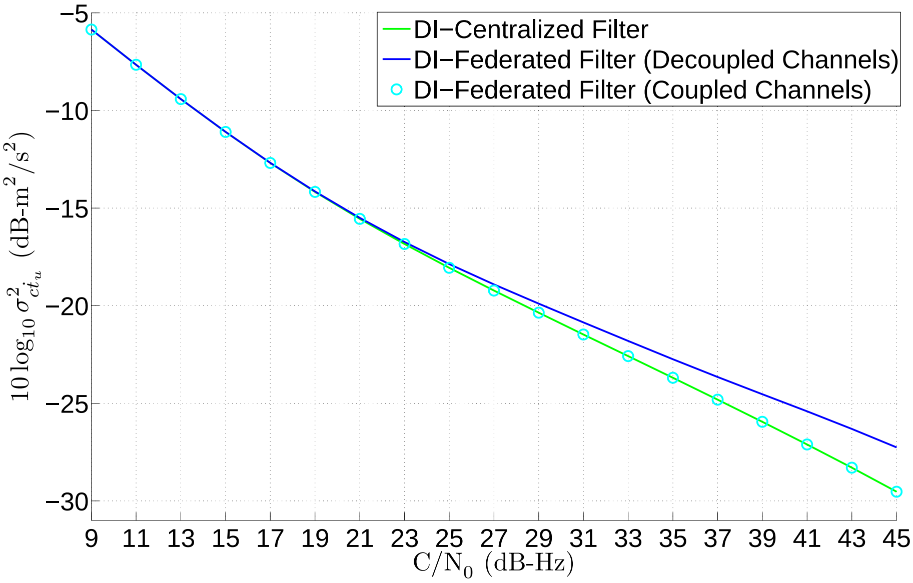

Figure 9.8: Covariance Analysis Results for the Clock Drift State with Eleven Satellites and Tactical Grade IMU

Figures 9.5 through 9.8 demonstrate that the performance of the three DI architectures converge together at low C/N_0 ratios. Furthermore, the DI algorithms that use vector tracking to process the GPS signals exhibt the same performance. The degree to which they outperform the decoupled channel DI architecture is itself a function of the C/N_0 ratio. In Figure 9.5, the coupled channel algorithms display an improvement in effective C/N_0 ratio of 4 to 5 dB at 45 dB-Hz. At 21 dB-Hz, the improvement in effective C/N_0 ratio is essentially zero. These results raise the question of why the coupled channel architectures outperform the decoupled channel architectures only at high C/N_0 ratios. To answer this question, it is useful to recall the results presented in Chapter 8.

The results presented in Figures 9.5 through 9.8 mirror the results presented in Chapter 8 for the different vector tracking architectures. Recall that the performance of two coupled channel algorithms and one decoupled channel algorithm are compared over a range of C/N_0 ratios. The coupled channel architectures display a significant improvement in performance at high C/N_0 ratios. However, at low C/N_0 ratios, the three algorithms perform identically. The increase in performance offered by the coupled channel architectures is a function of C/N_0 ratio, just as is shown in Figures 9.5 through 9.8.

A consistent improvement in effective C/N_0 ratio across all C/N_0 ratios is shown to be possible in Chapter 8. This is accomplished by scaling the process noise with the C/N_0 ratio. As the C/N_0 ratio declines, the process noise is scaled upward. The measurement noise is scaled automatically by the C/N_0 ratio (as shown in (9.55) and (9.57)). Scaling both noise sources yields a constant effective gain in C/N_0 ratio.

For the DI algorithms, the process noise is fixed by the IMU grade and oscillator type. Scaling the process noise with the C/N_0 therefore lacks verisimilitude. However, instead of attempting to produce a scenario where the effective gain in C/N_0 ratio is constant, it is useful to consider a situation where the performance of all the algorithms is identical.

As shown in Figures 9.5 through 9.8, the two coupled channel DI algorithms perform better than the decoupled channel DI algorithm. However, the algorithms can be manipulated so that all three DI algorithms perform virtually identical. This is accomplished by changing the update rate of the master filter in the federated architectures. For the covariance analysis results presented in Figures 9.5 through 9.8, an update rate of 1 Hz is used for the federated architectures. When the update rate is increased to 25 Hz, the performance of the decoupled DI algorithm approaches the performance of the coupled channel DI algorithms. Figures 9.9 and 9.10 show a covariance analysis of the x-axis position and velocity states at 45 dB-Hz with the tactical grade IMU and eleven satellites. Towards the end of the flight profile in the covariance analysis, the update rate of the federated filter is changed from 1 to 25 Hz. After the update rate is altered, the variances of the position and velocity states for the decoupled channel filter nearly converge to the variances for the coupled state filters.

The behavior of the decoupled DI algorithm in response to the change in update rates shown in Figures 9.9 and 9.10 is explained by examining the covariance of the predicted pseudoranges and pseudorange-rates of the residual filters. The covariances of the estimated pseudoranges and pseudorange-rates of the residual filters are important because they are effectively used as measurements by the master filters. The covariance matrix for the decoupled channel residual filter represents the estimated pseudoranges and pseudorange-rates. This is because the states of the residual filter are the pseudoranges and pseudorange-rates themselves. The observation matrix of the coupled channel residual filter is used to

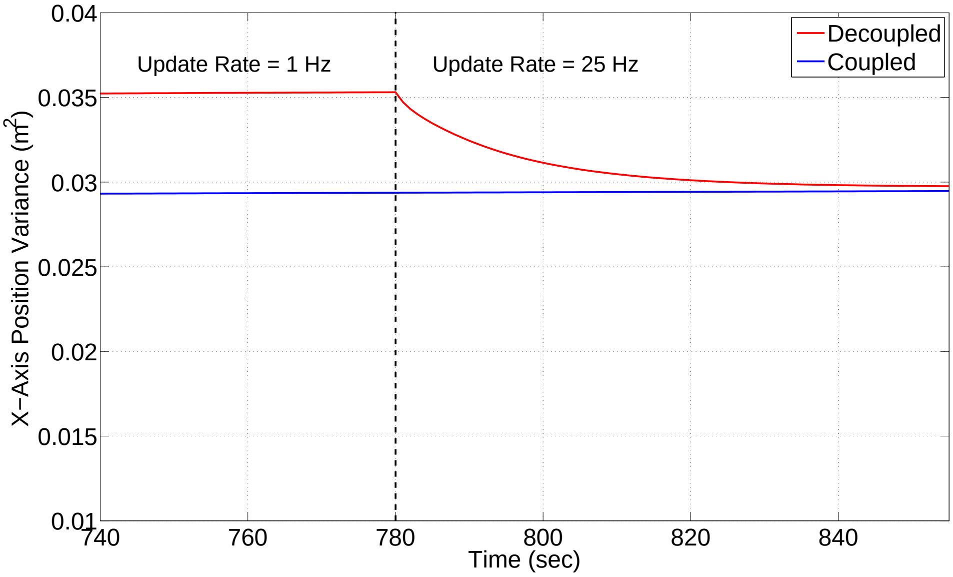

Figure 9.9: X-Axis Position Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 45 dB-Hz

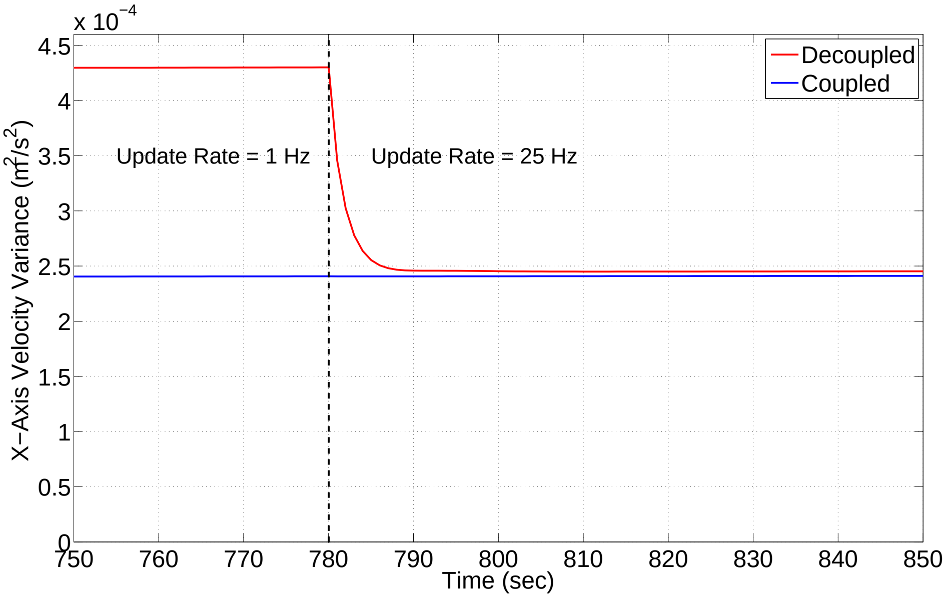

Figure 9.10: X-Axis Velocity Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 45 dB-Hz

transform the covariance of the filter's states (i.e. position, velocity, and clock states) into the covariance of the predicted pseudoranges and pseudorange-rates. The covariances of the pseudorange and pseudorange-rate predictions for the two residual filters are shown in Figures 9.11 and 9.12 for the tactical grade IMU and eleven satellite scenario, respectively.

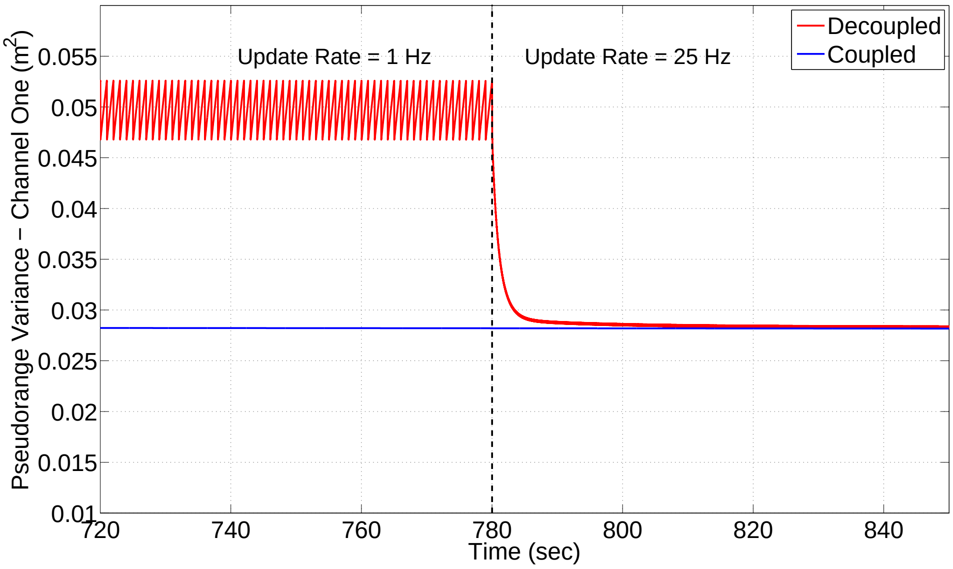

Figure 9.11: Channel One Pseudorange Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 45 dB-Hz

As shown in Figures 9.11 and 9.12, the covariances of the pseudorange and pseudorange-rate predictions for the coupled state residual filter don't change significantly during, or in between, updates of the master filter. In contrast, the covariances of the pseudorange and pseudorange-rates predictions for the decoupled state residual filter increase between updates of the master filter. The covariances decrease after a measurement update of the master filter, then increase until another master filter measurement update occurs. For the decoupled federated DI algorithm, the measurement update of the master filter uses

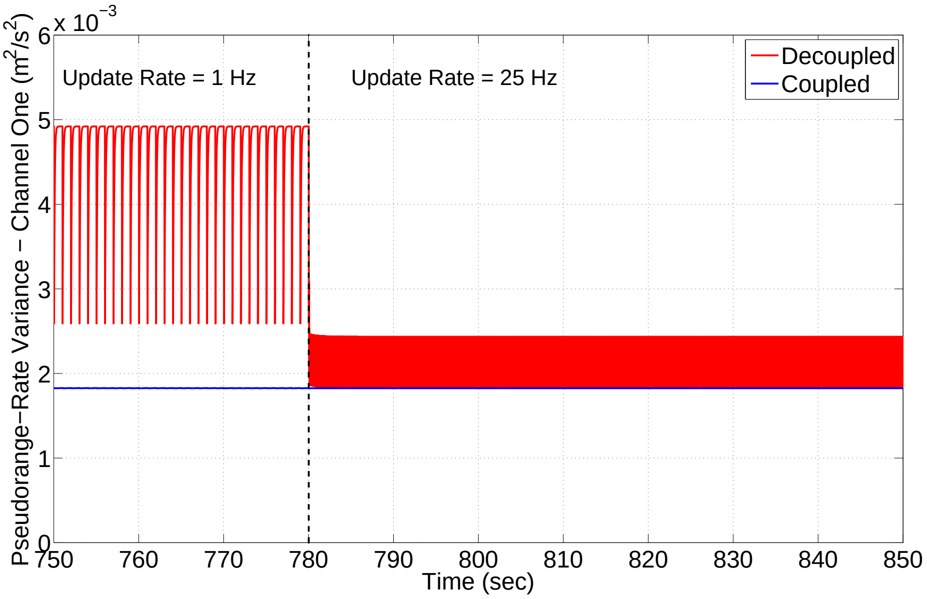

Figure 9.12: Channel One Pseudorange-Rate Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 45 dB-Hz

measurements from all the channels and effectively serves to correct the loss in optimality due to the use of scalar tracking by the residual filter. After the measurement update, the state vector and covariance matrix of the residual filters are reset. The residual filter state vector is zeroed and the master filter's covariance matrix is used to reinitialize the residual filter's covariance matrix. The covariances of the decoupled residual filter's pseudorange and pseudorange-rate estimates then diverge from their reset values. This loss in performance is caused by the lack of information sharing between channels.

Figures 9.13 and 9.14 show the covariances of the position and velocity states before and after an update rate change for the tactical grade IMU and eleven satellite scenario at 20 dB-Hz. The same type of behavior observed at 45 dB-Hz is seen at 20 dB-Hz. However, the difference in the covariance of the position and velocity states after the update rate change at 20 dB-Hz is less pronounced then it is at 45 dB-Hz.

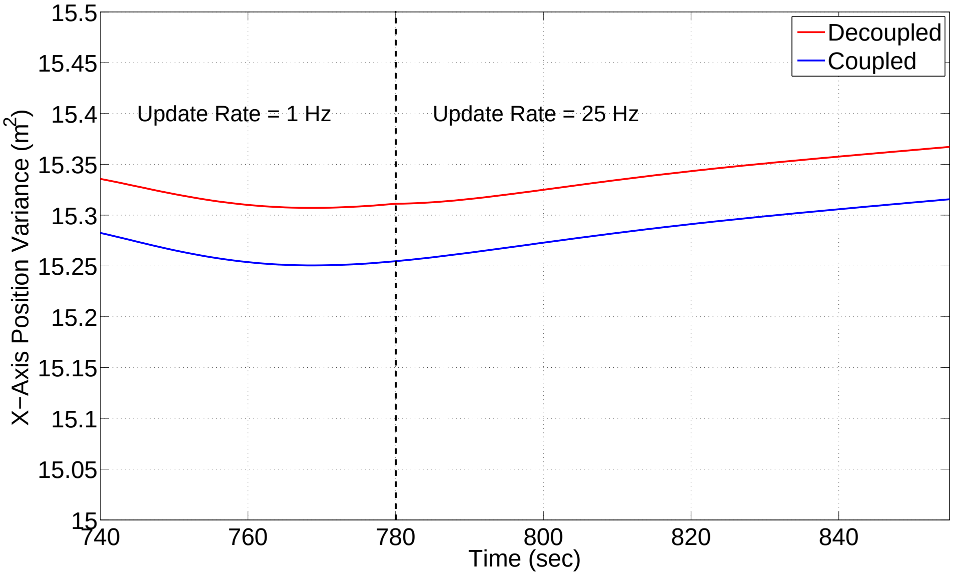

Figure 9.13: X-Axis Position Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 20 dB-Hz

Figures 9.11 and 9.12 demonstrate why the federated DI algorithm with decoupled channels performs more poorly than the DI algorithms using vector tracking. In these figures, the variance of the pseudoranges and pseudorange-rates between updates of the master filter for the coupled and decoupled channel DI architectures are compared. Recall that the residual filters in both federated DI architectures are reset after a measurement update of the master filter. Their state vectors are zeroed and their covariance matrices are reinitialized by using the master filter's covariance matrix. Figures 9.11 and 9.12 show that the pseudorange and pseudorange-rate variances for the decoupled channel DI filter increase between master filter measurement updates. At a measurement update of the master filter,

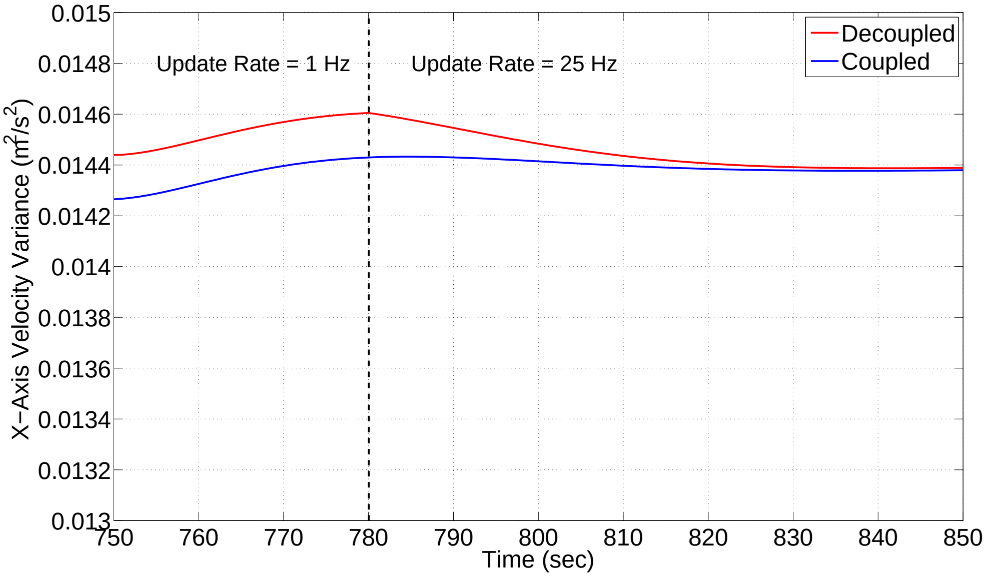

Figure 9.14: X-Axis Velocity Variance Before and After Update Rate Change with Eleven Satellites and Tactical Grade IMU at 20 dB-Hz

information is collected from all the channels and used to estimate the navigation states of the system. The residual filters are then reset using the collective information from all channels by the master filter.

It is interesting to consider the performance of the federated algorithms when the residual filters are not reset after a master filter measurement update. In this scenario, the estimates of the residual filters are not zeroed after being used by the master filter. Likewise, their covariance matrices are not reset using the master filter's covariance matrix. When used in this manner, the residual filters would have to track the receiver's actual navigation states, not just the errors in the master filter's estimates of the navigation states. The coupled channel residual filter would have to track the receiver's actual position, velocity, and clock states. Similarly, the decoupled channel residual filters would need to track the actual received pseudoranges and pseudorange-rates. When the residual filters are operated in this manner, they will be referred to here as Tightly Coupled (TC) algorithms. The federated architecture with coupled channels is referred to as tight coupling with vector tracking. Similarly, the federated architecture with decoupled channels is referred to as tight coupling with scalar tracking.

Figures 9.15 through 9.16 show the results of the covariance analysis for the x-axis position and velocity states for the eleven satellite constellation and tactical grade IMU for the two tightly coupled algorithms. The results for the three DI algorithms are also shown for comparison.

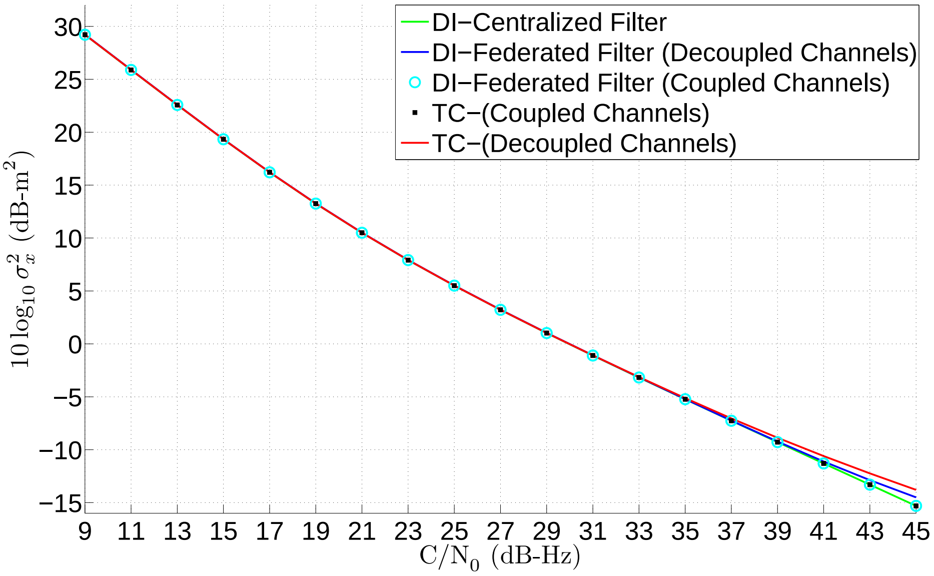

Figure 9.15: Covariance Analysis Results for the X-Axis Position State with Eleven Satellites and Tactical Grade IMU

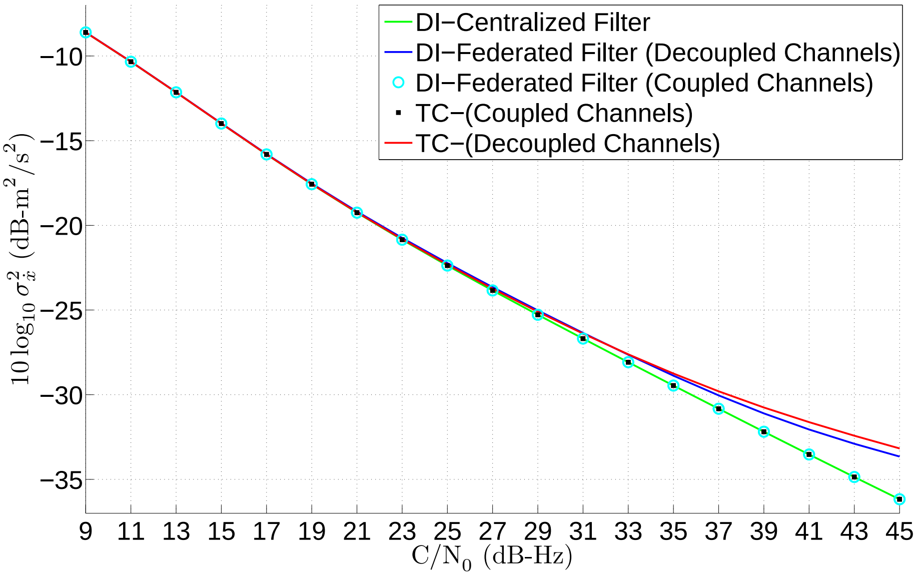

Figure 9.16: Covariance Analysis Results for the X-Axis Velocity State with Eleven Satellites and Tactical Grade IMU

Figures 9.15 and 9.16 show that the tightly coupled algorithm with vector tracking performs identical to the two coupled channel DI algorithms. This is not surprising when considering Figures 9.11 and 9.12. The pseudorange and pseudorange-rate variances do not increase between updates of the master filter for the federated DI algorithm with coupled channels. Consequently, the algorithm performs the same whether its residual filter is reset or not. In contrast, the variance of the pseudoranges and pseudorange-rates increase for the decoupled channel DI filter between master filter updates. When its residual filter is not reset, the pseudorange and pseudorange-rate variances are free to increase until they reach steady-state values. Consequently, the tightly coupled algorithm with decoupled channels performs worse than the decoupled channel DI algorithm. However, it is important to note that the performance of all five algorithms still converge as the C/N_0 ratio declines.

9.3.2 Automotive Grade IMU, Temperature Controlled Oscillator, and Five Satellite Scenario

A covariance analysis of the different architectures is carried out with the five satellite constellation and an automotive grade IMU. The covariance of the states of the filters are examined at the end of the flight profile. The results of the covariance analysis for the ECEF x-axis position and velocity states are shown in Figures 9.17 and 9.18, respectively. Compared to Figures 9.5 and 9.6, the performance of the filters are all noticeably worse. However, the performance of all the algorithms again converge as the C/N_0 ratio declines.

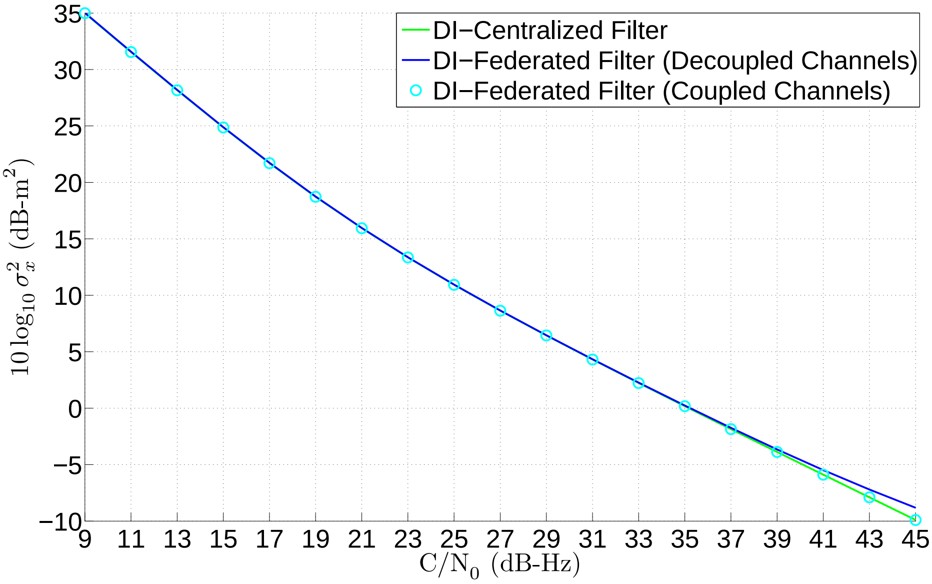

Figure 9.17: Covariance Analysis Results for the X-Axis Position State with Five Satellites and Automotive Grade IMU

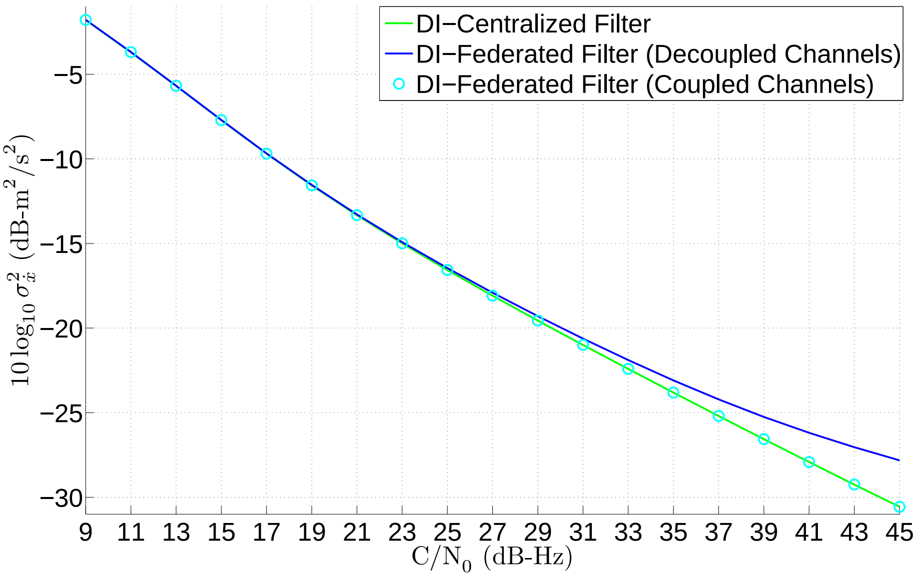

Figure 9.18: Covariance Analysis Results for the X-Axis Velocity State with Five Satellites and Automotive Grade IMU

Figures 9.19 and 9.20 show the results of the covariance analysis for the clock bias and clock drift states, respectively. As with the position and velocity states, the performance of the algorithms is much poorer with the automotive grade IMU and five satellite constellation. However, at low C/N_0 ratios the algorithms all yield the same performance.

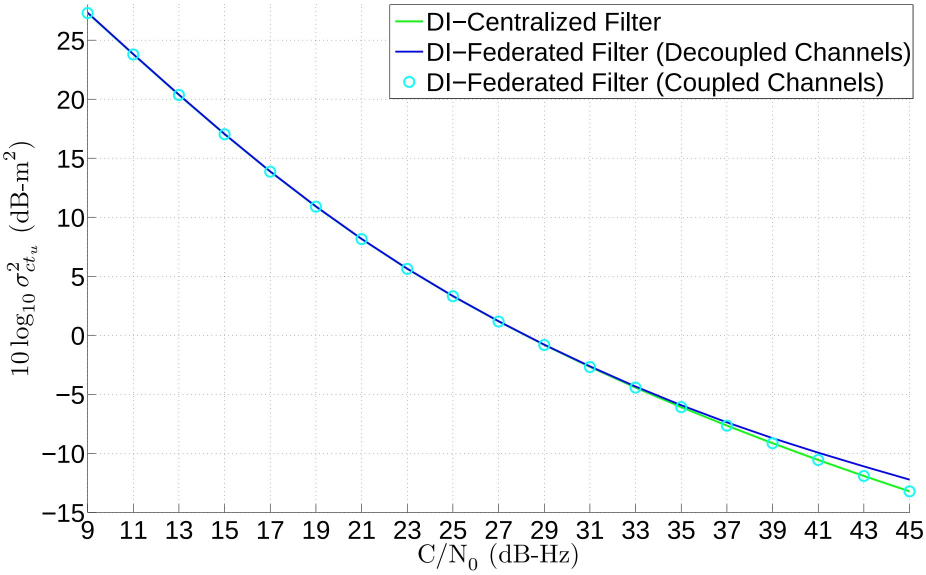

Figure 9.19: Covariance Analysis Results for the Clock Bias State with Five Satellites and Automotive Grade IMU

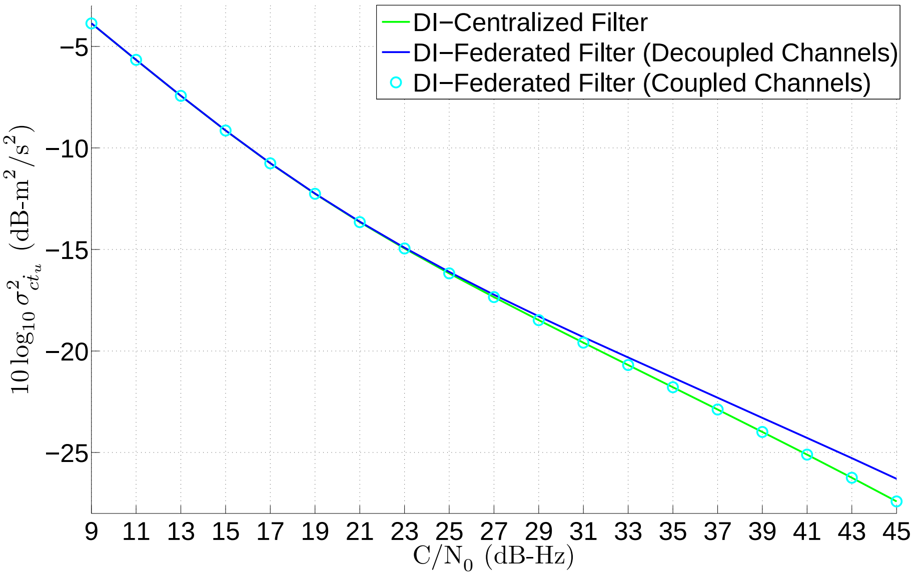

Figure 9.20: Covariance Analysis Results for the Clock Drift State with Five Satellites and Automotive Grade IMU

9.4 Monte Carlo Test Results of Deep Integration Architectures

In this section Monte Carlo simulations are used to verify the covariance analysis in the preceding section. Monte Carlo simulations are performed for the eleven satellite and tactical grade IMU scenario and the five satellite and automotive grade IMU case. The flight profile shown in Figure 9.4 is used to create uncorrupted IMU data. The stochastic errors are then added to the uncorrupted data using the noise statistics for the different IMU grades shown in Table 9.3.

Starting at 45 dB-Hz, a hundred Monte Carlo simulations are performed at decrements of 2 dB-Hz until the DI algorithms begin to loose lock. A pseudorange error of over 150 meters in any channel is used as the criteria for declaring loss of lock. The one hundred sets of IMU data used by the algorithms at each C/N_0 ratio are identical. The stochastic errors present in the IMU data for trial one are the same at all C/N_0 ratios. However, the stochastic errors in the IMU data vary from one Monte Carlo run to another. The same IMU data is used by all three DI algorithms. Furthermore, all three algorithm simulations use the same seed in the random number generator for a given trail number. All three algorithms are initialized with the correct state vector for the simulation. The covariance matrices of all three algorithms are initialized using results obtained from the covariance analysis.

The received GPS signals were simulated at the correlator output level. The various correlator outputs are modeled as shown in (9.59) [Sayre, 2003].

$$\begin{aligned} IE(k) &= AR(\epsilon+\tau)D(k)\cos\left(\pi f_{err}T+\theta_{err}\right)+\eta_{IE}(k) \\ IP(k) &= AR(\epsilon)D(k)\cos\left(\pi f_{err}T+\theta_{err}\right)+\eta_{IP}(k) \end{aligned}$$

$$
\begin{aligned}
IL(k) &= AR(\epsilon-\tau)D(k)\cos(\pi f_{err}T+\theta_{err})+\eta_{IL}(k) \qquad (9.59)\\
QE(k) &= AR(\epsilon+\tau)D(k)\sin(\pi f_{err}T+\theta_{err})+\eta_{QE}(k)\\
QP(k) &= AR(\epsilon)D(k)\sin(\pi f_{err}T+\theta_{err})+\eta_{QP}(k)\\
QL(k) &= AR(\epsilon-\tau)D(k)\sin(\pi f_{err}T+\theta_{err})+\eta_{QL}(k)\\
A &= \sqrt{\frac{2CT_{PDI}}{N_o}}\cdot\frac{\sin(\pi f_{err}T)}{(\pi f_{err}T)}\\
\eta &= N(0,1)
\end{aligned}
$$

In (9.59), the correlator outputs have been normalized so that the noise is unit variance. The variables f_{err} and θ_{err} are the frequency error (Hertz) and phase error (radians) between the received and locally generated carrier signals, respectively. The sinc (sin(x)/x) term in (9.59) is an effective loss in signal amplitude due to the frequency error, f_{err}. The term ϵ is the phase error between the received and locally generated PRN codes, measured in fractions of a chip. The function $R(\epsilon)$ is the auto-correlation function of the PRN sequences. The symbol τ is the correlator spacing of the Early and Late PRN replicas, expressed in fractions of a chip. A correlator spacing of one half chip is used for all the Monte Carlo simulation results. The autocorrelation of the Gold code as a function of the code phase error is:

$$
R(\epsilon)=\begin{cases} 1-|\epsilon| & \text{when } \epsilon<1 \text{ chip}\\ 0 & \text{when } \epsilon>1 \text{ chip.}\end{cases}
$$

The noise terms in (9.59) are all zero-mean and unit variance. The in-phase and quadrature noise terms are uncorrelated. Using a half chip correlator spacing results in the early and late noise terms being uncorrelated. The early and prompt noise terms, and the prompt and late noise terms, all have a correlation coefficient of 0.5 .

The pseudorange and pseudorange-rate discriminator functions in (9.55) and (9.57) are used in the simulations. An integrate and dump time of 20 milliseconds is used for the correlator outputs in (9.55). Two consecutive integrate and dump times of 10 ms are used for the correlator outputs in (9.57). The DI algorithms estimated the signal amplitudes and noise power using the techniques given in Appendix B.

A temperature compensated crystal oscillator is used for all the simulations. The uncorrupted clock data is generated using an initial clock bias of zero and clock drift of 177 m/s^2. The initial clock drift value is chose based off the typical clock drift displayed by the oscillator in the NordNav receiver that is used in Chapter 10. Stochastic errors are added to the clock data by using the power spectral density coefficients shown in (9.58) [Brown and Hwang, 1996].

9.4.1 Eleven Satellite and Tactical Grade IMU Monte Carlo Simulation Results

The Monte Carlo simulation results for the x-axis position and velocity states are shown in Figure 9.21 and 9.22 for the eleven satellite and tactical grade IMU scenario. The empirically calculated position and velocity error variances for all three DI algorithms are plotted versus C/N_0 ratio. The errors are the difference in the true position and velocity and the estimates of these quantities made by the master filters after a measurement update. Since a measurement update rate of 1 Hz is used, the errors are computed at 1 Hz. The predictions from the covariance analysis are also plotted for comparison. Overall the results from the Monte Carlo simulations match the predictions from the covariance analysis quite well.

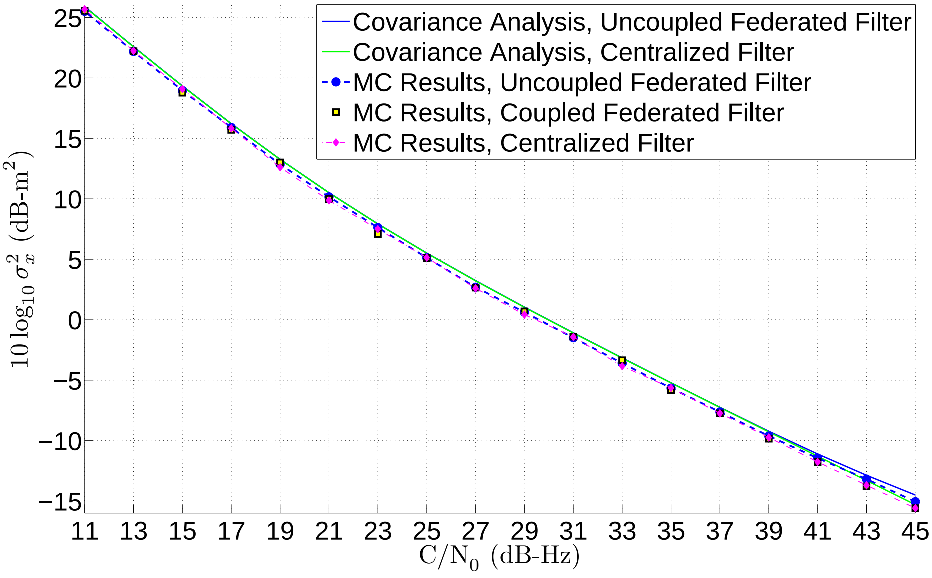

Figure 9.21: Monte Carlo Simulation Results for the X-Axis Position State with Eleven Satellites and Tactical Grade IMU

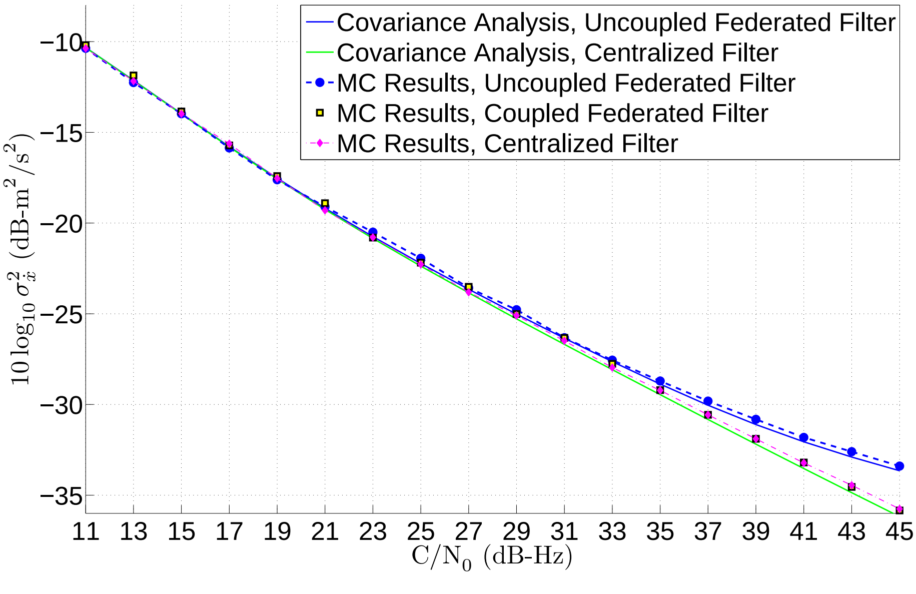

Figure 9.22: Monte Carlo Simulation Results for the X-Axis Velocity State with Eleven Satellites and Tactical Grade IMU

Figure 9.23 and 9.24 show the Monte Carlo simulation results for the clock bias and clock drift states, respectively. The results for the clock bias state match the predictions of the covariance analysis very closely. The results for the clock drift also match the covariance analysis for the two federated filters. However, the performance of the centralized filter is poorer than that predicted by the covariance analysis for the clock drift state. Additional plots for the Monte Carlo simulation results are included in Appendix C for the eleven satellite and tactical grade IMU scenario.

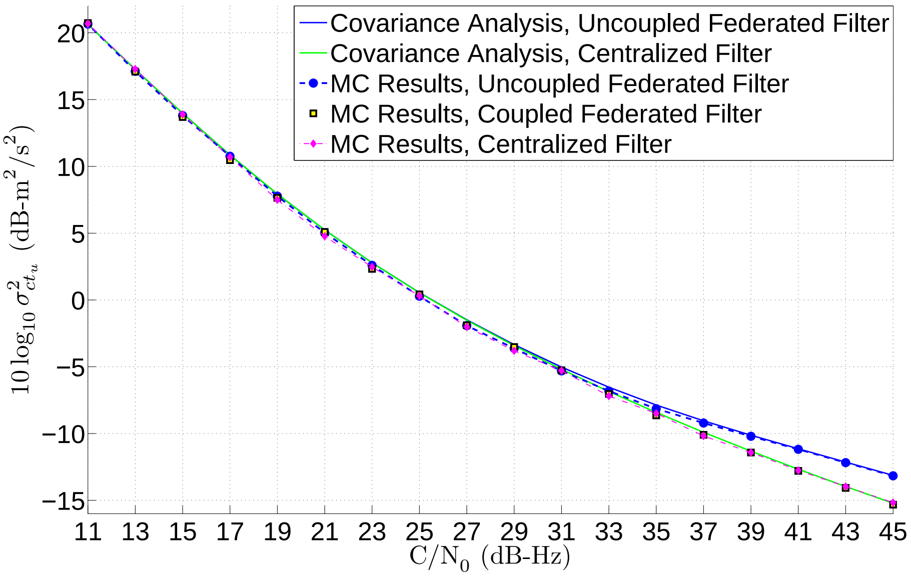

Figure 9.23: Monte Carlo Simulation Results for the Clock Bias State with Eleven Satellites and Tactical Grade IMU

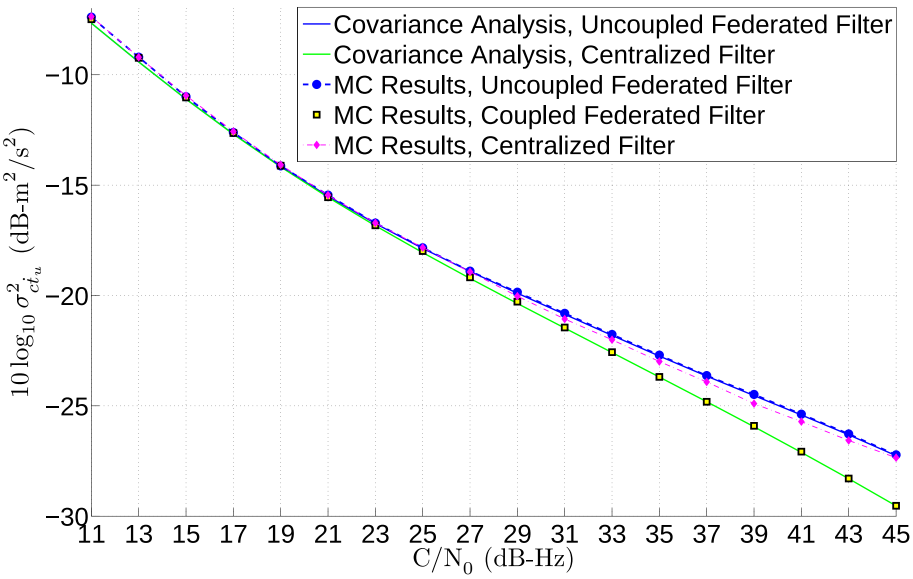

Figure 9.24: Monte Carlo Simulation Results for the Clock Drift State with Eleven Satellites and Tactical Grade IMU

9.4.2 Five Satellite and Automotive Grade IMU Monte Carlo Simulation Results

Figures 9.25 through 9.28 show the results of the Monte Carlo simulations for the five satellite and automotive grade IMU scenario for the x-axis position and velocity states and the clock bias and drift states. Overall the results from the Monte Carlo simulations match the predictions from the covariance analysis. However, the clock drift estimates of the centralized filter are again poorer than predicted. Additional plots for the Monte Carlo simulation results are included in Appendix C.

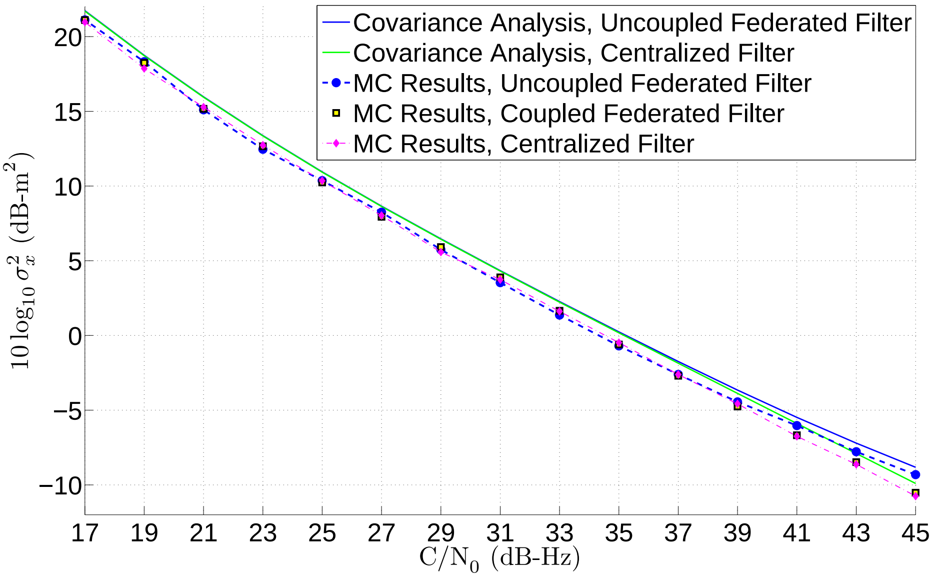

Figure 9.25: Monte Carlo Simulation Results for the X-Axis Position State with Five Satellites and Automotive Grade IMU

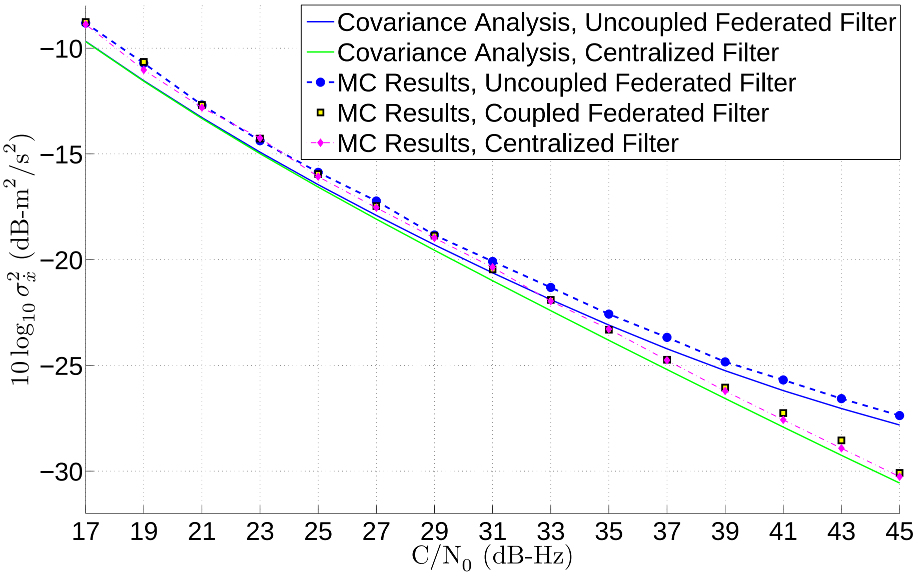

Figure 9.26: Monte Carlo Simulation Results for the X-Axis Velocity State with Five Satellites and Automotive Grade IMU

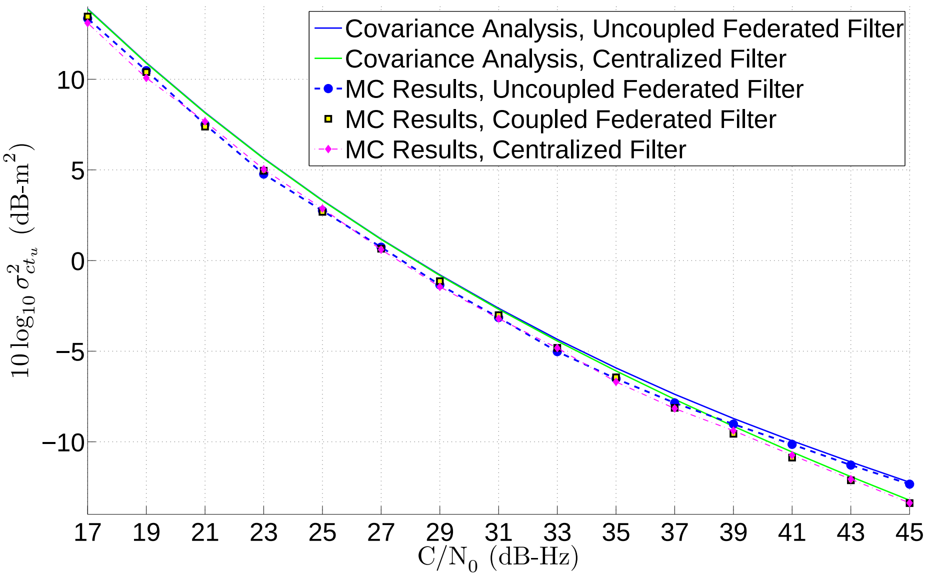

Figure 9.27: Monte Carlo Simulation Results for the Clock Bias State with Five Satellites and Automotive Grade IMU

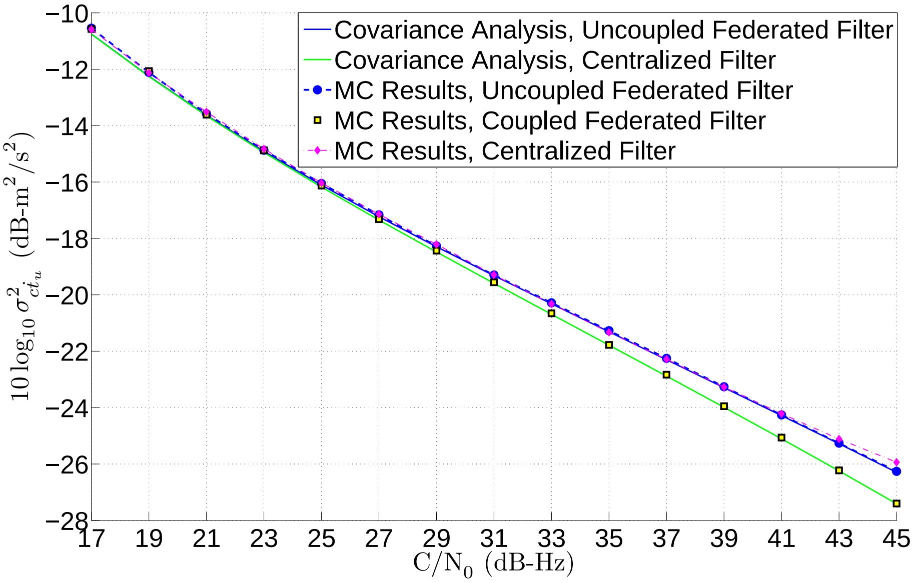

Figure 9.28: Monte Carlo Simulation Results for the Clock Drift State with Five Satellites and Automotive Grade IMU

9.5 Conclusion

The performance of three deeply integrated GPS/INS algorithms are analyzed in this chapter. The DI algorithms are differentiated by how the Kalman filter is implemented in each of them. One architecture uses a single filter to simultaneously track the GPS signals and estimate the navigation states of the receiver. The other two algorithms use a federated filtering architecture that divides the tasks of the central filter. In both federated architectures, a master filter is used to estimate the full navigation state vector and process measurements from the IMU. The states of the master filter are used to predict the incoming GPS signals. A series of smaller residual filters process the pseudorange and pseudorange-rate residuals that are produced in each channel of the receiver. The residual filters act to estimate the errors in the predictions made by the master filter. At regular intervals, the estimates of the residual filters are used as measurements by the master filter. The master filter in the federated architectures only uses the estimates of the residual filters as measurements and not the raw GPS pseudorange and pseudorange-rate residuals.

The two federated filtering architectures use identical master filters. However, the residual filters used by the algorithms differ. One of the federated architectures allows information to be shared between the channels of the receiver (i.e. vector tracking). In contrast, the other federated algorithm does not allow information sharing among the channels of the receiver. The residual filters are designed using the pseudorange-state and position-state formulations of vector tracking.

The three DI algorithms are identical except for the different filter architectures. This allows the effects of using a federated filtering architecture on DI's performance to be isolated. Furthermore, the effects of using a federated architecture with and without vector tracking can also be studied.

Covariance analysis of the three architectures is used to compare their performance. The performance of the algorithms is studied over a range of C/N_0 ratios. Two scenarios are used in the covariance analysis. An eleven satellite, tactical grade IMU scenario and a five satellite, automotive grade IMU scenario are considered. A TCXO grade oscillator is used in both scenarios. The results of the covariance analysis in both situations show the same characteristics in the performance of the three algorithms. The centralized architecture and the federated architecture with vector tracking yield essentially identical performance. The federated architecture without vector tracking performs poorer by comparison. However, the difference in the performance of the algorithms decreases with decreasing C/N_0 ratio. At low C/N_0 ratios, the difference in performance of the three algorithms is indistinguishable.

Chapter 10

Live Sky Test Results Using Vector Tracking

10.1 Introduction

This chapter verifies the ability of the vector tracking algorithms to function in adverse environments. The vector tracking algorithms are tested using data collected from an environment with dense foliage (having widely fluctuating signal levels) and from an urban canyon type environment. The data for the vector tracking algorithms is collected and stored using a software GPS receiver. Data is also collected from the same antenna with a NovAtel ProPak-V3 receiver for comparison. The performance of the vector tracking algorithms is then compared to that of the NovAtel receiver in the various scenarios.

10.2 Vector Tracking Test Results in Dense Foliage

The ability of the VDFLL algorithm to track weak GPS signals is demonstrated using data collected from Auburn University's Mini Baja racing course. The Mini Baja course is a narrow race track through an area of dense foliage. It is located adjacent to the National Center for Asphalt Testing (NCAT) in Opelika, AL. Figure 10.1 shows a bird's eye view of the Mini Baja course. The race course itself is outlined in red.

The GPS signals are attenuated by the foliage along the race course and their power levels fluctuate wildly. The majority of receivers that have been used to navigate about the course have experienced significant periods of GPS outages. The Mini Baja course presents a challenging environment for GPS receivers and is chosen for this reason to test the vector tracking algorithms.

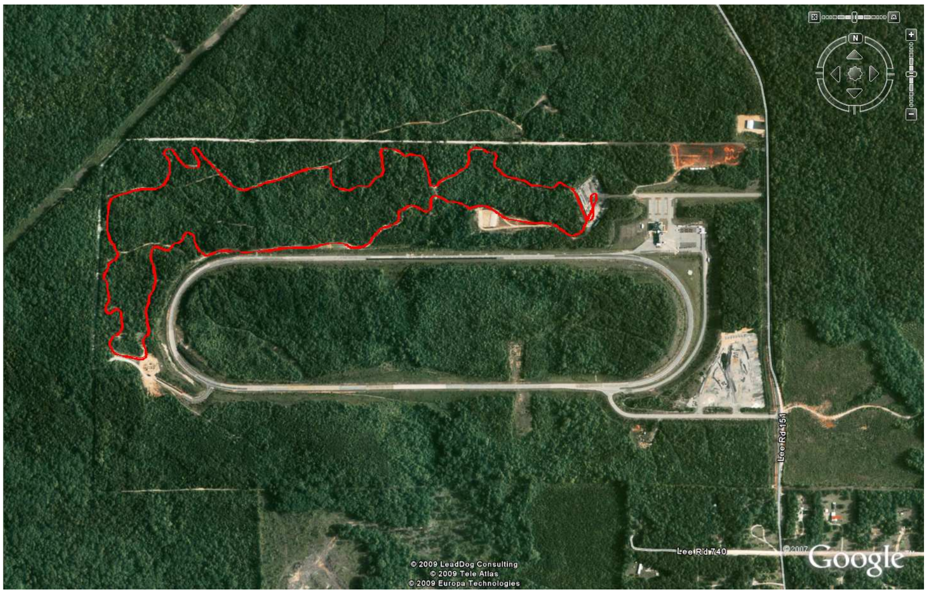

Figure 10.1: Mini Baja Course at NCAT (from Google Earth)

The data for the VDFLL was collected using a NordNav receiver. The RF signals from the satellites were downconverted, sampled, and saved by the NordNav RF front-end. The NordNav front-end uses an Intermediate Frequency (IF) of 4.1304 MHz and a sampling frequency of 16.3676 MHz. A two bit sample resolution was used in the IF collection. The sampled data was post-processed using MATLAB. A NovAtel ProPak-V3 receiver was used as a comparison to the VDFLL. Both receivers were mounted on an All Terrain Vehicle (ATV) and driven around the Mini Baja course. Figure 10.2 shows the ATV and setup used to collect the data. Both the receivers used the same NovAtel antenna, mounted to the top of the ATV. The NovAtel and VDFLL data were collected on the same runs around the Mini Baja course. The path taken by the ATV when collecting data for the different trials is approximately the same.

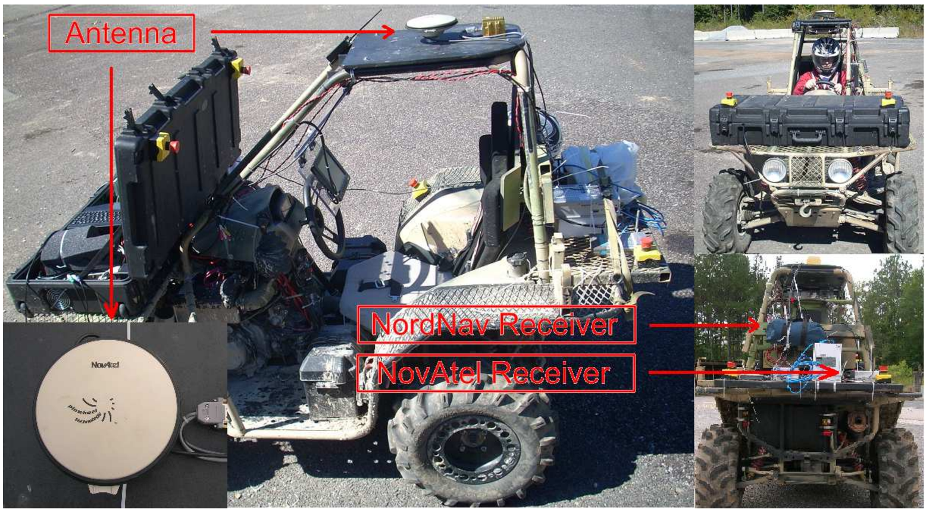

Figure 10.2: ATV and Hardware Used to Collect Data

A third-order VDFLL (as described in Chapter 5.3) was used to process the recorded IF data. The states of the filter are the receiver's position, velocity, and acceleration in the ECEF frame and the receiver's clock bias and clock drift. The acceleration states are modeled as being driven by separate white noise processes. A variance of 1 m^2/s^4 was used as the variance of the processes. The C/N_o ratios of the GPS signals were estimated using the techniques described in Appendix B. The VDFLL used the same pseudorange and pseudorange-rate discriminators used for the analysis in Chapter 7. An integrate and dump period of 20 milliseconds was used for the pseudorange discriminator and two consecutive 10 millisecond integrate and dump periods were used for the pseudorange-rate discriminator. A full chip Early-to-Late correlator spacing is used by the VDFLL.

The first data set from the Mini Baja course was collected on Monday, October 20, 2008 at approximately noon, eastern time. At the time the data was collected there were ten visible GPS satellites. Figure 10.3 shows the azimuth and elevation of the visible satellites.

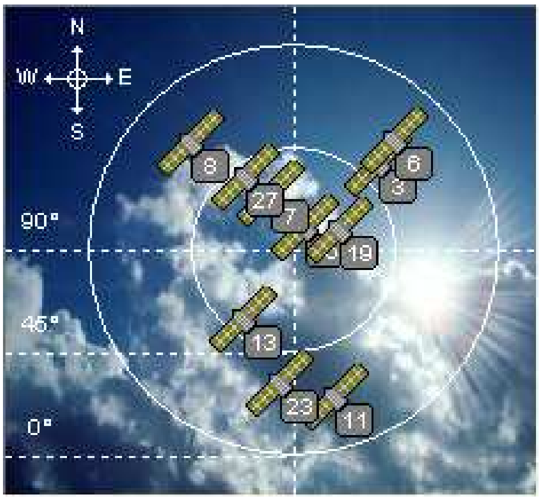

Figure 10.3: Skyplot of Visible Satellites for the First Data Set (from NordNav Receiver)

The estimated C/N_o ratios by the VDFLL are shown in Figure 10.4. Examining Figures 10.3 and 10.4, the lowest C/N_o ratios belong to the satellites closest to the horizon. Accordingly, the satellites with the weaker signals vary most throughout the race course. Since they are closest to the horizon, they tend to pass through more foliage than the satellites closer to zenith.

Figures 10.5 and 10.6 show the VDFLL's estimates of the receiver's velocity and acceleration during the first data collection interval, respectively. The position estimates of the VDFLL and NovAtel are shown in Figure 10.7. The position estimates have been converted from the ECEF frame into a local East-North-Up (ENU) coordinate frame. The East and North components are plotted to demonstrate the difference in coverage between the VDFLL and NovAtel receivers. The ATV started the course at the ENU coordinates (0 m, 0 m) (at the far right side of the figure), traversed the course going counter-clockwise, and then returned to roughly its starting position. Figures 10.8 and 10.9 show magnified versions of the position results.

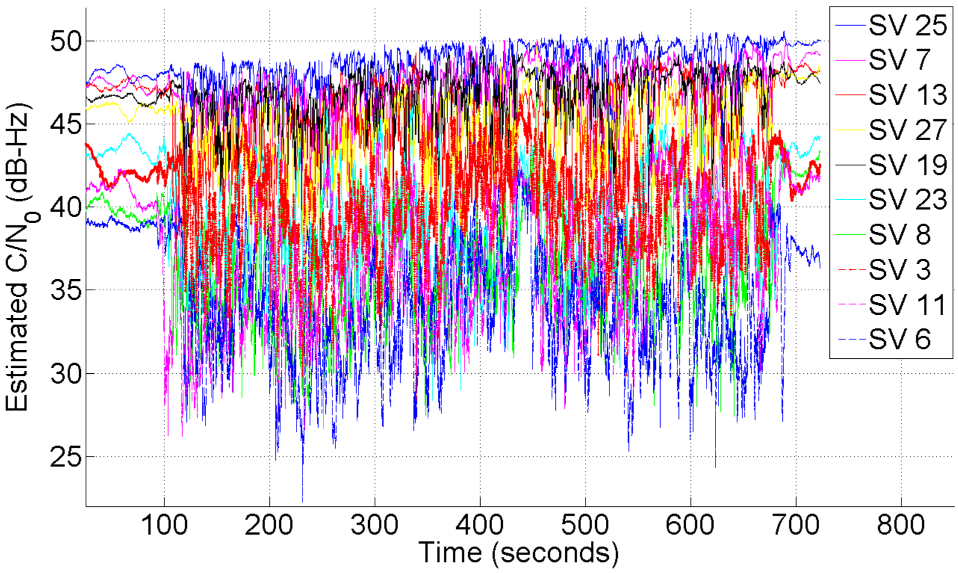

Figure 10.4: Estimated C/N_o Ratios for the First Data Set

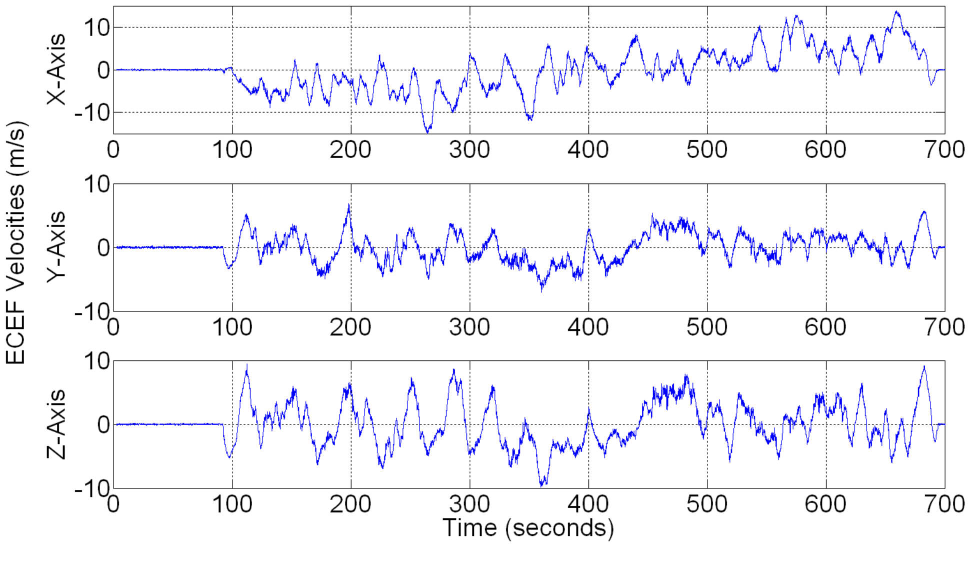

Figure 10.5: Estimated Velocities from the VDFLL for the First Data Set

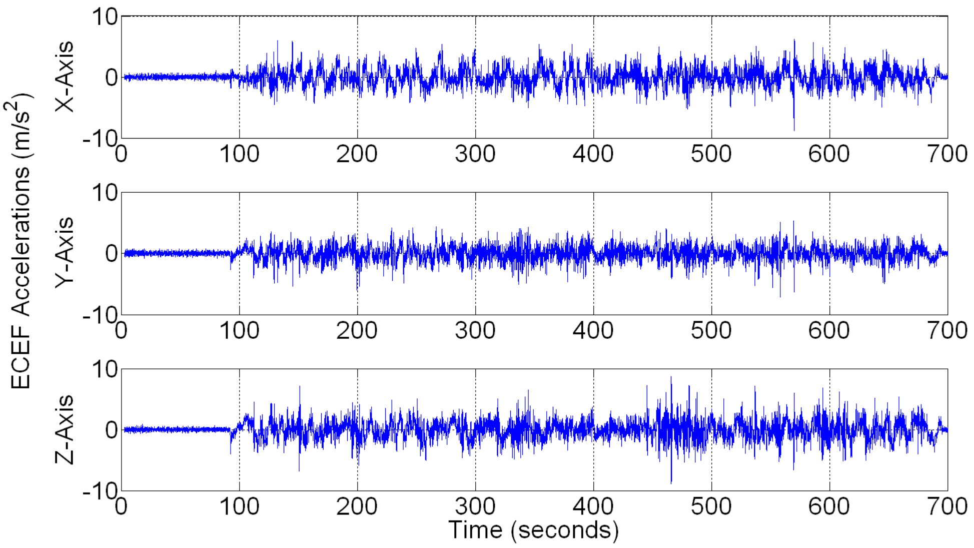

Figure 10.6: Estimated Accelerations from the VDFLL for the First Data Set

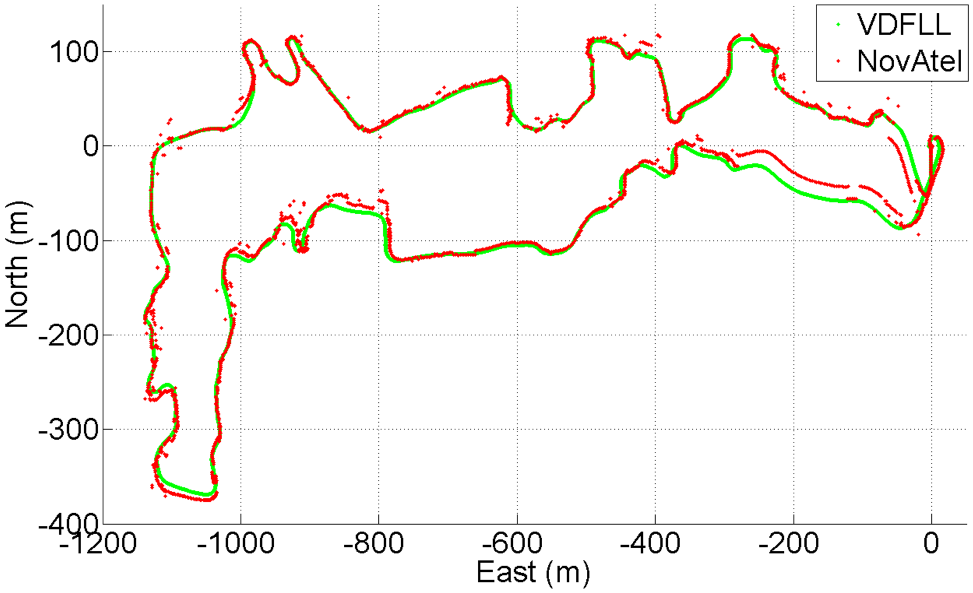

Figure 10.7: Position Estimates from the VDFLL and NovAtel for the First Data Set

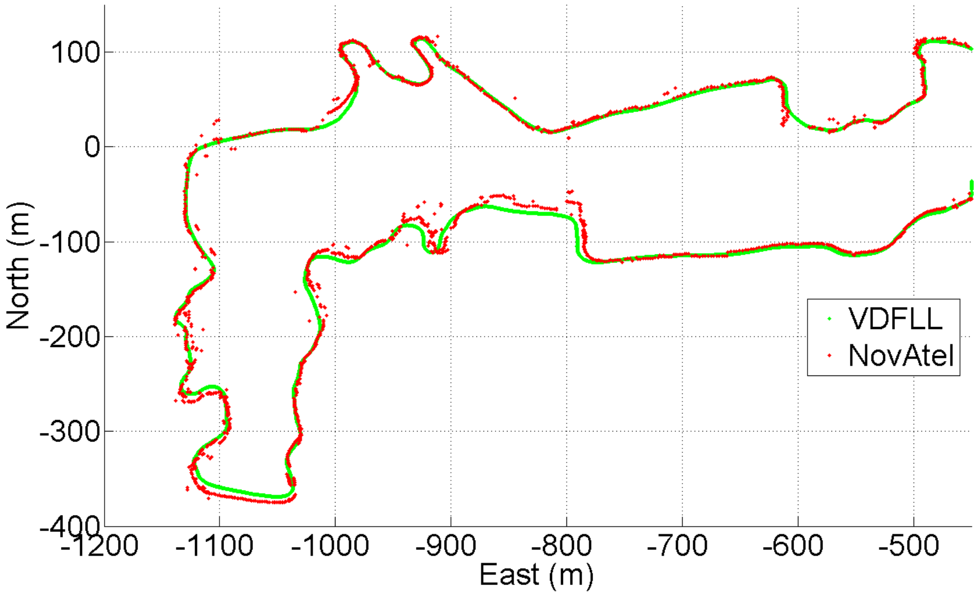

Figure 10.8: Position Estimates from the VDFLL and NovAtel for the First Data Set

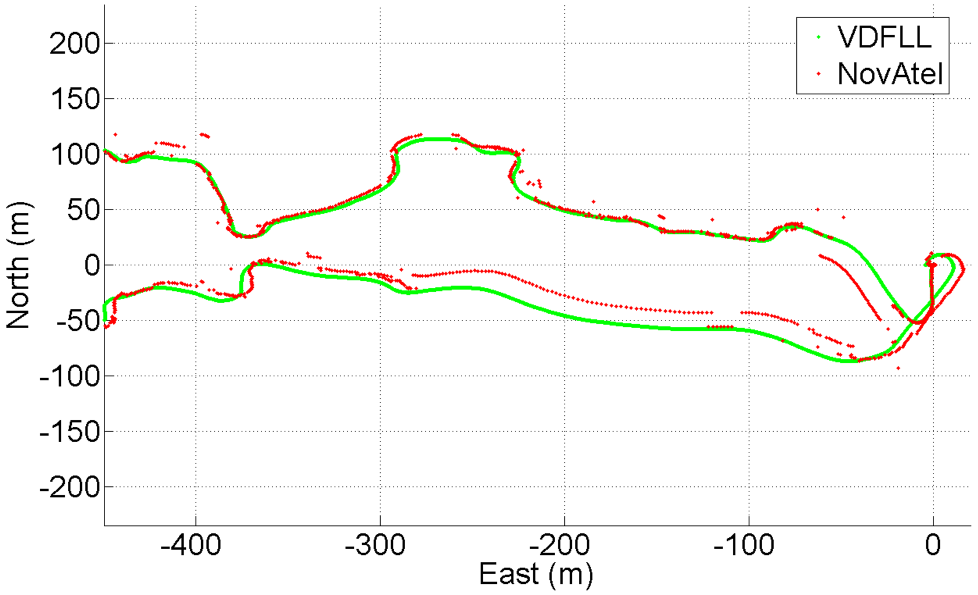

Figure 10.9: Position Estimates from the VDFLL and NovAtel for the First Data Set

Examining Figures 10.7 through 10.9, the VDFLL provides continuous coverage throughout the course and the NovAtel reports a receiver position during the majority of it. However, the NovAtel receiver reports multiple aberrant position solutions and the figures show significant jumps in its reported trajectory. This is seen in Figure 10.9 at the start and end of the course. At the start of the course, the reported position solutions lie essentially on top of each other. Then, the NovAtel reported position jumps by about 20 meters to the west of that given by the VDFLL. When the ATV enters the first turn at about (-60 m, 30 m), the NovAtel's position estimate jumps back, so as to coincide with the VDFLL's estimated position. Toward the end of the course, from about (-300 m, -20 m) to (-50 m, -80 m), the same type of phenomenon is observed. The reported positions differ by over 15 meters, except briefly when the NovAtel's reported trajectory shifts south to more closely coincide with the VDFLL's estimated trajectory.

The second data set from the Mini Baja course was recorded on Monday, October 20, 2008 at approximately 3 o'clock in the afternoon, eastern time. At the time the data was collected there were six visible GPS satellites. Figure 10.10 shows the azimuth and elevation of the visible satellites. The estimated C/N_o ratios by the VDFLL are shown in Figure 10.11. The weakest signals again correspond to the satellites closest to the horizon. Unlike the first data set though, only three satellites (SV's 4, 17, and 28) are in the vicinity of zenith.

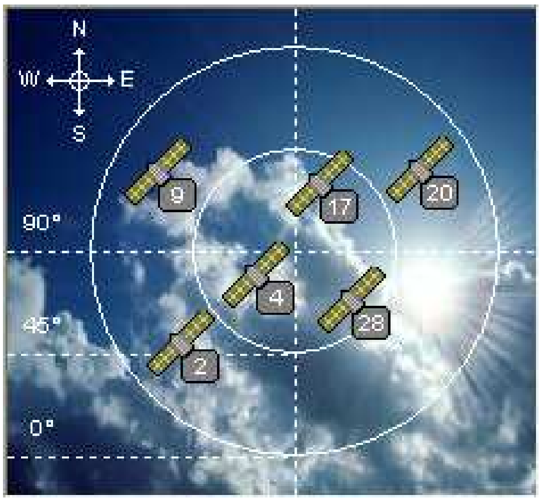

Figure 10.10: Skyplot of Visible Satellites for the Second Data Set (from NordNav Receiver)

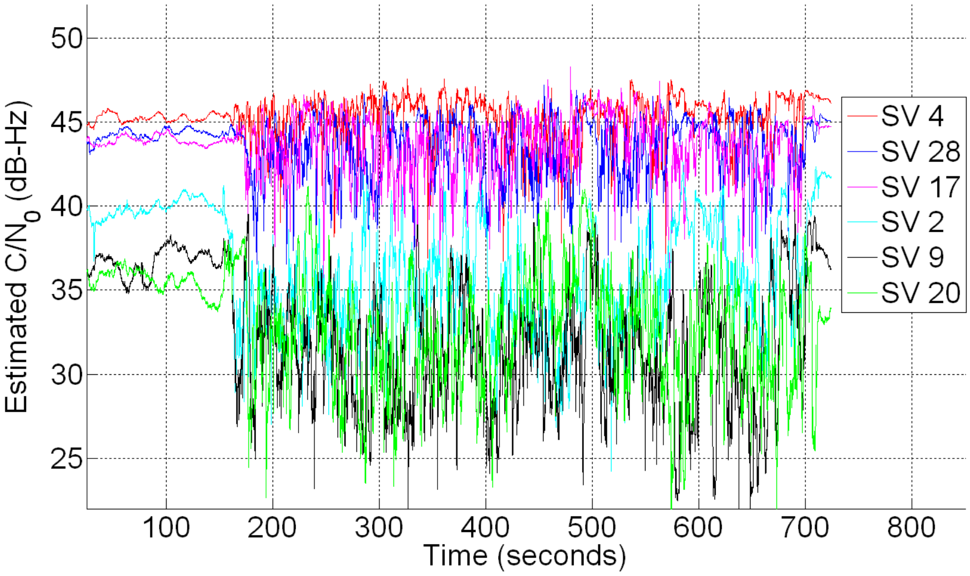

Figure 10.11: Estimated C/N_o Ratios for the Second Data Set

Figures 10.12 and 10.13 show the VDFLL's estimates of the receiver's velocity and acceleration during the second data collection interval, respectively. The position estimates of the VDFLL and NovAtel are shown in Figure 10.14 in the local ENU frame. The ATV traversed approximately the same course for the second data collection period as the first. Figures 10.15 and 10.16 show magnified versions of the position results.

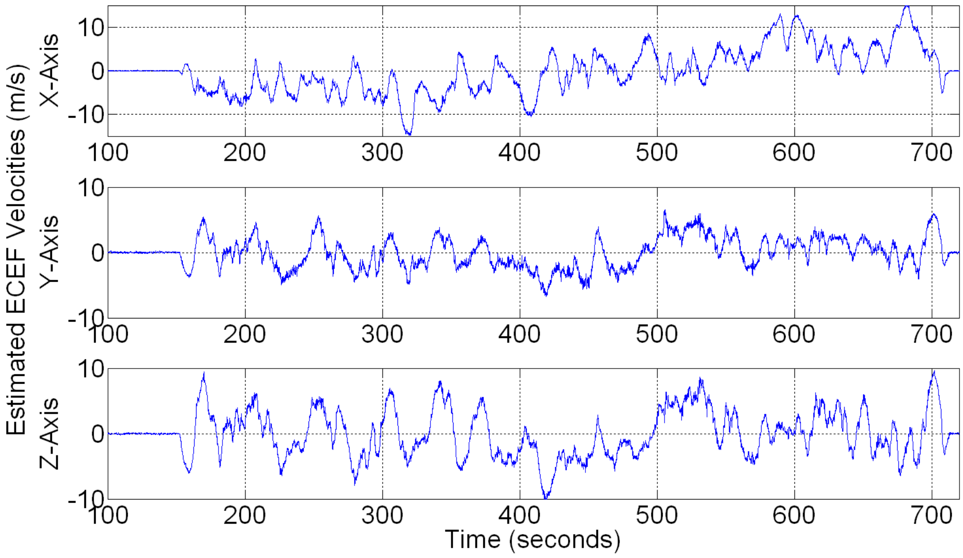

Figure 10.12: Estimated Velocities from the VDFLL for the Second Data Set

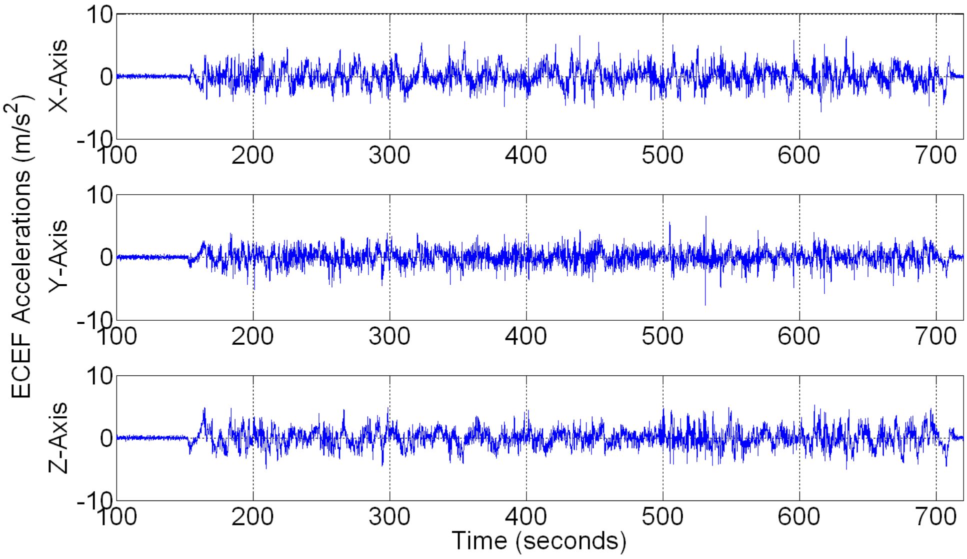

Figure 10.13: Estimated Accelerations from the VDFLL for the Second Data Set

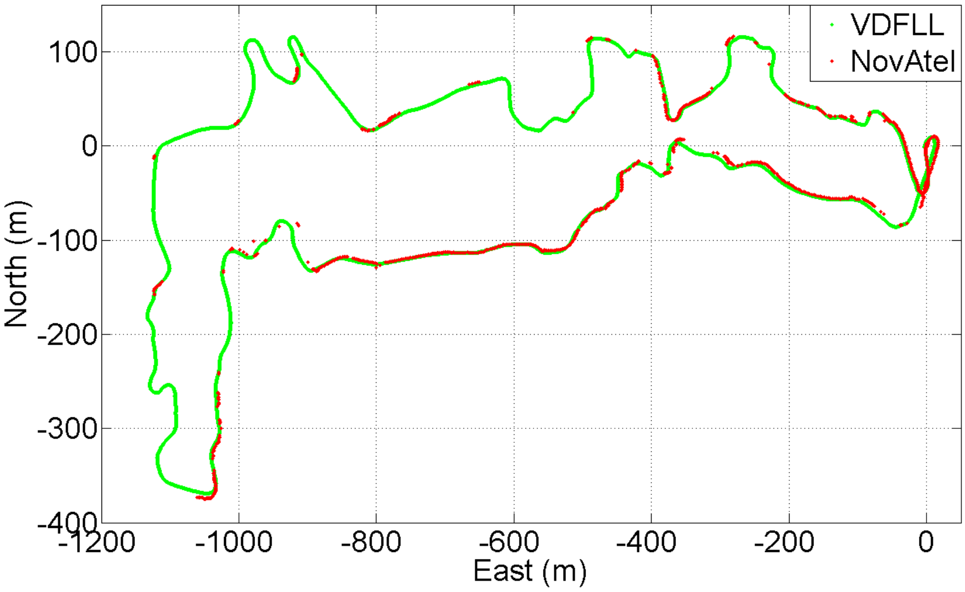

Figure 10.14: Position Estimates from the VDFLL and NovAtel for the Second Data Set

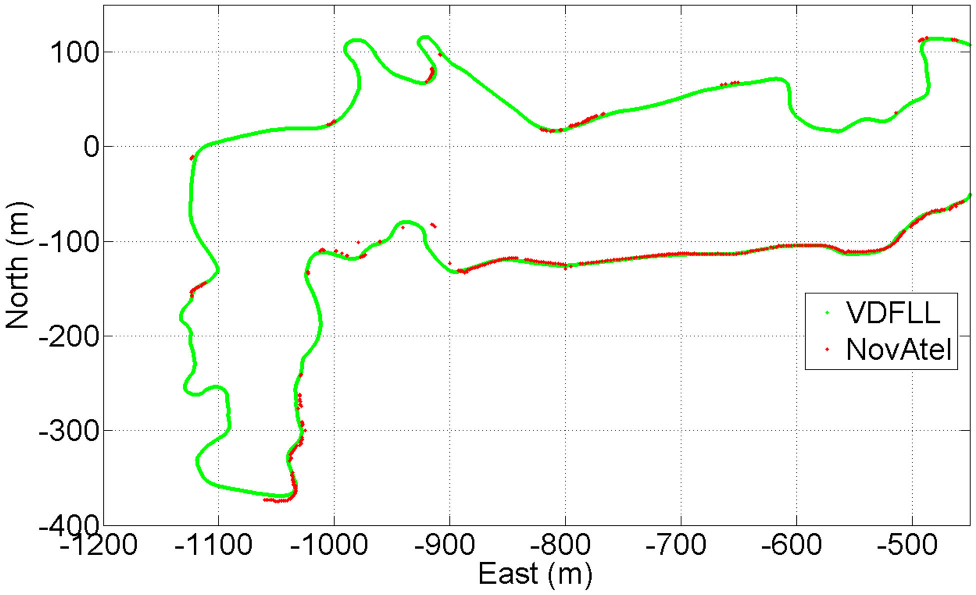

Figure 10.15: Position Estimates from the VDFLL and NovAtel for the Second Data Set

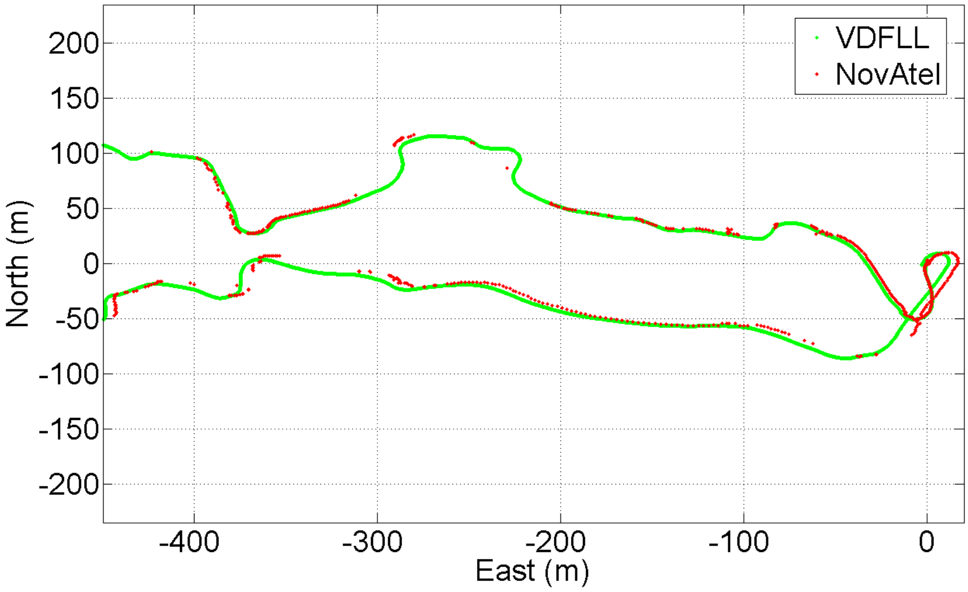

Figure 10.16: Position Estimates from the VDFLL and NovAtel for the Second Data Set

Figures 10.14 through 10.16 show the NovAtel receiver failing to produce position estimates during the majority of the course. In contrast, the VDFLL again provides continuous coverage throughout the entire course. The difference in coverage can especially be seen in Figure 10.15. The NovAtel provides very sparse coverage and distances of over 100 meters are traversed with no position estimates.

The difference in the NovAtel's performance from the first data set to second is primarily due to the number of visible satellites and their geometry. In the first data set, there were five satellites close to being overhead. Conversely, only three satellites were close to zenith in the second data set and a minimum of four satellites are required for the receiver to calculate its position. The satellites closer to zenith are impacted less by the foliage and are necessary for the NovAtel to provide continuous coverage.

The lack of truth data makes declaring which algorithm is more accurate very difficult. However, from examining the results it is clear that the VDFLL is more robust in the face of dense foliage. It provided continuous coverage in both cases and did not exhibit the rapid jumps in its estimated positions. At the very least, the VDFLL can provide a coarse estimate of the receivers position. Its estimates can also be used to fore go the re-acquisition process when the satellite signals are obstructed, and then reappear.

10.3 Vector Tracking Test Results in an Urban Canyon

The ability of the vector tracking algorithms to function in an urban canyon environment was also studied. Downtown Atlanta was used as the urban canyon environment. Data was collected using the NordNav software receiver. A two bit sample resolution was again used in the IF data collection. A NovAtel ProPak-V3 receiver was also used as a comparison for the VDFLL. Both receivers were mounted in the trunk of an Infiniti G35. Both the receivers used a NovAtel antenna that was mounted to the top of the automobile. The NovAtel and VDFLL data were collected on the same runs through downtown Atlanta. The NordNav IF data was post-processed using MATLAB.

The course taken by the automobile started in a parking lot south of downtown. The vehicle then drove north on interstate I-75/85 to just north of downtown. The car exited the interstate and then drove south on Peachtree St. NE. The stretch of Peachtree St. NE that was traversed contains many urban canyon scenarios. Figure 10.17 shows an example of the type of environment encountered in downtown Atlanta.

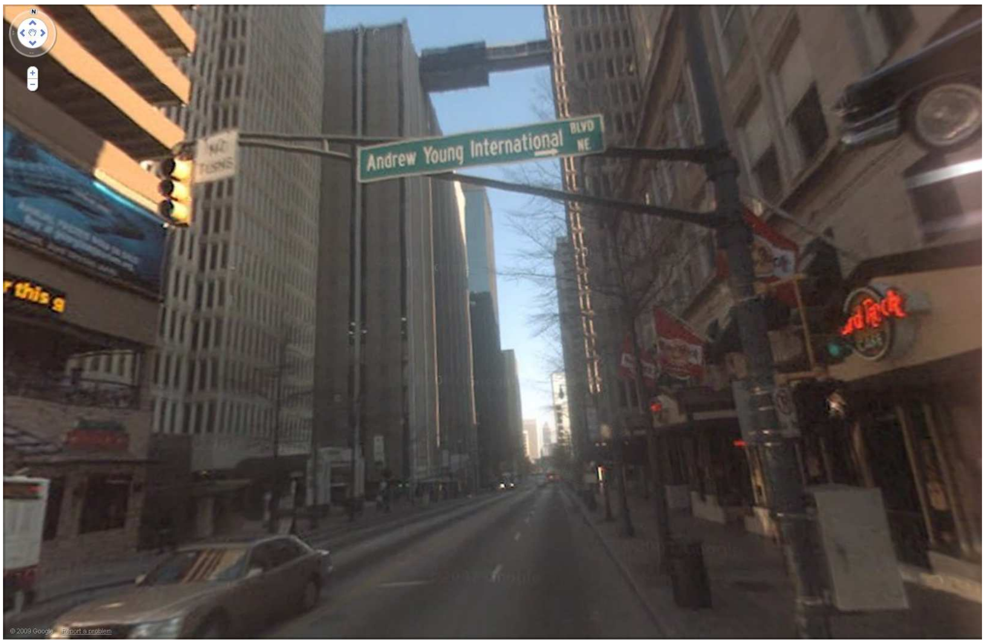

Figure 10.17: Urban Canyon at the Intersection of Peachtree ST. NE and Andrew Young International Blvd. NE, Looking North (from Google Street View)

Buildings on either side of the street obstruct the satellite signals from low and mid elevation satellites. The sides of the buildings also cause reflections of the GPS signals to be received by the antenna (i.e. multipath) [Soloviev and van Graas, 2008]. The high-rise buildings on either side of the street impede the performance of most GPS receivers.

A third-order VDFLL was used to process the recorded IF data. The states of the filter are the receiver's position, velocity, and acceleration in the ECEF frame and the receiver's clock bias and clock drift. The acceleration states are modeled as being driven by separate white noise processes. A variance of 1 m^2/s^4 was used as the variance of the processes. The C/N_o ratios of the GPS signals were estimated using the techniques described in Appendix B. The VDFLL used the same pseudorange and pseudorange-rate discriminators used for the analysis in Chapter 7. An integrate and dump period of 20 milliseconds was used for the pseudorange discriminator and two consecutive 10 millisecond integrate and dump periods were used for the pseudorange-rate discriminator. A full chip Early-to-Late correlator spacing is used by the VDFLL.

The data was collected on Sunday, June 7, 2009 at approximately one o'clock, central time. At the time the data was collected there were seven visible GPS satellites. Figure 10.18 shows the azimuth and elevation of the visible satellites. The estimated C/N_o ratios of the visible satellites are shown in Figure 10.19 for the entire data collection period.

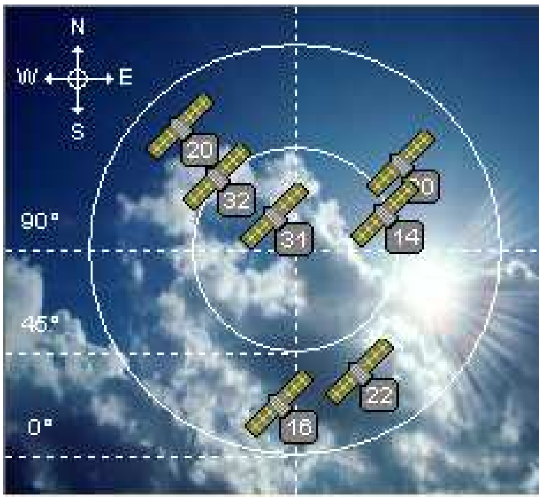

Figure 10.18: Skyplot of Visible Satellites for the Atlanta Data (from NordNav Receiver)

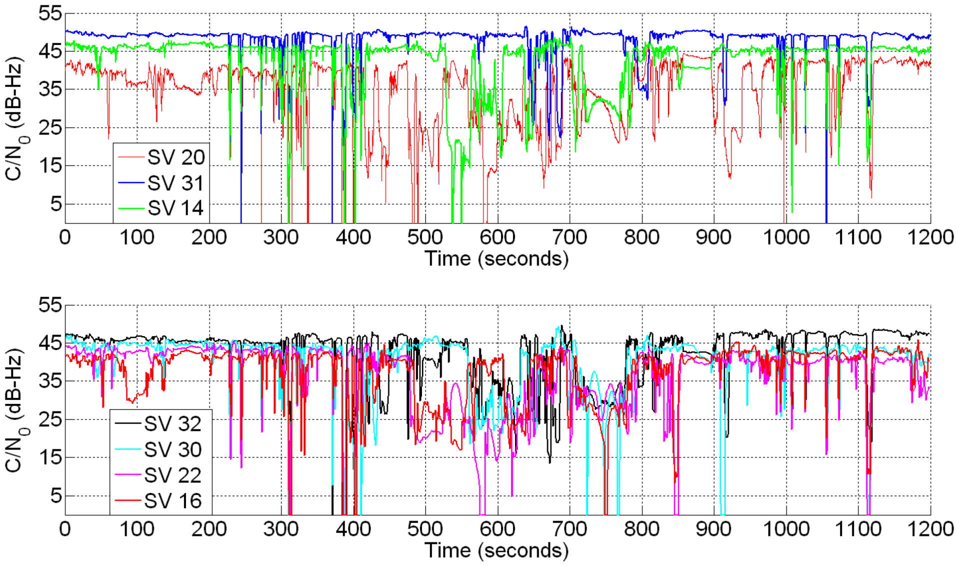

Figure 10.19: Estimated C/N_o Ratios for the Total Atlanta Data Set

The position estimates of the VDFLL and NovAtel for the entire data set are shown in Figure 10.20. The position estimates have been converted from the ECEF frame into a local East-North-Up (ENU) coordinate frame. The East and North components are plotted to demonstrate the difference in coverage between the VDFLL and NovAtel receivers. The vehicle started the course at the ENU coordinates (0 m, 0 m) (at the bottom of the figure) and then traversed the plotted course counter-clockwise. At the end of the data set, the car returned to its starting position.

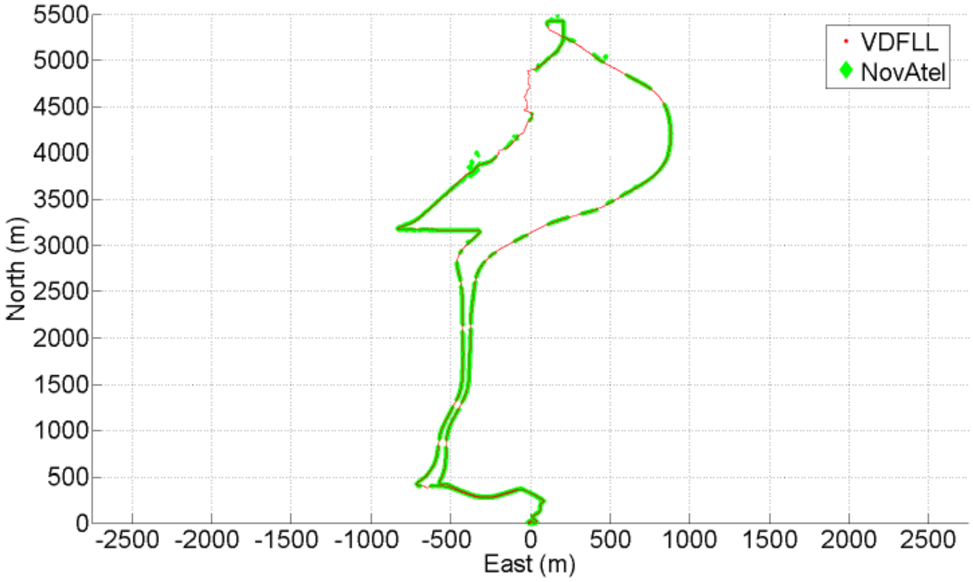

Figure 10.20: Position Estimates of the VDFLL and NovAtel

It is apparent from Figure 10.20 that there is a significant difference in the coverage offered by the VDFLL and the NovAtel. The areas where the NovAtel does not report a position correspond to the underpasses and urban canyon environments the vehicle encountered. The outages of the NovAtel can be correlated to the physical environment the car was in by overlaying the position results in Figure 10.20 on a map of the terrain. Figure 10.21

shows the positions reported by the VDFLL and the NovAtel on a map of downtown Atlanta. The positions reported by the NovAtel are plotted with red way points and the position results of the VDFLL are plotted with a green line.

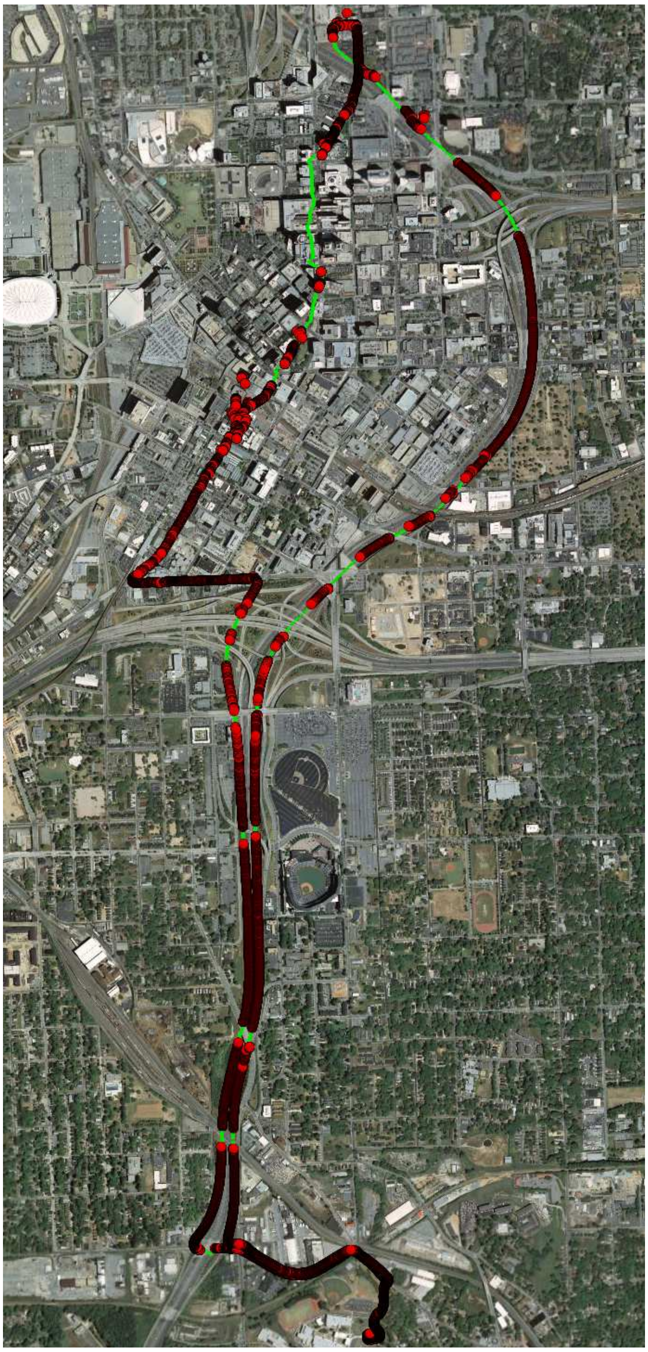

Figure 10.21: Position Estimates of the VDFLL and NovAtel Overladed on a Map of Downtown Atlanta (from Google Earth and GPS Visualizer)

Figure 10.21 shows the disparities between the VDFLL's and NovAtel's coverage in relation to the environment were the data was taken. While the vehicle is traveling on the interstate, the loss in the NovAtel's coverage is caused by overpasses and by tunnels. When the vehicle enters a tunnel, the GPS signals are significantly attenuated and the VDFLL effectively ignores measurements. The VDFLL propagates its states forward until the vehicle emerges from the tunnel. When the satellite signals return, the VDFLL corrects the errors in its states that have accumulated over the signal outage.

The disparities in coverage while in downtown are caused by the presence of high-rise buildings. The satellite signals are blocked and fluctuate as the car travels past the buildings. Figure 10.22 shows a magnified plot of the position results for the Northern half of the course. The magnified plot makes connecting the outages of the NovAtel to the physical features of the highways easier. The outages on the right side of the plot (while the car is driving north on the interstate) are clearly caused by the vehicle going under overpasses and into tunnels.

The position results while the car is driving through downtown are shown on the left side of the plot. This data was collected while the car was driving south on Peachtree St. NE. A significant outage in the NovAtel is seen in the north-west quadrant of the plot. The outage lasts for about 80 seconds and a third of a mile. The corridor where the outage occurred corresponds to the most severe urban canyon that was encountered. The position estimates reported by the VDFLL during the same stretch drift off the street on which the car was driving. The largest error between the position reported by the VDFLL and the street is about 40 meters.

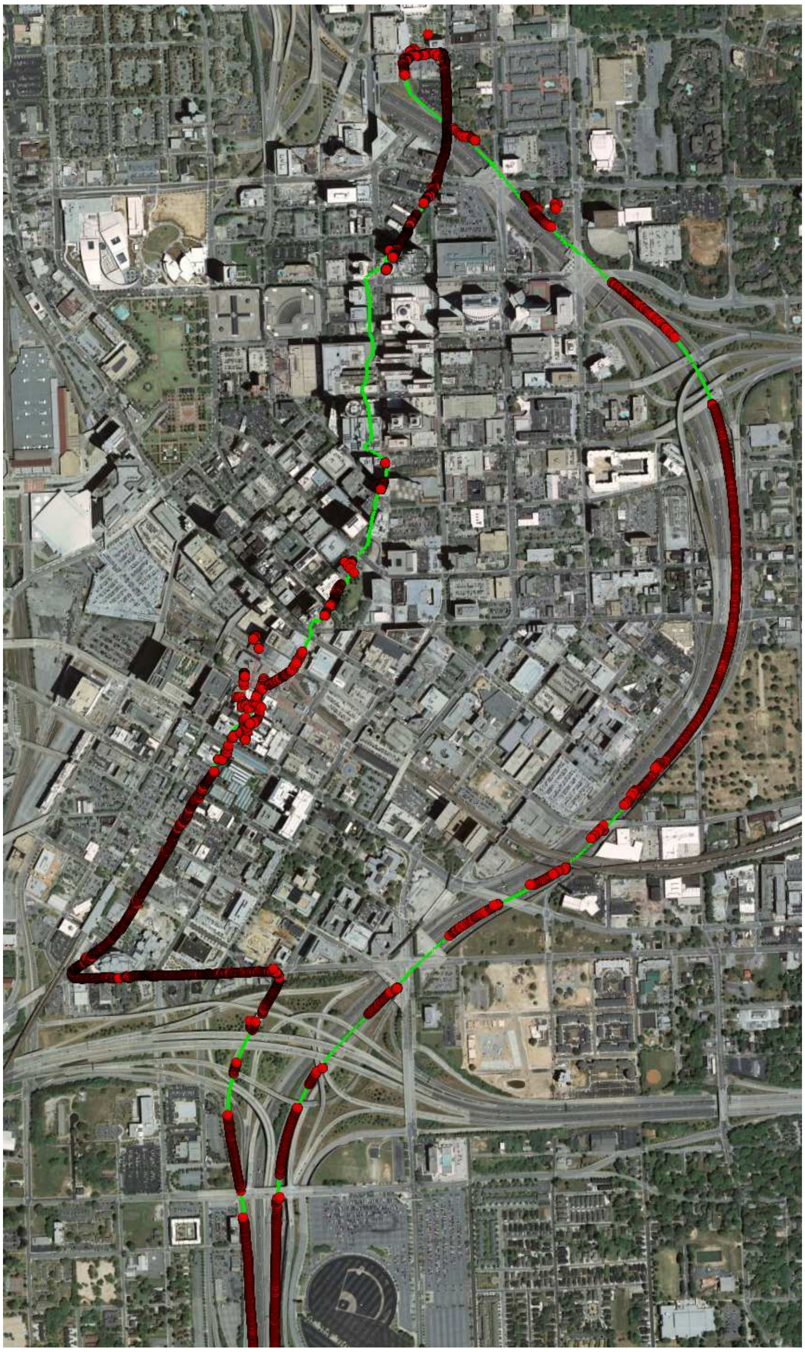

Figure 10.22: Position Estimates from the VDFLL and NovAtel for the Second Data Set (from Google Earth and GPS Visualizer)

Figure 10.23 shows a magnified plot of the position results during the extend outage of the NovAtel. The position results of the VDFLL in blue correspond to the period when the vehicle is inside the severe urban canyon. The vehicle is inside the urban canyon from about 548 to 628 seconds into the data set. The C/N_0 ratios estimated by the VDFLL during this period are shown in Figure 10.24. During the stretch of severe urban canyon, satellite 31 has a consistently high C/N_0 ratio and satellite 16 has the second highest consistent C/N_0 ratio. Referring back to Figure 10.18, satellite 31 is close to being overhead and satellite 16 is close to the southern horizon. The satellites on the east and west horizon are blocked by the buildings on either side of the street. Consequently, the VDFLL's estimates of the receiver's position in the east-west direction are poor. South of the extended outage, there is a cluster of erroneous position estimates from the NovAtel. Figure 10.25 shows a magnified plot of this stretch of data.

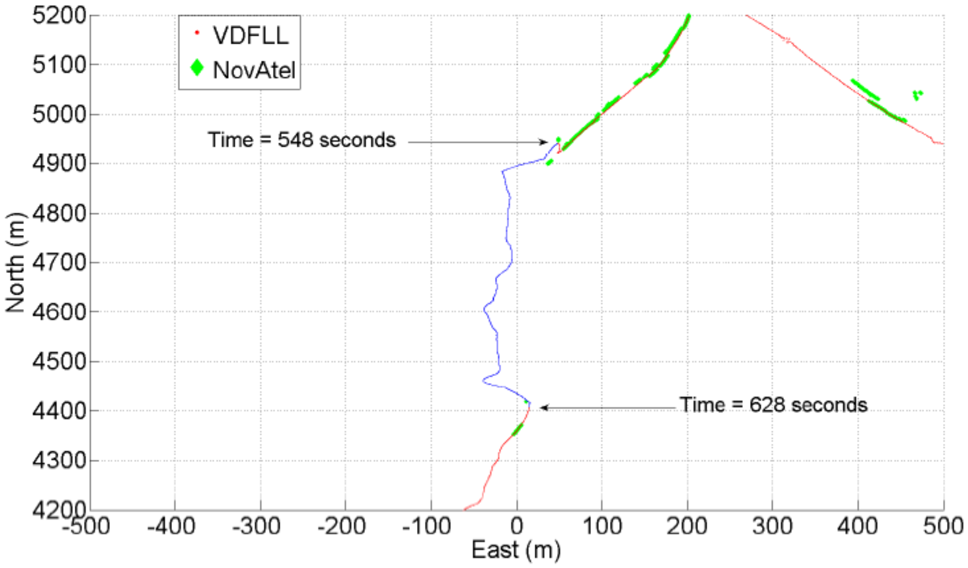

Figure 10.23: Position Estimates from the VDFLL and NovAtel While Inside Extreme Urban Canyon

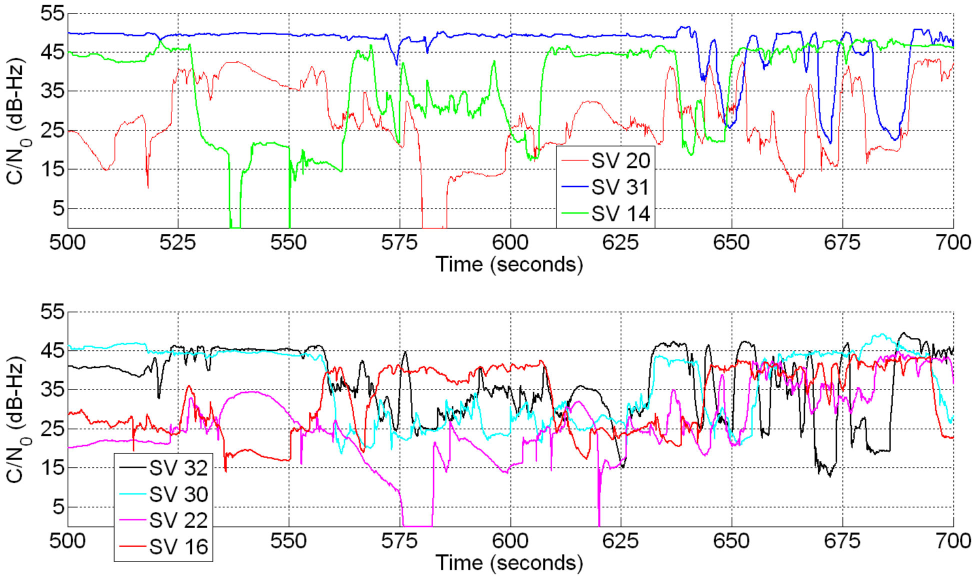

Figure 10.24: Estimated C/N_o Ratios While in Urban Canyon

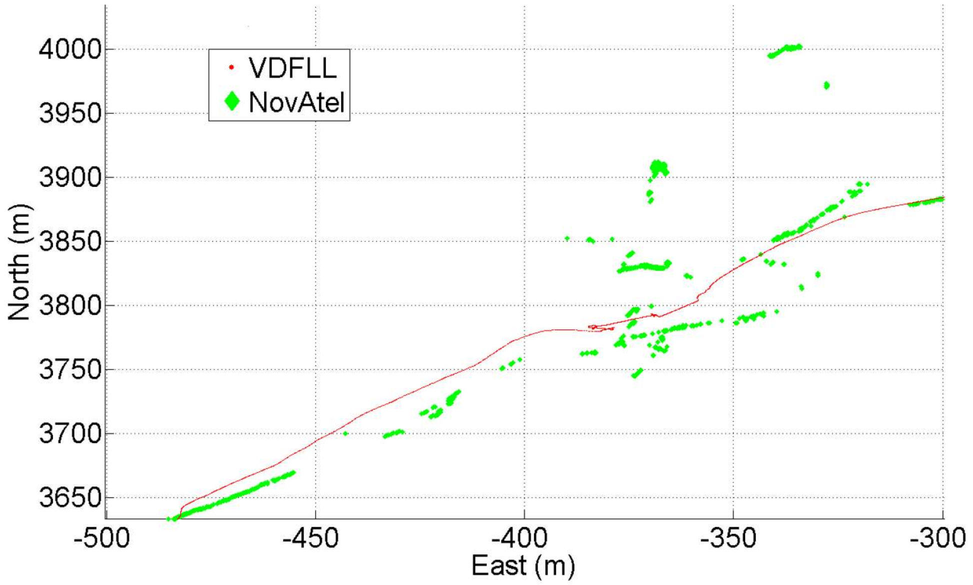

Figure 10.25: Position Estimates from the VDFLL and NovAtel

The errors in the positions reported by the NovAtel are quite large. The positions reported at about (-340 m, 4000 m) are approximately 140 meters from the street traveled by the automobile. Examining the VDFLL results, it's estimates are significantly more consistent than those of the NovAtel.

Figure 10.26 shows the position results from the VDFLL and the NovAtel for the southern half of the course. While the vehicle was on the interstate highway, the NovAtel and VDFLL reported positions are extremely similar. The only times the NovAtel failed to report a position was in the presence of an overpass. At the southern end of the plot, the vehicle is shown exiting the interstate and then turning east. As it passes under the interstate, there is a gap in the position estimates of the NovAtel. At the same time, the VDFLL's estimates drift off as the car passes under the highway. However, once the car is no longer under the interstate, the VDFLL's estimates converge back to the road.

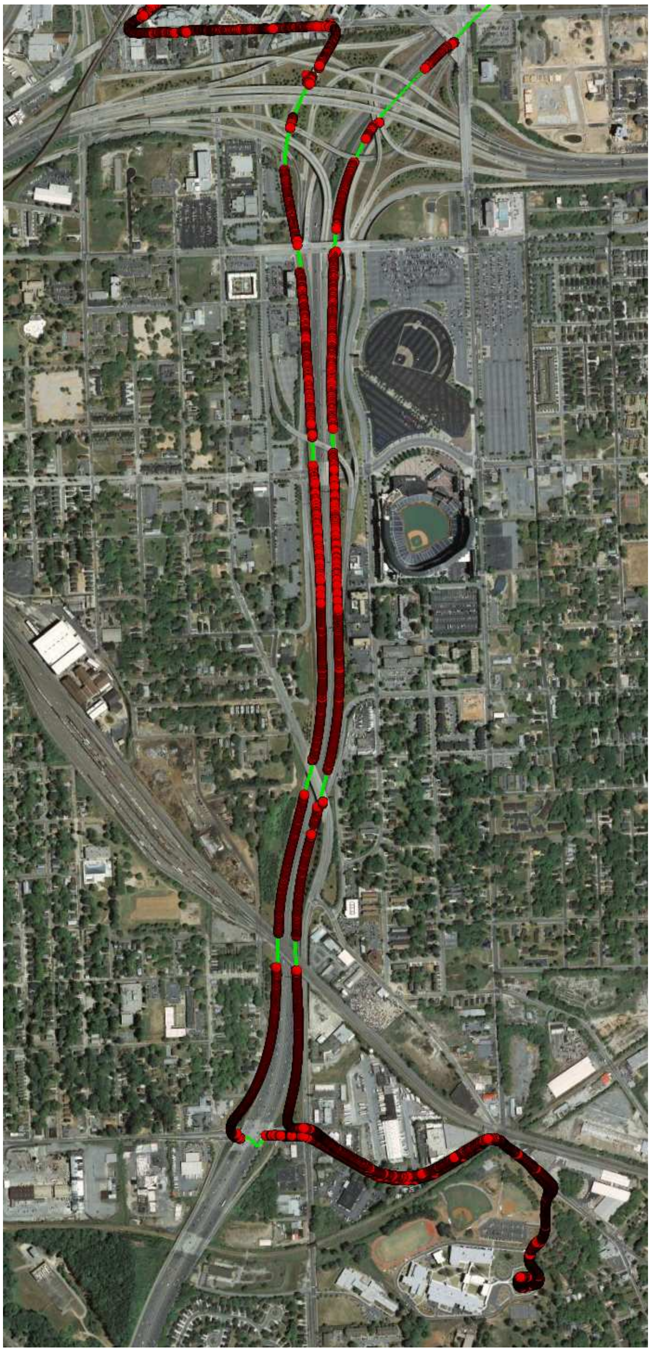

Figure 10.26: Position Estimates of the VDFLL and NovAtel Overladed on a Map of Atlanta (from Google Earth and GPS Visualizer)

10.4 Conclusion

The ability of the vector delay/frequency lock loop to work in challenging environments is demonstrated in this chapter. The VDFLL is implemented in software and is used to process data collected by a RF front-end. The RF front-end is used to capture data in two scenarios with dense foliage and one scenario in an urban canyon. Data is also collected using a NovAtel ProPak-V3 receiver for comparison.

The VDFLL provided continuous coverage in the two environments with dense foliage. In contrast, the NovAtel receiver displayed significant periods of GPS outage in the same scenarios. The position solutions of the NovAtel also contained multiple jumps in reported position of about 15 to 20 meters. The VDFLL also provided continuous coverage throughout the urban canyon scenario. However, its reported positions did drift off the path taken by the receiver during a stretch of particularly extreme urban canyon. During the same stretch, the NovAtel receiver provide no position solutions. Overall, the NovAtel receiver failed to provide navigation solutions during significant portions of the data collection period. The NovAtel receiver also reported erroneous position solutions in error by about 140 meters.

Chapter 11

Conclusion and Future Work

11.1 Conclusion

The research performed in this dissertation investigated the benefits and performance of vector tracking algorithms. A variety of different analysis tools were used to study the vector tracking algorithms and compare them to scalar tracking loops. The reason vector tracking yields superior performance is explained using weighted least squares. Vector tracking essentially reduces the number of unknowns being estimated. In the case N pseudorange measurements, scalar algorithms attempt to estimate N independent pseudorange tracking errors. In contrast, a vector based tracking loop uses the N pseudorange measurements to estimate four unknowns (three position errors and one clock bias error). When more than four satellites are available, vector tracking will generally perform better than scalar tracking. As the number N becomes larger, the number of parameters being estimated by vector tracking remains at four. This effect is similar to averaging, but is different in that the pseudorange measurements are related to the unknowns through the line-of-sight unit vectors. The advantage offered by vector tracking for a typical GPS receiver was calculated by recording the visible satellite constellation at Auburn University over an 18 hour period.

The general concept of vector tracking is manifested in two separate formulations. The position-state formulation uses a Kalman filter that directly tracks the receiver's navigation states (i.e. position, velocity, and clock states). From the estimated navigation states of the filter, all the received satellite signals are predicted. The differences between the predicted and received signals are then used as residuals by the filter. In contrast, the pseudorange-state formulation of vector tracking uses a Kalman filter with the pseudoranges and pseudorange-rates of the visible satellites as its states. The states of the filter are coupled together through the process noise covariance matrix of the filter in the pseudorange-state formulation. In the case of a block diagonal process noise covariance matrix, the pseudorange-state filter reverts back to a scalar tracking algorithm. The performance of the two filters in steady-state were shown to be equivalent.

Using the pseudorange-state filter, the states corresponding to the different satellites are easily decoupled. In this case, no information is shared between the channels of the receiver and the filter is performing scalar tracking. This fact was exploited in order to isolate the improvement in performance brought about by the coupling of the receiver channels. The difference in the performance of the pseudorange-state filter in the coupled and decoupled modes is due solely to the information sharing between the channels.

The loss of lock performance of the Vector Delay/Frequency Lock Loop (VDFLL) was studied using analytical tools and Monte Carlo simulations. The analytical tools take thresholds established for scalar Frequency Lock Loops and then extend them to the vector tracking case. The performance of the VDFLL with different satellite geometries and process noise tunings was compared to scalar FLL's of varying noise bandwidths. The predictions from the rule of thumb analysis were then tested using Monte Carlo simulations. The accuracy of the loss-of-lock predictions for the VDFLL are comparable to the accuracy of the rule of thumb thresholds for the scalar FLL. The effects of C/N_0 ratio estimation were also studied using Monte Carlo simulations. The same algorithms were simulated with and without *a priori* knowledge of the C/N_0 ratios of the satellite signals. The difference in the performance of the algorithms was altered at most by about .8 dB.

The rule of thumb analysis tools and Monte Carlo simulations were used to determine the improvement in loss-of-lock performance from vector tracking. In various scenarios, the performance of the VDFLL was compared to the performance of a scalar tracking algorithm. The scalar tracking algorithm was designed by taking a pseudorange-state filter with identical tunings to the VDFLL and then decoupling its channels. The improvement brought about by vector tracking ranged from 6.2 to 2.4 dB in various scenarios. The greatest improvements came with the largest number of satellites and the smallest improvement from the fewest number of satellites.

Radio Frequency (RF) data collected from a hardware simulator was used to study the performance of the VDFLL in high dynamic scenarios. The VDFLL was able to maintain signal lock through 2, 4, and 8 G coordinated turns at 19 dB-Hz with only five satellites. The VDFLL managed to operate through the 2 and 4 G turns at 16 dB-Hz with seven satellites, but could not maintain lock through the 8 G turn with eleven satellites at 16 dB-Hz. The rule of thumb analysis tools were applied to the 2, 4, and 8 G turns. The predictions from the rule of thumb analysis tend to be slightly pessimistic, but overall matched the simulation results reasonably well.

The performance of different vector tracking architectures was also studied. Three distinct architectures were investigated. The first uses a single centralized filter to perform all the tasks of signal tracking and navigation state estimation. The other two algorithms use federated architectures. Both federated architectures use a master filter in conjunction with smaller residual filters to track the received signals. The difference in the two federated architectures is in the design of the residual filters. One uses residual filters that are coupled together and the other uses residual filters that are decoupled and operated independently. The performance of the three algorithms was studied using covariance analysis and Monte Carlo simulations. The covariance analysis shows that the centralized filter and the federated filter with coupled channels yield the same performance. The federated filter with decoupled residual filters exhibited a loss in performance relative to the other two architectures. However, as the C/N_0 ratio declined the difference in the performance of the algorithms diminished. The Monte Carlo simulations verified the results of the covariance analysis.

The analysis of the vector tracking architectures was then extended to an analysis of deeply integrated architectures. The three vector tracking algorithms were combined with an Inertial Measurement Unit (IMU) to produce three deeply integrated GPS/INS systems. Just like in the case of vector tracking, one algorithm uses a centralized architecture and the other two use a federated architecture. One of the federated algorithms uses coupled channel residual filters and the other uses decoupled channel residual filters. The effect of using a federated architecture with and without coupled channels on DI's performance was then studied. The performance of the centralized DI algorithm and the federated algorithm with coupled channels were essentially identical. The federated algorithm with decoupled channels performed poorest. However, the difference in the performance of the architectures decreased as the C/N_0 ratio decreased. At low C/N_0 ratios the performance of the algorithms were virtually identical.

The final chapter of the dissertation used live-sky data to study the performance of the VDFLL in challenging environments. The data was collected using a software GPS receiver and the VDFLL was implemented in MATLAB. Data was also collected using a NovAtel

ProPak-V3 receiver for comparison. Auburn University's Mini Baja course was used as one location to test the VDFLL. The Mini Baja course is a challenging environment due to the dense foliage along the course. Two sets of data were collected from the course. The VDFLL provided continuous coverage around the course in both data sets. In contrast, the NovAtel provided more or less continuous position solutions in the first data set, but with aberrant jumps of 15 to 20 meters. In the second data set, the NovAtel did not report the receiver's positions for the majority of the course. Downtown Atlanta was used as the second location for data collection. The area of downtown Atlanta that was traversed contained an urban canyon environment. The VDFLL again provided continuous coverage. However, during a stretch of the course through high-rise buildings the positions reported by the VDFLL wandered off the highway by up to 40 meters. The NovAtel reported no positions for the same stretch of highway. Overall the VDFLL exhibited more robust performance than the NovAtel receiver.

11.2 Future Work

There are a multitude of avenues for future work in the area of vector tracking. All the algorithms in this dissertation are noncoherent. The phase of the carrier signals is not tracked, only their frequencies are estimated. Extending vector tracking to include the carrier phases would be an interesting research topic. There are several issues that have to be addressed when integrating the carrier phases into the Kalman filter used to track the pseudoranges and pseudorange-rates. One issue is the fact that the pseudoranges and pseudorange-rates can be directly predicted from the receiver's estimated position and velocity, and the broadcast ephemeris data. However, the carrier phases cannot be predicted unambiguously. Other issues to be addressed are the phenomenon of code-carrier divergence and other atmospheric effects. The carrier phase of the received signals is also more susceptible to interference, such as from foliage. Since the channels of the receiver are coupled together, a disturbance in one carrier phase would effect the carrier phase estimates for the other satellites.

Another topic of future research is integrity monitoring in vector tracking architectures. Since the tracking of all the satellite signals is done in unison, a fault in one channel of the receiver will affect all the other channels. The ability to monitor the tracking of the satellite signals and remove possible faults is very important.

The multipath performance of vector tracking algorithms has yet to be explored. The tracking of all the satellite signals is accomplished jointly and in a single filter in vector tracking. Intuitively it would seem that vector tracking should exhibt superior performance with respect to multipath. The effect that multipath has on tracking the Gold codes and carrier frequencies in a vector tracking algorithm is not clear. The pseudorange-state filter operated in the coupled channel and decoupled channel modes of operation could be used to study the multipath performance of vector tracking. The two modes of operation allow the same algorithm to be operated with or without information sharing between channels. The benefits of information sharing can therefore be easily determined.

The analysis of deeply integrated algorithms can be extended to study the performance of suboptimal GPS/INS systems. The federated DI algorithms that were studied used elaborate master filters with large state vectors. The master filter required the same seventeen states used in the centralized filter plus multiple additional states. The additional states are necessary to properly model the correlation between errors in the master filter and the residual filters. These additional states add significantly to the complexity and computational load of the federated algorithms. In practice, the additional states might not be included in the master filter's state vector due to processing constraints. Omitting the states would lead to a degradation in the algorithm's performance. The amount of degradation caused by using the reduced-order filter can be be determined through covariance analysis. The full state vector formulation can be used in conjunction with a suboptimal Kalman gain to determine the loss in performance. The ability of rule of thumb type techniques to mitigate the correlation in the errors between the two filters could also be ascertained.

The integration of GPS with other satellite navigation systems in a vector tracking architecture is another interesting topic for future research. The increase in performance offered by vector tracking is directly linked to the number of available satellites. Other satellite based navigation systems (such as GLONASS and Galileo) offer additional satellite signals for a receiver to track. The principles of operation for the other navigation systems are different than those used in GPS. However, an overall increase in performance through the integration of all the systems together might be possible.

BIBLIOGRAPHY

[Abbott and Lillo, 2003] Abbott, A. S. and Lillo, W. E. (2003). Global positioning systems and inertial measuring unit ultratight coupling method. U. S. Patent 6,516,021, The Aerospace Corporation, Segundo, CA.

[Alban et al., 2003] Alban, S., Akos, D. M., Rock, S. M., and Gebre-Egziabher, D. (2003). Performance analysis and architectures for INS-aided GPS tracking loops. In *Proceedings of the Institute of Navigation National Technical Meeting*, pages 611–622, Anaheim, CA. Institute of Navigation.

[Alban and Dietrich, 2004] Alban, S. and Dietrich, A. (2004). Inertial aiding of phase-tracking loops for automotive GPS attitude determination. In *Proceedings of the 60th Annual Meeting*, Dayton, OH. ION.

[Bar-Shalom et al., 2001] Bar-Shalom, Y., Li, X.-R., and Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, New York.

[Benson, 2007] Benson, D. (2007). Interference benefits of a vector delay lock loop (VDLL) GPS receiver. In *Proceedings of the 63rd Annual Meeting of the Institute of Navigation*, Cambridge, Massachusetts. Institute of Navigation.

[Besser et al., 2002] Besser, J., Alexander, S., Crane, R., Rounds, S., Wyman, J., and Baeder, B. (2002). TRUNAV: A low-cost guidance/navigation unit integrating a SAASM-based GPS and MEMS IMU in a deeply coupled mechanization. In *Proceedings of the Institute of Navigation GPS Conference*, pages 545–555. Institute of Navigation, Institute of Navigation 3975 University Drive Suite 390 Fairfax, VA 22030.

[Bowles, 1980] Bowles, W. M. (1980). *Correlation Tracking.* PhD thesis, Massachusetts Institute of Technology.

[Brown and Hwang, 1996] Brown, R. G. and Hwang, P. Y. C. (1996). *Introduction to Random Signals and Applied Kalman Filtering.* Wiley.

[Buck et al., 2006] Buck, T. M., Wilmot, J., and Cook, M. J. (2006). A high G, MEMS based, deeply integrated, INS/GPS, guidance, navigation and control flight management unit. In *Proceedings of IEEE/ION Position Location and Navigation Symposium Conference*, San Diego, CA. IEEE/ION.

[Bullock et al., 2006] Bullock, J. B., Foss, M., Geier, G. J., and King, M. (2006). Integration of GPS with other sensors and network assistance. In Kaplan, E. D., editor, *Understanding GPS: Principles and Applications*, Mobile Communication Series, chapter 9, pages 459–558. Artech House Publishers, 2 edition.

[Carlson, 1988] Carlson, N. A. (1988). Federated filter for fault-tolerant integrated navigation systems. In *Proceedings of IEEE/ION Position Location and Navigation Symposium Conference*, pages 110–119.

[Chan et al., 1979] Chan, Y. T., Hu, A. G. C., and Plant, J. B. (1979). A Kalman filter based tracking scheme with input estimation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-15:237–244.

[Chatfield, 1997] Chatfield, A. B. (1997). *Fundamentals of High Accuracy Inertial Navigation*, volume 174. American Institute of Aeronautics and Astronautics.

[Copps et al., 1980] Copps, E. M., Geier, G. J., Fidler, W. C., and Grundy, P. A. (1980). Optimal processing of GPS signals. *Navigation*, 27(3):171 – 182.

[Cox, 1978] Cox, D. B. (1978). Integration of GPS with inertial navigation systems. *Navigation*, 25(2):236 – 245.

[Crane, 2007] Crane, R. N. (2007). A simplified method for deep coupling of GPS and inertial data. In *Proceedings of the National Technical Meeting of the Institute of Navigation*, San Diego, California. Institute of Navigation.

[Dierendonck, 1996] Dierendonck, A. J. V. (1996). GPS receivers. In Parkinson, B. W., editor, *Global Positioning System: Theory and Applications, Volume 1*, volume 163 of *Progress in Astronautics and Aeronautics*, chapter 8. American Institute of Aeronautics and Astronautics, Washington, DC.

[Dyrud et al., 2007] Dyrud, L., Woessner, B., Jovancevic, A., and Ganguly, S. (2007). Ultra tightly coupled GPS/INS receiver for TSPI applications. In *Proceedings of the 20th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Fort Worth, TX. ION.

[Flenniken, IV, 2005] Flenniken, IV, W. S. (2005). Modeling inertial measurement units and analyzing the effect of their errors in navigation applications. Master's thesis, Auburn University.

[Flueckiger et al., 2005] Flueckiger, K., Gustafson, D., and Dowdle, J. (2005). INS/GPS deep integration navigation hardware testbed. In *Institute of Navigation, Annual Technical Meeting*, pages 1171–1178.

[Gautier, 2003] Gautier, J. D. (2003). *GPS/INS Generalized Evaluation Tool (GIGET) For The Design and Testing of Integrated Navigation Systems.* PhD thesis, Stanford University.

[Gebre-Egziabher et al., 2005] Gebre-Egziabher, D., Razavi, A., Enge, P. K., Gautier, J., Pullen, S., Pervan, B. S., and Akos, D. M. (2005). Sensitivity and performance analysis of doppler-aided GPS carrier-tracking loops. *Navigation*, 52(2):49–60.

[Godha, 2004] Godha, S. (2004). Strategies for GPS/INS integration. Presentation.

[Groves and Long, 2003] Groves, P. and Long, D. (2003). Adaptive tightly-coupled, a low cost alternative anti-jam INS/GPS integration technique. In *Proceedings of the 2003 National Technical Meeting of the Institute of Navigation*, Anaheim, CA. Institute of Navigation.

[Groves, 2008] Groves, P. D. (2008). *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House.

[Groves et al., 2007] Groves, P. D., Mather, C. J., and Macaulay, A. A. (2007). Demonstration of non-coherent deep INS/GPS integration for optimised signal-to-noise performance. In *Proceedings of the 20th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Fort Worth, Texas. Institute of Navigation.

[Gunawardena et al., 2004] Gunawardena, S., Soloviev, A., and van Graas, F. (2004). Real time implementation of deeply integrated software GPS receiver and low cost IMU for processing low-CNR GPS signals. In *Proceedings of the 60th Annual Meeting*, pages 108 – 114, Dayton, OH. ION.

[Gustafson and Dowdle, 2003] Gustafson, D. and Dowdle, J. (2003). Deeply integrated code tracking: Comparative performance analysis. In *Proceedings of Institute of Navigation GPS/GNSS Conference*, pages 2553–2561, Portland, OR. Institute of Navigation.

[Gustafson et al., 2000a] Gustafson, D., Dowdle, J., and Flueckiger, K. (2000a). A deeply integrated GPS-based navigator with extended range code tracking. In *Proceedings of the IEEE Position, Location, and Navigation Symposium*, pages 118–124. IEEE.

[Gustafson et al., 2000b] Gustafson, D., Dowdle, J., and Flueckiger, K. (2000b). A high anti-jam GPS-based navigator. In *Proceedings of the Institute of Navigation National Technical Meeting*, pages 495–503, Anaheim, CA. Institute of Navigation.

[Gustafson et al., 2000c] Gustafson, D., Dowdle, J., and Flueckiger, K. (2000c). A high anti-jam GPS-based navigator. In *Institute of Navigation, Annual Technical Meeting.*

[Gustafson et al., 2001] Gustafson, D. E., Dowdle, J. R., and John M. Elwell, J. (2001). Deeply-integrated adaptive GPS-based navigatior with extended-range code tracking. U. S. Patent 6,331835, The Charles Stark Draper Laboratory, Inc., Cambridge, MA.

[Hamm, 2005] Hamm, C. R. (2005). Analysis of simulated performance of integrated vector tracking and navigation loops for GPS. Master's thesis, Auburn University.

[Horslund and Hooker, 1999] Horslund, J. M. and Hooker, J. R. (1999). Increase jamming immunity by optimizing processing gain for GPS/INS systems. U. S. Patent 5,983160, Raytheon Company, Lexington, MA.

[Jovancevic and Ganguly, 2005] Jovancevic, A. and Ganguly, S. (2005). Real-time implementation of a deeply integrated GNSS-INS architecture. In *Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Long Beach, CA. ION.

[Jwo, 2001] Jwo, D. J. (2001). Optimization and sensitivity analysis of GPS receiver tracking loops in dynamic environments. *IEE Proceedings of Radar, Sonar Navigation*, 148:241–250.

[Kaplan et al., 1996] Kaplan, E. D., Leva, J. L., and Pavloff, M. S. (1996). Fundamentals of satellite navigation. In Kaplan, E. D., editor, *Understanding GPS: Principles and Applications*, Mobile Communication Series, chapter 2, pages 15–58. Artech House Publishers.

[Kennedy and Rossi, 2008] Kennedy, S. and Rossi, J. (2008). Performance of a deeply coupled commercial grade GPS/INS system from KVH and NovAtel Inc. In *Proceedings of IEEE/ION Position Location and Navigation Symposium Conference*, pages 17 – 24, Monterey, CA.

[Kiesel et al., 2007] Kiesel, S., Held, M. M., and Trommer, G. F. (2007). Realization of a deeply coupled GPS/INS navigation system based on INS-VDLL integration. In *Proceedings of the 2007 National Technical Meeting of the Institute of Navigation*, San Diego, California. Institute of Navigation.

[Kim et al., 2003] Kim, H.-S., Bu, S.-C., Jee, G.-I., and Park, C.-G. (2003). An ultra-tightly coupled GPS/INS integration using federated Kalman filter. In *Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Portland, OR. ION.

[Kreye et al., 2004] Kreye, C., Eissfeller, B., and Ameres, G. (2004). Architectures of GNSS/INS integrations: Theoretical approach and practical tests. In *Symposium on Gyro Technology*, pages 14.0–14.16.

[Landis et al., 2006] Landis, D., Thorvaldsen, T., Fink, B., Sherman, P., and Holmes, S. (2006). A deep integration estimator for urban ground navigation. In *Proceedings of IEEE/ION Position Location and Navigation Symposium Conference*, pages 927 – 932, San Diego. IEEE/ION.

[Lashley and Bevly, 2007] Lashley, M. and Bevly, D. M. (2007). Analysis of discriminator based vector tracking algorithms. In *Proceedings of the 2007 National Technical Meeting of the Institute of Navigation*, San Diego, California. Institute of Navigation.

[Lashley and Bevly, 2008] Lashley, M. and Bevly, D. M. (2008). Comparison of traditional tracking loops and vector based tracking loops for weak GPS signals. In *Proceedings of the 2008 National Technical Meeting of the Institute of Navigation*, San Diego, California. Institute of Navigation.

[Lashley et al., 2008] Lashley, M., Bevly, D. M., and Hung, J. Y. (2008). Impact of carrier to noise power density, platform dynamics, and IMU quality on deeply integrated navigation. In *Proceedings of IEEE/ION Position Location and Navigation Symposium Conference*, Monterey, California. Institute of Navigation/IEEE.

[Lashley et al., 2009] Lashley, M., Bevly, D. M., and Hung, J. Y. (2009). Performance analysis of vector tracking algorithms for weak GPS signals in high dynamics. *To appear in IEEE Journal of Selected Topics in Signal Processing.*

[Levy, 1997] Levy, L. J. (1997). The Kalman filter: Navigations integration workhorse. *GPS World*, 8:22.

[Lewis, 2003] Lewis, D. E. (2003). Ultra-tightly coupled GPS/INS tracking performance. In *Annual Aviation Technology, Integration, and Operations (ATIO) Forum*, Denver. AIAA.

[Li and Wang, 2006] Li, D. and Wang, J. (2006). Kalman filter design strategies for code tracking loop in ultra-tight GPS/INS/PL integration. In *Proceedings of the 2006 National Technical Meeting of the Institute of Navigation*, Monterey, CA. ION.

[Lin and Tsui, 2002] Lin, D. M. and Tsui, J. B. Y. (2002). A weak signal tracking technique for a stand-alone software GPS receiver. In *Proceedings of the 15th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Portland, OR. ION.

[Maybeck, 1978] Maybeck, P. S. (1978). *Stochastic Models, Estimation and Control*, volume 141. Academic Press.

[Misra and Enge, 2001] Misra, P. and Enge, P. (2001). *Global Positioning System: Signals, Measurements, and Performance.* Ganga-Jamuna Press.

[Negast, 2009] Negast, B. (2009). Personal communication.

[Ohlmeyer, 2006] Ohlmeyer, E. J. (2006). Analysis of an ultra-tightly coupled GPS/INS system in jamming. In *Proceedings of IEEE/ION Position Location and Navigation Symposium Conference*, pages 44–53, San Diego. IEEE/ION.

[Pany and Eissfeller, 2006] Pany, T. and Eissfeller, B. (2006). Use of a vector delay lock loop receiver for GNSS signal power analysis in bad signal conditions. In *Proceedings of IEEE/ION Position Location and Navigation Symposium Conference*, San Diego, California. Institute of Navigation and IEEE.

[Pany et al., 2005] Pany, T., Kaniuth, R., and Eissfeller, B. (2005). Deep integration of navigation solution and signal processing. In *Proceedings of Institute of Navigation GPS/GNSS Conference*, Long Beach, CA. Institute of Navigation.

[Pethel, 2006] Pethel, S. J. (2006). Test and evaluation of high performance micro electromechanical system based inertial measurement units. In *Proceedings of IEEE/ION Position, Location, and Navigation Symposium*, pages 405 – 414, San Diego,. IEEE/ION.

[Petovello and Lachapelle, 2006] Petovello, M. and Lachapelle, G. (2006). Comparison of vector-based software receiver implementations with application to ultra-tight GPS/INS integration. In *Proceedings of Institute of Navigation GPS/GNSS Conference*, Fort Worth, TX. Institute of Navigation.

[Psiaki and Jung, 2002] Psiaki, M. L. and Jung, H. (2002). Extended Kalman filter methods for tracking weak GPS signals. In *Proceedings of the 15th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Portland, Oregon. Institute of Navigation.

[Quinn et al., 2005] Quinn, P., Lewis, D., Berarducci, M., and Miller, M. (2005). Precise estimation and compensation of antenna lever arm flexure using auxiliary inertial sensors to improve ultra-tightly coupled GPS-IMU and ESM performance. In *Proceedings of the 2005 National Technical Meeting of the Institute of Navigation*, San Diego, CA. ION.

[Quinn et al., 2006] Quinn, P., Lewis, D., Berarducci, M., Miller, M., Campbell, J., and Howe, P. (2006). Theory and test results of antenna advanced inertial reference for enhanced sensors (ANTARES) lever arm flexure estimation and compensation for ultra-tightly coupled GPS-INS and RF emitter geo-location systems using auxiliary antenna-mounted inertial. In *Proceedings of the 19th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Fort Worth, TX. ION.

[Sayre, 2003] Sayre, M. M. (2003). Development of a block processing carrier to noise ratio estimator for the global positioning system. Master's thesis, Ohio University.

[Sennott and Senffner, 1995] Sennott, J. and Senffner, D. (1995). Comparison of continuity and integrity characteristics for integrated and decoupled demodulation/navigation receivers. In *Proceedings of the 8th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Palm Springs, CA. ION.

[Sennott and Senffner, 1997] Sennott, J. and Senffner, D. (1997). A GPS carrier phase processor for real-time high dynamics tracking. In *Proceedings of the 53rd Annual Meeting of the Institute of Navigation*, Albuquerque, NM. ION.

[Sennott, 1984] Sennott, J. W. (1984). A flexible GPS software development system and timing analyzer for present and future microprocessors. *Navigation*, 31(2):84–95.

[Sennott and Senffner, 1992a] Sennott, J. W. and Senffner, D. (1992a). Navigation receiver with coupled signal-tracking channels. U. S. Patent 5,343,209, Bloomington, IL.

[Sennott and Senffner, 1992b] Sennott, J. W. and Senffner, D. (1992b). The use of satellite geometry for prevention of cycle slips in a GPS processor. *Navigation*, 39(2):217 – 236.

[Sklar, 2001] Sklar, B. (2001). *Digital Communications: Fundamentals and Applications*. Prentice Hall PTR, 2nd edition.

[Soloviev et al., 2007] Soloviev, A., Bruckner, D., van Graas, F., and Marti, L. (2007). Assessment of GPS signal quality in urban environments using deeply integrated GPS/IMU. In *Proceedings of the 2007 National Technical Meeting of the Institute of Navigation*, pages 815 – 828, San Diego, CA. ION.

[Soloviev et al., 2004a] Soloviev, A., Gunawardena, S., and van Graas, F. (2004a). Deeply integrated GPS/low-cost IMU for low CNR signal processing: Flight test results and real time implementation. In *Proceedings of the 17th International Technical Meeting of the Satellite Division of the Institute of Navigation*, pages 1598 – 1608, Long Beach, CA.

[Soloviev et al., 2008] Soloviev, A., Gunawardena, S., and van Graas, F. (2008). Deeply integrated GPS/low-cost IMU for low CNR signal processing: Concept description and in-flight demonstration. *Navigation*, 55(1):1–13.

[Soloviev and van Graas, 2008] Soloviev, A. and van Graas, F. (2008). Utilizing multipath reflections in deeply integrated GPS/INS architecture for navigation in urban environments. In *Proceedings of IEEE/ION Position, Location, and Navigation Symposium*, pages 383 – 393, Monterey, CA. IEEE/ION.

[Soloviev et al., 2004b] Soloviev, A., van Graas, F., and Gunawardena, S. (2004b). Implementation of deeply integrated GPS/low-cost IMU for reacquisition and tracking of low CNR GPS signals. In *Proceedings of the Institute of Navigation National Technical Meeting*, pages 923–935, San Diego, CA. Institute of Navigation.

[Spilker, Jr., 1996a] Spilker, Jr., J. J. (1996a). Fundamentals of signal tracking theory. In Parkinson, B. W., editor, *Global Positioning System: Theory and Applications, Volume 1*, volume 163 of *Progress in Astronautics and Aeronautics*, chapter 4. American Institute of Aeronautics and Astronautics, Washington, DC.

[Spilker, Jr., 1996b] Spilker, Jr., J. J. (1996b). GPS signal structure and theoretical performance. In Parkinson, B. W., editor, *Global Positioning System: Theory and Applications, Volume 1*, volume 163 of *Progress in Astronautics and Aeronautics*, chapter 3. American Institute of Aeronautics and Astronautics, Washington, DC.

[Tsui, 2000] Tsui, J. B. Y. (2000). *Fundamentals of Global Positioning System Receivers, A Software Approach*. John Wiley & Sons.

[US Dept. of Defense, 2000] US Dept. of Defense (2000). NAVSTAR GPS space segment/navigation user interfaces. Iterface Control Document No. ICD-GPS-200C, Department of Defense.

[Wall, 2007] Wall, J. H. (2007). A study of the effects of stochastic inertial sensor errors in dead-reckoning navigation. Master's thesis, Auburn University, Auburn, AL.

[Ward, 1996a] Ward, P. (1996a). Effects of RF interference on GPS satellite signal receiver tracking. In Kaplan, E. D., editor, *Understanding GPS: Principles and Applications*, Mobile Communication Series, chapter 6, pages 209 – 236. Artech House Publishers.

[Ward, 1996b] Ward, P. (1996b). Satellite signal acquisition and tracking. In Kaplan, E. D., editor, *Understanding GPS: Principles and Applications*, Mobile Communication Series, chapter 5, pages 119 – 208. Artech House Publishers.

[Ward, 1997] Ward, P. W. (1997). Using a GPS receiver monte carlo simulator to predict RF interference performance. In *Proceedings of the 10th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Kansas City, MO. ION.

[Ward, 1998] Ward, P. W. (1998). Performance comparisons between FLL, PLL and a novel FLL-assisted-PLL carrier tracking loop under RF interference conditions. In *Proceedings of the 11th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Nashville, TN. ION.

[Weiss and Kee, 1995] Weiss, J. D. and Kee, D. S. (1995). A direct performance comparison between loosely coupled and tightly coupled GPS/INS integration techniques. In *Proceedings of the 51st Annual Meeting of the Institute of Navigation*, Colorado Springs, CO. ION.

[Wendel et al., 2006] Wendel, J., Metzger, J., Moenikes, R., Maier, A., and Trommer, G. F. (2006). A performance comparison of tightly coupled GPS/INS navigation systems based on extended and sigma point Kalman filters. *Navigation*, 53:21–31.

[Wenger, 2002] Wenger, L. (2002). An INS/GPS integrated navigation system for an autonomous micro air vehicle. Master's thesis, Swiss Federal Institute of Technology Zurich.

[Woessner et al., 2006] Woessner, W., Noronha, J., Jovancevic, A., and Ganguly, S. (2006). A software defined real-time ultra-tightly coupled (UTC) GNSS-INS architecture. In *Proceedings of the 19th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Fort Worth, TX. ION.

[Ziedan, 2006a] Ziedan, N. (2006a). Extended Kalman filter-based deeply integrated GNSS/low-cost INS for reliable navigation under GNSS weak interrupted signal conditions. In *Proceedings of the 19th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Fort Worth, TX. ION.

[Ziedan, 2005a] Ziedan, N. I. (2005a). Acquisition and fine acquisition of weak GPS L2C and L5 signals under high dynamic conditions for limited-resource applications. In *Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Long Beach, CA. Institute of Navigation.

[Ziedan, 2005b] Ziedan, N. I. (2005b). Extended Kalman filter tracking and navigation message decoding of weak GPS L2C and L5 signals. In *Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Long Beach, CA. Institute of Navigation.

[Ziedan, 2006b] Ziedan, N. I. (2006b). *GNSS Receivers for Weak signals.* Artech House, Norwood, MA.

APPENDICES

APPENDIX A

EQUIVALENCE OF POSITION-STATE AND PSEUDORANGE-STATE VECTOR TRACKING FORMULATIONS

A.1 Introduction

This appendix shows that the performance of the second order position-state VDFLL is identical to the performance of the second order pseudorange-state VDFLL. The metric used to quantify the performance of the algorithms is the variance of the pseudoranges and pseudorange-rates predicted by the filter. The steady state, state covariance matrices of the two filters are compared. It is shown that the steady state pseudorange and pseudorange-rate variances from the filters are identical. This result for the second order VDFLL is easily extended to higher order vector tracking loops.

A.2 Covariance Analysis

The error between the predictions and actual quantities is represented by the symbol ϵ and is defined in (A.1).

$$\epsilon = \begin{bmatrix} \rho_1 \\ \dot{\rho}_1 \\ \vdots \\ \rho_N \\ \dot{\rho}_N \end{bmatrix} - \begin{bmatrix} \hat{\rho}_1 \\ \hat{\dot{\rho}}_1 \\ \vdots \\ \hat{\rho}_N \\ \hat{\dot{\rho}}_N \end{bmatrix} \tag{A.1}$$

In (A.1), the pseudorange and pseudorange-rate predictions are are represented as $\hat{\rho}$ and $\hat{\dot{\rho}}$, respectively. The pseudorange and pseudorange-rate errors are the actual states of the pseudorange-state VDFLL filter. For the position-state VDFLL, the prediction errors are related to the states of the filter by (A.2).

$$\begin{aligned}
\epsilon &= C\hat{x} \qquad (A.2)\\
C &= \begin{bmatrix} a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 & 0 \\ 0 & a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 & 0 \\ 0 & a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 \end{bmatrix}
\end{aligned}$$

The mean of the vector ϵ is zero for both VDFLL filters. This is a result of the Kalman filter being an unbiased estimator. The covariances of the vector ϵ for the two filters are compared. The relation between the covariance of the vector ϵ and the state covariance matrix of the pseudorange-state VDFLL filter is shown in (A.3).

$$E\{\epsilon\epsilon^T\} = \tilde{P}^- \qquad (A.3)$$

In (A.3), the state covariance matrix of the pseudorange-state VDFLL filter prior to a measurement update is denoted as $\tilde{P}^-$. The covariance matrix of the position state filter is related to the covariance of the vector ϵ by (A.4).

$$E\{\epsilon\epsilon^T\} = CP^-C^T \qquad (A.4)$$

The steady state covariance of the vector ϵ is shown to be identical for both VDFLL formulations.

A.2.1 Position-State Covariance Analysis

The Kalman filter for the position-state formulation of the VDFLL is first examined. The continuous time state transition equations for the position-state Kalman filter are shown in (A.5).

$$\dot{x} = Ax + B^p_{dyn} w_{dyn} + B^p_{clk} w_{clk} \tag{A.5}$$

where

$$A = \begin{bmatrix} \kappa & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & \kappa & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & \kappa & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & 0_{2\times2} & \kappa \end{bmatrix}$$

$$\kappa = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$
\begin{aligned}
B^p_{dyn} &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
w_{dyn} &= \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \\
B^p_{clk} &= \begin{bmatrix} 0_{2\times2} \\ 0_{2\times2} \\ 0_{2\times2} \\ I_{2\times2} \end{bmatrix} \\
w_{clk} &= \begin{bmatrix} w_b \\ w_d \end{bmatrix}
\end{aligned}
$$

The noise sources (w_x, w_y, w_z) drive their respective velocity states. The terms (w_b, w_d) in w_{clk} represent the clock phase and frequency error of the user's local oscillator, expressed in units of meters and meters per second [Brown and Hwang, 1996], respectively. The statistics for the noise vector w_{dyn} are:

$$\begin{aligned} E\{w_{dyn}\} &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{(A.6)} \\ E\{w_{dyn}w_{dyn}^T\} &= \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \end{aligned}$$

The statistics for the noise vector w_{dyn} are:

$$\begin{aligned} E\{w_{clk}\} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \text{(A.7)} \\ E\{w_{clk}w_{clk}^T\} &= \begin{bmatrix} \sigma_b^2 & 0 \\ 0 & \sigma_d^2 \end{bmatrix} \end{aligned}$$

It is assumed that measurements are provided at intervals of T seconds. The discrete time state transition matrix used by the position-state Kalman filter is shown in (A.8).

$$\begin{aligned} A^p &= \begin{bmatrix} \alpha & 0_{2\times2} & \dots & 0_{2\times2} \\ 0_{2\times2} & \alpha & & \vdots \\ \vdots & & \ddots & \vdots \\ 0_{2\times2} & \dots & \dots & \alpha \end{bmatrix}_{8\times8} \qquad \text{(A.8)} \\ \alpha &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \end{aligned}$$

The discrete time process noise covariance matrix for the receiver dynamics process noise is shown in (A.9).

$$\begin{aligned}
Q^p_{dyn} &= \begin{bmatrix} Q_x & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & Q_y & 0_{2\times2} & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & Q_z & 0_{2\times2} \\ 0_{2\times2} & 0_{2\times2} & 0_{2\times2} & 0_{2\times2} \end{bmatrix} \qquad \text{(A.9)} \\
Q_x &= \begin{bmatrix} \sigma_x^2\frac{T^3}{3} & \sigma_x^2\frac{T^2}{2} \\ \sigma_x^2\frac{T^2}{2} & \sigma_x^2 T \end{bmatrix} \\
Q_y &= \begin{bmatrix} \sigma_y^2\frac{T^3}{3} & \sigma_y^2\frac{T^2}{2} \\ \sigma_y^2\frac{T^2}{2} & \sigma_y^2 T \end{bmatrix} \\
Q_z &= \begin{bmatrix} \sigma_z^2\frac{T^3}{3} & \sigma_z^2\frac{T^2}{2} \\ \sigma_z^2\frac{T^2}{2} & \sigma_z^2 T \end{bmatrix}
\end{aligned}$$

The discrete time process noise covariance matrix for the clock process noise is shown in (A.10).

$$\begin{aligned}
Q^p_{clk} &= \begin{bmatrix} 0_{2\times2} & \dots & 0_{2\times2} \\ \vdots & \ddots & \vdots \\ 0_{2\times2} & \dots & Q_c \end{bmatrix} \qquad \text{(A.10)} \\
Q_c &= \begin{bmatrix} \sigma_b^2 T + \sigma_d^2\frac{T^3}{3} & \sigma_d^2\frac{T^2}{2} \\ \sigma_d^2\frac{T^2}{2} & \sigma_d^2 T \end{bmatrix}
\end{aligned}$$

The relationship between the states of the position-state filter and the pseudorange and pseudorange-rate residuals is shown in (A.11).

$$\begin{aligned} y &= Cx + v \\ C &= \begin{bmatrix} a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 & 0 \\ 0 & a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 & 0 \\ 0 & a_{x,N} & 0 & a_{y,N} & 0 & a_{z,N} & 0 & 1 \end{bmatrix} \end{aligned} \tag{A.11}$$

The statistics for the measurement noise are shown in (A.12).

$$\begin{aligned} E\{v\} &= 0_{N\times 1} \\ E\{vv^T\} &= R_v \end{aligned} \tag{A.12}$$

Equations (A.8) through (A.12) provide the necessary information to perform the state covariance matrix updates in the Kalman filter recursions for the position-state filter.

A.2.2 Pseudorange-State Covariance Analysis

The pseudorange-state formulation of the VDFLL is know examined. The continuous time dynamics for the pseudorange-state Kalman filter are shown below in (A.13).

$$\dot{x} = Ax + B^{\rho}_{dyn} w_{dyn} + B^{\rho}_{clk} w_{clk} \tag{A.13}$$

where

$$
\begin{aligned}
A &= \begin{bmatrix} \kappa & 0_{2\times 2} & \dots & 0_{2\times 2} \\ 0_{2\times 2} & \kappa & \dots & \vdots \\ \vdots & \vdots & \ddots & 0_{2\times 2} \\ 0_{2\times 2} & \dots & 0_{2x2} & \kappa \end{bmatrix} \\
\kappa &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\
B^{\rho}_{dyn} &= \begin{bmatrix} 0 & 0 & 0 \\ a_{x,1} & a_{y,1} & a_{z,1} \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ a_{x,N} & a_{y,N} & a_{z,N} \end{bmatrix} \\
w_{dyn} &= \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \\
B^{\rho}_{clk} &= \begin{bmatrix} I_{2\times 2} \\ \vdots \\ I_{2\times 2} \end{bmatrix} \\
w_{clk} &= \begin{bmatrix} w_b \\ w_d \end{bmatrix}
\end{aligned}
$$

The noise terms in (A.13) are identical to those in (A.5).

The discrete time state transition matrix for the pseudorange-state Kalman filter is shown in (A.14).

$$\begin{aligned}
A^{\rho} &= \begin{bmatrix} \alpha & 0_{2\times 2} & \dots & 0_{2\times 2} \\ 0_{2\times 2} & \alpha & & \vdots \\ \vdots & & \ddots & \vdots \\ 0_{2\times 2} & \dots & \dots & \alpha \end{bmatrix}_{2N\times 2N} \qquad \text{(A.14)} \\
\alpha &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}
\end{aligned}$$

The discrete time process noise covariance matrix for the receiver dynamics process noise is shown in (A.15).

$$\begin{aligned}
Q^{\rho}_{dyn} &= \begin{bmatrix} Q^d_{1,1} & Q^d_{1,2} & \dots & Q^d_{1,N} \\ Q^d_{2,1} & Q^d_{2,2} & \dots & Q^d_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ Q^d_{N,1} & \dots & \dots & Q^d_{N,N} \end{bmatrix} \qquad \text{(A.15)} \\
Q^d_{i,j} &= \begin{bmatrix} \beta_{i,j}\frac{T^3}{3} & \beta_{i,j}\frac{T^2}{2} \\ \beta_{i,j}\frac{T^2}{2} & T\beta_{i,j} \end{bmatrix} \\
\beta_{i,j} &= \begin{bmatrix} \sigma_x^2 a_{x,i}a_{x,j} + \sigma_y^2 a_{y,i}a_{y,j} + \sigma_z^2 a_{z,i}a_{z,j} \end{bmatrix}
\end{aligned}$$

The discrete time process noise covariance matrix for the clock process noise is shown in (A.16).

$$\begin{aligned} Q^{\rho}_{clk} &= \begin{bmatrix} Q^c & \dots & Q^c \\ \vdots & \ddots & \vdots \\ Q^c & \dots & Q^c \end{bmatrix} \\ Q^c &= \begin{bmatrix} \sigma_b^2 T + \sigma_d^2 \frac{T^3}{3} & \sigma_d^2 \frac{T^2}{2} \\ \sigma_d^2 \frac{T^2}{2} & \sigma_d^2 T \end{bmatrix} \end{aligned} \tag{A.16}$$

The relationship between the states of the filter and the pseudorange and pseudorange-rate residuals is shown in (A.17).

$$y = I_{2N\times 2N} x + v \tag{A.17}$$

The measurement noise in (A.17) is identical to the noise in (A.11). The necessary information for performing the Kalman filter recursions for the pseudorange-state VDFLL filter are provided in equations (A.14) through (A.17).

A.2.3 Equivalence in Steady State Prediction Errors

To compare the steady state performance of the two VDFLL algorithms, it is first assumed that the position-state filter has reached steady state conditions. The steady state covariance matrices are denoted by the subscript *ss*. The steady state covariance matrix of the position-state filter is used to initialize the covariance matrix of the pseudorange-state

Kalman filter.

$$\tilde{P}^{+}(k) = C P_{ss}^{+} C^{T} \tag{A.18}$$

The symbol k in (A.18) is the index for the measurement epoch of the filter. The steady state value that the covariance matrix converges too will not depend on its initialization. The covariance matrices of the two formulations after propagating them forward to the next measurement epoch are now compared.

First, the position-state state covariance matrix is propagated forward to the next measurement updated using (A.19).

$$P_{ss}^{-} = A^{p} P_{ss}^{+} A^{pT} + Q_{dyn}^{p} + Q_{clk}^{p} \tag{A.19}$$

Likewise, the pseudorange-state state covariance matrix is propagated forward to the next measurement updated using (A.20).

$$\tilde{P}^{-}(k) = A^{\rho} \tilde{P}^{+}(k) A^{\rho T} + Q_{dyn}^{\rho} + Q_{clk}^{\rho} \tag{A.20}$$

The covariance of the predicted pseudorange and pseudorange-rate errors is now compared for the two VDFLL formulations. The covariance of the prediction errors is related to the state covariance matrix of the pseudorange-state filter by (A.21).

$$E\{\epsilon \epsilon^{T}\} = \tilde{P}^{-}(k) \tag{A.21}$$

The position-state state covariance matrix is related to the prediction errors by (A.22).

$$\begin{aligned}
E\{\epsilon\epsilon^T\} &= CP_{ss}^{-}C^T \\
CP_{ss}^{-}C^T &= CA^pP_{ss}^{+}A^{pT}C^T + CQ_{dyn}^{p}C^T \ldots \\
&\quad + CQ_{clk}^{p}C^T
\end{aligned} \tag{A.22}$$

The following relationships are now used:

$$\begin{aligned}
CQ_{dyn}^{p}C^T &= Q_{dyn}^{\rho} \\
CQ_{clk}^{p}C^T &= Q_{clk}^{\rho} \\
CA^p &= A^{\rho}C
\end{aligned} \tag{A.23}$$

Using the relations in (A.23), the following equivalences are made:

$$\begin{aligned}
CP_{ss}^{-}C^T &= CA^pP_{ss}^{+}A^{pT}C^T + CQ_{dyn}^{p}C^T + CQ_{clk}^{p}C^T \\
CP_{ss}^{-}C^T &= A^{\rho}CP_{ss}^{+}C^TA^{\rho T} + Q_{dyn}^{\rho} + Q_{clk}^{\rho} \\
CP_{ss}^{-}C^T &= A^{\rho}\tilde{P}^{-}(k)A^{\rho T} + Q_{dyn}^{\rho} + Q_{clk}^{\rho} \\
CP_{ss}^{-}C^T &= \tilde{P}^{-}(k)
\end{aligned} \tag{A.24}$$

Therefore, following the time propagation, the covariance of the pseudorange and pseudorange-rate prediction errors from the position-state and pseudorange-state filters are identical. The covariance matrices for the two filters are now examined after the measurement update.

The state covariance matrix for the pseudorange-state formulation after the measurement update is given below in (A.25).

$$\tilde{P}^{+}(k+1) = \tilde{P}^{-}(k) - \tilde{P}^{-}(k)\left[\tilde{P}^{-}(k) + R_v\right]^{-1}\tilde{P}^{-}(k) \tag{A.25}$$

The state covariance matrix for the position-state formulation after the measurement update is shown in (A.26).

$$P_{ss}^{+} = P_{ss}^{-} - P_{ss}^{-}C^{T}\left[CP_{ss}^{-}C^{T} + R_v\right]^{-1}CP_{ss}^{-} \tag{A.26}$$

Multiplying (A.26) by C and C^T, and using (A.24) and (A.25), results in (A.27).

$$\begin{aligned}
CP_{ss}^{+}C^{T} &= CP_{ss}^{-}C^{T}... \\
&\quad -CP_{ss}^{-}C^{T}\left[CP_{ss}^{-}C^{T} + R_v\right]^{-1}CP_{ss}^{-}C^{T} \\
\tilde{P}^{-}(k) &= CP_{ss}^{-}C^{T} \\
CP_{ss}^{+}C^{T} &= \tilde{P}^{-}(k) - \tilde{P}^{-}(k)\left[\tilde{P}^{-}(k) + R_v\right]^{-1}\tilde{P}^{-}(k) \\
CP_{ss}^{+}C^{T} &= \tilde{P}^{+}(k+1) \qquad \text{(A.27)}
\end{aligned}$$

For the pseudorange-state filter, the state covariance matrix after the measurement update at time k is equal to the state covariance matrix after the measurement update at time $k+1$. The initialization matrix in (A.18) is therefore the steady state covariance matrix of the pseudorange-state filter. This result is shown in (A.28).

$$\tilde{P}_{ss}^{+} = CP_{ss}^{+}C^{T} \tag{A.28}$$

Using (A.21), the steady state covariance matrix of the prediction errors for both formulations is shown in (A.29).

$$E\{\epsilon\epsilon^T\}_{ss} \quad = \quad \tilde{P}_{ss}^{-} = CP_{ss}^{+}C^T \tag{A.29}$$

The significance of (A.29) is that the steady state covariance of the prediction errors for both formulations of the VDFLL are identical. Therefore, the two algorithms yield the same performance with respect to the pseudorange and pseudorange-rate prediction errors. The analysis shown here has been performed for the second order VDLL. However, the same analysis can be performed for higher order VDFLL filters. The same results shown in (A.29) are obtained.

Appendix B

C/N_0 Ratio Estimation

B.1 Introduction

This appendix explains how the C/N_o ratio of the received signals is estimated. It is based on the method originally presented in [Crane, 2007]. Measurements of the noise power are generated by using noise correlators. A moving average filter is then used to smooth the noise power estimates. A single state Kalman filter is used to estimate the signal amplitude for each satellite signal. Large changes in signal amplitude are detected by monitoring the ratio of the signal power to noise power spectral density. The tunings used by the algorithms are included.

B.2 Noise Power Estimation

The variance of the noise is estimated by employing noise correlators. Over each integrate and dump operation, a series of correlator outputs are created which are solely noise (these are referred to as noise correlators). This is accomplished by taking the predicted code phase for a given channel, and adding a large offset to it. The large offset is chosen so that the autocorrelation function of the Gold code is effectively zero. For the algorithms implemented in this dissertation, 18 samples of pure noise are created every 20 milliseconds (i.e. at the end of every integrate and dump operation). These samples are used to compute an empirical measurement of the noise variance using the following:

$$\tilde{\sigma}_{\eta}^{2}(k) \quad = \quad \frac{1}{N_{\eta}} \sum_{m=1}^{N_{\eta}} (\eta_{k}(m))^{2} \tag{B.1}$$

In (B.1), the variable k is the index of the integrate and dump operation over which the noise correlators are generated and N_{η} is the number of noise samples generated. The noise samples created from the k-th integrate and dump operation are denoted as $\eta_{k}(m)$, where m ranges from one to N_{η}. The noise samples $\eta_{k}(m)$ are assumed to be zero mean, jointly Gaussian, and uncorrelated. Under these circumstances, the estimate given by (B.1) is the maximum likelihood estimate of the noise variance. The estimate is unbiased and has a variance given by:

$$\sigma_{ML}^{2} \quad = \quad \frac{2\sigma_{\eta}^{4}(k)}{N_{\eta}} \tag{B.2}$$

In (B.2), $\sigma_{\eta}(k)$ is the true standard deviation of the noise correlators from the k-th integrate and dump operation. The variance σ_{ML}^{2} is equal to the Cramer-Rao bound. Using 18 samples produces a large value for $\hat{\sigma}_{ML}^{2}$. However, the noise in the correlator outputs is assumed to be thermal noise and its variance changes slowly over time. The empirical measurements of the noise power are smoothed using a running average filter to produce accurate estimates of σ_{η}^{2} (B.3) [Crane, 2007].

$$\hat{\sigma}_{\eta}^{2}(k) \quad = \quad (1-\alpha_{\eta})\hat{\sigma}_{\eta}^{2}(k-1) + \alpha_{\eta}\tilde{\sigma}_{\eta}^{2}(k) \tag{B.3}$$

In (B.3), the estimated noise variance at time k is denoted as $\hat{\sigma}_\eta^2(k)$. For the algorithms implemented in this dissertation, a value of .001 is used for α_η. The estimated noise power is unbiased and the variance of the estimate in steady state is given in (B.4).

$$E\{(\hat{\sigma}_\eta^2 - \sigma_\eta^2)\} = \frac{\alpha_\eta^2}{1-(1-\alpha_\eta)^2}\frac{2\sigma_\eta^4}{N_\eta} \tag{B.4}$$

B.3 Signal Amplitude Measurements

Measurements of the signal amplitude are generated using the inphase early (IE), inphase late (IL), quadraphase early (QE), and quadraphase late (QL) correlator outputs (B.5).

$$\begin{aligned}
IE(k) &= A(k)R(\epsilon(k)+\tau)D(k)\cos\left(\pi f_{err}(k)T + \theta_{err}(k)\right) + \eta_{IE}(k)\\
IL(k) &= A(k)R(\epsilon(k)-\tau)D(k)\cos\left(\pi f_{err}(k)T + \theta_{err}(k)\right) + \eta_{IL}(k)\\
QE(k) &= A(k)R(\epsilon(k)+\tau)D(k)\sin\left(\pi f_{err}(k)T + \theta_{err}(k)\right) + \eta_{QE}(k)\\
QL(k) &= A(k)R(\epsilon(k)-\tau)D(k)\sin\left(\pi f_{err}(k)T + \theta_{err}(k)\right) + \eta_{QL}(k)\\
\eta(k) &\sim N(0, \sigma_\eta^2(k))
\end{aligned} \tag{B.5}$$

In (B.5), the variable $A(k)$ represents the effective amplitude of the signal over the k-th integrate and dump operation. It includes the correlation loss from Doppler frequency errors. The variables $f_{err}(k)$ and $\theta_{err}(k)$ are the frequency error (Hertz) and phase error (radians) between the received and locally generated carrier signals over the k-th integrate and dump operation, respectively. The sign of the navigation data bit over the integrate and dump operation is denoted as $D(k)$. The variable T is the period of the integrate and dump operation. The term $\epsilon(k)$ is the phase error between the received and locally generated PRN codes, measured in fractions of a chip. The function $R(\epsilon)$ is the auto-correlation function of the PRN sequences. The symbol τ is the correlator spacing of the Early and Late PRN replicas, expressed in fractions of a chip. A correlator spacing of one half chip is used for all algorithms implemented in this dissertation. The autocorrelation as a function of the code phase error is approximately given by:

$$R(\epsilon) = \begin{cases} 1 - |\epsilon| & \text{when } \epsilon < 1 \text{ chip} \\ 0 & \text{when } \epsilon > 1 \text{ chip.} \end{cases}$$

The correlator outputs are not normalized in (B.5) and the variance of the noise terms is represented by $\sigma_\eta^2(k)$. The subscripts IE, IL, QE, and QL represent the inphase early, inphase late, quadraphase early, and quadraphase late noise terms, respectively. The inphase and quadraphase noise sources are uncorrelated and the early and late noise terms are also uncorrelated. Measurements of the signal amplitude are generated by using the discriminator in (B.6).

$$\tilde{A}(k) = (IE(k) + IL(k))^2 + (QE(k) + QL(k))^2 \tag{B.6}$$

In (B.6), the variable k is the index of the integrate and dump operation used to generate the correlator outputs. The mean of the measurement $\tilde{A}(k)$ is shown in (B.7).

$$E\{\tilde{A}(k)\} = A^2(k) + 4\sigma_\eta^2(k) \tag{B.7}$$

The variable $\sigma_\eta^2(k)$ is the variance of the noise terms from (B.5). The variance of amplitude measurement is given in (B.8).

$$E\{(\tilde{A}(k) - E\{\tilde{A}(k)\})^2\} = 8\sigma_\eta^2(k)(2\sigma_\eta^2(k) + A^2(k)) \tag{B.8}$$

B.4 Amplitude Kalman Filter

The amplitude measurements generated from (B.6) are smoothed using a one state Kalman filter. The state, discrete time dynamics, and measurement update equation of the filter are shown in (B.9).

$$\begin{aligned} x(k) &= A^2(k) + 4\sigma_\eta^2(k) \\ x(k+1) &= x(k) + q(k) \\ y(k) &= x(k) + \zeta(k) \end{aligned} \tag{B.9}$$

The states of the filter are represented by $x(k)$ and the process noise is represent by $q(k)$. Measurements of the filter's state are denoted as $y(k)$ and the measurement noise is represented by $\zeta(k)$. The filter intentionally tracks a biased estimate of the squared amplitude. If the term $4\sigma_\eta^2$ is subtracted from the amplitude measurement in (B.6), an unbiased measurement of the amplitude squared can be produced. However, at low C/N_o ratios the subtraction will produce negative measurements of the amplitude squared. Including the negative measurements is quixotic, since the amplitude is a real non-negative number. Ignoring the negative measurements will result in biased estimates of the amplitude squared. To avoid these issues, the biased amplitude squared is processed.

The C/N_o ratio of the received signal is calculated by using the state of the Kalman filter and the empirically determined noise power (B.10).

$$\hat{C}/\hat{N}_0 = 10\log_{10}\left(\frac{\hat{x}(k) - 4\hat{\sigma}_\eta^2(k)}{2T\hat{\sigma}_\eta^2(k)}\right) \text{ in dB-Hz} \tag{B.10}$$

$$\hat{c}/\hat{n}_o = \left(\frac{\hat{x}(k) - 4\hat{\sigma}_\eta^2(k)}{2T\hat{\sigma}_\eta^2(k)}\right) \text{ in dB-Hz} \tag{B.11}$$

The formula in (B.10) is used only when the numerator is greater than zero.

The statistics of the noise term $\zeta(k)$ are given in (B.8). The current state of the filter and empirical noise power estimate are used at each measurement update in (B.8). The process noise in (B.9) is scaled based upon the current C/N_o ratio estimate. The process noise is determined from (B.12).

$$Q = \frac{1}{1000} \cdot 10^{[-3(28-\hat{C}/\hat{N}_0)\cdot u(28-\hat{C}/\hat{N}_0)]/10} \cdot \zeta(k) \tag{B.12}$$

In (B.12), $u(x)$ is the unit step function.

B.5 Adaptive Measures

For the algorithms used in this dissertation, the noise present in the correlator outputs is assumed to be thermal noise. The variance of the noise therefore changes relatively slowly over time. Rapid changes in the C/N_o ratio are therefore caused by fluctuations in the signal amplitude. Adaptive techniques are used by the amplitude Kalman filter to track rapid increases or decreases of the signal amplitude. The ratio of the signal power to noise power density is monitored to detect fluctuations in the amplitude of the received signal. The power ratio is given in (B.13).

$$\hat{c}/\hat{n}_o(k) = \left(\frac{\tilde{A}_s(k) - 4\hat{\sigma}^2_{\eta,s}(k)}{2T\hat{\sigma}^2_{\eta,s}(k)} \right) \tag{B.13}$$

The values for $\tilde{A}_s(k)$ are computed by averaging over the past N_s measurements from (B.6).

$$\tilde{A}_s(k) = \frac{1}{N_s} \sum_{m=0}^{N_s-1} \tilde{A}(k-m) \tag{B.14}$$

The term $\hat{\sigma}^2_{\eta,s}(k)$ is similarly calculated by averaging over the past N_s estimates of the noise power yielded from the formula in (B.3).

$$\hat{\sigma}^2_{\eta,s}(k) = \frac{1}{N_s} \sum_{m=0}^{N_s-1} \hat{\sigma}^2_{\eta}(k-m) \tag{B.15}$$

The expected value of (B.13) is approximately equal to (B.16).

$$E\{\hat{c}/\hat{n}_o\} \approx \frac{A^2(k)}{2T\sigma_\eta^2(k)} \tag{B.16}$$

The variance of the signal power to noise power density ratio is approximately equal to (B.17).

$$E(E\{\hat{c}/\hat{n}_o(k)\} - \hat{c}/\hat{n}_o(k)) = \sigma^2_{\hat{c}/\hat{n}_o}$$

$$\sigma^2_{\hat{c}/\hat{n}_o}(k) \approx \frac{8\sigma_\eta^2(k)(2\sigma_\eta^2(k) + A^2(k))}{N_s(2T\sigma_\eta^2(k))^2} + \frac{(A^2(k) + 4\sigma_\eta^2(k))^2}{(2T)^2 \cdot \sigma_\eta^8(k)} \frac{\alpha_\eta^2}{1-(1-\alpha_\eta)^2} \frac{2\sigma_\eta^4(k)}{N_\eta \cdot N_s} \tag{B.17}$$

The mean and variance of the power ratio is predicted using the current state of the amplitude Kalman filter and current estimate of the noise power from (B.4). The deviation of the calculated power ratio from the predicted mean is monitored. If the difference in the two quantities exceeds an upper threshold or drops below a lower threshold, then a significant change in the signal amplitude is declared. If a significant change in amplitude is declared at time k, the covariance of the amplitude filter is set to (B.18) and then the current amplitude measurement is processed by the filter.

$$P^-(k) = .35 \cdot 8\hat{\sigma}_\eta^2(k)(\hat{x}^-(k) - 2\hat{\sigma}_\eta^2(k)) \tag{B.18}$$

Table B.1: Thresholds Used to Detect Large Amplitude Fluctuations

C/N_0 (dB-Hz)	Lower Threshold	Upper Threshold
100	$4.0 \cdot \sigma_{\hat{c}/\hat{n}_o}$	$4.29 \cdot \sigma_{\hat{c}/\hat{n}_o}$
45	$4.0 \cdot \sigma_{\hat{c}/\hat{n}_o}$	$4.29 \cdot \sigma_{\hat{c}/\hat{n}_o}$
40	$3.9 \cdot \sigma_{\hat{c}/\hat{n}_o}$	$4.29 \cdot \sigma_{\hat{c}/\hat{n}_o}$
35	$3.5 \cdot \sigma_{\hat{c}/\hat{n}_o}$	$4.51 \cdot \sigma_{\hat{c}/\hat{n}_o}$
30	$3.5 \cdot \sigma_{\hat{c}/\hat{n}_o}$	$4.84 \cdot \sigma_{\hat{c}/\hat{n}_o}$
25	$3.4 \cdot \sigma_{\hat{c}/\hat{n}_o}$	$5.06 \cdot \sigma_{\hat{c}/\hat{n}_o}$
20	$3.1 \cdot \sigma_{\hat{c}/\hat{n}_o}$	$5.17 \cdot \sigma_{\hat{c}/\hat{n}_o}$
15	$3.0 \cdot \sigma_{\hat{c}/\hat{n}_o}$	$5.28 \cdot \sigma_{\hat{c}/\hat{n}_o}$
10	$3.0 \cdot \sigma_{\hat{c}/\hat{n}_o}$	$5.45 \cdot \sigma_{\hat{c}/\hat{n}_o}$
0	$3.0 \cdot \sigma_{\hat{c}/\hat{n}_o}$	$5.50 \cdot \sigma_{\hat{c}/\hat{n}_o}$

A period of 1.2 seconds is always allowed to elapse between resets of the covariance matrix. The thresholds used to detect fluctuations vary based on the estimated C/N_0 ratios. This due to the fact that the distribution of the linear power ratio becomes skewed as the C/N_0 ratio declines. The thresholds used in this dissertation are shown in Table B.1. A value of 15 is used for N_s. Figure B.1 shows the algorithms describe in this appendix being used to estimated the C/N_0 ratio of a signal with a fluctuating amplitude.

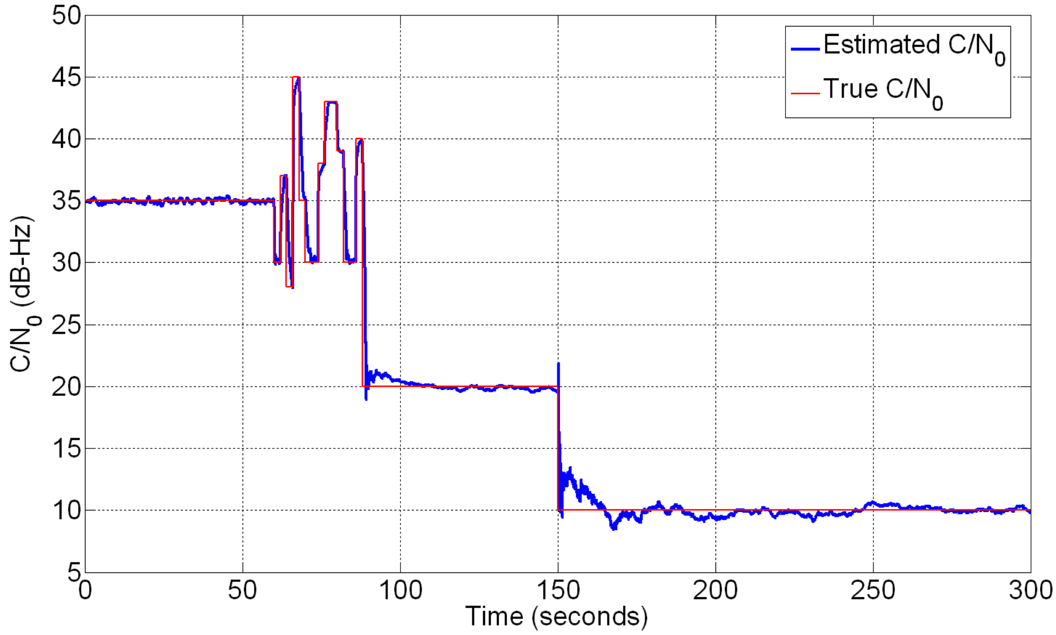

Figure B.1: C/N_0 Estimated Using Adaptive Filtering Techniques

APPENDIX C

ADDITIONAL PLOTS

This appendix includes additional plots from the different chapters of the dissertation. The plots are collected according to the chapter and section from which they come.

C.1 Additional Plots From Chapter 7 Section 3.2

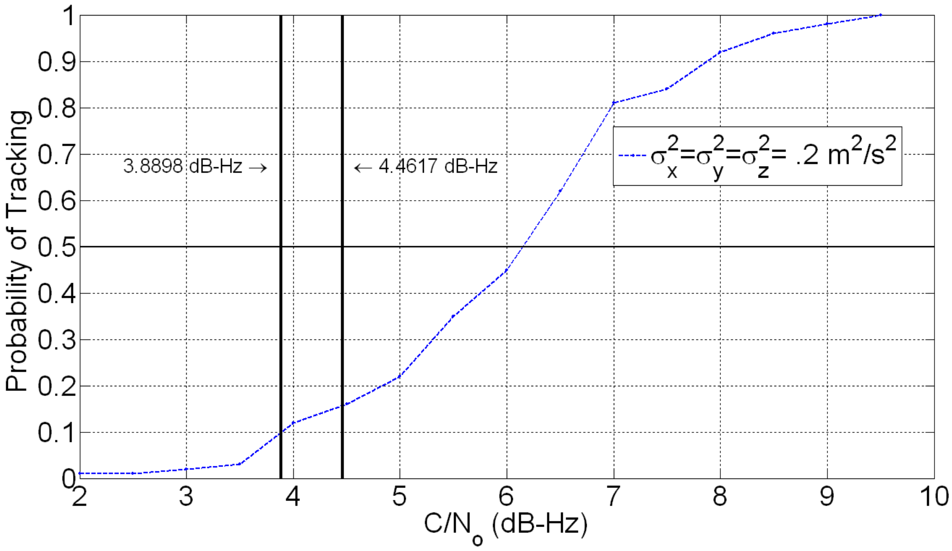

Figure C.1: VDFLL Monte Carlo Simulation Results for 5 Satellites and $\sigma^2 = .2\ m^2/s^2$

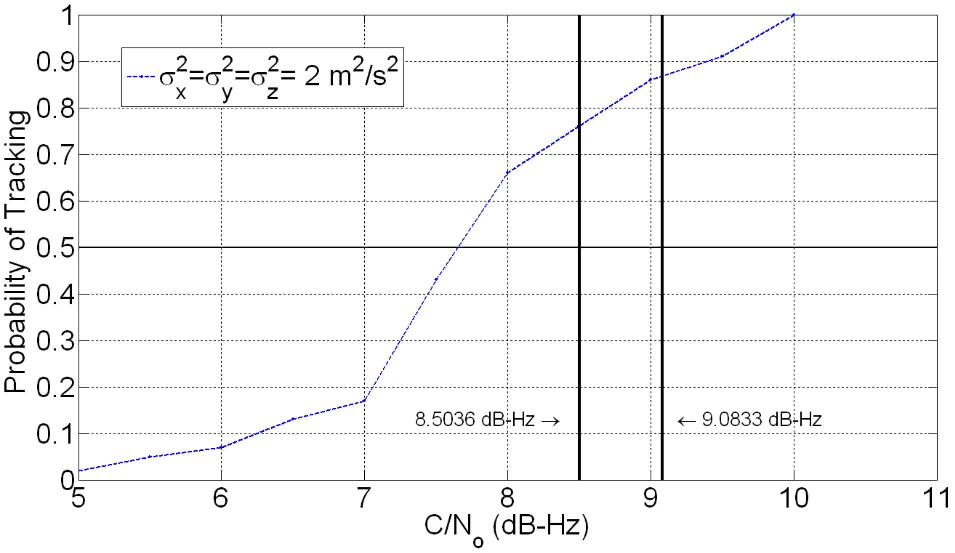

Figure C.2: VDFLL Monte Carlo Simulation Results for 5 Satellites and σ^2= 2 m^2/s^2

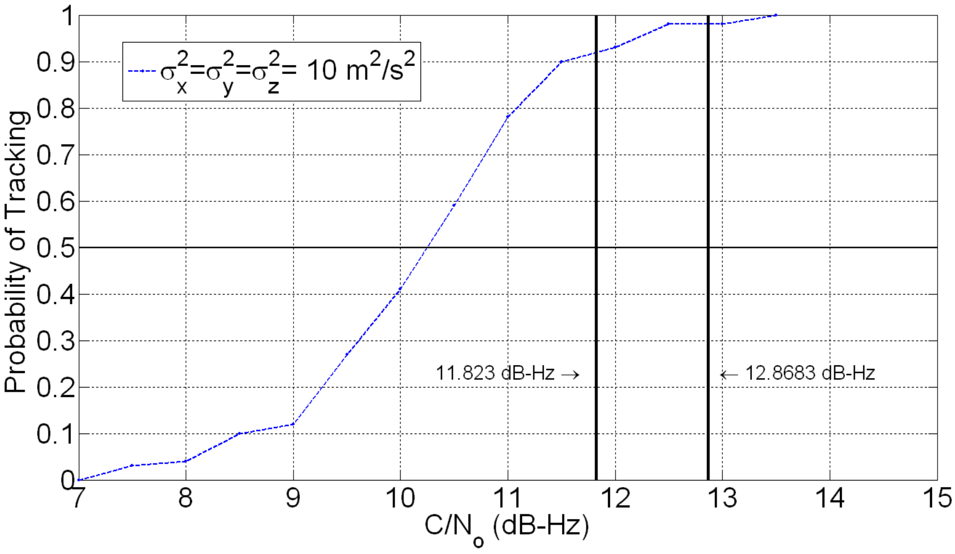

Figure C.3: VDFLL Monte Carlo Simulation Results for 5 Satellites and σ^2= 10 m^2/s^2

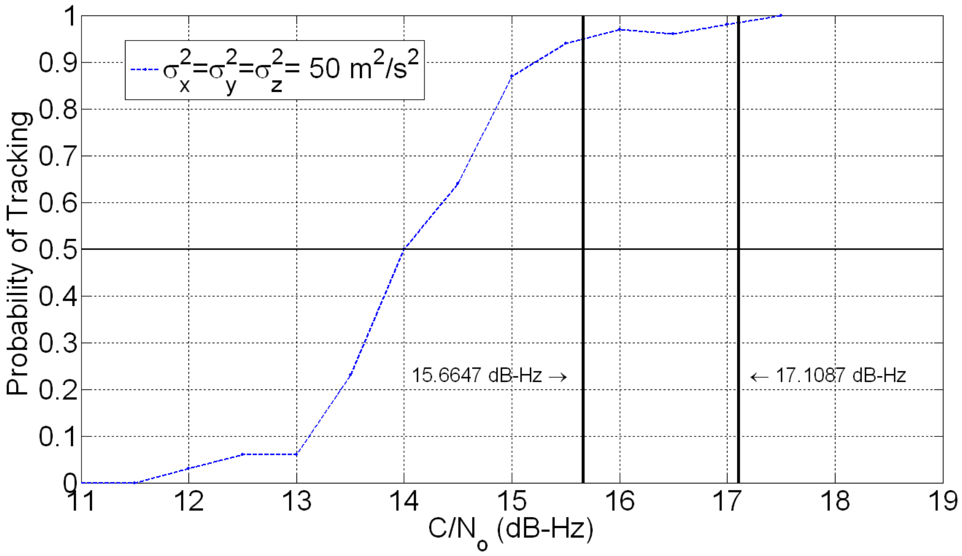

Figure C.4: VDFLL Monte Carlo Simulation Results for 5 Satellites and σ^2= 50 m^2/s^2

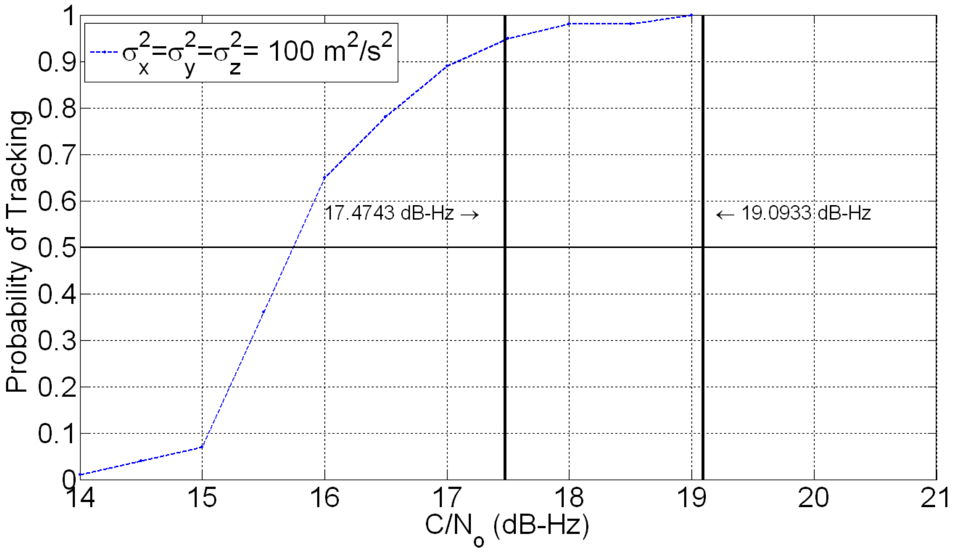

Figure C.5: VDFLL Monte Carlo Simulation Results for 5 Satellites and σ^2= 100 m^2/s^2

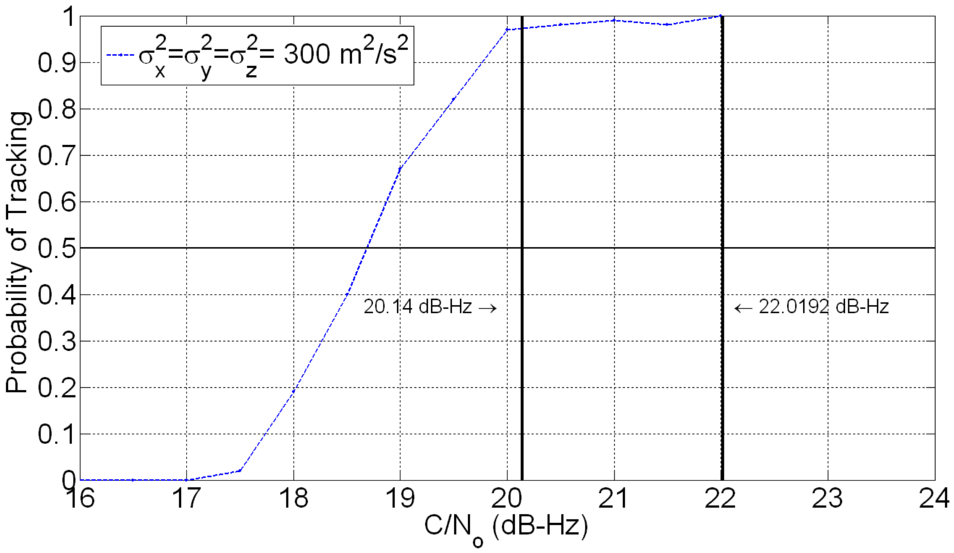

Figure C.6: VDFLL Monte Carlo Simulation Results for 5 Satellites and σ^2= 300 m^2/s^2

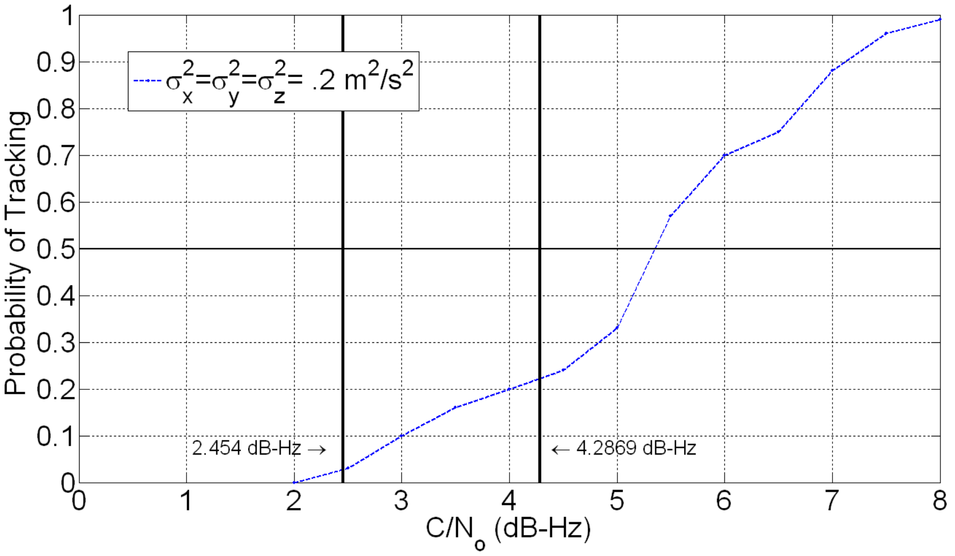

Figure C.7: VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= .2 m^2/s^2

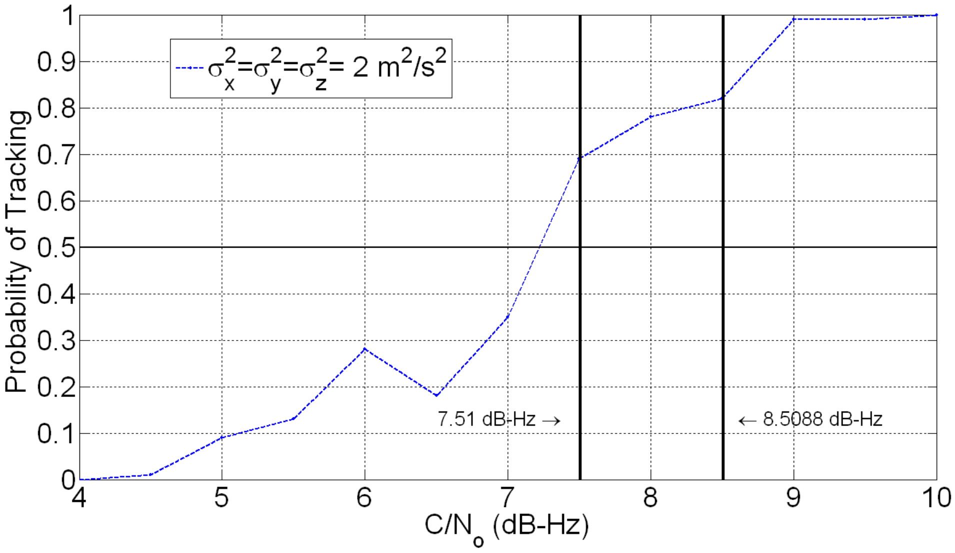

Figure C.8: VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= 2 m^2/s^2

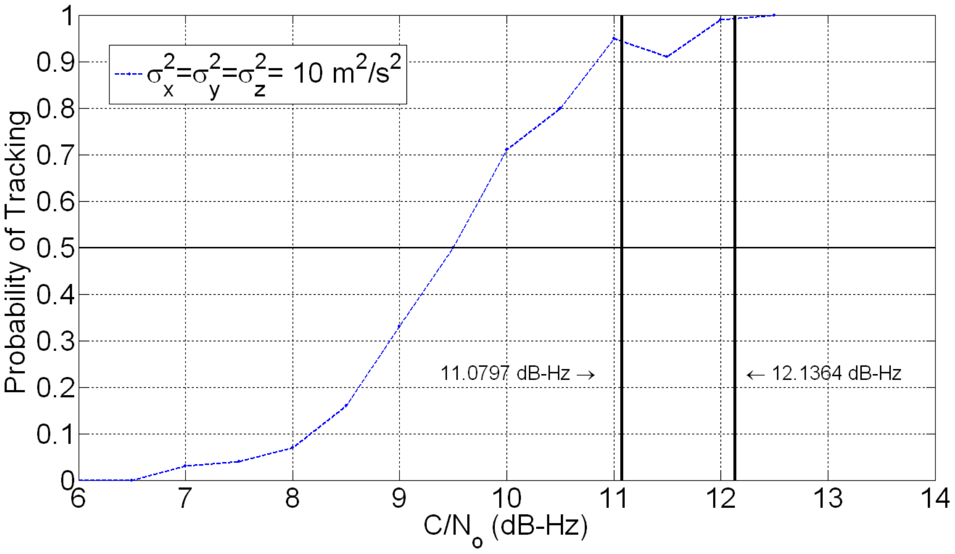

Figure C.9: VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= 10 m^2/s^2

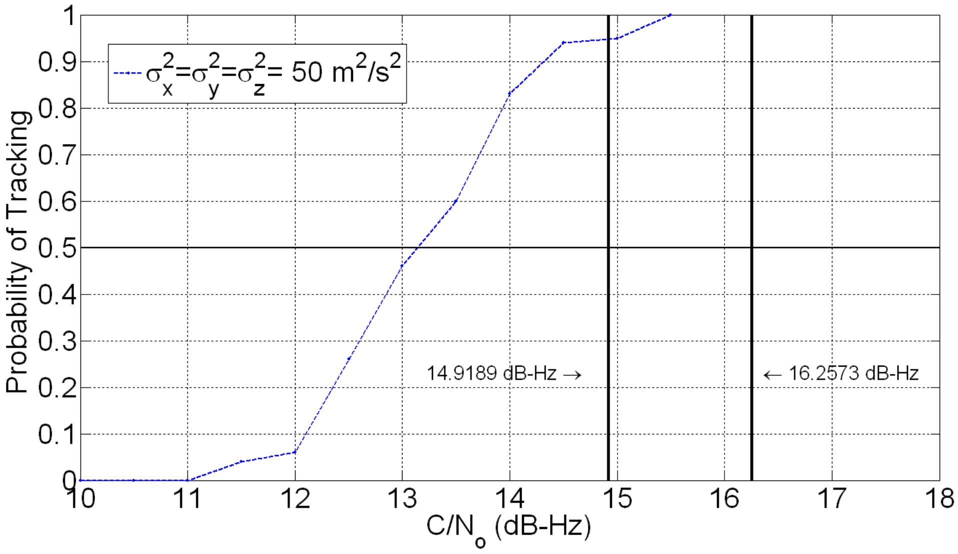

Figure C.10: VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= 50 m^2/s^2

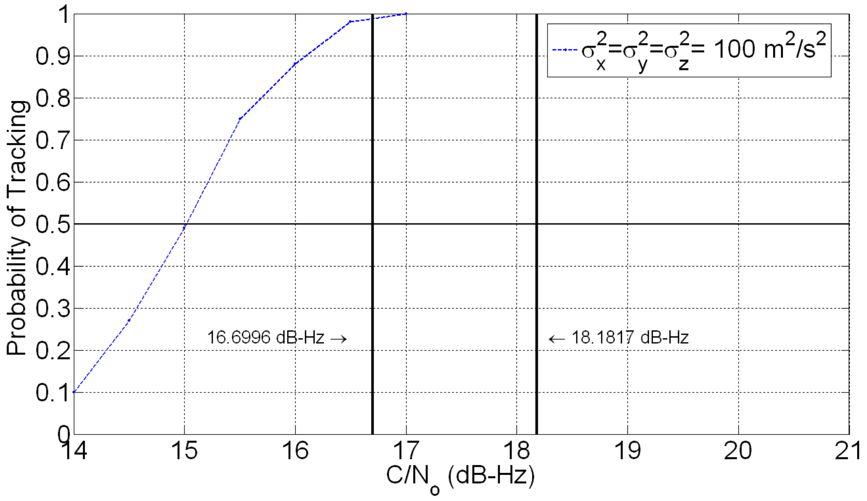

Figure C.11: VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= 100 m^2/s^2

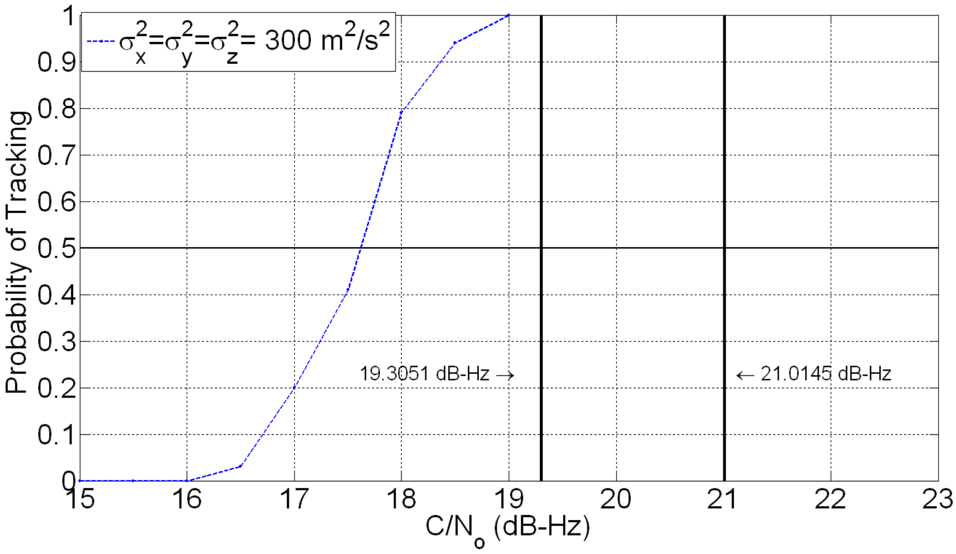

Figure C.12: VDFLL Monte Carlo Simulation Results for 7 Satellites and σ^2= 300 m^2/s^2

C.2 Additional Plots From Chapter 7 Section 3.4

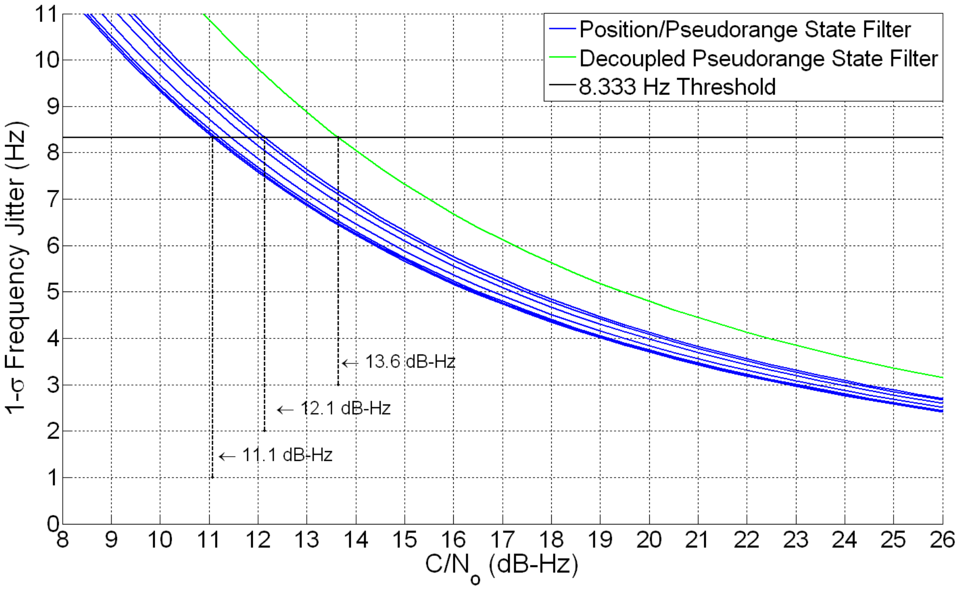

Figure C.13: Thermal Noise Analysis Results for 7 Satellites and σ^2= 10 m^2/s^2

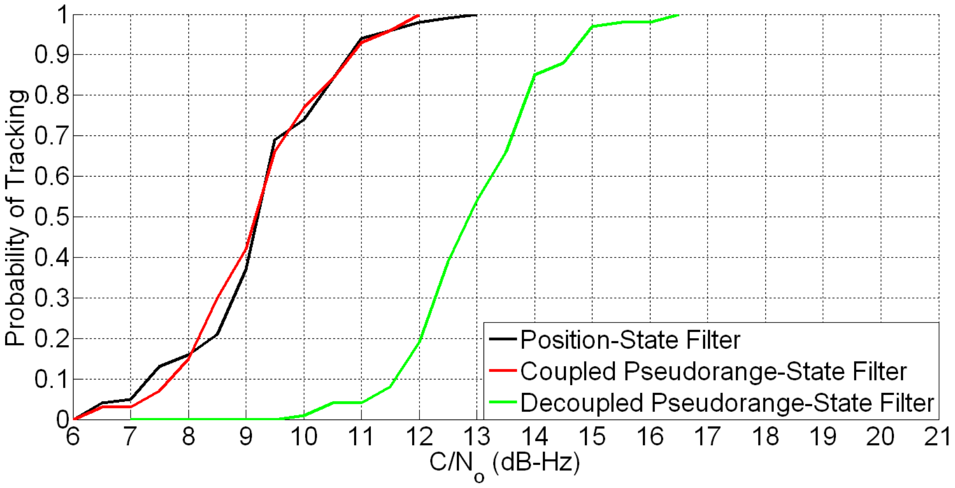

Figure C.14: Monte Carlo Simulation Results for 7 Satellites and $\sigma^2 = 10\ m^2/s^2$

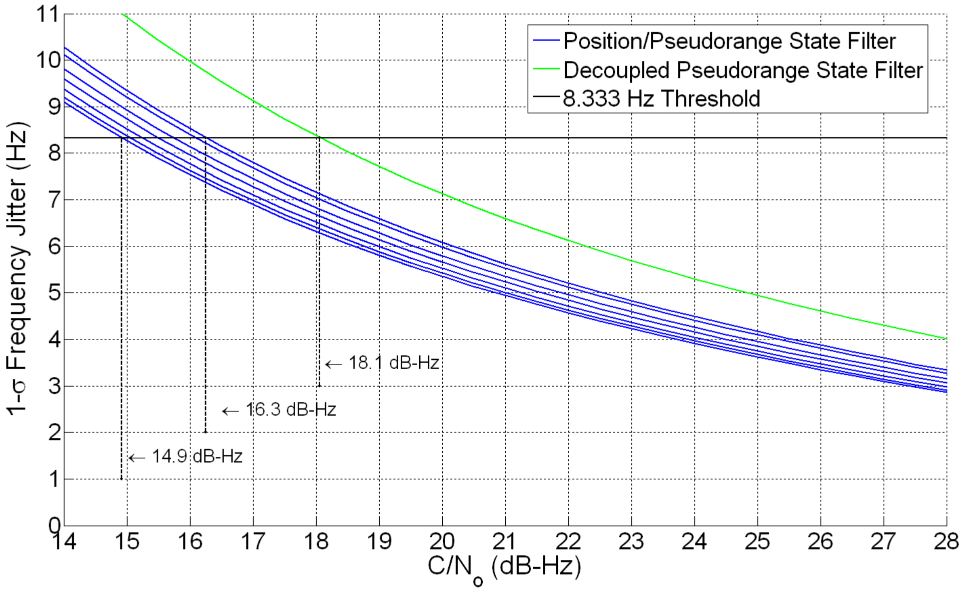

Figure C.15: Thermal Noise Analysis Results for 7 Satellites and $\sigma^2 = 50\ m^2/s^2$

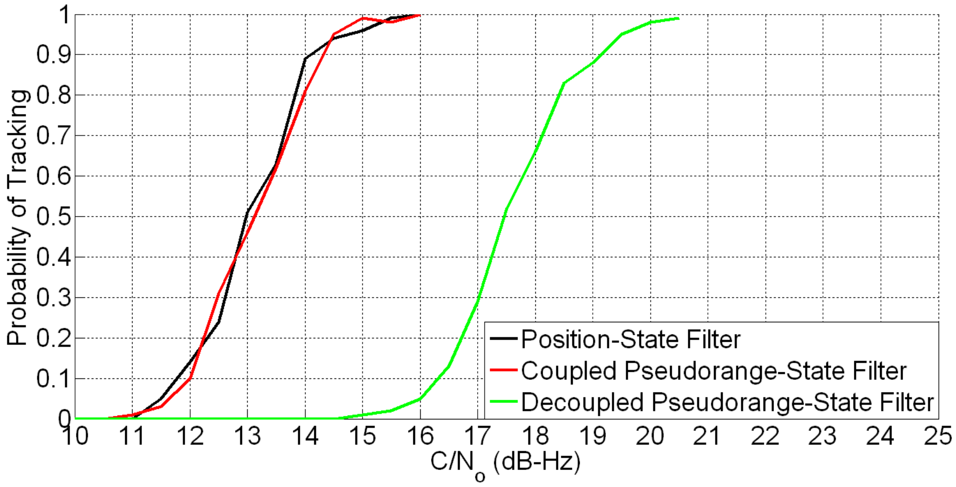

Figure C.16: Monte Carlo Simulation Results for 7 Satellites and σ^2= 50 m^2/s^2

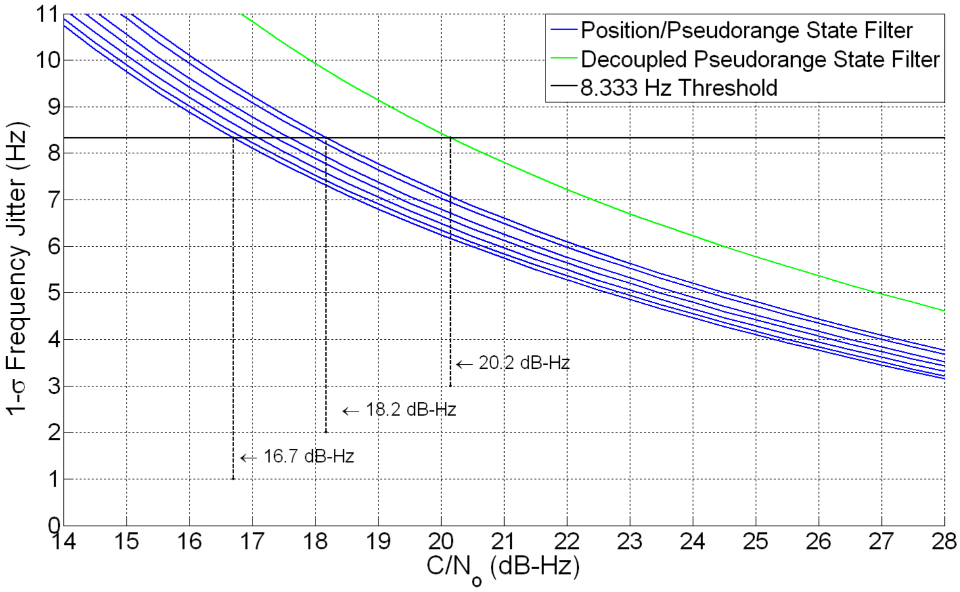

Figure C.17: Thermal Noise Analysis Results for 7 Satellites and σ^2= 100 m^2/s^2

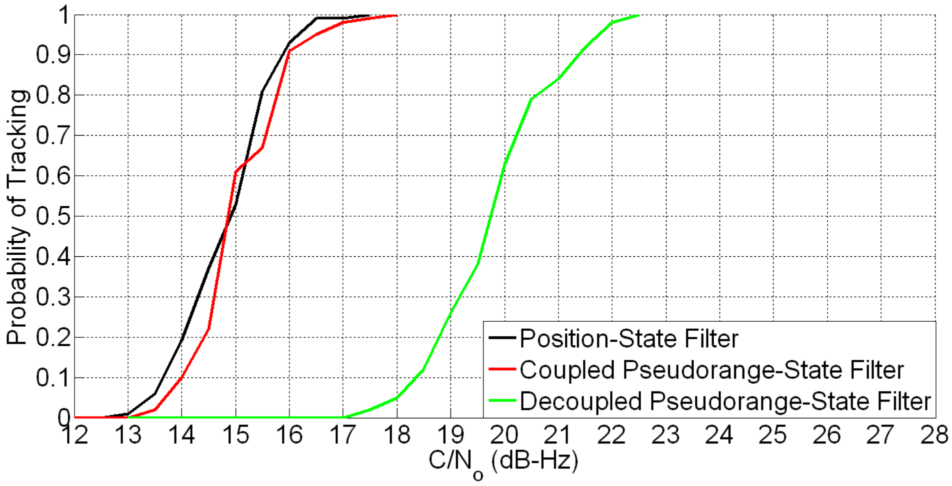

Figure C.18: 4.5Monte Carlo Simulation Results for 7 Satellites and σ^2= 100 m^2/s^2

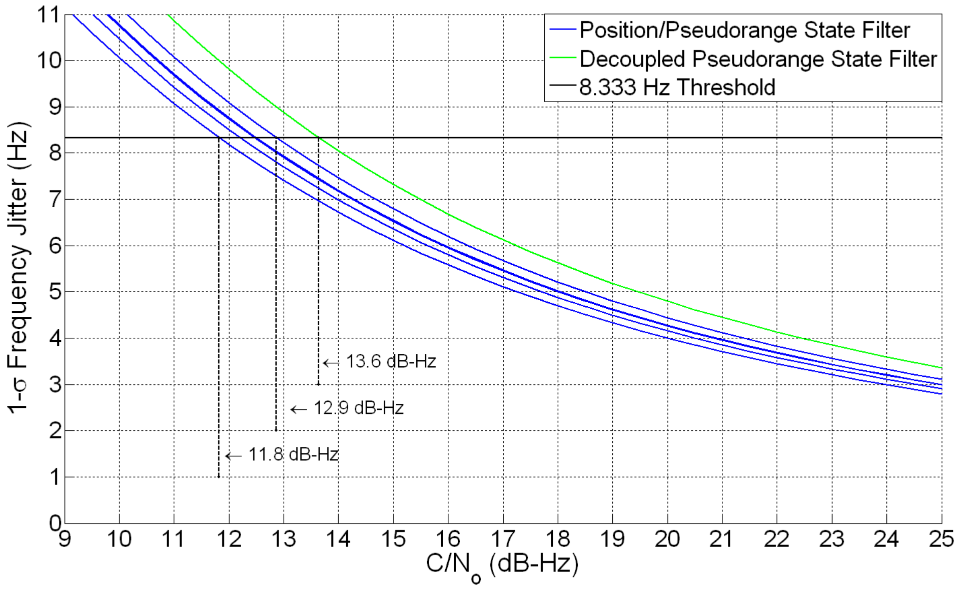

Figure C.19: Thermal Noise Analysis Results for 5 Satellites and σ^2= 10 m^2/s^2

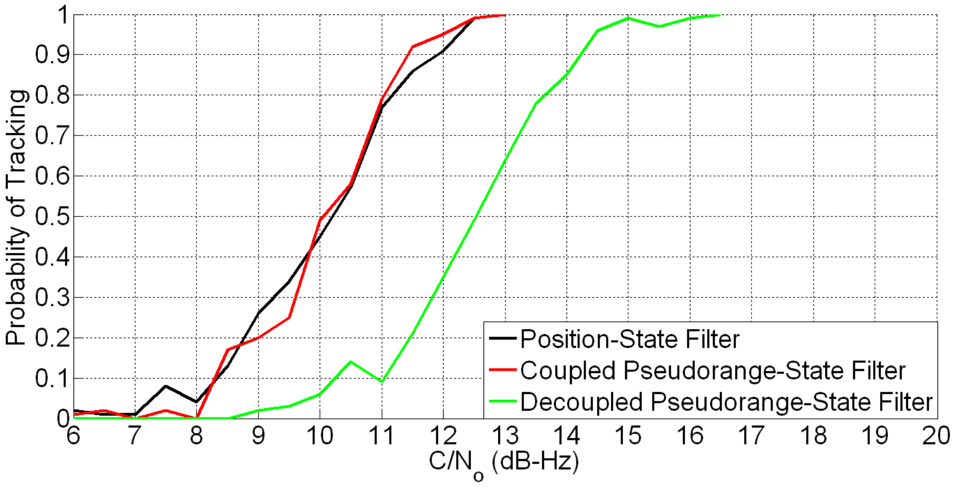

Figure C.20: Monte Carlo Simulation Results for 5 Satellites and σ^2= 10 m^2/s^2

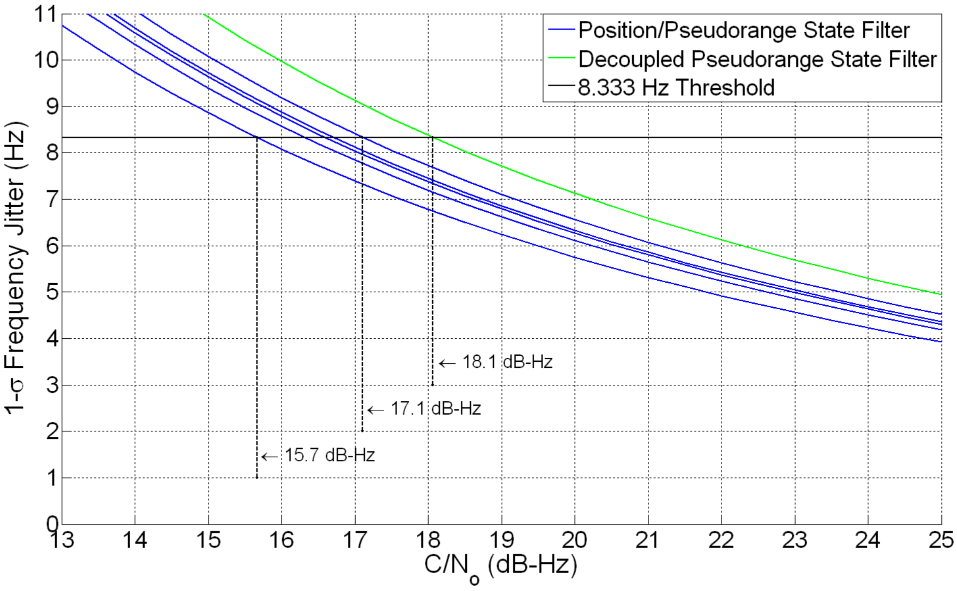

Figure C.21: Thermal Noise Analysis Results for 5 Satellites and σ^2= 50 m^2/s^2

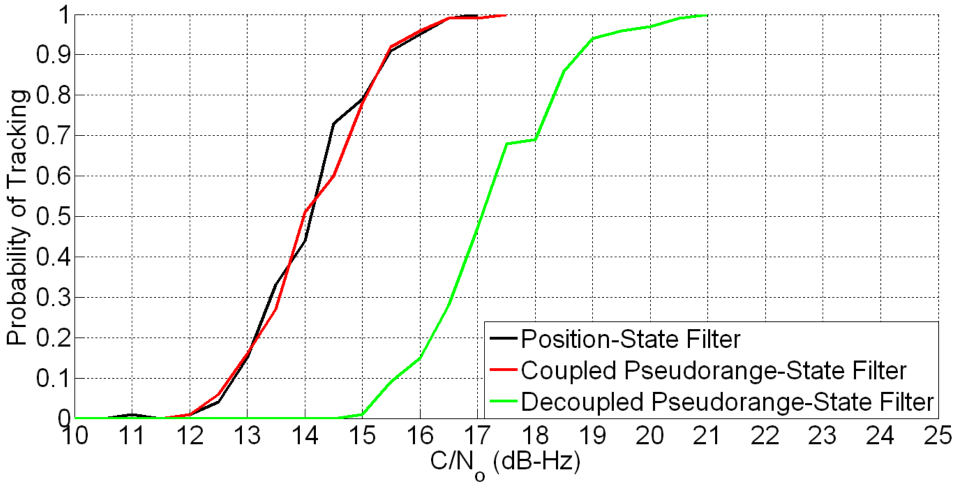

Figure C.22: Monte Carlo Simulation Results for 5 Satellites and σ^2= 50 m^2/s^2

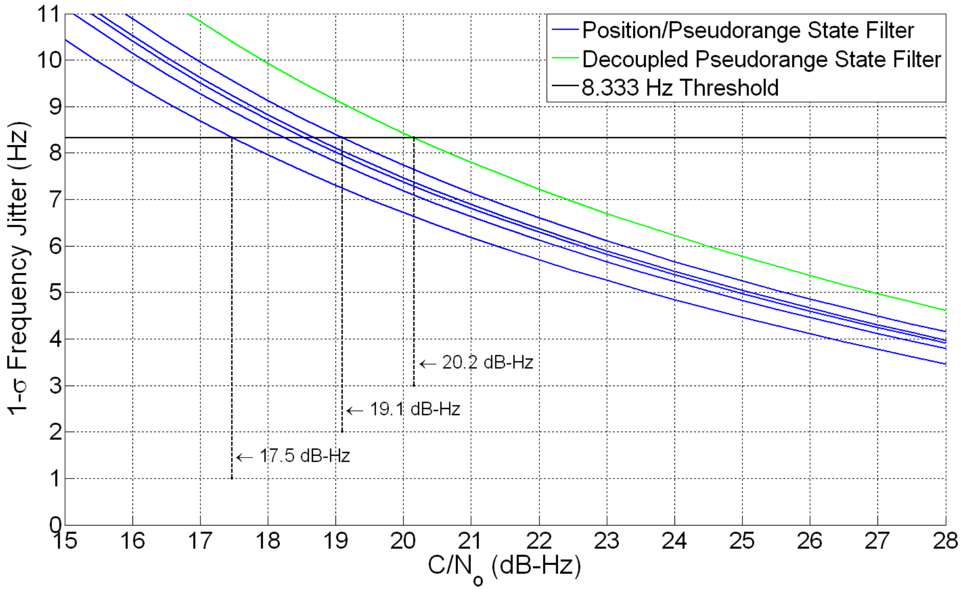

Figure C.23: Thermal Noise Analysis Results for 5 Satellites and σ^2= 100 m^2/s^2

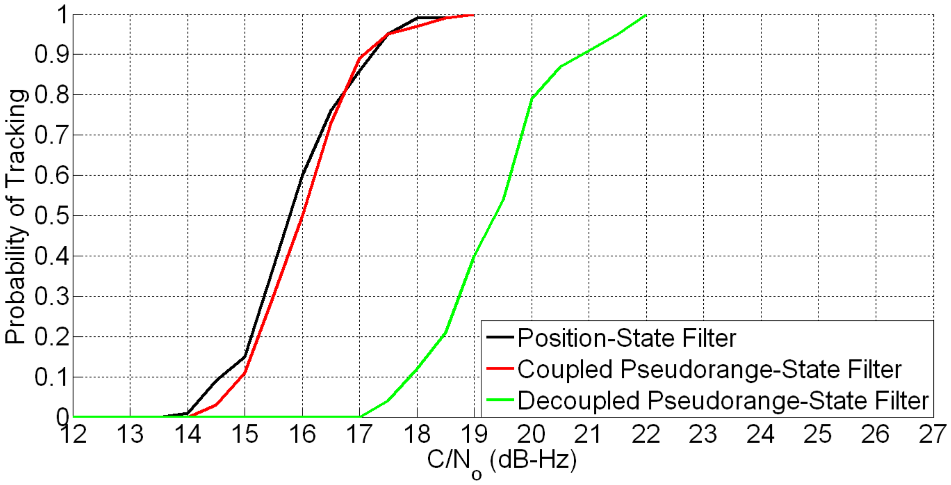

Figure C.24: Monte Carlo Simulation Results for 5 Satellites and σ^2= 100 m^2/s^2

C.3 Additional Plots From Chapter 8 Section 6.1

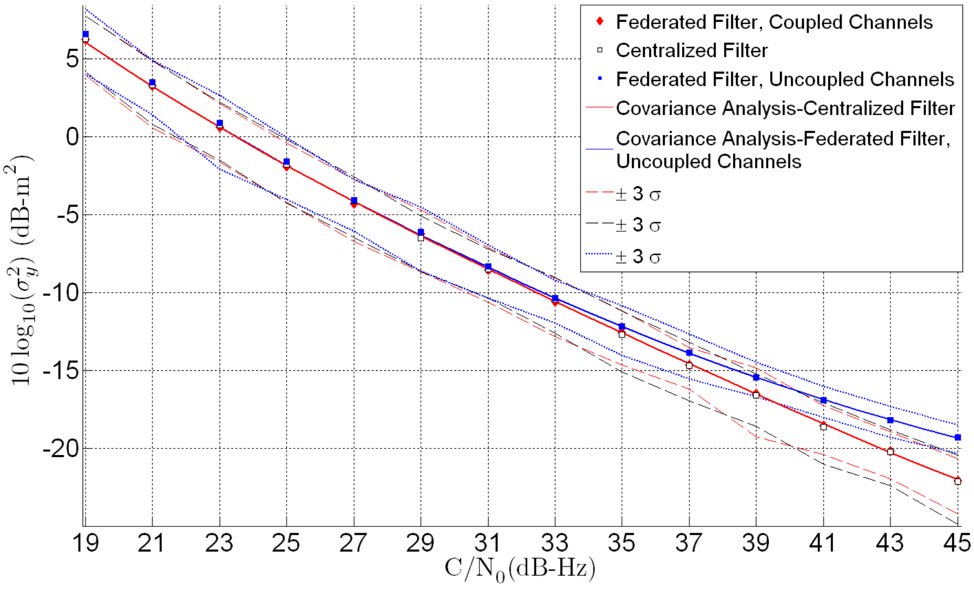

Figure C.25: Monte Carlo Simulation Results for the Y-Axis Position with Eleven Satellites and σ^2=2 m^2/s^2

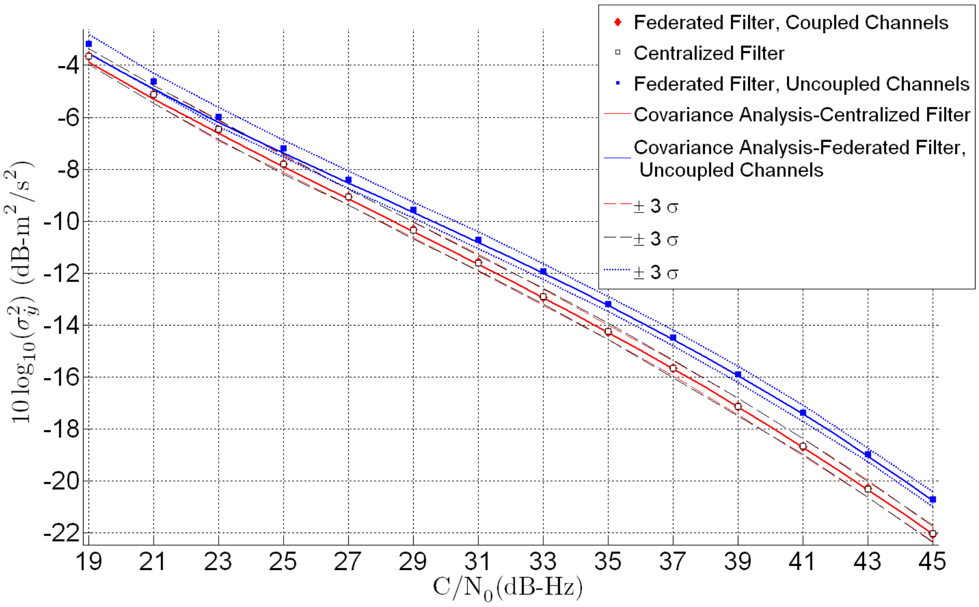

Figure C.26: Monte Carlo Simulation Results for the Y-Axis Velocity with Eleven Satellites and σ^2=2 m^2/s^2

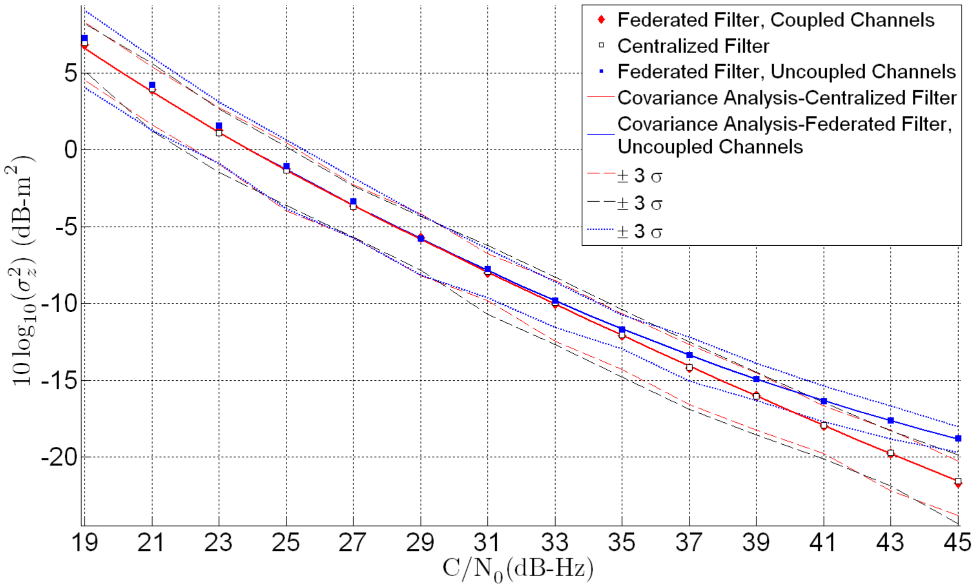

Figure C.27: Monte Carlo Simulation Results for the Z-Axis Position with Eleven Satellites and σ^2=2 m^2/s^2

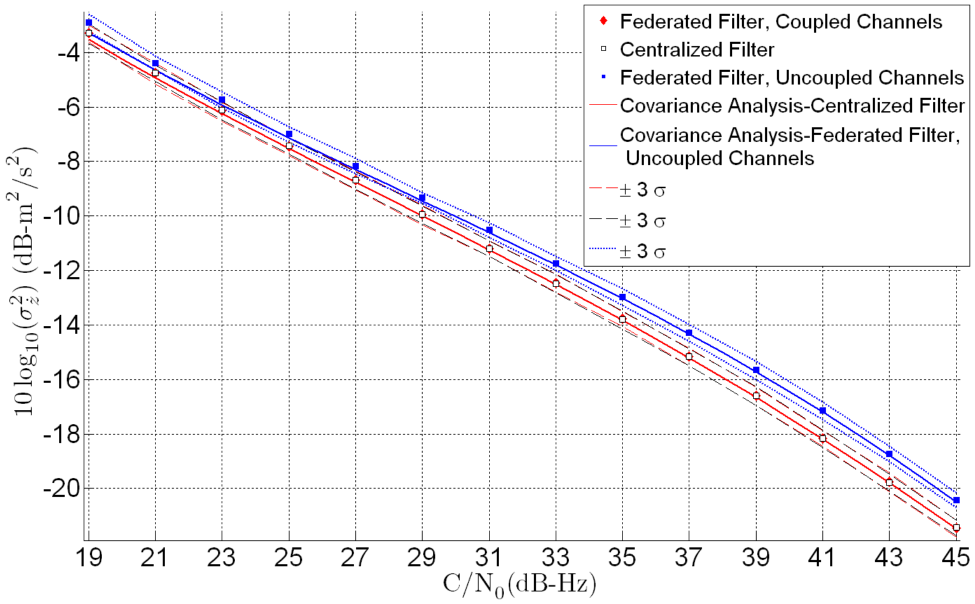

Figure C.28: Monte Carlo Simulation Results for the Z-Axis Velocity with Eleven Satellites and σ^2=2 m^2/s^2

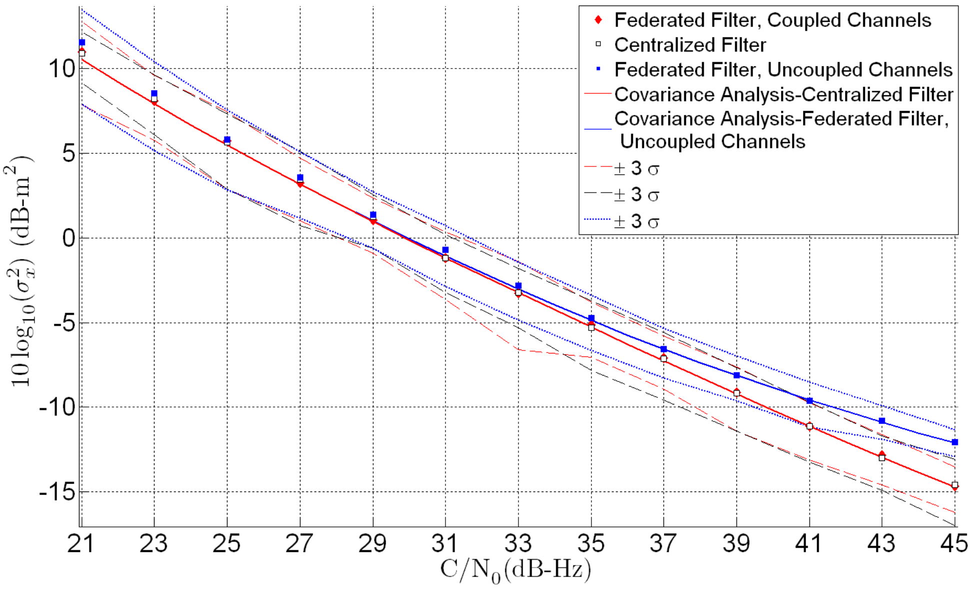

Figure C.29: Monte Carlo Simulation Results for the X-Axis Position with Eleven Satellites and σ^2=10 m^2/s^2

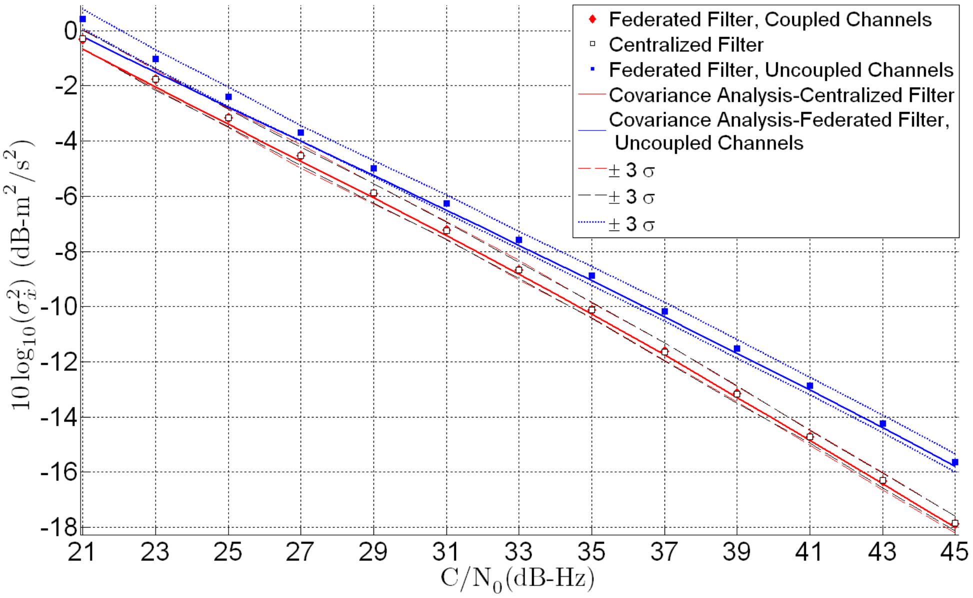

Figure C.30: Monte Carlo Simulation Results for the X-Axis Velocity with Eleven Satellites and σ^2=10 m^2/s^2

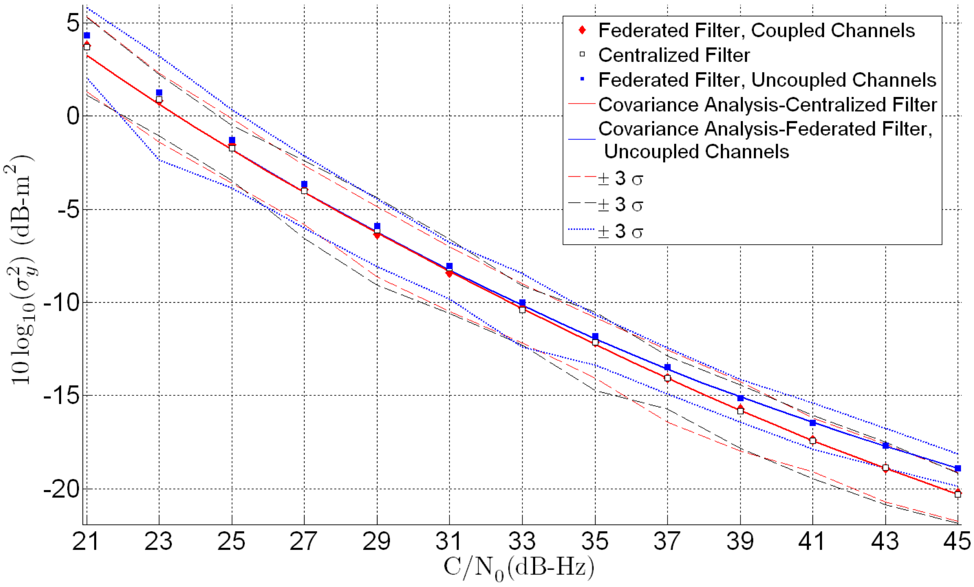

Figure C.31: Monte Carlo Simulation Results for the Y-Axis Position with Eleven Satellites and σ^2=10 m^2/s^2

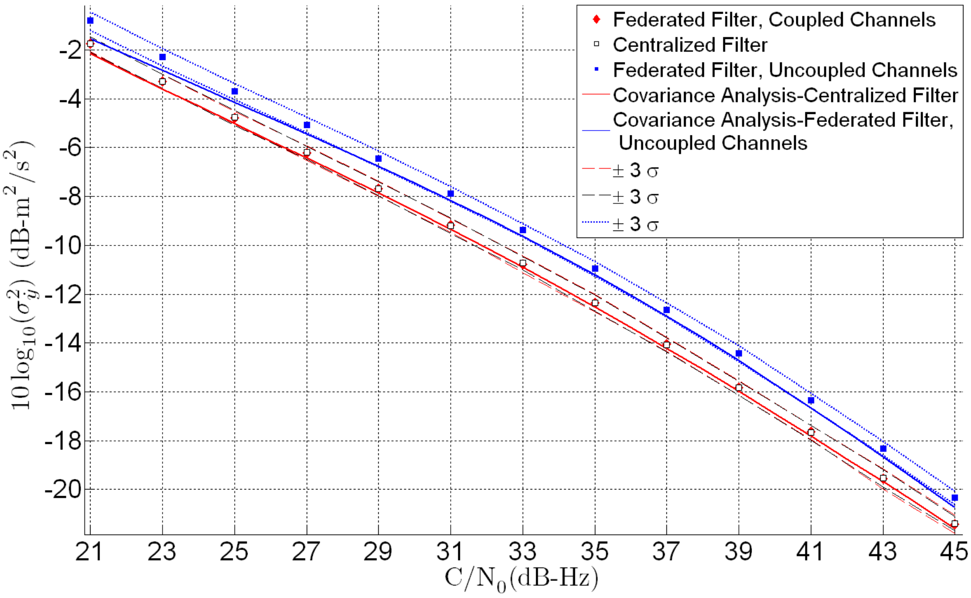

Figure C.32: Monte Carlo Simulation Results for the Y-Axis Velocity with Eleven Satellites and σ^2=10 m^2/s^2

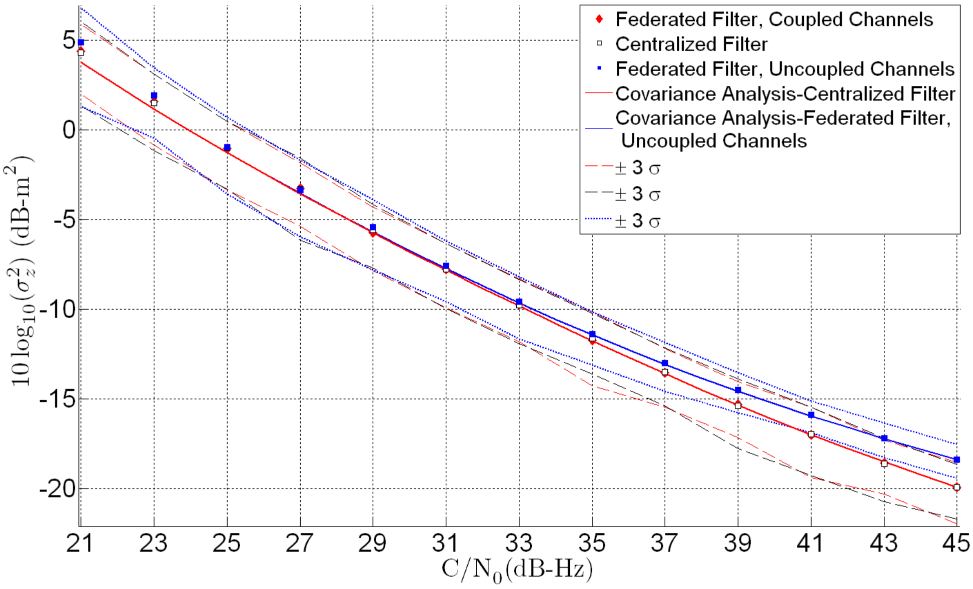

Figure C.33: Monte Carlo Simulation Results for the Z-Axis Position with Eleven Satellites and σ^2=10 m^2/s^2

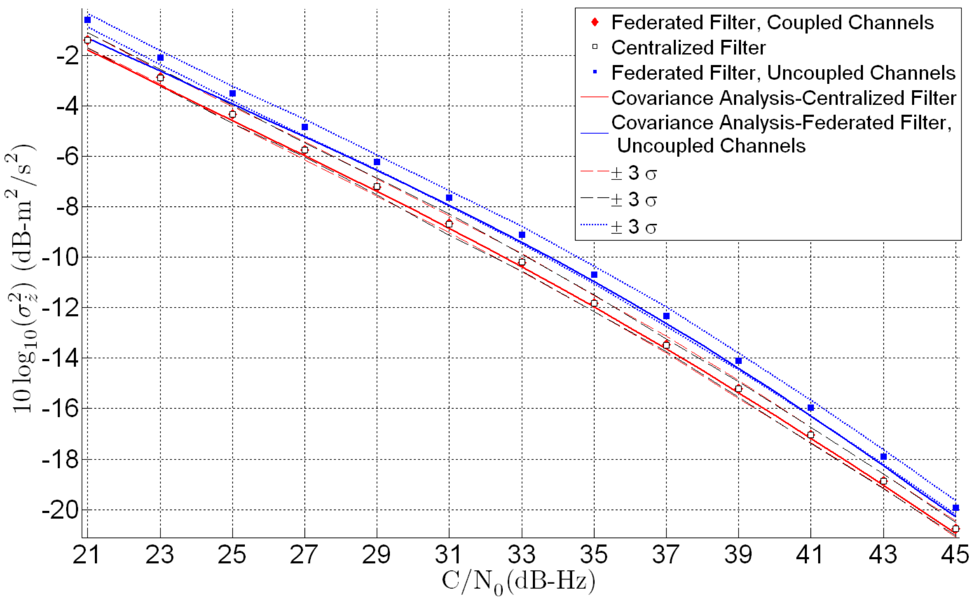

Figure C.34: Monte Carlo Simulation Results for the Z-Axis Velocity with Eleven Satellites and σ^2=10 m^2/s^2

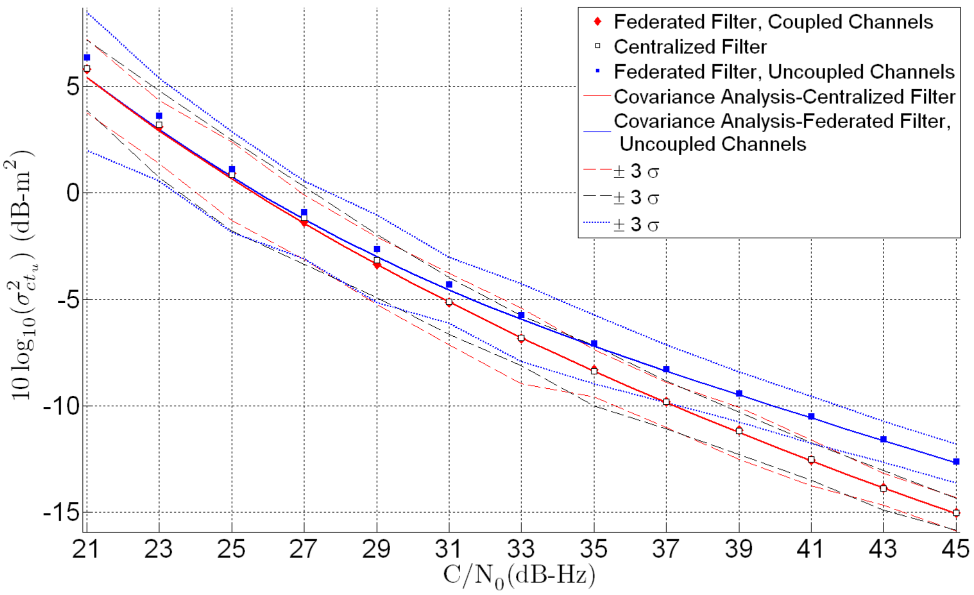

Figure C.35: Monte Carlo Simulation Results for the Clock Bias with Eleven Satellites and σ^2=10 m^2/s^2

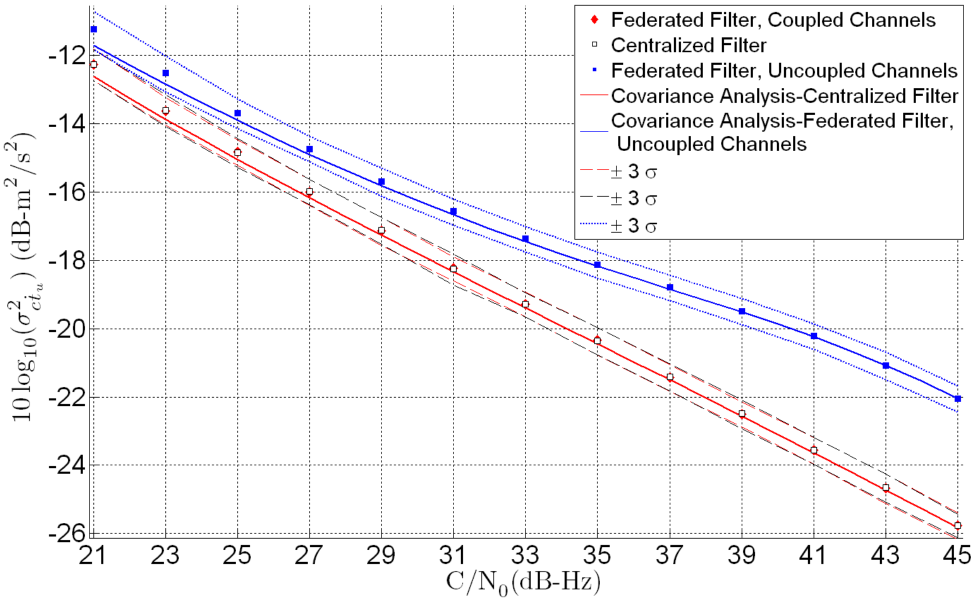

Figure C.36: Monte Carlo Simulation Results for the Clock Drift with Eleven Satellites and σ^2=10 m^2/s^2

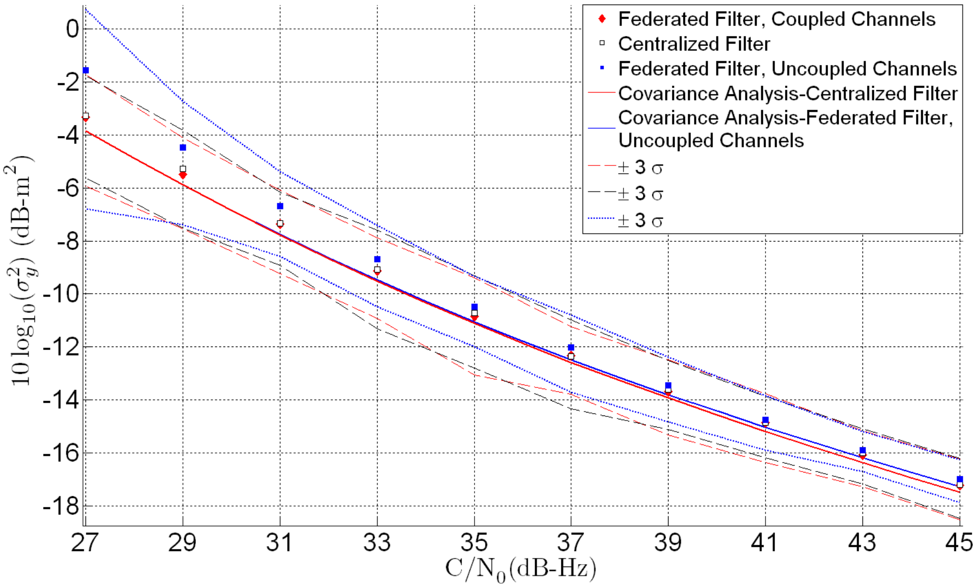

Figure C.37: Monte Carlo Simulation Results for the Y-Axis Position with Eleven Satellites and σ^2=50 m^2/s^2

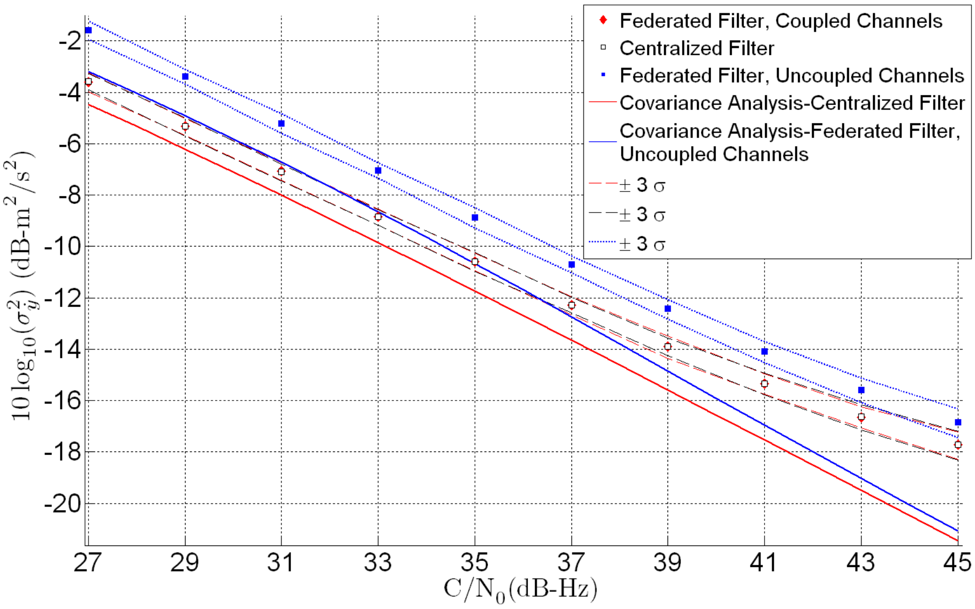

Figure C.38: Monte Carlo Simulation Results for the Y-Axis Velocity with Eleven Satellites and σ^2=50 m^2/s^2

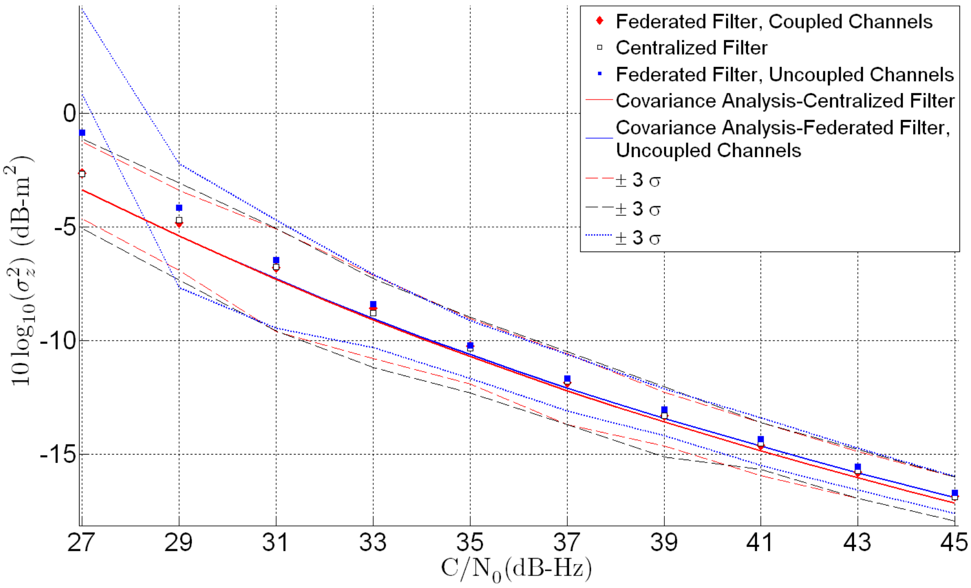

Figure C.39: Monte Carlo Simulation Results for the Z-Axis Position with Eleven Satellites and σ^2=50 m^2/s^2

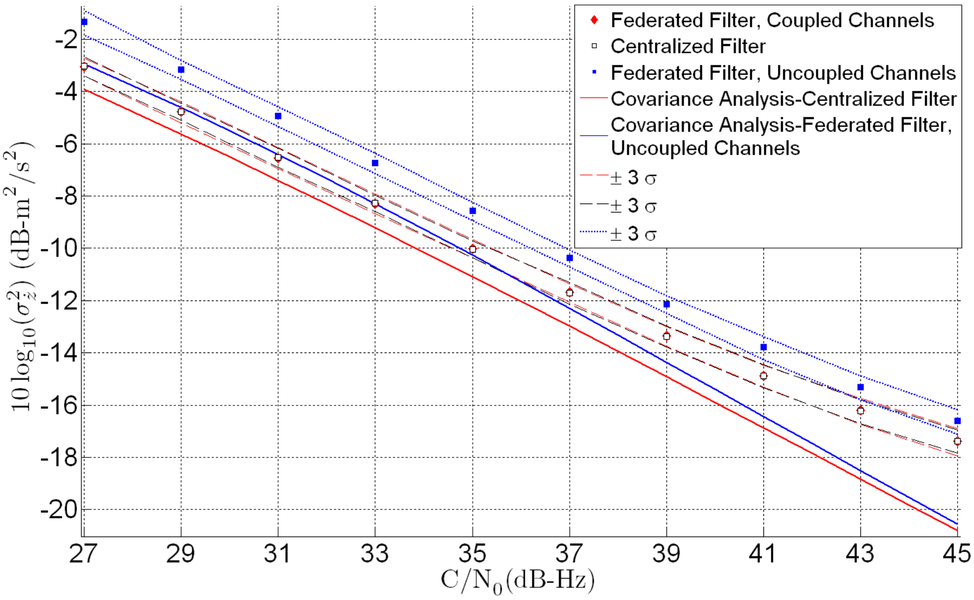

Figure C.40: Monte Carlo Simulation Results for the Z-Axis Velocity with Eleven Satellites and σ^2=50 m^2/s^2

C.4 Additional Plots From Chapter 8 Section 6.2

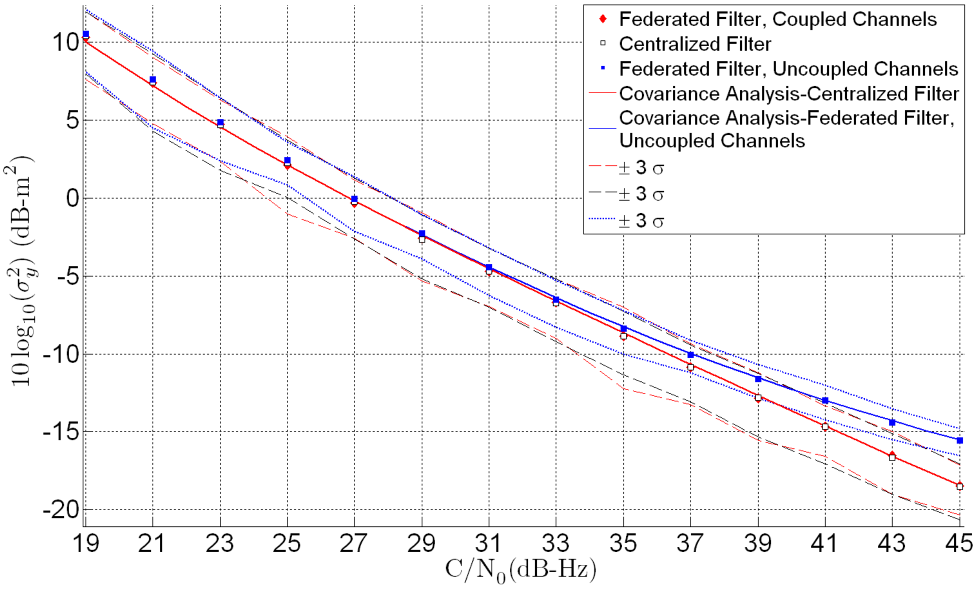

Figure C.41: Monte Carlo Simulation Results for the Y-Axis Position with Five Satellites and σ^2=2 m^2/s^2

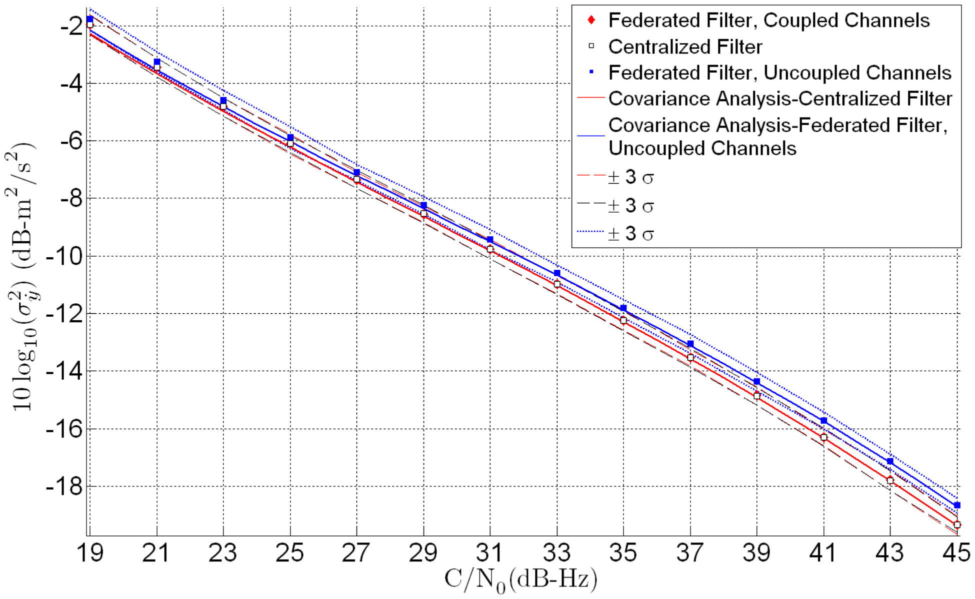

Figure C.42: Monte Carlo Simulation Results for the Y-Axis Velocity with Five Satellites and σ^2=2 m^2/s^2

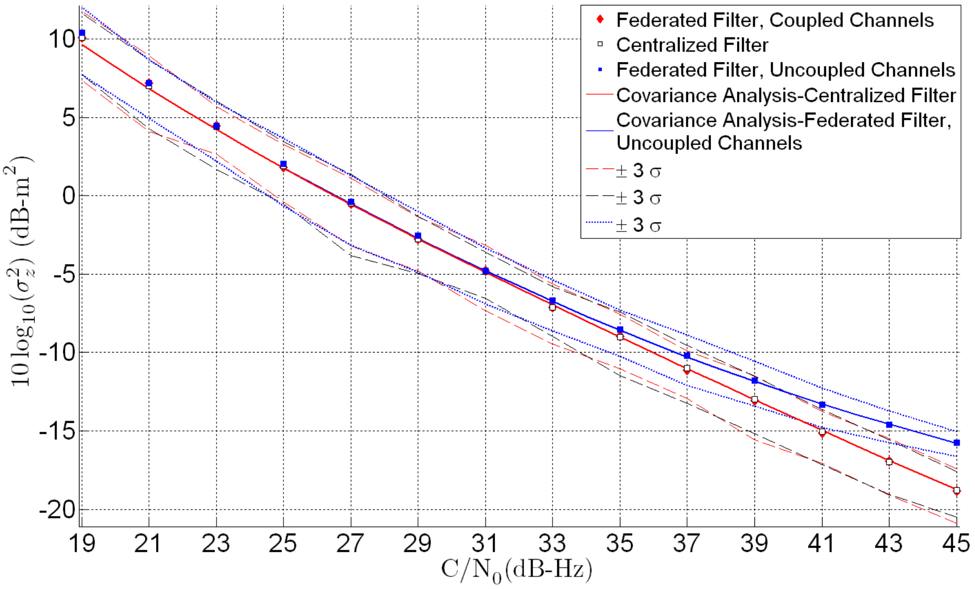

Figure C.43: Monte Carlo Simulation Results for the Z-Axis Position with Five Satellites and σ^2=2 m^2/s^2

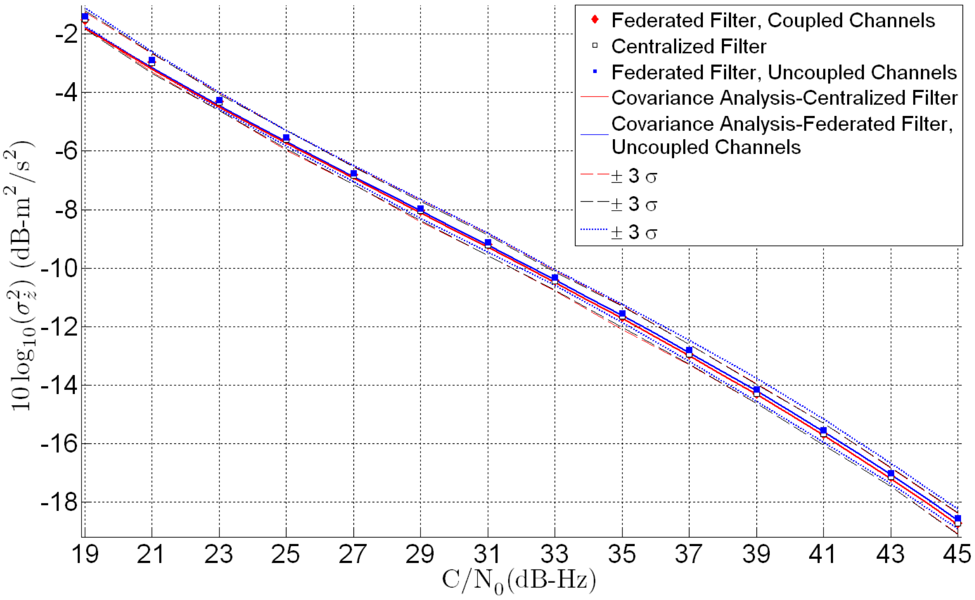

Figure C.44: Monte Carlo Simulation Results for the Z-Axis Velocity with Five Satellites and σ^2=2 m^2/s^2

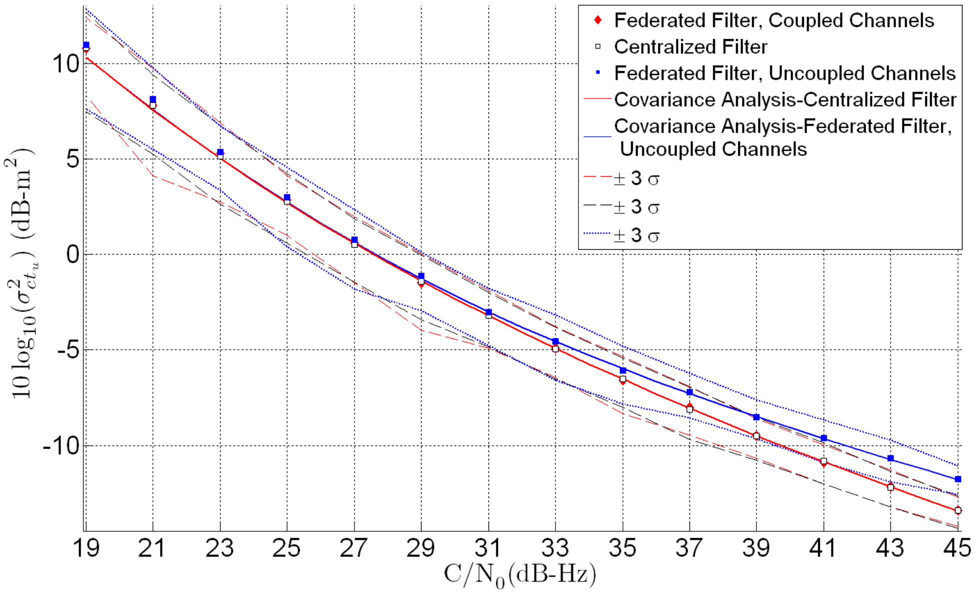

Figure C.45: Monte Carlo Simulation Results for the Clock Bias with Five Satellites and σ^2=2 m^2/s^2

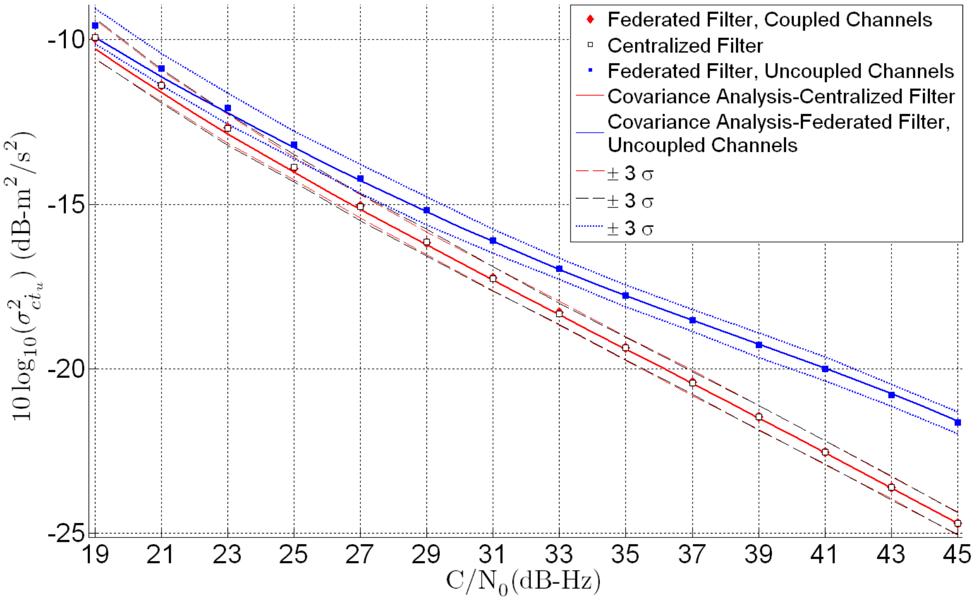

Figure C.46: Monte Carlo Simulation Results for the Clock Drift with Five Satellites and σ^2=2 m^2/s^2

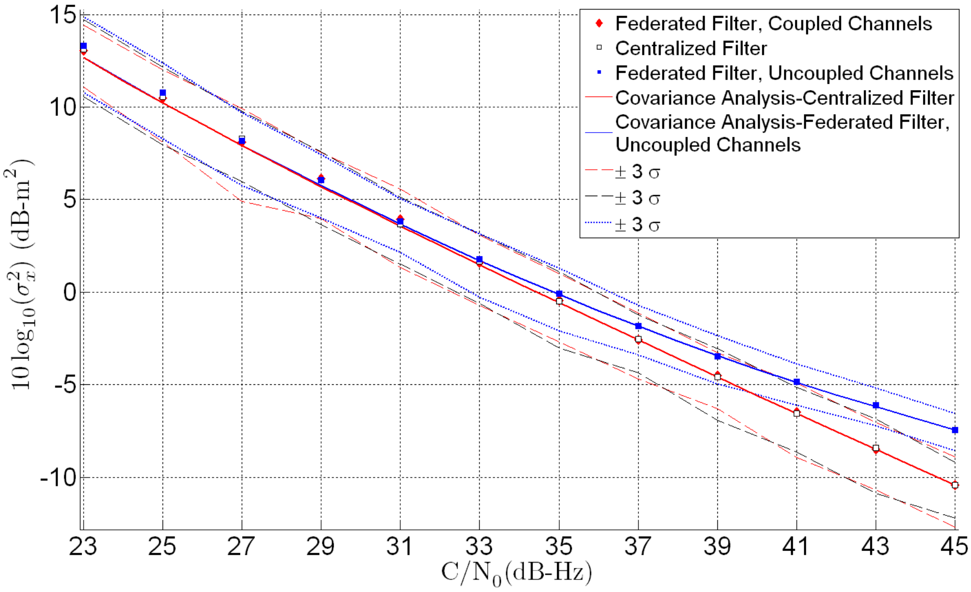

Figure C.47: Monte Carlo Simulation Results for the X-Axis Position with Five Satellites and σ^2=10 m^2/s^2

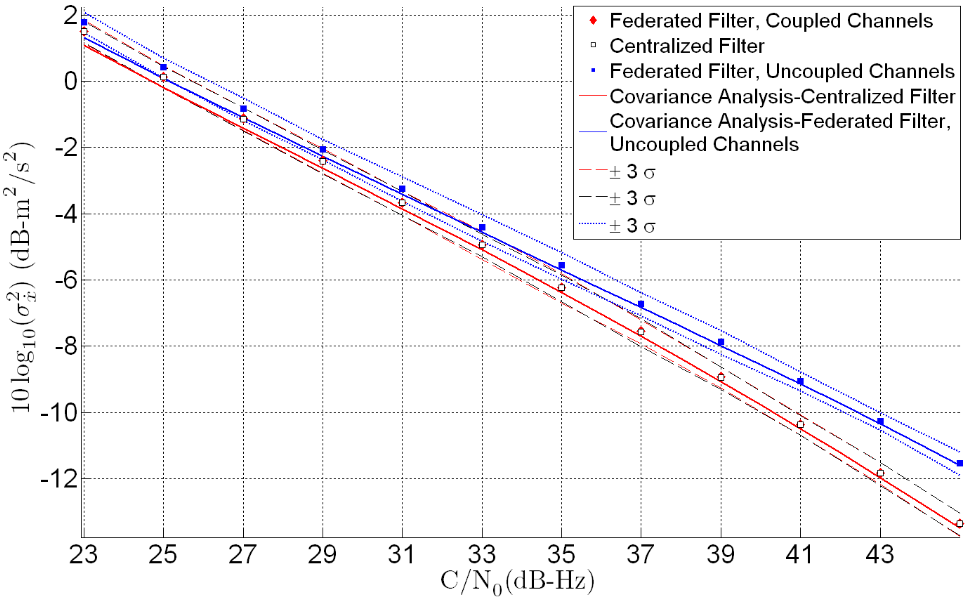

Figure C.48: Monte Carlo Simulation Results for the X-Axis Velocity with Five Satellites and σ^2=10 m^2/s^2

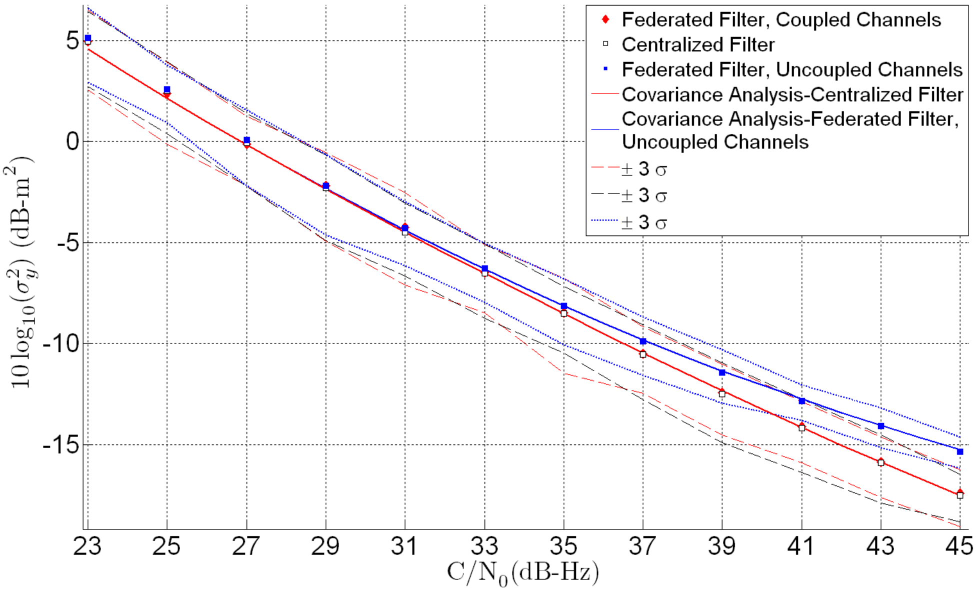

Figure C.49: Monte Carlo Simulation Results for the Y-Axis Position with Five Satellites and σ^2=10 m^2/s^2

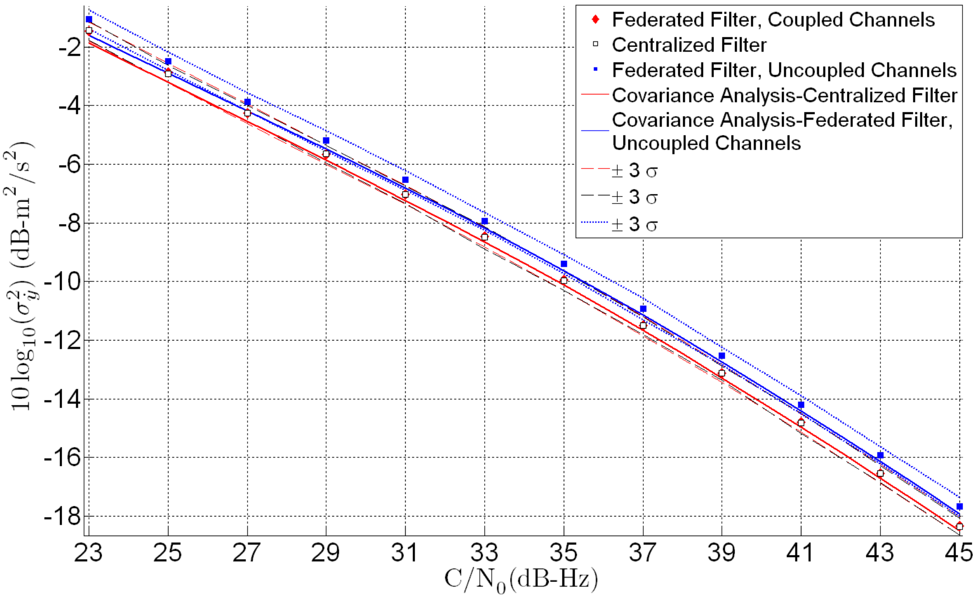

Figure C.50: Monte Carlo Simulation Results for the Y-Axis Velocity with Five Satellites and σ^2=10 m^2/s^2

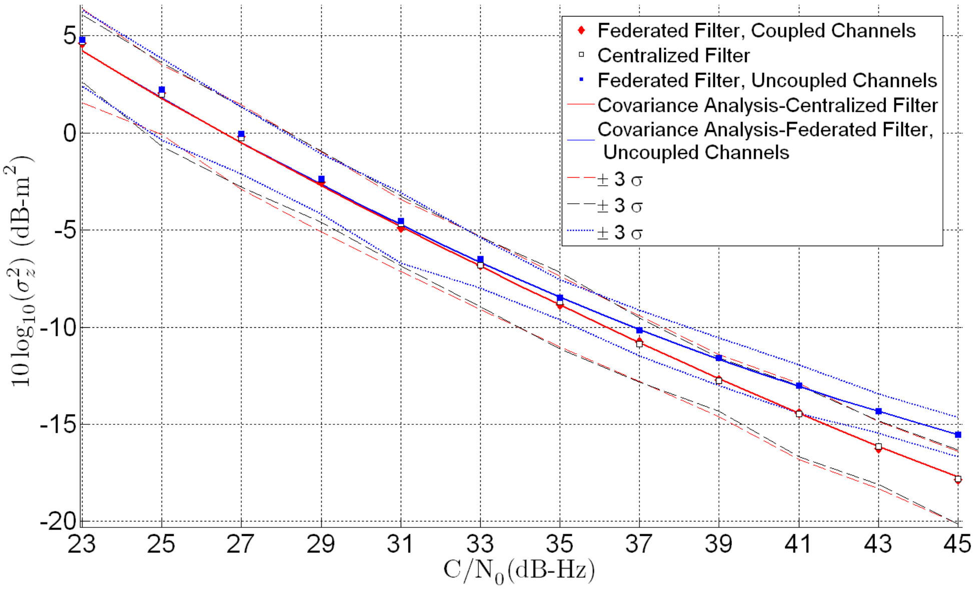

Figure C.51: Monte Carlo Simulation Results for the Z-Axis Position with Five Satellites and σ^2=10 m^2/s^2

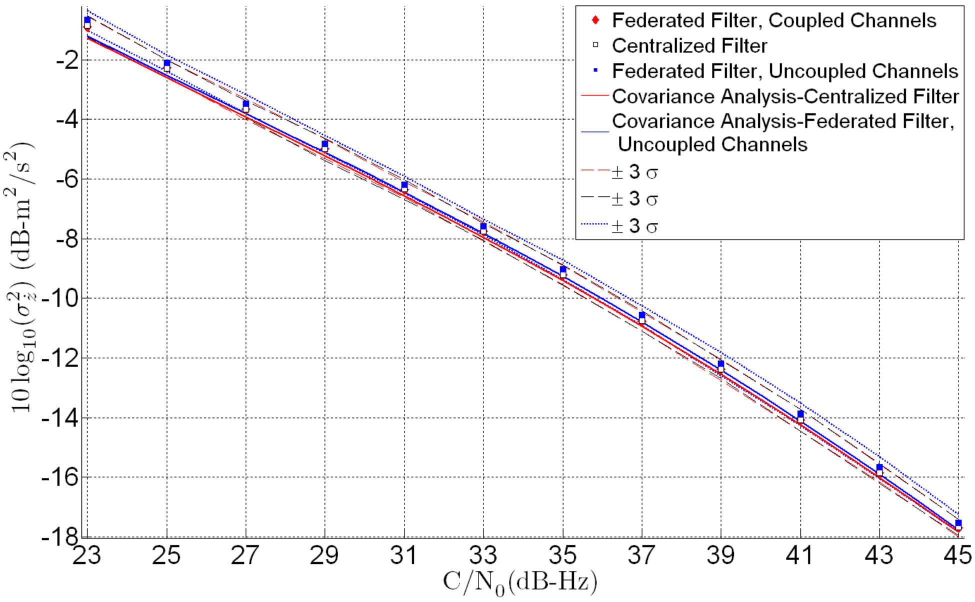

Figure C.52: Monte Carlo Simulation Results for the Z-Axis Velocity with Five Satellites and σ^2=10 m^2/s^2

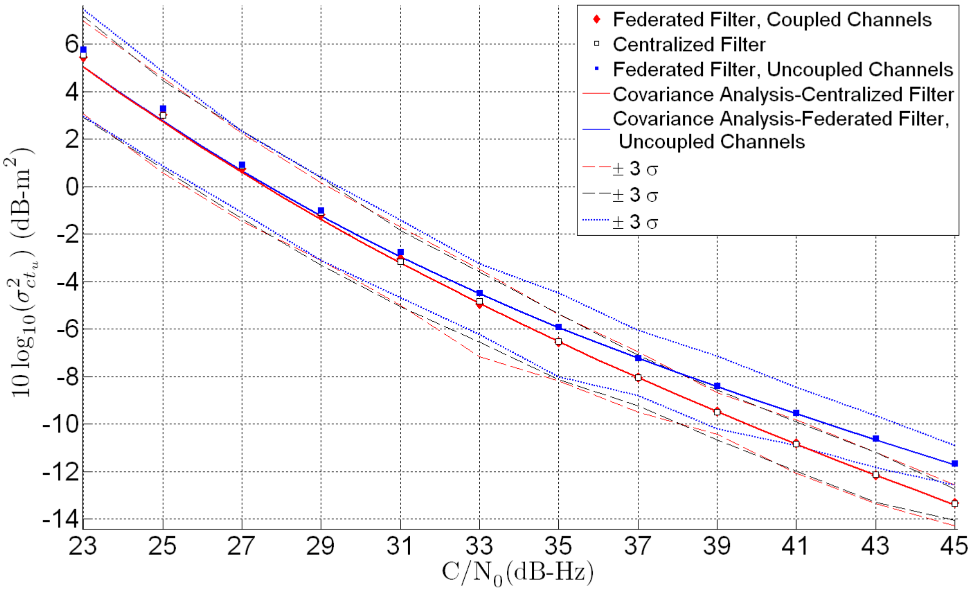

Figure C.53: Monte Carlo Simulation Results for the Clock Bias with Five Satellites and σ^2=10 m^2/s^2

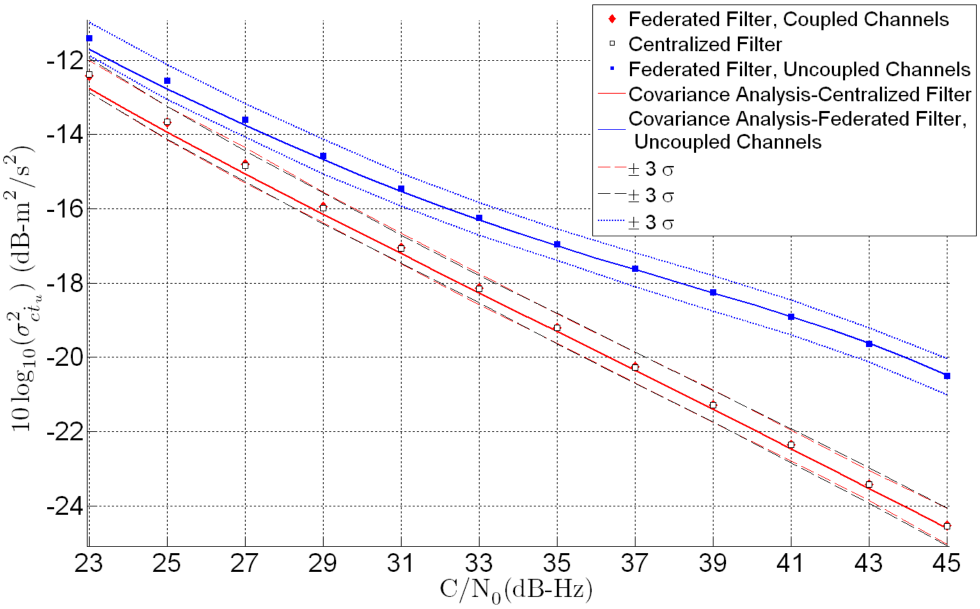

Figure C.54: Monte Carlo Simulation Results for the Clock Drift with Five Satellites and σ^2=10 m^2/s^2

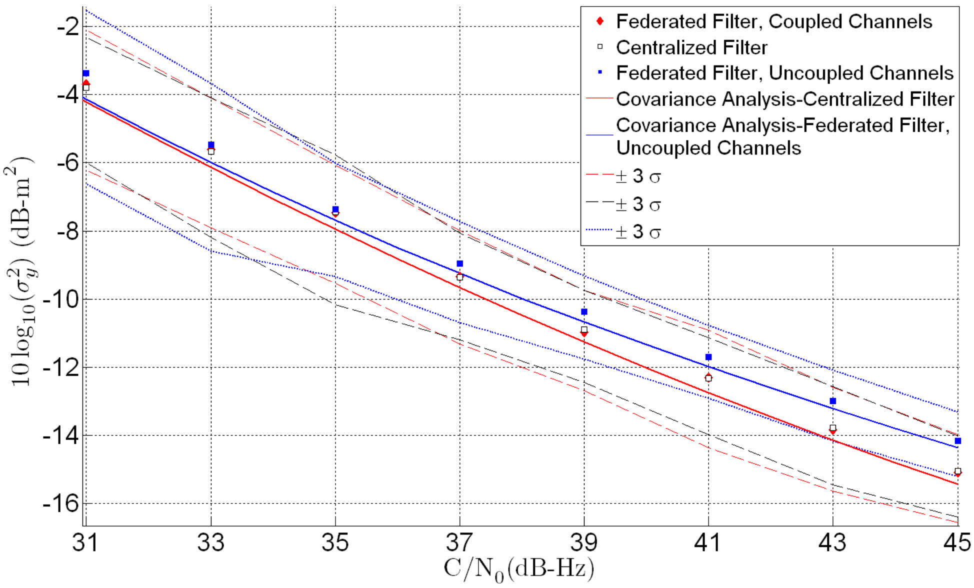

Figure C.55: Monte Carlo Simulation Results for the Y-Axis Position with Five Satellites and σ^2=50 m^2/s^2

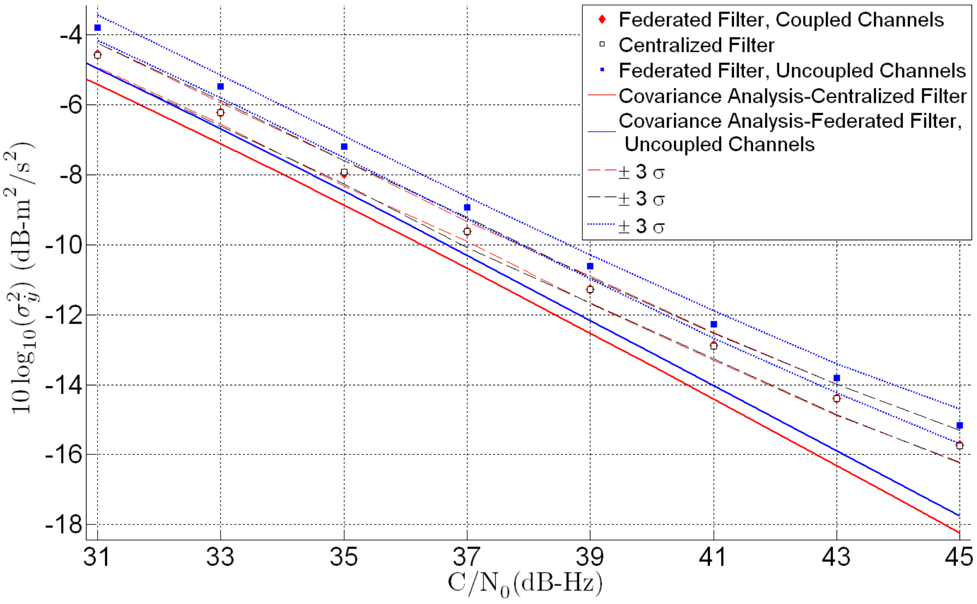

Figure C.56: Monte Carlo Simulation Results for the Y-Axis Velocity with Five Satellites and σ^2=50 m^2/s^2

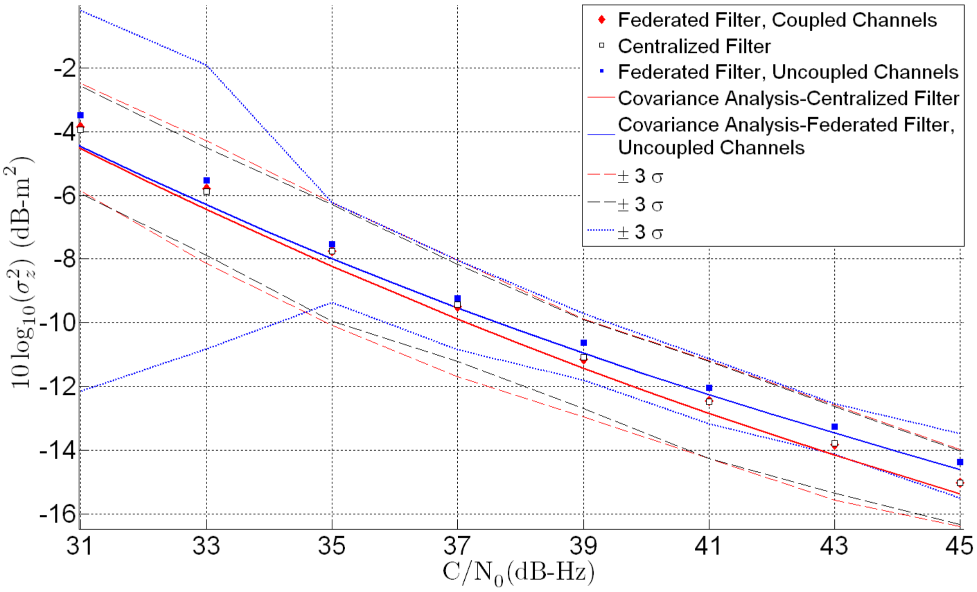

Figure C.57: Monte Carlo Simulation Results for the Z-Axis Position with Five Satellites and σ^2=50 m^2/s^2

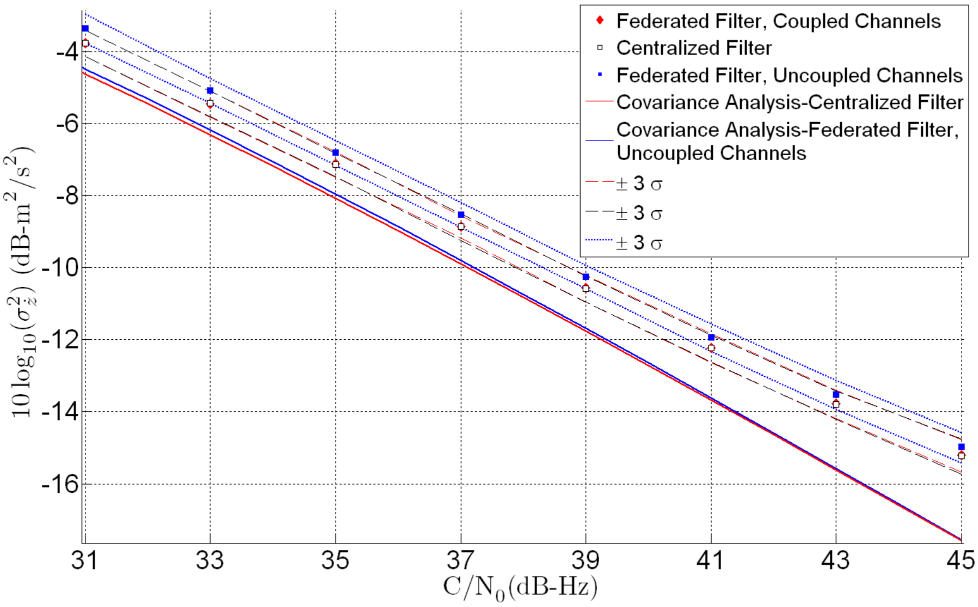

Figure C.58: Monte Carlo Simulation Results for the Z-Axis Velocity with Five Satellites and σ^2=50 m^2/s^2

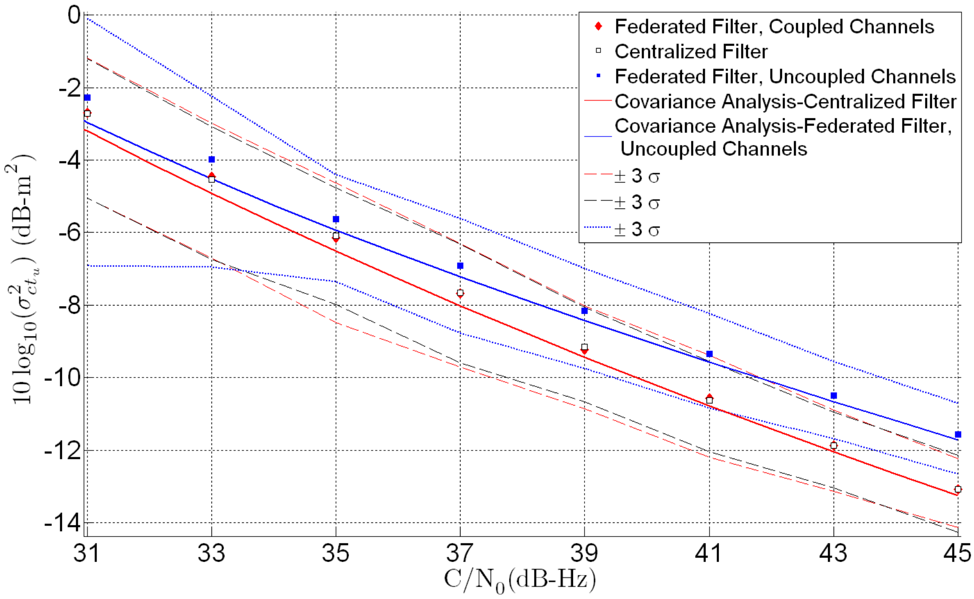

Figure C.59: Monte Carlo Simulation Results for the Clock Bias with Five Satellites and σ^2=50 m^2/s^2

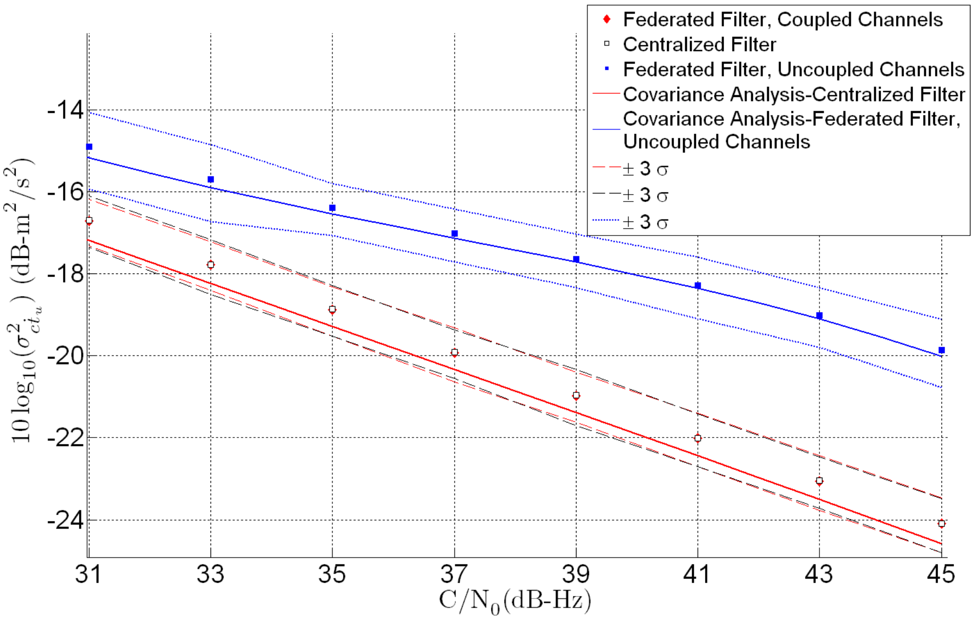

Figure C.60: Monte Carlo Simulation Results for the Clock Drift with Five Satellites and σ^2=50 m^2/s^2

C.5 Additional Plots From Chapter 9 Section 3.1

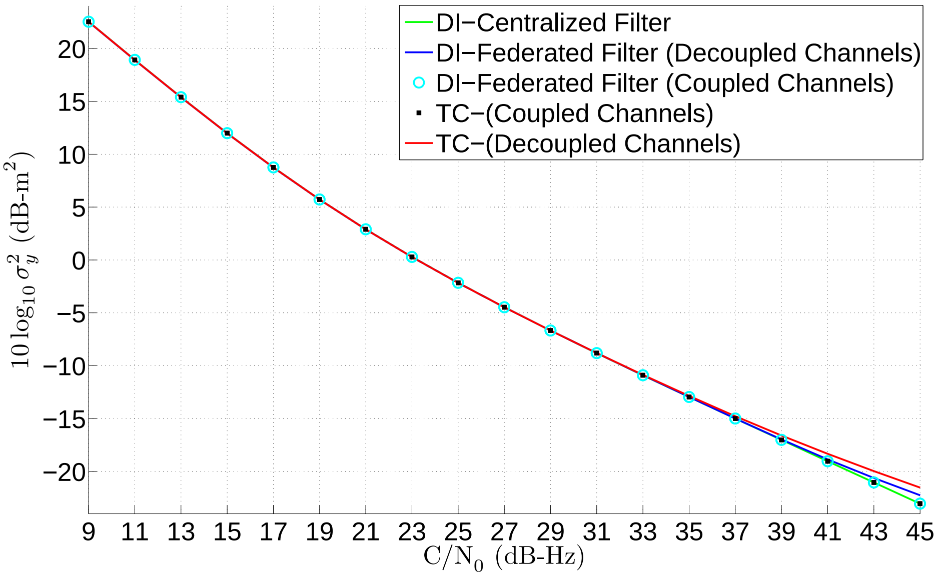

Figure C.61: Covariance Analysis Results for the Y-Axis Position State with Eleven Satellites and Tactical Grade IMU

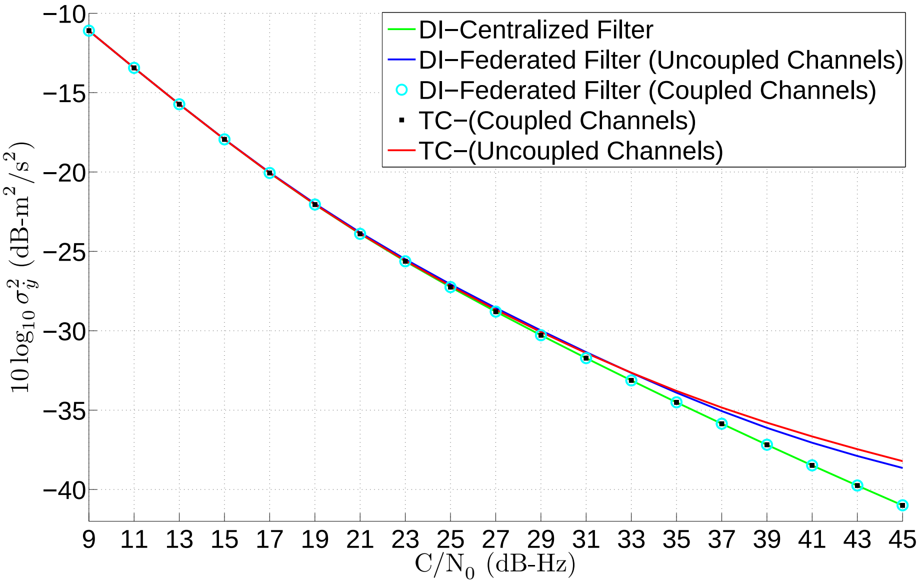

Figure C.62: Covariance Analysis Results for the Y-Axis Velocity State with Eleven Satellites and Tactical Grade IMU

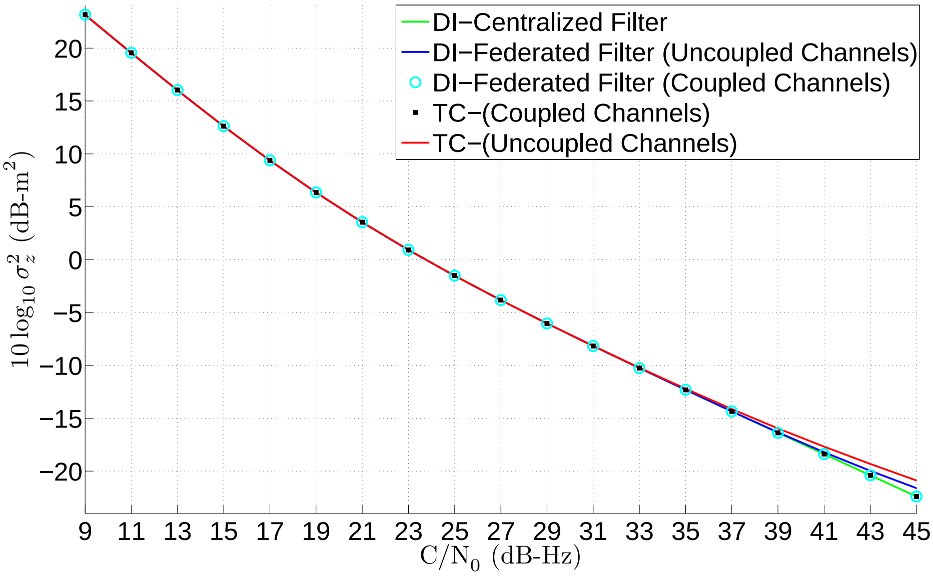

Figure C.63: Covariance Analysis Results for the Z-Axis Position State with Eleven Satellites and Tactical Grade IMU

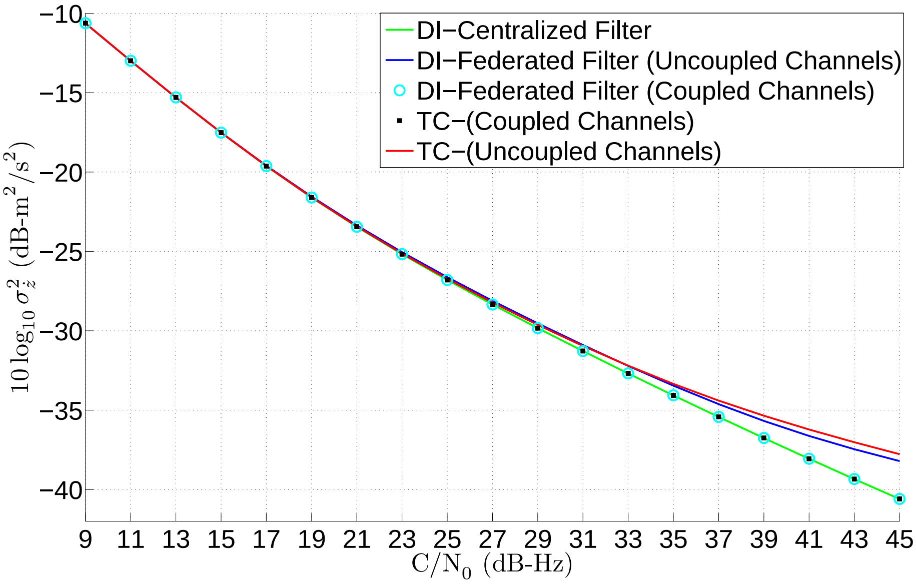

Figure C.64: Covariance Analysis Results for the Z-Axis Velocity State with Eleven Satellites and Tactical Grade IMU

C.6 Additional Plots From Chapter 9 Section 3.2

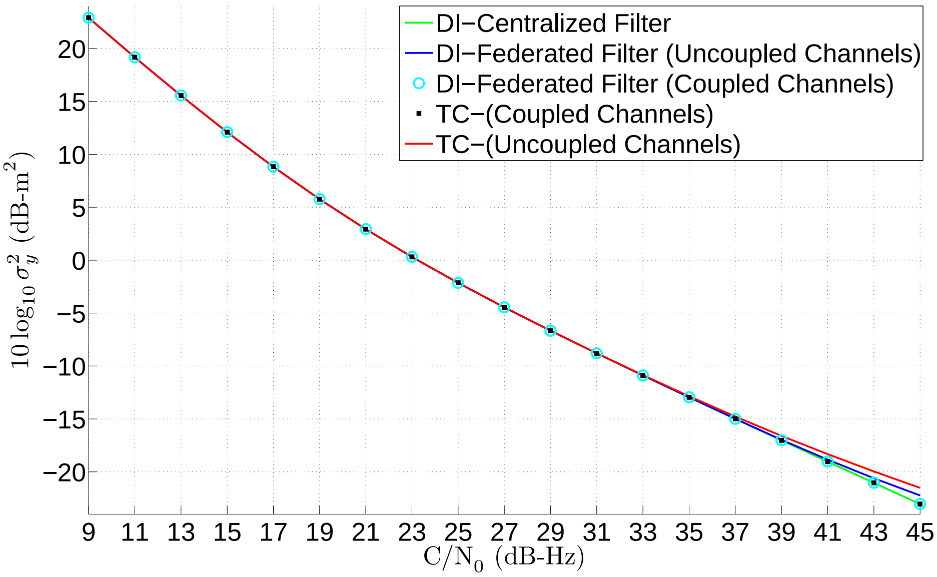

Figure C.65: Covariance Analysis Results for the Y-Axis Position State with Five Satellites and Automotive Grade IMU

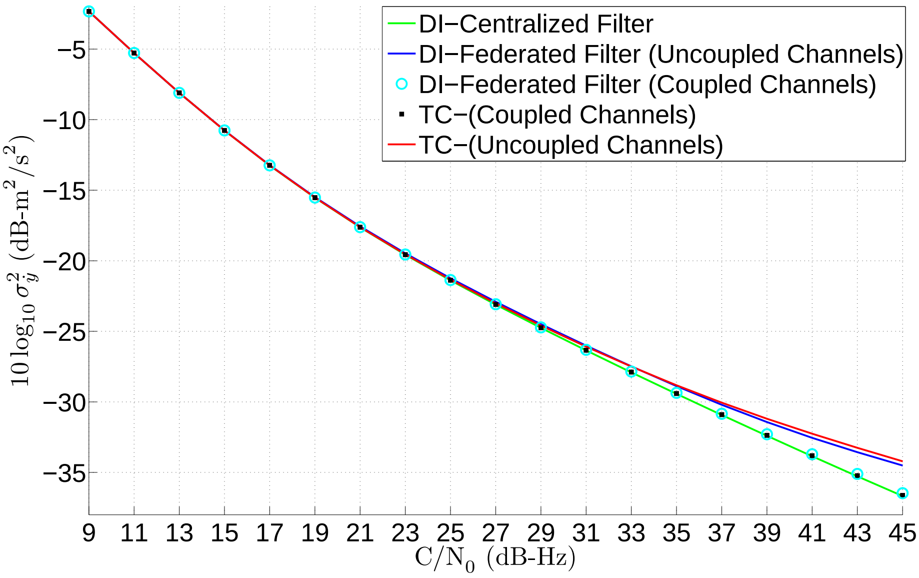

Figure C.66: Covariance Analysis Results for the Y-Axis Velocity State with Five Satellites and Automotive Grade IMU

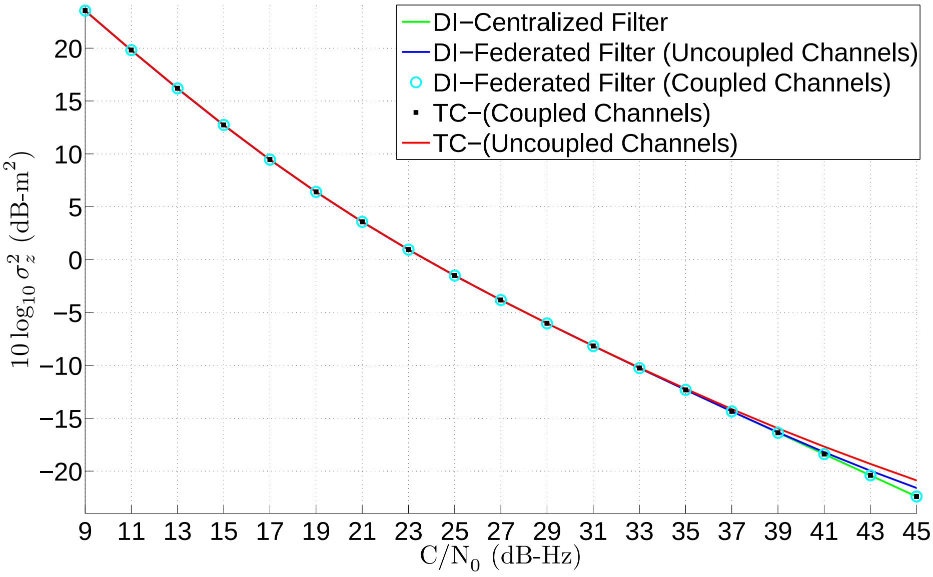

Figure C.67: Covariance Analysis Results for the Z-Axis Position State with Five Satellites and Automotive Grade IMU

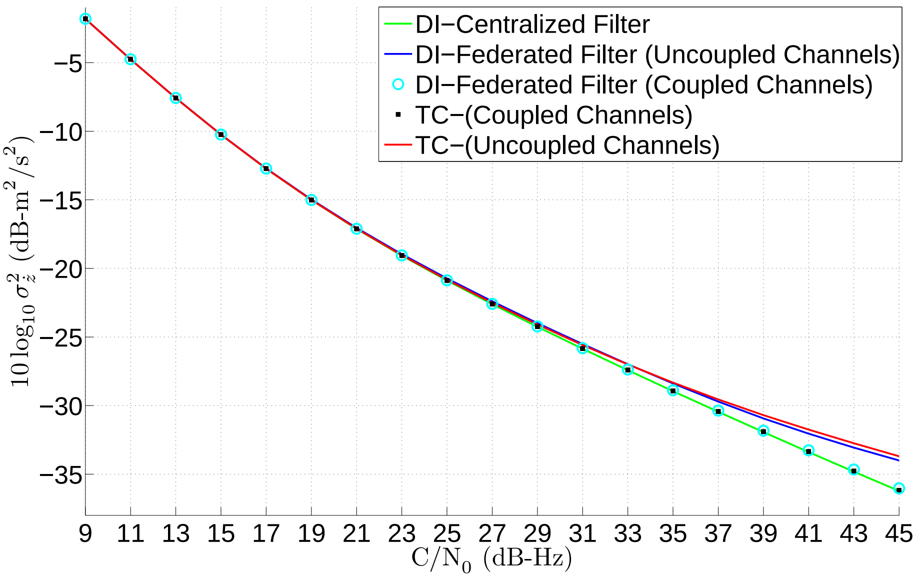

Figure C.68: Covariance Analysis Results for the Z-Axis Velocity State with Five Satellites and Automotive Grade IMU

C.7 Additional Plots From Chapter 9 Section 4.1

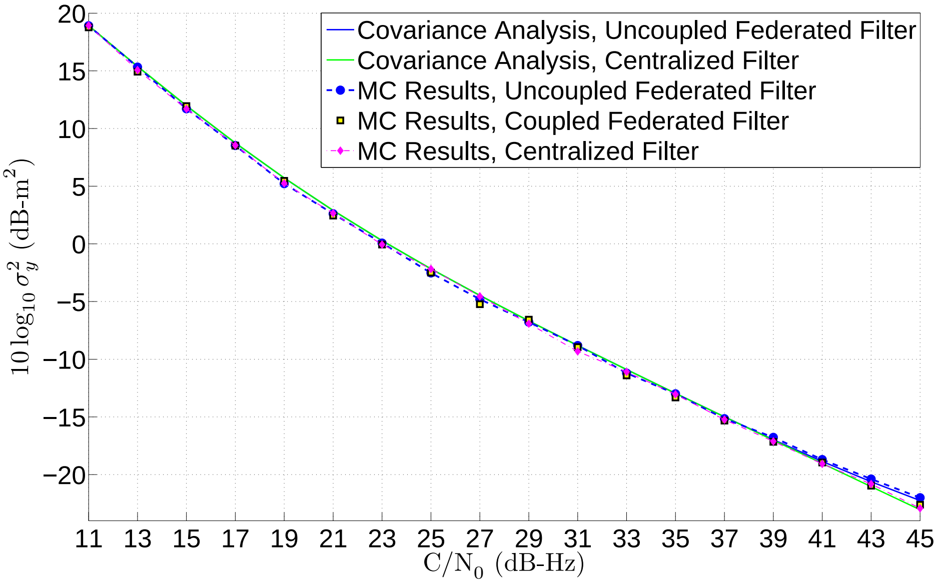

Figure C.69: Monte Carlo Simulation Results for the Y-Axis Position State with Eleven Satellites and Tactical Grade IMU

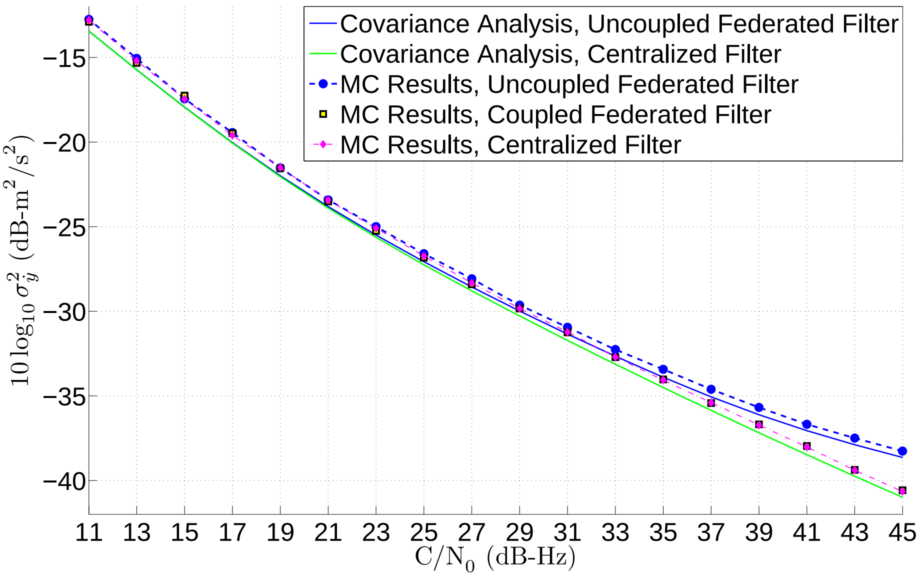

Figure C.70: Monte Carlo Simulation Results for the Y-Axis Velocity State with Eleven Satellites and Tactical Grade IMU

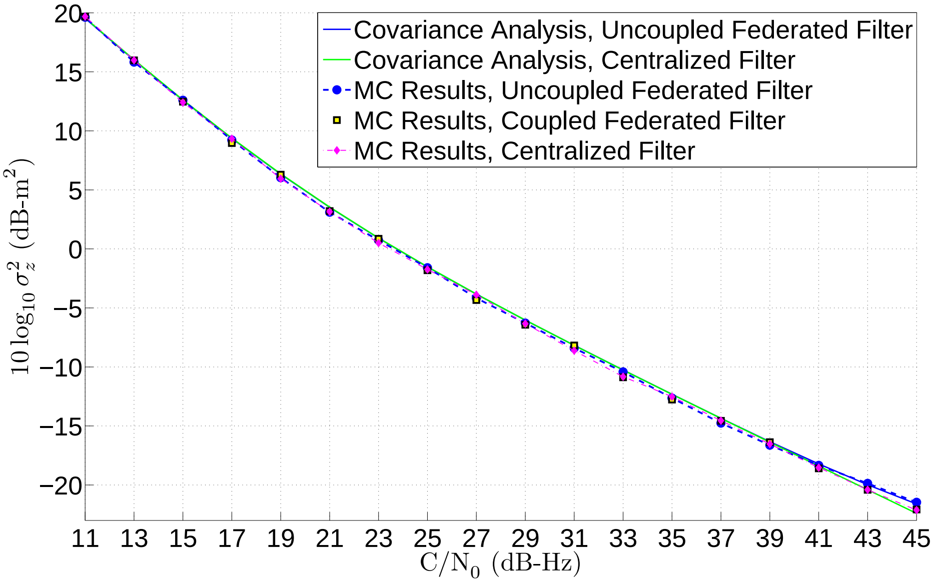

Figure C.71: Monte Carlo Simulation Results for the Z-Axis Position State with Eleven Satellites and Tactical Grade IMU

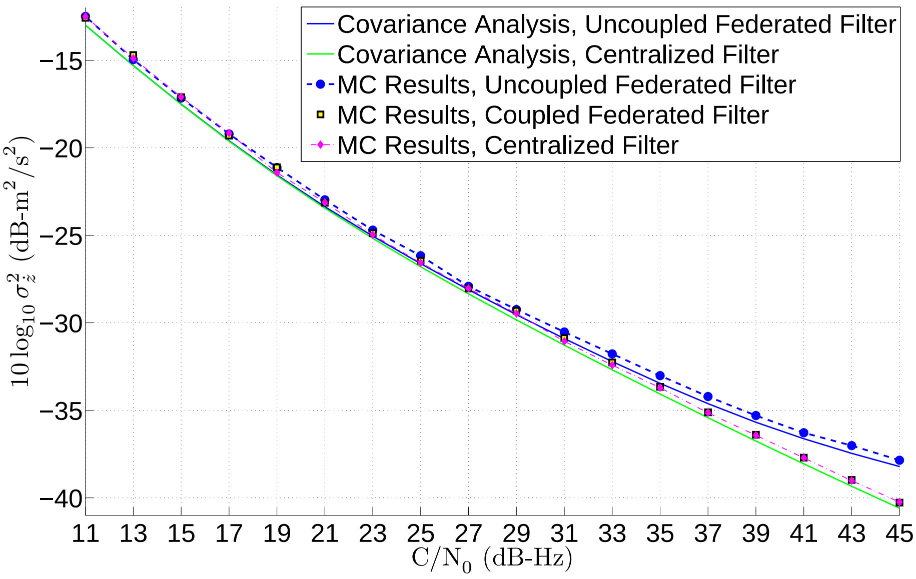

Figure C.72: Monte Carlo Simulation Results for the Z-Axis Velocity State with Eleven Satellites and Tactical Grade IMU

C.8 Additional Plots From Chapter 9 Section 4.2

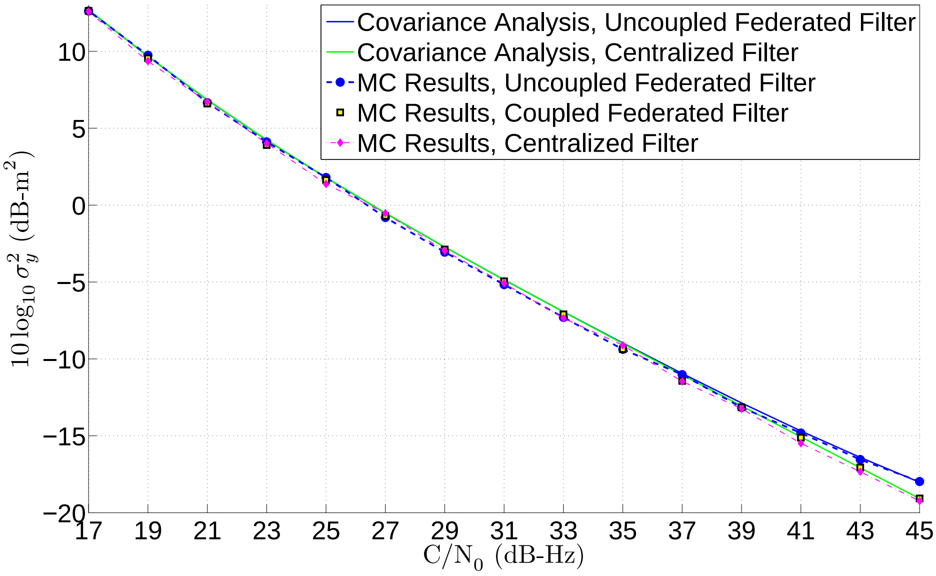

Figure C.73: Monte Carlo Simulation Results for the Y-Axis Position State with Five Satellites and Automotive Grade IMU

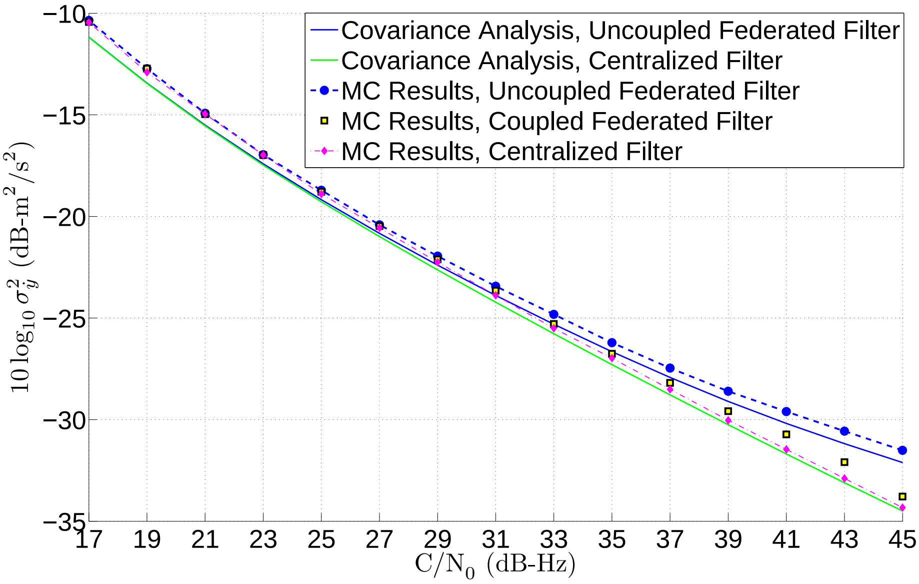

Figure C.74: Monte Carlo Simulation Results for the Y-Axis Velocity State with Five Satellites and Automotive Grade IMU

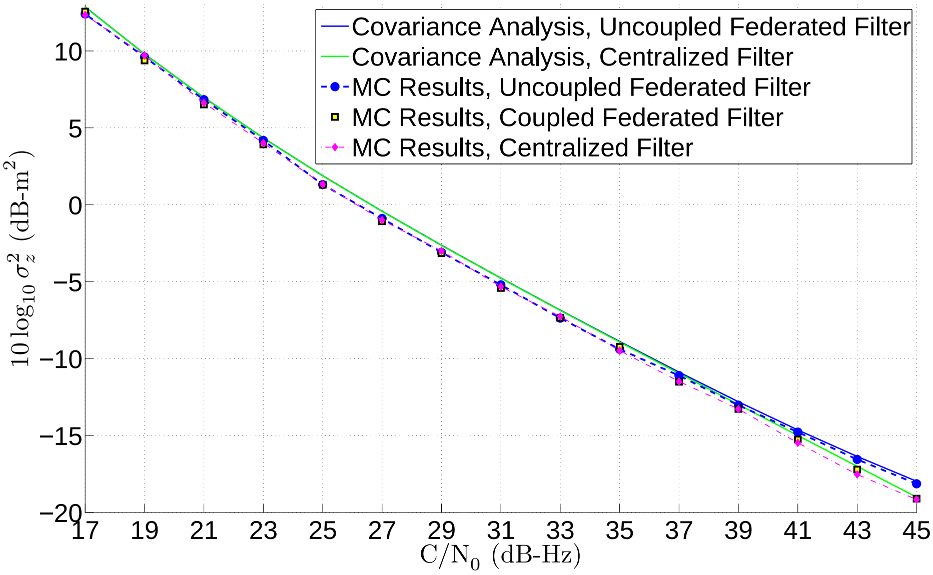

Figure C.75: Monte Carlo Simulation Results for the Z-Axis Position State with Five Satellites and Automotive Grade IMU

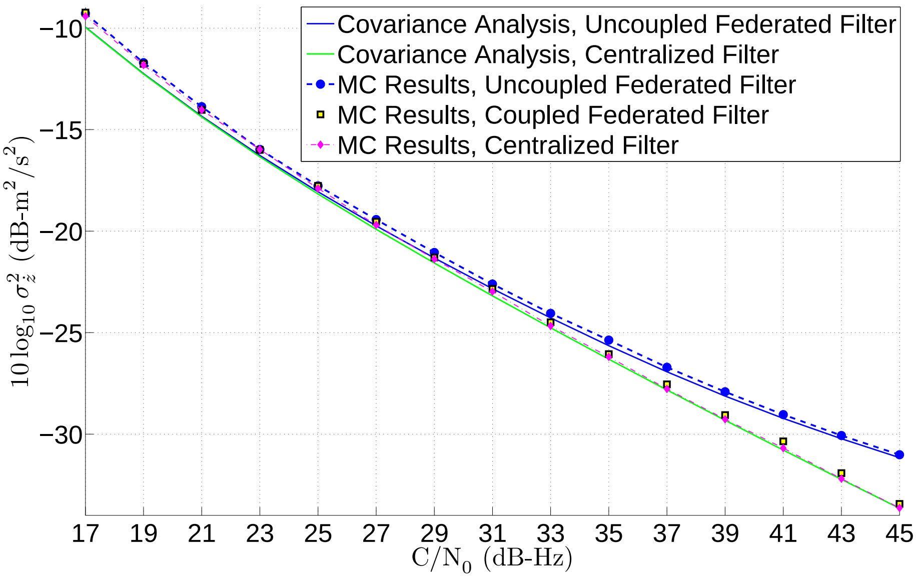

Figure C.76: Monte Carlo Simulation Results for the Z-Axis Velocity State with Five Satellites and Automotive Grade IMU

Appendix D

Error State Recursions

D.1 Introduction

This appendix discusses the error state recursions that are used in Chapters 8 and 9. The error state recursions provide a useful means of modeling how errors in the states of the filter propagate from one period to another. In this appendix, the error state recursions are defined and derived. The motivation for using the error state recursions in the federated filtering architectures is also explained.

D.2 Motivation

The motivation for using the error state recursions comes from the federated filtering architectures in Chapters 8 and 9. The federated filtering architectures all use a combination of a master filter and residual filters. The master filter estimates the full navigation states of the receiver and its state vector is used to predict the received pseudoranges and pseudorange-rates. The residual filters process the pseudorange and pseudorange-rate residuals and estimate the errors in the states of the master filter. Figure D.1 shows a block diagram of one of the federated vector tracking architectures.

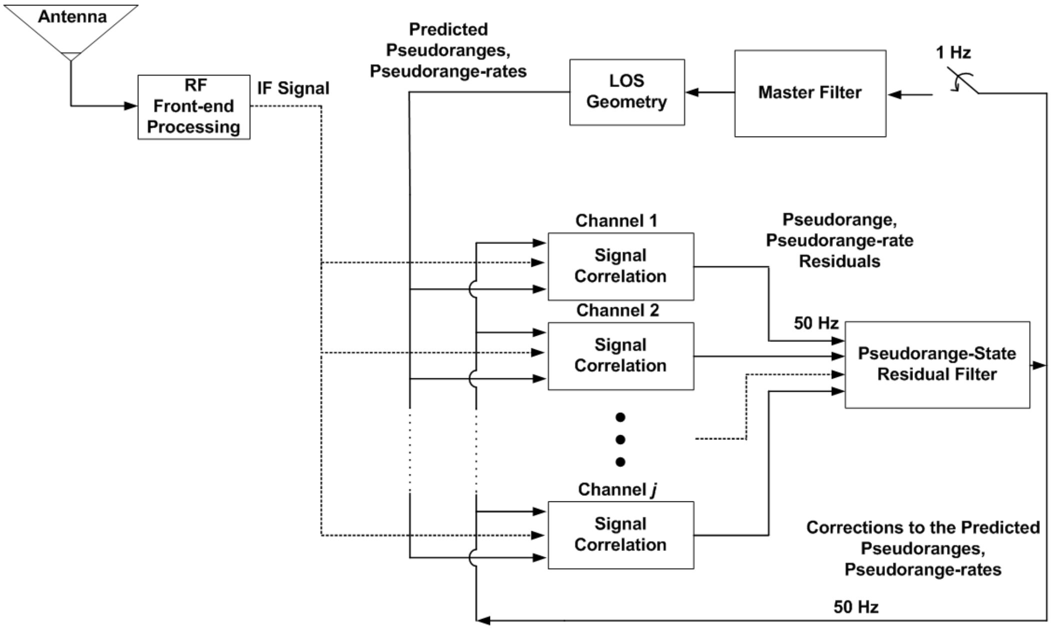
Antenna
RF Front-end Processing
IF Signal
Predicted Pseudoranges, Pseudorange-rates
LOS Geometry
Master Filter
1 Hz
Channel 1
Signal Correlation
Channel 2
Signal Correlation
Channel *j*
Signal Correlation
Pseudorange, Pseudorange-rate Residuals
50 Hz
Pseudorange-State Residual Filter
Corrections to the Predicted Pseudoranges, Pseudorange-rates
50 Hz

Figure D.1: Federated Filter Architecture with Decoupled-Channel Residual Filters

For the federated filtering architectures, the master filter never has access to the raw pseudorange and pseudorange-rate residuals. The estimates of the residual filters are used as measurements by the master filter. The states of the residual filters are cleared after they are used as measurements. The state vector of the residual filter is reset to zeros. This is because the residual filter is estimating the *errors* in the states of the master filter. After the measurement update, the mean of the errors in the states of the master filter is zero (this follows from the Kalman filter being an unbiased estimator). The covariance matrix of the residual filters is also reset. The covariance matrix of the residual filter is reinitialized using the covariance matrix of the master filter after the measurement update. This is done because the errors in the reset residual filter state vector are directly related to the errors in the master filter's states.

An important issue arises when carrying out the measurement update of the master filter. The Kalman filter requires knowledge of the covariance matrix of the errors present in the measurements. It is tempting to simply use the covariance matrix of the residual filter as the measurement noise covariance matrix. However, this is incorrect and can easily lead to filter divergence. This is because the errors in the state vector of the residual filters are correlated with errors in the state vector of the master filter. The Kalman filter equations are derived using the assumption that the measurement noise is white and uncorrelated with the errors in the state vector. These assumptions are violated by using the state covariance matrix of the residual filters as the measurement noise covariance matrix in the master filter update equations.

The problem with using the state covariance matrix of the residual filters as the measurement noise covariance matrix for the master filter can intuitively be understood by considering the case of tracking an extremely weak satellite signal. Assume that the master filter's states are used to predict the pseudorange and pseudorange-rate for the satellite. A residual filter is then used to process the pseudorange and pseudorange-rate residuals that are produced. The residual filter's state vector is initialized to zeros. Its state covariance matrix is initialized using the state covariance matrix of the master filter.

The residual filter processes multiple sets of pseudorange and pseudorange-rate residuals. Since the satellite signal is extremely weak, the residual filter essentially ignores the measurements and propagates its initial state vector forward. The state covariance matrix of the residual filter is also propagated forward accordingly.

After some period of time, the state vector of the residual filter is used by the master filter as a measurement. The residual filter state vector has changed little from the values it was initialized to (because the satellite signal is very weak). The covariance matrix of the residual filter is used as the measurement noise covariance matrix. From the perspective of the master filter, a new set of measurements have been supplied and the noise in the measurements is uncorrelated with its own state vector. In actuality, the errors present in the state vector of the residual filter are nearly identical to the errors present in the master filter. Because the satellite signal is extremely weak, the residual filter ignored the its measurements. Consequently, no new information has been added to the residual filter state vector.

The errors caused by operating the master and residual filters as described in the previous paragraph become particularly noticeable at low C/N_0 ratios. The filter rapidly diverges if the satellite signals start at nominal C/N_0 ratios and are then drastically lowered. The reason the algorithm diverges is that the covariance matrix of the residual filter is continually reinitialized with the covariance matrix of the master filter. Because the signals start at nominal C/N_0 ratios, the values in the master filter's covariance matrix are relatively small. When the signal levels are lowered, the residual filter's covariance matrix is reinitialized with the values from the master filter. At measurement updates of the master filter, the residual filter's covariance matrix is used as the measurement noise covariance matrix. Since the C/N_0 ratios are low, the residual filters basically ignore measurements and the values in it's covariance matrix remain relatively small. From the perspective of the master filter, it is continuing to receive high quality measurements. Consequently, the master filter covariance matrix does not properly increase when the C/N_0 ratios decrease.

The error state recursions are used in Chapters 8 and 9 to avoid the problems associated with the federated filtering architectures. The state vector of the master filter is appended with states that model the errors in the residual filter. At measurement updates of the master filter, the errors in the states of the master filter are observed along with the appended states. The draw back to this method is that no measurement noise is present in the measurements provided to the master filter. Consequently, the measurement noise covariance matrix is all zeros. This can cause numerical problems when calculating the Kalman gain. Equation (D.1) shows the equation typically used to calculate the Kalman gain.

$$K(k) \quad = \quad P^{-}(k)H^{T}\{HP^{-}(k)H^{T}+R\}^{-1} \tag{D.1}$$

When the measurement noise covariance matrix R is absent, the matrix resulting from the product $HP^{-}(k)H^{T}$ may not have full rank and the matrix inside the braces will not be invertible. Techniques exist for dealing with situations where there is no measurement noise and the matrix $HP^{-}(k)H^{T}$ does not have full rank [Maybeck, 1978]. Luckily, an ill conditioned matrix resulting from the product $HP^{-}(k)H^{T}$ was not encounter in the architectures used in Chapters 8 and 9. However, the federated filtering architecture with a coupled channel pseudorange-state residual filter does introduce ill conditioned matrices and numerical problems. The coupled channel pseudorange-state filter was generally avoided for these reasons throughout the dissertation.

The error states appended to the master filter state vector are modeled on the error state recursions. The recursions are derived in the following section.

D.3 Error State Recursions

The state update equation of the Kalman filter can be arranged into the form shown in (D.2).

$$\hat{x}^+(k) = A\hat{x}^+(k-1) + K(k)\epsilon(k) \tag{D.2}$$

In (D.2), the states of the filter are denoted by $\hat{x}$ and the superscripts + and - are used to represent the states of the filter before and after measurement updates, respectively. The discrete-time state transition matrix of the filter is denoted by A and the Kalman gain by K. The residuals of the filter are represented by ϵ and are defined in (D.3).

$$\begin{aligned}
\epsilon(k) &= Hx(k) - H\hat{x}^-(k) + v(k) \\
\epsilon(k) &= Hx(k) - H\{A\hat{x}^+(k)\} + v(k) \\
\epsilon(k) &= Hx(k) - HA\hat{x}^+(k) + v(k)
\end{aligned} \tag{D.3}$$

Equation (D.3) expresses the filter residuals in terms of the true state vector $x(k)$ and the measurement noise vector $v(k)$. Equations (D.2) and (D.3) are combined to form the equations in (D.4).

$$\begin{aligned}
\hat{x}^+(k) &= A\hat{x}^+(k-1) + K(k)\epsilon(k) \\
\hat{x}^+(k) &= A\hat{x}^+(k-1) + K(k)\{Hx(k) - HA\hat{x}^+(k) + v(k)\} \\
x(k) &= Ax(k-1) + Bw(k-1) \\
\hat{x}^+(k) &= A\hat{x}^+(k-1) + K(k)\{HAx(k-1) + HBw(k-1) - HA\hat{x}^+(k) + v(k)\}
\end{aligned} \tag{D.4}$$

$$\begin{aligned}\hat{x}^{+}(k) &= A\hat{x}^{+}(k-1)+K(k)HAx(k-1)+K(k)HBw(k-1)-K(k)HA\hat{x}^{+}(k)...\\ &\quad +K(k)v(k)\end{aligned}$$

In (D.4), the process noise vector is denoted by $w(k)$. The error states of the filter prior to and after the measurement update are defined in (D.5) and are expressed in terms of the estimated states of the filter $\hat{x}^{+}(k)$ and $\hat{x}^{-}(k)$.

$$\begin{aligned}x(k)-\hat{x}^{+}(k) &= \delta x^{+}(k) \qquad \text{(D.5)}\\ x(k)-\hat{x}^{-}(k) &= \delta x^{-}(k)\\ \delta x^{+}(k) &= x(k)-A\hat{x}^{+}(k-1)-K(k)HAx(k-1)-K(k)HBw(k-1)...\\ &\quad +K(k)HA\hat{x}^{+}(k-1)-K(k)v(k)\\ \delta x^{+}(k) &= Ax(k-1)+Bw(k-1)-A\hat{x}^{+}(k-1)-K(k)HAx(k-1)...\\ &\quad -K(k)HBw(k-1)+K(k)HA\hat{x}^{+}(k-1)-K(k)v(k)\end{aligned}$$

Using the definition of the error states $\delta x^{+}(k-1)$, the propagation of the error states after the measurement from time $k-1$ to time k is given in (D.6).

$$\begin{aligned}x(k-1)-\hat{x}^{+}(k-1) &= \delta x^{+}(k-1)\\ \delta x^{+}(k) &= \{A-K(k)HA\}\{x(k-1)-\hat{x}^{+}(k-1)\}...\\ &\quad +\{B-K(k)HB\}w(k-1)-K(k)v(K) \qquad \text{(D.6)}\\ \delta x^{+}(k) &= \{A-K(k)HA\}\delta x^{+}(k-1)+\{B-K(k)HB\}w(k-1)...\\ &\quad -K(k)v(k)\end{aligned}$$

Similarly, using the definition of the error states $\delta x^{-}(k-1)$, the propagation of the error states before the measurement from time $k-1$ to time k is given in (D.7).

$$\begin{aligned} \delta x^{-}(k) &= A\{I - K(k-1)H\}\delta x^{-}(k-1) + w(k-1)... \\ &\quad -AK(k-1)v(k-1) \end{aligned} \tag{D.7}$$

The error state recursions can also be used to derive the autocorrelation sequence of the Kalman filter. The autocorrelation of the error states prior to and after the measurement update in steady state are given in (D.8).

$$\begin{aligned} E\{[x(k+n) - \hat{x}^{-}(k+n)][x(k) - \hat{x}^{-}(k)]^T\} &= [A - AK_{ss}H]^n P_{ss}^{-} \\ E\{[x(k+n) - \hat{x}^{+}(k+n)][x(k) - \hat{x}^{+}(k)]^T\} &= [A - K_{ss}HA]^n P_{ss}^{+} \end{aligned} \tag{D.8}$$

The steady state Kalman gain is denoted by K_{ss} and the steady state covariance matrices are represented by P_{ss}.

The autocorrelation sequence of the Kalman filter can be particularly useful when using suboptimal federated filtering architectures. If the error states are not appended to the master filter's state vector (a likely situation due to processing constraints), enough time can be allowed to lapse between measurement updates of the master filter such that the errors in the measurements appear to be white. The amount of time needed for the errors in the residual filter to become sufficiently decorrelated for different situations can be determined from the autocorrelation sequence.

Appendix E

Inertial Sensor Integration and Error Modeling

E.1 Introduction

This appendix describes how inertial measurement units (IMU's) are integrated with the GPS system in this dissertation. The measurements provided by IMU's and how they correspond to a receiver's navigation states are explained. How the errors in the IMU's measurements are modeled are described. The basic Extended Kalman Filter (EKF) used to blend the GPS and IMU measurements is also explained.

E.2 Strap down Inertial Measurement Units

Strapdown inertial measurements units are used in this dissertation. Strapdown IMU's provide specific force and angular rate measurements of the body upon which they are fixed. Six degree of freedom IMU's are used throughout the dissertation. A six degree of freedom IMU provides measurements from three orthogonally mounted accelerometers and from three orthogonally mounted gyros. The three accelerometers each measure the specific force along their sensitive axes. Similarly, the three gyros each measure the angular rotation rate about the three orthogonal axes [Chatfield, 1997]. The three orthogonal axes of the IMU are referred to here as the body frame coordinate system.

The specific force measurements of the IMU are made in reference to the body frame coordinate system. To process the these measurements, they are first converted into the navigation frame. The navigation frame is the coordinate frame being used to track the receiver's navigation states. The coordinate frame rotation used to go from the body frame to the navigation frame is shown in (E.1) [Wenger, 2002].

$$x^n = C_b^n x^b \tag{E.1}$$

where

$$C_b^n = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \tag{E.2}$$

In (E.1), the matrix C_b^n is referred to as the direction cosine matrix. The vectors x^b and x^n represent the accelerometer measurements in the body and navigation frames, respectively. The angles ϕ, θ, and ψ are Euler angles and are generally referred to as roll, pitch, and yaw when referenced to a local-level frame [Negast, 2009], respectively. The Euler angles define the orientation of the body frame relative to the navigation frame.

Throughout this dissertation, the Earth-Centered Earth-Fixed (ECEF) coordinate frame is used as the navigation frame. The ideal, noise-free specific force measurements of the accelerometers in the navigation frame are related to the receiver's acceleration by (E.3) [Chatfield, 1997] .

$$\ddot{P}^e + 2\varpi_{ie}^e \times \dot{P}^e + \varpi_{ie}^e \times \varpi_{ie}^e \times P^e = g^e + S^e$$

$$\text{where} \tag{E.3}$$

$$\varpi_{ie}^{e} = \begin{bmatrix} 0 \\ 0 \\ \omega_{ie} \end{bmatrix}$$

In (E.3), the position of the receiver is denoted by P^e. The variable ω_{ie} is the rotation rate of the Earth in radians per second and g^e represents the effects of gravity. The variable S^e represents the output of the accelerometer cluster after it has been rotated into the ECEF frame [Chatfield, 1997]. Once expressed in the ECEF frame, the specific force measurements of the IMU are easily integrated to compute the receiver's change in velocity ($\dot{P}^e$) and position (P^e).

In order to perform the coordinate frame rotation in (E.1), the attitude of the IMU relative to the navigation frame must be known. After being initialized with the attitude of the IMU, the change in the IMU's orientation is calculated by using the angular rate measurements of the three gyros. The ideal, noise-free angular rate measurements of the gyro are related to changes in the Euler angles by (E.4) [Wenger, 2002].

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \frac{\sin\phi\sin\theta}{\cos\theta} & \frac{\cos\phi\sin\theta}{\cos\theta} \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{E.4}$$

$$T_\omega = \begin{bmatrix} 1 & \frac{\sin\phi\sin\theta}{\cos\theta} & \frac{\cos\phi\sin\theta}{\cos\theta} \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \tag{E.5}$$

The angular rate measurements of the IMU are denoted by p, q, and r. The change in the IMU's orientation is computed by integrating the equations in (E.4). It is important to note that a singularity occurs in the equations in (E.4) when a pitch angle of $\pi/2$ radians is encountered.

E.3 Inertial Measurement Unit Error Modeling

The formulas relating the IMU measurements to the receiver's acceleration (E.3) and change in orientation (E.4) are ideal. In reality, the IMU's measurements contain a multitude of errors. In this dissertation, deterministic and stochastic IMU errors are modeled. The deterministic errors are scale factor and cross-coupling errors. The stochastic errors are white and time-correlated noise processes.

Scale factor errors are deviations of the input to output gradient of the instrument from unity [Groves, 2008]. Cross-coupling errors are caused by misalignment of the three axes in which the sensors are mounted. The scale factor and cross-coupling errors can be expressed by the following matrices [Groves, 2008]:

$$M = \begin{bmatrix} s_x & m_{xy} & m_{xz} \\ m_{yx} & s_y & m_{yz} \\ m_{zx} & m_{zy} & s_z \end{bmatrix} \tag{E.6}$$

The matrix M is different for the accelerometer and gyro outputs. The corrupted output of the sensor is related to the actual measured quantities by [Groves, 2008]:

$$\tilde{f} = (I_{3\times3} + M) \cdot f \tag{E.7}$$

In (E.7), the true measured value is represented by f and the sensor output is denoted by $\tilde{f}$. The symbol $I_{3\times 3}$ represents a three by three identity matrix. The magnitude of the values for the scale factor and cross coupling terms in (E.6) are determined by the grade of the sensor in question.

Two types of noise process are modeled in the IMU outputs. A white noise process and a $1^{st}-order\ Gauss-Markov$ process are included in the IMU measurements. The white noise process models errors in the measurements that are uncorrelated from one measurement to another. The 1^{st}-order Gauss-Markov process is used to model a slowly wandering error in the IMU output [Wall, 2007]. The total error ϵ at time k is modeled as :

$$\epsilon(k) = b(k) + w(k) \tag{E.8}$$

The white noise component of the error is denoted by w and the Gauss-Markov process in represented by b. The statistical properties of the white noise term are given in (E.9).

$$\begin{aligned} w(k) &\sim N(0, \sigma_w^2) \\ E\{w(i)w(j)\} &= \sigma_w^2 \delta(i-j) \end{aligned} \tag{E.9}$$

The Gauss-Markov sequence is generated from the difference equation in (E.10) [Wall, 2007].

$$b(k) = e^{-\frac{\Delta T}{\tau}} \cdot b(k-1) + v(k) \tag{E.10}$$

$$\begin{aligned} v(k) &\sim N(0,\sigma_v^2) \\ E\{v(i)v(j)\} &= \sigma_v^2\delta(i-j) \end{aligned}$$

In (E.10), the variable τ is referred to as the time constant of the process. The values for σ_w^2, σ_v^2, and τ are dependent on the grade of IMU that is used.

E.4 Extended Kalman Filter for Tight and Deeply Integrated Architectures

An extended Kalman filter is used to estimate the navigation states of the receiver in the GPS/INS systems studied in this dissertation. The states of the basic filter are shown in (E.11).

$$\begin{bmatrix} P_{3\times1} \\ V_{3\times1} \\ A_{3\times1} \\ B^{accel}_{3\times1} \\ B^{gyro}_{3\times1} \\ B^{clk}_{2\times1} \end{bmatrix} = \begin{bmatrix} \text{Position States} \\ \text{Velocity States} \\ \text{Attitude States} \\ \text{Accelerometer Bias States} \\ \text{Gyro Bias States} \\ \text{Clock Bias and Drift States} \end{bmatrix} \tag{E.11}$$

In general, the dynamics of the states are nonlinear. The state propagation equations for the position, velocity, attitude, bias, and clock states are given below.

$$x(k+1) = f(x(k),u(k),w(k))$$

$$\begin{aligned}
P(k+1) &= P(k) + \Delta T \cdot V(k) + \frac{\Delta T^2}{2} \cdot C_b^n \cdot \tilde{f}^{accel}(k) - \frac{\Delta T^2}{2} \cdot C_b^n \cdot B^{accel}(k) ... \\
&\quad + \frac{\Delta T^2}{2} \cdot C_b^n \cdot w^{accel}(k) + \frac{\Delta T^2}{2} \cdot G(P(k)) - \Delta T^2 \cdot \varpi_{ie}^e \times V(k) ... \\
&\quad - \frac{\Delta T^2}{2} \cdot \varpi_{ie}^e \times \varpi_{ie}^e \times P(k) \\
V(k+1) &= V(k) + \Delta T \cdot C_b^n \cdot \tilde{f}^{accel}(k) - \Delta T \cdot C_b^n \cdot B^{accel} + \Delta T \cdot C_b^n \cdot w^{accel}(k) ... \\
&\quad - \Delta T \cdot G(P(k)) - 2\Delta T \cdot \varpi_{ie}^e \times V(k) - \Delta T \cdot \varpi_{ie}^e \times \varpi_{ie}^e \times P(k) \\
A(k+1) &= A(k) + \Delta T \cdot T_\omega \cdot \tilde{f}^{gyro}(k) + \Delta T \cdot T_\omega \cdot w^{gyro}(k) - \Delta T \cdot T_\omega \cdot B^{gyro}(k) \\
B^{accel}(k+1) &= e^{-\frac{\Delta T}{\tau_{accel}}} \cdot I_{3\times 3} \cdot B^{accel}(k) + v^{accel}(k) \\
B^{gyro}(k+1) &= e^{-\frac{\Delta T}{\tau_{gyro}}} \cdot I_{3\times 3} \cdot B^{gyro}(k) + v^{gyro}(k) \\
B^{clk}(k+1) &= \kappa \cdot B^{clk}(k) + v_{2\times 1}^{clock}(k) \\
\kappa &= \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}
\end{aligned} \tag{E.12}$$

In (E.12), the function $G(P(k))$ is an Earth gravity model and is a function of the receiver's estimated position. The time between state updates is represented by ΔT. The symbol ϖ_{ie}^e denotes the rotation rate of the Earth. The measurements of the accelerometer and gyro cluster are denoted by the symbols $\tilde{f}^{accel}$ and $\tilde{f}^{gyro}$, respectively. The variables τ_{gyro} and τ_{accel} are the time constants associated with the gyro and accelerometer biases, respectively. The symbol $I_{3\times 3}$ represents a 3 by 3 identity matrix. The wideband noise present in the accelerometer and gyro measurements is denoted by w^{accel} and w^{gyro}, respectively. Similarly, the noise driving the accelerometer and gyro bias states is represented by v^{accel} and v^{gyro}. The process noise driving the clock states is denoted by v^{clock}

An error state extended Kalman filter is used to estimate errors in the propagated states. The true states of the receiver $x(k)$, the estimates of the filter $\hat{x}(k)$, and the errors in the states of the filter $\delta x(k)$ at time k are related to each other through (E.13).

$$x(k) \equiv \hat{x}(k) + \delta x(k) \tag{E.13}$$

In general, the states of the system at the next time $k+1$ are equal to a nonlinear function of the states at time k, input vector $u(k)$, and noise vector $w(k)$ (E.14).

$$x(k+1) = f(x(k), u(k), w(k)) \tag{E.14}$$

For the specific system in question here, the input vector $u(k)$ represents the uncorrupted measurements from the IMU and the noise vector $w(k)$ models the noise processes in the IMU and receiver clock. Equation (E.14) is linearized about the current estimates of the filter to produce (E.15).

$$\begin{aligned} x(k+1) &= f(\hat{x}(k) + \delta x(k), u(k), w(k)) && \text{(E.15)} \\ x(k+1) &\approx f(\hat{x}(k), u(k), 0) + A \cdot \delta x(k) + B \cdot w(k) \end{aligned}$$

where

$$\begin{aligned} A &= \left.\frac{\partial f(\cdot)}{\partial x}\right|_{x=\hat{x}(k)} \\ B &= \left.\frac{\partial f(\cdot)}{\partial w}\right|_{x=\hat{x}(k)} \end{aligned}$$

The vectors $\delta x(k)$ and $w(k)$ are random variables and are assumed to be zero mean. The state vector of the system is propagated forward in time by using (E.16).

$$\hat{x}(k+1) = f(\hat{x}(k), u(k), 0) \tag{E.16}$$

The errors in the propagated state vector at time $(k+1)$ are therefore given in (E.17).

$$\begin{aligned} \delta x(k+1) &\equiv x(k+1) - \hat{x}(k+1) \\ \delta x(k+1) &\equiv x(k+1) - f(\hat{x}(k), u(k), 0) \end{aligned} \tag{E.17}$$

Using the linearization in (E.15), the mean and covariance of the state vector errors are shown in (E.18).

$$\begin{aligned} E\{\delta x(k+1)\} &\approx 0_{N\times 1} \\ E\{\delta x(k+1)\delta x^T(k+1)\} &\approx AP(k)A^T + BQ(k)B^T \end{aligned} \tag{E.18}$$

In (E.18), $P(k)$ is the covariance matrix of the state errors at time k and $Q(k)$ is the covariance matrix of the process noise vector $w(k)$.

In general, the measurements available to the filter are nonlinear functions of the states. The measurements are subtracted by the nonlinear measurement function evaluated at the propagated state vector $\hat{x}(k)$. The difference of the two quantities is approximately equal to the state error vector multiplied by the partial of the measurement function with respect to the state vector (E.19).

$$\begin{aligned} y(k) &= g(x(k)) + v(k) \\ y(k) - g(\hat{x}(k)) &\approx H \cdot \delta x(k) + v(k) \\ \text{where} & \\ H &= \left.\frac{\partial g(\cdot)}{\partial x}\right|_{x=\hat{x}(k)} \end{aligned} \tag{E.19}$$

In (E.19), the measurement noise is denoted as $v(k)$. The state vector errors are then estimated using the standard Kalman filter measurement update equations by using the matrix H as the observation matrix. The estimated state vector errors are then added to the propagated states $\hat{x}(k)$. The updated state vector is then propagated forward in time to the next measurement update time.