

BOUNDING ESTIMATION INTEGRITY RISK FOR LINEAR SYSTEMS  
WITH STRUCTURED STOCHASTIC MODELING UNCERTAINTY

BY

STEVEN EDWARD LANGEL

Submitted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy in Mechanical and Aerospace Engineering  
in the Graduate College of the  
Illinois Institute of Technology

Approved \_\_\_\_\_  
Adviser

Chicago, Illinois  
May 2014

© Copyright by  
STEVEN LANGEL  
2014

## ACKNOWLEDGEMENT

I would like to thank my doctoral advisor Dr. Boris Pervan for providing me the opportunity to complete this dissertation. His invaluable guidance and belief in my abilities gave me the confidence and desire to reach this milestone in my life. I would also like to extend my deepest gratitude to my committee, comprised of Drs. Hassan Nagib, Matthew Spenko, Yongyi Yang and Jason Rife, for their incredible patience in reviewing my dissertation. Special thanks are also due to my research sponsors at NAVAIR and Dynamic Analytical Solutions, in particular Frank Allen and Marie Lage. Without their endless support, this research would never have been possible.

To my research partner Dr. Samer Khanafseh, I am forever grateful. He was my primary mentor when I entered graduate school at IIT and provided the foundation on which this entire work is built. His enthusiasm and dedication to research has been a source of inspiration throughout this journey. I would also like to thank Drs. Fang-Cheng Chan and Mathieu Joerger for their precious time and valuable insights into my work, and my fellow colleagues Jason Neale, Jing Jing, Stefan Stevanovic, Aniruddha Katre and Naeem Roshan who have made my tenure as a PhD candidate as memorable as possible.

Finally, I would like to extend my most heartfelt appreciation to my family, especially my parents Edward and Kimberly Langel, who have fostered my interests in science, engineering and mathematics since childhood. It is because of their continual love and support that I have finally reached the pinnacle of academic achievement. I dedicate this dissertation to them.

## TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENT .....	iii
LIST OF TABLES .....	vi
LIST OF FIGURES .....	vii
LIST OF ABBREVIATIONS.....	ix
ABSTRACT.....	x
 CHAPTER	
1. INTRODUCTION .....	1
1.1 Kalman Filtering with Colored Noise.....	1
1.2 Batch Weighted Least Squares with Colored Noise .....	2
1.3 Mixed Real/Integer Estimation .....	3
1.4 Integrity Risk Assessment with Modeling Uncertainty .....	5
1.5 Previous and Related Work .....	6
1.6 Dissertation Contributions .....	9
2. INTEGRITY RISK BOUNDING IN KALMAN FILTERING .....	11
2.1 Measurement and State Dynamic Models .....	11
2.2 Quantifying Integrity Risk in Kalman Filtering .....	13
2.3 Estimate Error Vector Difference Equations .....	18
2.4 Generalized Covariance Matrix for the Kalman Filter .....	21
2.5 Estimate Error Variance as a Function of Autocorrelation.....	25
2.6 Variance Bounding with Autocorrelation Uncertainty .....	28
2.7 One-Dimensional Estimation Application.....	29
3. EXTENSIONS TO BATCH ESTIMATION.....	42
3.1 Linearized Measurement and State Dynamic Models .....	42
3.2 Batch Measurement Model and WLS Estimator .....	45
3.3 Generalized Covariance Matrix for Batch WLS.....	48
3.4 Estimate Error Variance as a Function of Autocorrelation.....	50
3.5 Variance Bounding with Autocorrelation Uncertainty .....	51
3.6 Two-Dimensional Estimation Application .....	52

4. BOUND REDUCTION VIA SEMI-DEFINITE OPTIMIZATION.....	60
4.1 Problem Statement.....	60
4.2 Conversion to Standard Primal Form.....	62
4.3 Primal-Dual Optimality and Interior Point Algorithms.....	64
4.4 Comparison of Bounding Methods.....	65
5. MIXED REAL/INTEGER INTEGRITY RISK BOUNDING.....	71
5.1 Integer Bootstrap Estimator.....	71
5.2 Integrity Risk in Mixed Real/Integer Estimation.....	74
5.3 Natural Formulation of Mixed Integer Integrity Risk Bound.....	77
5.4 Bounding Integrity Risk over a Hyper-Rectangular Region.....	80
5.5 Static Baseline Estimation Application.....	84
5.6 Bounding Integrity Risk over Polytopic Region.....	93
5.7 Comparison of Rectangular and Polytopic Bounding Methods.....	98
6. CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK.....	99
6.1 Summary of Contributions.....	99
6.2 Recommended Topics for Future Research.....	101

## APPENDIX

A. GENERAL SOLUTION TO LINEAR SYSTEM OF DIFFERENCE EQUATIONS.....	103
B. DISCRETE-TIME AUTOCORRELATION FUNCTION OF INTEGRATED RANDOM PROCESS.....	108
C. CONTINUOUS-TIME DYNAMIC MODEL FOR TWO-DIMENSIONAL ESTIMATION PROBLEM.....	112
D. DISCRETIZATION AND LINEARIZATION FOR TWO-DIMENSIONAL ESTIMATION PROBLEM.....	116
E. CONDITIONS FOR PRIMAL-DUAL OPTIMALITY.....	123
F. INFEASIBLE INTERIOR POINT ALGORITHM FOR SEMI-DEFINITE OPTIMIZATION.....	125
G. PROOF THAT GLOBAL MINIMUM RESIDES AT AN ENDPOINT.....	131
H. DETERMINATION OF POLYTOPIC FEASIBLE REGION.....	136
BIBLIOGRAPHY.....	140

## LIST OF TABLES

Table	Page
2.1 ACF Bound Parameters Used for 1D Estimation Problem.....	35
3.1 Accelerometer and Gyroscope ACF Bounds Used for 2D Application.....	55
3.2 VOR and DME ACF Bounds Used for 2D Estimation Application.....	56
3.3 Eigenvalues of ACFs Leading to Radial Position State Variance Bound .....	59
4.1 Symmetric Toeplitz Constraints for $4 \times 4$ Matrix .....	66
4.2 Eigenvalues of SDP ACFs Yielding Radial Position State Variance Bound..	70
5.1 Parameters Used for Convexity Analysis.....	79
5.2 SD Carrier Phase ACF Bounds Used for Baseline Estimation Problem.....	87
5.3 Simulation Parameters for Baseline Estimation Problem .....	87
5.4 Bounding Conditional Variances after One Bootstrapping Phase .....	88
5.5 Bounding Conditional Variances after Two Bootstrapping Phases .....	90
H.1 Number of Inequality Constraints Introduced from Column Combinations...	139

## LIST OF FIGURES

Figure	Page
2.1 Problem Setup for 1D Estimation Application .....	29
2.2 Ranging Beacon Autocorrelation Bounding Functions.....	37
2.3 Accelerometer Autocorrelation Bounding Functions.....	38
2.4 Ranging Beacon ACF Resulting in 1D Position State Variance Bound .....	38
2.5 Accelerometer ACF Resulting in 1D Position State Variance Bound .....	39
2.6 Ranging Beacon ACF Resulting in 1D Velocity State Variance Bound .....	40
2.7 Accelerometer ACF Resulting in 1D Velocity State Variance Bound .....	40
3.1 Problem Setup for 2D Estimation Application.....	52
3.2 Vehicle Body Frame and Heading Angle.....	53
3.3 DME ACF Resulting in Position State Variance Bound.....	57
3.4 VOR ACF Resulting in Position State Variance Bound .....	57
3.5 Accelerometer 1 ACF Resulting in Position State Variance Bound .....	57
3.6 Accelerometer 2 ACF Resulting in Position State Variance Bound .....	58
3.7 Gyroscope ACF Resulting in Position State Variance Bound .....	58
4.1 Percent Reduction in Variance Bound Achieved Using SDP Approach .....	67
4.2 DME ACF via SDP Leading to Position State Variance Bound.....	68
4.3 VOR ACF via SDP Leading to Position State Variance Bound .....	68
4.4 Accelerometer 1 ACF via SDP Leading to Position State Variance Bound	69
4.5 Accelerometer 2 ACF via SDP Leading to Position State Variance Bound	69
4.6 Gyroscope ACF via SDP Leading to Position State Variance Bound .....	69

5.1	Cost Function Behavior after Affine Transformation .....	79
5.2	Problem Setup for Static Baseline Estimation Application.....	83
5.3	Bound Reduction via Large Subdivisions for One Bootstrapping Phase.....	89
5.4	Bound Reduction via Small Subdivisions for One Bootstrapping Phase.....	89
5.5	Bound Reduction via Large Subdivisions for Two-Phase Bootstrapping....	90
5.6	Bound Reduction via Small Subdivisions for Two-Phase Bootstrapping ...	91
5.7	Example of Subdivided Polytopic Feasible Region .....	93
5.8	Polytopic and Rectangular Feasible Regions after One Bootstrap Phase ....	96
5.9	Polytopic Bound Performance for One Bootstrapping Phase .....	97
5.10	Polytopic Bound Performance for Two-Phase Bootstrapping .....	97
C.1	Definition of Inertial and Polar Coordinate Systems .....	113

## LIST OF ABBREVIATIONS

Abbreviation	Definition
ACF	Autocorrelation Function
ARE	Algebraic Riccati Equation
CDF	Cumulative Distribution Function
DD	Double Difference
DME	Distance Measuring Equipment
GLS	Generalized Least Squares
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
LMI	Linear Matrix Inequality
LP	Linear Programming
MAP	Maximum a Posteriori
PDF	Probability Density Function
PSD	Power Spectral Density
SD	Single Difference
SDP	Semi-Definite Programming
VHF	Very High Frequency
VOR	VHF Omnidirectional Ranging
WLS	Weighted Least Squares

## ABSTRACT

Safety critical estimation applications require quantification of integrity risk, which is the probability of the state estimate error exceeding predefined bounds of acceptability. Integrity risk can only be evaluated when the state estimate error probability density function is precisely known, necessitating stochastic models that exactly describe measurement noise and disturbance inputs. Uncertainty in these models directly results in inaccurate assessments of integrity risk. This dissertation develops the first implementable methods to upper bound integrity risk when the autocorrelation functions of stochastic inputs reside between upper and lower bounding functions.

The first part of this work considers real-valued estimation applications that use the Kalman filter or batch weighted least squares estimator. Explicit relations are developed between the estimate error variance and autocorrelation functions using a new generalized covariance matrix derived in this dissertation. From these expressions, two methods are provided to upper bound integrity risk. The first method enables fast computation of a conservative bound, and the second method produces the minimum upper bound via semi-definite optimization.

Mixed real/integer estimation applications utilizing integer bootstrapping are the focus of the second part of this work. The integrity risk bound is formally defined as the global solution to a non-convex optimization problem over a polytope. Determination of the polytopic region is difficult, and two bounding approaches are initially developed for a circumscribing hyper-rectangular feasible region. Using an innovative method to define the polytope together with linear programming, a third method is derived to upper bound integrity risk over the true polytopic feasible region.

## CHAPTER 1

### INTRODUCTION

State estimation is essential to the solution of numerous engineering problems, including nuclear reactor control, aircraft navigation, and orbit determination of interplanetary spacecraft and earth-orbiting satellites. It also lies at the heart of extremely challenging military operations like autonomous shipboard landing and airborne refueling of unmanned aerial vehicles. For safety critical estimation applications, it is necessary to quantify *integrity risk*, which is the probability that the state estimate error exceeds a given acceptable bound. Accurate integrity risk assessment relies on precise knowledge of the estimate error probability density function, which can only be determined when exact statistical descriptions of measurement noise and disturbance inputs are available. Inevitably, there will be uncertainty in characterizing these random processes. This dissertation provides the first known methods to quantify the impact of stochastic model uncertainty on integrity risk for the Kalman filter, batch WLS estimator and integer bootstrap estimator.

#### **1.1 Kalman Filtering with Colored Noise**

For estimation applications where the state vector evolves according to a dynamic model, measurements are typically processed (or filtered) over time. Sequential algorithms are primarily used in linear estimation problems because of their obvious computational advantage over batch estimation techniques. The Kalman filter will be the only sequential estimator considered in this dissertation due to its widespread use in engineering. A fundamental assumption used in its derivation is that the measurement

noise and disturbance inputs are white noise processes. While this assumption may be valid in certain situations, it will not be true in general. GPS multipath error and inertial sensor bias errors are two examples of random processes that exhibit significant time correlation [Yan04] [Wal06]. Fortunately, several variants of the Kalman filter have already been developed to accommodate colored noise.

State augmentation is one well-known approach that accounts for time correlation by using additional filter states [And79]. The resulting increase in state vector dimension can sometimes have adverse practical ramifications, and this has resulted in alternative solutions that do not require additional states. Two examples of such alternatives are the measurement differencing filter [Bry68] and the Schmidt-Kalman filter [Sch66]. Because the choice of which method to use is mainly a practical consideration, this dissertation will use the standard state augmentation paradigm to account for time correlation. The Kalman filter and its variants have been successfully utilized to solve a wide variety of estimation problems. However, there are situations where it is advantageous or even necessary to process sequential measurements simultaneously in a batch estimator.

## **1.2 Batch Weighted Least Squares with Colored Noise**

In non-linear estimation problems like radar tracking, batch estimators have been shown to possess better convergence properties than sequential algorithms [Sat01]. When the measurement noise and disturbance inputs can only be accurately modeled over a finite time interval, or if they cannot be described using a finite state model (e.g., flicker noise in quartz oscillators), state vector estimation can only be properly implemented using batch algorithms [Cha10] [Joe10]. Weighted Least Squares (WLS) is most

commonly used in the situations described above, and it will be the only batch estimator considered in this research.

For nonlinear estimation problems, WLS is typically implemented by iteratively re-linearizing the measurement model until the state estimate converges. The weighting matrix is determined from the autocorrelation functions (ACFs) of measurement noise and disturbance inputs and in general will be fully populated. It is not necessary to specify a linear, finite state model for correlated noise in WLS estimation. Therefore, WLS can solve a larger class of estimation problems compared to the Kalman filter. However, WLS is considerably more computationally demanding, and therefore should only be used in applications that cannot be treated using a Kalman filter.

Both the WLS estimator and the Kalman filter are real estimators. That is, they produce a real-valued estimate of the state vector. In a select number of estimation applications, a complete description of the system state requires specification of both real and integer variables. For example, tracking a maneuvering aircraft involves estimating position and velocity as well as the current flight mode, which is specified by an integer value. Enforcing the integer constraint can significantly improve estimation quality.

### **1.3 Mixed Real/Integer Estimation**

Hybrid estimation applications like aircraft and target tracking have measurement and state dynamic models that are defined separately for each mode, or integer state [Boe02] [Hwa06]. In practice, the integer state is often modeled as a finite state Markov chain to account for the fact that mode transitions are typically unknown *a priori* [Cos05]. For a relatively small number of modes, a bank of Kalman filters (one designed

for each mode) can be used to estimate the mixed real/integer state vector [Bar01] [May82]. The real-valued state estimate is a weighted average of the Kalman filter estimates and the integer estimate is defined as the most probable mode. In [Zym08], a maximum *a posteriori* (MAP) estimator is developed for a special class of hybrid systems based on the solution to an associated mixed real/integer quadratic programming problem.

Mixed real/integer estimation also appears in differential navigation applications that utilize carrier phase signals from Global Navigation Satellite Systems (GNSS). The number of whole carrier cycles between the vehicle and satellite is an unknown integer, commonly referred to as the *cycle ambiguity* [Mis01]. Real-time navigation applications utilizing carrier phase measurements require fast, efficient algorithms for estimation of cycle ambiguities, which has led to the development of numerous mixed real/integer estimators [Teu01a] [Teu03] [Teu10] [Ver05]. The majority of these algorithms convert appropriate elements of the real-valued estimate vector (obtained from either a Kalman filter or WLS estimator) to an integer value.

Integer least squares is one technique that minimizes a least squares cost function to determine the integer estimate [Teu95]. Integer bootstrapping is another approach that combines conditional least squares estimation with rounding to sequentially estimate each component of the integer vector [Teu01b]. The sequential nature of the bootstrapping method allows one to estimate only a partial subset of the integer vector. Furthermore, it is possible to directly quantify the probability of successful integer estimation, which is essential for integrity risk computation. Because of these features, the bootstrap estimator is the only mixed real/integer estimator considered in this research.

#### 1.4 Integrity Risk Assessment with Modeling Uncertainty

Quantifying integrity risk requires specification of the state estimate error's probability density function (PDF). Throughout this dissertation, it will be assumed that the measurement noise and disturbance inputs can be accurately described as zero-mean Gaussian random processes. Under this assumption, the estimate error vectors associated with the Kalman filter and WLS estimator are also zero-mean Gaussian because both estimators are linear and unbiased [Sim06]. Therefore, the estimate error covariance matrix completely specifies the joint PDF of the state estimate error vector. For the bootstrap estimator, it can be shown that the estimate error probability distribution is a weighted sum of Gaussian distributions when the measurement noise and disturbance inputs are Gaussian [Teu99]. The weights and covariance matrix associated with each distribution are determined from the real-valued estimate error covariance matrix.

For all three estimators considered in this dissertation, the estimate error covariance matrix plays a critical role in quantifying integrity risk. In the presence of colored measurement noise and time correlated disturbance inputs, this matrix will only be accurate when state augmentation can be performed precisely or when the weighting matrix in WLS estimation can be prescribed exactly. Since the true statistical nature of measurement noise and disturbance inputs are rarely known, approximate, reduced order models are often employed. Under these circumstances, integrity risk cannot be quantified precisely, which can have catastrophic consequences in safety-of-life applications. Therefore, an upper bound on integrity risk must be obtained subject to a specified time correlation uncertainty structure.

## 1.5 Previous and Related Work

The issue of uncertainty in characterizing measurement noise has been studied extensively in the aircraft navigation community. In [DeC00] and [Rif04], it is shown how to compute an upper bound on integrity risk using the concept of cumulative distribution function (CDF) overbounding. These results apply for the special case where measurement errors are mutually independent. However, when state estimation is accomplished through measurement filtering, the presence of any time correlation in the measurement noise violates the independence assumption. In response, the symmetric overbounding theorem was developed in [Rif07] and [Pul08] to generalize CDF overbounding techniques to the case where measurement errors are correlated. The theorem provides a solid theoretical foundation for integrity risk bounding with correlated errors but does not provide a readily implementable solution.

**1.5.1 Recursive Filtering with Model Uncertainty.** Numerous publications can be found in the robust estimation literature that consider the integrity risk bounding problem for specialized cases of modeling uncertainty. For example, in [Kha10a] it is shown how to upper bound integrity risk when the measurement noise is governed by a first order Gauss-Markov model with an unknown time constant. More general uncertainty structures can be handled using guaranteed cost filtering [Xie05], where a linear estimator is sought such that the estimate error variance is guaranteed to be smaller than a certain bound. The design matrices are chosen to minimize the upper bound subject to a specified uncertainty structure on the state transition matrix and observation matrix.

Norm-bounded uncertainty is one type of uncertainty structure that can be handled using guaranteed cost filtering [Xie91] [Xie94] [Pet96] [Zhu02]. Under this structural model, the minimum upper bound on the estimate error variance is obtained by solving algebraic Riccati equations (AREs). The polytopic uncertainty structure is another example, where the state transition matrix and observation matrix are expressed as unknown linear combinations of a set of matrices [Ger98] [Tua01] [Ger00] [Sha01] [Xie04] [Ger02]. The advantage of this formulation is that the robust filtering problem can be written in terms of Linear Matrix Inequalities (LMIs), which can be solved efficiently using existing algorithms [Boy94]. A thorough overview of norm bounded and polytopic uncertainty is provided in [Xie05] and [Lew08].

Guaranteed cost filtering is an effective solution to the integrity risk bounding problem provided that the size of the state transition matrix is known a priori. However, when statistical models for measurement noise and disturbance inputs are not precisely known, the number of states necessary to accurately model these processes is also unknown. Furthermore, certain random processes like flicker noise considered earlier cannot even be modeled with a finite number of states. These problems cannot be addressed using guaranteed cost filtering. New methods must be developed to quantify integrity risk when measurement noise and disturbance inputs have an unknown correlation structure.

**1.5.2 Batch Estimation with Model Uncertainty.** There is no shortage of batch estimation techniques that are robust to specific types of model uncertainty. A nonlinear estimator is designed in [Ham92] when the measurement noise power spectral density

function (PSD) is known to be within a specified neighborhood of a nominal PSD. Using a diagonalized form of the measurement noise covariance matrix, a minimax mean square error estimator is designed in [Eld06] for the case where the diagonal elements lie in a specified convex region. Bayesian methods are employed in [Won92] and [Rei92] to construct a MAP estimator when the measurement noise vector has an unknown positive semi-definite covariance matrix. Maximum likelihood estimators are developed in [Ye95] and [Gor99] for the case where the noise covariance matrix can be parameterized by a small set of parameters. A comprehensive survey of robust estimation techniques is provided in [Kas85].

All of the methods described in the previous paragraph yield optimal state estimators subject to particular measurement noise uncertainty structures. The resulting estimators are either nonlinear or must be determined numerically by solving an optimization problem. However, none of these methods provide the means to determine the probability distribution of the estimate error and they are therefore incapable of quantifying integrity risk. Researchers have made progress toward resolving this issue when the measurement noise covariance matrix can be parameterized by a small set of parameters.

Generalized Least Squares (GLS) estimation can be used in these instances to estimate the covariance matrix parameters in addition to the state vector [Bal11]. Edgeworth expansions are used in [Rot84] to approximate the probability density function of the GLS estimate error. In [Lan12], a method was proposed to upper bound estimation integrity risk for a weighted least squares estimator when the measurement noise could be characterized by a first order Gauss Markov process with unknown

variance and time constant. The methods developed in [Rot84] and [Lan12] depend on the ability to parameterize the noise covariance matrix. They cannot be generalized to situations where the mathematical structure of the measurement noise and disturbance input autocorrelation functions is unknown.

Unlike robust estimation techniques, this dissertation defines the estimator in the traditional manner using the best autocorrelation function models available. This enables implementable algorithms to be derived for the first time that provide integrity risk bounds for real and mixed real/integer estimators when the autocorrelation functions of stochastic inputs are only known to reside between upper and lower bounding functions.

## **1.6 Dissertation Contributions**

**1.6.1 Integrity Risk Bounding in Kalman Filtering.** A new set of linear difference equations is derived in Chapter 2 that explicitly show how measurement noise and disturbance input autocorrelation functions map into the Kalman state estimate error vector. An efficient algorithm is developed to solve these equations and determine the worst-case autocorrelation functions that define the integrity risk bound. The algorithm is applied to an illustrative example of one-dimensional position and velocity estimation.

**1.6.2 Bounding Integrity Risk in Batch Estimation.** A new form of the batch WLS estimate error vector is derived in Chapter 3 for the case where the weighting matrix has unknown, but bounded elements. This expression allows the algorithms developed in Chapter 2 to be used in defining an integrity risk bound for batch WLS estimation. The

method is demonstrated using a two-dimensional position and velocity estimation application.

**1.6.3 Bounding Integrity Risk via Semi-Definite Optimization.** The autocorrelation functions producing the integrity risk bounds described above are not necessarily positive semi-definite. A new bounding approach is developed in Chapter 4 that incorporates positive semi-definiteness as a constraint within a semi-definite programming framework. The method is applicable to both Kalman filters and batch WLS estimators. Performance comparisons are made for the two-dimensional estimation problem.

**1.6.4 Mixed Real/Integer Integrity Risk Bounding.** For the integer bootstrap estimator, the integrity risk bound is defined in Chapter 5 as the solution of a non-convex optimization problem over a polytope. Two novel approaches are developed to conservatively upper bound integrity risk by solving the optimization problem over a circumscribing hyper-rectangle. Their performance is compared for the problem of estimating the scalar distance (baseline) between two stationary antennas using carrier phase differential GPS measurements.

**1.6.5 Mixed Real/Integer Integrity Risk Bounding over a Polytope.** Using a new approach to define the polytope in conjunction with linear programming, a third method is developed in Chapter 5 to upper bound the mixed real/integer integrity risk over the true polytopic feasible region. Simulation results and comparison studies are provided for the static baseline estimation application.

## CHAPTER 2

## INTEGRITY RISK BOUNDING IN KALMAN FILTERING

This chapter is concerned with quantifying the impact of stochastic model uncertainty on integrity risk for the Kalman filter. In Section 2.1, linear measurement and state dynamic models are defined, and assumptions regarding the measurement noise and disturbance inputs are clearly stated. Section 2.2 introduces the method of state augmentation and establishes the role of the estimate error variance in quantifying integrity risk. A set of linear difference equations is derived in Section 2.3 whose solution is used in Section 2.4 to define a new generalized covariance matrix for the Kalman filter.

Section 2.5 uses the generalized covariance matrix to derive an expression for the estimate error variance in terms of measurement noise and disturbance input autocorrelation function values. An upper bound on estimate error variance is derived in Section 2.6 for the bounded autocorrelation uncertainty structure, which is subsequently used to compute the integrity risk bound. Section 2.7 implements the bounding algorithms developed in this chapter for a one-dimensional position and velocity estimation problem.

## 2.1 Measurement and State Dynamic Models

Consider the linear measurement model

$$\mathbf{z}_k = \mathbf{H}_{\theta,k} \boldsymbol{\theta}_k + \mathbf{J}_{v,k} \mathbf{v}_k \quad (2.1)$$

where  $\mathbf{z}_k \in \mathbb{R}^{n_z}$  is the measurement vector,  $\mathbf{H}_{\theta,k} \in \mathbb{R}^{n_z \times n_\theta}$  is the observation matrix,

$\boldsymbol{\theta}_k \in \mathbb{R}^{n_\theta}$  is the state vector,  $\mathbf{J}_{v,k} \in \mathbb{R}^{n_z \times n_v}$  is the measurement noise mapping matrix

and  $\mathbf{v}_k \in \mathbb{R}^{n_v}$  is a zero-mean Gaussian random vector. The subscript  $k$  indicates that a quantity is defined at time index  $k$ .

The state vector  $\boldsymbol{\theta}$  evolves in time according to the linear dynamic model

$$\boldsymbol{\theta}_{k+1} = \mathbf{F}_{\theta,k} \boldsymbol{\theta}_k + \mathbf{G}_{\theta,k} \mathbf{w}_k \quad (2.2)$$

where  $\mathbf{F}_{\theta,k} \in \mathbb{R}^{n_\theta \times n_\theta}$  is the state transition matrix,  $\mathbf{G}_{\theta,k} \in \mathbb{R}^{n_\theta \times n_w}$  is the disturbance input mapping matrix and  $\mathbf{w}_k \in \mathbb{R}^{n_w}$  is a zero-mean Gaussian random vector. In this dissertation,  $\mathbf{H}_{\theta,k}$ ,  $\mathbf{J}_{v,k}$ ,  $\mathbf{F}_{\theta,k}$  and  $\mathbf{G}_{\theta,k}$  are assumed to be precisely known.

The components of  $\mathbf{v}_k$  and  $\mathbf{w}_k$  represent sensor measurement errors and external disturbance inputs, respectively, at time index  $k$ . In the most general case, these components are correlated over time and cross-correlated with each other. Even though cross-correlation can be accommodated by the algorithms provided herein, its presence does not alter the conclusions of this research. Therefore, the assumption of zero cross-correlation will be made for brevity.

Let  $v_{i,k}$  denote the  $i^{\text{th}}$  component of  $\mathbf{v}$  at time index  $k$ , and let  $\{v_i\}$  be a time series of arbitrary length of  $v_i$ . Then  $\mathbf{v}$  is statistically modeled as

$$E[v_{i,k} v_{j,l}] = r_{v_i,kl} \delta_{ij} \quad , \quad \begin{array}{l} i = 1, \dots, n_v \\ j = 1, \dots, n_v \end{array} \quad (2.3)$$

where  $E$  is the expectation operator,  $\delta_{ij}$  is the Kronecker delta, and  $r_{v_i,kl}$  is the value of the autocorrelation function of  $\{v_i\}$  at time indices  $k$  and  $l$ .

A similar statistical model exists for  $\mathbf{w}$ . That is,

$$E[w_{i,k} w_{j,l}] = r_{w_i,kl} \delta_{ij} \quad , \quad \begin{array}{l} i=1, \dots, n_w \\ j=1, \dots, n_w \end{array} \quad (2.4)$$

where  $r_{w_i,kl}$  is the value of the autocorrelation function of  $\{w_i\}$  at time indices  $k$  and  $l$ .

Because the components of  $\mathbf{v}$  and  $\mathbf{w}$  are uncorrelated, it follows that

$$E[w_{i,k} v_{j,l}] = 0 \quad , \quad \forall i, j, k, \text{ and } l \quad (2.5)$$

## 2.2 Quantifying Integrity Risk in Kalman Filtering

In general,  $r_{v_i,kl}$  and  $r_{w_i,kl}$  are non-zero for  $k \neq l$ , indicating that there is time correlation in  $\{v_i\}$  and  $\{w_i\}$ . If  $\mathbf{v}$  and  $\mathbf{w}$  can be dynamically modeled as the output of a linear system driven by white Gaussian noise, then the well-known method of state augmentation can be utilized to recast equations 2.1 through 2.5 in the form of an estimation problem amenable to a Kalman filtering solution.

**2.2.1 Method of State Augmentation.** Suppose that  $\mathbf{v}_k$  and  $\mathbf{w}_k$  can be linearly decomposed as

$$\mathbf{v}_k = \mathbf{C}_k \boldsymbol{\eta}_k + \mathbf{D}_k \mathbf{r}_k \quad (2.6)$$

$$\mathbf{w}_k = \mathbf{A}_k \boldsymbol{\xi}_k + \mathbf{B}_k \mathbf{q}_{\theta,k} \quad (2.7)$$

where  $\mathbf{C}_k \in \mathbb{R}^{n_v \times n_\eta}$ ,  $\boldsymbol{\eta} \in \mathbb{R}^{n_\eta}$  is a zero-mean Gaussian random vector,  $\mathbf{D}_k \in \mathbb{R}^{n_v \times n_r}$  and  $\mathbf{r} \in \mathbb{R}^{n_r}$  is a zero-mean white Gaussian random vector with  $n_r \times n_r$  covariance matrix  $E[\mathbf{r}_k \mathbf{r}_l^T] = \mathbf{R}_k \delta_{kl}$ . Similarly,  $\mathbf{A}_k \in \mathbb{R}^{n_w \times n_\xi}$ ,  $\boldsymbol{\xi} \in \mathbb{R}^{n_\xi}$  is a zero-mean Gaussian

random vector,  $\mathbf{B}_k \in \mathbb{R}^{n_w \times n_{q,\theta}}$  and  $\mathbf{q}_\theta \in \mathbb{R}^{n_{q,\theta}}$  is a zero-mean white Gaussian random vector with  $n_{q,\theta} \times n_{q,\theta}$  covariance matrix  $E[\mathbf{q}_{\theta,k} \mathbf{q}_{\theta,l}^T] = \mathbf{Q}_{\theta,k} \delta_{kl}$ .

It is assumed that  $\boldsymbol{\eta}$  is uncorrelated with  $\mathbf{r}$  and that  $\boldsymbol{\xi}$  is uncorrelated with  $\mathbf{q}_\theta$ , i.e.,  $E[\boldsymbol{\eta}_k \mathbf{r}_l^T] = \mathbf{0}$  and  $E[\boldsymbol{\xi}_k \mathbf{q}_{\theta,l}^T] = \mathbf{0}$  for all  $k$  and  $l$ . The vectors  $\boldsymbol{\eta}$  and  $\boldsymbol{\xi}$  capture the time correlated nature of  $\mathbf{v}$  and  $\mathbf{w}$ , respectively, and evolve in time according to the linear dynamic models

$$\boldsymbol{\eta}_{k+1} = \mathbf{F}_{\eta,k} \boldsymbol{\eta}_k + \mathbf{G}_{\eta,k} \mathbf{q}_{\eta,k} \quad (2.8)$$

$$\boldsymbol{\xi}_{k+1} = \mathbf{F}_{\xi,k} \boldsymbol{\xi}_k + \mathbf{G}_{\xi,k} \mathbf{q}_{\xi,k} \quad (2.9)$$

where  $\mathbf{F}_{\eta,k} \in \mathbb{R}^{n_\eta \times n_\eta}$ ,  $\mathbf{G}_{\eta,k} \in \mathbb{R}^{n_\eta \times n_{q,\eta}}$ ,  $\mathbf{q}_\eta \in \mathbb{R}^{n_{q,\eta}}$  is a zero-mean white Gaussian random vector with  $n_{q,\eta} \times n_{q,\eta}$  covariance matrix  $E[\mathbf{q}_{\eta,k} \mathbf{q}_{\eta,l}^T] = \mathbf{Q}_{\eta,k} \delta_{kl}$ ,  $\mathbf{F}_{\xi,k} \in \mathbb{R}^{n_\xi \times n_\xi}$ ,  $\mathbf{G}_{\xi,k} \in \mathbb{R}^{n_\xi \times n_{q,\xi}}$  and  $\mathbf{q}_\xi \in \mathbb{R}^{n_{q,\xi}}$  is a zero-mean white Gaussian random vector with  $n_{q,\xi} \times n_{q,\xi}$  covariance matrix  $E[\mathbf{q}_{\xi,k} \mathbf{q}_{\xi,l}^T] = \mathbf{Q}_{\xi,k} \delta_{kl}$ .

Substituting equation 2.7 into equation 2.2 and appending equations 2.8 and 2.9 to equation 2.2 results in the new linear dynamic model

$$\begin{bmatrix} \boldsymbol{\theta}_{k+1} \\ \boldsymbol{\xi}_{k+1} \\ \boldsymbol{\eta}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{\theta,k} & \mathbf{G}_{\theta,k} \mathbf{A}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{\xi,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{\eta,k} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_k \\ \boldsymbol{\xi}_k \\ \boldsymbol{\eta}_k \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{\theta,k} \mathbf{B}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{\xi,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_{\eta,k} \end{bmatrix} \begin{bmatrix} \mathbf{q}_{\theta,k} \\ \mathbf{q}_{\xi,k} \\ \mathbf{q}_{\eta,k} \end{bmatrix} \quad (2.10)$$

which can be written more succinctly as

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{q}_k \quad (2.11)$$

Defining  $n_x = n_\theta + n_\eta + n_\varepsilon$  and  $n_q = n_{q,\theta} + n_{q,\eta} + n_{q,\xi}$ ,  $\mathbf{x}_k \in \mathbb{R}^{n_x}$  is the state vector,  $\mathbf{F}_k \in \mathbb{R}^{n_x \times n_x}$  is the state transition matrix,  $\mathbf{G}_k \in \mathbb{R}^{n_x \times n_q}$  and  $\mathbf{q} \in \mathbb{R}^{n_q}$  is a zero-mean white Gaussian random vector with  $n_q \times n_q$  covariance matrix

$$\mathbf{Q}_k = E[\mathbf{q}_k \mathbf{q}_k^T] = \begin{bmatrix} \mathbf{Q}_{\theta,k} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\xi,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_{\eta,k} \end{bmatrix} \delta_{kl} \quad (2.12)$$

Inserting equation 2.6 into equation 2.1 results in the new measurement model

$$\mathbf{z}_k = \mathbf{H}_{\theta,k} \boldsymbol{\theta}_k + \mathbf{J}_{v,k} \mathbf{C}_k \boldsymbol{\eta}_k + \mathbf{J}_{v,k} \mathbf{D}_k \mathbf{r}_k \quad (2.13)$$

Defining  $\mathbf{H}_k \in \mathbb{R}^{n_z \times n_x}$  and  $\mathbf{J}_k \in \mathbb{R}^{n_z \times n_r}$  as

$$\mathbf{H}_k = [\mathbf{H}_{\theta,k} \quad \mathbf{0} \quad \mathbf{J}_{v,k} \mathbf{C}_k] \quad (2.14)$$

$$\mathbf{J}_k = \mathbf{J}_{v,k} \mathbf{D}_k \quad (2.15)$$

allows equation 2.13 to be written more compactly as

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{J}_k \mathbf{r}_k \quad (2.16)$$

An optimal estimate of the state vector  $\mathbf{x}$  can now be obtained using the Kalman filter.

**2.2.2 State Estimation via Kalman Filtering.** The Kalman filter is composed of a measurement update, producing the estimate vector  $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$ , and a time update, which yields the estimate vector  $\hat{\mathbf{x}}_{k+1}^- = \mathbf{F}_k \hat{\mathbf{x}}_k$  [Bro97]. The vectors  $\hat{\mathbf{x}}_k$ ,  $\hat{\mathbf{x}}_k^-$  and  $\hat{\mathbf{x}}_{k+1}^-$  have dimension  $n_x \times 1$  and  $\mathbf{K}_k \in \mathbb{R}^{n_x \times n_z}$  is the Kalman gain matrix, to be defined shortly.

The estimate error vectors associated with  $\hat{\mathbf{x}}_k$  and  $\hat{\mathbf{x}}_{k+1}^-$  are defined as  $\boldsymbol{\varepsilon}_k = \hat{\mathbf{x}}_k - \mathbf{x}_k$  and  $\boldsymbol{\varepsilon}_{k+1}^- = \hat{\mathbf{x}}_{k+1}^- - \mathbf{x}_{k+1}$ , respectively, whose covariance matrices are determined from the expressions [Bro97]

$$\hat{\mathbf{P}}_k = E[\boldsymbol{\varepsilon}_k \boldsymbol{\varepsilon}_k^T] = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \hat{\mathbf{P}}_k^- \quad (2.17)$$

$$\hat{\mathbf{P}}_{k+1}^- = E[\boldsymbol{\varepsilon}_{k+1}^- (\boldsymbol{\varepsilon}_{k+1}^-)^T] = \mathbf{F}_k \hat{\mathbf{P}}_k \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (2.18)$$

$$\mathbf{K}_k = \hat{\mathbf{P}}_k^- \mathbf{H}_k^T (\mathbf{H}_k \hat{\mathbf{P}}_k^- \mathbf{H}_k^T + \mathbf{J}_k \mathbf{R}_k \mathbf{J}_k^T)^{-1} \quad (2.19)$$

where  $\hat{\mathbf{P}}_k^- \in \mathbb{R}^{n_x \times n_x}$  is output during the time update and  $\hat{\mathbf{P}}_k \in \mathbb{R}^{n_x \times n_x}$  is output during the measurement update.

Throughout the dissertation,  $\boldsymbol{\varepsilon}^-$  and  $\boldsymbol{\varepsilon}$  will commonly be referred to as the *a priori* and *a posteriori* estimate error vectors, respectively. Notice that  $\boldsymbol{\varepsilon}^-$  and  $\boldsymbol{\varepsilon}$  are both Gaussian random vectors, which follows from the Gaussianity of  $\mathbf{r}$  and  $\mathbf{q}$  and the fact that the Kalman filter is a linear estimator. In general, the mean vector and covariance matrix completely specify the joint probability density function of a Gaussian random vector. The mean of  $\boldsymbol{\varepsilon}^-$  and  $\boldsymbol{\varepsilon}$  is zero because  $\mathbf{r}$  and  $\mathbf{q}$  are both zero-mean, and because the Kalman filter is an unbiased estimator.

Therefore, equations 2.17 and 2.18 completely specify the joint probability density function of the state estimate error vector after a measurement update and time update of the Kalman filter.  $\hat{\mathbf{P}}_k$  and  $\hat{\mathbf{P}}_{k+1}^-$  play a critical role in verifying that the likelihood of entering a hazardous situation is acceptably small for safety-critical estimation applications.

**2.2.3 Definition of Integrity Risk.** Let  $y$  be a scalar linear combination of  $\mathbf{x}$ . That is,  $y = \boldsymbol{\alpha}_y^T \mathbf{x}$ , where  $\boldsymbol{\alpha}_y \in \mathbb{R}^{n_x}$  is a known vector. The estimation integrity risk of  $y$  is defined in this dissertation as

$$I_y = P(\varepsilon_y \notin [-\ell_y, \ell_y]) \quad (2.20)$$

where  $\ell_y$  is a specified real number.

Given that  $\varepsilon_y$  is a zero-mean Gaussian random variable,  $I_y$  evaluates to

$$I_y = \operatorname{erfc}\left(\frac{\ell_y}{\sqrt{2\hat{\sigma}_y^2}}\right) \quad (2.21)$$

where  $\operatorname{erfc}(\bullet)$  is the complementary error function and  $\hat{\sigma}_y^2$  is the variance of  $\varepsilon_y$  obtained from the covariance transformation  $\hat{\sigma}_y^2 = \boldsymbol{\alpha}_y^T \hat{\mathbf{P}} \boldsymbol{\alpha}_y$ . The *a priori* integrity risk is computed in an analogous manner by replacing  $\hat{\sigma}_y^2$  with  $(\hat{\sigma}_y^-)^2$  and  $\hat{\mathbf{P}}$  with  $\hat{\mathbf{P}}^-$ .

It is clear that the evaluation of  $I_y$  is straightforward once  $\hat{\mathbf{P}}$  has been specified. However, it was shown in Section 2.2.1 that when the components of  $\mathbf{v}$  and  $\mathbf{w}$  are time correlated, dynamic models for  $\mathbf{v}$  and  $\mathbf{w}$  needed to be specified in order to propagate  $\hat{\mathbf{P}}$  and  $\hat{\mathbf{P}}^-$ . In practice, these dynamic models will never be known precisely because of the uncertainty in the true nature of the time correlation. As a result, the computed  $\hat{\mathbf{P}}$ , even using the best dynamic models available, will not accurately describe the probability distribution of the *a posteriori* estimate error vector.

### 2.3 Estimate Error Vector Difference Equations

Notice that  $\boldsymbol{\theta}$  is the only subset of  $\boldsymbol{x}$  pertinent to integrity risk. The vectors  $\boldsymbol{\xi}$  and  $\boldsymbol{\eta}$  were appended to  $\boldsymbol{\theta}$  solely for the purpose of ensuring that  $\hat{\mathbf{P}}_\theta$  faithfully describes the probability distribution of  $\boldsymbol{\varepsilon}_\theta$ . In this section, two vector difference equations will be derived for the Kalman filter that explicitly reveal how  $\boldsymbol{v}$  and  $\boldsymbol{w}$  map into  $\boldsymbol{\varepsilon}_\theta$ .

**2.3.1 A Posteriori Generalized Error Vector.** It was shown in Section 2.2.2 that the measurement update is given by  $\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \mathbf{K}_k (\boldsymbol{z}_k - \mathbf{H}_k \hat{\boldsymbol{x}}_k^-)$ . Replacing  $\boldsymbol{z}_k$  with the expression on the right hand side of equation 2.1 yields

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \mathbf{K}_k (\mathbf{H}_{\theta,k} \boldsymbol{\theta}_k + \mathbf{J}_{\nu,k} \boldsymbol{\nu}_k - \mathbf{H}_k \hat{\boldsymbol{x}}_k^-) \quad (2.22)$$

Substituting equation 2.14 for  $\mathbf{H}_k$  and partitioning  $\hat{\boldsymbol{x}}_k$  and  $\hat{\boldsymbol{x}}_k^-$  results in

$$\begin{bmatrix} \hat{\boldsymbol{\theta}}_k \\ \hat{\boldsymbol{\xi}}_k \\ \hat{\boldsymbol{\eta}}_k \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\theta}}_k^- \\ \hat{\boldsymbol{\xi}}_k^- \\ \hat{\boldsymbol{\eta}}_k^- \end{bmatrix} - \mathbf{K}_k \begin{bmatrix} \mathbf{H}_{\theta,k} & \mathbf{0} & \mathbf{J}_{\nu,k} \mathbf{C}_k \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\theta}}_k^- \\ \hat{\boldsymbol{\xi}}_k^- \\ \hat{\boldsymbol{\eta}}_k^- \end{bmatrix} + \mathbf{K}_k (\mathbf{H}_{\theta,k} \boldsymbol{\theta}_k + \mathbf{J}_{\nu,k} \boldsymbol{\nu}_k) \quad (2.23)$$

which can be simplified into the form

$$\begin{bmatrix} \hat{\boldsymbol{\theta}}_k \\ \hat{\boldsymbol{\xi}}_k \\ \hat{\boldsymbol{\eta}}_k \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\theta}}_k^- \\ \hat{\boldsymbol{\xi}}_k^- \\ \hat{\boldsymbol{\eta}}_k^- \end{bmatrix} - \mathbf{K}_k \mathbf{H}_k \begin{bmatrix} \hat{\boldsymbol{\theta}}_k^- - \boldsymbol{\theta}_k \\ \hat{\boldsymbol{\xi}}_k^- \\ \hat{\boldsymbol{\eta}}_k^- \end{bmatrix} + \mathbf{K}_k \mathbf{J}_{\nu,k} \boldsymbol{\nu}_k \quad (2.24)$$

Subtracting the  $n_x \times 1$  vector  $[\boldsymbol{\theta}_k^T \quad \mathbf{0}^T \quad \mathbf{0}^T]^T$  from both sides yields

$$\begin{bmatrix} \hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k \\ \hat{\boldsymbol{\xi}}_k \\ \hat{\boldsymbol{\eta}}_k \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\theta}}_k^- - \boldsymbol{\theta}_k \\ \hat{\boldsymbol{\xi}}_k^- \\ \hat{\boldsymbol{\eta}}_k^- \end{bmatrix} - \mathbf{K}_k \mathbf{H}_k \begin{bmatrix} \hat{\boldsymbol{\theta}}_k^- - \boldsymbol{\theta}_k \\ \hat{\boldsymbol{\xi}}_k^- \\ \hat{\boldsymbol{\eta}}_k^- \end{bmatrix} + \mathbf{K}_k \mathbf{J}_{v,k} \mathbf{v}_k \quad (2.25)$$

Defining the  $n_\theta \times 1$  vectors  $\boldsymbol{\varepsilon}_{\theta,k} = \hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k$  and  $\boldsymbol{\varepsilon}_{\theta,k}^- = \hat{\boldsymbol{\theta}}_k^- - \boldsymbol{\theta}_k$  allows equation 2.25 to be written as

$$\begin{bmatrix} \boldsymbol{\varepsilon}_{\theta,k} \\ \hat{\boldsymbol{\xi}}_k \\ \hat{\boldsymbol{\eta}}_k \end{bmatrix} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \begin{bmatrix} \boldsymbol{\varepsilon}_{\theta,k}^- \\ \hat{\boldsymbol{\xi}}_k^- \\ \hat{\boldsymbol{\eta}}_k^- \end{bmatrix} + \mathbf{K}_k \mathbf{J}_{v,k} \mathbf{v}_k \quad (2.26)$$

Now define  $\mathbf{e}_k = [\boldsymbol{\varepsilon}_{\theta,k}^T \quad \hat{\boldsymbol{\xi}}_k^T \quad \hat{\boldsymbol{\eta}}_k^T]^T$ ,  $\mathbf{L}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)$  and  $\mathbf{M}_k = \mathbf{K}_k \mathbf{J}_{v,k}$ . Then equation 2.26 can be expressed more succinctly as

$$\mathbf{e}_k = \mathbf{L}_k \mathbf{e}_k^- + \mathbf{M}_k \mathbf{v}_k \quad (2.27)$$

where  $\mathbf{e}_k \in \mathbb{R}^{n_x}$ ,  $\mathbf{L}_k \in \mathbb{R}^{n_x \times n_x}$  and  $\mathbf{M}_k \in \mathbb{R}^{n_x \times n_v}$ .

**2.3.2 A Priori Generalized Error Vector.** A similar procedure can be followed for the Kalman time update. First write the update equation  $\hat{\mathbf{x}}_{k+1}^- = \mathbf{F}_k \hat{\mathbf{x}}_k$  in the partitioned form

$$\begin{bmatrix} \hat{\boldsymbol{\theta}}_{k+1}^- \\ \hat{\boldsymbol{\xi}}_{k+1}^- \\ \hat{\boldsymbol{\eta}}_{k+1}^- \end{bmatrix} = \mathbf{F}_k \begin{bmatrix} \hat{\boldsymbol{\theta}}_k \\ \hat{\boldsymbol{\xi}}_k \\ \hat{\boldsymbol{\eta}}_k \end{bmatrix} \quad (2.28)$$

Subtracting the  $n_x \times 1$  vector  $[\boldsymbol{\theta}_{k+1}^T \quad \mathbf{0}^T \quad \mathbf{0}^T]^T$  from both sides yields

$$\begin{bmatrix} \hat{\boldsymbol{\theta}}_{k+1}^- - \boldsymbol{\theta}_{k+1} \\ \hat{\boldsymbol{\xi}}_{k+1}^- \\ \hat{\boldsymbol{\eta}}_{k+1}^- \end{bmatrix} = \mathbf{F}_k \begin{bmatrix} \hat{\boldsymbol{\theta}}_k \\ \hat{\boldsymbol{\xi}}_k \\ \hat{\boldsymbol{\eta}}_k \end{bmatrix} - \begin{bmatrix} \boldsymbol{\theta}_{k+1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad (2.29)$$

From equation 2.2,  $\boldsymbol{\theta}_{k+1} = \mathbf{F}_{\theta,k} \boldsymbol{\theta}_k + \mathbf{G}_{\theta,k} \mathbf{w}_k$ , which can also be written as

$$\begin{bmatrix} \boldsymbol{\theta}_{k+1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{\theta,k} & \mathbf{G}_{\theta,k} \mathbf{A}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{\xi,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{\eta,k} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_k \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{\theta,k} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \mathbf{w}_k \quad (2.30)$$

The  $3 \times 3$  block matrix on the right hand side of equation 2.30 is identical to  $\mathbf{F}_k$  defined in equation 2.11. Therefore, equation 2.30 can be simplified to

$$\begin{bmatrix} \boldsymbol{\theta}_{k+1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \mathbf{F}_k \begin{bmatrix} \boldsymbol{\theta}_k \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{\theta,k} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \mathbf{w}_k \quad (2.31)$$

Replacing the vector  $[\boldsymbol{\theta}_{k+1}^T \ \mathbf{0}^T \ \mathbf{0}^T]^T$  in equation 2.29 with the right hand side of equation 2.31 and simplifying yields

$$\begin{bmatrix} \hat{\boldsymbol{\theta}}_{k+1}^- - \boldsymbol{\theta}_{k+1} \\ \hat{\boldsymbol{\xi}}_{k+1}^- \\ \hat{\boldsymbol{\eta}}_{k+1}^- \end{bmatrix} = \mathbf{F}_k \begin{bmatrix} \hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k \\ \hat{\boldsymbol{\xi}}_k \\ \hat{\boldsymbol{\eta}}_k \end{bmatrix} - \begin{bmatrix} \mathbf{G}_{\theta,k} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \mathbf{w}_k \quad (2.32)$$

Recalling the definitions of  $\mathbf{e}_k$  and  $\mathbf{e}_k^-$  given after equation 2.25, equation 2.32 can also be written as

$$\mathbf{e}_{k+1}^- = \mathbf{F}_k \mathbf{e}_k + \mathbf{N}_k \mathbf{w}_k \quad (2.33)$$

where the definition of  $\mathbf{N}_k \in \mathbb{R}^{n_x \times n_w}$  is obvious by comparing equation 2.33 to equation 2.32.

## 2.4 Generalized Covariance Matrix for the Kalman Filter

Together, equations 2.27 and 2.33 constitute a system of linear, time-varying difference equations driven by  $\mathbf{v}$  and  $\mathbf{w}$ . Throughout the derivation in Section 2.3, no assumptions were made regarding the mathematical structure of the vector time series  $\{\mathbf{v}\}$  and  $\{\mathbf{w}\}$ . Hence, equations 2.27 and 2.33 represent the true error in the estimate of the vector  $\boldsymbol{\theta}$  after a measurement update and time update of the Kalman filter. In this section, these two equations will be used to derive a new generalized covariance matrix for the Kalman filter.

**2.4.1 Preliminary Analysis of Covariance.** From equation 2.27, the generalized covariance matrix  $\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T]$  can be written as

$$\mathbf{P}_k = E[(\mathbf{L}_k \mathbf{e}_k^- + \mathbf{M}_k \mathbf{v}_k)(\mathbf{L}_k \mathbf{e}_k^- + \mathbf{M}_k \mathbf{v}_k)^T] \quad (2.34)$$

Expanding equation 2.34 results in the expression

$$\mathbf{P}_k = \mathbf{L}_k \mathbf{P}_k^- \mathbf{L}_k^T + \mathbf{L}_k E[\mathbf{e}_k^- \mathbf{v}_k^T] \mathbf{M}_k^T + \mathbf{M}_k E[\mathbf{v}_k (\mathbf{e}_k^-)^T] \mathbf{L}_k^T + \mathbf{M}_k \mathbf{V}_k \mathbf{M}_k^T \quad (2.35)$$

where  $\mathbf{V}_k \in \mathbb{R}^{n_v \times n_v}$  is the covariance matrix of  $\mathbf{v}_k$ .

The expected values on the right hand side of equation 2.35 are non-zero. To see this, first consider the case when  $k=0$ . From equation 2.33,  $\mathbf{e}_1^- = \mathbf{F}_0 \mathbf{e}_0 + \mathbf{N}_0 \mathbf{w}_0$ .

Replacing  $\mathbf{e}_0$  by the expression on the right hand side of equation 2.27 yields

$$\mathbf{e}_1^- = \mathbf{F}_0 (\mathbf{L}_0 \mathbf{e}_0^- + \mathbf{M}_0 \mathbf{v}_0) + \mathbf{N}_0 \mathbf{w}_0 \quad (2.36)$$

Post multiplying both sides by  $\mathbf{v}_1^T$  and taking the expected value results in

$$E[\mathbf{e}_1^- \mathbf{v}_1^T] = \mathbf{F}_0 \mathbf{L}_0 E[\mathbf{e}_0^- \mathbf{v}_1^T] + \mathbf{F}_0 \mathbf{M}_0 E[\mathbf{v}_0 \mathbf{v}_1^T] + \mathbf{N}_0 E[\mathbf{w}_0 \mathbf{v}_1^T] \quad (2.37)$$

The initial error  $\mathbf{e}_0^-$  is uncorrelated with  $\mathbf{v}_1$  and  $\mathbf{w}_0$  is uncorrelated with  $\mathbf{v}_1$  by virtue of equation 2.5. Therefore,  $E[\mathbf{e}_0^- \mathbf{v}_1^T] = \mathbf{0}$  and  $E[\mathbf{w}_0 \mathbf{v}_1^T] = \mathbf{0}$ . Defining  $\Delta_1 \in \mathbb{R}^{n_x \times n_v}$  as  $\Delta_1 = \mathbf{F}_0 \mathbf{M}_0$ , equation 2.37 becomes

$$E[\mathbf{e}_1^- \mathbf{v}_1^T] = \Delta_1 E[\mathbf{v}_0 \mathbf{v}_1^T] \quad (2.38)$$

When  $k=1$ , equation 2.33 can be written as  $\mathbf{e}_2^- = \mathbf{F}_1 \mathbf{e}_1 + \mathbf{N}_1 \mathbf{w}_1$ . Replacing  $\mathbf{e}_1$  by the expression on the right hand side of equation 2.27 yields

$$\mathbf{e}_2^- = \mathbf{F}_1 (\mathbf{L}_1 \mathbf{e}_1^- + \mathbf{M}_1 \mathbf{v}_1) + \mathbf{N}_1 \mathbf{w}_1 \quad (2.39)$$

Post multiplying both sides by  $\mathbf{v}_2^T$  and taking the expected value results in

$$E[\mathbf{e}_2^- \mathbf{v}_2^T] = \mathbf{F}_1 \mathbf{L}_1 E[\mathbf{e}_1^- \mathbf{v}_2^T] + \mathbf{F}_1 \mathbf{M}_1 E[\mathbf{v}_1 \mathbf{v}_2^T] \quad (2.40)$$

where the relation  $E[\mathbf{w}_1 \mathbf{v}_2^T] = \mathbf{0}$  has been used.

From equation 2.38,  $E[\mathbf{e}_1^- \mathbf{v}_2^T] = \Delta_1 E[\mathbf{v}_0 \mathbf{v}_2^T]$ . Defining  $\Delta_2 \in \mathbb{R}^{n_x \times n_v}$  as  $\Delta_2 = \mathbf{F}_1 \mathbf{L}_1 \Delta_1$  and redefining  $\Delta_1 \in \mathbb{R}^{n_x \times n_v}$  as  $\Delta_1 = \mathbf{F}_1 \mathbf{M}_1$ , equation 2.40 can be written as

$$E[\mathbf{e}_2^- \mathbf{v}_2^T] = \Delta_2 E[\mathbf{v}_0 \mathbf{v}_2^T] + \Delta_1 E[\mathbf{v}_1 \mathbf{v}_2^T] \quad (2.41)$$

When  $k=2$ , it can be shown that

$$E[\mathbf{e}_3^- \mathbf{v}_3^T] = \Delta_3 E[\mathbf{v}_0 \mathbf{v}_3^T] + \Delta_2 E[\mathbf{v}_1 \mathbf{v}_3^T] + \Delta_1 E[\mathbf{v}_2 \mathbf{v}_3^T] \quad (2.42)$$

where  $\Delta_3 = \mathbf{F}_2 \mathbf{L}_2 \Delta_2$ ,  $\Delta_2 = \mathbf{F}_2 \mathbf{L}_2 \Delta_1$  and  $\Delta_1 = \mathbf{F}_2 \mathbf{M}_2$ .

In general,  $E[\mathbf{e}_k^- \mathbf{v}_k^T]$  can be expressed in the form

$$E[\mathbf{e}_k^- \mathbf{v}_k^T] = \sum_{i=1}^k \Delta_i E[\mathbf{v}_{k-i} \mathbf{v}_k^T] \quad (2.43)$$

where  $\Delta_k = \mathbf{F}_{k-1} \mathbf{L}_{k-1} \Delta_{k-1}$ ,  $\Delta_{k-1} = \mathbf{F}_{k-1} \mathbf{L}_{k-1} \Delta_{k-2}$ ,  $\dots$ ,  $\Delta_2 = \mathbf{F}_{k-1} \mathbf{L}_{k-1} \Delta_1$ ,  $\Delta_1 = \mathbf{F}_{k-1} \mathbf{M}_{k-1}$ .

The fact that  $E[\mathbf{e}_k^- \mathbf{v}_k^T] \neq \mathbf{0}$   $E[\mathbf{e}_k^- \mathbf{v}_k^T] \neq \mathbf{0}$  implies that a recursive update equation cannot be derived for  $\mathbf{P}_k$ . In order to obtain an efficient algorithm for the computation of  $\mathbf{P}_k$  and  $\mathbf{P}_{k+1}^-$ , equations 2.27 and 2.33 must be treated as a system of linear, time-varying difference equations.

**2.4.2 Closed-Form Expressions for Generalized Covariance.** The solution to any linear system of difference equations can be written in terms of an initial condition response and an impulse response [Che99].

$$\mathbf{e}_k = \Phi_k \mathbf{e}_0^- + \sum_{j=1}^{n_w} \Gamma_{j,k} \{w_j\}_0^{k-1} + \sum_{l=1}^{n_v} \Lambda_{l,k} \{v_l\}_0^k \quad (2.44)$$

$$\mathbf{e}_k^- = \Phi_k^- \mathbf{e}_0^- + \sum_{j=1}^{n_w} \Gamma_{j,k}^- \{w_j\}_0^{k-1} + \sum_{l=1}^{n_v} \Lambda_{l,k}^- \{v_l\}_0^{k-1} \quad (2.45)$$

where  $\Phi_k$  and  $\Phi_k^-$  are  $n_x \times n_x$  matrices,  $\Gamma_{j,k}$  and  $\Gamma_{j,k}^-$  are  $n_x \times k$  matrices,  $\Lambda_{l,k}$  is an  $n_x \times (k+1)$  matrix,  $\Lambda_{l,k}^-$  is an  $n_x \times k$  matrix,  $\{w_j\}_0^{k-1}$  is a  $k \times 1$  time series vector for  $w_j$  from time index 0 to time index  $k-1$  and  $\{v_l\}_0^{k-1}$  is a  $k \times 1$  time series vector for  $v_l$  from time index 0 to time index  $k-1$ .

An algorithm for the computation of  $\Phi_k$ ,  $\Phi_k^-$ ,  $\Gamma_{j,k}$ ,  $\Gamma_{j,k}^-$ ,  $\Lambda_{l,k}$  and  $\Lambda_{l,k}^-$  is provided in Appendix A. Evaluating  $E[\mathbf{e}_k \mathbf{e}_k^T]$  and  $E[\mathbf{e}_k^- (\mathbf{e}_k^-)^T]$  results in the following expressions for  $\mathbf{P}_k$  and  $\mathbf{P}_k^-$

$$\mathbf{P}_k = \Phi_k \mathbf{P}_0^- \Phi_k^T + \sum_{j=1}^{n_w} \Gamma_{j,k} \mathbf{W}_{j,k-1} \Gamma_{j,k}^T + \sum_{l=1}^{n_v} \Lambda_{l,k} \mathbf{V}_{l,k} \Lambda_{l,k}^T \quad (2.46)$$

$$\mathbf{P}_k^- = \Phi_k^- \mathbf{P}_0^- (\Phi_k^-)^T + \sum_{j=1}^{n_w} \Gamma_{j,k}^- \mathbf{W}_{j,k-1} (\Gamma_{j,k}^-)^T + \sum_{l=1}^{n_v} \Lambda_{l,k}^- \mathbf{V}_{l,k-1} (\Lambda_{l,k}^-)^T \quad (2.47)$$

where  $\mathbf{P}_0^- \in \mathbb{R}^{n_x \times n_x}$ ,  $\mathbf{W}_{j,k-1} \in \mathbb{R}^{k \times k}$ ,  $\mathbf{V}_{l,k} \in \mathbb{R}^{(k+1) \times (k+1)}$  and  $\mathbf{V}_{l,k-1} \in \mathbb{R}^{k \times k}$  are given

by

$$\mathbf{P}_0^- = E[\mathbf{e}_0^- (\mathbf{e}_0^-)^T] \quad (2.48)$$

$$\mathbf{W}_{j,k-1} = E[\{w_j\}_0^{k-1} (\{w_j\}_0^{k-1})^T] \quad (2.49)$$

$$\mathbf{V}_{l,k} = E[\{v_l\}_0^k (\{v_l\}_0^k)^T] \quad (2.50)$$

$$\mathbf{V}_{l,k-1} = E[\{v_l\}_0^{k-1} (\{v_l\}_0^{k-1})^T] \quad (2.51)$$

$\mathbf{W}_{j,k-1}$ ,  $\mathbf{V}_{l,k-1}$  and  $\mathbf{V}_{l,k}$  all have a similar structure and are populated using equations 2.3 and 2.4. For example

$$\mathbf{W}_{j,k-1} = \begin{bmatrix} r_{w_j,11} & r_{w_j,12} & \cdots & r_{w_j,1k} \\ r_{w_j,12} & r_{w_j,22} & \cdots & r_{w_j,2k} \\ \vdots & \vdots & \ddots & \vdots \\ r_{w_j,1k} & r_{w_j,2k} & \cdots & r_{w_j,kk} \end{bmatrix} \quad (2.52)$$

$\mathbf{V}_{l,k-1}$  is similar to  $\mathbf{W}_{j,k-1}$  except that each  $r_{w_j,mn}$  is replaced by  $r_{v_l,mn}$ .

## 2.5 Estimate Error Variance as a Function of Autocorrelation

For the linear combination  $y = \boldsymbol{\alpha}^T \mathbf{x}$ , the *a posteriori* estimate error variance is determined by applying a covariance transformation to equation 2.46. That is

$$\sigma_{y,k}^2 = \boldsymbol{\alpha}_y^T \boldsymbol{\Phi}_k \mathbf{P}_0^- \boldsymbol{\Phi}_k^T \boldsymbol{\alpha}_y + \sum_{j=1}^{n_w} \boldsymbol{\alpha}_y^T \boldsymbol{\Gamma}_{j,k} \mathbf{W}_{j,k-1} \boldsymbol{\Gamma}_{j,k}^T \boldsymbol{\alpha}_y + \sum_{l=1}^{n_v} \boldsymbol{\alpha}_y^T \boldsymbol{\Lambda}_{l,k} \mathbf{V}_{l,k} \boldsymbol{\Lambda}_{l,k}^T \boldsymbol{\alpha}_y \quad (2.53)$$

A similar expression can be obtained for  $(\sigma_{y,k}^-)^2$  by applying the same transformation to equation 2.47. The two summations on the right hand side of equation 2.53 are sums of quadratic forms in  $\mathbf{W}_{j,k-1}$  and  $\mathbf{V}_{l,k}$ . Through proper manipulation, they can be expressed directly in terms of the autocorrelation function values  $r_{v_l,kl}$  and  $r_{w_j,kl}$ .

To illustrate, consider the term  $\boldsymbol{\alpha}_y^T \boldsymbol{\Gamma}_{j,k} \mathbf{W}_{j,k-1} \boldsymbol{\Gamma}_{j,k}^T \boldsymbol{\alpha}_y$ . Defining  $\boldsymbol{\psi} \in \mathbb{R}^k$  as  $\boldsymbol{\psi} = \boldsymbol{\Gamma}_{j,k}^T \boldsymbol{\alpha}_y$  allows the quadratic form to be written as

$$\boldsymbol{\alpha}_y^T \boldsymbol{\Gamma}_{j,k} \mathbf{W}_{j,k-1} \boldsymbol{\Gamma}_{j,k}^T \boldsymbol{\alpha}_y = \sum_{l=1}^k \psi_l [\mathbf{W}_{j,k-1} \boldsymbol{\psi}]_l \quad (2.54)$$

Expressing the product  $\mathbf{W}_{j,k-1} \boldsymbol{\psi}$  using summation notation results in

$$\boldsymbol{\alpha}_y^T \boldsymbol{\Gamma}_{j,k} \mathbf{W}_{j,k-1} \boldsymbol{\Gamma}_{j,k}^T \boldsymbol{\alpha}_y = \sum_{l=1}^k \psi_l \left\{ \sum_{m=1}^k (\mathbf{W}_{j,k-1})_{lm} \psi_m \right\} \quad (2.55)$$

which can also be written as

$$\boldsymbol{\alpha}_y^T \boldsymbol{\Gamma}_{j,k} \mathbf{W}_{j,k-1} \boldsymbol{\Gamma}_{j,k}^T \boldsymbol{\alpha}_y = \sum_{l=1}^k \sum_{m=1}^k (\mathbf{W}_{j,k-1})_{lm} \psi_l \psi_m \quad (2.56)$$

Defining  $\boldsymbol{\Psi} \in \mathbb{R}^{k \times k}$  as  $\boldsymbol{\Psi} = \boldsymbol{\psi} \boldsymbol{\psi}^T$ , equation 2.56 can be written in the form

$$\boldsymbol{\alpha}_y^T \boldsymbol{\Gamma}_{j,k} \mathbf{W}_{j,k-1} \boldsymbol{\Gamma}_{j,k}^T \boldsymbol{\alpha}_y = \sum_{l=1}^k \sum_{m=1}^k [\mathbf{W}_{j,k-1} \circ \boldsymbol{\Psi}]_{lm} \quad (2.57)$$

where the symbol  $\circ$  indicates the entry-wise product of two matrices.

With the aid of equation 2.52, and temporarily dropping the  $j$  and  $k$  subscripts, the right hand side of equation 2.57 can be written as

$$\begin{aligned} \sum_{l=1}^k \sum_{m=1}^k [\mathbf{W} \circ \boldsymbol{\Psi}]_{lm} &= \gamma_{w,11} r_{w,11} + \cdots + \gamma_{w,kk} r_{w,kk} + \\ &2\gamma_{w,12} r_{w,12} + \cdots + 2\gamma_{w,(k-1)k} r_{w,(k-1)k} + \\ &2\gamma_{w,13} r_{w,13} + \cdots + 2\gamma_{w,(k-2)k} r_{w,(k-2)k} + \cdots + 2\gamma_{w,1k} r_{w,1k} \end{aligned} \quad (2.58)$$

where  $\gamma_{w,ij}$  is the  $(i, j)$  element of  $\boldsymbol{\Psi}$ .

Equation 2.58 can also be written in the more compact form

$$\sum_{l=1}^k \sum_{m=1}^k [\mathbf{W} \circ \boldsymbol{\Psi}]_{lm} = \sum_{m=1}^k \gamma_{w,mm} r_{w,mm} + 2 \sum_{m=1}^{k-1} \sum_{n=m+1}^k \gamma_{w,mn} r_{w,mn} \quad (2.59)$$

Together, equations 2.57 and 2.59 lead to the conclusion that

$$\boldsymbol{\alpha}_y^T \boldsymbol{\Gamma}_{j,k} \mathbf{W}_{j,k-1} \boldsymbol{\Gamma}_{j,k}^T \boldsymbol{\alpha}_y = \sum_{m=1}^k \gamma_{w,mm} r_{w,mm} + 2 \sum_{m=1}^{k-1} \sum_{n=m+1}^k \gamma_{w,mn} r_{w,mn} \quad (2.60)$$

The same procedure can be applied to the second summation on the right hand side of equation 2.53, resulting in the following equation for  $\sigma_{y,k}^2$

$$\begin{aligned} \sigma_{y,k}^2 &= \boldsymbol{\alpha}_y^T \boldsymbol{\Phi}_k \mathbf{P}_0^- \boldsymbol{\Phi}_k^T \boldsymbol{\alpha}_y + \sum_{j=1}^{n_w} \left[ \sum_{m=1}^k \gamma_{w_j,mm} r_{w_j,mm} + 2 \sum_{m=1}^{k-1} \sum_{n=m+1}^k \gamma_{w_j,mn} r_{w_j,mn} \right] \\ &+ \sum_{l=1}^{n_v} \left[ \sum_{m=1}^{k+1} \gamma_{v_l,mm} r_{v_l,mm} + 2 \sum_{m=1}^k \sum_{n=m+1}^{k+1} \gamma_{v_l,mn} r_{v_l,mn} \right] \end{aligned} \quad (2.61)$$

**2.5.1 Covariance Initialization.** The initial covariance matrix  $\mathbf{P}_0^-$  is given by

$$\mathbf{P}_0^- = \begin{bmatrix} \mathbf{P}_{\theta,0}^- & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & E[\hat{\boldsymbol{\xi}}_0^- (\hat{\boldsymbol{\xi}}_0^-)^T] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & E[\hat{\boldsymbol{\eta}}_0^- (\hat{\boldsymbol{\eta}}_0^-)^T] \end{bmatrix} \quad (2.62)$$

where  $\mathbf{P}_{\theta,0}^- = E[\boldsymbol{\varepsilon}_{\theta,0}^- (\boldsymbol{\varepsilon}_{\theta,0}^-)^T]$  is the  $n_\theta \times n_\theta$  covariance matrix of the initial estimate error vector,  $\boldsymbol{\varepsilon}_{\theta,0}^-$ .

Both  $\hat{\boldsymbol{\xi}}_0^-$  and  $\hat{\boldsymbol{\eta}}_0^-$  are estimates of the initial state vectors  $\boldsymbol{\xi}_0$  and  $\boldsymbol{\eta}_0$ . Recall that  $\boldsymbol{\xi}$  and  $\boldsymbol{\eta}$  were added to model the colored nature of  $\boldsymbol{v}$  and  $\boldsymbol{w}$ , respectively. Given that  $\boldsymbol{v}$  and  $\boldsymbol{w}$  are zero-mean Gaussian random vectors, the only estimates of  $\boldsymbol{\xi}_0$  and  $\boldsymbol{\eta}_0$  consistent with the zero-mean Gaussian model are  $\hat{\boldsymbol{\xi}}_0^- = \mathbf{0}$  and  $\hat{\boldsymbol{\eta}}_0^- = \mathbf{0}$ . The initial state estimate is a deterministic vector, which leads to the conclusion that  $E[\hat{\boldsymbol{\xi}}_0^- (\hat{\boldsymbol{\xi}}_0^-)^T] = \hat{\boldsymbol{\xi}}_0^- (\hat{\boldsymbol{\xi}}_0^-)^T = \mathbf{0}$  and  $E[\hat{\boldsymbol{\eta}}_0^- (\hat{\boldsymbol{\eta}}_0^-)^T] = \hat{\boldsymbol{\eta}}_0^- (\hat{\boldsymbol{\eta}}_0^-)^T = \mathbf{0}$ . Substituting these expressions into equation 2.62 results in the initial covariance matrix

$$\mathbf{P}_0^- = \begin{bmatrix} \mathbf{P}_{\theta,0}^- & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (2.63)$$

Equation 2.61 expresses the estimate error variance in terms of the autocorrelation function values,  $r_{v_i,kl}$  and  $r_{w_i,kl}$ . When  $r_{v_i,kl}$  and  $r_{w_i,kl}$  are precisely known, they are simply inserted into equation 2.61 to compute  $\sigma_{y,k}^2$  which is subsequently used to compute integrity risk. However, when  $r_{v_i,kl}$  and  $r_{w_i,kl}$  are unknown,  $\sigma_{y,k}^2$  (and consequently the integrity risk  $I_y$  in equation 2.21) is also unknown. In this case it is necessary to produce a variance  $\bar{\sigma}_{y,k}^2$  that upper bounds  $\sigma_{y,k}^2$  so that the computed integrity risk  $\bar{I}_y$  always upper bounds the true risk. Before the variance bound can be constructed, an uncertainty structure on  $r_{v_i,kl}$  and  $r_{w_i,kl}$  must be specified.

## 2.6 Variance Bounding with Autocorrelation Uncertainty

This dissertation introduces the bounded uncertainty model

$$a_{v_l, mn} \leq r_{v_l, mn} \leq b_{v_l, mn} \quad , \quad l=1, \dots, n_v \quad (2.64)$$

$$a_{w_j, mn} \leq r_{w_j, mn} \leq b_{w_j, mn} \quad , \quad j=1, \dots, n_w \quad (2.65)$$

Given this uncertainty structure, the variance upper bound  $\bar{\sigma}_{y,k}^2$  can be constructed from equation 2.61 by setting  $r_{v_l, mn}$  and  $r_{w_j, mn}$  equal to their lower bound value whenever  $\gamma_{v_l, mn} < 0$  and  $\gamma_{w_j, mn} < 0$ , respectively, and by setting  $r_{v_l, mn}$  and  $r_{w_j, mn}$  equal to their upper bound value whenever  $\gamma_{v_l, mn} \geq 0$  and  $\gamma_{w_j, mn} \geq 0$ , respectively. Mathematically, the variance bound can be stated as

$$\begin{aligned} \bar{\sigma}_{y,k}^2 = & \alpha_y^T \Phi_k \mathbf{P}_0^- \Phi_k^T \alpha_y + \sum_{j=1}^{n_w} \left[ \sum_{m=1}^k \gamma_{w_j, mm} \bar{r}_{w_j, mm} + 2 \sum_{m=1}^{k-1} \sum_{n=m+1}^k \gamma_{w_j, mn} \bar{r}_{w_j, mn} \right] \\ & + \sum_{l=1}^{n_v} \left[ \sum_{m=1}^{k+1} \gamma_{v_l, mm} \bar{r}_{v_l, mm} + 2 \sum_{m=1}^k \sum_{n=m+1}^{k+1} \gamma_{v_l, mn} \bar{r}_{v_l, mn} \right] \end{aligned} \quad (2.66)$$

where  $\bar{r}_{v_l, mn}$  and  $\bar{r}_{w_j, mn}$  are defined as

$$\bar{r}_{w_j, mn} = \begin{cases} a_{w_j, mn} & , \quad \gamma_{w_j, mn} < 0 \\ b_{w_j, mn} & , \quad \gamma_{w_j, mn} \geq 0 \end{cases} \quad \text{and} \quad \bar{r}_{v_l, mn} = \begin{cases} a_{v_l, mn} & , \quad \gamma_{v_l, mn} < 0 \\ b_{v_l, mn} & , \quad \gamma_{v_l, mn} \geq 0 \end{cases} \quad (2.67)$$

**2.6.1 Remarks.** The estimate error variance given in equation 2.61 was derived for the general case where  $\{v_l\}$ ,  $l=1, \dots, n_v$  and  $\{w_j\}$ ,  $j=1, \dots, n_w$  are non-stationary, i.e.,  $r_{v_l, mn}$  and  $r_{w_j, mn}$  depend on the two time indices  $m$  and  $n$ . In certain applications, additional knowledge about the autocorrelation functions might be available. For instance, it may be known that the autocorrelation functions are wide sense stationary, in

which case  $r_{v_l, mn}$  and  $r_{w_j, mn}$  only depend on  $m-n$ . Under this scenario, equation 2.61 can be re-expressed as a linear combination of  $r_{v_l, 0}, \dots, r_{v_l, k}$  and  $r_{w_j, 0}, \dots, r_{w_j, (k-1)}$  prior to forming the variance bound. This special case will appear in Section 2.7.

Upper bounding the estimate error variance relies on the ability to upper and lower bound the autocorrelation functions associated with each measurement noise component and disturbance input. There are two ways to obtain the autocorrelation bounds. First, it might be possible to determine the bounds analytically based on a physical understanding of what is causing the time correlation. If it is not possible or is intractable to use a physical approach, then the bounds must be determined through experimental means.

## 2.7 One-Dimensional Estimation Application

The algorithms developed above will now be applied to the estimation problem shown in Figure 2.1.

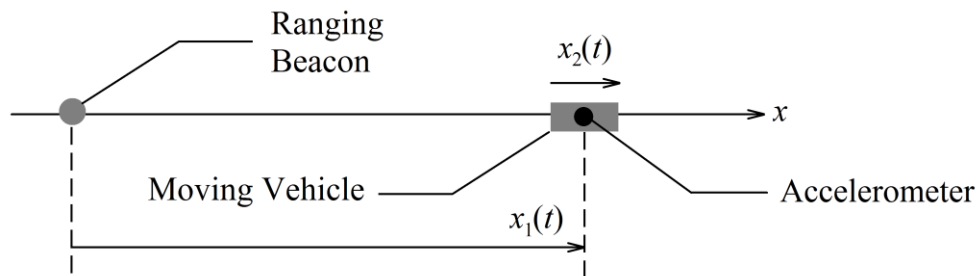


Figure 2.1. One-Dimensional Estimation Problem

The position  $x_1(t)$  and velocity  $x_2(t)$  are estimated using an accelerometer attached to the vehicle and a ranging beacon located at the origin. Assume that the vehicle is moving in a horizontal, inertial reference frame.

**2.7.1 Measurement and State Dynamic Models.** The measurement model for the ranging beacon is given by

$$z_k = x_{1,k} + v_k \quad (2.68)$$

where  $z_k$  is the beacon measurement at time index  $k$ ,  $x_{1,k}$  is the position of the vehicle at time index  $k$  and  $v_k$  is the ranging measurement noise at time index  $k$ .

Continuous-time dynamic models for position and velocity are given by

$$\dot{x}_1(t) = x_2(t) \quad (2.69)$$

$$\dot{x}_2(t) = a(t) \quad (2.70)$$

where  $a(t) \in \mathbb{R}$  is the vehicle's inertial acceleration sensed by the accelerometer.

Accelerometer measurements are output in the form of a *velocity increment*, which is the integral of acceleration over a given time interval. They operate in this fashion because the process of integration allows them to capture changes in acceleration over the sampling interval, which is especially important for applications with high vehicle dynamics [Jek01].

In addition to integrating vehicle acceleration, the accelerometer will also integrate random disturbances, denoted by  $\beta(t)$ . Therefore, the signal being integrated can be expressed as  $\tilde{a}(t) = a(t) + \beta(t)$ . Using this relation, the velocity dynamic model in equation 2.70 can also be written as

$$\dot{x}_2(t) = \tilde{a}(t) - \beta(t) \quad (2.71)$$

**2.7.2 Conversion to Discrete-Time.** In Section 2.1, linear measurement and state dynamic models were introduced. Namely,

$$\mathbf{z}_k = \mathbf{H}_{\theta,k} \boldsymbol{\theta}_k + \mathbf{J}_{\nu,k} \boldsymbol{\nu}_k \quad (2.72)$$

$$\boldsymbol{\theta}_{k+1} = \mathbf{F}_{\theta,k} \boldsymbol{\theta}_k + \mathbf{G}_{\theta,k} \mathbf{w}_k \quad (2.73)$$

The ranging beacon measurement model in equation 2.68 is already in the format of equation 2.72 with  $\mathbf{H}_{\theta,k} = [1 \ 0]$ ,  $\boldsymbol{\theta}_k = [x_{1,k} \ x_{2,k}]^T$  and  $\mathbf{J}_{\nu,k} = 1$ . However, the dynamic models in equations 2.69 and 2.71 are defined in continuous-time. These two equations will now be converted to discrete-time form.

Integrating both sides of equation 2.71 from time index  $k$  to time index  $k + 1$  yields

$$x_{2,k+1} = x_{2,k} + \int_k^{k+1} [\tilde{a}(\tau) - \beta(\tau)] d\tau \quad (2.74)$$

Making the definitions  $\Delta v_k = \int_k^{k+1} \tilde{a}(\tau) d\tau$  and  $w_k = \int_k^{k+1} \beta(\tau) d\tau$  allows equation 2.74

to be written in the form

$$x_{2,k+1} = x_{2,k} + \Delta v_k - w_k \quad (2.75)$$

where  $\Delta v_k$  is the accelerometer measurement (i.e., the velocity increment) and  $w_k$  is the associated accelerometer measurement noise.

Similarly, integrating both sides of equation 2.69 from time index  $k$  to time index  $k + 1$  results in

$$x_{1,k+1} = x_{1,k} + \int_k^{k+1} x_2(\tau) d\tau \quad (2.76)$$

In order to perform the integration, a functional expression for  $x_2(t)$  is required. Assuming that  $a(t)$  is approximately constant over the integration interval allows equation 2.70 to be written as  $x_2(t) \approx x_{2,k} + a(t - t_k)$ ,  $t \in [t_k, t_{k+1}]$ . Inserting this expression into equation 2.76 and performing the integration yields

$$x_{1,k+1} = x_{1,k} + x_{2,k} \Delta t + \frac{1}{2} a \Delta t^2 \quad , \quad \Delta t = t_{k+1} - t_k \quad (2.77)$$

Over the interval  $[t_k, t_{k+1}]$ ,  $a \approx (x_{2,k+1} - x_{2,k}) / \Delta t$ . Replacing  $x_{2,k+1}$  with the right hand side of equation 2.75 results in the approximation  $a \approx (\Delta v_k - w_k) / \Delta t$ . Substituting this relation into equation 2.77 yields the discrete-time position dynamic model

$$x_{1,k+1} = x_{1,k} + x_{2,k} \Delta t + \frac{\Delta t}{2} \Delta v_k - \frac{\Delta t}{2} w_k \quad (2.78)$$

Equations 2.75 and 2.78 can be written in the state space form

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \begin{bmatrix} \Delta t / 2 \\ 1 \end{bmatrix} \Delta v_k - \begin{bmatrix} \Delta t / 2 \\ 1 \end{bmatrix} w_k \quad (2.79)$$

which can finally be expressed as

$$\boldsymbol{\theta}_{k+1} = \mathbf{F}_{\theta,k} \boldsymbol{\theta}_k + \mathbf{G}_{v,k} \Delta v_k + \mathbf{G}_{\theta,k} w_k \quad (2.80)$$

The dynamic model in equation 2.80 is slightly different from equation 2.73 due to the presence of the term  $\mathbf{G}_{v,k} \Delta v_k$ . However, this vector is entirely known and therefore its presence does not impact the estimate error covariance matrix.

**2.7.3 Discrete-time Filter Model.** Suppose that  $\nu_k$  and  $w_k$  are time-correlated, and that state augmentation is used to define the filter model. While this step is straightforward for the measurement model, it is difficult to perform state augmentation directly on equation 2.80 because  $w_k$  is an integrated random process. Therefore, the state-augmented dynamic model will first be defined in continuous-time, followed by a conversion to discrete-time.

For this example,  $\nu_k$  can be decomposed as  $\nu_k = \eta_k + r_k$ , where  $\{r\}$  is a zero-mean, white Gaussian random process with variance  $\sigma_r^2$  and  $\{\eta\}$  is a first order Gauss-Markov process. The continuous time dynamic model of  $\eta$  is given by

$$\dot{\eta}(t) = -\frac{1}{\tau_\eta} \eta(t) + q_\eta(t) \quad , \quad \eta(0) \sim N(0, \sigma_\eta^2) \quad (2.81)$$

where  $\tau_\eta$  is the time constant of the Gauss-Markov process,  $\sigma_\eta^2$  is the variance of the Gauss-Markov process and  $\{q_\eta\}$  is a zero-mean, white Gaussian random process with power spectral density  $Q_\eta = 2\sigma_\eta^2 / \tau_\eta$ .

Similarly, the disturbance input  $\beta(t)$  can be decomposed as  $\beta(t) = \xi(t) + q_\theta(t)$ , where  $\{q_\theta\}$  is a zero-mean, white Gaussian random process with power spectral density  $Q_\theta$  and  $\{\xi\}$  is a first-order Gauss-Markov process governed by the dynamic model

$$\dot{\xi}(t) = -\frac{1}{\tau_\xi} \xi(t) + q_\xi(t) \quad , \quad \xi(0) \sim N(0, \sigma_\xi^2) \quad (2.82)$$

where  $\tau_\xi$  is the time constant of the Gauss-Markov process,  $\sigma_\xi^2$  is the variance of the Gauss-Markov process and  $\{q_\xi\}$  is a zero-mean, white Gaussian noise process with power spectral density  $Q_\xi = 2\sigma_\xi^2/\tau_\xi$ .

Collecting equations 2.69, 2.71, 2.81 and 2.82 into one matrix-vector equation results in the state-augmented dynamic model

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{\xi}(t) \\ \dot{\eta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & -1/\tau_\xi & 0 \\ 0 & 0 & 0 & -1/\tau_\eta \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \xi(t) \\ \eta(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tilde{a}(t) + \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_\theta(t) \\ q_\xi(t) \\ q_\eta(t) \end{bmatrix} \quad (2.83)$$

where the substitution  $\beta(t) = \xi(t) + q_\theta(t)$  has been made in equation 2.71.

Well known methods exist to convert equation 2.83 to the discrete-time form  $\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k \Delta v_k + \mathbf{q}_k$ . Expressions for  $\mathbf{F}_k$  and  $\mathbf{G}_k$  can be found in [Ste94], and the covariance matrix of  $\mathbf{q}_k$  can be determined using the Van Loan algorithm [Van78].

**2.7.4 Autocorrelation Bound Determination.** The autocorrelation bounds for  $v_k$  and  $\beta(t)$  are defined using the same mathematical structure as the filter model, i.e., they are based on the sum of a white Gaussian noise process and a first order Gauss-Markov process. The parameters defining these bounds are specified in Table 2.1.

Table 2.1. Autocorrelation Bound Parameters

Ranging Beacon		Accelerometer	
Lower Bound	Upper Bound	Lower Bound	Upper Bound
$\sigma_r = 0.25$ m	$\sigma_r = 0.50$ m	$Q_\theta^{1/2} = 5$ $\mu\text{g} \cdot \text{s}^{1/2}$	$Q_\theta^{1/2} = 10$ $\mu\text{g} \cdot \text{s}^{1/2}$
$\sigma_\eta = 0.75$ m	$\sigma_\eta = 1.0$ m	$\sigma_\xi = 30$ $\mu\text{g}$	$\sigma_\xi = 50$ $\mu\text{g}$
$\tau_\eta = 50$ sec	$\tau_\eta = 200$ sec	$\tau_\xi = 50$ sec	$\tau_\xi = 200$ sec

In what follows,  $r_{v_i, mn}$  and  $r_{w_j, mn}$  will be replaced by  $r_{v, mn}$  and  $r_{w, mn}$ , respectively, because  $v$  and  $w$  are scalars for this example. It is shown in Appendix B that the autocorrelation function of  $w$  is given by

$$r_{w, mn} = \begin{cases} 2(\sigma_\xi \tau_\xi)^2 \left[ \frac{\Delta t}{\tau_\xi} - 1 + \exp\left(-\frac{\Delta t}{\tau_\xi}\right) \right] + Q_\theta \Delta t & , m = n \\ (\sigma_\xi \tau_\xi)^2 \exp\left(-\frac{|m-n| \Delta t}{\tau_\xi}\right) \left[ 1 - \exp\left(-\frac{\Delta t}{\tau_\xi}\right) \right] \left[ \exp\left(\frac{\Delta t}{\tau_\xi}\right) - 1 \right] & , m \neq n \end{cases} \quad (2.84)$$

The autocorrelation function of  $v$  is more straightforward. For the Gauss-Markov component of the beacon measurement noise, the autocorrelation function is given by [Gel74]

$$r_{v, mn} = \sigma_\eta^2 \exp\left(-\frac{|m-n| \Delta t}{\tau_\eta}\right) \quad (2.85)$$

The white noise component of  $v$  has the autocorrelation function

$$r_{r, mn} = \begin{cases} \sigma_r^2 & , m = n \\ 0 & , m \neq n \end{cases} \quad (2.86)$$

When equations 2.85 and 2.86 are combined, the autocorrelation function of  $v$  is described by the equation

$$r_{v,mn} = \begin{cases} \sigma_\eta^2 + \sigma_r^2 & , \quad m = n \\ \sigma_\eta^2 \exp\left(-\frac{|m-n|\Delta t}{\tau_\eta}\right) & , \quad m \neq n \end{cases} \quad (2.87)$$

Notice that both  $r_{v,mn}$  and  $r_{w,mn}$  are stationary autocorrelation functions, i.e., they are only a function of the time difference,  $m-n$ . The autocorrelation bounding functions are obtained by substituting the upper and lower bound values from Table 2.1 into equations 2.84 and 2.87.

Design values of  $\tau_\xi$ ,  $\tau_\eta$ ,  $\sigma_\xi^2$ ,  $\sigma_\eta^2$ ,  $Q_\theta$  and  $\sigma_r^2$  must be specified in order to define the Kalman filter. Typically, the upper bound values are used, and the same approach is adopted in this dissertation. The rationale behind this choice is that measurements with highly correlated errors do not provide as much new information as measurements whose errors are less correlated. This in turn should produce a larger estimate error variance. However, it will be shown shortly that these qualitative arguments do not always work.

**2.7.5 Covariance Initialization.** The last step to complete before conducting the simulation is to specify the initial covariance matrices,  $\hat{\mathbf{P}}_0^-$  and  $\mathbf{P}_0^-$ . For this example,  $\hat{\mathbf{P}}_0^-$  is set equal to

$$\hat{\mathbf{P}}_0^- = \begin{bmatrix} (10 \text{ m})^2 & 0 & 0 & 0 \\ 0 & (1 \text{ m/s})^2 & 0 & 0 \\ 0 & 0 & (50 \mu\text{g})^2 & 0 \\ 0 & 0 & 0 & (1 \text{ m})^2 \end{bmatrix} \quad (2.88)$$

Recall from equation 2.63 that the only non-zero elements of  $\mathbf{P}_0^-$  are those belonging to the upper left  $n_\theta \times n_\theta$  matrix which are identical to the values in the upper left  $n_\theta \times n_\theta$  matrix of  $\hat{\mathbf{P}}_0^-$ . Given that  $n_\theta = 2$ , this results in the following form for  $\mathbf{P}_0^-$

$$\mathbf{P}_0^- = \begin{bmatrix} (10 \text{ m})^2 & 0 & 0 & 0 \\ 0 & (1 \text{ m/s})^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.89)$$

The ranging beacon and accelerometer measurement sampling intervals are equal to 5 sec (i.e.  $\Delta t = 5$  sec). Sixty measurement epochs are simulated, for a total simulation time scale of 300 seconds. In this case, the autocorrelation bounding functions can be represented graphically as in Figures 2.2 and 2.3.

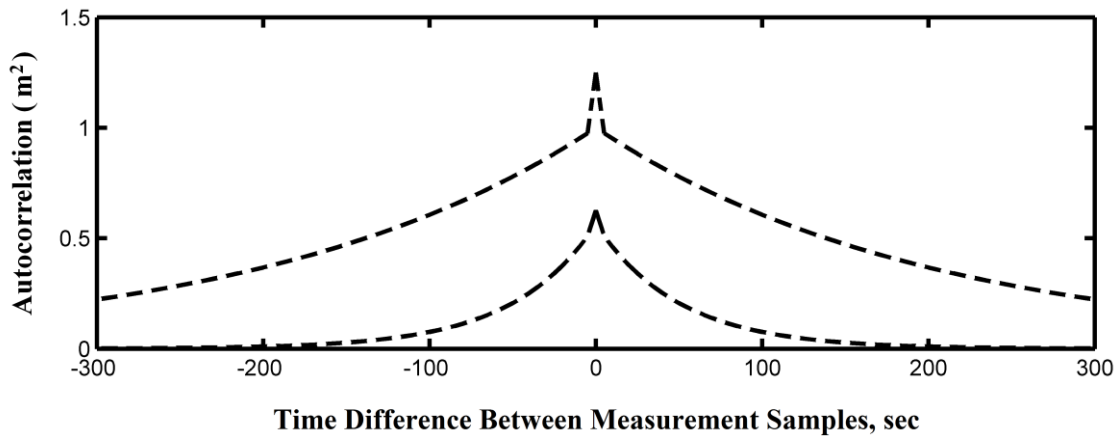


Figure 2.2. Ranging Beacon Measurement Error Autocorrelation Bounds

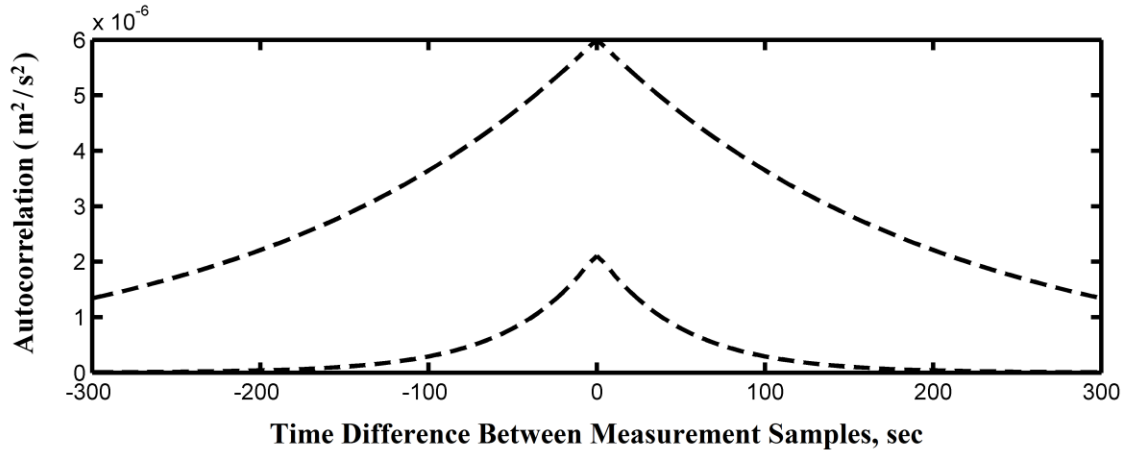


Figure 2.3. Accelerometer Measurement Error Autocorrelation Bounds

**2.7.6 Position State Simulation Results.** For the position state, the bounding variance  $\bar{\sigma}_{x_1}^2$  is determined using the coefficient vector  $\alpha_{x_1} = [1 \ 0 \ 0 \ 0]^T$ . Since the *a posteriori* state estimate is the best available estimate at any given time index, only the posterior bound will be considered. Figures 2.4 and 2.5 show the ranging beacon and accelerometer error autocorrelation functions producing the variance bound at an elapsed time of 300 seconds.

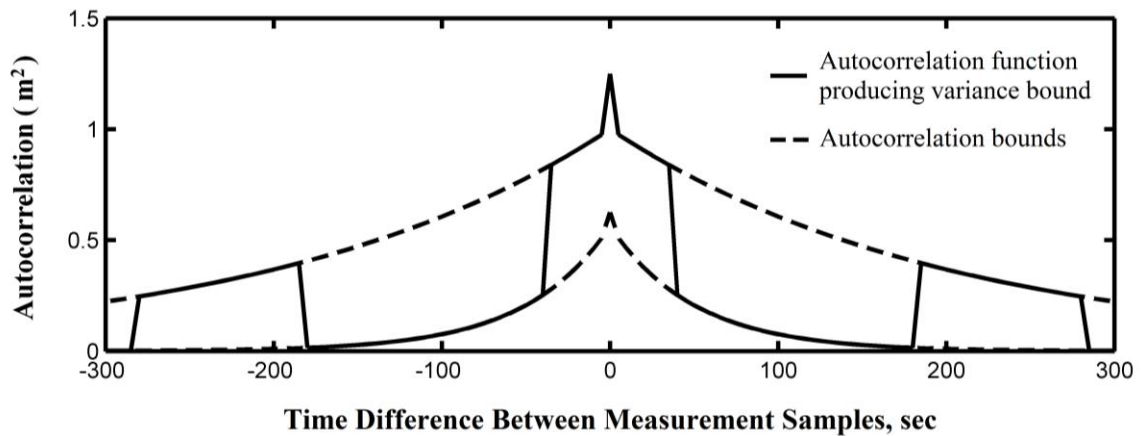


Figure 2.4. Beacon ACF Producing Position State Variance Bound

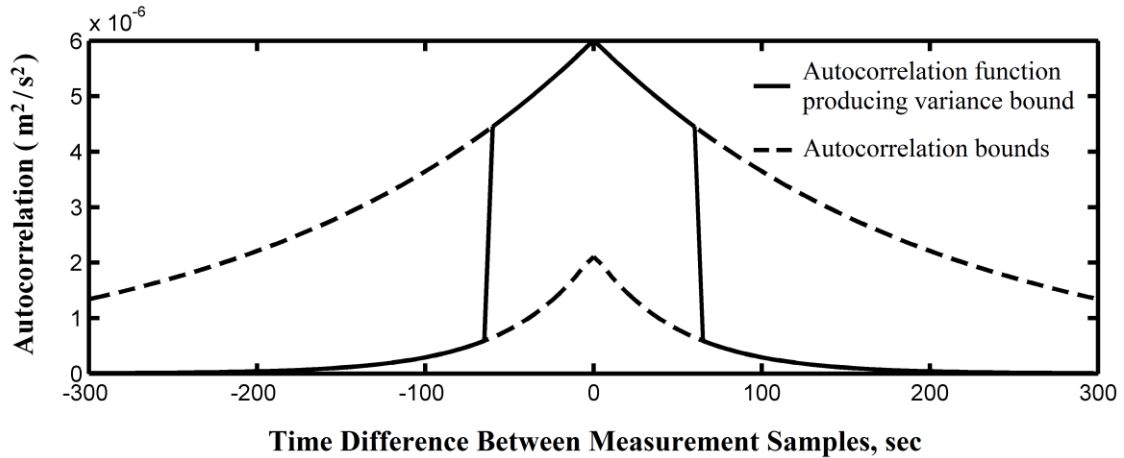


Figure 2.5. Accelerometer ACF Producing Position State Variance Bound

The autocorrelation functions producing the position estimate error variance bound are characterized by sharp transitions between the upper and lower bounding functions. Where the transitions occur as well as their frequency is different for the ranging beacon and the accelerometer. Physically, the autocorrelation functions in Figures 2.4 and 2.5 represent the worst-case scenario. That is, if the ranging beacon and accelerometer errors had the autocorrelation functions shown in Figures 2.4 and 2.5, respectively, then the position estimate error variance would attain its largest value. Therefore, it is not necessarily the upper bound autocorrelation function that results in the worst-case estimate error variance.

**2.7.7 Velocity State Simulation Results.** For the velocity state, the bounding variance  $\bar{\sigma}_{x_2}^2$  is determined using the coefficient vector  $\alpha_{x_2} = [0 \ 1 \ 0 \ 0]^T$ . Figures 2.6 and 2.7 show the ranging beacon and accelerometer error autocorrelation functions producing the variance bound at an elapsed time of 300 seconds.

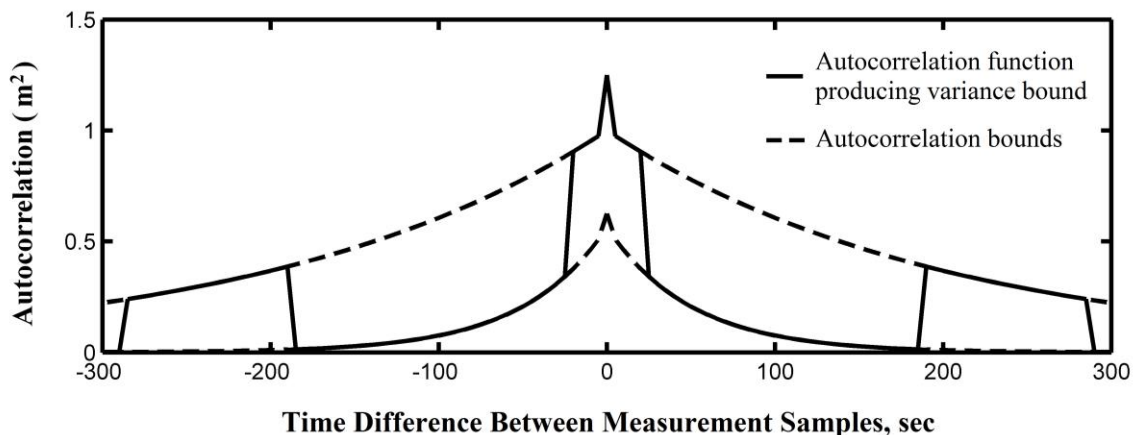


Figure 2.6. Beacon ACF Producing Velocity State Variance Bound

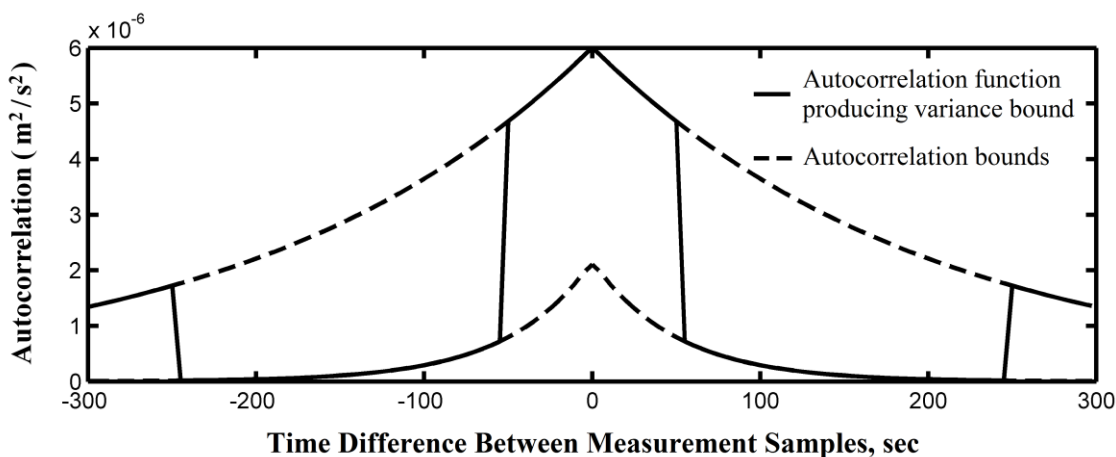


Figure 2.7. Accelerometer ACF Producing Velocity State Variance Bound

Figure 2.6 shows that the beacon measurement error autocorrelation function producing the velocity bound is nearly identical to that shown in Figure 2.4, except that the transition points occur at slightly different times. The accelerometer error autocorrelation function shown in Figure 2.7 has two transitions and is distinctly different from the function shown in Figure 2.5. Therefore, the worst-case autocorrelation functions for one state are not necessarily the worst-case for another state. If integrity risk

needs to be quantified for multiple states, the bounding algorithm must be applied separately for each state of interest.

The methods developed in this chapter are effective at bounding integrity risk for linear estimation applications that use the Kalman filter. However, there are situations where it is advantageous or even necessary to process sequential measurements simultaneously in a batch estimator. For example, when the measurement noise and disturbance inputs cannot be described by a finite state model, state vector estimation can only be properly implemented using batch algorithms. In nonlinear estimation applications, the convergence properties of batch estimators make them more suitable than sequential algorithms like the extended Kalman filter. Therefore, it is beneficial to extend the bounding methods developed in this chapter to batch estimation.

## CHAPTER 3

### EXTENSIONS TO BATCH ESTIMATION

This chapter considers integrity risk bounding for estimation applications that use batch algorithms for state estimation. In Section 3.1, linearized measurement and state dynamic models are derived, which are merged into one batch measurement equation in Section 3.2. A new, generalized covariance matrix for batch WLS is defined in Section 3.3 that explicitly accounts for the fact that the weighting matrix is only an approximation of the true batch measurement noise covariance matrix. In Section 3.4, the estimate error variance is obtained from the generalized covariance matrix and is expressed directly in terms of measurement noise and disturbance input autocorrelation function values using the results from Section 2.5. An upper bound on the estimate error variance is defined in Section 3.5 for the bounded autocorrelation uncertainty structure, which is subsequently used to compute an upper bound on integrity risk. Section 3.6 implements the bounding algorithms developed in this chapter for a two-dimensional nonlinear estimation problem.

#### 3.1 Linearized Measurement and State Dynamic Models

Consider the measurement model

$$\zeta_k = \mathbf{h}(\boldsymbol{\theta}_k, \mathbf{v}_k) \quad (3.1)$$

where  $\zeta_k \in \mathbb{R}^{n_\zeta}$  is the measurement vector,  $\mathbf{h}(\bullet) \in \mathbb{R}^{n_\zeta}$  is a known, nonlinear vector function,  $\boldsymbol{\theta}_k \in \mathbb{R}^{n_\theta}$  is the state vector and  $\mathbf{v}_k \in \mathbb{R}^{n_\nu}$  is a zero-mean Gaussian random vector.

Expanding equation 3.1 in a first order Taylor series yields the linearized measurement model

$$\zeta_k = \mathbf{h}(\boldsymbol{\theta}_k^*, \mathbf{v}_k^*) + \left. \frac{\partial \mathbf{h}}{\partial \boldsymbol{\theta}_k} \right|_{\boldsymbol{\theta}_k^*, \mathbf{v}_k^*} (\boldsymbol{\theta}_k - \boldsymbol{\theta}_k^*) + \left. \frac{\partial \mathbf{h}}{\partial \mathbf{v}_k} \right|_{\boldsymbol{\theta}_k^*, \mathbf{v}_k^*} (\mathbf{v}_k - \mathbf{v}_k^*) \quad (3.2)$$

where  $\boldsymbol{\theta}_k^*$  is a nominal value of  $\boldsymbol{\theta}_k$  and  $\mathbf{v}_k^*$  is a nominal value of  $\mathbf{v}_k$ . Given that  $\mathbf{v}_k$  is zero-mean, it is simplest to choose  $\mathbf{v}_k^* = \mathbf{0}$ .

Defining  $\mathbf{H}_{\theta,k} \in \mathbb{R}^{n_\zeta \times n_\theta}$ ,  $\mathbf{J}_{v,k} \in \mathbb{R}^{n_\zeta \times n_v}$  and  $\mathbf{z}_k \in \mathbb{R}^{n_\zeta}$  as

$$\mathbf{H}_{\theta,k} = \left. \frac{\partial \mathbf{h}}{\partial \boldsymbol{\theta}_k} \right|_{\boldsymbol{\theta}_k^*, \mathbf{v}_k^*} \quad (3.3)$$

$$\mathbf{J}_{v,k} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{v}_k} \right|_{\boldsymbol{\theta}_k^*, \mathbf{v}_k^*} \quad (3.4)$$

$$\mathbf{z}_k = \zeta_k - \mathbf{h}(\boldsymbol{\theta}_k^*, \mathbf{v}_k^*) + \mathbf{H}_{\theta,k} \boldsymbol{\theta}_k^* \quad (3.5)$$

allows equation 3.2 to be written more compactly as

$$\mathbf{z}_k = \mathbf{H}_{\theta,k} \boldsymbol{\theta}_k + \mathbf{J}_{v,k} \mathbf{v}_k \quad (3.6)$$

Notice that if  $n_\zeta < n_\theta$ , then  $\mathbf{z}_k$  will not provide sufficient information to estimate  $\boldsymbol{\theta}_k$ . In this case, it is necessary to use measurements collected over time to estimate the state vector. At each measurement epoch, the total number of measurements available for estimation increases by  $n_\zeta$ , but the total number of states also increases by  $n_\theta$ . Therefore, in order for the measurement batch to provide sufficient information for estimation, a dynamic model must also be available for  $\boldsymbol{\theta}$ .

Suppose that the dynamic model of a subset ( $\boldsymbol{\eta}$ ) of  $\boldsymbol{\theta}$  is known. Then equation 3.6 can be partitioned as

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{H}_{\xi,k} & \mathbf{H}_{\eta,k} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_k \\ \boldsymbol{\eta}_k \end{bmatrix} + \mathbf{J}_{v,k} \mathbf{v}_k \quad (3.7)$$

where  $\mathbf{H}_{\xi,k} \in \mathbb{R}^{n_\zeta \times n_\xi}$ ,  $\mathbf{H}_{\eta,k} \in \mathbb{R}^{n_\zeta \times n_\eta}$ ,  $\boldsymbol{\xi}_k \in \mathbb{R}^{n_\xi}$  and  $\boldsymbol{\eta}_k \in \mathbb{R}^{n_\eta}$ , with  $n_\eta \geq (n_\theta - n_\zeta)$ .

The dynamic model of  $\boldsymbol{\eta}$  is given by

$$\boldsymbol{\eta}_{k+1} = \mathbf{f}(\boldsymbol{\eta}_k, \mathbf{w}_k) \quad (3.8)$$

where  $\mathbf{f}(\bullet) \in \mathbb{R}^{n_\eta}$  is a known, nonlinear vector function and  $\mathbf{w}_k \in \mathbb{R}^{n_w}$  is a zero-mean Gaussian random vector.

Expanding equation 3.8 in a first order Taylor series results in the linear dynamic model

$$\boldsymbol{\eta}_{k+1} = \mathbf{f}(\boldsymbol{\eta}_k^*, \mathbf{w}_k^*) + \left. \frac{\partial \mathbf{f}}{\partial \boldsymbol{\eta}_k} \right|_{\boldsymbol{\eta}_k^*, \mathbf{w}_k^*} (\boldsymbol{\eta}_k - \boldsymbol{\eta}_k^*) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{w}_k} \right|_{\boldsymbol{\eta}_k^*, \mathbf{w}_k^*} (\mathbf{w}_k - \mathbf{w}_k^*) \quad (3.9)$$

where  $\boldsymbol{\eta}_k^*$  is a nominal value of  $\boldsymbol{\eta}_k$  and  $\mathbf{w}_k^*$  is a nominal value of  $\mathbf{w}_k$ , taken to be  $\mathbf{w}_k^* = \mathbf{0}$ .

Defining  $\mathbf{F}_{\eta,k} \in \mathbb{R}^{n_\eta \times n_\eta}$ ,  $\mathbf{G}_{\eta,k} \in \mathbb{R}^{n_\eta \times n_w}$  and  $\mathbf{u}_k \in \mathbb{R}^{n_\eta}$  as

$$\mathbf{F}_{\eta,k} = \left. \frac{\partial \mathbf{f}}{\partial \boldsymbol{\eta}_k} \right|_{\boldsymbol{\eta}_k^*, \mathbf{w}_k^*} \quad (3.10)$$

$$\mathbf{G}_{\eta,k} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{w}_k} \right|_{\boldsymbol{\eta}_k^*, \mathbf{w}_k^*} \quad (3.11)$$

$$\mathbf{u}_k = \mathbf{f}(\boldsymbol{\eta}_k^*, \mathbf{w}_k^*) - \mathbf{F}_{\eta,k} \boldsymbol{\eta}_k^* \quad (3.12)$$

allows equation 3.9 to be written more compactly as

$$\boldsymbol{\eta}_{k+1} = \mathbf{F}_{\eta,k} \boldsymbol{\eta}_k + \mathbf{u}_k + \mathbf{G}_{\eta,k} \mathbf{w}_k \quad (3.13)$$

Notice that  $\mathbf{u}_k$  is known because it is determined from the nominal values  $\boldsymbol{\eta}_k^*$  and  $\mathbf{w}_k^*$ .

Equations 3.7 and 3.13 describe a linear estimation problem in the state vector  $[\boldsymbol{\xi}_k^T \quad \boldsymbol{\eta}_k^T]^T$ . These two equations, collected over the time interval  $(t_1, t_Q)$ , can be combined into one batch measurement model that allows existing algorithms to be used for state vector estimation.

### 3.2 Batch Measurement Model and WLS Estimator

At time index  $k = 1$ , equations 3.7 and 3.13 can be written as

$$\mathbf{z}_1 = \mathbf{H}_{\xi,1} \boldsymbol{\xi}_1 + \mathbf{H}_{\eta,1} \boldsymbol{\eta}_1 + \mathbf{J}_{v,1} \mathbf{v}_1 \quad (3.14)$$

$$\boldsymbol{\eta}_2 = \mathbf{F}_{\eta,1} \boldsymbol{\eta}_1 + \mathbf{u}_1 + \mathbf{G}_{\eta,1} \mathbf{w}_1 \quad (3.15)$$

Solving equation 3.15 for  $\boldsymbol{\eta}_1$  results in  $\boldsymbol{\eta}_1 = \mathbf{F}_{\eta,1}^{-1} \boldsymbol{\eta}_2 - \mathbf{F}_{\eta,1}^{-1} \mathbf{u}_1 - \mathbf{F}_{\eta,1}^{-1} \mathbf{G}_{\eta,1} \mathbf{w}_1$ . Substituting this expression into equation 3.14 yields the measurement model

$$\mathbf{z}_1 = \mathbf{H}_{\xi,1} \boldsymbol{\xi}_1 + \mathbf{H}_{\eta,1} \mathbf{F}_{\eta,1}^{-1} \boldsymbol{\eta}_2 - \mathbf{H}_{\eta,1} \mathbf{F}_{\eta,1}^{-1} \mathbf{u}_1 - \mathbf{H}_{\eta,1} \mathbf{F}_{\eta,1}^{-1} \mathbf{G}_{\eta,1} \mathbf{w}_1 + \mathbf{J}_{v,1} \mathbf{v}_1 \quad (3.16)$$

The measurement and state dynamic models at  $k = 2$  are given by

$$\mathbf{z}_2 = \mathbf{H}_{\xi,2} \boldsymbol{\xi}_2 + \mathbf{H}_{\eta,2} \boldsymbol{\eta}_2 + \mathbf{J}_{v,2} \mathbf{v}_2 \quad (3.17)$$

$$\boldsymbol{\eta}_3 = \mathbf{F}_{\eta,2} \boldsymbol{\eta}_2 + \mathbf{u}_2 + \mathbf{G}_{\eta,2} \mathbf{w}_2 \quad (3.18)$$

Rearranging equation 3.18 into the form  $\boldsymbol{\eta}_2 = \mathbf{F}_{\eta,2}^{-1} \boldsymbol{\eta}_3 - \mathbf{F}_{\eta,2}^{-1} \mathbf{u}_2 - \mathbf{F}_{\eta,2}^{-1} \mathbf{G}_{\eta,2} \mathbf{w}_2$  and using this relation in equations 3.16 and 3.17 yield the measurement equations

$$\begin{aligned} \mathbf{z}_1 = & \mathbf{H}_{\xi,1} \boldsymbol{\xi}_1 + \mathbf{H}_{\eta,1} \mathbf{F}_{\eta,1}^{-1} \mathbf{F}_{\eta,2}^{-1} \boldsymbol{\eta}_3 - \mathbf{H}_{\eta,1} \mathbf{F}_{\eta,1}^{-1} \mathbf{F}_{\eta,2}^{-1} \mathbf{u}_2 - \mathbf{H}_{\eta,1} \mathbf{F}_{\eta,1}^{-1} \mathbf{F}_{\eta,2}^{-1} \mathbf{G}_{\eta,2} \mathbf{w}_2 \\ & - \mathbf{H}_{\eta,1} \mathbf{F}_{\eta,1}^{-1} \mathbf{u}_1 - \mathbf{H}_{\eta,1} \mathbf{F}_{\eta,1}^{-1} \mathbf{G}_{\eta,1} \mathbf{w}_1 + \mathbf{J}_{v,1} \mathbf{v}_1 \end{aligned} \quad (3.19)$$

$$\mathbf{z}_2 = \mathbf{H}_{\xi,2} \boldsymbol{\xi}_2 + \mathbf{H}_{\eta,2} \mathbf{F}_{\eta,2}^{-1} \boldsymbol{\eta}_3 - \mathbf{H}_{\eta,2} \mathbf{F}_{\eta,2}^{-1} \mathbf{u}_2 - \mathbf{H}_{\eta,2} \mathbf{F}_{\eta,2}^{-1} \mathbf{G}_{\eta,2} \mathbf{w}_2 + \mathbf{J}_{v,2} \mathbf{v}_2 \quad (3.20)$$

In general, the batch measurement model can be written as

$$\mathbf{z}_Q = \mathbf{H}_Q \mathbf{x}_Q + \mathbf{G}_Q \mathbf{u}_Q + \mathbf{J}_Q \mathbf{v}_Q \quad (3.21)$$

where

$$\mathbf{x}_Q = [\boldsymbol{\xi}_1^T \quad \boldsymbol{\xi}_2^T \quad \cdots \quad \boldsymbol{\xi}_q^T \quad \boldsymbol{\eta}_q^T]^T \quad (3.22)$$

$$\mathbf{u}_Q = [\mathbf{u}_1^T \quad \mathbf{u}_2^T \quad \cdots \quad \mathbf{u}_{q-1}^T]^T \quad (3.23)$$

$$\mathbf{v}_Q = [\mathbf{v}_1^T \quad \mathbf{v}_2^T \quad \cdots \quad \mathbf{v}_q^T \quad \mathbf{w}_1^T \quad \cdots \quad \mathbf{w}_{q-1}^T]^T \quad (3.24)$$

$$\mathbf{H}_Q = \begin{bmatrix} \mathbf{H}_{\xi,1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}_{\eta,1} (\mathbf{F}_{\eta,q-1} \cdots \mathbf{F}_{\eta,2} \mathbf{F}_{\eta,1})^{-1} \\ \mathbf{0} & \mathbf{H}_{\xi,2} & \cdots & \mathbf{0} & \mathbf{H}_{\eta,2} (\mathbf{F}_{\eta,q-1} \cdots \mathbf{F}_{\eta,2})^{-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{H}_{\xi,q} & \mathbf{H}_{\eta,q} \end{bmatrix} \quad (3.25)$$

$$\mathbf{G}_Q = \begin{bmatrix} -\mathbf{H}_{\eta,1} \mathbf{F}_{\eta,1}^{-1} & \cdots & -\mathbf{H}_{\eta,1} (\mathbf{F}_{\eta,q-1} \cdots \mathbf{F}_{\eta,1})^{-1} \\ \mathbf{0} & \cdots & -\mathbf{H}_{\eta,2} (\mathbf{F}_{\eta,q-1} \cdots \mathbf{F}_{\eta,2})^{-1} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \quad (3.26)$$

$$\mathbf{J}_Q = \begin{bmatrix} \mathbf{J}_{v,1} & \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{H}_{\eta,1} \mathbf{F}_{\eta,1}^{-1} \mathbf{G}_{\eta,1} & \cdots & -\mathbf{H}_{\eta,1} (\mathbf{F}_{\eta,q-1} \cdots \mathbf{F}_{\eta,1})^{-1} \mathbf{G}_{\eta,q-1} \\ \mathbf{0} & \mathbf{J}_{v,2} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{H}_{\eta,2} (\mathbf{F}_{\eta,q-1} \cdots \mathbf{F}_{\eta,2})^{-1} \mathbf{G}_{\eta,q-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{J}_{v,q} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \quad (3.27)$$

Defining  $n_x = qn_\xi + n_\eta$ ,  $n_z = qn_\zeta$ ,  $n_u = (q-1)n_\eta$  and  $n_v = qn_\nu + (q-1)n_w$ , then

$\mathbf{z}_Q \in \mathbb{R}^{n_z}$ ,  $\mathbf{H}_Q \in \mathbb{R}^{n_z \times n_x}$ ,  $\mathbf{x}_Q \in \mathbb{R}^{n_x}$ ,  $\mathbf{G}_Q \in \mathbb{R}^{n_z \times n_u}$ ,  $\mathbf{u}_Q \in \mathbb{R}^{n_u}$ ,  $\mathbf{J}_Q \in \mathbb{R}^{n_z \times n_v}$  and

$\mathbf{v}_Q \in \mathbb{R}^{n_v}$ . The capital letter  $Q$  will be used to indicate the size of the batch.

An estimate of  $\mathbf{x}_Q$  can be obtained using weighted least squares [Bro97]

$$\hat{\mathbf{x}}_Q = [\mathbf{H}_Q^T (\mathbf{J}_Q \mathbf{P}_{v,Q} \mathbf{J}_Q^T)^{-1} \mathbf{H}_Q]^{-1} \mathbf{H}_Q^T (\mathbf{J}_Q \mathbf{P}_{v,Q} \mathbf{J}_Q^T)^{-1} (\mathbf{z}_Q - \mathbf{G}_Q \mathbf{u}_Q) \quad (3.28)$$

where  $\mathbf{P}_{v,Q} = E[\mathbf{v}_Q \mathbf{v}_Q^T]$  is the  $n_v \times n_v$  measurement noise covariance matrix. Equations

2.3, 2.4 and 2.5 (restated below for convenience) are used to populate  $\mathbf{P}_{v,Q}$ .

$$E[v_{i,k} v_{j,l}] = r_{v_i,kl} \delta_{ij} \quad , \quad \begin{array}{l} i = 1, \dots, n_v \\ j = 1, \dots, n_v \end{array} \quad (3.29)$$

$$E[w_{i,k} w_{j,l}] = r_{w_i,kl} \delta_{ij} \quad , \quad \begin{array}{l} i = 1, \dots, n_w \\ j = 1, \dots, n_w \end{array} \quad (3.30)$$

$$E[w_{i,k} v_{j,l}] = 0 \quad , \quad \forall i, j, k, \text{ and } l \quad (3.31)$$

The WLS estimator is a linear, unbiased estimator. Given that  $\mathbf{v}_Q \sim N(\mathbf{0}, \mathbf{P}_{v,Q})$ ,

this implies that  $\boldsymbol{\varepsilon}_Q = \hat{\mathbf{x}}_Q - \mathbf{x}_Q$  is also a zero-mean Gaussian random vector, whose

probability density function is completely specified by the covariance matrix

$\mathbf{P}_\varrho = E[\boldsymbol{\varepsilon}_\varrho \boldsymbol{\varepsilon}_\varrho^T]$ . For  $y = \boldsymbol{\alpha}^T \mathbf{x}$ , where  $\boldsymbol{\alpha} \in \mathbb{R}^{n_x}$  is a known vector, the estimate error variance is given by  $\sigma_y^2 = \boldsymbol{\alpha}^T \mathbf{P}_\varrho \boldsymbol{\alpha}$ , which is used to compute integrity risk via the relation  $I_y = \text{erfc}[\ell_y (2\sigma_y^2)^{-1/2}]$ . Just as in the case of the Kalman filter, the estimate error covariance matrix is also a critical aspect of integrity risk assessment in batch estimation.

### 3.3 Generalized Covariance Matrix for Batch WLS

In what follows, equation 3.28 will be written in the form  $\hat{\mathbf{x}}_\varrho = \mathbf{S}_\varrho (\mathbf{z}_\varrho - \mathbf{G}_\varrho \mathbf{u}_\varrho)$ , where  $\mathbf{S}_\varrho \in \mathbb{R}^{n_x \times n_z}$  is given by  $\mathbf{S}_\varrho = [\mathbf{H}_\varrho^T (\mathbf{J}_\varrho \mathbf{P}_{v,\varrho} \mathbf{J}_\varrho^T)^{-1} \mathbf{H}_\varrho]^{-1} \mathbf{H}_\varrho^T (\mathbf{J}_\varrho \mathbf{P}_{v,\varrho} \mathbf{J}_\varrho^T)^{-1}$ . Notice that  $\mathbf{S}_\varrho$  is a function of the measurement noise and disturbance input autocorrelation function values comprising  $\mathbf{P}_{v,\varrho}$ . Therefore,  $\mathbf{S}_\varrho$  will be denoted by  $\mathbf{S}_\varrho(\mathbf{r})$ , where  $\mathbf{r} \in \mathbb{R}^{n_v}$  is a vector of autocorrelation function values. However,  $\mathbf{r}$  is an unknown vector, and the WLS estimator is formed using an approximate matrix,  $\hat{\mathbf{S}}_\varrho(\hat{\mathbf{r}})$ . That is,  $\hat{\mathbf{x}}_\varrho = \hat{\mathbf{S}}_\varrho(\hat{\mathbf{r}})(\mathbf{z}_\varrho - \mathbf{G}_\varrho \mathbf{u}_\varrho)$ , where  $\hat{\mathbf{r}}$  is based on the best available knowledge of the measurement noise and disturbance input autocorrelation functions.

To establish the estimate error vector associated with  $\hat{\mathbf{x}}_\varrho = \hat{\mathbf{S}}_\varrho(\hat{\mathbf{r}})(\mathbf{z}_\varrho - \mathbf{G}_\varrho \mathbf{u}_\varrho)$ , use equation 3.21 to replace  $(\mathbf{z}_\varrho - \mathbf{G}_\varrho \mathbf{u}_\varrho)$  with  $(\mathbf{H}_\varrho \mathbf{x}_\varrho + \mathbf{J}_\varrho \mathbf{v}_\varrho)$ , which results in

$$\hat{\mathbf{x}}_\varrho = \hat{\mathbf{S}}_\varrho(\hat{\mathbf{r}})[\mathbf{H}_\varrho \mathbf{x}_\varrho + \mathbf{J}_\varrho \mathbf{v}_\varrho] \quad (3.32)$$

Noting that  $\hat{\mathbf{S}}_\varrho(\hat{\mathbf{r}})\mathbf{H}_\varrho = \mathbf{I}$ , equation 3.32 can be simplified to

$$\hat{\mathbf{x}}_\varrho = \mathbf{x}_\varrho + \hat{\mathbf{S}}_\varrho(\hat{\mathbf{r}})\mathbf{J}_\varrho \mathbf{v}_\varrho \quad (3.33)$$

which can also be written as  $\boldsymbol{\varepsilon}_Q = \hat{\mathbf{S}}_Q(\hat{\mathbf{r}}) \mathbf{J}_Q \boldsymbol{\nu}_Q$ .

Evaluating  $E[\boldsymbol{\varepsilon}_Q \boldsymbol{\varepsilon}_Q^T]$  results in

$$\mathbf{P}_Q(\mathbf{r}, \hat{\mathbf{r}}) = E[\boldsymbol{\varepsilon}_Q \boldsymbol{\varepsilon}_Q^T] = \hat{\mathbf{S}}_Q(\hat{\mathbf{r}}) \mathbf{J}_Q \mathbf{P}_{\nu, Q}(\mathbf{r}) \mathbf{J}_Q^T \hat{\mathbf{S}}_Q^T(\hat{\mathbf{r}}) \quad (3.34)$$

where  $\mathbf{P}_Q(\mathbf{r}, \hat{\mathbf{r}}) \in \mathbb{R}^{n_x \times n_x}$  is the generalized covariance matrix of  $\boldsymbol{\varepsilon}_Q$  which accurately reflects the fact that the WLS estimator is using approximate autocorrelation function values.

The noise vector  $\boldsymbol{\nu}_Q$  in equation 3.24 is ordered according to time, but it can also be arranged according to measurement noise and disturbance input components by applying a linear transformation. That is

$$[\boldsymbol{\nu}_1^T \quad \boldsymbol{\nu}_2^T \quad \cdots \quad \boldsymbol{\nu}_q^T \quad \mathbf{w}_1^T \quad \cdots \quad \mathbf{w}_{q-1}^T]^T = \mathbf{T} [\bar{\mathbf{v}}_1^T \quad \bar{\mathbf{v}}_2^T \quad \cdots \quad \bar{\mathbf{v}}_{n_\nu}^T \quad \bar{\mathbf{w}}_1^T \quad \cdots \quad \bar{\mathbf{w}}_{n_w}^T]^T \quad (3.35)$$

$$\boldsymbol{\nu}_Q = \mathbf{T} \bar{\boldsymbol{\nu}}_Q \quad (3.36)$$

where  $\bar{\mathbf{v}}_i \in \mathbb{R}^q$  is a time series of the measurement error from sensor  $i$ ,  $\bar{\mathbf{w}}_i \in \mathbb{R}^{q-1}$  is a time series of disturbance input  $i$  and  $\mathbf{T} \in \mathbb{R}^{n_\nu \times n_\nu}$  is a matrix consisting of 0's and 1's.

Realizing that  $\mathbf{P}_{\nu, Q}(\mathbf{r}) = \mathbf{T} \mathbf{P}_{\bar{\nu}, Q}(\mathbf{r}) \mathbf{T}^T$ , equation 3.34 can be also be written as

$$\mathbf{P}_Q(\mathbf{r}, \hat{\mathbf{r}}) = \hat{\mathbf{S}}_Q(\hat{\mathbf{r}}) \mathbf{J}_Q \mathbf{T} \mathbf{P}_{\bar{\nu}, Q}(\mathbf{r}) \mathbf{T}^T \mathbf{J}_Q^T \hat{\mathbf{S}}_Q^T(\hat{\mathbf{r}}) \quad (3.37)$$

The advantage of this formulation is that  $\mathbf{P}_{\bar{\nu}, Q}(\mathbf{r})$  is block diagonal, which follows from equations 3.29 through 3.31. This fact will be exploited in the next section to derive an expression for the estimate error variance in terms of  $r_{v_i, kl}$  and  $r_{w_i, kl}$ .

### 3.4 Estimate Error Variance as a Function of Autocorrelation

Applying a covariance transformation to equation 3.37 results in the following expression for  $\sigma_y^2$

$$\sigma_y^2 = \boldsymbol{\alpha}_y^T \hat{\mathbf{S}}_Q(\hat{\mathbf{r}}) \mathbf{J}_Q \mathbf{T} \mathbf{P}_{\bar{v}, Q}(\mathbf{r}) \mathbf{T}^T \mathbf{J}_Q^T \hat{\mathbf{S}}_Q^T(\hat{\mathbf{r}}) \boldsymbol{\alpha}_y \quad (3.38)$$

Defining  $\boldsymbol{\omega} \in \mathbb{R}^{n_v}$  as  $\boldsymbol{\omega} = \mathbf{T}^T \mathbf{J}_Q^T \hat{\mathbf{S}}_Q^T(\hat{\mathbf{r}}) \boldsymbol{\alpha}_y$  allows  $\sigma_y^2$  to be written as

$$\sigma_y^2 = \boldsymbol{\omega}^T \mathbf{P}_{\bar{v}, Q}(\mathbf{r}) \boldsymbol{\omega} \quad (3.39)$$

Clearly,  $\sigma_y^2$  is a quadratic form in  $\mathbf{P}_{\bar{v}, Q}(\mathbf{r})$ . It was shown in equation 2.57 that equation 3.39 can be expressed in the equivalent form

$$\sigma_y^2 = \sum_{l=1}^{n_v} \sum_{m=1}^{n_v} [\mathbf{P}_{\bar{v}, Q}(\mathbf{r}) \circ \boldsymbol{\Omega}]_{lm} \quad (3.40)$$

where  $\boldsymbol{\Omega} \in \mathbb{R}^{n_v \times n_v}$  is defined as  $\boldsymbol{\Omega} = \boldsymbol{\omega} \boldsymbol{\omega}^T$ .

The covariance matrix  $\mathbf{P}_{\bar{v}, Q}(\mathbf{r})$  is given by

$$\mathbf{P}_{\bar{v}, Q}(\mathbf{r}) = \text{diag}(\bar{\mathbf{V}}_1, \dots, \bar{\mathbf{V}}_{n_v}, \bar{\mathbf{W}}_1, \dots, \bar{\mathbf{W}}_{n_w}) \quad (3.41)$$

where  $\bar{\mathbf{V}}_i \in \mathbb{R}^{q \times q}$ ,  $i=1, \dots, n_v$  and  $\bar{\mathbf{W}}_i \in \mathbb{R}^{(q-1) \times (q-1)}$ ,  $i=1, \dots, n_w$ .

Equation 3.41 implies that equation 3.40 can be written as

$$\sigma_y^2 = \sum_{l=1}^{n_v} \sum_{m=1}^q \sum_{n=1}^q (\boldsymbol{\Lambda}_l \circ \bar{\mathbf{V}}_l)_{mn} + \sum_{j=1}^{n_w} \sum_{m=1}^{q-1} \sum_{n=1}^{q-1} (\boldsymbol{\Gamma}_j \circ \bar{\mathbf{W}}_j)_{mn} \quad (3.42)$$

where  $\boldsymbol{\Lambda}_l \in \mathbb{R}^{q \times q}$  is the diagonal block of  $\boldsymbol{\Omega}$  corresponding to  $\bar{\mathbf{V}}_l$  and

$\boldsymbol{\Gamma}_j \in \mathbb{R}^{(q-1) \times (q-1)}$  is the diagonal block of  $\boldsymbol{\Omega}$  corresponding to  $\bar{\mathbf{W}}_j$ .

Each  $\bar{\mathbf{V}}_l$  and  $\bar{\mathbf{W}}_j$  is structured as

$$\bar{\mathbf{V}}_l = \begin{bmatrix} r_{v_l,11} & r_{v_l,12} & \cdots & r_{v_l,1q} \\ r_{v_l,12} & r_{v_l,22} & \cdots & r_{v_l,2q} \\ \vdots & \vdots & \ddots & \vdots \\ r_{v_l,1q} & r_{v_l,2q} & \cdots & r_{v_l,qq} \end{bmatrix} \quad (3.43)$$

$$\bar{\mathbf{W}}_j = \begin{bmatrix} r_{w_j,11} & r_{w_j,12} & \cdots & r_{w_j,1(q-1)} \\ r_{w_j,12} & r_{w_j,22} & \cdots & r_{w_j,2(q-1)} \\ \vdots & \vdots & \ddots & \vdots \\ r_{w_j,1(q-1)} & r_{w_j,2(q-1)} & \cdots & r_{w_j,(q-1)(q-1)} \end{bmatrix} \quad (3.44)$$

Substituting equations 3.43 and 3.44 into equation 3.42 and using the result from equation 2.59 results in

$$\sigma_y^2 = \sum_{l=1}^{n_v} \left[ \sum_{m=1}^q \varphi_{v_l,mm} r_{v_l,mm} + 2 \sum_{m=1}^{q-1} \sum_{n=m+1}^q \varphi_{v_l,mn} r_{v_l,mn} \right] + \sum_{j=1}^{n_w} \left[ \sum_{m=1}^{q-1} \varphi_{w_j,mm} r_{w_j,mm} + 2 \sum_{m=1}^{q-2} \sum_{n=m+1}^{q-1} \varphi_{w_j,mn} r_{w_j,mn} \right] \quad (3.45)$$

where  $\varphi_{v_l,mm}$  is the  $(m,n)$  element of  $\Lambda_l$  and  $\varphi_{w_j,mm}$  is the  $(m,n)$  element of  $\Gamma_j$ .

### 3.5 Variance Bounding with Autocorrelation Uncertainty

For unknown  $r_{v_i,kl}$  and  $r_{w_i,kl}$  that adhere to the bounded uncertainty structure given in equations 2.64 and 2.65, an upper bounding variance  $\bar{\sigma}_y^2$  can be obtained by following the same procedure outlined in Section 2.6. That is

$$\begin{aligned} \bar{\sigma}_y^2 = & \sum_{l=1}^{n_v} \left[ \sum_{m=1}^q \varphi_{v_l,mm} \bar{r}_{v_l,mm} + 2 \sum_{m=1}^{q-1} \sum_{n=m+1}^q \varphi_{v_l,mn} \bar{r}_{v_l,mn} \right] + \\ & \sum_{j=1}^{n_w} \left[ \sum_{m=1}^{q-1} \varphi_{w_j,mm} \bar{r}_{w_j,mm} + 2 \sum_{m=1}^{q-2} \sum_{n=m+1}^{q-1} \varphi_{w_j,mn} \bar{r}_{w_j,mn} \right] \end{aligned} \quad (3.46)$$

where

$$\bar{r}_{v_l,mm} = \begin{cases} a_{v_l,mm} & , \quad \varphi_{v_l,mm} < 0 \\ b_{v_l,mm} & , \quad \varphi_{v_l,mm} \geq 0 \end{cases} \quad \text{and} \quad \bar{r}_{w_j,mm} = \begin{cases} a_{w_j,mm} & , \quad \varphi_{w_j,mm} < 0 \\ b_{w_j,mm} & , \quad \varphi_{w_j,mm} \geq 0 \end{cases} \quad (3.47)$$

### 3.6 Two-Dimensional Estimation Application

The algorithms developed in Sections 3.4 and 3.5 will now be applied to the two-dimensional estimation problem shown in Figure 3.1.

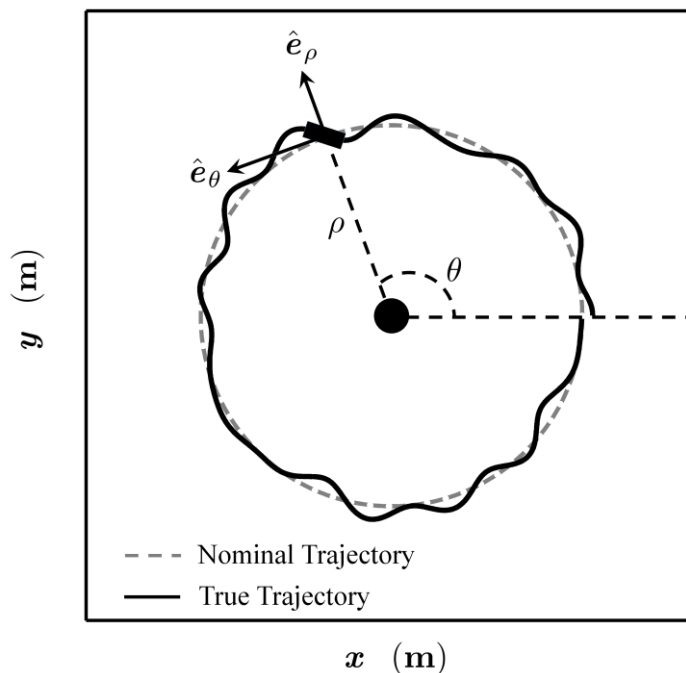


Figure 3.1. Two-Dimensional Estimation Problem

**3.6.1 Linearized Measurement and State Dynamic Models.** It is desired to estimate the position, velocity and heading of the vehicle. At any time index  $k$ , the state vector is the  $5 \times 1$  vector

$$\mathbf{x}_k = [\rho_k \quad \theta_k \quad v_{\rho,k} \quad v_{\theta,k} \quad \psi_k]^T \quad (3.48)$$

where  $\rho_k$  and  $\theta_k$  are the polar coordinates of the vehicle,  $v_{\rho,k}$  and  $v_{\theta,k}$  are the velocity components of the vehicle along the  $\hat{e}_\rho$  and  $\hat{e}_\theta$  directions, respectively, and  $\psi_k$  is the heading angle, which determines the orientation of the vehicle body frame with respect to inertial space. The body frame and heading angle  $\psi_k$  are defined in Figure 3.2.

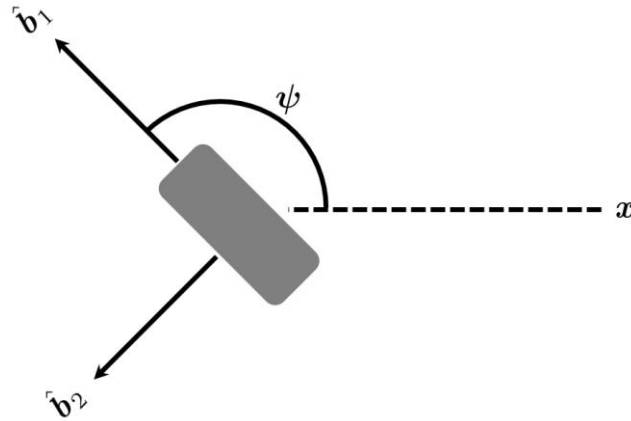


Figure 3.2. Vehicle Body Frame and Heading Angle

Two accelerometers mounted along the  $\hat{b}_1$  and  $\hat{b}_2$  axes and a rate gyroscope are attached to the vehicle. The outputs from these sensors, denoted by  $\Delta v$  and  $\Delta \phi$ , respectively, are the integrals of inertial acceleration and inertial angular velocity over a specified time interval  $\delta t$ . In addition, measurements of  $\rho$  and  $\theta$  are available from a radionavigation system: a VHF Omnidirectional Ranging beacon (VOR) colocated with

Distance Measuring Equipment (DME) at the origin [For08]. The sampling interval of the VOR/DME is  $\Delta t$ .

The measurement models are given by

$$z_{v,k} = \theta_k + v_{v,k} \quad (3.49)$$

$$z_{d,k} = \rho_k + v_{d,k} \quad (3.50)$$

where  $z_{v,k}$  and  $z_{d,k}$  are the VOR and DME measurements, respectively, and  $v_{v,k}$  and  $v_{d,k}$  are the associated measurement noise.

It is shown in Appendix C that the continuous time dynamic model of  $\mathbf{x}$  is given by

$$\begin{bmatrix} \dot{\rho} \\ \dot{\theta} \\ \dot{v}_\rho \\ \dot{v}_\theta \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} v_\rho \\ v_\theta / \rho \\ \cos(\theta - \psi) a_{b_1} + \sin(\theta - \psi) a_{b_2} + v_\theta^2 / \rho \\ -\sin(\theta - \psi) a_{b_1} + \cos(\theta - \psi) a_{b_2} - v_\rho v_\theta / \rho \\ \omega \end{bmatrix} \quad (3.51)$$

where  $a_{b_1}$  and  $a_{b_2}$  are the inertial acceleration components along the  $\hat{\mathbf{b}}_1$  and  $\hat{\mathbf{b}}_2$  axes, respectively, and  $\omega$  is the inertial angular velocity of the vehicle.

The variance upper bound in equation 3.46 was derived on the basis of linearized, discrete-time measurement and state dynamic models. Hence, before the variance bound can be computed for this example, equation 3.51 must first be converted to the linear, discrete-time form

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{u}_k + \mathbf{G}_k \mathbf{w}_k \quad (3.52)$$

A procedure to determine  $\mathbf{F}_k \in \mathbb{R}^{5 \times 5}$ ,  $\mathbf{u}_k \in \mathbb{R}^5$  and  $\mathbf{G}_k \in \mathbb{R}^{5 \times 6}$  is provided in Appendix D.

**3.6.2 Autocorrelation Bounding Functions.** This example will use the same autocorrelation function models introduced in the one-dimensional estimation problem of Section 2.7. That is, the autocorrelation bounds on  $\delta\Delta v$  and  $\delta\Delta\phi$  are determined from

$$r_{w_j, mn} = \begin{cases} 2(\sigma_j \tau_j)^2 \left[ \frac{\delta t}{\tau_j} - 1 + \exp\left(-\frac{\delta t}{\tau_j}\right) \right] + Q_j \delta t & , m = n \\ (\sigma_j \tau_j)^2 \exp\left(-\frac{|m-n| \delta t}{\tau_j}\right) \left[ 1 - \exp\left(-\frac{\delta t}{\tau_j}\right) \right] \left[ \exp\left(\frac{\delta t}{\tau_j}\right) - 1 \right] & , m \neq n \end{cases} \quad (3.53)$$

and the autocorrelation bounds on  $v_{v,k}$  and  $v_{d,k}$  are determined from the equation

$$r_{v_l, mn} = \begin{cases} \sigma_{l,g}^2 + \sigma_{l,r}^2 & , m = n \\ \sigma_{l,g}^2 \exp\left(-\frac{|m-n| \Delta t}{\tau_l}\right) & , m \neq n \end{cases} \quad (3.54)$$

where the index  $j$  varies from 1 to 3 and the index  $l$  varies from 1 to 2.

The parameters defining the bounds are provided in Tables 3.1 and 3.2.

Table 3.1. Accelerometer and Gyroscope Autocorrelation Bound Parameters

Accelerometer		Gyroscope	
Lower Bound	Upper Bound	Lower Bound	Upper Bound
$Q^{1/2} = 5e^{-3} \text{ m/s}^{3/2}$	$Q^{1/2} = 10e^{-3} \text{ m/s}^{3/2}$	$Q^{1/2} = 0.05 \text{ deg/s}^{1/2}$	$Q^{1/2} = 0.1 \text{ deg/s}^{1/2}$
$\sigma = 0.25 \text{ m/s}^2$	$\sigma = 0.50 \text{ m/s}^2$	$\sigma = 0.15 \text{ deg/s}$	$\sigma = 0.30 \text{ deg/s}$
$\tau = 25 \text{ sec}$	$\tau = 50 \text{ sec}$	$\tau = 25 \text{ sec}$	$\tau = 50 \text{ sec}$

Table 3.2. VOR and DME Autocorrelation Bound Parameters

VOR		DME	
Lower Bound	Upper Bound	Lower Bound	Upper Bound
$\sigma_r = 0.25$ deg	$\sigma_r = 0.50$ deg	$\sigma_r = 0.125$ m	$\sigma_r = 0.25$ m
$\sigma_g = 0.50$ deg	$\sigma_g = 1.0$ deg	$\sigma_g = 0.25$ m	$\sigma_g = 0.50$ m
$\tau = 25$ sec	$\tau = 50$ sec	$\tau = 25$ sec	$\tau = 50$ sec

Substituting the upper and lower bound values into equations 3.53 and 3.54 defines the autocorrelation bounding functions. In addition, the upper bound values will be used to specify the weighting matrix of the WLS estimator. For the numerical simulation, the VOR/DME sampling interval is 2 sec (i.e.,  $\Delta t = 2$  sec) and the accelerometer and gyroscope sampling interval is 0.5 sec (i.e.,  $\delta t = 0.5$  sec).

**3.6.3 Radial Position State Simulation Results.** For the radial position state, the bounding variance  $\bar{\sigma}_\rho^2$  is determined using the vector  $\alpha_\rho = [1 \ 0 \ 0 \ 0 \ 0]^T$ . Figures 3.3 through 3.7 show the autocorrelation functions producing the variance bound at an elapsed time of 30 seconds.

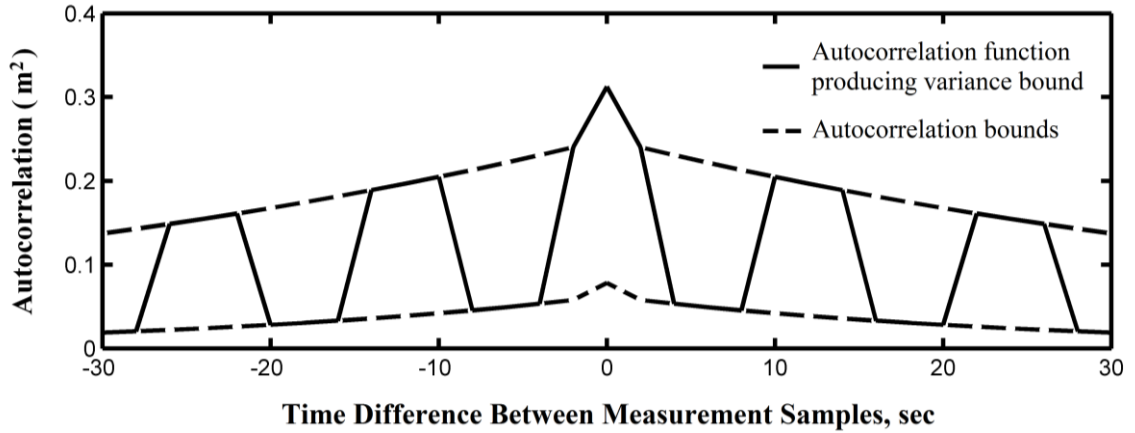


Figure 3.3. DME ACF Producing Position State Variance Bound

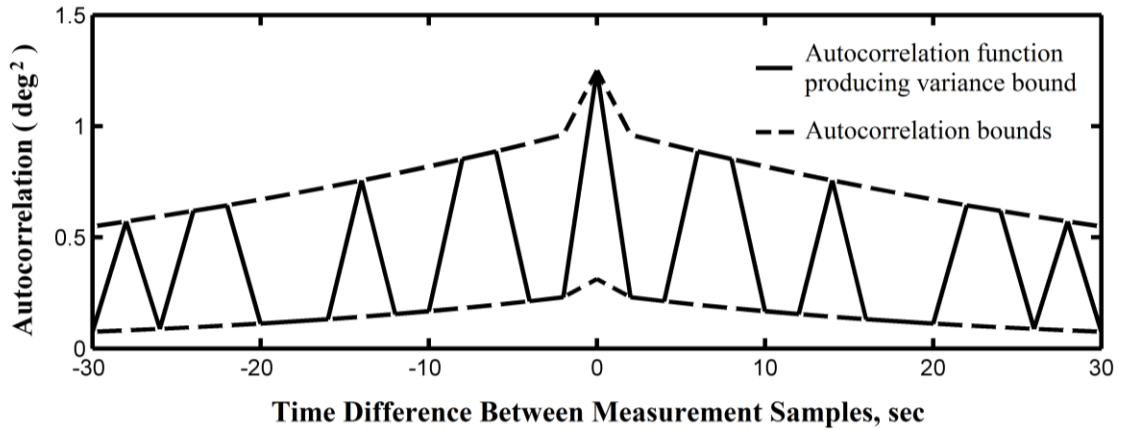


Figure 3.4. VOR ACF Producing Position State Variance Bound

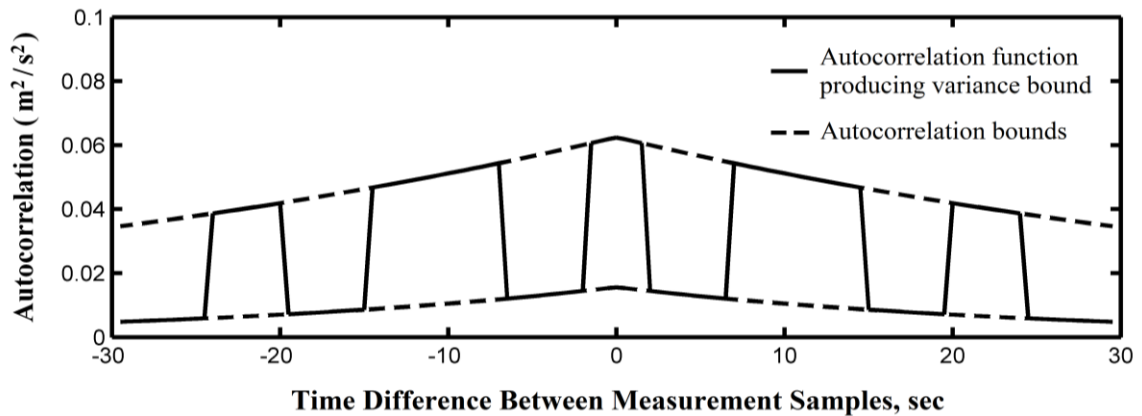


Figure 3.5. Accelerometer # 1 ACF Producing Position State Variance Bound

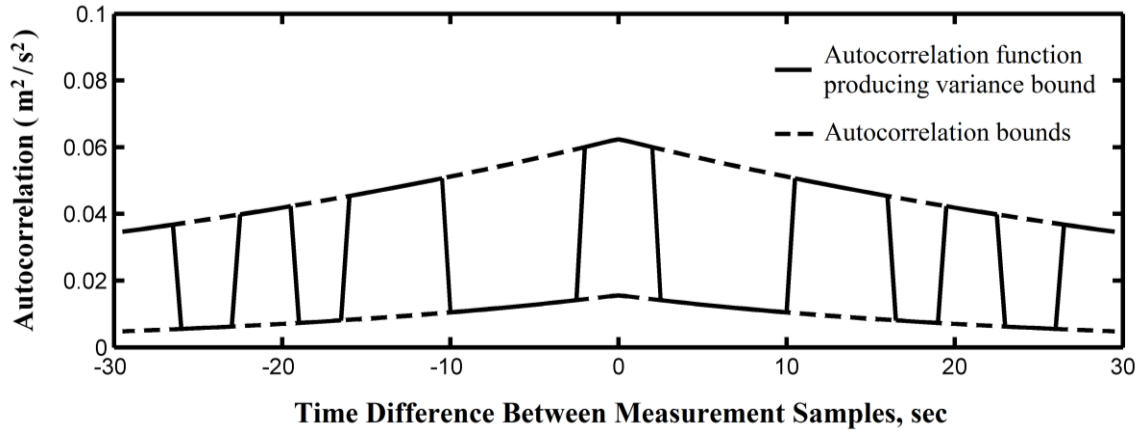


Figure 3.6. Accelerometer # 2 ACF Producing Position State Variance Bound

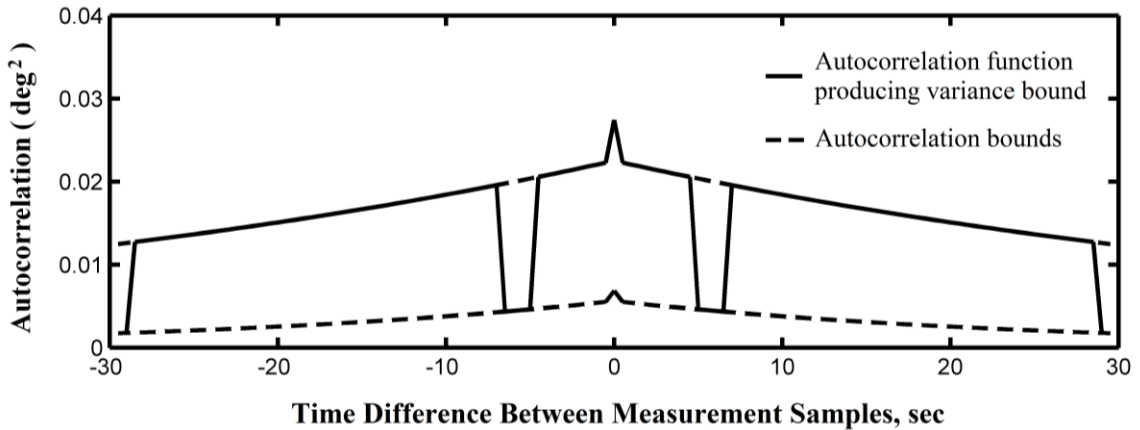


Figure 3.7. Gyroscope ACF Producing Position State Variance Bound

For the WLS estimator defined using the upper bound values from Tables 3.1 and 3.2, the estimate error variance  $\sigma_p^2$  will attain its largest value when the VOR, DME and inertial sensor errors have the autocorrelation functions shown above. However, there is one important constraint that is not being satisfied: that the autocorrelation function values from Figures 3.3 through 3.7 must produce matrices  $\bar{\mathbf{V}}_l$  and  $\bar{\mathbf{W}}_j$  that are positive semi-definite. This is most easily seen by checking the eigenvalues of  $\bar{\mathbf{V}}_l$  and  $\bar{\mathbf{W}}_j$ . If the

eigenvalues are all non-negative, then the constraint is satisfied. The results of this test are summarized in Table 3.3.

Table 3.3. Range of Eigenvalues for Worst-Case Autocorrelation Functions

Sensor	Eigenvalue Range
DME	$[-0.360, 2.08] \text{ m}^2$
VOR	$[-0.831, 7.30] \text{ deg}^2$
Accelerometer # 1	$[-0.243, 1.97] \text{ m}^2/\text{s}^2$
Accelerometer # 2	$[-0.139, 1.75] \text{ m}^2/\text{s}^2$
Gyroscope	$[-8.41\text{e}^{-2}, 1.02] \text{ deg}^2$

It can be seen from Table 3.3 that none of the autocorrelation functions in Figures 3.3 through 3.7 are positive semi-definite. This implies that the estimate error variance bound given in equation 3.46 is overly conservative. A tighter bound on the variance can be achieved by requiring the autocorrelation functions to be positive semi-definite in addition to being contained between upper and lower bounding functions. This approach will be pursued in the next chapter.

## CHAPTER 4

## BOUND REDUCTION VIA SEMI-DEFINITE OPTIMIZATION

The integrity risk bounding problem is formally defined in this chapter in terms of semi-definite optimization. Section 4.1 formulates the problem statement, which is converted in Section 4.2 into the standard primal form of a semi-definite programming problem. Equations for primal-dual optimality are introduced in Section 4.3 that are solved using an infeasible interior point algorithm. Comparisons between the SDP bounding method and the direct approach developed in Chapters 2 and 3 are made in Section 4.4 for the two-dimensional estimation application.

#### 4.1 Problem Statement

The variance bounds given in equations 2.66 and 3.46 can both be stated formally as the solution to the optimization problem

$$\begin{aligned} \bar{\sigma}_y^2 = \beta + \sum_{l=1}^{n_v} \max_{\mathbf{V}_l} \sum_{m=1}^{n_\lambda} \sum_{n=1}^{n_\lambda} (\Lambda_l \circ \mathbf{V}_l)_{mn} + \sum_{j=1}^{n_w} \max_{\mathbf{W}_j} \sum_{m=1}^{n_\gamma} \sum_{n=1}^{n_\gamma} (\Gamma_j \circ \mathbf{W}_j)_{mn} \end{aligned} \quad (4.1)$$

s. t.  $a_i \leq r_i \leq b_i$  ,  $i = 1, \dots, n_r$

where  $n_\lambda$  is the size of  $\Lambda_l$  for  $l = 1, \dots, n_v$  and  $n_\gamma$  is the size of  $\Gamma_j$  for  $j = 1, \dots, n_w$ .

Notice that  $\beta = 0$  for the batch variance bound. Because  $\mathbf{V}_l$  and  $\mathbf{W}_j$  are symmetric matrices, the total number of autocorrelation function values  $n_r$  is given by

$$n_r = n_v \left[ \frac{n_\lambda(n_\lambda + 1)}{2} \right] + n_w \left[ \frac{n_\gamma(n_\gamma + 1)}{2} \right] \quad (4.2)$$

A tighter variance bound can be defined by including positive semi-definite constraints on  $\mathbf{V}_l$  and  $\mathbf{W}_j$  in equation 4.1, resulting in the new optimization problem

$$\begin{aligned}
(\sigma_y^*)^2 = & \beta + \sum_{l=1}^{n_v} \max_{\mathbf{V}_l} \sum_{m=1}^{n_\lambda} \sum_{n=1}^{n_\lambda} (\Lambda_l \circ \mathbf{V}_l)_{mn} + \sum_{j=1}^{n_w} \max_{\mathbf{W}_j} \sum_{m=1}^{n_\gamma} \sum_{n=1}^{n_\gamma} (\Gamma_j \circ \mathbf{W}_j)_{mn} \\
\text{s. t. } & a_i \leq r_i \leq b_i, \quad i=1, \dots, n_r \\
& \mathbf{V}_l \succeq 0, \quad l=1, \dots, n_v \\
& \mathbf{W}_j \succeq 0, \quad j=1, \dots, n_w
\end{aligned} \tag{4.3}$$

Consider the cost function of the maximization problem corresponding to  $l=1$ .

Using index notation and temporarily dropping the  $l$  subscript, notice that

$$\sum_{m=1}^{n_\lambda} \sum_{n=1}^{n_\lambda} (\Lambda \circ \mathbf{V})_{mn} = \Lambda_{mn} \mathbf{V}_{mn} = \Lambda_{mn} \mathbf{V}_{nm}^T \tag{4.4}$$

The expression  $\Lambda_{mn} \mathbf{V}_{nm}^T$  is recognized as the trace of the product,  $\Lambda \mathbf{V}^T$ . Given that  $\mathbf{V}$  is symmetric, equation 4.4 leads to the conclusion that

$$\sum_{m=1}^{n_\lambda} \sum_{n=1}^{n_\lambda} (\Lambda \circ \mathbf{V})_{mn} = \text{Tr}(\Lambda \mathbf{V}) \tag{4.5}$$

Therefore, equation 4.3 can be written in the equivalent form

$$\begin{aligned}
(\sigma_y^*)^2 = & \beta + \sum_{l=1}^{n_v} \max_{\mathbf{V}_l} \text{Tr}(\Lambda_l \mathbf{V}_l) + \sum_{j=1}^{n_w} \max_{\mathbf{W}_j} \text{Tr}(\Gamma_j \mathbf{W}_j) \\
\text{s. t. } & a_i \leq r_i \leq b_i, \quad i=1, \dots, n_r \\
& \mathbf{V}_l \succeq 0, \quad l=1, \dots, n_v \\
& \mathbf{W}_j \succeq 0, \quad j=1, \dots, n_w
\end{aligned} \tag{4.6}$$

In what follows, the constant term  $\beta$  will be omitted because it does not contribute to the optimization process. There are  $n_v + n_w$  independent maximization problems in equation 4.6, each with the prototypical structure

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \text{Tr}(-\mathbf{K}\mathbf{Y}) \\ \text{s. t.} \quad & a_{ij} \leq y_{ij} \leq b_{ij}, \quad \begin{array}{l} i = 1, \dots, n \\ j = i, \dots, n \end{array} \\ & \mathbf{Y} \succeq 0 \end{aligned} \quad (4.7)$$

The negative sign is introduced into the cost function to convert the maximization problem to a minimization problem.

## 4.2 Conversion to Standard Primal Form

In standard form, the primal SDP problem is given by

$$\begin{aligned} \min_{\mathbf{X}} \quad & \text{Tr}(\mathbf{C}\mathbf{X}) \\ \text{SDP (Primal)} \quad & \text{s. t. } \text{Tr}(\mathbf{A}_i \mathbf{X}) = d_i, \quad i = 1, \dots, m \\ & \mathbf{X} \succeq 0 \end{aligned} \quad (4.8)$$

To convert equation 4.7 to standard primal form, the inequality constraints must be converted to equality constraints and then written in terms of a matrix trace. For given indices  $i$  and  $j$ , the two inequality constraints can be represented in the equivalent form

$$a_{ij} \leq y_{ij} \leq b_{ij} \quad \rightarrow \quad \begin{array}{l} y_{ij} - e_{ij} = a_{ij}, \quad e_{ij} \geq 0 \\ y_{ij} + s_{ij} = b_{ij}, \quad s_{ij} \geq 0 \end{array} \quad (4.9)$$

Let  $\mathbf{d} \in \mathbb{R}^{n(n+1)}$  be equal to

$$\mathbf{d} = [e_{11} \quad s_{11} \quad \dots \quad e_{nn} \quad s_{nn}]^T \quad (4.10)$$

and define  $\mathbf{D} \in \mathbb{R}^{n(n+1) \times n(n+1)}$  as a diagonal matrix with the elements of  $\mathbf{d}$  on the main diagonal. Combining  $\mathbf{Y}$  and  $\mathbf{D}$  into one  $n(n+2) \times n(n+2)$  matrix  $\mathbf{X}$  yields

$$\mathbf{X} = \begin{bmatrix} \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \quad (4.11)$$

To illustrate how the two equality constraints in equation 4.9 can be written in the form  $\text{Tr}(\mathbf{A}_i \mathbf{X}) = d_i$ , consider the case where  $n = 2$ . Then  $\mathbf{X}$  is given by

$$\mathbf{X} = \begin{bmatrix} y_{11} & y_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ y_{12} & y_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & s_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_{22} \end{bmatrix} \quad (4.12)$$

The constraint  $y_{11} - e_{11} = a_{11}$  can be written as  $\text{Tr}(\mathbf{A} \mathbf{X}) = a_{11}$ , where  $\mathbf{A}$  is defined as

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.13)$$

Similarly, the constraint  $y_{12} + s_{12} = b_{12}$  can be written as  $\text{Tr}(\mathbf{A} \mathbf{X}) = b_{12}$  by defining  $\mathbf{A}$  as

$$\mathbf{A} = \begin{bmatrix} 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.14)$$

The same procedure can be followed for the other constraints in equation 4.9.

Defining  $\mathbf{C} \in \mathbb{R}^{n(n+2) \times n(n+2)}$  as

$$\mathbf{C} = \begin{bmatrix} -\mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (4.15)$$

equation 4.7 can be written in the standard form

$$\begin{aligned} & \min_{\mathbf{X}} \text{Tr}(\mathbf{C}\mathbf{X}) \\ \text{SDP (Primal)} \quad & \text{s. t. } \text{Tr}(\mathbf{A}_i \mathbf{X}) = d_i, \quad i = 1, \dots, n(n+1) \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned} \quad (4.16)$$

### 4.3 Primal-Dual Optimality and Interior Point Algorithms

The dual problem associated with the primal SDP in equation 4.16 is given by

$$\begin{aligned} & \max_{\boldsymbol{\lambda}} \mathbf{d}^T \boldsymbol{\lambda} \\ \text{SDP (Dual)} \quad & \text{s. t. } \sum_{i=1}^{n(n+1)} \lambda_i \mathbf{A}_i + \mathbf{S} = \mathbf{C} \\ & \mathbf{S} \succeq \mathbf{0} \end{aligned} \quad (4.17)$$

where  $\boldsymbol{\lambda} \in \mathbb{R}^{n(n+1)}$  is the Lagrange multiplier vector for the primal equality constraints and  $\mathbf{S} \in \mathbb{R}^{n(n+2) \times n(n+2)}$  is the dual slack matrix.

It is shown in Appendix E that the optimal solution to the primal and dual problems must satisfy the system of matrix equations

$$\text{Tr}(\mathbf{A}_i \mathbf{X}) = d_i, \quad i = 1, \dots, n(n+1) \quad (4.18)$$

$$\sum_{i=1}^{n(n+1)} \lambda_i \mathbf{A}_i + \mathbf{S} = \mathbf{C} \quad (4.19)$$

$$\mathbf{X} \mathbf{S} = \mathbf{0} \quad (4.20)$$

$$\mathbf{X} \succeq \mathbf{0}, \quad \mathbf{S} \succeq \mathbf{0} \quad (4.21)$$

In this dissertation, an infeasible interior point algorithm is used to solve equations 4.18 through 4.21. Specific details concerning the algorithm can be found in Appendix F. The variance bound  $(\sigma_y^*)^2$  is obtained by solving this set of equations for each of the  $n_v + n_w$  maximization problems in equation 4.6.

#### 4.4 Comparison of Bounding Methods

A comparison will now be made between the SDP bounding approach developed in this chapter and the direct approach presented in Chapters 2 and 3 for the two-dimensional estimation problem. In that example, the VOR, DME, accelerometer and gyroscope measurement error autocorrelation functions are wide sense stationary. Therefore, in addition to being symmetric,  $\mathbf{V}_l$  and  $\mathbf{W}_j$  are also Toeplitz. This extra information about the structure of  $\mathbf{V}_l$  and  $\mathbf{W}_j$  is incorporated in the SDP solution via additional constraints.

**4.4.1 Symmetric Toeplitz Constraints.** The nature of the symmetric Toeplitz constraints is most easily revealed by examining a specific case. Suppose that the matrix variable  $\mathbf{Y}$  in equation 4.7 is  $4 \times 4$ .

$$\mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} \\ y_{12} & y_{22} & y_{23} & y_{24} \\ y_{13} & y_{23} & y_{33} & y_{34} \\ y_{14} & y_{24} & y_{34} & y_{44} \end{bmatrix} \quad (4.22)$$

The symmetric Toeplitz form of  $\mathbf{Y}$  (denoted by  $\mathbf{Y}_t$ ) is given by

$$\mathbf{Y}_t = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 \\ y_2 & y_1 & y_2 & y_3 \\ y_3 & y_2 & y_1 & y_2 \\ y_4 & y_3 & y_2 & y_1 \end{bmatrix} \quad (4.23)$$

Comparing  $\mathbf{Y}$  to  $\mathbf{Y}_t$ , it can be seen that  $\mathbf{Y}$  will be symmetric Toeplitz when the constraints in Table 4.1 are satisfied.

Table 4.1. Symmetric Toeplitz Constraints for  $4 \times 4$  Matrix

Diagonal Constraints	Off-Diagonal Constraints
$y_{11} - y_{22} = 0$	$y_{12} - y_{23} = 0$
$y_{22} - y_{33} = 0$	$y_{23} - y_{34} = 0$
$y_{33} - y_{44} = 0$	$y_{13} - y_{24} = 0$

In general, there are  $(n-1) + (n-1)(n-2)/2$  symmetric Toeplitz constraints, each of which can be written in the form  $\text{Tr}(\mathbf{A}_i \mathbf{X}) = 0$  and included in the constraint set of equation 4.16.

**4.4.2 Radial Position State Simulation Results.** Figure 4.1 shows the percent reduction in  $\bar{\sigma}_p^2$  achieved using the SDP bounding method. The results are shown in 2 second intervals. For this example, the state vector is  $5 \times 1$ , which can only be observed when at least three VOR/DME measurements populate the batch. Therefore, the results begin at an elapsed time of four seconds.

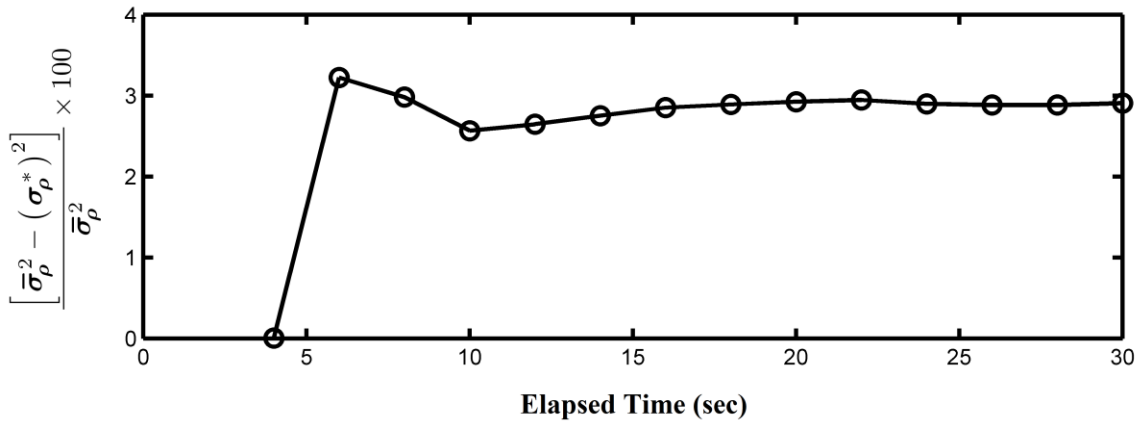


Figure 4.1. Percent Reduction in Variance Bound Achieved Using SDP

At an elapsed time of six seconds, the maximum percent reduction of 3.2% is achieved which quickly settles to approximately 3%. For a fixed value of the variance bound, the SDP method can tolerate a larger separation between the autocorrelation bounding functions, thereby reducing the required level of knowledge of the measurement noise and disturbance input statistics. Whether or not it is worth using an SDP approach over the direct bounding method will be problem dependent.

For real-time applications, the computational efficiency of the direct bounding method far outweighs the bound reduction benefits of the SDP approach. In these instances, if necessary levels of integrity cannot be achieved with the direct approach,

more effort should be placed in reducing the separation between the autocorrelation bounding functions. On the other hand, if the integrity risk bound is not required in real-time and/or it is not possible to reduce autocorrelation bound separation, then SDP may be the only way to achieve the required degree of integrity.

The sensor error autocorrelation functions producing the estimate error variance bound at an elapsed time of 30 seconds are shown in Figures 4.2 through 4.6.

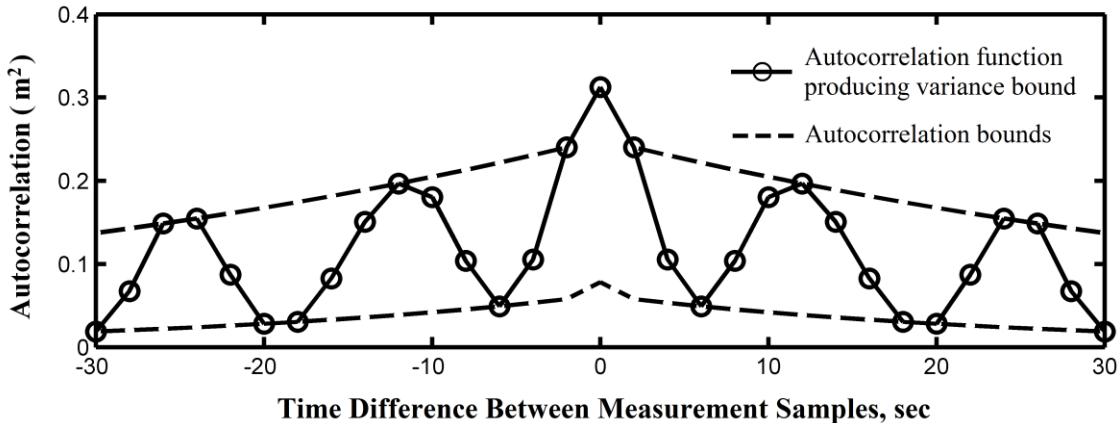


Figure 4.2. DME ACF via SDP Leading to Position State Variance Bound

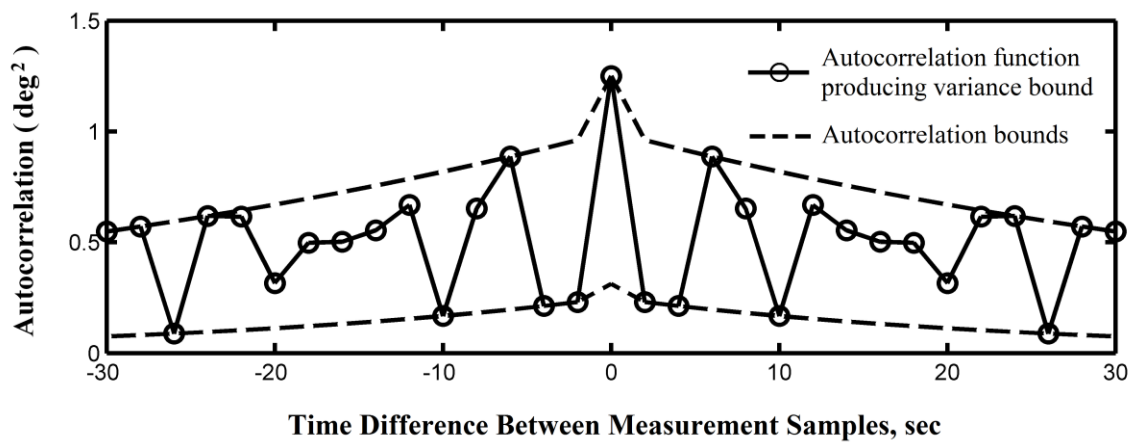


Figure 4.3. VOR ACF via SDP Leading to Position State Variance Bound

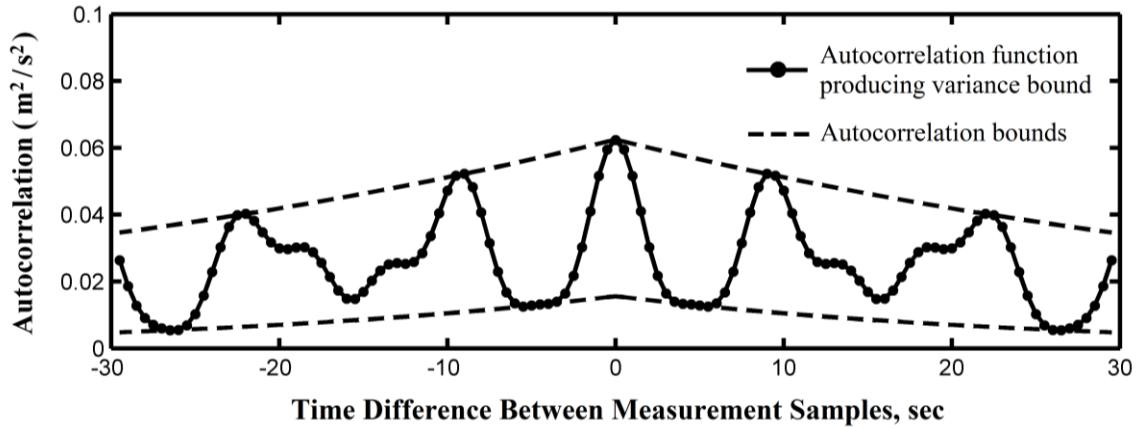


Figure 4.4. Accelerometer # 1 ACF via SDP Leading to Position State Variance Bound

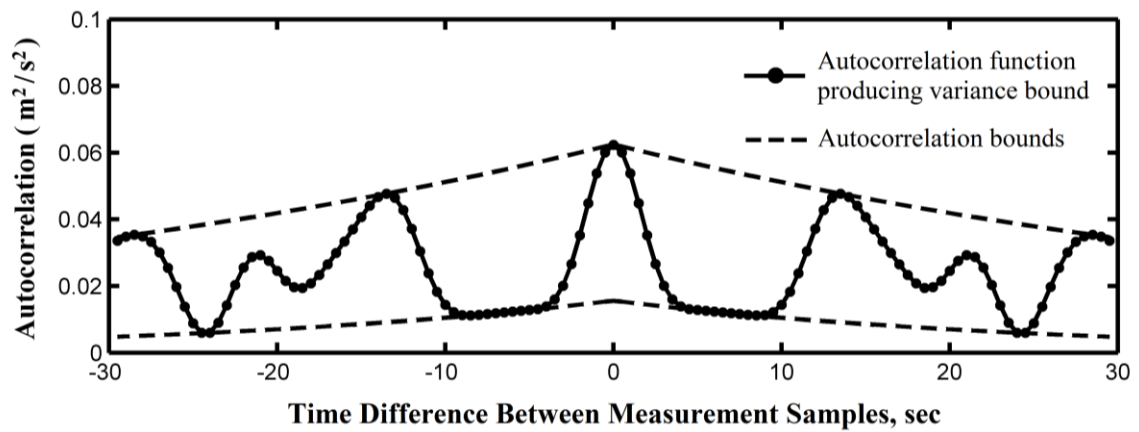


Figure 4.5. Accelerometer # 2 ACF via SDP Leading to Position State Variance Bound

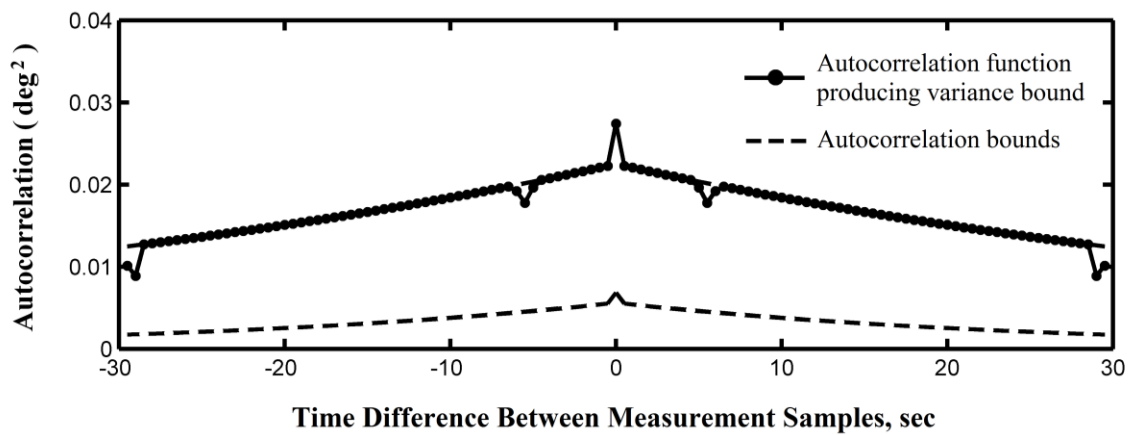


Figure 4.6. Gyroscope ACF via SDP Leading to Position State Variance Bound

Table 4.2. Eigenvalues of SDP ACFs Producing Position State Variance Bound

Sensor	Eigenvalue Range
DME	$[9.77e^{-9}, 2.19] \text{ m}^2$
VOR	$[1.34e^{-11}, 8.29] \text{ deg}^2$
Accelerometer # 1	$[3.74e^{-8}, 1.73] \text{ m}^2/\text{s}^2$
Accelerometer # 2	$[7.82e^{-8}, 1.57] \text{ m}^2/\text{s}^2$
Gyroscope	$[5.18e^{-10}, 1.12] \text{ deg}^2$

Table 4.2 shows that the autocorrelation functions leading to the radial position state variance bound are indeed positive semi-definite. Chapters 2 through 4 have considered integrity risk bounding for unconstrained estimation applications. However, certain applications contain additional information about the state vector in the form of constraints. The next chapter will address the integrity risk bounding problem for a particular class of constrained estimation applications.

## CHAPTER 5

## MIXED REAL/INTEGER INTEGRITY RISK BOUNDING

This chapter considers estimation applications where a subset of the state vector is constrained to be integer-valued. Section 5.1 introduces the integer bootstrap estimator and shows that the impact of bootstrapping on the state estimate error vector can be accounted for via successive measurement updates. The mixed real/integer integrity risk equation is derived in Section 5.2 as a function of a vector of conditional variances. For the bounded autocorrelation uncertainty structure, Section 5.3 defines the integrity risk bound as the solution of a non-convex optimization problem.

Section 5.4 provides two approaches to solve the optimization problem globally using a hyper-rectangular feasible region that circumscribes the true polytopic feasible region. The performance of these methods is demonstrated for a static baseline estimation application in Section 5.5. A third method is developed in Section 5.6 to upper bound integrity risk over the polytopic feasible region, and Section 5.7 compares the performance of the polytopic and rectangular bounds for the baseline estimation problem.

### 5.1 Integer Bootstrap Estimator

Suppose that  $\mathbf{x}$  can be partitioned as

$$\mathbf{x} = [\mathbf{a}^T \quad \mathbf{p}^T]^T \quad (5.1)$$

where  $\mathbf{a} \in \mathbb{Z}^{n_a}$  is the integer subset of  $\mathbf{x}$  and  $\mathbf{p} \in \mathbb{R}^{n_p}$  is the real-valued subset of  $\mathbf{x}$  with  $n_a + n_p = n_x$ .

Integer bootstrapping is one method available for converting a real-valued estimate  $\hat{\mathbf{a}}$  to an integer estimate  $\tilde{\mathbf{a}}$  [Teu01b]. The vector  $\hat{\mathbf{a}}$  is random because it is dependent on the noisy measurement vector,  $\mathbf{z}$ . This implies that there is a certain probability that  $\tilde{\mathbf{a}}$  will be equal to a given integer vector  $\boldsymbol{\zeta}$ , with correct integer estimation being the most desirable. It is possible to increase the probability of successful integer estimation significantly by operating on a vector  $\mathbf{b}$  that is a specific integer linear combination of  $\mathbf{a}$ . That is,  $\mathbf{b} = \mathbf{Z}^T \mathbf{a}$ , where  $\mathbf{Z}^T \in \mathbb{Z}^{n_a \times n_a}$  is determined from the estimate error covariance matrix  $\hat{\mathbf{P}}_a \in \mathbb{R}^{n_a \times n_a}$  associated with  $\hat{\mathbf{a}}$  [Teu95]. Both  $\mathbf{Z}$  and  $\mathbf{Z}^{-1}$  must have integer entries to ensure that transformations between the original domain and the  $z$ -domain do not change the integer nature of  $\mathbf{a}$  and  $\mathbf{b}$ .

**5.1.1 Accounting for Bootstrapping Via Measurement Update.** Let the estimate of  $\mathbf{x}$  and its associated estimate error covariance matrix prior to bootstrapping be  $\hat{\mathbf{x}}_0$  and  $\hat{\mathbf{P}}_0$ , respectively, where  $\hat{\mathbf{P}}_a$  is the upper left  $n_a \times n_a$  block of  $\hat{\mathbf{P}}_0$ . The bootstrapping process begins by rounding the real-valued estimate  $\hat{b}_{1|0} = \mathbf{Z}_1^T \hat{\mathbf{a}}_0$  to its nearest integer, where  $\mathbf{Z}_1^T$  is the first row of  $\mathbf{Z}^T$ . Mathematically, this is denoted by  $\tilde{b}_{1|0} = [\hat{b}_{1|0}]$ , where  $[\bullet]$  indicates the rounding operation.

In general, the estimate error covariance matrix  $\hat{\mathbf{P}}_0$  is fully populated. Therefore,  $\tilde{b}_{1|0}$  conveys information about the other components of  $\mathbf{x}$  by virtue of their correlation with  $b_1$ . By viewing  $\tilde{b}_{1|0}$  as a noise-free measurement of  $b_1$ , the impact of the initial

bootstrap step is captured by updating  $\hat{\mathbf{x}}_0$  and  $\hat{\mathbf{P}}_0$  to the conditional quantities  $\hat{\mathbf{x}}_1$  and  $\hat{\mathbf{P}}_1$  via a Kalman measurement update. That is

$$\hat{\mathbf{x}}_1 = \hat{\mathbf{x}}_0 + \mathbf{K}_1(\tilde{b}_{1|0} - \mathbf{H}_1 \hat{\mathbf{x}}_0) \quad (5.2)$$

$$\hat{\mathbf{P}}_1 = (\mathbf{I} - \mathbf{K}_1 \mathbf{H}_1) \hat{\mathbf{P}}_0 \quad (5.3)$$

where  $\mathbf{H}_1 \in \mathbb{Z}^{1 \times n_x}$  and  $\mathbf{K}_1 \in \mathbb{R}^{n_x}$  are given by

$$\mathbf{H}_1 = [\mathbf{Z}_1^T \quad \mathbf{0}^T] \quad (5.4)$$

$$\mathbf{K}_1 = \hat{\mathbf{P}}_0 \mathbf{H}_1^T (\mathbf{H}_1 \hat{\mathbf{P}}_0 \mathbf{H}_1^T)^{-1} \quad (5.5)$$

During the second step of bootstrapping, the real-valued estimate  $\hat{b}_{2|1} = \mathbf{Z}_2^T \hat{\mathbf{a}}_1$  is rounded to its nearest integer followed by another measurement update.

$$\tilde{b}_{2|1} = [\hat{b}_{2|1}] \quad (5.6)$$

$$\mathbf{H}_2 = [\mathbf{Z}_2^T \quad \mathbf{0}^T] \quad (5.7)$$

$$\mathbf{K}_2 = \hat{\mathbf{P}}_1 \mathbf{H}_2^T (\mathbf{H}_2 \hat{\mathbf{P}}_1 \mathbf{H}_2^T)^{-1} \quad (5.8)$$

$$\hat{\mathbf{x}}_2 = \hat{\mathbf{x}}_1 + \mathbf{K}_2(\tilde{b}_{2|1} - \mathbf{H}_2 \hat{\mathbf{x}}_1) \quad (5.9)$$

$$\hat{\mathbf{P}}_2 = (\mathbf{I} - \mathbf{K}_2 \mathbf{H}_2) \hat{\mathbf{P}}_1 \quad (5.10)$$

In general, the process above can be executed  $m$  times, with  $0 \leq m \leq n_a$ . The size of  $m$  is ultimately dictated by the integrity risk requirements of a given application. The integer estimates will be combined in the  $m \times 1$  vector  $\tilde{\mathbf{b}}_B = [\tilde{b}_{1|0} \quad \tilde{b}_{2|1} \quad \cdots \quad \tilde{b}_{m|m-1}]^T$ .

**5.1.2 Bootstrap Success Rate.** Let  $\hat{\mathbf{P}}_b \in \mathbb{R}^{n_a \times n_a}$  be the matrix  $\hat{\mathbf{P}}_b = \mathbf{Z}^T \hat{\mathbf{P}}_a \mathbf{Z}$  and let  $\mathbf{L} \in \mathbb{R}^{n_a \times n_a}$  and  $\mathbf{D} \in \mathbb{R}^{n_a \times n_a}$  be the matrices associated with an  $\mathbf{LDL}^T$  decomposition of  $\hat{\mathbf{P}}_b$ . It is shown in [Teu01b] that the probability of the event  $\check{\mathbf{b}}_B = \boldsymbol{\zeta}$  is given by

$$P(\check{\mathbf{b}}_B = \boldsymbol{\zeta}) = \prod_{j=1}^m \left[ \Phi \left( \frac{1 - 2\mathbf{l}_j^T (\mathbf{b}_B - \boldsymbol{\zeta})}{2\hat{\sigma}_{\check{b}_{j|j-1}}} \right) + \Phi \left( \frac{1 + 2\mathbf{l}_j^T (\mathbf{b}_B - \boldsymbol{\zeta})}{2\hat{\sigma}_{\check{b}_{j|j-1}}} \right) - 1 \right], \boldsymbol{\zeta} \in \mathbb{Z}^m \quad (5.11)$$

where  $\mathbf{l}_j \in \mathbb{R}^m$  is the vector of the first  $m$  components of the  $j^{\text{th}}$  column of  $\mathbf{L}^{-T}$ ,  $\mathbf{b}_B \in \mathbb{Z}^m$  is the true integer vector that  $\check{\mathbf{b}}_B$  is an estimate of, and  $\hat{\sigma}_{\check{b}_{j|j-1}}^2$  is the error variance of  $\hat{b}_{j|j-1}$  obtained from  $\hat{\mathbf{P}}_{j-1}$ . The function  $\Phi(x)$  is the cumulative distribution function for a standard normal random variable, i.e.,

$$\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right) du \quad (5.12)$$

## 5.2 Integrity Risk in Mixed Real/Integer Estimation

The set of all possible integer vectors  $\boldsymbol{\zeta}$  constitute a countably infinite set. Therefore, the law of total probability can be used to write the integrity risk associated with  $\mathbf{y} = \boldsymbol{\alpha}^T \mathbf{x}$  as

$$I_y = \sum_{i=1}^{\infty} P(\varepsilon_y \notin [-\ell_y, \ell_y] \mid \check{\mathbf{b}}_B = \boldsymbol{\zeta}_i) P(\check{\mathbf{b}}_B = \boldsymbol{\zeta}_i) \quad (5.13)$$

Equation 5.13 cannot be used to quantify integrity risk because it is an infinite series. A conservative approximation for  $I_y$  can be obtained by first splitting the series into two parts, resulting in [Kha10b]

$$I_y = \sum_{i=1}^n P(\varepsilon_y \notin [-\ell_y, \ell_y] \mid \tilde{\mathbf{b}}_B = \zeta_i) P(\tilde{\mathbf{b}}_B = \zeta_i) + \sum_{i=n+1}^{\infty} P(\varepsilon_y \notin [-\ell_y, \ell_y] \mid \tilde{\mathbf{b}}_B = \zeta_i) P(\tilde{\mathbf{b}}_B = \zeta_i) \quad (5.14)$$

Conservatively assuming that  $P(\varepsilon_y \notin [-\ell_y, \ell_y] \mid \tilde{\mathbf{b}}_B = \zeta_i) = 1$  for all  $i \geq n+1$ ,

equation 5.14 simplifies to

$$\hat{I}_y = \sum_{i=1}^n P(\varepsilon_y \notin [-\ell_y, \ell_y] \mid \tilde{\mathbf{b}}_B = \zeta_i) P(\tilde{\mathbf{b}}_B = \zeta_i) + \sum_{i=n+1}^{\infty} P(\tilde{\mathbf{b}}_B = \zeta_i) \quad (5.15)$$

where  $\hat{I}_y$  is used to indicate that equation 5.15 is a conservative approximation of  $I_y$ .

Noting that

$$\sum_{i=n+1}^{\infty} P(\tilde{\mathbf{b}}_B = \zeta_i) = 1 - \sum_{i=1}^n P(\tilde{\mathbf{b}}_B = \zeta_i) \quad (5.16)$$

equation 5.15 can be re-written as

$$\hat{I}_y = \sum_{i=1}^n P(\varepsilon_y \notin [-\ell_y, \ell_y] \mid \tilde{\mathbf{b}}_B = \zeta_i) P(\tilde{\mathbf{b}}_B = \zeta_i) + 1 - \sum_{i=1}^n P(\tilde{\mathbf{b}}_B = \zeta_i) \quad (5.17)$$

which can finally be simplified to

$$\hat{I}_y = 1 - \sum_{i=1}^n P(\varepsilon_y \in [-\ell_y, \ell_y] \mid \tilde{\mathbf{b}}_B = \zeta_i) P(\tilde{\mathbf{b}}_B = \zeta_i) \quad (5.18)$$

In principle,  $\hat{I}_y$  can be computed using any set of  $n$  integer vectors. A method is provided in [Kha10b] to choose this set so that  $\hat{I}_y$  is a tight upper bound on  $I_y$ .

**5.2.1 Integrity Risk as a Function of Conditional Variance.** The first probability on the right hand side of equation 5.18 is the conditional probability that  $\varepsilon_y \in [-\ell_y, \ell_y]$  given that bootstrapping resulted in the integer estimate  $\tilde{\mathbf{b}}_B = \zeta_i$ . This probability will be

written using the shorthand notation  $P(\varepsilon_{y|m} \in [-\ell_y, \ell_y])$ , where  $\varepsilon_{y|m}$  is the estimate error of  $y$  after  $m$  steps of bootstrapping. Because bootstrapping can be represented as a linear measurement updating process and the fact that the measurement noise and disturbance inputs are Gaussian,  $\varepsilon_{y|m}$  is a Gaussian random variable.

The variance of  $\varepsilon_{y|m}$  is obtained from the transformation  $\hat{\sigma}_{y|m}^2 = \boldsymbol{\alpha}_y^T \hat{\mathbf{P}}_m \boldsymbol{\alpha}_y$  and it is shown in [Kha10b] that the mean of  $\varepsilon_{y|m}$  for a given index  $i$  has the form

$$\boldsymbol{\mu}_{i,y} = \boldsymbol{\alpha}_y^T \hat{\mathbf{P}}_0 \mathbf{H}^T (\mathbf{H} \hat{\mathbf{P}}_0 \mathbf{H}^T)^{-1} (\mathbf{b}_B - \boldsymbol{\zeta}_i) \quad (5.19)$$

where  $\mathbf{H} \in \mathbb{Z}^{m \times n_x}$  is defined as  $\mathbf{H} = [\mathbf{Z}_{1:m}^T \quad \mathbf{0}^T]$ , with  $\mathbf{Z}_{1:m}^T \in \mathbb{Z}^{m \times n_a}$  being the first  $m$  rows of  $\mathbf{Z}^T$ .

Using these definitions,  $P(\varepsilon_y \in [-\ell_y, \ell_y] \mid \check{\mathbf{b}}_B = \boldsymbol{\zeta}_i)$  evaluates to

$$P(\varepsilon_y \in [-\ell_y, \ell_y] \mid \check{\mathbf{b}}_B = \boldsymbol{\zeta}_i) = \Phi\left(\frac{\ell_y - \boldsymbol{\mu}_{i,y}}{\hat{\sigma}_{y|m}}\right) + \Phi\left(\frac{\ell_y + \boldsymbol{\mu}_{i,y}}{\hat{\sigma}_{y|m}}\right) - 1 \quad (5.20)$$

Substituting equations 5.11 and 5.20 into equation 5.18 results in the following expression for integrity risk

$$\begin{aligned} \hat{I}_y = & 1 - \sum_{i=1}^n \left[ \Phi\left(\frac{\ell_y - \boldsymbol{\mu}_{i,y}}{\hat{\sigma}_{y|m}}\right) + \Phi\left(\frac{\ell_y + \boldsymbol{\mu}_{i,y}}{\hat{\sigma}_{y|m}}\right) - 1 \right] \\ & \prod_{j=1}^m \left[ \Phi\left(\frac{1 - 2\mathbf{l}_j^T (\mathbf{b}_B - \boldsymbol{\zeta}_i)}{2\hat{\sigma}_{\hat{\mathbf{b}}_{j|j-1}}}\right) + \Phi\left(\frac{1 + 2\mathbf{l}_j^T (\mathbf{b}_B - \boldsymbol{\zeta}_i)}{2\hat{\sigma}_{\hat{\mathbf{b}}_{j|j-1}}}\right) - 1 \right] \end{aligned} \quad (5.21)$$

Let  $\ell_j = 1/2$ ,  $\boldsymbol{\mu}_{i,j} = \mathbf{l}_j^T (\mathbf{b}_B - \boldsymbol{\zeta}_i)$ , and  $\hat{\sigma}_j = \hat{\sigma}_{\hat{\mathbf{b}}_{j|j-1}}$  for all  $j = 1, \dots, m$ . Also define

$\ell_{m+1} = \ell_y$ ,  $\boldsymbol{\mu}_{i,m+1} = \boldsymbol{\mu}_{i,y}$  and  $\hat{\sigma}_{m+1} = \hat{\sigma}_{y|m}$ . Then equation 5.21 can be re-written in the

compact form

$$\hat{I}_y = 1 - \sum_{i=1}^n \prod_{j=1}^{m+1} \left[ \Phi \left( \frac{\ell_j - \mu_{i,j}}{\sqrt{\hat{\sigma}_j^2}} \right) + \Phi \left( \frac{\ell_j + \mu_{i,j}}{\sqrt{\hat{\sigma}_j^2}} \right) - 1 \right] \quad (5.22)$$

Equation 5.22 indicates that the computed integrity risk is a function of the conditional variances  $\hat{\sigma}_1^2, \dots, \hat{\sigma}_{m+1}^2$ . However, these do not represent the true conditional variances because the Kalman filter (or WLS estimator) is using approximate models for the unknown measurement noise and disturbance input autocorrelation functions.

### 5.3 Natural Formulation of Mixed Integer Integrity Risk Bound

The fact that bootstrapping can be accounted for through a Kalman measurement update means that the results from Chapter 2 can be applied to each conditional variance. Specifically, the true conditional variance  $\sigma_j^2$  can be written in the form of equation 2.61 using the coefficient vector  $\boldsymbol{\alpha}_j^T = \mathbf{Z}_j^T$ . Defining  $\mathbf{r} \in \mathbb{R}^{n_r}$  as a vector containing all measurement noise and disturbance input autocorrelation function values,  $\sigma_j^2$  can be expressed in the generic form

$$\sigma_j^2 = c_j + \mathbf{d}_j^T \mathbf{r} \quad (5.23)$$

where  $c_j$  is a scalar and  $\mathbf{d}_j \in \mathbb{R}^{n_r}$ .

For the bounded autocorrelation uncertainty structure introduced in equations 2.64 and 2.65, the mixed integer integrity risk bound is given formally as the solution to the optimization problem

$$I_y^* = 1 - \min_{\mathbf{r}} \sum_{i=1}^n \prod_{j=1}^{m+1} \left[ \Phi \left( \frac{\ell_j - \mu_{i,j}}{\sqrt{c_j + \mathbf{d}_j^T \mathbf{r}}} \right) + \Phi \left( \frac{\ell_j + \mu_{i,j}}{\sqrt{c_j + \mathbf{d}_j^T \mathbf{r}}} \right) - 1 \right] \quad (5.24)$$

$$\text{s.t. } a_k \leq r_k \leq b_k, \quad k = 1, \dots, n_r$$

Equation 5.24 is the most natural form of the integrity risk bound because it is defined as an optimization problem directly in terms of what is unknown: the autocorrelation function values. In order for the integrity risk bound to be meaningful, it must be greater than or equal to the global optimal solution. However, global optimality of a candidate solution can only be guaranteed if the optimization problem is convex.

**5.3.1 Convexity Analysis.** One test for convexity involves using the fact that the composition of a convex function with an affine mapping preserves convexity [Boy04]. Let  $f(\mathbf{r})$  denote the cost function in equation 5.24, and suppose that  $f(\mathbf{r})$  is a convex function of  $\mathbf{r}$ . If this hypothesis is true, then  $f(\mathbf{g} + \mathbf{Q}\mathbf{r})$  must also be a convex function for every  $\mathbf{g}$  and  $\mathbf{Q}$ . Assume that  $\mathbf{r}$  is a  $3 \times 1$  vector and that  $\mathbf{Q}$  has the structure

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & q_{13} \\ 0 & 0 & q_{23} \\ 0 & 0 & q_{33} \end{bmatrix} \quad (5.25)$$

The composite function  $f(\mathbf{g} + \mathbf{Q}\mathbf{r})$  is formed by replacing  $c_j + \mathbf{d}_j^T \mathbf{r}$  with  $c_j + \mathbf{d}_j^T (\mathbf{g} + \mathbf{Q}\mathbf{r})$ , which, for the particular form of  $\mathbf{Q}$  given above, can be written as

$$c_j + \mathbf{d}_j^T (\mathbf{g} + \mathbf{Q}\mathbf{r}) = \gamma_j + \delta_j r_3 \quad (5.26)$$

Substituting equation 5.26 into the cost function of equation 5.24 results in

$$f(\mathbf{g} + \mathbf{Q}\mathbf{r}) = \sum_{i=1}^n \prod_{j=1}^{m+1} \left[ \Phi \left( \frac{\ell_j - \mu_{i,j}}{\sqrt{\gamma_j + \delta_j r_3}} \right) + \Phi \left( \frac{\ell_j + \mu_{i,j}}{\sqrt{\gamma_j + \delta_j r_3}} \right) - 1 \right] \quad (5.27)$$

For  $n=2$ ,  $m=1$ ,  $\ell_1=1/2$  and  $\ell_2=1$  with the parameter values contained in Table 5.1,  $f(\mathbf{g} + \mathbf{Q}\mathbf{r})$  has the form shown in Figure 5.1.

Table 5.1. Parameters Used for Convexity Analysis

$i$	$j$	$\mu_{i,j}$	$\gamma_j$	$\delta_j$
1	1	-1.72	-0.28	1.34
1	2	0.66	0.71	-0.20
2	1	0.79	—	—
2	2	-0.86	—	—

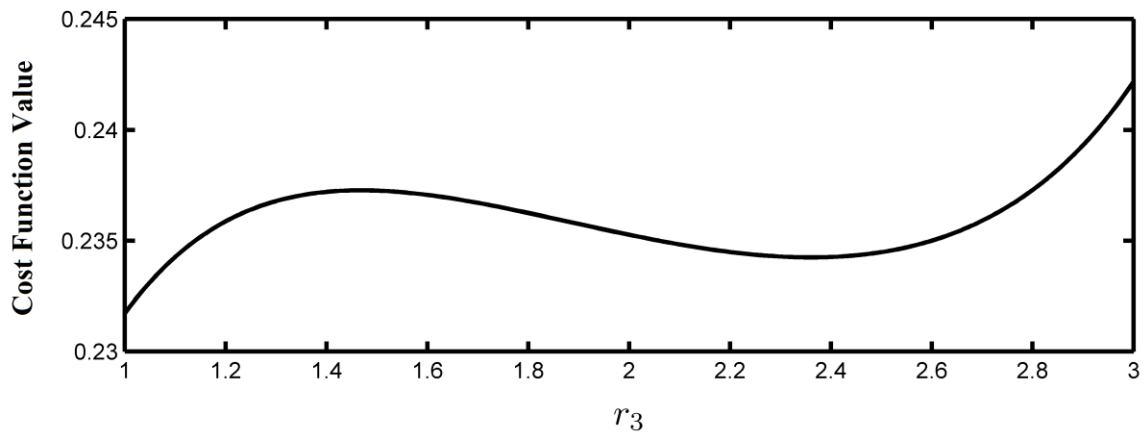


Figure 5.1. Behavior of Cost Function after Affine Composition

Figure 5.1 clearly shows that  $f(\mathbf{g} + \mathbf{Q}\mathbf{r})$  is not convex. Therefore,  $f(\mathbf{r})$  is not a convex function, implying that well-known optimization techniques will at best produce a local minimum. The only way to guarantee that a global solution has been found to a

non-convex optimization problem is through an exhaustive search scheme using specified grid points that span the feasible region. However, extensive analysis is required to ensure that the variation of the cost function between grid points is sufficiently small so that the global minimum has been determined with an acceptable level of accuracy. This difficulty coupled with the long computation times incurred from exhaustive search strategies generally makes them an undesirable solution. An alternative formulation will be developed in the next section that alleviates these deficiencies.

#### 5.4 Bounding Integrity Risk over a Hyper-Rectangular Region

In multi-sensor systems with multiple disturbance inputs,  $m \ll n_r$ . Therefore, it is more efficient to work in the  $\sigma^2$ –domain instead of the  $\mathbf{r}$ –domain. Using equation 5.23, the optimization problem in equation 5.24 takes the alternative form

$$I_y^* = 1 - \min_{\sigma^2} \sum_{i=1}^n \prod_{j=1}^{m+1} \left[ \Phi \left( \frac{\ell_j - \mu_{i,j}}{\sqrt{\sigma_j^2}} \right) + \Phi \left( \frac{\ell_j + \mu_{i,j}}{\sqrt{\sigma_j^2}} \right) - 1 \right] \quad (5.28)$$

s.t.  $\sigma^2 \in \Omega$

where  $\sigma^2 \in \mathbb{R}^{m+1}$  is defined as  $\sigma^2 = [\sigma_1^2 \ \sigma_2^2 \ \dots \ \sigma_{m+1}^2]^T$  and  $\Omega$  is the feasible region of  $\sigma^2$ .

Notice from equation 5.24 that if  $c_j = 0$  and  $\mathbf{d}_j$  is a vector of zeros except that the  $j^{\text{th}}$  element is 1, then  $f(\mathbf{r})$  has the same form as the cost function in equation 5.28. The special forms of  $c_j$  and  $\mathbf{d}_j$  have no impact on the convexity analysis given in Section 5.3.1. Therefore, equation 5.28 still constitutes a non-convex optimization problem. Mathematically defining  $\Omega$  is not a straightforward matter, and this will be

deferred until Section 5.6. In the present discussion, a simpler feasible region is used that contains  $\Omega$  within its interior to enable determination of an integrity risk bound  $\bar{I}_y^* \geq I_y^*$ .

Upper and lower bounds on  $\sigma_j^2$  can be obtained for the bounded autocorrelation uncertainty structure using the method provided in equations 2.66 and 2.67. Collectively, these bounds describe a hyper-rectangle in  $\mathbb{R}^{m+1}$  that circumscribes  $\Omega$ , which will be denoted by  $\bar{\Omega}$ . Defining

$$f_{i,j}(\sigma_j^2) = \Phi\left(\frac{\ell_j - \mu_{i,j}}{\sqrt{\sigma_j^2}}\right) + \Phi\left(\frac{\ell_j + \mu_{i,j}}{\sqrt{\sigma_j^2}}\right) - 1 \quad (5.29)$$

the integrity risk bound obtained from solving equation 5.28 over  $\bar{\Omega}$  is given by

$$\begin{aligned} \bar{I}_y^* &= 1 - \min_{\sigma^2} \sum_{i=1}^n \prod_{j=1}^{m+1} f_{i,j}(\sigma_j^2) \\ \text{s.t. } &\sigma^2 \in \bar{\Omega} \end{aligned} \quad (5.30)$$

An upper bound on  $\bar{I}_y^*$ , denoted by  $\bar{\bar{I}}_y^*$ , can be found by minimizing the product for each  $i$ , and then summing over  $i$ . Now, because each  $\sigma_j^2$  can vary independently over  $\bar{\Omega}$ , the minimum of the product is the product of minimums, leading to the new optimization problem

$$\begin{aligned} \bar{\bar{I}}_y^* &= 1 - \sum_{i=1}^n \prod_{j=1}^{m+1} \min_{\sigma_j^2} f_{i,j}(\sigma_j^2) \\ \text{s.t. } &l_j \leq \sigma_j^2 \leq u_j \end{aligned} \quad (5.31)$$

Appendix G shows that the global minimum of  $f_{i,j}(\sigma_j^2)$  must occur at either  $l_j$  or  $u_j$ . This useful fact allows  $\bar{\bar{I}}_y^*$  to be obtained in a quick and direct manner by simply setting  $\sigma_j^2$  equal to  $l_j$  or  $u_j$  for every index  $i$ .  $\bar{\bar{I}}_y^*$  is the most straightforward bound

considered in this chapter and will be used as a basis for comparison with more advanced techniques. A major deficiency of the method described above is that the optimal set of conditional variances will most likely be different for each  $i$ , when in fact there can only be one set. Therefore,  $\bar{I}_y^*$  has a propensity to be overly conservative.

**5.4.1 Rectangular Bound Using Subdivisions.** The fact that the global minimum of  $f_{i,j}(\sigma_j^2)$  must reside at either  $l_j$  or  $u_j$  is true regardless of the size of the interval  $[l_j, u_j]$ .  $\bar{\Omega}$  can be partitioned into small rectangular elements by subdividing each  $\sigma_j^2$  into  $N$  equally spaced segments. Solving equation 5.31 over the  $q^{\text{th}}$  rectangular element defines a local integrity risk bound  $\bar{I}_{y,q}^*$ . The global bound is then defined as the maximum value of the local bounds, i.e.

$$\bar{I}_{y,max}^* = \max(\bar{I}_{y,1}^*, \bar{I}_{y,2}^*, \dots, \bar{I}_{y,n_E}^*) \quad (5.32)$$

where  $n_E$  is the total number of rectangular elements.

The case  $n_E = 1$  corresponds to no partitioning of  $\bar{\Omega}$ , and therefore  $\bar{I}_{y,max}^* = \bar{I}_y^*$ . As  $n_E \rightarrow \infty$ , each rectangular element becomes a single point, and  $\bar{I}_{y,max}^* \rightarrow \bar{I}_y^*$ . Therefore, for any  $1 < n_E < \infty$ , the method of subdivisions will produce an integrity risk bound  $\bar{I}_{y,max}^*$  that is smaller than  $\bar{I}_y^*$ . In the next section, the two bounding methods developed above will be compared for a specific mixed real/integer estimation problem.

## 5.5 Static Baseline Estimation Application

Consider the problem illustrated in Figure 5.2. The baseline length  $p$  between stationary antennas A and B is to be estimated using carrier phase measurements from three GPS satellites in a circular orbit. The radius of the orbit (relative to the Earth's center O) is  $r_s$  and all satellites travel counterclockwise with constant speed  $v_s$ .

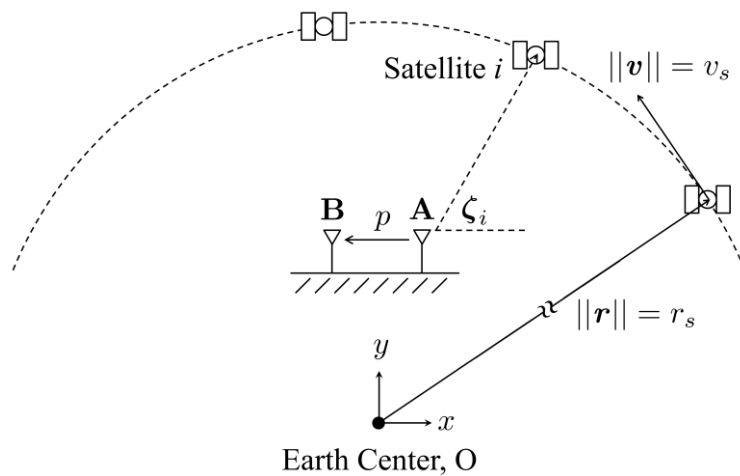


Figure 5.2. Problem Setup for Static Baseline Estimation

**5.5.1 Linearized Measurement Models.** This example will use the simplified carrier phase measurement models

$$z_{Ai,k} = \|\mathbf{x}_{i,k} - \mathbf{x}_A\| + b_{A,k} + \lambda a_{Ai} + v_{Ai,k} \quad (5.33)$$

$$z_{Bi,k} = \|\mathbf{x}_{i,k} - \mathbf{x}_B\| + b_{B,k} + \lambda a_{Bi} + v_{Bi,k} \quad (5.34)$$

where  $z_{Ai,k} \in \mathbb{R}$  is the carrier phase measurement received by antenna A from satellite  $i$  at time index  $k$ ,  $\mathbf{x}_{i,k} \in \mathbb{R}^2$  is the position vector from O to satellite  $i$ ,  $\mathbf{x}_A \in \mathbb{R}^2$  is the position vector from O to antenna A,  $b_{A,k} \in \mathbb{R}$  is the clock bias of the receiver

connected to antenna A,  $\lambda$  is the wavelength of the GPS carrier signal,  $a_{Ai} \in \mathbb{R}$  is the *integer* ambiguity associated with antenna A and satellite  $i$  and  $v_{Ai,k} \in \mathbb{R}$  is receiver measurement noise for antenna A and satellite  $i$  at time index  $k$ .

Similar definitions exist for the terms in equation 5.34. The vector norms appearing in equations 5.33 and 5.34 can be linearized using a first order Taylor series expansion, resulting in the linearized measurement models

$$z_{Ai,k} \approx \|\mathbf{x}_{i,k} - \mathbf{x}_A^*\| - \frac{\mathbf{x}_{i,k}^T - (\mathbf{x}_A^*)^T}{\|\mathbf{x}_{i,k} - \mathbf{x}_A^*\|} (\mathbf{x}_A - \mathbf{x}_A^*) + b_{A,k} + \lambda a_{Ai} + v_{Ai,k} \quad (5.35)$$

$$z_{Bi,k} \approx \|\mathbf{x}_{i,k} - \mathbf{x}_B^*\| - \frac{\mathbf{x}_{i,k}^T - (\mathbf{x}_B^*)^T}{\|\mathbf{x}_{i,k} - \mathbf{x}_B^*\|} (\mathbf{x}_B - \mathbf{x}_B^*) + b_{B,k} + \lambda a_{Bi} + v_{Bi,k} \quad (5.36)$$

where a superscript \* indicates an approximate quantity used for linearization.

Let  $\Delta\mathbf{x} \in \mathbb{R}^2$  be the vector originating at antenna A and terminating at antenna B. Then  $\mathbf{x}_B = \mathbf{x}_A + \Delta\mathbf{x}$ . The row vector  $\mathbf{x}_{i,k}^T - (\mathbf{x}_A^*)^T$  is a line-of-sight vector originating at antenna A and terminating at satellite  $i$ . Given that the antenna-to-satellite distance is many orders of magnitude larger than the antenna baseline length, it is reasonable to assume that  $\mathbf{x}_{i,k}^T - (\mathbf{x}_A^*)^T \approx \mathbf{x}_{i,k}^T - (\mathbf{x}_B^*)^T$  or, more simply, that  $\mathbf{x}_B^* = \mathbf{x}_A^*$ . Substituting these relations into equation 5.36 results in

$$z_{Bi,k} \approx \|\mathbf{x}_{i,k} - \mathbf{x}_A^*\| - \frac{\mathbf{x}_{i,k}^T - (\mathbf{x}_A^*)^T}{\|\mathbf{x}_{i,k} - \mathbf{x}_A^*\|} (\mathbf{x}_A + \Delta\mathbf{x} - \mathbf{x}_A^*) + b_{B,k} + \lambda a_{Bi} + v_{Bi,k} \quad (5.37)$$

Defining the line-of-sight unit vector

$$\hat{\mathbf{e}}_{i,k}^T = \frac{\mathbf{x}_{i,k}^T - (\mathbf{x}_A^*)^T}{\|\mathbf{x}_{i,k} - \mathbf{x}_A^*\|} \quad (5.38)$$

and subtracting equation 5.35 from equation 5.37 yields the single-difference (SD) measurement model

$$z_{Bi,k} - z_{Ai,k} = -\hat{\mathbf{e}}_{i,k}^T \Delta \mathbf{x} + (b_{B,k} - b_{A,k}) + \lambda(a_{Bi} - a_{Ai}) + (v_{Bi,k} - v_{Ai,k}) \quad (5.39)$$

At this point, the antenna subscripts A and B will be dropped, and equation 5.39 will be re-written as

$$z_{i,k} = -\hat{\mathbf{e}}_{i,k}^T \Delta \mathbf{x} + b_k + \lambda a_i + v_{i,k} \quad (5.40)$$

The clock bias term  $b_k$  can be eliminated by subtracting the SD carrier phase measurement corresponding to an arbitrary ‘master’ satellite  $j$  from  $z_{i,k}$ . This results in the double-difference (DD) carrier phase measurement model

$$z_{i,k} - z_{j,k} = -(\hat{\mathbf{e}}_{i,k}^T - \hat{\mathbf{e}}_{j,k}^T) \Delta \mathbf{x} + \lambda(a_i - a_j) + (v_{i,k} - v_{j,k}) \quad (5.41)$$

Defining  $z_{ij,k} = z_{i,k} - z_{j,k}$  and  $a_{ij} = a_i - a_j$ , equation 5.41 becomes

$$z_{ij,k} = -(\hat{\mathbf{e}}_{i,k}^T - \hat{\mathbf{e}}_{j,k}^T) \Delta \mathbf{x} + \lambda a_{ij} + (v_{i,k} - v_{j,k}) \quad (5.42)$$

For this example,  $\Delta \mathbf{x} = [-p \ 0]^T$ , where  $p$  is the unknown distance to be estimated. In addition, Figure 5.2 indicates that  $\hat{\mathbf{e}}_{i,k} = [\cos \zeta_{i,k} \ \sin \zeta_{i,k}]^T$ . Using these two facts, equation 5.42 takes the final form

$$z_{ij,k} = [\cos \zeta_{i,k} - \cos \zeta_{j,k}] p + \lambda a_{ij} + (v_{i,k} - v_{j,k}) \quad (5.43)$$

**5.5.2 Batch Measurement Model.** Suppose that the master satellite index  $j=3$ . Because the satellites can be ordered arbitrarily, there is no loss of generality in making this assumption. Defining  $\mathbf{z}_k \in \mathbb{R}^2$ ,  $\mathbf{H}_k \in \mathbb{R}^{2 \times 3}$ ,  $\boldsymbol{\theta} \in \mathbb{R}^3$ ,  $\mathbf{J}_k \in \mathbb{R}^{2 \times 3}$  and  $\mathbf{v}_k \in \mathbb{R}^3$  as

$$\mathbf{z}_k = [z_{13,k} \ z_{23,k}]^T \quad (5.44)$$

$$\mathbf{H}_k = \begin{bmatrix} \cos \zeta_{1,k} - \cos \zeta_{3,k} & 1 & 0 \\ \cos \zeta_{2,k} - \cos \zeta_{3,k} & 0 & 1 \end{bmatrix} \quad (5.45)$$

$$\boldsymbol{\theta} = [p \quad a_{13} \quad a_{23}]^T \quad (5.46)$$

$$\mathbf{J}_k = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \quad (5.47)$$

$$\mathbf{v}_k = [v_{1,k} \quad v_{2,k} \quad v_{3,k}]^T \quad (5.48)$$

allows the complete set of measurements at time index  $k$  to be written as

$$\mathbf{z}_k = \mathbf{H}_k \boldsymbol{\theta} + \mathbf{J}_k \mathbf{v}_k \quad (5.49)$$

The vector  $\boldsymbol{\theta}$  is estimated using a batch WLS estimator. Stacking measurements from  $k=1$  to  $k=q$  results in the batch measurement model

$$\mathbf{z}_Q = \mathbf{H}_Q \boldsymbol{\theta} + \mathbf{J}_Q \mathbf{v}_Q \quad (5.50)$$

where  $\mathbf{H}_Q \in \mathbb{R}^{2q \times 3}$ ,  $\mathbf{J}_Q \in \mathbb{R}^{2q \times 3q}$  and  $\mathbf{v}_Q \in \mathbb{R}^{3q}$  are defined as

$$\mathbf{H}_Q = [\mathbf{H}_1^T \quad \cdots \quad \mathbf{H}_q^T]^T \quad (5.51)$$

$$\mathbf{J}_Q = \text{diag}(\mathbf{J}_1, \dots, \mathbf{J}_q) \quad (5.52)$$

$$\mathbf{v}_Q = [\mathbf{v}_1^T \quad \cdots \quad \mathbf{v}_q^T]^T \quad (5.53)$$

**5.5.3 Autocorrelation Bounding Functions and Simulation Parameters.** One set of upper and lower bounding functions are used to bound the autocorrelation function associated with each satellite's SD measurement error. The bounds are defined using the same model given in equation 2.87. That is,

$$r_{v_i,kl} = \begin{cases} \sigma_\eta^2 + \sigma_r^2 & , \quad k = l \\ \sigma_\eta^2 \exp\left(-\frac{|k-l|\Delta t}{\tau_\eta}\right) & , \quad k \neq l \end{cases} \quad \text{for } i = 1, 2, 3 \quad (5.54)$$

The parameters defining the upper and lower bounds are provided in Table 5.2.

Table 5.2. Autocorrelation Bound Parameters for SD Measurement Error

Lower Bound	Upper Bound
$\sigma_r = 0.15 \text{ cm}$	$\sigma_r = 0.30 \text{ cm}$
$\sigma_\eta = 0.75 \text{ cm}$	$\sigma_\eta = 1.50 \text{ cm}$
$\tau_\eta = 200 \text{ sec}$	$\tau = 400 \text{ sec}$

It is assumed that GPS L1 carrier phase measurements are used to estimate  $\theta$ , which correspond to a wavelength of  $\lambda = 19.03 \text{ cm}$ . The batch WLS estimator is designed using the upper bound values from Table 5.2, and is run for 15 minutes. Additional parameters used for the simulation are summarized in Table 5.3.

Table 5.3. Simulation Parameters for Baseline Estimation Problem

Parameter	Assigned Value
Orbital Radius	$r_s = 26560 \text{ km}$
Satellite Speed	$v_s = 4 \text{ km/s}$
GPS Sampling Interval	$\Delta t = 60 \text{ sec}$
Initial Satellite Angles, $\zeta_i$	$20^\circ, 85^\circ, 120^\circ$
Baseline Error Acceptability Bound	$\ell_p = 1 \text{ m}$

There are two integer-valued states in this example. Hence, two stages of bootstrapping are required to form an integer estimate of the carrier phase cycle ambiguities.

**5.5.4 Analysis Results for First Bootstrapping Phase.** During the first stage of bootstrapping, the integrity risk bound for the baseline state  $p$  is determined using the five integer candidates

$$S = \{ 0, 1, 2, -1, -2 \} \quad (5.55)$$

The integrity risk bound  $\bar{I}_p^*$  is defined by setting  $\sigma_j^2$  equal to the values given in Table 5.4 for each candidate in  $S$ .

Table 5.4. Conditional Variance Sets that Produce  $\bar{I}_p^*$

Conditional Variance	Integer Candidate Index				
	1	2	3	4	5
$\sigma_1^2$	$u_1$	$l_1$	$l_1$	$l_1$	$l_1$
$\sigma_2^2$	$u_2$	$u_2$	$u_2$	$u_2$	$u_2$

By subdividing  $\sigma_1^2$  and  $\sigma_2^2$  into  $N$  equally spaced segments, a tighter integrity risk bound can be achieved using the method of subdivisions. Figures 5.3 and 5.4 show the percent reduction in the integrity risk bound relative to  $\bar{I}_p^*$  for small and large  $N$ . With  $N \geq 20$ , more than 90% reduction in the integrity risk bound is achieved, which represents a full order of magnitude reduction. For this example, subdividing  $\bar{\Omega}$  provides substantial improvement in the integrity risk bound.

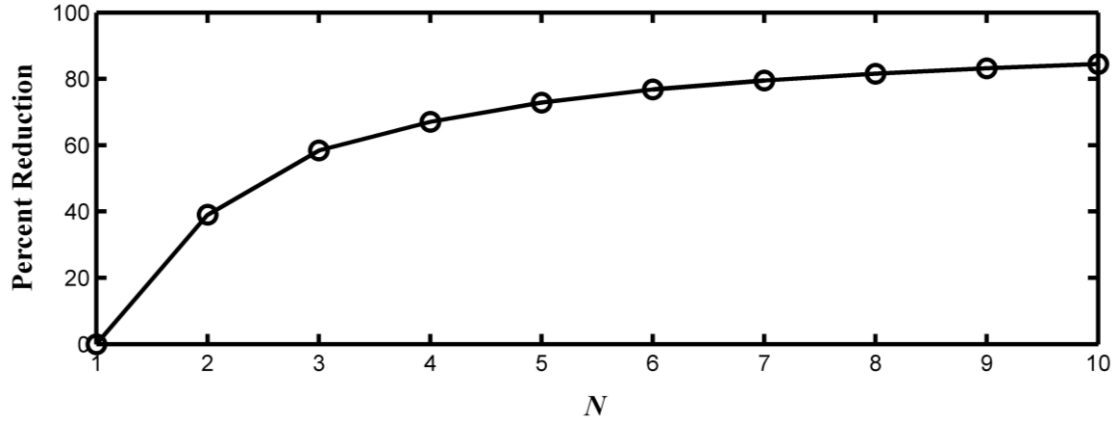


Figure 5.3. Bound Reduction Achieved Using Subdivisions (Small  $N$ )

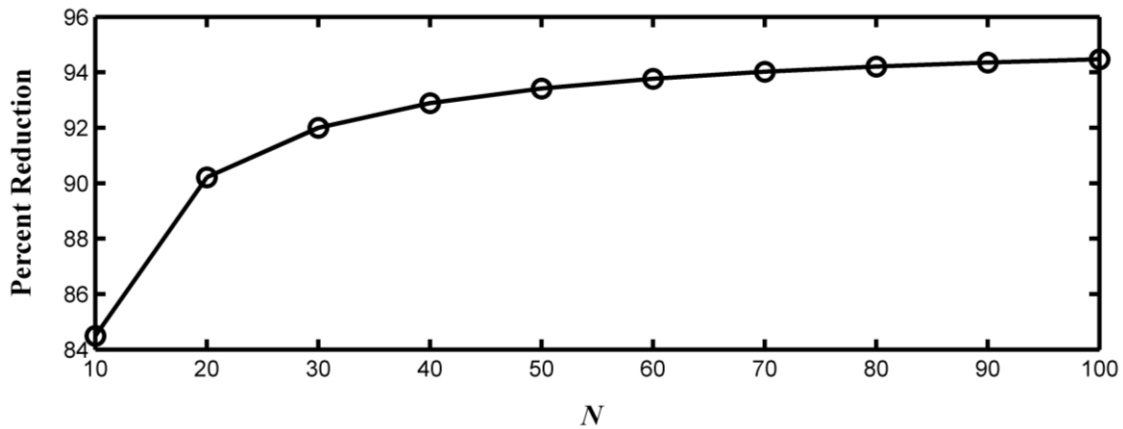


Figure 5.4. Bound Reduction Achieved Using Subdivisions (Large  $N$ )

**5.5.5 Analysis Results for Second Bootstrapping Phase.** After two stages of bootstrapping, the integrity risk bound is determined using the seven integer candidate vectors

$$S = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix} \right\} \quad (5.56)$$

which are selected independently of  $S$  given in equation 5.55.

Table 5.5 contains the values of  $\sigma_j^2$  for each candidate vector that produce the integrity risk bound  $\bar{I}_p^*$ .

Table 5.5. Conditional Variance Sets that Produce  $\bar{I}_p^*$

Conditional Variance	Integer Candidate Index						
	1	2	3	4	5	6	7
$\sigma_1^2$	$u_1$	$u_1$	$l_1$	$l_1$	$u_1$	$l_1$	$l_1$
$\sigma_2^2$	$u_2$	$l_2$	$u_2$	$l_2$	$l_2$	$u_2$	$l_2$
$\sigma_3^2$	$u_3$	$u_3$	$u_3$	$u_3$	$u_3$	$u_3$	$u_3$

Figures 5.5 and 5.6 show the percent reduction in the integrity risk bound relative to  $\bar{I}_p^*$  for small and large  $N$ .

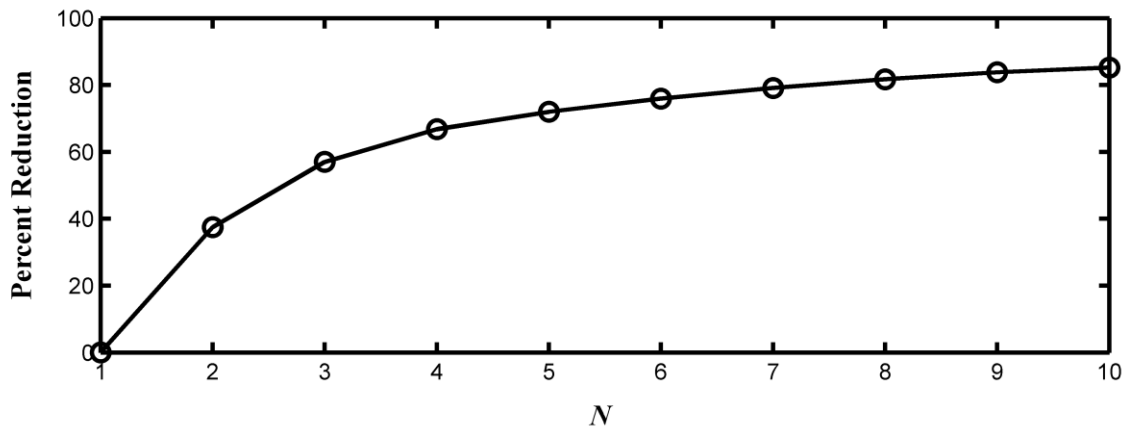


Figure 5.5. Bound Reduction Achieved Using Subdivisions (Small  $N$ )

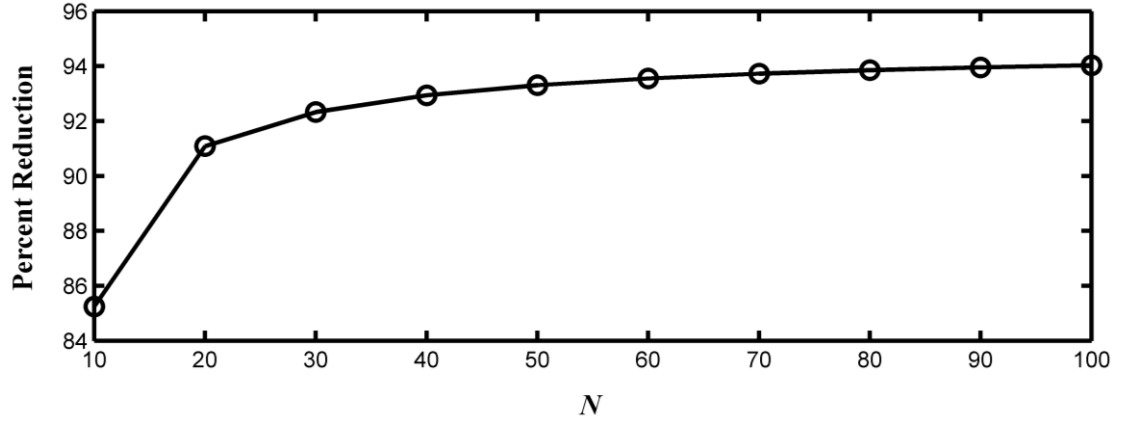


Figure 5.6. Bound Reduction Achieved Using Subdivisions (Large  $N$ )

These figures lead to the same conclusion that the potential reduction in the integrity risk bound achieved by the method of subdivisions is significant. However, subdivisions cannot eliminate or even reduce the conservatism introduced into the bound by approximating the polytope by a circumscribing hyper-rectangular region. Therefore, if smaller subdivisions do not reduce the integrity risk bound sufficiently, the only avenue for reduction is to define the bound over the true polytopic feasible region.

## 5.6 Bounding Integrity Risk over Polytopic Region

It is of practical and theoretical importance to understand how much conservatism, if any, is introduced into the bound by approximating  $\Omega$  by  $\bar{\Omega}$ . This section develops a new method to upper bound integrity risk over  $\Omega$ .

**5.6.1 Determination of Feasible Region.** Equation 5.23 shows that each conditional variance is a linear function of  $r$ . Collecting all of the conditional variances in one vector  $\sigma^2$  results in the linear system of equations

$$\boldsymbol{\sigma}^2 = \mathbf{c} + \mathbf{D}\mathbf{r} \quad (5.57)$$

where  $\boldsymbol{\sigma}^2 \in \mathbb{R}^{m+1}$ ,  $\mathbf{c} \in \mathbb{R}^{m+1}$  and  $\mathbf{D} \in \mathbb{R}^{(m+1) \times n_r}$ .

Equation 5.57 describes an affine transformation from the  $\mathbf{r}$ -domain to the  $\boldsymbol{\sigma}^2$ -domain. Given that the feasible region of  $\mathbf{r}$  is a closed, convex hyper-box in  $\mathbb{R}^{n_r}$  and the fact that affine transformations preserve convexity [Boy04], it follows that  $\Omega$  is a closed, convex polytope in  $\mathbb{R}^{m+1}$ . A closed, convex polytope can be described by a set of linear inequalities [Lue08]. Therefore,  $\Omega$  is defined mathematically as

$$\Omega = \{ \boldsymbol{\sigma}^2 \mid \mathbf{g}_k^T \boldsymbol{\sigma}^2 \leq \beta_k, k = 1, \dots, n_e \} \quad (5.58)$$

An efficient method to determine  $\mathbf{g}_k$  and  $\beta_k$  is derived in Appendix H, where it is also shown that the number of inequalities necessary to define  $\Omega$  obeys the relation

$$n_e = 2 + 2 \sum_{k=1}^m C(n_r - m + k - 1, k) \quad (5.59)$$

where  $C(n, r)$  denotes how many ways  $n$  objects can be taken  $r$  at a time. That is

$$C(n, r) = \frac{n!}{r!(n-r)!} \quad (5.60)$$

The integrity risk bound from equation 5.28 can now be stated more precisely as

$$I_y^* = 1 - \min_{\boldsymbol{\sigma}^2} \sum_{i=1}^n \prod_{j=1}^{m+1} \left[ \Phi \left( \frac{\ell_j - \mu_{i,j}}{\sqrt{\sigma_j^2}} \right) + \Phi \left( \frac{\ell_j + \mu_{i,j}}{\sqrt{\sigma_j^2}} \right) - 1 \right] \quad (5.61)$$

s.t.  $\mathbf{g}_k^T \boldsymbol{\sigma}^2 \leq \beta_k, k = 1, \dots, n_e$

**5.6.2 Partitioning  $\Omega$  into Rectangular Elements.** An upper bound on  $I_y^*$  can be obtained by partitioning  $\Omega$  into small rectangular regions, just as was done in Section

5.4. Figure 5.7 shows an example of what this partitioned region might look like when  $m=1$  and  $n_r=3$ .

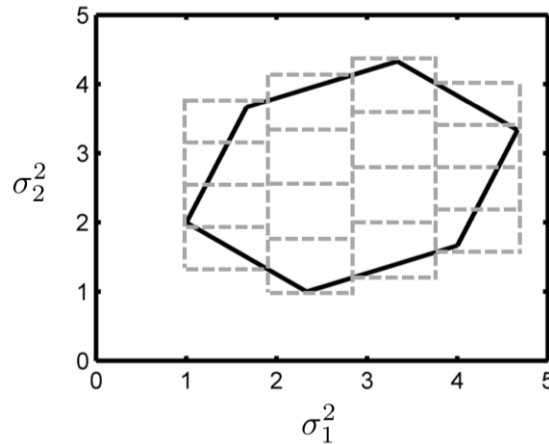


Figure 5.7. Example of Partitioned Polytopic Feasible Region

One approach for generating these elements begins by partitioning  $[l_1, u_1]$  into  $N$  equally spaced segments. For the  $p^{\text{th}}$  segment, the minimum value of  $\sigma_2^2$  is determined by solving the optimization problem

$$\begin{aligned}
 l_{2,p} &= \min_{\sigma^2} \sigma_2^2 \\
 \text{s.t. } & \mathbf{g}_k^T \boldsymbol{\sigma}^2 \leq \beta_k, \quad k=1, \dots, n_e \\
 & l_{1,p} \leq \sigma_1^2 \leq u_{1,p} \\
 & \boldsymbol{\sigma}^2 \geq 0
 \end{aligned} \tag{5.62}$$

and the maximum value  $u_{2,p}$  is found by minimizing  $-\sigma_2^2$ .

The interval  $[l_{2,p}, u_{2,p}]$  can be partitioned into  $N$  equally spaced segments, and the lower bound on  $\sigma_3^2$  associated with  $[l_{1,p}, u_{1,p}]$  and  $[l_{2,q}, u_{2,q}]$  is determined by solving the optimization problem

$$\begin{aligned}
l_{3,pq} &= \min_{\sigma^2} \sigma_3^2 \\
\text{s.t. } & \mathbf{g}_k^T \boldsymbol{\sigma}^2 \leq \beta_k, \quad k=1, \dots, n_e \\
& l_{1,p} \leq \sigma_1^2 \leq u_{1,p} \\
& l_{2,q} \leq \sigma_2^2 \leq u_{2,q} \\
& \boldsymbol{\sigma}^2 \geq 0
\end{aligned} \tag{5.63}$$

Minimizing  $-\sigma_3^2$  instead of  $\sigma_3^2$  produces the upper bound  $u_{3,pq}$ .

The interval  $[l_{3,pq}, u_{3,pq}]$  can be partitioned into  $N$  equally spaced segments, and the process above can be continued to determine hyper-rectangular elements that encompass  $\Omega$ .

Equations 5.62 and 5.63 can each be written in the form

$$\begin{aligned}
y^* &= \min_{\mathbf{x}} \mathbf{c}^T \mathbf{x} \\
\text{s.t. } & \mathbf{a}_j^T \mathbf{x} + s_j = b_j, \quad j=1, \dots, q \\
& \mathbf{x} \geq 0, \quad \mathbf{s} \geq 0
\end{aligned} \tag{5.64}$$

which is recognized as the standard form of a linear programming (LP) problem.

Numerous algorithms exist to solve LP problems, and this dissertation uses the active set method described in [Gil84]. After determining the  $n_E$  rectangular elements, the integrity risk bound  $I_{y,max}^*$  is defined as the maximum value of the individual bounds associated with each element.

**5.6.3 Efficacy of the Polytopic Bound.** Suppose that the element  $S_r$  producing  $\bar{I}_{y,max}^*$  resides within the interior of  $\Omega$ . In this case, the polytopic bound will not provide any improvement over the rectangular bound, and  $I_{y,max}^* = \bar{I}_{y,max}^*$ . One way to ascertain if

$I_{y,max}^* = \bar{I}_{y,max}^*$  is to simply compute both bounds and compare their values. However, this approach is unnecessarily time consuming if the bounds do happen to be identical. A more efficient method involves determining if the intervals  $[\bar{l}_j, \bar{u}_j]$  defining  $S_r$  are simultaneously feasible in  $\Omega$ .

It is already known that  $[\bar{l}_1, \bar{u}_1] \in \Omega$  because  $\bar{\Omega}$  circumscribes  $\Omega$ . The minimum permissible value of  $\sigma_2^2$  associated with  $[\bar{l}_1, \bar{u}_1]$  is determined by solving the LP problem

$$\begin{aligned}
 l_2^* &= \min_{\sigma^2} \sigma_2^2 \\
 \text{s.t. } & \mathbf{g}_k^T \boldsymbol{\sigma}^2 \leq \beta_k, \quad k=1, \dots, n_e \\
 & \bar{l}_1 \leq \sigma_1^2 \leq \bar{u}_1 \\
 & \boldsymbol{\sigma}^2 \geq 0
 \end{aligned} \tag{5.65}$$

and the maximum permissible value  $u_2^*$  is attained by minimizing  $-\sigma_2^2$ .

If  $\bar{l}_2 \geq l_2^*$  and  $\bar{u}_2 \leq u_2^*$ , then  $[\bar{l}_2, \bar{u}_2]$  is simultaneously feasible with  $[\bar{l}_1, \bar{u}_1]$ . The minimum permissible value of  $\sigma_3^2$  associated with  $[\bar{l}_1, \bar{u}_1]$  and  $[\bar{l}_2, \bar{u}_2]$  is determined by solving the LP problem

$$\begin{aligned}
 l_3^* &= \min_{\sigma^2} \sigma_3^2 \\
 \text{s.t. } & \mathbf{g}_k^T \boldsymbol{\sigma}^2 \leq \beta_k, \quad k=1, \dots, n_e \\
 & \bar{l}_1 \leq \sigma_1^2 \leq \bar{u}_1 \\
 & \bar{l}_2 \leq \sigma_2^2 \leq \bar{u}_2 \\
 & \boldsymbol{\sigma}^2 \geq 0
 \end{aligned} \tag{5.66}$$

and the maximum permissible value  $u_3^*$  is obtained by minimizing  $-\sigma_3^2$ .

If  $\bar{l}_3 \geq l_3^*$  and  $\bar{u}_3 \leq u_3^*$ , then  $[\bar{l}_1, \bar{u}_1]$ ,  $[\bar{l}_2, \bar{u}_2]$  and  $[\bar{l}_3, \bar{u}_3]$  are simultaneously feasible. This process can be continued to determine if the intervals  $[\bar{l}_j, \bar{u}_j]$  are concurrently feasible in  $\Omega$ . Whether or not the polytopic approach has any potential for bound reduction can therefore be ascertained by solving up to  $2m$  LP problems.

**5.7 Comparison of Rectangular and Polytopic Bounding Methods.** This section compares the rectangular and polytopic bounds for the static baseline estimation application. The rectangular bound is fundamentally conservative because it replaces  $\Omega$  with  $\bar{\Omega}$  and because it uses finite size rectangular elements to subdivide  $\bar{\Omega}$ . The polytopic bound eliminates the former source of conservatism, suggesting that it can potentially achieve the same value as the rectangular bound using larger (i.e., fewer) rectangular elements. To illustrate, consider Figure 5.8, which shows  $\Omega$  and  $\bar{\Omega}$  after one bootstrapping step.

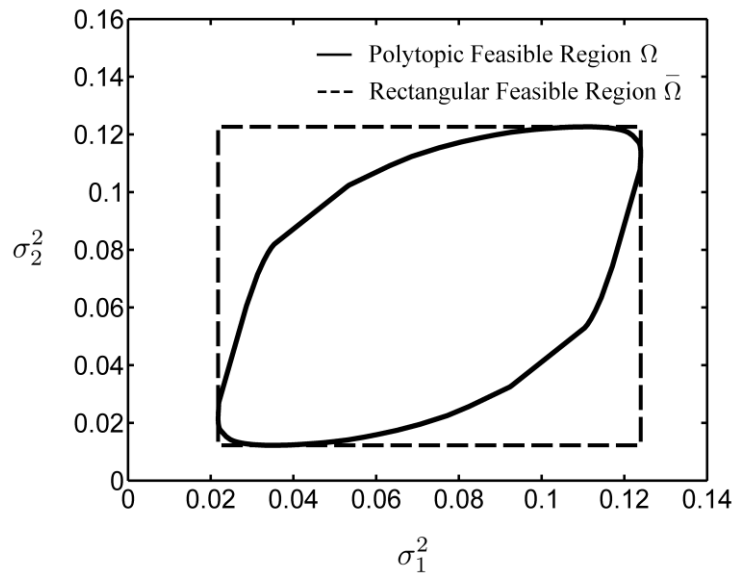


Figure 5.8. Feasible Regions after One Bootstrapping Step

If  $\bar{I}_{y,max}^*$  is determined from an element in the upper left or lower right regions of  $\bar{\Omega}$ , then the polytopic bound will achieve the same value using larger rectangular elements because it will not consider these infeasible regions. For comparison purposes, let the rectangular bound  $\bar{I}_{p,max}^*$  be fixed using  $N=100$  and let the polytopic bound  $I_{p,max}^*$  be defined for different numbers of segments,  $N_p$ . Figures 5.9 and 5.10 show the percent reduction in the integrity risk bound relative to  $\bar{I}_{p,max}^*$  achieved by using  $I_{p,max}^*$ .

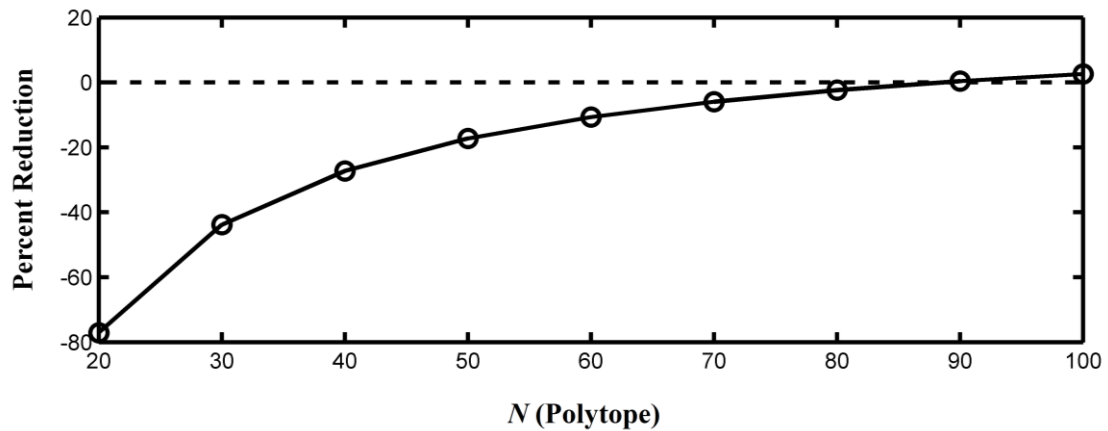


Figure 5.9. Polytopic Bound Performance after One Bootstrapping Phase

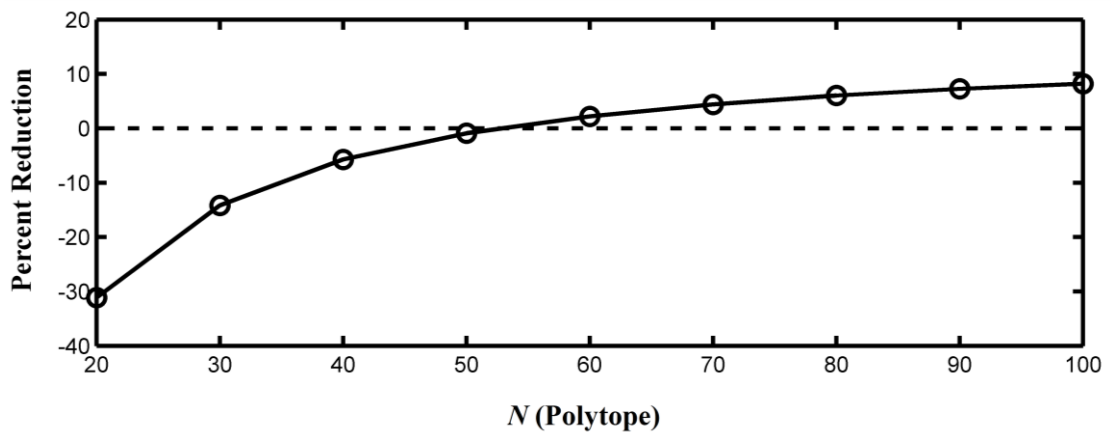


Figure 5.10. Polytopic Bound Performance after Two Bootstrapping Phases

Figure 5.9 shows that the polytopic bound achieves the same value as  $\bar{I}_{p,max}^*$  when  $N_p = 90$ . Hence, after one phase of bootstrapping, the polytopic bound requires almost the same number of elements as the rectangular bound. After two bootstrapping phases, Figure 5.10 indicates that  $I_{p,max}^*$  determined using  $N_p = 50$  achieves the same value as  $\bar{I}_{p,max}^*$  using  $N = 100$ . Hence, the polytopic bound has to consider 87.5% fewer rectangular elements ( $50^3$  versus  $100^3$ ). If both bounds use 100 segments, it is evident from Figure 5.10 that  $I_{p,max}^*$  reduces the integrity risk bound by 9% relative to  $\bar{I}_{p,max}^*$ .

Even though the polytopic bound is significantly more computationally demanding than the rectangular bound, the reduction in bound magnitude and number of elements can make up for this deficiency when  $\sigma^2$  is higher dimensional. Approaches to improve the efficiency of the polytopic bound are discussed at the conclusion of this dissertation.

## CHAPTER 6

### CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

Quantifying integrity risk is essential in safety critical estimation applications. When state estimation is accomplished by filtering measurements over time, accurate assessments of integrity risk are generally not possible due to uncertainty in the stochastic models of measurement noise and disturbance inputs. In this case, an upper bound on integrity risk must be obtained. This dissertation developed the first implementable methods to upper bound integrity risk when the measurement noise and disturbance inputs have a bounded autocorrelation uncertainty structure. These methods were derived for both real-valued and mixed real/integer estimation applications that use the Kalman filter, batch WLS estimator and integer bootstrap estimator.

#### **6.1 Summary of Contributions**

A main objective of this work has been to derive two classes of bounding methods. One that uses simplifying approximations to obtain a conservative bound quickly and efficiently, and another that uses more advanced mathematical techniques to define a tight bound on integrity risk. Below is a summary of this dissertation's contributions.

**6.1.1 Integrity Risk Bounding in Kalman Filtering.** A new set of linear difference equations was derived in Chapter 2 that enabled determination of how measurement noise and disturbance input autocorrelation functions map into the state estimate error variance. The resulting expressions were used to develop a fast method to determine the worst-case

autocorrelation functions that lead to an upper bound on integrity risk. A one-dimensional position and velocity estimation application was used to illustrate this method.

**6.1.2 Bounding Integrity Risk in Batch Estimation.** Chapter 3 derived a new form of the batch WLS estimate error vector when the weighting matrix has unknown but bounded elements. This expression allowed the methods from Chapter 2 to be used to define an integrity risk bound for the batch WLS estimator. The method was demonstrated for a two-dimensional position and velocity estimation application.

**6.1.3 Real-Valued Integrity Risk Bounding via Semi-Definite Optimization.** The integrity risk bound was formally defined in Chapter 4 as a semi-definite programming problem that was subsequently solved using a primal-dual interior point algorithm. Formulating the integrity risk bound this way allowed enforcement of the constraint that the autocorrelation functions must be positive semi-definite in addition to being bounded. Performance comparisons were made for the two-dimensional estimation problem.

**6.1.4 Mixed Real/Integer Integrity Risk Bounding.** For the integer bootstrap estimator, the mixed integer integrity risk bound was defined in Chapter 5 as a non-convex optimization problem over a polytope. Two methods were developed to define a conservative integrity risk bound by solving a relaxed version of the optimization problem in which the polytopic region is replaced by a circumscribing hyper-rectangle. The performance of these methods was compared for a static baseline estimation application.

**6.1.5 Mixed Real/Integer Integrity Risk Bounding over a Polytope.** A third approach was derived in Chapter 5 to upper bound the mixed real/integer integrity risk by solving the non-convex optimization problem over the polytope. This required creation of a new procedure to determine the set of inequalities that define the polytope, which was used in conjunction with linear programming to define the integrity risk bound. Simulation results and comparison studies were made with the rectangular bound for the static baseline estimation problem.

## **6.2 Recommended Topics for Future Research**

**6.2.1 Autocorrelation Bound Determination.** A fundamental assumption throughout this dissertation has been that the autocorrelation bounding functions are given. A method must be devised to determine these bounds either through experimental or analytical means. Experimental methods will most likely produce bounds that do not contain the true autocorrelation function with probability one. To address this issue, the methods developed in this dissertation must be adapted to accommodate ‘soft’ bounds that are only known to contain the autocorrelation functions with a specified probability.

**6.2.2 Minimize Integrity Risk Bound by Exploiting Estimator Structure.** All of the integrity risk bounds in this dissertation have been derived for a pre-defined estimator. However, the magnitude of the integrity risk bound is dependent on the estimator design, which provides an interesting topic for future research: examine how estimator structure can be exploited to produce the minimum upper bound on integrity risk.

**6.2.3 Improving Efficiency of the Polytopic Integrity Risk Bound.** The polytopic bound has the potential to significantly reduce the magnitude of the rectangular bound. However, it is computationally intensive due to the large number of inequality constraints needed to define the polytope. One way to improve efficiency is to compute the bound using an approximate polytopic region defined by a small set of linear inequality constraints. How this set of inequalities is chosen and the impact on the magnitude of the integrity risk bound is a topic for future research.

APPENDIX A  
GENERAL SOLUTION TO LINEAR SYSTEM  
OF DIFFERENCE EQUATIONS

This appendix derives the general solution to the following linear system of difference equations

$$\mathbf{e}_k = \mathbf{L}_k \mathbf{e}_k^- + \mathbf{M}_k \mathbf{v}_k \quad (\text{A.1})$$

$$\mathbf{e}_{k+1}^- = \mathbf{F}_k \mathbf{e}_k + \mathbf{N}_k \mathbf{w}_k \quad (\text{A.2})$$

When  $k=0$ , equation A.1 can be written in the form

$$\mathbf{e}_0 = \mathbf{\Phi}_0 \mathbf{e}_0^- + \sum_{l=1}^{n_v} \mathbf{\Lambda}_{l,0} \nu_{l,0} \quad (\text{A.3})$$

where  $\mathbf{\Phi}_0 = \mathbf{L}_0$ ,  $\mathbf{\Lambda}_{l,0} \in \mathbb{R}^{n_x}$  is equal to the  $l^{\text{th}}$  column of  $\mathbf{M}_0$  (i.e.  $\mathbf{\Lambda}_{l,0} = \mathbf{M}_0(:, l)$ ) and  $\nu_{l,0}$  is a scalar corresponding to the  $l^{\text{th}}$  component of  $\mathbf{v}_0$ .

Similarly, when  $k=0$ , equation A.2 can be written as

$$\mathbf{e}_1^- = \mathbf{F}_0 \mathbf{e}_0 + \sum_{j=1}^{n_w} \mathbf{N}_0(:, j) w_{j,0} \quad (\text{A.4})$$

where  $\mathbf{N}_0(:, j)$  is an  $n_x \times 1$  vector corresponding to the  $j^{\text{th}}$  column of  $\mathbf{N}_0$  and  $w_{j,0}$  is a scalar corresponding to the  $j^{\text{th}}$  component of  $\mathbf{w}_0$ . Replacing  $\mathbf{e}_0$  by the expression on the right hand side of equation A.3 results in

$$\mathbf{e}_1^- = \mathbf{F}_0 \mathbf{\Phi}_0 \mathbf{e}_0^- + \sum_{l=1}^{n_v} \mathbf{F}_0 \mathbf{\Lambda}_{l,0} \nu_{l,0} + \sum_{j=1}^{n_w} \mathbf{N}_0(:, j) w_{j,0} \quad (\text{A.5})$$

Defining the  $n_x \times n_x$  matrix  $\mathbf{\Phi}_1^- = \mathbf{F}_0 \mathbf{\Phi}_0$ , the  $n_x \times 1$  vector  $\mathbf{\Lambda}_{l,1}^- = \mathbf{F}_0 \mathbf{\Lambda}_{l,0}$  and the  $n_x \times 1$  vector  $\mathbf{\Gamma}_{j,1}^- = \mathbf{N}_0(:, j)$  allows equation A.5 to be expressed more compactly as

$$\mathbf{e}_1^- = \mathbf{\Phi}_1^- \mathbf{e}_0^- + \sum_{l=1}^{n_v} \mathbf{\Lambda}_{l,1}^- \nu_{l,0} + \sum_{j=1}^{n_w} \mathbf{\Gamma}_{j,1}^- w_{j,0} \quad (\text{A.6})$$

When  $k=1$ , equation A.1 becomes

$$\mathbf{e}_1 = \mathbf{L}_1 \mathbf{e}_1^- + \sum_{l=1}^{n_v} \mathbf{M}_1(:, l) v_{l,1} \quad (\text{A.7})$$

where  $\mathbf{M}_1(:, l)$  is an  $n_x \times 1$  vector corresponding to the  $l^{\text{th}}$  column of  $\mathbf{M}_1$  and  $v_{l,1}$  is a scalar corresponding to the  $l^{\text{th}}$  component of  $\mathbf{v}_1$ . Replacing  $\mathbf{e}_1^-$  by the expression on the right hand side of equation A.6 results in

$$\mathbf{e}_1 = \mathbf{L}_1 \Phi_1^- \mathbf{e}_0^- + \sum_{l=1}^{n_v} \mathbf{L}_1 \Lambda_{l,1}^- v_{l,0} + \sum_{j=1}^{n_w} \mathbf{L}_1 \Gamma_{j,1}^- w_{j,0} + \sum_{l=1}^{n_v} \mathbf{M}_1(:, l) v_{l,1} \quad (\text{A.8})$$

Defining  $\Phi_1 \in \mathbb{R}^{n_x \times n_x}$  as  $\Phi_1 = \mathbf{L}_1 \Phi_1^-$ ,  $\Lambda_{l,1} \in \mathbb{R}^{n_x \times 2}$  as  $\Lambda_{l,1} = [\mathbf{L}_1 \Lambda_{l,1}^- | \mathbf{M}_1(:, l)]$ ,  $\Gamma_{j,1} \in \mathbb{R}^{n_x}$  as  $\Gamma_{j,1} = \mathbf{L}_1 \Gamma_{j,1}^-$  and the  $2 \times 1$  vector  $\{v_l\}_0^1 = [v_{l,0} \ v_{l,1}]^T$  allows equation A.8 to be expressed more compactly as

$$\mathbf{e}_1 = \Phi_1 \mathbf{e}_0^- + \sum_{l=1}^{n_v} \Lambda_{l,1} \{v_l\}_0^1 + \sum_{j=1}^{n_w} \Gamma_{j,1} w_{j,0} \quad (\text{A.9})$$

Similarly, when  $k=1$ , equation A.2 becomes

$$\mathbf{e}_2^- = \mathbf{F}_1 \mathbf{e}_1 + \sum_{j=1}^{n_w} \mathbf{N}_1(:, j) w_{j,1} \quad (\text{A.10})$$

where  $\mathbf{N}_1(:, j)$  is an  $n_x \times 1$  vector corresponding to the  $j^{\text{th}}$  column of  $\mathbf{N}_1$  and  $w_{j,1}$  is a scalar corresponding to the  $j^{\text{th}}$  component of  $\mathbf{w}_1$ . Replacing  $\mathbf{e}_1$  by the expression on the right hand side of equation A.9 results in

$$\mathbf{e}_2^- = \mathbf{F}_1 \Phi_1 \mathbf{e}_0^- + \sum_{l=1}^{n_v} \mathbf{F}_1 \Lambda_{l,1} \{v_l\}_0^1 + \sum_{j=1}^{n_w} \mathbf{F}_1 \Gamma_{j,1} w_{j,0} + \sum_{j=1}^{n_w} \mathbf{N}_1(:, j) w_{j,1} \quad (\text{A.11})$$

Defining  $\Phi_2^- \in \mathbb{R}^{n_x \times n_x}$  as  $\Phi_2^- = \mathbf{F}_1 \Phi_1^-$ ,  $\Lambda_{l,2}^- \in \mathbb{R}^{n_x \times 2}$  as  $\Lambda_{l,2}^- = \mathbf{F}_1 \Lambda_{l,1}^-$ ,  $\Gamma_{j,2}^- \in \mathbb{R}^{n_x \times 2}$  as

$\Gamma_{j,2}^- = [\mathbf{F}_1 \Gamma_{j,1}^- | \mathbf{N}_1(:, j)]$  and the  $2 \times 1$  vector  $\{w_j\}_0^1 = [w_{j,0} \ w_{j,1}]^T$  allows equation

A.11 to be expressed more compactly as

$$\mathbf{e}_2^- = \Phi_2^- \mathbf{e}_0^- + \sum_{l=1}^{n_v} \Lambda_{l,2}^- \{v_l\}_0^1 + \sum_{j=1}^{n_w} \Gamma_{j,2}^- \{w_j\}_0^1 \quad (\text{A.12})$$

This process can be continued, resulting in the expressions

$$\mathbf{e}_k = \Phi_k \mathbf{e}_0^- + \sum_{j=1}^{n_w} \Gamma_{j,k} \{w_j\}_0^{k-1} + \sum_{l=1}^{n_v} \Lambda_{l,k} \{v_l\}_0^k \quad (\text{A.13})$$

$$\mathbf{e}_k^- = \Phi_k^- \mathbf{e}_0^- + \sum_{j=1}^{n_w} \Gamma_{j,k}^- \{w_j\}_0^{k-1} + \sum_{l=1}^{n_v} \Lambda_{l,k}^- \{v_l\}_0^{k-1} \quad (\text{A.14})$$

where  $\Phi_k$  and  $\Phi_k^-$  are  $n_x \times n_x$  matrices,  $\Gamma_{j,k}$  and  $\Gamma_{j,k}^-$  are  $n_x \times k$  matrices,  $\Lambda_{l,k}$  is an  $n_x \times (k+1)$  matrix,  $\Lambda_{l,k}^-$  is an  $n_x \times k$  matrix,  $\{w_j\}_0^{k-1}$  is a  $k \times 1$  time series vector for the  $j^{\text{th}}$  component of  $\mathbf{w}$  from time index 0 to time index  $k-1$ ,  $\{v_l\}_0^{k-1}$  is a  $k \times 1$  time series vector for the  $l^{\text{th}}$  component of  $\mathbf{v}$  from time index 0 to time index  $k-1$  and  $\{v_l\}_0^k$  is a  $(k+1) \times 1$  time series vector for the  $l^{\text{th}}$  component of  $\mathbf{v}$  from time index 0 to time index  $k$ .

The matrices in equations A.13 and A.14 are defined as

$$\Phi_k = \mathbf{L}_k \Phi_k^- \quad , \quad \Phi_0^- = \mathbf{I} \quad (\text{A.15})$$

$$\Lambda_{l,k} = [\mathbf{L}_k \Lambda_{l,k}^- | \mathbf{M}_k(:, l)] \quad , \quad \Lambda_{l,0}^- = [] \quad (\text{A.16})$$

$$\Gamma_{j,k} = \mathbf{L}_k \Gamma_{j,k}^- \quad , \quad \Gamma_{j,0}^- = [] \quad (\text{A.17})$$

$$\Phi_k^- = \mathbf{F}_{k-1} \Phi_{k-1}^- \quad , \quad \Phi_0^- = \mathbf{I} \quad (\text{A.18})$$

$$\Lambda_{l,k}^- = \mathbf{F}_{k-1} \Lambda_{l,k-1}^- \quad (\text{A.19})$$

$$\Gamma_{j,k}^- = [\mathbf{F}_{k-1} \Gamma_{j,k-1}^- | \mathbf{N}_{k-1}(:, j)] \quad (\text{A.20})$$

APPENDIX B  
DISCRETE-TIME AUTOCORRELATION FUNCTION  
OF INTEGRATED RANDOM PROCESS

This appendix derives the discrete-time autocorrelation function of the integrated random process

$$w_j = \int_{t_j}^{t_j+\Delta t} [\xi_a(\tau) + q_g(\tau)] d\tau \quad (\text{B.1})$$

where  $\xi_a(t)$  is a scalar first order Gauss-Markov process,  $q_g(t)$  is a zero-mean, Gaussian white noise process and  $\Delta t$  is the discrete-time sampling interval.

The autocorrelation function of  $q_g(t)$  is given by

$$E[q_g(t_1)q_g(t_2)] = Q_g \delta(t_1 - t_2) \quad (\text{B.2})$$

where  $Q_g$  is the power spectral density of the white noise process and  $\delta(t)$  is the Dirac delta function.

For the first order Gauss-Markov process, the autocorrelation function is well-known [Gel74]

$$E[\xi_a(t_1)\xi_a(t_2)] = \sigma_a^2 \exp\left(-\frac{|t_1 - t_2|}{\tau_a}\right) \quad (\text{B.3})$$

where  $\sigma_a^2$  and  $\tau_a$  are the variance and time constant of the process, respectively.

Now consider the values of  $w$  at time index  $j$  and time index  $k$ .

$$w_j = \int_{t_j}^{t_j+\Delta t} [\xi_a(u) + q_g(u)] du \quad (\text{B.4})$$

$$w_k = \int_{t_k}^{t_k+\Delta t} [\xi_a(u) + q_g(u)] du \quad (\text{B.5})$$

The discrete-time autocorrelation function of  $w$  is defined as  $E[w_j w_k]$ . From equations B.4 and B.5, this expectation can be written as

$$E[w_j w_k] = \int_{t_k}^{t_k + \Delta t} \int_{t_j}^{t_j + \Delta t} E\{[\xi_a(u) + q_g(u)][\xi_a(v) + q_g(v)]\} du dv \quad (\text{B.6})$$

Assuming that  $\xi_a(t)$  and  $q_g(t)$  are mutually uncorrelated, equation B.6 can be expanded into the form

$$E[w_j w_k] = \int_{t_k}^{t_k + \Delta t} \int_{t_j}^{t_j + \Delta t} E[\xi_a(u) \xi_a(v)] du dv + \int_{t_k}^{t_k + \Delta t} \int_{t_j}^{t_j + \Delta t} E[q_g(u) q_g(v)] du dv \quad (\text{B.7})$$

For simplicity, equation B.7 will be written in the form  $E[w_j w_k] = I_1 + I_2$ . In order to compute  $I_1$ , the integral must be evaluated separately for the cases  $k = j$  and  $k \neq j$  because of the absolute value in equation B.3.

For  $k = j$ , substituting equation B.3 into the first integral on the right hand side of equation B.7 results in

$$I_1 = \int_{t_k}^{t_k + \Delta t} \int_{t_k}^{t_k + \Delta t} \sigma_a^2 \exp\left(-\frac{|u-v|}{\tau_a}\right) du dv \quad (\text{B.8})$$

The absolute value can be eliminated by writing  $I_1$  as the sum of two integrals

$$I_1 = \int_{t_k}^{t_k + \Delta t} \int_{t_k}^v \sigma_a^2 \exp\left(-\frac{v-u}{\tau_a}\right) du dv + \int_{t_k}^{t_k + \Delta t} \int_v^{t_k + \Delta t} \sigma_a^2 \exp\left(-\frac{u-v}{\tau_a}\right) du dv \quad (\text{B.9})$$

It is straightforward to verify that the integrals evaluate to the expression

$$I_1 = 2(\sigma_a \tau_a)^2 \left[ \frac{\Delta t}{\tau_a} - 1 + \exp\left(-\frac{\Delta t}{\tau_a}\right) \right], \quad k = j \quad (\text{B.10})$$

Considering the case now where  $k > j$ ,  $I_1$  becomes

$$I_1 = \int_{t_k}^{t_k + \Delta t} \int_{t_j}^{t_j + \Delta t} \sigma_a^2 \exp\left(-\frac{v-u}{\tau_a}\right) du dv \quad (\text{B.11})$$

It is straightforward to show that the double integral in equation B.11 evaluates to

$$I_1 = (\sigma_a \tau_a)^2 \exp\left(-\frac{t_k - t_j}{\tau_a}\right) \left[ 1 - \exp\left(-\frac{\Delta t}{\tau_a}\right) \right] \left[ \exp\left(\frac{\Delta t}{\tau_a}\right) - 1 \right] , \quad k > j \quad (\text{B.12})$$

Because the autocorrelation function is symmetric about the origin, the case where  $k < j$  is easily handled by changing the argument  $t_k - t_j$  to  $t_j - t_k$  in equation B.12. In general,

$I_1$  can be written as

$$I_1 = \begin{cases} 2(\sigma_a \tau_a)^2 \left[ \frac{\Delta t}{\tau_a} - 1 + \exp\left(-\frac{\Delta t}{\tau_a}\right) \right] & , \quad j = k \\ (\sigma_a \tau_a)^2 \exp\left(-\frac{|j-k|\Delta t}{\tau_a}\right) \left[ 1 - \exp\left(-\frac{\Delta t}{\tau_a}\right) \right] \left[ \exp\left(\frac{\Delta t}{\tau_a}\right) - 1 \right] & , \quad j \neq k \end{cases} \quad (\text{B.13})$$

Substituting equation B.2 into the second integral on the right hand side of equation B.7 and using the fact that  $Q_g$  is a constant yields

$$I_2 = Q_g \int_{t_k}^{t_k + \Delta t} \int_{t_j}^{t_j + \Delta t} \delta(u - v) du dv \quad (\text{B.14})$$

Because the Dirac delta function is non-zero only when  $u = v$ , the double integral in equation B.14 is non-zero only when  $k = j$ . Therefore,  $I_2$  evaluates to

$$I_2 = \begin{cases} Q_g \Delta t & , \quad j = k \\ 0 & , \quad j \neq k \end{cases} \quad (\text{B.15})$$

Combining equations B.13 and B.15 results in the complete autocorrelation function

$$r_{jk} = \begin{cases} 2(\sigma_a \tau_a)^2 \left[ \frac{\Delta t}{\tau_a} - 1 + \exp\left(-\frac{\Delta t}{\tau_a}\right) \right] + Q_g \Delta t & , \quad j = k \\ (\sigma_a \tau_a)^2 \exp\left(-\frac{|j-k|\Delta t}{\tau_a}\right) \left[ 1 - \exp\left(-\frac{\Delta t}{\tau_a}\right) \right] \left[ \exp\left(\frac{\Delta t}{\tau_a}\right) - 1 \right] & , \quad j \neq k \end{cases} \quad (\text{B.16})$$

APPENDIX C  
CONTINUOUS-TIME DYNAMIC MODEL FOR  
TWO-DIMENSIONAL ESTIMATION PROBLEM

For the estimation problem shown in Figure 3.1, the state vector is given by

$$\mathbf{x} = [\rho \quad \theta \quad v_\rho \quad v_\theta \quad \psi]^T \quad (\text{C.1})$$

The continuous time dynamic models of  $\rho$  and  $\theta$  are given by

$$\dot{\rho} = v_\rho \quad (\text{C.2})$$

$$\dot{\theta} = \frac{v_\theta}{\rho} \quad (\text{C.3})$$

To derive the velocity dynamic model, two coordinate frames will be defined. An inertial coordinate frame (denoted by I) and a polar coordinate frame (denoted by P). These two frames are depicted in Figure C.1.

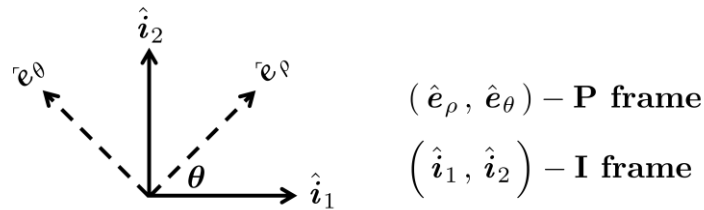


Figure C.1. Inertial and Polar Coordinate Frames

It is most convenient to express the vehicle velocity vector in the P-frame, i.e.,  $\mathbf{v} = v_\rho \hat{e}_\rho + v_\theta \hat{e}_\theta$ . Therefore, the dynamic model will be derived by differentiating in P. However, the derivative of  $\mathbf{v}$  in the P-frame must be related to the derivative of  $\mathbf{v}$  in the I-frame because accelerometers sense inertial acceleration of the vehicle. These derivatives are related via the relation [Wie10]

$$\frac{{}^P d\mathbf{v}}{dt} = \frac{{}^I d\mathbf{v}}{dt} - {}^I \boldsymbol{\omega}^P \times \mathbf{v} \quad (\text{C.4})$$

where  ${}^I \boldsymbol{\omega}^P$  is the angular velocity of the P-frame with respect to the I-frame, and is given by  ${}^I \boldsymbol{\omega}^P = \dot{\theta} \hat{e}_z$ . The vector  $\hat{e}_z$  is a unit vector orthogonal to  $\hat{e}_\rho$  and  $\hat{e}_\theta$ .

The derivative on the right hand side of equation C.4 is the inertial acceleration of the vehicle, and will be denoted by  $\mathbf{a}$ . Defining the left hand side of equation C.4 as  $\dot{\mathbf{v}}$ , this equation can be written as

$$\dot{\mathbf{v}} = \mathbf{a} - {}^I\boldsymbol{\omega}^P \times \mathbf{v} \quad (\text{C.5})$$

Accelerometers sense inertial acceleration in the vehicle body frame (B-frame). Therefore, equation C.5 is re-expressed in the form

$$\dot{\mathbf{v}} = {}^P\mathbf{R}^B \mathbf{a} - {}^I\boldsymbol{\omega}^P \times \mathbf{v} \quad (\text{C.6})$$

The matrix  ${}^P\mathbf{R}^B$  is a  $2 \times 2$  rotation matrix that rotates a vector from the B-frame (depicted in Figure 3.2) to the P-frame. Its determination is most easily accomplished by decomposing  ${}^P\mathbf{R}^B$  into the product of two rotation matrices

$${}^P\mathbf{R}^B = {}^P\mathbf{R}^I {}^I\mathbf{R}^B \quad (\text{C.7})$$

where  ${}^P\mathbf{R}^I \in \mathbb{R}^{2 \times 2}$  rotates a vector from the I-frame to the P-frame and  ${}^I\mathbf{R}^B \in \mathbb{R}^{2 \times 2}$  rotates a vector from the B-frame to the I-frame.

From Figure 3.2

$${}^I\mathbf{R}^B = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \quad (\text{C.8})$$

and from Figure C.1

$${}^P\mathbf{R}^I = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (\text{C.9})$$

Substituting equations C.8 and C.9 into equation C.7, it can be shown that

$${}^P\mathbf{R}^B = \begin{bmatrix} \cos(\theta - \psi) & \sin(\theta - \psi) \\ -\sin(\theta - \psi) & \cos(\theta - \psi) \end{bmatrix} \quad (\text{C.10})$$

The cross product in equation C.6 can be simplified to

$${}^I \boldsymbol{\omega}^P \times \mathbf{v} = \dot{\theta} \hat{\mathbf{e}}_z \times (v_\rho \hat{\mathbf{e}}_\rho + v_\theta \hat{\mathbf{e}}_\theta) \quad (\text{C.11})$$

Distributing the cross product and replacing  $\dot{\theta}$  with the right hand side of equation C.3 yields

$${}^I \boldsymbol{\omega}^P \times \mathbf{v} = -v_\theta \dot{\theta} \hat{\mathbf{e}}_\rho + v_\rho \dot{\theta} \hat{\mathbf{e}}_\theta \quad (\text{C.12})$$

$${}^I \boldsymbol{\omega}^P \times \mathbf{v} = -\frac{v_\theta^2}{\rho} \hat{\mathbf{e}}_\rho + \frac{v_\rho v_\theta}{\rho} \hat{\mathbf{e}}_\theta \quad (\text{C.13})$$

Gyroscopes sense the vehicle's inertial angular velocity, which for this example, is equal to the heading rate,  $\dot{\psi}$ . The dynamic model for vehicle heading is then given by

$$\dot{\psi} = \omega \quad (\text{C.14})$$

When taken together, equations C.2, C.3, C.6 and C.14 constitute a set of nonlinear differential equations that can be written in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{a}, \omega) \quad (\text{C.15})$$

where  $\mathbf{f}(\mathbf{x}, \mathbf{a}, \omega)$  is defined as

$$\mathbf{f}(\mathbf{x}, \mathbf{a}, \omega) = \begin{bmatrix} x_3 \\ x_4/x_1 \\ \begin{bmatrix} \cos(x_2 - x_5) & \sin(x_2 - x_5) \\ -\sin(x_2 - x_5) & \cos(x_2 - x_5) \end{bmatrix} \begin{bmatrix} a_{b_1} \\ a_{b_2} \end{bmatrix} - \begin{bmatrix} -x_4^2/x_1 \\ x_3 x_4/x_1 \end{bmatrix} \\ \omega \end{bmatrix} \quad (\text{C.16})$$

and the elements of the state vector  $\mathbf{x}$  are defined in equation C.1.

APPENDIX D  
DISCRETIZATION AND LINEARIZATION FOR  
TWO-DIMENSIONAL ESTIMATION PROBLEM

This appendix derives a procedure for the discretization and linearization of the following system of nonlinear differential equations

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{a}, \omega) \quad (\text{D.1})$$

where  $\mathbf{f}(\bullet) \in \mathbb{R}^5$  was derived in Appendix C and is given by

$$\mathbf{f}(\mathbf{x}, \mathbf{a}, \omega) = \begin{bmatrix} x_3 \\ x_4/x_1 \\ \begin{bmatrix} \cos(x_2 - x_5) & \sin(x_2 - x_5) \\ -\sin(x_2 - x_5) & \cos(x_2 - x_5) \end{bmatrix} \begin{bmatrix} a_{b_1} \\ a_{b_2} \end{bmatrix} - \begin{bmatrix} -x_4^2/x_1 \\ x_3 x_4/x_1 \end{bmatrix} \\ \omega \end{bmatrix} \quad (\text{D.2})$$

Equation D.1 can be converted to a discrete-time nonlinear system using a numerical integration scheme. For the purposes of this example, a fourth order Runge-Kutta algorithm is used, which is defined by the recursion [Kin02]

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \quad (\text{D.3})$$

where the  $5 \times 1$  vectors  $\mathbf{k}_1$  through  $\mathbf{k}_4$  are given by

$$\mathbf{k}_1 = h \mathbf{f}(\mathbf{x}_k, \mathbf{a}(t_k), \omega(t_k)) \quad (\text{D.4})$$

$$\mathbf{k}_2 = h \mathbf{f}\left(\mathbf{x}_k + \frac{1}{2}\mathbf{k}_1, \mathbf{a}(t_k + h/2), \omega(t_k + h/2)\right) \quad (\text{D.5})$$

$$\mathbf{k}_3 = h \mathbf{f}\left(\mathbf{x}_k + \frac{1}{2}\mathbf{k}_2, \mathbf{a}(t_k + h/2), \omega(t_k + h/2)\right) \quad (\text{D.6})$$

$$\mathbf{k}_4 = h \mathbf{f}(\mathbf{x}_k + \mathbf{k}_3, \mathbf{a}(t_k + h), \omega(t_k + h)) \quad (\text{D.7})$$

The parameter  $h$  is the step size of the fourth order Runge-Kutta integrator. In what follows, the shorthand notation  $\mathbf{a}_1$ ,  $\mathbf{a}_2$  and  $\mathbf{a}_3$  will be used to denote  $\mathbf{a}(t_k)$ ,

$\mathbf{a}(t_k + h/2)$  and  $\mathbf{a}(t_k + h)$ , respectively, and  $\omega_1$ ,  $\omega_2$  and  $\omega_3$  will be used to denote  $\omega(t_k)$ ,  $\omega(t_k + h/2)$  and  $\omega(t_k + h)$ , respectively.

Equation D.3 can be approximated by a linear system of difference equations using a first order Taylor series expansion. Making the following definitions

$$\mathbf{F}_k = \mathbf{I} + \frac{1}{6} \left( \frac{\partial \mathbf{k}_1}{\partial \mathbf{x}_k} + 2 \frac{\partial \mathbf{k}_2}{\partial \mathbf{x}_k} + 2 \frac{\partial \mathbf{k}_3}{\partial \mathbf{x}_k} + \frac{\partial \mathbf{k}_4}{\partial \mathbf{x}_k} \right) \quad (\text{D.8})$$

$$\mathbf{B}_{a_1} = \frac{1}{6} \left( \frac{\partial \mathbf{k}_1}{\partial \mathbf{a}_1} + 2 \frac{\partial \mathbf{k}_2}{\partial \mathbf{a}_1} + 2 \frac{\partial \mathbf{k}_3}{\partial \mathbf{a}_1} + \frac{\partial \mathbf{k}_4}{\partial \mathbf{a}_1} \right) \quad (\text{D.9})$$

$$\mathbf{B}_{a_2} = \frac{1}{6} \left( 2 \frac{\partial \mathbf{k}_2}{\partial \mathbf{a}_2} + 2 \frac{\partial \mathbf{k}_3}{\partial \mathbf{a}_2} + \frac{\partial \mathbf{k}_4}{\partial \mathbf{a}_2} \right) \quad (\text{D.10})$$

$$\mathbf{B}_{a_3} = \frac{1}{6} \frac{\partial \mathbf{k}_4}{\partial \mathbf{a}_3} \quad (\text{D.11})$$

$$\mathbf{B}_{\omega_1} = \frac{1}{6} \left( \frac{\partial \mathbf{k}_1}{\partial \omega_1} + 2 \frac{\partial \mathbf{k}_2}{\partial \omega_1} + 2 \frac{\partial \mathbf{k}_3}{\partial \omega_1} + \frac{\partial \mathbf{k}_4}{\partial \omega_1} \right) \quad (\text{D.12})$$

$$\mathbf{B}_{\omega_2} = \frac{1}{6} \left( 2 \frac{\partial \mathbf{k}_2}{\partial \omega_2} + 2 \frac{\partial \mathbf{k}_3}{\partial \omega_2} + \frac{\partial \mathbf{k}_4}{\partial \omega_2} \right) \quad (\text{D.13})$$

$$\mathbf{B}_{\omega_3} = \frac{1}{6} \frac{\partial \mathbf{k}_4}{\partial \omega_3} \quad (\text{D.14})$$

the Taylor series expansion can be written as

$$\begin{aligned} \mathbf{x}_{k+1} = & \mathbf{x}_k^* + \frac{1}{6} (\mathbf{k}_1^* + 2\mathbf{k}_2^* + 2\mathbf{k}_3^* + \mathbf{k}_4^*) + \mathbf{F}_k (\mathbf{x}_k - \mathbf{x}_k^*) + \\ & \mathbf{B}_{a_1} \delta \mathbf{a}_1 + \mathbf{B}_{a_2} \delta \mathbf{a}_2 + \mathbf{B}_{a_3} \delta \mathbf{a}_3 + \mathbf{B}_{\omega_1} \delta \omega_1 + \mathbf{B}_{\omega_2} \delta \omega_2 + \mathbf{B}_{\omega_3} \delta \omega_3 \end{aligned} \quad (\text{D.15})$$

The chain rule of differentiation must be applied repeatedly in order to evaluate the partial derivatives in equations D.8 through D.14. To illustrate, consider the term  $\partial \mathbf{k}_2 / \partial \mathbf{x}_k$ , which appears in equation D.8. In the definition of  $\mathbf{k}_2$  given in equation D.5, let  $\boldsymbol{\xi} = \mathbf{x}_k + \mathbf{k}_1/2$ . Then

$$\frac{\partial \mathbf{k}_2}{\partial \mathbf{x}_k} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}_k} \quad (\text{D.16})$$

$$\frac{\partial \mathbf{k}_2}{\partial \mathbf{x}_k} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}} \left( \mathbf{I} + \frac{1}{2} \frac{\partial \mathbf{k}_1}{\partial \mathbf{x}_k} \right) \quad (\text{D.17})$$

where  $\partial \mathbf{f} / \partial \boldsymbol{\xi}$  is determined by evaluating the gradient of the function  $f(\boldsymbol{\xi}, \mathbf{a}, \boldsymbol{\omega})$  defined in equation D.2 and  $\partial \mathbf{k}_1 / \partial \mathbf{x}_k$  is determined by taking the gradient of  $\mathbf{k}_1$  defined in equation D.4. The same procedure can be applied to compute the other partial derivatives in equations D.8 through D.14.

Defining  $\mathbf{u}_k \in \mathbb{R}^5$ ,  $\boldsymbol{\eta}_k \in \mathbb{R}^9$  and  $\mathbf{B}_k \in \mathbb{R}^{5 \times 9}$  as

$$\mathbf{u}_k = \mathbf{x}_k^* + \frac{1}{6}(\mathbf{k}_1^* + 2\mathbf{k}_2^* + 2\mathbf{k}_3^* + \mathbf{k}_4^*) - \mathbf{F}_k \mathbf{x}_k^* \quad (\text{D.18})$$

$$\boldsymbol{\eta}_k = [\delta \mathbf{a}_1^T \quad \delta \boldsymbol{\omega}_1^T \quad \delta \mathbf{a}_2^T \quad \delta \boldsymbol{\omega}_2^T \quad \delta \mathbf{a}_3^T \quad \delta \boldsymbol{\omega}_3^T]^T \quad (\text{D.19})$$

$$\mathbf{B}_k = [\mathbf{B}_{a_1} \quad \mathbf{B}_{\omega_1} \quad \mathbf{B}_{a_2} \quad \mathbf{B}_{\omega_2} \quad \mathbf{B}_{a_3} \quad \mathbf{B}_{\omega_3}] \quad (\text{D.20})$$

allows equation D.15 to be written as

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{u}_k + \mathbf{B}_k \boldsymbol{\eta}_k \quad (\text{D.21})$$

In order to compute  $\mathbf{F}_k$ ,  $\mathbf{u}_k$  and  $\mathbf{B}_k$ , the  $2 \times 1$  acceleration vector  $\mathbf{a}$  and scalar inertial angular velocity  $\boldsymbol{\omega}$  are required at three points in time. However, the

accelerometer and gyroscope output the integral of inertial acceleration and inertial angular velocity over a specified time interval,  $\delta t$ . That is

$$\Delta v = \int_{\delta t} a(\tau) d\tau \quad (\text{D.22})$$

$$\Delta \phi = \int_{\delta t} \omega(\tau) d\tau \quad (\text{D.23})$$

Therefore, it is necessary to relate  $a_1$  through  $a_3$  and  $\omega_1$  through  $\omega_3$  to  $\Delta v$  and  $\Delta \phi$ . This requires an acceleration and angular velocity profile over the integration interval. The Runge-Kutta step size is twice the accelerometer and gyroscope output rate (i.e.,  $h = 2\delta t$ ), which implies that there are two  $\Delta v$  and  $\Delta \phi$  measurements available over the interval  $[t_k, t_{k+1}]$ . Therefore, the highest order acceleration and angular velocity model that can be used is a linear model. For acceleration, this model is given by

$$a(t) = a_1 + \beta(t - t_k) \quad (\text{D.24})$$

where  $a_1$  and  $\beta$  are to be determined.

Substituting the right hand side of equation D.24 into equation D.22 results in

$$\Delta v_1 = \int_{t_k}^{t_k + \delta t} [a_1 + \beta(\tau - t_k)] d\tau = a_1 \delta t + \beta \frac{\delta t^2}{2} \quad (\text{D.25})$$

Similarly, the second  $\Delta v$  measurement can be expressed as

$$\Delta v_2 = \int_{t_k + \delta t}^{t_k + 2\delta t} [a_1 + \beta(\tau - t_k)] d\tau = a_1 \delta t + 3\beta \frac{\delta t^2}{2} \quad (\text{D.26})$$

Solving equations D.25 and D.26 for  $a_1$  and  $\beta$ , results in

$$\beta = \frac{\Delta v_2 - \Delta v_1}{\delta t^2} \quad (\text{D.27})$$

$$a_1 = \frac{3\Delta v_1 - \Delta v_2}{2\delta t} \quad (\text{D.28})$$

From equation D.24, notice that

$$a_2 = a_1 + \beta(t_k + \delta t - t_k) \quad (\text{D.29})$$

$$a_3 = a_1 + \beta(t_k + 2\delta t - t_k) \quad (\text{D.30})$$

Substituting the expressions for  $a_1$  and  $\beta$  from equations D.27 and D.28 into equations D.29 and D.30 yields

$$a_2 = \frac{\Delta v_1 + \Delta v_2}{2\delta t} \quad (\text{D.31})$$

$$a_3 = \frac{3\Delta v_2 - \Delta v_1}{2\delta t} \quad (\text{D.32})$$

Analogous expressions can be derived for  $\omega_1$ ,  $\omega_2$  and  $\omega_3$ .

$$\omega_1 = \frac{3\Delta\phi_1 - \Delta\phi_2}{2\delta t} \quad (\text{D.33})$$

$$\omega_2 = \frac{\Delta\phi_1 + \Delta\phi_2}{2\delta t} \quad (\text{D.34})$$

$$\omega_3 = \frac{3\Delta\phi_2 - \Delta\phi_1}{2\delta t} \quad (\text{D.35})$$

Equations D.28 and D.31 through D.35 are used to populate  $\mathbf{F}_k$ ,  $\mathbf{u}_k$  and  $\mathbf{B}_k$  in equation D.21. The last step is to use equations D.28 and D.31 through D.35 to convert  $\boldsymbol{\eta}_k$  from errors in  $\mathbf{a}$  and  $\boldsymbol{\omega}$  to errors in  $\Delta\mathbf{v}$  and  $\Delta\boldsymbol{\phi}$ . Let  $\mathbf{w}_k \in \mathbb{R}^6$  be defined as

$$\mathbf{w}_k = [\delta\Delta\mathbf{v}_1^T \quad \delta\Delta\boldsymbol{\phi}_1 \quad \delta\Delta\mathbf{v}_2^T \quad \delta\Delta\boldsymbol{\phi}_2]^T \quad (\text{D.36})$$

Then  $\boldsymbol{\eta}_k$  and  $\boldsymbol{w}_k$  are related by the linear transformation  $\boldsymbol{\eta}_k = \mathbf{M}\boldsymbol{w}_k$ , where  $\mathbf{M} \in \mathbb{R}^{9 \times 6}$  is given by

$$\mathbf{M} = \begin{bmatrix} 3\mathbf{I} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & 3 & \mathbf{0} & -1 \\ \mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} & 1 \\ -\mathbf{I} & \mathbf{0} & 3\mathbf{I} & \mathbf{0} \\ \mathbf{0} & -1 & \mathbf{0} & 3 \end{bmatrix} \quad (\text{D.37})$$

Replacing  $\boldsymbol{\eta}_k$  in equation D.21 by  $\mathbf{M}\boldsymbol{w}_k$  and defining  $\mathbf{G}_k \in \mathbb{R}^{5 \times 6}$  as  $\mathbf{G}_k = \mathbf{B}_k \mathbf{M}$ , the linearized dynamic model is finally given by

$$\boldsymbol{x}_{k+1} = \mathbf{F}_k \boldsymbol{x}_k + \boldsymbol{u}_k + \mathbf{G}_k \boldsymbol{w}_k \quad (\text{D.38})$$

APPENDIX E  
CONDITIONS FOR PRIMAL-DUAL OPTIMALITY

This appendix derives a set of equations that must be satisfied by the optimal solution to the primal and dual SDPs stated in equations 4.16 and 4.17. As long as the primal and dual problems have non-empty interiors (i.e., there exists an  $\mathbf{X} \succ \mathbf{0}$  and  $\mathbf{S} \succ \mathbf{0}$  that satisfy the equality constraints in equations 4.16 and 4.17), optimal solutions  $\mathbf{X}^*$ ,  $\mathbf{S}^*$  and  $\boldsymbol{\lambda}^*$  exist for the primal and dual SDPs such that  $\text{Tr}(\mathbf{C}\mathbf{X}^*) = \mathbf{d}^T \boldsymbol{\lambda}^*$  [Van96]. Replacing  $\mathbf{C}$  with the left hand side of the equality constraint in equation 4.17 results in

$$\text{Tr} \left\{ \left( \sum_{i=1}^{n(n+1)} \lambda_i^* \mathbf{A}_i + \mathbf{S}^* \right) \mathbf{X}^* \right\} = \mathbf{d}^T \boldsymbol{\lambda}^* \quad (\text{E.1})$$

$$\sum_{i=1}^{n(n+1)} \lambda_i^* \text{Tr}(\mathbf{A}_i \mathbf{X}^*) + \text{Tr}(\mathbf{S}^* \mathbf{X}^*) = \mathbf{d}^T \boldsymbol{\lambda}^* \quad (\text{E.2})$$

Because  $\mathbf{X}^*$  is primal feasible,  $\text{Tr}(\mathbf{A}_i \mathbf{X}^*) = d_i$ . Therefore, equation E.2 reduces to

$$\sum_{i=1}^{n(n+1)} \lambda_i^* d_i + \text{Tr}(\mathbf{S}^* \mathbf{X}^*) = \mathbf{d}^T \boldsymbol{\lambda}^* \Rightarrow \text{Tr}(\mathbf{S}^* \mathbf{X}^*) = 0 \quad (\text{E.3})$$

Note that for two symmetric, positive semi-definite matrices  $\mathbf{A}$  and  $\mathbf{B}$ ,  $\text{Tr}(\mathbf{A}\mathbf{B}) = 0 \rightarrow \mathbf{A}\mathbf{B} = \mathbf{0}$ . Therefore, at a primal-dual optimal solution, equation E.3 implies that  $\mathbf{S}^* \mathbf{X}^* = \mathbf{0}$ . Re-introducing the constraints from equations 4.16 and 4.17, primal-dual optimal solutions must satisfy the following equations

$$\text{Tr}(\mathbf{A}_i \mathbf{X}^*) = d_i, \quad i = 1, \dots, n(n+1) \quad (\text{E.4})$$

$$\sum_{i=1}^{n(n+1)} \lambda_i \mathbf{A}_i + \mathbf{S} = \mathbf{C} \quad (\text{E.5})$$

$$\mathbf{X}\mathbf{S} = \mathbf{0} \quad (\text{E.6})$$

$$\mathbf{X} \succeq \mathbf{0}, \quad \mathbf{S} \succeq \mathbf{0} \quad (\text{E.7})$$

APPENDIX F  
INFEASIBLE INTERIOR POINT ALGORITHM  
FOR SEMI-DEFINITE OPTIMIZATION

Interior point algorithms work by restricting the iterates to a certain neighborhood of the *central path*, defined by the set of equations [Wri97]

$$\text{Tr}(\mathbf{A}_i \mathbf{X}) = d_i, \quad i = 1, \dots, n(n+1) \quad (\text{F.1})$$

$$\sum_{i=1}^{n(n+1)} \lambda_i \mathbf{A}_i + \mathbf{S} = \mathbf{C} \quad (\text{F.2})$$

$$\mathbf{X} \mathbf{S} = \tau \mathbf{I}, \quad \tau > 0 \quad (\text{F.3})$$

$$\mathbf{X} \succeq 0, \quad \mathbf{S} \succeq 0 \quad (\text{F.4})$$

As  $\tau \rightarrow 0$ , the solution to equations F.1 through F.4 approaches the primal-dual optimal solution. Most interior point algorithms set  $\tau$  equal to  $\sigma \bar{\mu}$ , where  $\sigma \in [0, 1]$  is a centering parameter and  $\bar{\mu}$  is the *normalized duality gap*, i.e.,  $\bar{\mu} = \text{Tr}(\mathbf{X} \mathbf{S}) / [n(n+2)]$  [Lue08].

In order to solve equations F.1 through F.4, they must be converted to a set of linear equations. For guess solutions  $\bar{\mathbf{X}} \succeq 0$ ,  $\bar{\mathbf{S}} \succeq 0$ , and  $\bar{\boldsymbol{\lambda}}$ , the linearization is accomplished by making the substitutions  $\mathbf{X} = \bar{\mathbf{X}} + \Delta \mathbf{X}$ ,  $\mathbf{S} = \bar{\mathbf{S}} + \Delta \mathbf{S}$ , and  $\boldsymbol{\lambda} = \bar{\boldsymbol{\lambda}} + \Delta \boldsymbol{\lambda}$ , resulting in

$$\text{Tr}(\mathbf{A}_i \Delta \mathbf{X}) = -\text{Tr}(\mathbf{A}_i \bar{\mathbf{X}}) + d_i, \quad i = 1, \dots, n(n+1) \quad (\text{F.5})$$

$$\sum_{i=1}^{n(n+1)} \Delta \lambda_i \mathbf{A}_i + \Delta \mathbf{S} = -\sum_{i=1}^{n(n+1)} \bar{\lambda}_i \mathbf{A}_i - \bar{\mathbf{S}} + \mathbf{C} \quad (\text{F.6})$$

$$\Delta \mathbf{X} \bar{\mathbf{S}} + \bar{\mathbf{X}} \Delta \mathbf{S} = -\bar{\mathbf{X}} \bar{\mathbf{S}} + \sigma \bar{\mu} \mathbf{I} \quad (\text{F.7})$$

where the 2<sup>nd</sup> order term  $\Delta \mathbf{X} \Delta \mathbf{S}$  has been dropped from equation F.7.

Defining the scalar  $r_{b,i}$  and matrix  $\mathbf{R}_c \in \mathbb{R}^{n(n+2) \times n(n+2)}$  as

$$r_{b,i} = \text{Tr}(\mathbf{A}_i \bar{\mathbf{X}}) - d_i \quad (\text{F.8})$$

$$\mathbf{R}_c = \sum_{i=1}^{n(n+1)} \bar{\lambda}_i \mathbf{A}_i + \bar{\mathbf{S}} - \mathbf{C} \quad (\text{F.9})$$

allows equations F.5 through F.7 to be written as

$$\text{Tr}(\mathbf{A}_i \Delta \mathbf{X}) = -r_{b,i} \quad , \quad i = 1, \dots, n(n+1) \quad (\text{F.10})$$

$$\sum_{i=1}^{n(n+1)} \Delta \lambda_i \mathbf{A}_i + \Delta \mathbf{S} = -\mathbf{R}_c \quad (\text{F.11})$$

$$\Delta \mathbf{X} \bar{\mathbf{S}} + \bar{\mathbf{X}} \Delta \mathbf{S} = -\bar{\mathbf{X}} \bar{\mathbf{S}} + \sigma \bar{\mu} \mathbf{I} \quad (\text{F.12})$$

Solving equations F.10 through F.12 for  $\Delta \mathbf{X}$ ,  $\Delta \mathbf{S}$  and  $\Delta \lambda$  will, in general, produce matrices  $\Delta \mathbf{X}$  and  $\Delta \mathbf{S}$  that are not symmetric. In response, the symmetrization operator, defined in equation F.13 for an arbitrary matrix  $\mathbf{M}$ , is introduced [Kou07].

$$\mathcal{H}_p(\mathbf{M}) = \frac{1}{2} [\mathbf{PMP}^{-1} + (\mathbf{PMP}^{-1})^T] \quad , \quad \mathbf{P} = [\bar{\mathbf{X}}^{-1/2} (\bar{\mathbf{X}}^{1/2} \bar{\mathbf{S}} \bar{\mathbf{X}}^{1/2})^{1/2} \bar{\mathbf{X}}^{-1/2}]^{1/2} \quad (\text{F.13})$$

Applying  $\mathcal{H}_p$  to both sides of equation F.12 results in

$$\mathcal{H}_p(\Delta \mathbf{X} \bar{\mathbf{S}} + \bar{\mathbf{X}} \Delta \mathbf{S}) = -\mathcal{H}_p(\bar{\mathbf{X}} \bar{\mathbf{S}}) + \sigma \bar{\mu} \mathbf{I} \quad (\text{F.14})$$

Equations F.10, F.11 and F.14 constitute a set of linear matrix equations. In order to obtain the search directions  $\Delta \mathbf{X}$ ,  $\Delta \mathbf{S}$  and  $\Delta \lambda$ , these equations must first be written in matrix-vector form. To facilitate the conversion, the following relations are used

$$\text{Tr}(\mathbf{A}\mathbf{B}) = \text{vec}(\mathbf{A}^T)^T \text{vec}(\mathbf{B}) \quad (\text{F.15})$$

$$\text{vec}(\mathbf{A}\mathbf{X}\mathbf{B}) = (\mathbf{B}^T \otimes \mathbf{A}) \text{vec}(\mathbf{X}) \quad (\text{F.16})$$

where  $\text{vec}(\mathbf{X})$  is a column vector obtained by stacking each column of  $\mathbf{X}$  underneath the immediately preceding column and  $\otimes$  indicates the Kronecker product of two matrices [Bel97].

The resulting linear system of equations has the form  $\mathbf{M}\mathbf{x} = \mathbf{b}$ . Defining  $n_m$  as  $n_m = 2[n(n+2)]^2 + n(n+1)$ , then  $\mathbf{M} \in \mathbb{R}^{n_m \times n_m}$ ,  $\mathbf{x} \in \mathbb{R}^{n_m}$  and  $\mathbf{b} \in \mathbb{R}^{n_m}$  with the following structure

$$\mathbf{M} = \begin{bmatrix} \text{vec}(\mathbf{A}_1^T)^T & 0 & \cdots & 0 & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \text{vec}(\mathbf{A}_{n(n+1)}^T)^T & 0 & \cdots & 0 & \mathbf{0} \\ \mathbf{0} & \text{vec}(\mathbf{A}_1) & \cdots & \text{vec}(\mathbf{A}_{n(n+1)}) & \mathbf{I} \\ \mathbf{F}_{\Delta\mathbf{X}} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{F}_{\Delta\mathbf{S}} \end{bmatrix} \quad (\text{F.17})$$

$$\mathbf{x} = [\text{vec}(\Delta\mathbf{X})^T \quad \Delta\lambda_1 \quad \cdots \quad \Delta\lambda_{n(n+1)} \quad \text{vec}(\Delta\mathbf{S})^T]^T \quad (\text{F.18})$$

$$\mathbf{b} = [\mathbf{r}_b^T \quad \mathbf{R}_{c,1}^T \quad \cdots \quad \mathbf{R}_{c,n(n+2)}^T \quad \text{vec}[-\mathcal{H}_p(\bar{\mathbf{X}}\bar{\mathbf{S}}) + \sigma\bar{\mu}\mathbf{I}]^T]^T \quad (\text{F.19})$$

The vector  $\mathbf{r}_b \in \mathbb{R}^{n(n+1)}$  is a column vector containing the  $r_{b,i}$ 's,  $\mathbf{R}_{c,i} \in \mathbb{R}^{n(n+2)}$  is the  $i^{\text{th}}$  column of  $\mathbf{R}_c$  and  $\mathbf{F}_{\Delta\mathbf{X}}$ ,  $\mathbf{F}_{\Delta\mathbf{S}}$  are  $[n(n+2)]^2 \times [n(n+2)]^2$  matrices defined as

$$\mathbf{F}_{\Delta\mathbf{X}} = \frac{1}{2}[(\mathbf{P}^{-T}\bar{\mathbf{S}}) \otimes \mathbf{P} + \mathbf{P} \otimes (\mathbf{P}^{-T}\bar{\mathbf{S}})] \quad (\text{F.20})$$

$$\mathbf{F}_{\Delta\mathbf{S}} = \frac{1}{2}[\mathbf{P}^{-T} \otimes (\mathbf{P}\bar{\mathbf{X}}) + (\mathbf{P}\bar{\mathbf{X}}) \otimes \mathbf{P}^{-T}] \quad (\text{F.21})$$

It is not difficult to imagine that even for relatively small  $n$ , the size of  $\mathbf{M}$  is exceedingly large. In order to solve the linear system of equations efficiently, sparseness of the coefficient matrix must be taken advantage of and parallel computation techniques must be employed. A detailed discussion of this topic can be found in Chapter 11 of [Wri97].

After solving equations F.10, F.11 and F.14 for the search directions  $\Delta\mathbf{X}$ ,  $\Delta\mathbf{S}$  and  $\Delta\boldsymbol{\lambda}$ , the guess solutions  $\bar{\mathbf{X}}$ ,  $\bar{\mathbf{S}}$  and  $\bar{\boldsymbol{\lambda}}$  are updated according to

$$\mathbf{X} = \bar{\mathbf{X}} + q\Delta\mathbf{X} \quad (\text{F.22})$$

$$\mathbf{S} = \bar{\mathbf{S}} + q\Delta\mathbf{S} \quad (\text{F.23})$$

$$\boldsymbol{\lambda} = \bar{\boldsymbol{\lambda}} + q\Delta\boldsymbol{\lambda} \quad (\text{F.24})$$

where the scalar  $q$  is the largest value in the interval  $[0, 1]$  that keeps  $\mathbf{X}$ ,  $\mathbf{S}$  and  $\boldsymbol{\lambda}$  in a specified neighborhood of the central path.

In this dissertation, an infeasible interior point algorithm is used that operates in a neighborhood of the central path that admits infeasible iterates. The advantage of this type of algorithm is that it does not require an initialization phase. Instead, it merely requires initial values of  $\mathbf{X}$  and  $\mathbf{S}$  (denoted by  $\bar{\mathbf{X}}_0$  and  $\bar{\mathbf{S}}_0$ , respectively) that are symmetric positive definite. The degree of infeasibility in  $\mathbf{X}$ ,  $\mathbf{S}$  and  $\boldsymbol{\lambda}$  is captured by computing the vector norm

$$\rho = \|[r_b^T \quad \text{vec}(\mathbf{R}_c)^T]\| \quad (\text{F.25})$$

Notice that if  $\bar{\mathbf{X}}$ ,  $\bar{\mathbf{S}}$  and  $\bar{\boldsymbol{\lambda}}$  are feasible then  $r_b = \mathbf{0}$  and  $\mathbf{R}_c = \mathbf{0}$  (i.e.,  $\rho = 0$ ). Also notice from equation E.3 in Appendix E that the value of the normalized duality gap is equal to zero at  $\mathbf{X}^*$  and  $\mathbf{S}^*$ . Therefore, at each step of the algorithm, progress must be

made toward reducing  $\mu$  and  $\rho$  to zero. This is accomplished by choosing the step length  $q \in [0, 1]$  to be the largest value such that

$$\mu \leq (1 - 0.01q) \bar{\mu} \quad (\text{F.26})$$

$$\frac{\rho}{\bar{\rho}_0} \leq \beta \frac{\mu}{\bar{\mu}_0} \quad (\text{F.27})$$

where  $\bar{\rho}_0$  and  $\bar{\mu}_0$  are computed at the initial point  $\bar{\mathbf{X}}_0$ ,  $\bar{\mathbf{S}}_0$  and  $\bar{\boldsymbol{\lambda}}_0$  and  $\beta \geq 1$  is a given parameter that controls how much priority is placed on improving feasibility at each iteration. In the most aggressive case where  $\beta = 1$ , the relative decrease in the degree of infeasibility must be at least as large as the relative decrease in the normalized duality gap.

Equation F.27 is one characteristic of the infeasible neighborhood used in this work, which is defined as [Wri97]

$$\begin{aligned} \mathcal{N}_{-\infty}(\gamma, \beta) = \{ \mathbf{X}, \mathbf{S}, \boldsymbol{\lambda} \mid \rho \leq (\bar{\rho}_0 / \bar{\mu}_0) \beta \mu, \\ \mathbf{X} \succ 0, \mathbf{S} \succ 0, \text{eig}_i(\mathbf{XS}) \geq \gamma \mu, i = 1, \dots, n(n+2) \} \end{aligned} \quad (\text{F.28})$$

where  $\gamma \in (0, 1)$  is a given value and  $\text{eig}_i(\mathbf{XS})$  indicates the  $i^{\text{th}}$  eigenvalue of  $\mathbf{XS}$ . In this dissertation, the configurable parameters are set equal to:  $\sigma = 0.25$ ,  $\beta = 2$  and  $\gamma = 0.01$ .

## APPENDIX G

## PROOF THAT GLOBAL MINIMUM RESIDES AT AN ENDPOINT

Consider the minimization problem

$$\min_{\sigma_j^2} \left[ \Phi\left(\frac{\ell - \mu}{\sqrt{\sigma^2}}\right) + \Phi\left(\frac{\ell + \mu}{\sqrt{\sigma^2}}\right) - 1 \right] \quad (\text{G.1})$$

$$\text{s.t. } l \leq \sigma^2 \leq u$$

where

$$\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right) du \quad (\text{G.2})$$

Let the cost function be denoted by  $f(\sigma^2)$ . Differentiating  $f(\sigma^2)$  with respect to  $\sigma^2$  results in

$$\begin{aligned} \frac{df}{d(\sigma^2)} = & \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma^2}(\ell^2 - 2\mu\ell + \mu^2)\right] \left[\frac{-\ell + \mu}{\sqrt{(\sigma^2)^3}}\right] + \\ & \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma^2}(\ell^2 + 2\mu\ell + \mu^2)\right] \left[\frac{-\ell - \mu}{\sqrt{(\sigma^2)^3}}\right] \end{aligned} \quad (\text{G.3})$$

Factoring equation G.3 yields

$$\begin{aligned} \frac{df}{d(\sigma^2)} = & \frac{1}{\sqrt{2\pi}(\sigma^2)^3} \exp\left[-\frac{1}{2\sigma^2}(\ell^2 + \mu^2)\right] \\ & [(-\ell + \mu)e^{\mu\ell/\sigma^2} + (-\ell - \mu)e^{-\mu\ell/\sigma^2}] \end{aligned} \quad (\text{G.4})$$

Making the definition  $\xi = \mu\ell/\sigma^2$ , equation G.4 can be rearranged into the form

$$\frac{df}{d(\sigma^2)} = \frac{1}{\sqrt{2\pi}(\sigma^2)^3} \exp\left[-\frac{1}{2\sigma^2}(\ell^2 + \mu^2)\right] [-\ell(e^\xi + e^{-\xi}) + \mu(e^\xi - e^{-\xi})] \quad (\text{G.5})$$

Recognizing the last factor on the right hand side of equation G.5 as  $-2\ell \cosh \xi + 2\mu \sinh \xi$ , the first derivative becomes

$$\frac{df}{d(\sigma^2)} = \sqrt{\frac{2}{\pi(\sigma^2)^3}} \exp\left[-\frac{1}{2\sigma^2}(\ell^2 + \mu^2)\right] (-\ell \cosh \xi + \mu \sinh \xi) \quad (\text{G.6})$$

The roots of equation G.6 are the critical points of  $f(\sigma^2)$ . Notice that  $\exp(-x^2)$  is non-zero for any finite value of  $x$ . This implies that the critical points must satisfy the relation

$$-\ell \cosh \xi + \mu \sinh \xi = 0 \quad (\text{G.7})$$

Given that  $\cosh \xi$  is non-zero for all  $\xi$ , dividing both sides of equation G.7 by  $\cosh \xi$  yields

$$-\ell + \mu \tanh \xi = 0 \quad (\text{G.8})$$

The fact that  $\tanh \xi$  is a monotonically increasing function implies that there is only one solution to equation G.8, which is given by

$$\xi_c = \tanh^{-1}(\ell/\mu) \quad (\text{G.9})$$

Re-introducing the definition  $\xi = \mu\ell/\sigma^2$ , equation G.9 implies that

$$\sigma_c^2 = \frac{\mu\ell}{\tanh^{-1}(\ell/\mu)} \quad (\text{G.10})$$

Whether or not the critical point in equation G.10 is a local minimum or a local maximum can be ascertained using the second derivative test. However, before differentiating equation G.6, the following definitions will be made

$$g(\sigma^2) = \sqrt{\frac{2}{\pi(\sigma^2)^3}} \exp\left[-\frac{1}{2\sigma^2}(\ell^2 + \mu^2)\right] \quad (\text{G.11})$$

$$h(\xi) = (-\ell \cosh \xi + \mu \sinh \xi) \quad (\text{G.12})$$

which allows equation G.6 to be written more compactly as

$$\frac{df}{d(\sigma^2)} = g(\sigma^2)h(\xi) \quad (\text{G.13})$$

The second derivative is then given by

$$\frac{d^2f}{d(\sigma^2)^2} = g(\sigma^2) \frac{dh}{d\xi} \frac{d\xi}{d(\sigma^2)} + \frac{dg}{d(\sigma^2)} h(\xi) \quad (\text{G.14})$$

At the critical point  $\xi = \xi_c$  specified in equation G.9,  $h(\xi_c) = 0$ . Therefore, it is only necessary to evaluate the first term on the right hand side of equation G.14.

$$\begin{aligned} \left. \frac{d^2f}{d(\sigma^2)^2} \right|_{\sigma_c^2, \xi_c} &= \sqrt{\frac{2}{\pi(\sigma_c^2)^3}} \exp\left[-\frac{1}{2\sigma_c^2}(\ell^2 + \mu^2)\right] \\ &\quad (-\ell \sinh \xi_c + \mu \cosh \xi_c) \left( \frac{-\ell \mu}{(\sigma_c^2)^2} \right) \end{aligned} \quad (\text{G.15})$$

Equation G.15 simplifies to

$$\left. \frac{d^2f}{d(\sigma^2)^2} \right|_{\sigma_c^2, \xi_c} = -\ell \mu \sqrt{\frac{2}{\pi(\sigma_c^2)^7}} \exp\left[-\frac{1}{2\sigma_c^2}(\ell^2 + \mu^2)\right] (-\ell \sinh \xi_c + \mu \cosh \xi_c) \quad (\text{G.16})$$

Making the substitution  $\xi_c = \tanh^{-1}(\ell/\mu)$  from equation G.9 and using the relations

$$\sinh(\tanh^{-1} x) = \frac{x}{\sqrt{1-x^2}} \quad (\text{G.17})$$

$$\cosh(\tanh^{-1} x) = \frac{1}{\sqrt{1-x^2}} \quad (\text{G.18})$$

equation G.16 becomes

$$\left. \frac{d^2f}{d(\sigma^2)^2} \right|_{\sigma_c^2} = -\ell \mu \sqrt{\frac{2}{\pi(\sigma_c^2)^7}} \exp\left[-\frac{1}{2\sigma_c^2}(\ell^2 + \mu^2)\right] \left[ \frac{-\ell^2 + \mu^2}{\mu \sqrt{1-(\ell/\mu)^2}} \right] \quad (\text{G.19})$$

The last factor on the right hand side of equation G.19 can be simplified further, depending on whether  $\mu$  is positive or negative.

$$\frac{-\ell^2 + \mu^2}{\mu\sqrt{1-(\ell/\mu)^2}} = \begin{cases} \sqrt{\mu^2 - \ell^2} & , \mu \geq 0 \\ -\sqrt{\mu^2 - \ell^2} & , \mu < 0 \end{cases} \quad (\text{G.20})$$

Upon substituting equation G.20 into equation G.19, the second derivative at the critical point becomes

$$\left. \frac{d^2 f}{d(\sigma^2)^2} \right|_{\sigma_c^2} = -\ell |\mu| \sqrt{\frac{2(\mu^2 - \ell^2)}{\pi(\sigma_c^2)^7}} \exp\left[-\frac{1}{2\sigma_c^2}(\ell^2 + \mu^2)\right] \quad (\text{G.21})$$

Because  $\ell$  is always positive, the right hand side of equation G.21 is negative at  $\sigma_c^2$  (i.e.,  $\sigma_c^2$  is a local maximum). Therefore, the global minimum to the optimization problem in equation G.1 must occur at either  $l$  or  $u$ .

## APPENDIX H

## DETERMINATION OF POLYTOPIC FEASIBLE REGION

The vector of conditional variances obeys the linear system of equations

$$\boldsymbol{\sigma}^2 = \mathbf{c} + \mathbf{D}\mathbf{r} \quad (\text{H.1})$$

where  $\boldsymbol{\sigma}^2 \in \mathbb{R}^{m+1}$ ,  $\mathbf{c} \in \mathbb{R}^{m+1}$  and  $\mathbf{D} \in \mathbb{R}^{(m+1) \times n_r}$ .

Equation H.1 can also be written in the form

$$\boldsymbol{\sigma}^2 = \mathbf{c} + \mathbf{d}_1 r_1 + \mathbf{d}_2 r_2 + \cdots + \mathbf{d}_{n_r} r_{n_r} \quad (\text{H.2})$$

where  $\mathbf{d}_i \in \mathbb{R}^{m+1}$  is the  $i^{\text{th}}$  column of  $\mathbf{D}$  and  $r_i$  is the  $i^{\text{th}}$  component of  $\mathbf{r}$ .

Now suppose that  $m+1$  of the  $\mathbf{d}_i$  are chosen to form a matrix  $\mathbf{A} \in \mathbb{R}^{(m+1) \times (m+1)}$

and the remaining  $n_r - m - 1$   $\mathbf{d}_i$  are used to form a matrix  $\mathbf{B} \in \mathbb{R}^{(m+1) \times (n_r - m - 1)}$ . Then

equation H.2 becomes

$$\boldsymbol{\sigma}^2 = \mathbf{c} + [\mathbf{A} \quad \mathbf{B}] \begin{bmatrix} \mathbf{r}_a \\ \mathbf{r}_b \end{bmatrix} \quad (\text{H.3})$$

with  $\mathbf{r}_a \in \mathbb{R}^{m+1}$  and  $\mathbf{r}_b \in \mathbb{R}^{n_r - m - 1}$  formed from components of  $\mathbf{r}$  that correspond to the columns of  $\mathbf{A}$  and  $\mathbf{B}$ , respectively.

Notice that  $\mathbf{A}$  is a square matrix. Pre-multiplying both sides of equation H.3 by  $\mathbf{A}^{-1}$  results in

$$\mathbf{A}^{-1} \boldsymbol{\sigma}^2 = \mathbf{A}^{-1} \mathbf{c} + [\mathbf{I} \quad \mathbf{A}^{-1} \mathbf{B}] \begin{bmatrix} \mathbf{r}_a \\ \mathbf{r}_b \end{bmatrix} \quad (\text{H.4})$$

Defining  $\mathbf{c}_A \in \mathbb{R}^{m+1}$ ,  $\mathbf{L}_A \in \mathbb{R}^{(m+1) \times n_r}$  and  $\mathbf{r}_A \in \mathbb{R}^{n_r}$  as  $\mathbf{c}_A = \mathbf{A}^{-1} \mathbf{c}$ ,  $\mathbf{L}_A = [\mathbf{I} \quad \mathbf{A}^{-1} \mathbf{B}]$ ,

and  $\mathbf{r}_A = [\mathbf{r}_a^T \quad \mathbf{r}_b^T]^T$  allows equation H.4 to be written more concisely as

$$\mathbf{A}^{-1} \boldsymbol{\sigma}^2 - \mathbf{c}_A = \mathbf{L}_A \mathbf{r}_A \quad (\text{H.5})$$

The components of  $\mathbf{A}^{-1}\boldsymbol{\sigma}^2 - \mathbf{c}_A$  are known affine combinations of the  $\sigma_j^2$ , whereas  $\mathbf{L}_A \mathbf{r}_A$  has unknown but bounded elements. For a given component of  $\mathbf{L}_A \mathbf{r}_A$ , upper and lower bounds are determined by setting each component of  $\mathbf{r}_A$  equal to its upper or lower bounding value, depending on the sign of the corresponding element in  $\mathbf{L}_A$ . This is analogous to the procedure described in Section 2.6 to upper and lower bound  $\sigma_j^2$ . Therefore, equation H.5 describes a set of  $2(m+1)$  linear inequality constraints of the form

$$\mathbf{g}_k^T \boldsymbol{\sigma}^2 \leq \beta_k \quad (\text{H.6})$$

where  $\mathbf{g}_k \in \mathbb{R}^{m+1}$  and  $\beta_k$  is a scalar.

Selecting a different set of  $\mathbf{d}_i$  for  $\mathbf{A}$  and assigning the remaining  $\mathbf{d}_i$  to  $\mathbf{B}$  will yield a new set of inequalities. This process can be performed for all possible  $\mathbf{A}$  matrices. However, it is only necessary to consider sets of  $\mathbf{d}_i$  that include the first column of  $\mathbf{D}$ ; those that do not include the first column result in repeated inequality constraints. Hence,  $C(n_r - 1, m)$  combinations of the  $\mathbf{d}_i$  need to be considered, where  $C(n, r)$  denotes how many ways  $n$  objects can be taken  $r$  at a time. That is

$$C(n, r) = \frac{n!}{r!(n-r)!} \quad (\text{H.7})$$

For the specific case where  $m=3$  and  $n_r=7$ , equation H.7 indicates that there are 20 relevant sets of  $\mathbf{d}_i \in \mathbb{R}^4$ . Table H.1 lists these sets together with the number of new inequalities contributed by equation H.5 for each set.

Table H.1. Number of Constraints Introduced from each Column Combination

Group One					Group Two				
Columns of $\mathbf{D}$				Number of New Inequalities	Columns of $\mathbf{D}$				Number of New Inequalities
1	2	3	4	8	1	3	4	5	2
1	2	3	5	6	1	3	4	6	2
1	2	3	6	6	1	3	4	7	2
1	2	3	7	6	1	3	5	6	2
1	2	4	5	4	1	3	5	7	2
1	2	4	6	4	1	3	6	7	2
1	2	4	7	4	1	4	5	6	2
1	2	5	6	4	1	4	5	7	2
1	2	5	7	4	1	4	6	7	2
1	2	6	7	4	1	5	6	7	2

There are a total of 70 inequality constraints when  $m=3$  and  $n_r=7$ .

Mathematically, the number of inequalities  $n_e$  can be written as

$$n_e = 2 + C(n_r - m, 1) + C(n_r - m + 1, 2) + C(n_r - m + 2, 3) \quad (\text{H.8})$$

Equation H.8 can be generalized to arbitrary  $m$  and  $n_r$ , resulting in the following expression for the number of inequalities necessary to define the polytopic feasible region

$$n_e = 2 + 2 \sum_{k=1}^m C(n_r - m + k - 1, k) \quad (\text{H.9})$$

## BIBLIOGRAPHY

- [And79] Anderson, B. D. O., and J. B. Moore. "Colored Noise and Suboptimal Reduced Order Filters." *Optimal Filtering*. New York: Dover Publications, 1979. 288-305.
- [Bal11] Baltagi, B. H. "Generalized Least Squares." *Econometrics*. 5<sup>th</sup> Ed. Berlin: Springer-Verlag, 2011. 223-240.
- [Bar01] Bar-Shalom, Y., X. R. Li, and T. Kirubarajan. "Adaptive Estimation and Maneuvering Targets." *Estimation with Applications to Tracking and Navigation*. New York: John Wiley & Sons, 2001. 441-465.
- [Bel97] Bellman, R. "Symmetric Functions, Kronecker Products and Circulants." *Introduction to Matrix Analysis*. 2<sup>nd</sup> Ed. Philadelphia: SIAM, 1997. 231-248.
- [Boe02] Boers, Y., and H. Driessen. "Hybrid State Estimation: A Target Tracking Application." *Automatica* 38.12 (2002): 2153-2158.
- [Boy94] Boyd, S., L. El Ghaoui, E. Feron, and V. Balakrishnan. "Linear Matrix Inequalities in System and Control Theory." *SIAM Studies in Applied Mathematics*. Vol. 15. Philadelphia: Society for Industrial and Applied Mathematics, 1994.
- [Boy04] Boyd, S., and L. Vandenberghe. "Convex Functions." *Convex Optimization*. New York: Cambridge University Press, 2004. 67-126.
- [Bro97] Brown, R. G., and P. Y. C Hwang. "The Discrete Kalman Filter, State-Space Modeling, and Simulation." *Introduction to Random Signals and Applied Kalman Filtering*. 3<sup>rd</sup> Ed. New York: John Wiley & Sons, 1997. 190-241.
- [Bry68] Bryson, A. E., and L. J. Henrikson. "Estimation Using Sampled Data Containing Sequentially Correlated Noise." *Journal of Spacecraft and Rockets* 5.6 (1968): 662-665.
- [Cha10] Chan, F. C., M. Joerger, and B. Pervan. "High Integrity Stochastic Modeling of GPS Receiver Clock for Improved Positioning and Fault Detection Performance." *Proceedings of the IEEE/ION Position, Location and Navigation Symposium*. Indian Wells, CA (May 2010): 1245-1257.
- [Che99] Chen, C-T. "Mathematical Descriptions of Systems." *Linear System Theory and Design*. 3<sup>rd</sup> Ed. New York: Oxford University Press, 1999. 5-43.
- [Cos05] Costa, O. L. V., M. D. Fragoso, and R. P. Marques. *Discrete-Time Markov Jump Linear Systems*. London: Springer-Verlag, 2005.

- [Dec00] DeCleene, B. "Defining Pseudorange Integrity-Overbounding." *Proceedings of the 13<sup>th</sup> International Technical Meeting of the Satellite Division of the Institute of Navigation*. Salt Lake City, UT (September 2000): 1916-1924.
- [Eld06] Eldar, Y. C. "Minimax MSE Estimation of Deterministic Parameters with Noise Covariance Uncertainties." *IEEE Transactions on Signal Processing* 54.1 (2006): 138-145.
- [For08] Forssell, B. "Aircraft Systems." *Radionavigation Systems*. Massachusetts: Artech House, Inc., 2008. 160-192.
- [Gel74] Gelb, A., ed. "Review of Underlying Mathematical Techniques." *Applied Optimal Estimation*. Cambridge, MA: The M.I.T. Press, 1974. 10-50.
- [Ger98] Geromel, J. C., and M. C. de Oliveira. " $H_2$  and  $H_\infty$  Robust Filtering for Convex Bounded Uncertain Systems." *Proceedings of the 37<sup>th</sup> IEEE Conference on Decision and Control*. Tampa, FL (December 1998): 146-151.
- [Ger00] Geromel, J. C., J. Bernussou, G. Garcia, and M. C. de Oliveira. " $H_2$  and  $H_\infty$  Robust Filtering for Discrete-Time Linear Systems." *SIAM Journal on Control and Optimization* 38.5 (2000): 1353-1368.
- [Ger02] Geromel, J. C., M. C. de Oliveira, and J. Bernussou. "Robust Filtering of Discrete-Time Linear Systems with Parameter Dependent Lyapunov Functions." *SIAM Journal on Control and Optimization* 41.3 (2002): 700-711.
- [Gil84] Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright. "Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints." *ACM Transactions on Mathematical Software* 10.3 (1984): 282-298.
- [Gor99] Goransson B., and B. Ottersten. "Direction Estimation in Partially Unknown Noise Fields." *IEEE Transactions on Signal Processing* 47.9 (1999): 2375-2385.
- [Ham92] Hambaba, M. L. "The Robust Generalized Least Squares Estimator." *Signal Processing* 26 (1992): 359-368.
- [Hwa06] Hwang, I., H. Balakrishnan, and C. Tomlin. "State Estimation for Hybrid Systems: Applications to Aircraft Tracking." *IEE Proceedings-Control Theory and Applications* 153.5 (2006): 556-566.
- [Jek01] Jekeli, C. "Inertial Navigation System." *Inertial Navigation Systems with Geodetic Applications*. New York: de Gruyter, 2001. 101-138.

- [Joe10] Joerger, M., L. Gratton, B. Pervan, and C. Cohen. "Analysis of Iridium-Augmented GPS for Floating Carrier Phase Positioning". *NAVIGATION* 57.2 (2010): 137-160.
- [Kas85] Kassam, S. A., H. V. Poor. "Robust Techniques for Signal Processing: A Survey." *Proceedings of the IEEE* 73.3 (1985): 433-481.
- [Kha10a] Khanafseh, S., S. Langel, and B. Pervan. "Overbounding Position Errors in the Presence of Carrier Phase Multipath Error Model Uncertainty." *Proceedings of the IEEE/ION Position, Location and Navigation Symposium*. Indian Wells, CA (May 2010): 575-584.
- [Kha10b] Khanafseh, S., and B. Pervan. "A New Approach for Calculating Position Domain Integrity Risk for Cycle Resolution in Carrier Phase Navigation Systems." *IEEE Transactions on Aerospace and Electronic Systems* 46.1 (2010): 1-12.
- [Kin02] Kincaid, D., and W. Cheney. "Numerical Solution of Ordinary Differential Equations." *Numerical Analysis: Mathematics of Scientific Computing*. 3<sup>rd</sup> Ed. Pacific Grove, CA: Brooks/Cole, 2002. 524-614.
- [Kou07] Koulaei, M. H., and T. Terlaky. "On the Extension of a Mehrotra-Type Algorithm for Semidefinite Optimization." Advanced Optimization Lab., Technical Report 4, Hamilton, Ontario: McMaster University Department of Computing and Software, 2007. 1-20.
- [Lan12] Langel, S., S. Khanafseh, and B. Pervan. "Bounding Integrity Risk in the Presence of Parametric Time Correlation Uncertainty." *Proceedings of the 2012 International Technical Meeting of the Institute of Navigation*. Newport Beach, CA (2012): 1666-1680.
- [Lue08] Luenberger, D. G., and Y. Yinyu. "Primal-Dual Methods." *Linear and Nonlinear Programming*. 3<sup>rd</sup> Ed. New York: Springer Science + Business Media, LLC, 2008. 469-506.
- [Lew08] Lewis, F. L., L. Xie, and D. Popa. *Optimal and Robust Estimation*. 2<sup>nd</sup> ed. Boca Raton: CRC Press, 2008.
- [May82] Maybeck, P. S. "Parameter Uncertainties and Adaptive Estimation." *Stochastic Models, Estimation, and Control*. Vol. 2. New York: Academic Press, 1982. 129-136.
- [Mis01] Misra, P., and P. Enge. "Precise Positioning With Carrier Phase." *Global Positioning System: Signals, Measurements and Performance*. 2<sup>nd</sup> ed. Massachusetts: Ganga-Jamuna Press, 2001. 233-280.

- [Pet96] Petersen, I. R., and D. C. McFarlane. "Optimal Guaranteed Cost Filtering for Uncertain Discrete-Time Linear Systems." *International Journal of Robust and Nonlinear Control* 6.4 (1996): 267-280.
- [Pul08] Pullford, G. W. "A Proof of the Spherically Symmetric Overbounding Theorem for Linear Systems." *NAVIGATION* 55.4 (2008): 283-292.
- [Rei92] Reilly, J. P., and K. M. Wong. "Estimation of the Directions of Arrival of Signals in Unknown Correlated Noise, Part II: Asymptotic Behavior and Performance of the MAP Approach." *IEEE Transactions on Signal Processing* 40.8 (1992): 2018-2028.
- [Rif04] Rife, J., S. Pullen, B. Pervan, and P. Enge. "Paired Overbounding and Application to GPS Augmentation." *Proceedings of the IEEE Position, Location and Navigation Symposium*. Monterey, CA (April 2004): 439-446.
- [Rif07] Rife, J., and D. Gebre-Egziabher. "Symmetric Overbounding of Correlated Errors." *NAVIGATION* 54.2 (2007): 109-124.
- [Rot84] Rothenberg, T. J. "Approximate Normality of Generalized Least Squares Estimates." *Econometrica* 54.2 (1984): 811-825.
- [Sat01] Satz, H., and T. H. Kerr. "Comparison of Batch and Kalman Filtering for Radar Tracking." *Proceedings of the 10<sup>th</sup> Annual AIAA/BMDO Technology Conference*. Williamsburg, VA (July 2001): 1-7.
- [Sch66] Schmidt, S. "Applications of State-Space Methods to Navigation Problems." *Advances in Control Systems*. Ed. C.T. Leondes. Vol. 3. New York: Academic Press, 1966.
- [Sha01] Shaked, U., L. Xie, and Y. C. Soh. "New Approaches to Robust Minimum Variance Filter Design." *IEEE Transactions on Signal Processing* 49.11 (2001): 2620-2629.
- [Sim06] Simon, D. *Optimal State Estimation: Kalman,  $H_\infty$ , and Nonlinear Approaches*. New Jersey: John Wiley & Sons, 2006.
- [Ste94] Stengel, R. F. "The Mathematics of Control and Estimation." *Optimal Control and Estimation*. New York: Dover Publications, 1994. 69-85.
- [Teu01a] Teunissen, P. J. G. "Statistical GNSS Carrier Phase Ambiguity Resolution: A Review." *Proceedings of the 11<sup>th</sup> IEEE Signal Processing Workshop on Statistical Signal Processing*. Singapore (August 2001): 4-12.

- [Teu01b] Teunissen, P. J. G. "GNSS Ambiguity Bootstrapping: Theory and Application." *Proceedings of the International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation*. Banff, Canada (June 2001): 246-254.
- [Teu03] Teunissen, P. J. G. "Theory of Carrier Phase Ambiguity Resolution." *Wuhan University Journal of Natural Sciences* 8.2B (2003): 471-484.
- [Teu10] Teunissen, P. J. G. "Mixed Integer Estimation and Validation for Next Generation GNSS." *Handbook of Geomathematics*. Ed. W. Freeden, M. Z. Nashed, and T. Sonar. Berlin: Springer-Verlag, 2010. 1101-1129.
- [Teu95] Teunissen, P. J. G. "The Least-Squares Ambiguity Decorrelation Adjustment: A Method for Fast GPS Integer Ambiguity Estimation." *Journal of Geodesy* 70.1-2 (1995): 65-82.
- [Teu99] Teunissen, P. J. G. "The Probability Distribution of the GPS Baseline for a Class of Integer Ambiguity Estimators." *Journal of Geodesy* 73.5 (1999): 275-284.
- [Tua01] Tuan H. D., P. Apkarian, and T. Q. Nguyen. "Robust and Reduced-Order Filtering: New LMI-Based Characterizations and Methods." *IEEE Transactions on Signal Processing* 49.12 (2001): 2975-2984.
- [Van78] Van Loan, C. F. "Computing Integrals Involving the Matrix Exponential." *IEEE Transactions on Automatic Control* 23.3 (1978): 395-404.
- [Van96] Vandenberghe, L, and S. Boyd. "Semidefinite Programming." *SIAM Review* 38.1 (1996): 49-95.
- [Ver05] Verhagen, S. "Integer Ambiguity Resolution." *The GNSS Integer Ambiguities: Estimation and Validation*. Delft, The Netherlands: Netherlands Geodetic Commission, 2005. 27-66.
- [Wal06] Wall, J. H., and D. M. Bevly. "Characterization of Inertial Sensor Measurements for Navigation Performance Analysis." *Proceedings of the 19<sup>th</sup> International Technical Meeting of the Satellite Division of the Institute of Navigation*. Fort Worth, TX (September 2006): 2678-2685.
- [Wie10] Wiesel, W. E. "Particle Dynamics." *Spaceflight Dynamics*. 3<sup>rd</sup> Ed. Beaver creek, OH: Aphelion Press, 2010. 47-78.
- [Won92] Wong, K. M., J. P. Reilly, Q. Wu, and S. Qiao. "Estimation of the Directions of Arrival of Signals in Unknown Correlated Noise, Part I: The MAP Approach and its Implementation." *IEEE Transactions on Signal Processing* 40.8 (1992): 2007-2017.

- [Wri97] Wright, S. J. *Primal-Dual Interior-Point Methods*. Philadelphia, PA: SIAM, 1997.
- [Xie91] Xie, L., C. E. de Souza, and M. Fu. "H<sub>∞</sub> Estimation for Discrete-Time Linear Uncertain Systems." *International Journal of Robust and Nonlinear Control* 1.2 (1991): 111-123.
- [Xie94] Xie, L., Y. C. Soh, and C. E. de Souza. "Robust Kalman Filtering for Uncertain Discrete-Time Systems." *IEEE Transactions on Automatic Control* 39.6 (1994): 1310-1314.
- [Xie04] Xie, L., L. Lu, D. Zhang, and H. Zhang. "Improved Robust H<sub>2</sub> and H<sub>∞</sub> Filtering for Uncertain Discrete-Time Systems." *Automatica* 40.5 (2004): 873-880.
- [Xie05] Xie, L. "On Robust H<sub>2</sub> Estimation." *ACTA Automatica Sinica* 31.1 (2005): 1-12.
- [Yan04] Yang, Y., R. R. Hatch, and R. T. Sharpe. "GPS Multipath Mitigation in Measurement Domain and its Applications for High Accuracy Navigation." *Proceedings of the 17<sup>th</sup> International Technical Meeting of the Satellite Division of the Institute of Navigation*. Long Beach, CA (September 2004): 1124-1130.
- [Ye95] Ye, H., and R. D. DeGroat. "Maximum Likelihood DOA Estimation and Asymptotic Cramer-Rao Bounds for Additive Unknown Colored Noise." *IEEE Transactions on Signal Processing* 43.4 (1995): 938-949.
- [Zhu02] Zhu, X., Y. C. Soh, and L. Xie. "Design and Analysis of Discrete-Time Robust Kalman Filters." *Automatica* 38.6 (2002): 1069-1077.
- [Zym08] Zymnis, A., S. Boyd, and D. Gorinevsky. "Mixed State Estimation for a Linear Gaussian Markov Model." *Proceedings of the 47<sup>th</sup> IEEE Conference on Decision and Control*. Cancun, Mexico (December 2008): 3219-3226.