



NGA.SIG.0026.03\_1.0\_ACCPRED  
2016-11-17

# **NGA STANDARDIZATION DOCUMENT**

## **Accuracy and Predicted Accuracy in the NSG: Predictive Statistics**

### **Technical Guidance Document (TGD) 2a**

**(2016-11-17)**

**Version 1.0**

## Forward

This handbook is approved for use by all Departments and Agencies of the Department of Defense.

Comments, suggestions, or questions on this document should be addressed to the GWG World Geodetic System (WGS) and Geomatics (WGSG) Focus Group, ATTN : Chair, WGS/Geomatics Standards Focus Group, [ncgis-mail.nga.mil](mailto:ncgis-mail.nga.mil) or to the National Geospatial-Intelligence Agency Office of Geomatics (SFN), Mail Stop L-41, 3838 Vogel Road, Arnold, MO 63010 or emailed to [GandG@nga.mil](mailto:GandG@nga.mil).

## Summary of Changes and Modifications

Revision	Date	Status	Description
1.0	20161117	FINAL	INITIAL PUBLICATION

## Contents

<b>Forward</b> .....	ii
Summary of Changes and Modifications .....	ii
1 Scope.....	1
2 Applicable Documents .....	3
2.1 Government specifications, standards, and handbooks.....	3
3 Definitions.....	4
3.1 Key Terms Used in the Document.....	5
3.1.1 Accuracy.....	5
3.1.2 Circular Error .....	5
3.1.3 Error .....	5
3.1.4 Fusion.....	5
3.1.5 Linear Error .....	5
3.1.6 Monte-Carlo Simulation.....	5
3.1.7 National System for Geospatial Intelligence (NSG) .....	5
3.1.8 Predicted Accuracy.....	6
3.1.9 Predictive Statistics .....	6
3.1.10 Quality Assurance .....	6
3.1.11 Quality Assessment.....	6
3.1.12 Sample Statistics .....	6
3.1.13 Scalar Accuracy Metrics .....	6
3.1.14 Spherical Error (SE) .....	6
3.1.15 Statistical Error Model .....	6
3.1.16 Validation .....	7
3.1.17 Variance .....	7
3.1.18 Verification.....	7
3.2 Other Relevant Terms .....	8
3.3 Abbreviations and Acronyms .....	9
4 Introduction to Predictive Statistics in the NSG .....	10
4.1 Guide to Detailed Technical Content.....	11

5	Methodologies and Algorithms in Predictive Statistics .....	13
5.1	Fundamentals of Predictive Statistics .....	13
5.1.1	Random Error Vector statistic definitions.....	15
5.1.1.1	Underlying probabilistic foundations.....	17
5.1.2	Example: scalar Gaussian pdf and related statistics .....	17
5.1.3	Example: Multi-variate Gaussian pdf and related statistics .....	19
5.1.4	Conditional expectations and correlation.....	21
5.1.5	Coordinate Systems .....	22
5.2	Error Covariance Matrices: Properties and Rigorous Descriptors .....	22
5.2.1	Formal derivation/definition of a valid error covariance matrix .....	23
5.2.2	Additional properties of related matrices.....	25
5.2.3	Possible issues with the mean-value .....	25
5.2.4	Assurance of valid and realistic error covariance matrices required for practical applications.....	26
5.3	Error Ellipsoids .....	28
5.3.1	Error Ellipsoid Examples.....	30
5.3.2	Derivation of relationship between probability $p$ & ellipsoidal normalized distance $d$ .....	34
5.3.3	Additional properties of the Error Ellipsoid .....	36
5.3.4	Rendering the Error Ellipsoid .....	37
5.3.5	Comparison of Covariance Matrices .....	38
5.3.6	Error ellipsoids: intersection and union.....	42
5.4	Predictive Scalar Accuracy Metrics: Linear Error, Circular Error, and Spherical Error .....	44
5.4.1	Linear Error (LE) .....	50
5.4.1.1	Algorithm for Computing LE_XX .....	51
5.4.1.2	Examples of LE_XX computation.....	51
5.4.2	Circular Error (CE).....	51
5.4.2.1	Baseline Computation Method: Table Interpolation.....	54
5.4.2.2	Alternate Computation Method: Monte Carlo Matrix Square Root.....	55
5.4.2.3	Examples of Monte Carlo Matrix Square Root Method.....	56
5.4.2.4	CE_XX Computation Method Selection .....	58
5.4.2.5	Examples of CE_XX computation .....	59
5.4.2.6	CE_XX Performance Summary .....	60

5.4.3	Spherical Error (SE) .....	61
5.4.3.1	Baseline Computation Method: Table Interpolation .....	63
5.4.3.2	Alternate Computation Method: Monte Carlo Matrix Square Root.....	64
5.4.3.3	SE_XX Computation Method Selection .....	64
5.4.3.4	Examples of SE_XX computation .....	65
5.4.3.5	SE_XX Performance Summary .....	66
5.5	Multi-state vector Error Covariance Matrix Definition .....	67
5.5.1	Details regarding a single state vector .....	68
5.5.2	Details regarding two different state vectors .....	68
5.5.3	Applicability of definitions .....	69
5.5.4	Generation of the Relative Error Covariance Matrix .....	69
5.6	Propagation of Multi-State Vector Error Covariance Matrices.....	70
5.6.1	Error Covariance Representation in Different Coordinate Systems .....	70
5.6.2	First-order Taylor Series Expansion .....	71
5.6.3	Other propagations.....	71
5.7	Generic Methods for Generation of the Multi-State Vector Error Covariance Matrix.....	72
5.7.1	<i>A priori</i> modeling.....	72
5.7.1.1	Gauss-Markov as an underlying error model .....	73
5.7.1.2	Use of sample statistics.....	75
5.7.2	Batch WLS .....	76
5.7.3	Kalman filter or smoother.....	77
5.7.3.1	Kalman filter example using the “A matrix” required for cross-covariance .....	78
5.8	Generic Methods for Representation/Dissemination of the Multi-State Vector Error Covariance Matrix .....	79
5.8.1	Direct method .....	79
5.8.2	“A matrix” method.....	80
5.8.3	Spdcf method .....	80
5.8.3.1	Correlation subgroups .....	81
5.8.3.2	Spdcf properties and examples.....	82
5.8.4	Bandwidth .....	85

5.9	Approximation of the Multi-State Vector Error Covariance Matrix .....	86
5.9.1	Spdcf Method: approximation example .....	87
5.9.2	Future bandwidth-reduction research.....	94
5.10	Overview of References by Section .....	95
6	Notes.....	96
6.1	Intended Use.....	96
7	References .....	96
Appendix A: Additional Terms and Definitions.....		98
Appendix B: Pseudo-code for Rendering the Error Ellipsoid .....		111
B.1	Plot Error Ellipse.....	111
B.2	Plot Error Ellipsoid .....	112
Appendix C: Supplementary Material on Predictive Scalar Accuracy Metrics .....		115
C.1	Solution Comparisons Supporting Sections 5.4.2.6 (CE) and 5.4.3.5 (SE) .....	115
C.1.1	Circular Error (CE) .....	116
C.1.1.1	CE Method Repeatability Tests.....	117
C.1.1.2	CE Method versus Method Comparison Tests.....	120
C.1.2	Spherical Error (SE) .....	121
C.1.2.1	SE Method Repeatability Tests .....	122
C.1.2.2	SE Method versus Method Comparison Tests.....	125
C.1.3	Summary of Tests.....	126
C.1.4	Miscellaneous other Tests .....	127
C.2	Complete Set of SE Interpolation Tables .....	127
C.3	Pseudo-code .....	130
C.3.1	Pseudo-code for LE .....	130
C.3.2	Pseudo-code for CE.....	131
C.3.3	Pseudo-code for SE .....	135

# 1 Scope

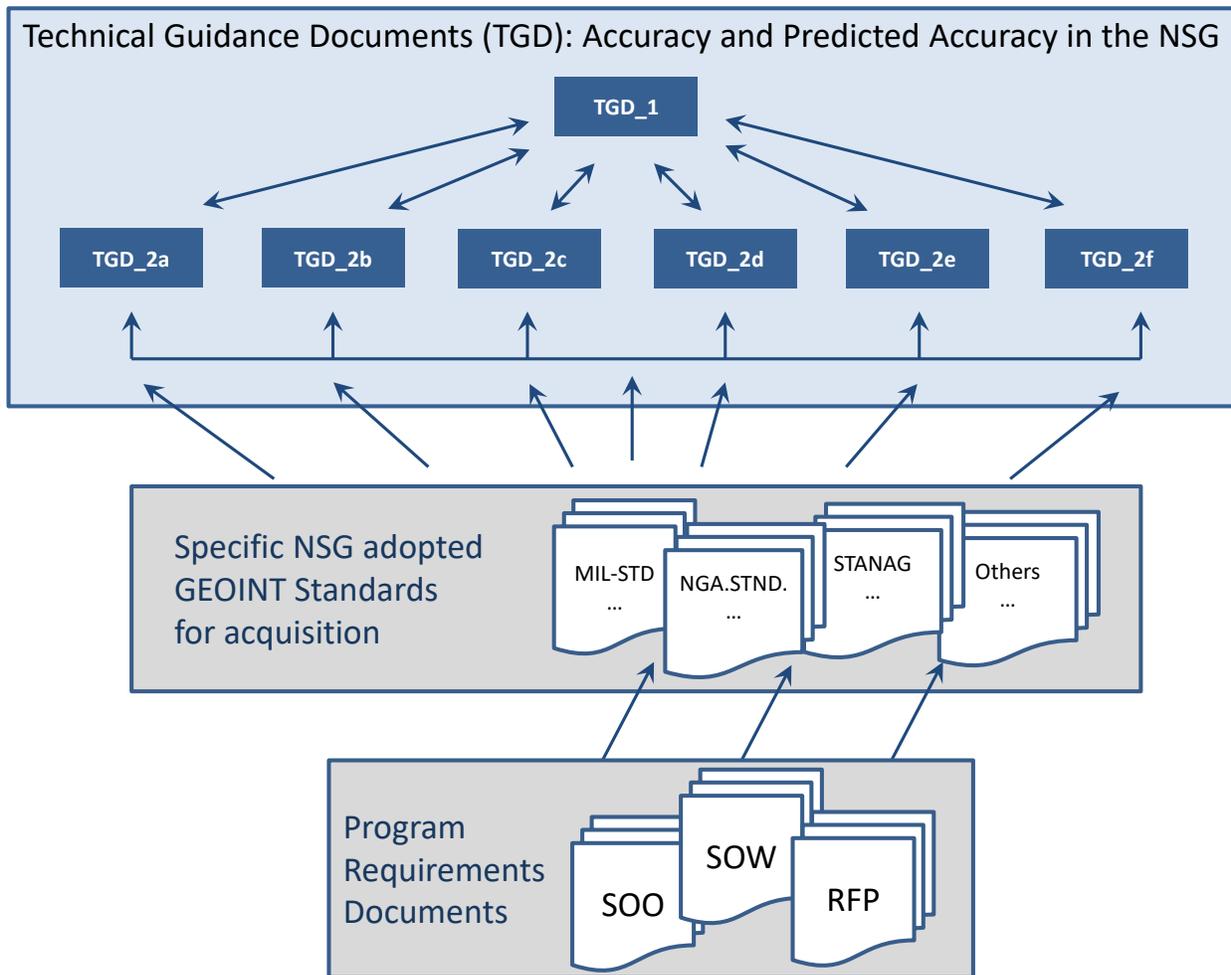
This Technical Guidance Document (TGD 2a) is a specific topic document on Predictive Statistics, part of a series of information and guidance documents regarding Accuracy and Predicted Accuracy in the National System for Geospatial Intelligence (NSG). As the title suggests, it focuses on methods, practices and applications of predictive statistics within the context of a larger scope of work which includes a more generalized overview and additional topic specific technical guidance. Documents in this series are listed below:

TGD 1	Accuracy and Predicted Accuracy in the NSG:	Overview and Methodologies
TGD 1-G	Accuracy and Predicted Accuracy in the NSG:	Glossary of Terms
TGD 2a	Accuracy and Predicted Accuracy in the NSG:	Predictive Statistics
TGD 2b	Accuracy and Predicted Accuracy in the NSG:	Sample Statistics
TGD 2c	Accuracy and Predicted Accuracy in the NSG:	Specification and Validation
TGD 2d	Accuracy and Predicted Accuracy in the NSG:	Estimators and Quality Control
TGD 2e	Accuracy and Predicted Accuracy in the NSG:	Monte-Carlo Simulation
TGD 2f	Accuracy and Predicted Accuracy in the NSG:	External Data and Quality Assessment

All documents in the series, “Accuracy and Predicted Accuracy in the NSG”, are intended to provide technical guidance to inform the development of geospatial data accuracy characterization for NSG GEOINT collectors, producers and consumers -- accuracy characterization as required to describe the trustworthiness of geolocations for defense and intelligence use and to support practices that acquire, generate, process, exploit, and provide geolocation data and information based on geolocation data. Today, both the sources and desired uses for geospatial data are quickly expanding. Throughout the NSG, trusted conveyance of geospatial accuracy is broadly required for a variety of traditional and evolving missions including those supported by manual, man-in-the-loop, and automated processes. This guidance is the foundation layer for a collection of common techniques, methods, and algorithms ensuring that geospatial data within the NSG can be clearly requested, delivered and evaluated as fit for desired purpose whether by decision makers, intelligence analysts, or as input to further processing techniques.

TGD 2a contains references to and is referenced by other Technical Guidance Documents. The documents in this series, TGD 1 and TGD 2b – TGD 2f, also have cross-references among themselves. All Technical Guidance Documents also reference external public as well as “NGA approved for public release” documents for further insight/details. While each individual document contains definitions for important relevant terms, TGD 1-G compiles all important terms and respective definitions of use particular to this series of documents to ensure continuity and provide ease of reference.

The TGD 2 documents, including this document focused on predictive statistics, are also considered somewhat top-level in that they are not directed at specific systems. They do provide general guidance, technical insight, and recommended algorithms. The relationship of the Technical Guidance Documents with specific GEOINT Standards documents and specific Program Requirements documents is presented in Figure 1-1, where arrows refer to references. That is, in general, specific product requirement documents reference specific GEOINT standards documents which reference specific technical guidance documents.



**Figure 1-1:** The relationships between the Technical Guidance Documents, GEOINT Standards Documents, and Program Requirement Documents

Accuracy and Predicted Accuracy in the NSG: Predictive Statistics, Technical Guidance Document (TGD) 2a is for guidance only and cannot be cited as a requirement.

## 2 Applicable Documents

The documents listed below are not necessarily all of the documents referenced herein, but are those needed to understand the information provided by this information and guidance document.

### 2.1 Government specifications, standards, and handbooks

NGA.SIG.0026.01\_1.0\_ACCOVER, Accuracy and Predicted Accuracy in the NSG: Overview and Methodologies, Technical Guidance Document (TGD) 1

NGA.SIG.0026.02\_1.0\_ACCGLOS, Accuracy and Predicted Accuracy in the NSG: Glossary of Terms, Technical Guidance Document (TGD) 1-G

NGA.SIG.0026.04\_1.0\_ACCSAMP, Accuracy and Predicted Accuracy in the NSG: Sample Statistics, Technical Guidance Document (TGD) 2b

NGA.SIG.0026.05\_1.0\_ACCSPEC, Accuracy and Predicted Accuracy in the NSG: Specification and Validation, Technical Guidance Document (TGD) 2c

NGA.SIG.0026.06\_1.0\_ACCESQC, Accuracy and Predicted Accuracy in the NSG: Estimators and Quality Control, Technical Guidance Document (TGD) 2d

NGA.SIG.0026.07\_1.0\_ACCMTCO, Accuracy and Predicted Accuracy in the NSG: Monte-Carlo Simulation, Technical Guidance Document (TGD) 2e

NGA.SIG.0026.08\_1.0\_ACCXDQA, Accuracy and Predicted Accuracy in the NSG: External Data and Quality Assessment, Technical Guidance Document (TGD) 2f

### 3 Definitions

There are a number of authoritative guides as well as existing standards within the NSG and Department of Defense for definitions of the identified key terms used in this technical guidance document. In many cases, the existing definitions provided by these sources are either too general or, in some cases, too narrow or dated by intended purposes contemporary to the document's development and publication. The definitions provided in this document have been expanded and refined to explicitly address details relevant to the current and desired future use of accuracy in the NSG. To acknowledge the basis and/or lineage of certain terms Section 3.1, we reference the following sources considered as either foundational or contributory:

- [a] Anderson, James M. and Mikhail, E., *Surveying: Theory and Practice*, 7<sup>th</sup> Edition, WCB/McGraw-Hill, 1998.
- [b] DMA-TR-8400.1, DMA Technical Report: Error Theory as Applied to Mapping, Charting, and Geodesy.
- [c] Defense Mapping Agency, *Glossary of Mapping, Charting, and Geodetic Terms*, 4<sup>th</sup> Edition, Defense Mapping Agency Hydrographic/Topographic Center, 1981.
- [d] ISO TC/211 211n2047, Text for ISO 19111 Geographic Information - Spatial referencing by coordinates, as sent to the ISO Central Secretariat for issuing as FDIS, July 17, 2006.
- [e] Joint Publication (JP) 1-02, Department of Defense Dictionary of Military and Associated Terms, November 8, 2010 as amended through January 15, 2016.
- [f] MIL-HDBK-850, *Military Handbook: Glossary of Mapping, Charting, and Geodetic Terms*, January 21, 1994.
- [g] MIL-STD-2401, Department of Defense Standard Practice; Department of Defense World Geodetic System (WGS), January 11, 1994
- [h] MIL-STD-600001, Department of Defense Standard Practice; Mapping, Charting and Geodesy Accuracy, February 26, 1990.
- [i] *National System for Geospatial Intelligence* [Brochure] Public Release Case #15-489.
- [j] NGA.STND.0046\_1.0, *The Generic Point-cloud Model (GPM): Implementation and Exploitation*, Version 1.0, October 03, 2015.
- [k] Oxford Dictionaries ([www.oxforddictionaries.com/us/](http://www.oxforddictionaries.com/us/)) copyright © 2016 by Oxford University Press.
- [l] Soler, Tomas and Hothem, L., "Coordinate Systems Used in Geodesy: Basic Definitions and Concepts", *Journal of Surveying Engineering*, Vol. 114, No. 2, May 1988.

## 3.1 Key Terms Used in the Document

### 3.1.1 Accuracy

The range of values for the error in an object's metric value with respect to an accepted reference value expressed as a probability. [f]

- Statements of accuracy may be developed through applications of predictive statistics or by sample statistics based on multiple independent samples of errors.

### 3.1.2 Circular Error

See **Scalar Accuracy Metrics**.

### 3.1.3 Error

The difference between the observed or estimated value and its ideal or true value. See Appendix A for a more detailed and augmented definition. [f]

### 3.1.4 Fusion

A process that combines or relates different sources of (typically independent) information.

### 3.1.5 Linear Error

See **Scalar Accuracy Metrics**.

### 3.1.6 Monte-Carlo Simulation

A technique in which a large number of independent sample inputs for a system are randomly generated using an assumed *a priori* statistical model to analyze corresponding system output samples statistically and support derivation of a statistical model of the system output. This technique is valuable for complex systems, non-linear systems, and those where no insight to internal algorithms is provided ("black box" systems).

### 3.1.7 National System for Geospatial Intelligence (NSG)

The operating framework supported by producers, consumers or influencers of geospatial intelligence (GEOINT). Spanning defense, intelligence, civil, commercial, academic and international sectors, the NSG contributes to the overall advancement of the GEOINT function within the strategic priorities identified by the Functional Manager for Geospatial Intelligence in the role established by Executive Order 12333. The framework facilitates community strategy, policy, governance, standards and requirements to ensure responsive, integrated national security capabilities. [i]

### 3.1.8 Predicted Accuracy

The range of values for the error in a specific object's metric value expressed as a probability derived from an underlying and accompanying detailed statistical error model.

- If the statistical error model does not include the identification of a specific probability distribution, a Gaussian (or Normal) probability distribution is typically assumed in order to generate probabilities.
- The term "Predicted" in Predicted Accuracy corresponds to the use of predictive statistics in the detailed statistical error model; it does not correspond to a prediction of accuracy applicable to the future since the corresponding error corresponds to a geolocation already extracted.

### 3.1.9 Predictive Statistics

Statistics corresponding to the mathematical modeling of assumed *a priori* error characteristics contained in a statistical error model.

### 3.1.10 Quality Assurance

The maintenance of a desired level of quality in a service or product, especially by means of attention to every stage of the process of delivery or production. [k]

### 3.1.11 Quality Assessment

Processes and procedures intended to verify the reliability of provided data and processes, typically performed independent of collection or production. For example, If ground truth is available, then comparison of actual (sample) errors to predicted errors (statistical values via rigorous error propagation) is a key part of this process.

### 3.1.12 Sample Statistics

Statistics corresponding to the analysis of a collection of physical observations, a sample of the population, as compared to an assumed true or an *a priori* value.

### 3.1.13 Scalar Accuracy Metrics

Convenient one-number summaries of geolocation accuracy and geolocation predicted accuracy expressed as a probability: (1) Linear Error (LE) corresponds to 90% probable vertical error, (2) Circular Error (CE) correspond to 90% probable horizontal radial error, and (3) Spherical Error (SE) corresponds to 90% spherical radial error. See Appendix A for a more detailed and augmented definition. [b],[f], and [h]

### 3.1.14 Spherical Error (SE)

See **Scalar Accuracy Metrics**.

### 3.1.15 Statistical Error Model

Information which describes the error data corresponding to a given state vector. The information includes the type of corresponding error representation (random variable, random vector, stochastic process, or random process), the category of statistics (predictive or sample), and associated statistical information including at a minimum the mean-value and covariance data.

**3.1.16 Validation**

The process of determining the degree to which a model is an accurate representation of the real world from the perspective of its intended use/s. In the NSG, this includes validation of accuracy and predicted accuracy specified capabilities. [e]

**3.1.17 Variance**

The measure of the dispersion of a random variable about its mean-value, also the standard deviation squared. [b]

**3.1.18 Verification**

The process of determining that an implemented model accurately represents the developer's conceptual description and specifications. [e]

### 3.2 Other Relevant Terms

Appendix A contains definitions of the following additional terms relevant to the content of this document:

- *A priori*
- *A posteriori*
- Absolute Horizontal Accuracy
- Absolute Vertical Accuracy
- Bias Error
- CE-LE Error Cylinder
- Confidence Ellipsoid
- Correlated Error
- Correlated Values
- Covariance
- Covariance Function
- Covariance Matrix
- Cross-covariance Matrix
- Deterministic Error
- Distance Constant
- Earth Centered Earth Fixed Cartesian Coordinate System
- Error (augmented definition)
- Error Ellipsoid
- Estimator
- Gaussian (or Normal) probability distribution
- Geodetic Coordinate System
- Ground Truth
- Homogeneous
- Horizontal Error
- Inter-state Vector Correlation
- Intra-state Vector Correlation
- Local Tangent Plane Coordinate System
- Mean-Value
- Metadata
- Multi-image Geopositioning (MIG)
- Multi-state Vector Error Covariance Matrix
- Order Statistics
- Percentile
- Precision
- Principal Matrix Square Root
- Probability density function
- Probability distribution
- Probability distribution function
- Radial Error
- Random Error
- Random Error Vector
- Random Field
- Random Variable
- Random Vector
- Realization
- Relative Horizontal Accuracy
- Relative Vertical Accuracy
- Scalar Accuracy Metrics (augmented definition)
- Sensor support data
- Spatial Correlation
- Standard Deviation
- State Vector
- State Vector Error
- Stationary
- Stochastic Process
- Strictly Positive Definite Correlation Function (spdcf)
- Systematic Error
- Temporal Correlation
- Time Constant
- Uncertainty
- Uncorrelated Error
- Uncorrelated Values
- Vertical Error
- WGS84

### 3.3 Abbreviations and Acronyms

Abbreviation/Acronym	Definition
1d	One Dimensional
2d	Two Dimensional
3d	Three Dimensional
cdf	cumulative probability distribution function
CE	Circular Error
CSM	Community Sensor Model
ECF	Earth Centered Fixed
ENU	East North Up
GEOINT	Geospatial Intelligence
GPS	Global Positioning System
LE	Linear Error
NSG	National System for Geospatial Intelligence
pdf	probability density function
SE	Spherical Error
spdcf	strictly positive definite correlation function
TGD	Technical Guidance Document
UAV	Unmanned Aerial Vehicle
WGS84	World Geodetic System 1984
WLS	Weighted Least Squares

## 4 Introduction to Predictive Statistics in the NSG

This document describes predictive statistics and provides detailed technical guidance regarding their recommended use in the NSG. We first start with some background definitions required for context:

Accuracy in the NSG is defined as: “the range of values for the error in an object’s metric value expressed as a probability”. Furthermore, this general definition can be sub-allocated to more specific accuracies. For example, we can define horizontal accuracy for a specific system as: “the 90<sup>th</sup> percentile of horizontal (radial) geolocation error, where location is relative to a specified geodetic reference system”.

Predicted accuracy in the NSG is defined as: “the range of values for the error in a specific object’s metric value expressed as a probability derived from an underlying and accompanying detailed statistical error model.” The detailed statistical error model includes predictive statistics when in an operational environment. (Sample statistics are used for accuracy/performance validation and verification, as well as inputs to the *a priori* modelling of predictive statistics.) Underlying errors are represented as random vectors (variables), stochastic processes, and random fields.

A top-level discussion of accuracy, predicted accuracy, predictive statistics, sample statistics, and their various differences and interrelationships are provided in TGD 1: “Accuracy and Predicted Accuracy in the NSG: Overview and Methodologies”. Predicted accuracy is identified as critical to the optimal and reliable performance of an NSG system. Predictive statistics are identified as the key component of predicted accuracy.

Now that background definitions have been presented, we go on to present an overview of predictive statistics per section of this document:

Section 5.1 begins by defining relevant predictive statistics: mean-value, covariance matrix, probability density function (pdf), and strictly positive definite correlation function (spdcf). The first three of these statistics correspond to a single, multi-component ( $nx1$ ) state vector, and its corresponding multi-component state vector error, an  $nx1$  “random vector”.

As discussed in Section 5.1, the most important and practical predictive statistic is the  $nxn$  error covariance matrix, and as such, Section 5.2 presents important properties and rigorous descriptors. Section 5.3 then goes on to describe how to compute and interpret the error ellipsoid, which is a rigorous, equivalent, and important visual counterpart to the error covariance matrix. Section 5.4 details practical approximations of the error covariance matrix and accuracy: the ubiquitous scalar accuracy metrics LE, CE, and SE.

In the remaining sections we “switch gears” to use of the more general multi-state vector and its corresponding multi-state vector random error and error covariance matrix. Section 5.5 first defines the multi-state vector as simply consisting of a collection of individual state vectors, all “stacked” together. It then defines its corresponding multi-state vector error covariance matrix as consisting of the previously defined error covariance matrices for each of the individual state vectors, plus the various cross-covariance matrices corresponding to each pair of individual state vectors.

Most, but not all, of the sections following Section 5.5 make use of the multi-state vector error covariance matrix. It is a practical predictive statistic for a collection of related individual state vectors and their random error vectors. Such a collection may consist of temporally correlated state vector errors corresponding to a time series of realizations from a stochastic process, such as a time series of satellite position and/or attitude (pointing) metadata and their errors. Alternatively, such a collection may consist of individual state vectors adjusted simultaneously in a batch weighted least squares (WLS) estimator and their subsequent solution errors.

Section 5.6 discusses propagation of the multi-state vector error covariance matrix, an important process in “rigorous error propagation”. Section 5.7 discusses generic methods for the generation of multi-state vector error covariance matrices, including the use of spdcf. Section 5.8 discusses generic methods for the practical representation and dissemination of the multi-state vector error covariance matrix. Section 5.9 details the approximation of the multi-state vector error covariance matrix via spdcf for bandwidth reduction, as many of these error covariance matrices can be very large. Of course, in all of these sections, as well as proceeding sections, such multi-state vector error covariance matrices must be theoretically valid (positive definite, invertible, etc.), and corresponding generation, representation, and dissemination techniques presented guarantee as such.

## 4.1 Guide to Detailed Technical Content

The following is a corresponding top-level roadmap of Section 5 of this document, Methodologies and Algorithms in Predictive Statistics:

- Section 5.1: Fundamentals of Predictive statistics
  - Role in the NSG
  - Differences from sample statistics
  - Overview: mean, error covariance matrix, probability density function (pdf), strictly positive definite correlation function (spdcf)
  - Definitions of mean, error covariance matrix from the pdf
  - The Gaussian or Normal multi-variate pdf
- Section 5.2: Properties and rigorous descriptors of the error covariance matrix
  - Positive definite; hence, positive eigenvalues and invertible
  - Captures correlation between error components – essential to many applications
  - “Valid” versus “pseudo-valid” versus “invalid” error covariance matrices; “realistic” error covariance matrices
- Section 5.3: Error ellipsoids: equivalent and visual descriptions of the error covariance matrix
  - How to interpret, compute, and render
  - (Covariance matrix B)  $\succeq$  (Covariance matrix A) and implications
  - Definition and applications of the “union error ellipsoid” and “intersection error ellipsoid”
- Section 5.4: Scalar accuracy metrics: Linear Error, Circular Error, and Spherical Error
  - Definitions and how to interpret
  - Ubiquitous and in need of computational standardization

- Rigorous derivations and practical algorithms to compute
- Section 5.5: Multi-state vector error covariance matrix definition
  - Error covariance matrix and error cross-covariance matrix
  - Intra-state vector correlation and inter-state vector correlation
  - Definition and computation of the relative error covariance matrix
- Section 5.6: Propagation of the multi-state vector error covariance matrix
  - Methods and properties
- Section 5.7: Generic methods for generation of the multi-state vector error covariance matrix
  - *A priori* modeling and use of sample statistics
  - Batch and real-time estimator output
- Section 5.8: Generic methods for representation/dissemination of the multi-state vector error covariance matrix
  - Direct, “A matrix”, spdcf
- Section 5.9: Approximation of the multi-state vector error covariance matrix
  - Summary methods using CE and LE over geographic regions are problematic
  - Spdcf method ensures a valid error covariance matrix and bandwidth compression
- Section 5.10: An overview of useful references and their content relative to various sections of the document. Some are also referenced directly in the text, particularly when content was deemed essential but space limited.

Some sections contain examples that are based on various aspects of image-based geopositioning; however, the same principles apply across the entire scope of the NSG.

## 5 Methodologies and Algorithms in Predictive Statistics

### 5.1 Fundamentals of Predictive Statistics

Predictive statistics are used throughout the NSG. They form important inputs and outputs between various collection, value-added processing, and exploitation modules. There are four general predictive statistics for associated errors:

- Mean-value
- Covariance matrix
- Probability density function
- Strictly positive definite correlation function.

The above predictive statistics actually correspond to  $n \times 1$  random error vectors associated with  $n \times 1$  state vectors within various modules of the NSG, i.e., errors in the state vectors' values relative to "truth".

The term "random error vector" is more precisely defined as a random vector representation of error. As such, a random error vector contains  $n$  random errors as components, each represented as a random variable. The "origin" of a random error vector can either be "stand-alone", or associated with a particular time in a stochastic process, or associated with a particular location in a random field, as discussed in TGD 1. A component in a random error vector may be correlated with other components in the same random error vector (intra-state vector correlation). In addition, a random error vector may be correlated with other random error vectors (inter-state vector correlation), i.e., their components are correlated.

A random error vector's  $n \times 1$  mean-value and its  $n \times n$  covariance matrix about that mean-value are standard and reasonably well-defined predictive statistics. The mean-value of a predictive error is typically zero unless specifically stated otherwise, i.e., all of its  $n$  components are equal to zero.

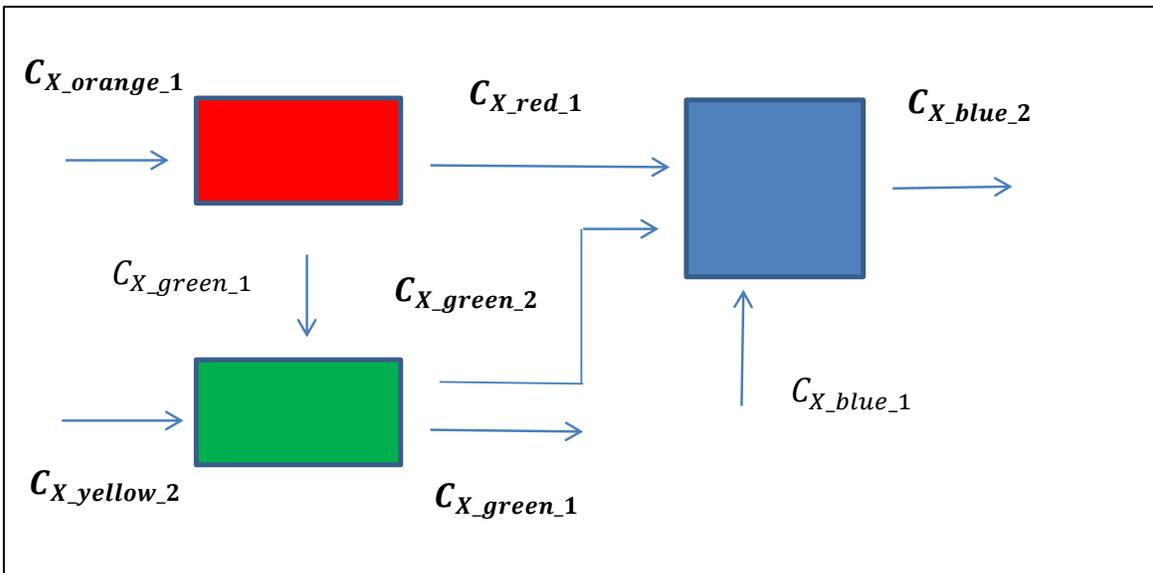
A random error vector's probability density function is an optional "statistic". It defines the probability distribution of underlying error components or random variables. If it corresponds to a Gaussian or Normal multi-variate distribution (the two terms are used interchangeably), its identity as such, along with the aforementioned mean and covariance, completely specify the distribution. The term "multi-variate" is used when the number of components in the random error vector is greater than 1, i.e.,  $n > 1$ . If a (multi-variate) probability density function is not identified (defined), it is usually assumed Gaussian. Note that for many processes, identification of a specific (multi-variate) probability distribution is not required. For example, only the mean-value and error covariance matrix are required in order to implement a Best Linear Unbiased Estimator. No specific probability density need be assumed.

Strictly positive definite correlation functions (spdcf) are used to model temporal correlation or spatial correlation between various random error vectors which typically correspond to a stochastic process at specific times or a random field at specific spatial locations, respectively. The appropriate evaluation of an spdcf allows for generation of the cross-covariance matrix between the random error vectors. The

cross-covariance matrices are placed into the more general multi-state vector error covariance matrices discussed in Section 5.5. Like the probability density functions, spdcf are optional “statistics”, and typically characterized by a few parameters, such as temporal (de)correlation time constants.

Predictive statistics are “modelled” statistics, in that they correspond to an *a priori* (mathematical) model or are the output of a computational process, like an estimator. They are in contrast to sample statistics, which are typically generated “off-line” from a set of sampled errors using corresponding “ground truth”. Of course, there is interplay between the two types of statistics: predictive statistics affect system errors which are then (occasionally) sampled. And sample errors can be used to better refine the predictive statistics and underlying predictive error models.

Finally, various predictive statistics throughout the NSG are inter-related in that they are both module inputs and outputs. They affect each other, as outlined in Figure 5.1-1. This is one reason why it is so important to properly generate and disseminate them within a given module.



**Figure 5.1-1:** The interrelationship between predictive statistics, as represented by the error covariance matrix  $C_X$ , across different (color-coded) modules in the NSG

In the above figure, the primary predictive statistic, the error covariance matrix  $C_X$ , is illustrated. Bold-face text indicates an error covariance matrix generated by the identified module, typically an *a posteriori* solution error covariance matrix generated by an estimator in the module. This error covariance matrix is an output of the identified module and may be an input to one or more “down-stream” modules. Non-bold-face text indicates *a priori* information input to a module, such as an error covariance matrix (or parameters to generate this matrix) corresponding to measurement errors, used by an estimator in the module in order to weight the measurements used in the solution.

We now go on to define the predictive statistics for a random error vector. The definition for the spdcf refers to multiple random error vectors and is postponed until Section 5.8.

### 5.1.1 Random Error Vector statistic definitions

In the following definitions, the superscript T indicates vector (or matrix) transpose, the “overbar” indicates mean-value,  $E\{\}$  indicates expected value,  $\epsilon x_i$  indicates random variable, and all numbers are assumed real-valued. Both  $E\{\}$  and random variables are defined themselves near the end of the definitions, along with the probability density function  $pdf_X$ .

The state vector represents an arbitrary column vector of quantities of interest. Its value differs from the (unknown) true value by a random error vector:

**State vector:**

$$X = [X(1) \quad \dots \quad X(n)]^T \equiv [x_1 \quad \dots \quad x_n]^T, \quad (5.1.1-1)$$

an  $n \times 1$  column vector;

**Random error vector:**

$$\epsilon X = [\epsilon X(1) \quad \dots \quad \epsilon X(n)]^T \equiv [\epsilon x_1 \quad \dots \quad \epsilon x_n]^T, \quad (5.1.1-2)$$

an  $n \times 1$  column vector;

**Mean-value** of the random error vector:

$$\bar{\epsilon X} = E\{\epsilon X\} = [E\{\epsilon x_1\} \quad \dots \quad E\{\epsilon x_n\}]^T, \quad (5.1.1-3)$$

an  $n \times 1$  column vector;

**Error covariance matrix** of the random error vector about its mean-value:

$$C_X = E\{(\epsilon X - \bar{\epsilon X})(\epsilon X - \bar{\epsilon X})^T\} = \begin{bmatrix} \dots & \dots & \dots & \dots \\ \dots & E\{(\epsilon x_i - \bar{\epsilon x}_i)(\epsilon x_j - \bar{\epsilon x}_j)\} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} = \quad (5.1.1-4)$$

$$\begin{bmatrix} \dots & \dots & \dots \\ \dots & C_X(i,j) & \dots \\ \dots & \dots & \dots \end{bmatrix} \equiv \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2n} \\ \dots & \dots & \dots & \dots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_n^2 \end{bmatrix},$$

an  $n \times n$  matrix.

The  $\sigma_i^2 = E\{(\epsilon x_i - \bar{\epsilon x}_i)^2\}$  are variances about the (component) mean, the  $\sigma_i$  are standard deviations about the mean, and the  $\sigma_{ij} = E\{(\epsilon x_i - \bar{\epsilon x}_i)(\epsilon x_j - \bar{\epsilon x}_j)\}$  are covariances about the mean. Note that since  $(\epsilon x_i - \bar{\epsilon x}_i)(\epsilon x_j - \bar{\epsilon x}_j) = (\epsilon x_j - \bar{\epsilon x}_j)(\epsilon x_i - \bar{\epsilon x}_i)$  by the properties of real-numbers, the error covariance matrix is symmetric by definition.

Further note that:

$$\text{if } \bar{\epsilon X} = 0, \quad C_X = E\{(\epsilon X)(\epsilon X)^T\}, \text{ the latter also termed the “second moment”}; \quad (5.1.1-5)$$

If  $\bar{\epsilon X} \neq 0$ ,  $C_X = E\{(\epsilon X)(\epsilon X)^T\} - (\bar{\epsilon X})(\bar{\epsilon X})^T$ , the latter termed the “square” of the “first moment”. (5.1.1-6)

The (cumulative) probability distribution function  $cdf_X$  of the random error vector  $\epsilon X$  is (5.1.1-7) another term for the joint probability distribution of the random variables  $\epsilon x_1, \dots, \epsilon x_n$ , a scalar function of the n-dimensional (component) value  $\epsilon X$ , and “loosely” defined as the probability that an arbitrary value  $\epsilon X'$  is contained within the n-dimensional interval  $[(-\infty, \epsilon x_1'), \dots, (-\infty, \epsilon x_n')]$ . (See Section 5.1.1.1 for further discussion.)

A probability density function  $pdf_X$  of the random error vector  $\epsilon X$  is “loosely” defined as the  $n$ -th order partial derivative of  $cdf_X$ , a scalar function of the n-dimensional value  $\epsilon X$ :

$$pdf_X(\epsilon X) = \frac{\partial^n cdf(\epsilon X)}{\partial \epsilon x_1 \dots \partial \epsilon x_n} \quad (5.1.1-8)$$

In Equations (5.1.1-2) – (5.1.1-6),  $E\{\}$  corresponds to the expected value taken over an (arbitrary) probability density function  $pdf_X$ . For an arbitrary scalar function  $g(X)$ , its expected value is defined as:

$$E\{g(\epsilon X)\} \equiv \iint \dots \int_{-\infty}^{+\infty} g(\epsilon X) pdf_X d\epsilon x_1 \dots d\epsilon x_n, \quad (5.1.1-9)$$

where the integration is taken over  $(-\infty, +\infty)$  for each of the  $n$  error components, i.e.,  $n$  joint integrals taken over the entire n-dimensional (real-valued) space  $R^n$ .

Thus, the mean-value is simply the expected value of ( $n$  components of) the random vector  $\epsilon X$  and the error covariance matrix about the mean-value is the expected value of ( $n^2$  components of)  $(\epsilon X - \bar{\epsilon X})(\epsilon X - \bar{\epsilon X})^T$ , where  $\bar{\epsilon X}$  is considered a deterministic (pre-computed) statistic, i.e.,  $C_X = E\{(\epsilon X - \bar{\epsilon X})(\epsilon X - \bar{\epsilon X})^T\}$ .

The variance of a component of error  $\epsilon x_i$  in  $\epsilon X$  corresponds to matrix element  $C_X(i, i) \equiv \sigma_i^2$ . The covariance (not the “covariance matrix”) between two components of error  $\epsilon x_i$  and  $\epsilon x_j$ , in the same  $\epsilon X$ , corresponds to matrix element  $C_X(i, j) \equiv \sigma_{ij}$ . It further defines the correlation (coefficient)  $\rho_{ij}$  between these two components of error as follows:

$$\rho_{ij} \equiv \sigma_{ij} / (\sigma_i \sigma_j); \text{ thus, } \sigma_{ij} = \rho_{ij} \sigma_i \sigma_j \quad (5.1.1-10)$$

Note that  $\rho_{ij}$  is unit-less, and that  $|\rho_{ij}| < 1, i \neq j$ , as will be demonstrated later.

### 5.1.1.1 Underlying probabilistic foundations

The definitions presented in Section 5.1.1 are at an “engineering overview” level, and underlying probabilistic foundations are not needed for further applications in this or related documents. However, for the sake of completeness, the following (jointly labeled “Equation” (5.1.1.1-1) for convenience) are briefly mentioned, with further details found in reference [24], as well as more formal definitions for the probability distribution function and the probability density function:

- There is an underlying Probability Space consisting of the triple (5.1.1.1-1)  
{Experiment Space of all possible outcomes, Subsets of all possible collections of outcomes or Events (Borel field), the Probability Measure of an arbitrary event}.
- A random (error) vector consists of a vector of random variables ( $\epsilon x_i$ ), each a mapping from every outcome in the Experiment Space to a number. We assume that random variables are of the continuous type in this document, but these can be extended to include random variables of the discrete type.
- The above number is assumed to be real-valued in this and related documents, but definitions can be augmented in a straightforward manner to include random variables as mapping to complex numbers. Augmentation essentially consists of defining a probability density function as a joint density between the real and imaginary parts making up the complex numbers, and defining vector and matrix transpose as the transpose of the complex conjugate of the vector or matrix.

### 5.1.2 Example: scalar Gaussian pdf and related statistics

The above predictive statistics are further defined/illustrated for  $n = 1$  and the common (scalar) Gaussian or Normal distribution with probability density function  $pdf_x$  as defined below (the explicit error notation “ $\epsilon$ ” was dropped for convenience in the following, a practice sometimes used throughout this document; furthermore,  $x \equiv x_1$ ):

$$\text{Given the Gaussian probability density function, } pdf_x \equiv \frac{1}{\sigma_x \sqrt{2\pi}} e^{-1/2((x-\bar{x})/\sigma_x)^2}: \quad (5.1.2-1)$$

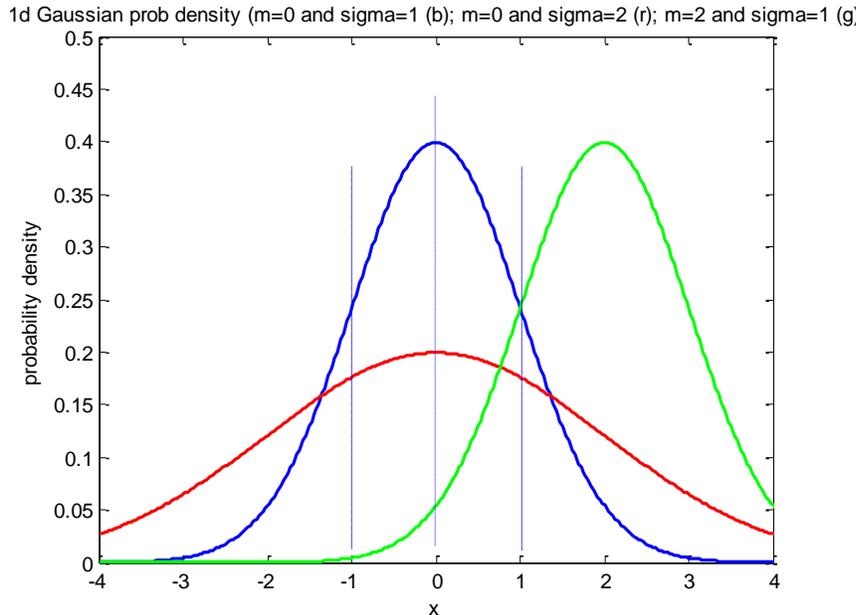
$$E\{x\} = \int_{-\infty}^{+\infty} x pdf_x dx = \frac{1}{\sigma_x \sqrt{2\pi}} \int_{-\infty}^{\infty} x e^{-1/2((x-\bar{x})/\sigma_x)^2} dx = \bar{x}, \text{ the mean-value;} \quad (5.1.2-2)$$

$$E\{(x - \bar{x})^2\} = \int_{-\infty}^{+\infty} (x - \bar{x})^2 pdf_x dx = \frac{1}{\sigma_x \sqrt{2\pi}} \int_{-\infty}^{\infty} (x - \bar{x})^2 e^{-1/2((x-\bar{x})/\sigma_x)^2} dx = \sigma_x^2, \quad (5.1.2-3)$$

the variance about the mean (the mean and variance are directly embedded in the Gaussian pdf itself);

$$p = \int_{region R}^* pdf_x dx = \frac{1}{\sigma_x \sqrt{2\pi}} \int_{region R}^* e^{-1/2((x-\bar{x})/\sigma_x)^2} dx, \quad (5.1.2-4)$$

the probability contained within a *region R* (line), where the notation “ $\int_{region R}^* \dots dx$ ” indicates integration over the specified region.



**Figure 5.1.2-1:** Examples of Gaussian probability density functions for a scalar error  $x$

In Figure 5.1.2-1 above, the blue curve corresponds to a pdf with mean=0 and standard deviation or “sigma” of 1 about the mean; the green curve corresponds to an identical pdf except that the mean=2; the red curve corresponds to a mean=0 but a standard deviation of 2. Note that the standard deviation is a measure of the dispersion about the mean.

And, of course, by definition of a Probability Space, the area under any of the  $pdf_x$  curves above, taken over the interval  $(-\infty, +\infty)$ , equals 1. (When taken over the interval  $(-\infty, x)$  this is a function of  $x$  and termed the cumulative probability distribution function  $cdf_x$ .)

Note that an arbitrary Gaussian or Normal distribution with mean  $m$  and standard deviation  $\sigma$  is sometimes designated  $N(m, \sigma)$ .

### 5.1.3 Example: Multi-variate Gaussian pdf and related statistics

These statistics are now generalized to an arbitrary random (error) vector and multi-variate or multi-component Gaussian or Normal distribution (and again dropping the explicit error notation “ $\epsilon$ ” for convenience):

Given a pdf of a multi-variate Normal distribution of a random vector  $X = [x_1 \ \dots \ x_n]^T$ ,

$$pdf_X \equiv \frac{1}{(2\pi)^{n/2} \det(C_X)^{1/2}} e^{-1/2((X-\bar{X})^T C_X^{-1} (X-\bar{X}))} ; \quad (5.1.3-1)$$

$$E\{X\} \equiv \iint \dots \int_{-\infty}^{+\infty} X pdf_X dx_1 \dots dx_n = \quad (5.1.3-2)$$

$$\frac{1}{(2\pi)^{n/2} \det(C_X)^{1/2}} \iint \dots \int_{-\infty}^{+\infty} X e^{-1/2((X-\bar{X})^T C_X^{-1} (X-\bar{X}))} dx_1 \dots dx_n = \bar{X},$$

which is the mean-value of  $X$ .

(Note: the above is the notational equivalent of the  $n \times 1$  column vector  $\left[ \iint \dots \int_{-\infty}^{+\infty} x_i pdf_X dx_1 \dots dx_n \right]$ .)

$$E\{(X - \bar{X})(X - \bar{X})^T\} = \iint \dots \int_{-\infty}^{+\infty} (X - \bar{X})(X - \bar{X})^T pdf_X dx_1 \dots dx_n = \quad (5.1.3-3)$$

$$\frac{1}{(2\pi)^{n/2} \det(C_X)^{1/2}} \iint \dots \int_{-\infty}^{+\infty} (X - \bar{X})(X - \bar{X})^T e^{-1/2((X-\bar{X})^T C_X^{-1} (X-\bar{X}))} dx_1 \dots dx_n = C_X,$$

which is the covariance of  $X$  about the mean-value of  $X$ .

(Note: the above is the notational equivalent of the  $n \times n$  matrix

$$\begin{bmatrix} \dots & \dots & \dots \\ \dots & \iint \dots \int_{-\infty}^{+\infty} (x_i - \bar{x}_i)(x_j - \bar{x}_j) pdf_X dx_1 \dots dx_n & \dots \\ \dots & \dots & \dots \end{bmatrix}.)$$

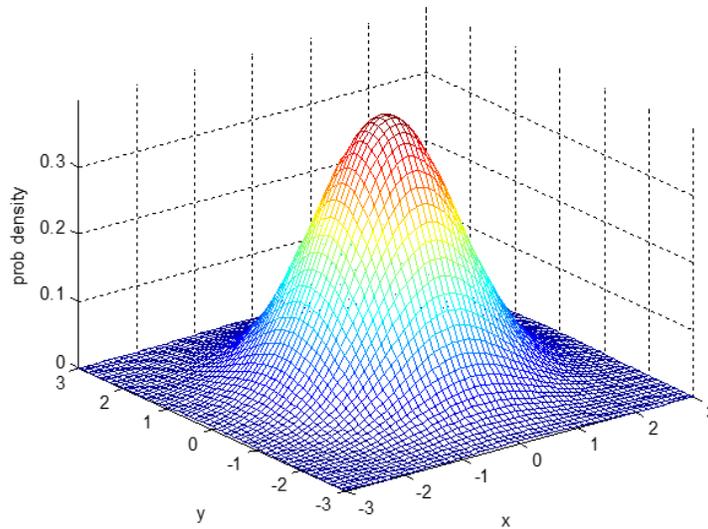
$$p = \frac{1}{(2\pi)^{n/2} \det(C_X)^{1/2}} \iint \dots \int_{region R}^* e^{-1/2((X-\bar{X})^T C_X^{-1} (X-\bar{X}))} dx_1 \dots dx_n, \quad (5.1.3-4)$$

which the probability contained within the  $n$ -dimensional region  $R$ .

(In the above equations, the notation “ $\iint \dots \int_{-\infty}^{+\infty} \dots dx_1 \dots dx_n$ ” indicates integration from  $-\infty$  to  $\infty$  over each of the variables  $x_i$ ,  $i = 1, \dots, n$ ; the notation “ $\iint \dots \int_{region R}^* \dots dx_1 \dots dx_n$ ” indicates integration within the specified  $n$ -dimensional region  $R$  over each of the variables  $x_i$ ,  $i = 1, \dots, n$ .)

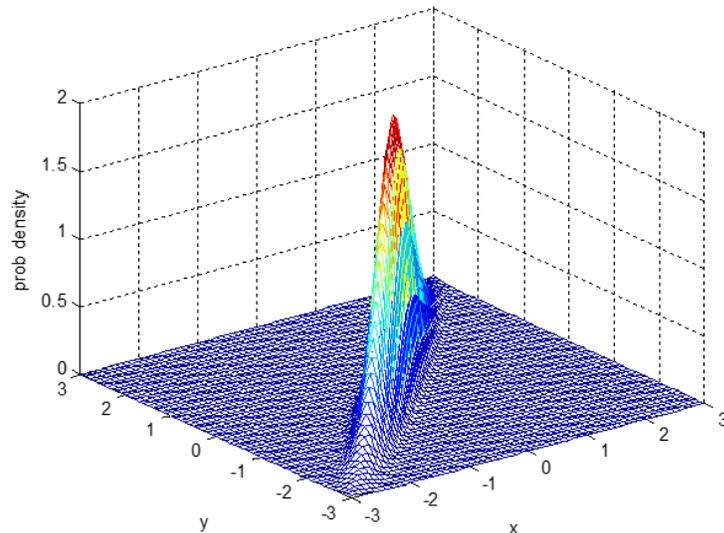
The following Figures 5.1.3-1 and 5.1.3-2 correspond to a multi-variate Gaussian probability density function for two (error) components ( $n = 2$ ). Both pdf's have a vector mean-value of zero. The first pdf has an  $2 \times 2$  error covariance matrix with diagonal elements (variance) equal to 1 and zero covariance between the two components (correlation=0), the second has the same error covariance matrix but has a non-zero covariance corresponding to high positive correlation (correlation coefficient  $\rho_{12} = 0.98$ ) between the two components.

2d Gaussian prob density (means = 0, sigmas = 1, correlation between x and y = 0)



**Figure 5.1.3-1:** Two-dimensional Gaussian pdf with zero correlation between components

2d Gaussian prob density (means = 0, sigmas = 1, correlation between x and y = 0.98)



**Figure 5.1.3-2:** Two-dimensional Gaussian pdf with high positive correlation between components

**5.1.4 Conditional expectations and correlation**

We can also define conditional distributions or pdfs as well. In general, for two components  $x$  and  $y$ , the conditional pdf of  $x$  given  $y$  is (and dropping the explicit error notation “ $\epsilon$ ” for convenience):

$$pdf(x|y) = pdf_{x,y}/pdf_y$$

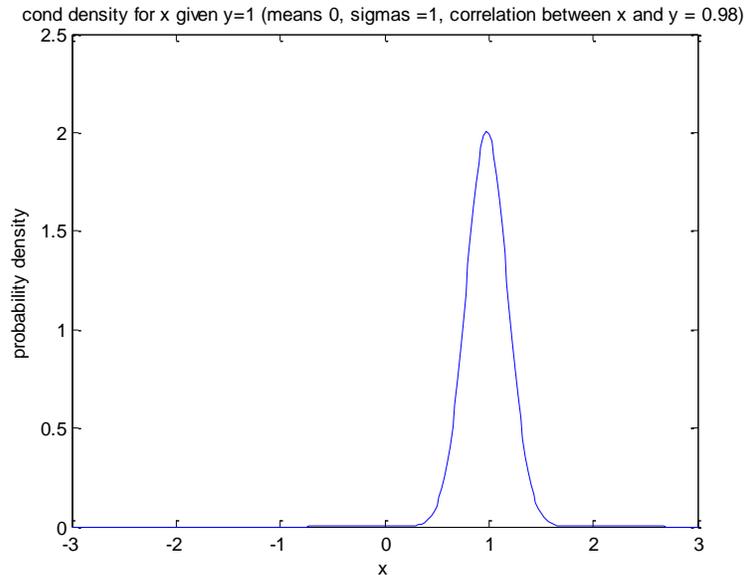
(5.1.4-1)

Assuming a Gaussian joint distribution (aka a bivariate normal distribution) with a mean-value of zero and with the value of  $y$  given, the conditional pdf equals:

$$pdf(x|y) = \frac{1}{\sigma_x \sqrt{2\pi(1-\rho_{x,y}^2)}} e^{-\frac{(x - \frac{\rho_{x,y}\sigma_x y}{\sigma_y})^2}{2\sigma_x^2(1-\rho_{x,y}^2)}}, \text{ where} \tag{5.1.4-2}$$

the above has a mean-value of  $\rho_{x,y}\sigma_x y/\sigma_y$  and a variance about the mean of  $\sigma_x^2(1 - \rho_{x,y}^2)$ .

Thus, given that the specific (joint)  $pdf_X = pdf_{x,y}$  corresponding to Figure 5.1.3-2, and that  $y$  has a value of 1, the conditional pdf corresponds to Figure 5.1.4-1 below.



**Figure 5.1.4-1: Conditional pdf of x given y equals 1**

The conditional pdf has a mean-value of  $\frac{\rho_{x,y}\sigma_x y}{\sigma_y} = 0.98$  and variance about the mean-value of  $\sigma_x^2(1 - \rho_{x,y}^2) = 0.0396$  or “sigma”  $\cong 0.2$ . That is, the expected value of  $x$  is very close to the given value of  $y$  with very little uncertainty (dispersion). Obviously, the given or “*a priori*” correlation  $\rho_{x,y} = 0.98$  has a tremendous influence on the results.

Another way to look at correlation: even though we do not know the value of either  $x$  or  $y$  now, if they are highly correlated, and given the value of  $y$  (or  $x$ ) in the future, we can then accurately predict that the value of  $x$  is a positive scalar multiple of the given value  $y$  (negative scalar multiple, if the correlation is highly negative). The scalar multiple approaches 1 (or -1) if the correlation is high and the variance of the two components are approximately the same value. This is one illustration of the importance of correlation, and the need to reliably “capture it” within the appropriate error covariance matrix.

### 5.1.5 Coordinate Systems

Note that the underlying (assumed) Cartesian coordinate system for representation of the random error vector (Equation (5.1.1-2)) may differ from that for the state vector (Equation (5.1.1-1)) itself for practical modeling of predictive errors. This is perfectly valid as long as there is an associated deterministic transformation between the two systems. For example, if the state vector contains 3d location, it may be relative to the WGS84 (earth-centered, earth-fixed) system, whereas the coordinate system for its errors may be relative to a local tangent plane (ENU) system where component errors may be modeled as uncorrelated when appropriate. Note that if a state vector  $X$  in WGS84 corresponds to the state vector  $X'$  in ENU, we have:

$$X' = \Omega X + X_0, \text{ where} \tag{5.1.5-1}$$

$\Omega$  and  $X_0$  are a deterministic  $3 \times 3$  rotation matrix and  $3 \times 1$  vector offset, respectively.

Therefore, the random error vector expressed in ENU is simply the following function of the random error vector expressed in WGS84, which can be accounted for in “rigorous error propagation” (see Equation (5.2.2-1) and Section 5.6):

$$\epsilon X' = \Omega \epsilon X \tag{5.1.5-2}$$

## 5.2 Error Covariance Matrices: Properties and Rigorous Descriptors

This section of the document makes no assumptions regarding the underlying probability distribution of errors, i.e., is applicable whether a Gaussian distribution or not. It details the error covariance matrix and its various properties and descriptors.

Assume a single  $n \times 1$  state vector, and its corresponding (previously defined)  $n \times 1$  random error vector, its mean-value, and  $n \times n$  valid error covariance matrix about the mean:

$$\epsilon X = [\epsilon X(1) \quad \dots \quad \epsilon X(n)]^T \equiv [\epsilon x_1 \quad \dots \quad \epsilon x_n]^T; \tag{5.2-1}$$

$$\bar{\epsilon X} = [\bar{\epsilon X}(1) \quad \dots \quad \bar{\epsilon X}(n)]^T \equiv [\bar{\epsilon x}_1 \quad \dots \quad \bar{\epsilon x}_n]^T; \tag{5.2-2}$$

$$C_X = \begin{bmatrix} \dots & \dots & \dots & \dots \\ \dots & E\{(\epsilon x_i - \bar{\epsilon x}_i)(\epsilon x_j - \bar{\epsilon x}_j)\} & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \equiv \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2n} \\ \dots & \dots & \dots & \dots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_n^2 \end{bmatrix}. \tag{5.2-3}$$

A valid  $n \times n$  error covariance matrix is a wonderful object. It is completely characterized/defined, in conjunction with Equation (5.2-3), as follows:

- Any symmetric,  $n \times n$  positive definite matrix is a valid  $n \times n$  error covariance matrix (5.2-4)
- A valid  $n \times n$  error covariance matrix is a symmetric,  $n \times n$  positive definite matrix.

From this top-level characterization/definition, we have the following properties for a valid  $n \times n$  error covariance matrix  $C_X$  (the following properties are jointly labeled “Equation” (5.2-5) for convenience):

- It is invertible ( $C_X^{-1}$  exists) (5.2-5)
- It has strictly positive diagonal entries (error variances)
- It has a strictly positive determinant ( $\det(C_X) > 0$ )
- The absolute value of the correlation coefficient for an arbitrary pair of distinct error components is less than 1
- It has  $n$  (not necessarily distinct) strictly positive ( $>0$ ) eigenvalues and corresponding eigenvectors
- Its trace and determinant are the sum and product of its eigenvalues, respectively
- An orthonormal  $n \times n$  (rotation) matrix  $\Phi$ , consisting of the unit eigenvectors as matrix rows, maps the original coordinate system to the eigenvector-aligned coordinate system
- The original error covariance matrix  $C_X$  can be represented as an  $n \times n$  diagonal error covariance matrix  $D$  with the eigenvalues as diagonals in the eigenvector-aligned Cartesian coordinate system, i.e.,  $D = \Phi C_X \Phi^T$

In this document, unless specifically stated otherwise, an error covariance matrix  $C_X$  is assumed valid.

The actual formal definition of a valid error covariance matrix (as opposed to its properties or characteristics) follows from the definitions and interrelationships between a positive semi-definite matrix, a positive definite matrix, and the expected value operator as detailed in the following section.

### 5.2.1 Formal derivation/definition of a valid error covariance matrix

From linear algebra, an arbitrary  $n \times n$  matrix  $Q$  is positive semi-definite by definition if the following scalar is always positive (non-negative):

$$Z^T Q Z = \sum_{i,j} z_i z_j Q_{ij} \geq 0, \text{ for all } n \times 1 \text{ column vectors } Z. \quad (5.2.1-1)$$

An  $n \times n$  matrix  $Q$  is positive definite by definition if the following scalar is always strictly positive:

$$Z^T Q Z = \sum_{i,j} z_i z_j Q_{ij} > 0, \quad (5.2.1-2)$$

for all  $n \times 1$  column vectors  $Z$  not identically equal to zero.

An  $n \times n$  error covariance matrix  $C_X$  is guaranteed positive semi-definite or “pseudo-valid” by the linear properties of the expectation operator and the fact that the absolute value of an arbitrary quantity is always  $\geq 0$ :

$$E\{z_1(\epsilon x_1 - \bar{\epsilon x}_1) + \dots + z_n(\epsilon x_n - \bar{\epsilon x}_n)\}^2 = \quad (5.2.1-3)$$

$$\sum_{i,j} z_i z_j E\{(\epsilon x_i - \bar{\epsilon x}_i)(\epsilon x_j - \bar{\epsilon x}_j)\} = Z^T C_X Z \geq 0.$$

However, we further add the stipulation that all (centered) random variables  $(\epsilon x_i - \bar{\epsilon x}_i)$  are to be linearly independent, i.e., by the definition of linearly independent [24]:

$$E\{z_1(\epsilon x_1 - \bar{\epsilon x}_1) + \dots + z_n(\epsilon x_n - \bar{\epsilon x}_n)\}^2 > 0, \text{ for } z_i, i = 1, \dots, n, \text{ not all zero;} \quad (5.2.1-4)$$

$$\text{and thus, } \sum_{i,j} z_i z_j E\{(\epsilon x_i - \bar{\epsilon x}_i)(\epsilon x_j - \bar{\epsilon x}_j)\} = Z^T C_X Z > 0, \text{ i.e., } C_X \text{ is positive definite.} \quad (5.2.1-5)$$

(Note: linear independence of (centered) random variables is also equivalent to linear independence of random variables, since one is just the other plus a deterministic offset.)

With this added (required) property of linear independence, the error covariance matrix  $C_X$  is valid, i.e., positive definite and invertible.

Further note that linear independence between random variables does not imply that they are not correlated. However, it does imply that they are not “totally” correlated. This is defined/derived as follows. From Equation (5.2.1-4) and for  $z_i=1$  and  $z_j = \pm 1$ :

$$E\{(\epsilon x_i - \bar{\epsilon x}_i)^2 \pm 2(\epsilon x_i - \bar{\epsilon x}_i)(\epsilon x_j - \bar{\epsilon x}_j) + (\epsilon x_j - \bar{\epsilon x}_j)^2\} = \quad (5.2.1.6)$$

$$\sigma_i^2 \pm 2\sigma_{ij} + \sigma_j^2 = \sigma_i^2 \pm 2\rho_{ij}\sigma_i\sigma_j + \sigma_j^2 > 0,$$

which implies that  $\rho_{ij} < 1$  if  $z_j = -1$ , and  $-1 < \rho_{ij}$  if  $z_j = 1$ , or that:

$$|\rho_{ij}| < 1, \quad (5.2.1.7)$$

for an arbitrary pair of component errors. This is a necessary condition for a valid error covariance matrix, but not a sufficient condition as will be discussed later.

Linear independence, and hence a positive definite error covariance matrix, is a common assumption/definition. It guarantees the existence of  $C_X^{-1}$ , which is required for many related definitions and practical applications, such as: (1) the definition of the multi-variate Gaussian pdf, (2) computation of the error ellipsoid (Section 5.3), and computation of the scalar accuracy metrics LE, CE, and SE (Section 5.4).

### 5.2.2 Additional properties of related matrices

The properties listed in Equation (5.2-5) for valid error covariance matrices are straightforward, easily stated, and very useful. Another property for valid error covariance matrices that is not quite as “straightforward”, but certainly useful, particularly regarding the propagation of error covariance matrices (Section 5.6), is as follows:

- Any “new”  $mx1$  random error vector defined as  $\epsilon X' = \Omega \epsilon X$ , and thus  $\overline{\epsilon X'} = \Omega \overline{\epsilon X}$ , (5.2.2-1) where the  $mxn$  mapping matrix  $\Omega$  is full rank, has a valid  $mxm$  error covariance matrix equal to  $C_{X'} \equiv E\{(\epsilon X' - \overline{\epsilon X'})(\epsilon X' - \overline{\epsilon X'})^T\} = \Omega C_X \Omega^T$ .
  - If the mapping matrix is not full rank (e.g.,  $m > n$ ), the resultant “pseudo-valid” error covariance matrix is still positive semi-definite, i.e., positive eigenvalues ( $\geq 0$ ) but not invertible.

Additional properties of valid error covariance and related matrices are:

- If the  $nxn$  matrix  $A$  is a valid error covariance matrix, its  $nxn$  inverse  $A^{-1}$  (5.2.2-2) is also a valid error covariance matrix since it is symmetric and positive definite as well. In addition, its eigenvalues are the reciprocal of the eigenvalues of  $A$ .
- If the  $nxn$  matrix  $A$  is a valid error covariance matrix and the  $nxn$  matrix  $B$  (5.2.2-3) is a pseudo-valid (or valid) error covariance matrix, then the matrix  $A + B$  is a valid error covariance matrix.
- If the  $nxn$  matrix  $A$  is a valid error covariance matrix and  $Y$  an arbitrary  $nx1$  (5.2.2-4) vector, the  $nxn$  matrix  $A' = A + YY^T$  is a valid  $nxn$  error covariance matrix, based on Equation (5.2.2-3) and the following:
  - $YY^T$  is symmetric and positive semi-definite (pseudo-valid error covariance matrix), since it is obviously symmetric and for an arbitrary  $nx1$  vector  $Z$ ,  $Z^T Y$  is a scalar  $s$ , and  $Z^T YY^T Z$  equals  $s^2 \geq 0$ ; hence,  $YY^T$  is positive semi-definite by definition.
- If the  $nxn$  matrix  $A$  is a valid error covariance matrix, the  $nxn$  matrix  $B = kA$  (5.2.2-5) is also a valid error covariance matrix, where the scalar  $k > 0$  and the multiplication  $kA$  corresponds to multiplying each element of  $A$  by the scalar  $k$ .

Further note, that virtually all commercial pseudo-code (e.g., MATLAB) have straight-forward functions to determine eigenvalues and corresponding eigenvectors, as well as other functions from linear algebra and probability /statistics. The linear algebra related properties/proofs of Sections 5.2, 5.2.1, and 5.2.2 are found or can be readily derived via reference [22].

### 5.2.3 Possible issues with the mean-value

In Section 5.2, we defined the  $nx1$  mean-value  $\overline{\epsilon X}$  and the  $nxn$  error covariance matrix  $C_X$  about the mean-value. Since, in this document, these are predictive statistics, the mean-value is typically assumed zero and need not be accounted for explicitly. Also, if it were non-zero, in many instances the corresponding vector  $X$  could be corrected “ahead of time”, such as for a satellite metadata error (“bias”) corrected for during preprocessing at a ground (collection) station; and thus, the mean-value  $\overline{\epsilon X}$  subsequently zero thereafter.

In summary, typically for predictive statistics within the NSG, either: (1)  $\overline{\epsilon X}$  with a value of zero and  $C_X$  are both accounted for/disseminated explicitly, or (2) only  $C_X$  is accounted for/disseminate explicitly, with  $\overline{\epsilon X}$  assumed zero.

However, if this is not the case, the equations in this document can still be utilized by various modules/applications as they account (sometimes as an option) for a non-zero mean-value. If this is not possible due to design limitations (i.e., a non-zero mean-value of random error cannot be disseminated/implemented by various modules/applications), we can mitigate this problem as follows:

$$\{ \overline{\epsilon X} \neq 0 \text{ and } C_X \} \rightarrow \{ \overline{\epsilon X} \text{ assumed zero and the original } C_X \text{ modified to } (C_X + \overline{\epsilon X} \overline{\epsilon X}^T) \} \quad (5.2.3-1)$$

The original (left side of the above equation) is statistically correct, and the right side an approximation to be used only if necessary, i.e., this technique is a “last resort” to account for a non-trivial mean-value when confronted with design limitations. We know, by Equations (5.2.2-2) and (5.2.2-3), that the approximation corresponds to a valid error covariance matrix (symmetric and positive definite). The form of the approximation follows from Equation (5.1.1-6).

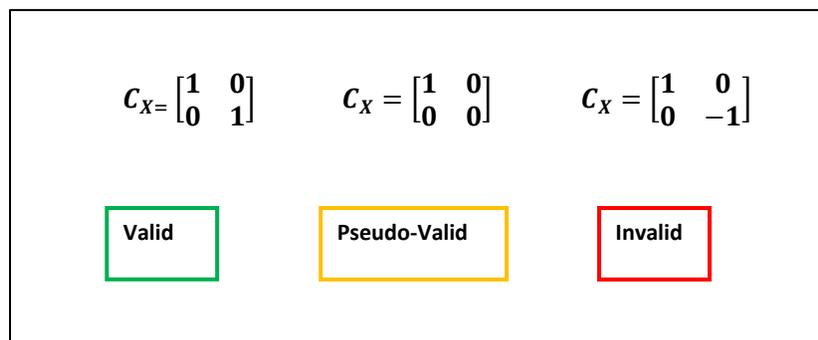
In the upcoming Section 5.3 on error ellipsoids, Figure 5.3.1-5 presents an example of Equation (5.2.3-1) in terms of corresponding error ellipsoids.

#### 5.2.4 Assurance of valid and realistic error covariance matrices required for practical applications

A candidate error covariance matrix is either valid (positive definite), pseudo-valid (positive semi-definite), or invalid (not positive semi-definite). Of course, by our definitions, a valid error covariance matrix is also positive semi-definite, but a pseudo-valid error covariance matrix is not positive definite. And as such, it is not invertible, a property required for many applications, such as weighting measurements in an estimator.

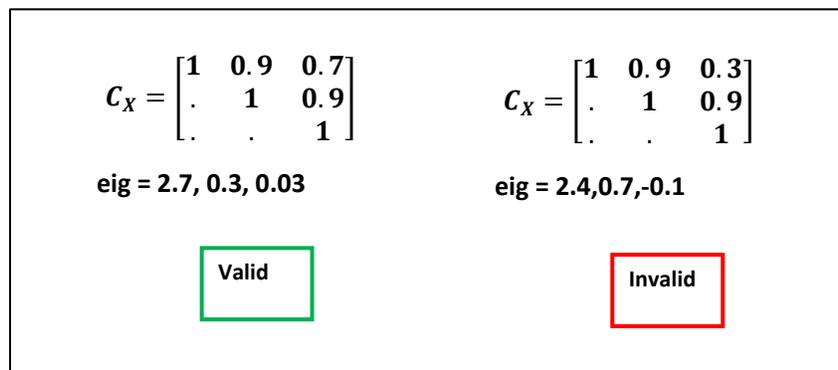
A candidate error covariance matrix is invalid if it has one or more negative eigenvalues. As such, there is some linear combination of its underlying error components that has a negative variance – physically impossible, and a “time-bomb” for any application attempting to use it.

Thus, it certainly made sense to define a valid error covariance matrix as a symmetric and positive definite matrix. And hopefully, it is easy to identify a “valid” versus “pseudo-valid” versus “invalid” error covariance matrix. Figure 5.2.4-1 presents the simplest possible case for all three categories.



**Figure 5.2.4-1:** Valid, pseudo-valid, and invalid error covariance matrices.

Of course, with more realistic applications, it is not so easy to identify validity without computing corresponding eigenvalues. For example, the 3x3 error covariance matrix on the left side of Figure 5.2.4-2 is valid, while the one on the right side is invalid. The latter’s correlation coefficients are statistically inconsistent with each other. The first component of error is highly correlated (statistically similar; correlation coefficient equal to 0.9) with the second, and the second component of error is highly correlated with the third. This would imply that the first and third components must be reasonably correlated as well ( $> 0.62$ ), but they are not. Hence, the error covariance matrix is invalid, i.e., it has a negative eigenvalue. This is true even though the necessary condition that the absolute values of all correlations were less than 1 was met. (Only for a 2x2 error covariance matrix is this condition sufficient, assuming of course, diagonal elements greater than zero.)



**Figure 5.2.4-2:** Valid and invalid error covariance matrices

The above example (Figure 5.2.4-2) was also more realistic than the first example (Figure 5.2.4-1) in that most error covariance matrices of interest are larger than 2x2 matrices (3x3 to 10000x10000 not unrealistic), and usually non-diagonal, particular if an output from an estimator (e.g., Kalman Filter or WLS). The off-diagonal components (correlations) are extremely important for the characterization of predicted accuracies as well as optimal performance of subsequent processing – they are not to be ignored or removed. Methods for the generation, dissemination, and representation of error covariance matrices presented in this document insure valid and full error covariance matrices.

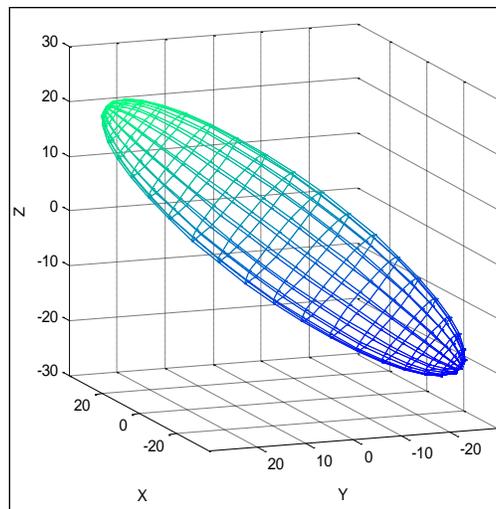
Finally, although we consider a valid error covariance matrix as a necessary condition, we also want a “realistic” error covariance matrix - one that reasonably approximates the “true” and generally unknown error covariance matrix that corresponds to the true error. This, of course, is a challenge. Throughout this document we present various methods to reasonably model errors and to assemble corresponding error covariance matrices. Of course, modeling is “user-specific” and usually iterative, in that corresponding predictive statistics need verification with sample statistics/“ground truth”, and the predictive error model subsequently “tuned”. QC checks in various estimators that rely on predictive statistics can also be employed. These topics are covered more deeply in TGD 1 (Overview and Methodologies), TGD 2d (Estimators and Quality Control), and TGD 2c (Specification and Validation).

### 5.3 Error Ellipsoids

This section of the document assumes that the underlying probability distribution of errors is Gaussian in order to assign probabilities to the error ellipsoids.

An error ellipsoid is a graphical representation of the error covariance  $C_x$  and an intuitive representation of predicted accuracy. It displays, among other things, the directions of greatest and least expected error (magnitude). It is typically defined for three or fewer components of error for visualization.

The error ellipsoid presented in Figure 5.3-1 corresponds to geographic 3d location error as represented in a local tangent plane (ENU) Cartesian coordinate system. It was computed as a 90% (0.9p) error ellipsoid, which means that there is a 90% probability that the location (solution) error is within the ellipsoid. Alternatively, if the 90% error ellipsoid is centered at the target solution  $X$  instead of zero, there is a 90% probability that the true target location is within the ellipsoid. When centered at the target solution, the error ellipsoid is typically called a confidence ellipsoid. We are 90% confident that the true target location is within the 90% confidence ellipsoid.



**Figure 5.3-1:** The 90% (0.9p) probability error ellipsoid corresponding and equivalent to  $C_x$

The specific underlying error covariance matrix in the above example is equal to:

$$C_X = \begin{bmatrix} \sigma_x^2 & \rho_{xy}\sigma_x\sigma_y & \rho_{xz}\sigma_x\sigma_z \\ \cdot & \sigma_y^2 & \rho_{yz}\sigma_y\sigma_z \\ \cdot & \cdot & \sigma_z^2 \end{bmatrix} = \begin{bmatrix} 10^2 & 0.75 \cdot 10 \cdot 12 & 0.95 \cdot 10 \cdot 9 \\ \cdot & 12^2 & 0.8 \cdot 12 \cdot 9 \\ \cdot & \cdot & 9^2 \end{bmatrix}. \quad (5.3-1)$$

(Note: the above is an error covariance matrix relative to a single  $3 \times 1$  state vector error  $\epsilon X = [\epsilon x_1 \ \epsilon x_2 \ \epsilon x_3]^T \equiv [\epsilon x \ \epsilon y \ \epsilon z]^T$ .)

The general equation for an error ellipsoid is:

$$\epsilon X^T C_X^{-1} \epsilon X = d^2, \tag{5.3-2}$$

where  $\epsilon X$  is  $n \times 1$ ,  $C_x$  is  $n \times n$ ,  $n$  an integer and typically  $1 \leq n \leq 3$ ,  $d$  a positive scalar, and the mean-value  $\bar{\epsilon X}$  or the error ellipsoid origin is assumed 0.

Figure 5.3-2 presents a corresponding summary of the general equation for different values of  $n$ .

The general equation for an error ellipsoid is given by: $\epsilon X^T C_X^{-1} \epsilon X = d^2$		
<u>For dim <math>n = 1</math>:</u>	<u>For dim <math>n = 2</math>:</u>	<u>For dim <math>n = 3</math>:</u>
$\epsilon X = \epsilon x$	$\epsilon X = [\epsilon x \quad \epsilon y]^T$	$\epsilon X = [\epsilon x \quad \epsilon y \quad \epsilon z]^T$
$C_X = [E\{\epsilon x^2\}]$	$C_X = \begin{bmatrix} E\{\epsilon x^2\} & E\{\epsilon x \epsilon y\} \\ . & E\{\epsilon y^2\} \end{bmatrix}$	$C_X = \begin{bmatrix} E\{\epsilon x^2\} & E\{\epsilon x \epsilon y\} & E\{\epsilon x \epsilon z\} \\ . & E\{\epsilon y^2\} & E\{\epsilon y \epsilon z\} \\ . & . & E\{\epsilon z^2\} \end{bmatrix}$

**Figure 5.3-2:** General Equation for the Error Ellipsoid

This general equation is both simple yet elegant. However, the key to its use and interpretation is the value of the “distance” or normalized radius  $d$  (unitless)

Assuming a Gaussian (multi-variate) probability distribution of errors (i.e., the normal or “bell-shaped” distribution of errors), various values of the normalized radius  $d$  correspond to various probabilities that the  $n$ -dimensional solution error lies within the interior of the error ellipsoid. Tables 5.3-1 and 5.3-2 present the correspondence between probability  $p$ , normalized radius or distance  $d$ , and dimension  $n$ . Note: when  $n = 2$  and  $n = 1$ , the ellipsoid “collapses” to an ellipse and line, respectively.

**Table 5.3-1** Distance  $d$  versus probability  $p$  and dimension  $n$

probability $p$	$n=1$	$n=2$	$n=3$
0.5	0.6745	1.1774	1.5382
0.9	1.6449	2.1460	2.5003
0.95	1.9600	2.4477	2.7955
0.99	2.5758	3.0349	3.3682
0.999	3.2905	3.7169	4.0336

**Table 5.3-2** Probability  $p$  versus distance  $d$  and dimension  $n$

distance $d$	$n=1$	$n=2$	$n=3$
1	0.6827	0.3935	0.1987
2	0.9545	0.8647	0.7385
3	0.9973	0.9889	0.9707

Therefore, for example, the 90% error ellipsoid ( $n=3$ ) presented in Figure 5.3-1 corresponds to the equation  $\epsilon X^T C_X^{-1} \epsilon X = (2.5003)^2$ .

The entries for the above tables were derived via the equations detailed in Section 5.3.2.

### 5.3.1 Error Ellipsoid Examples

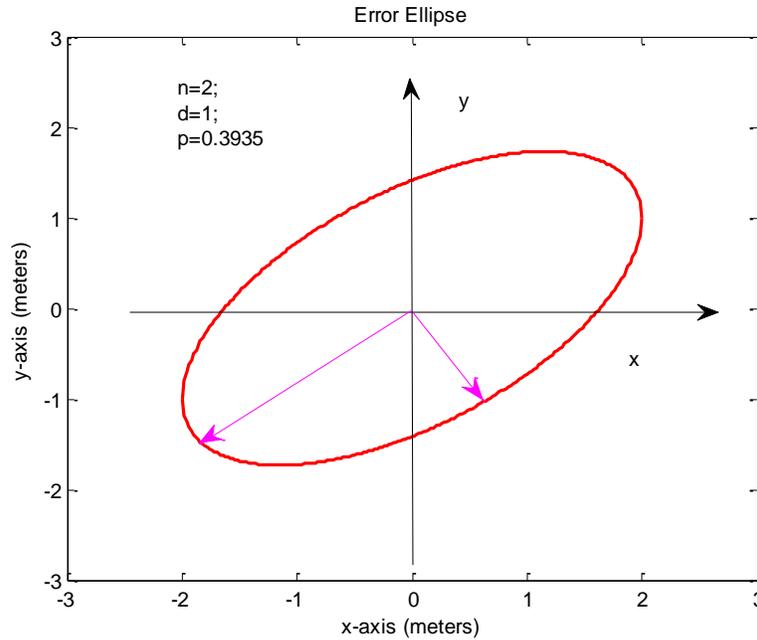
Let  $n=2$ , and  $C_X = \begin{bmatrix} 4 & 2 \\ 2 & 3 \end{bmatrix}$  meters-squared. The error ellipsoid is an ellipse whose boundary is specified

by  $\epsilon X C_X^{-1} \epsilon X^T = d^2$  or more specifically, since  $\epsilon X^T = [\epsilon x \quad \epsilon y]$  and  $C_X^{-1} = \begin{bmatrix} 3/8 & -1/4 \\ -1/4 & 1/2 \end{bmatrix}$ :

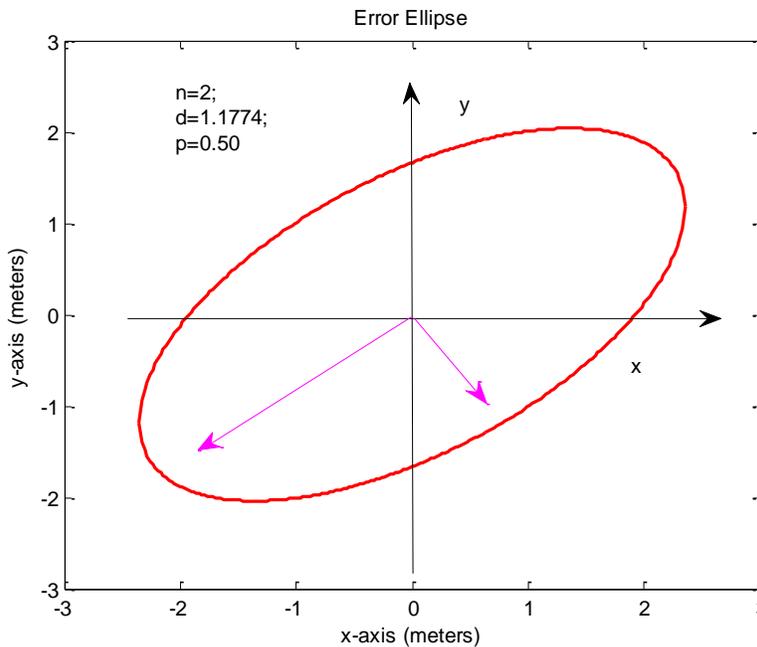
$$(3/8)\epsilon x^2 - 2(1/4)\epsilon x \epsilon y + (1/2)\epsilon y^2 = d^2 \tag{5.3.1-1}$$

This is an equation for an ellipse and guaranteed valid since  $C_X$  is positive definite, and thus,  $\det(C_X^{-1}) > 0$ .

Figure 5.3.1-1 plots the error ellipse corresponding to  $d = 1$ , which is often referred to as the standard error ellipse and has a probability level of only  $p = 0.3935$  per Table 5.3-2. Figure 5.3.1-2 plots the error ellipse corresponding to  $d = 1.1774$ , with a probability level  $p = 0.50$  per Table 5.3-1.



**Figure 5.3.1-1:** Error Ellipse (red) with  $d=1$  and  $p=0.39$  (often called standard error ellipse)



**Figure 5.3.1-2:** Error Ellipse (red) with  $d=1.1774$  and  $p=0.50$

The magnitude and direction of the semi-major axis of the first ellipse corresponds to the square root of the maximum eigenvalue and its (unit) eigenvector from the error covariance matrix  $C_X$ , respectively.

Similarly, the semi-minor axis corresponds to the square root of the minimum eigenvalue and its (unit) eigenvector. These are shown as the magenta lines (vectors) in the plot.

If  $d \neq 1$ , only the directions match, as illustrated in the second ellipse. In general, the magnitude of the semi-major and minor-axis equal  $d\sqrt{eigenvalue_{max}}$  and  $d\sqrt{eigenvalue_{min}}$ , respectively.

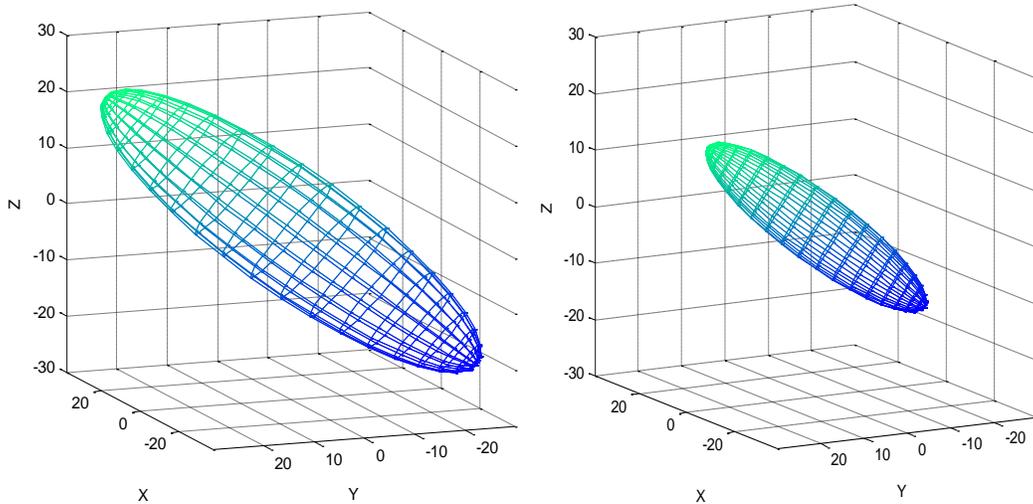
The above plots implemented Equation (5.3.1-1), expressing  $\epsilon y$  as a function of  $\epsilon x$  using the quadratic formula:

$$\epsilon y = (\epsilon x \pm \sqrt{-2\epsilon x^2 + 8d^2})/2, \epsilon x = -2, \dots, 2, \text{ where} \tag{5.3.1-2}$$

the end points in the range for  $\epsilon x$  correspond to a value of zero under the square-root, i.e., a single value for  $\epsilon y$ .

The above error ellipses can also be rendered “more naturally” using an eigenvector aligned coordinate system followed by a rotation back to the original Cartesian coordinate system – see Section 5.3.4.

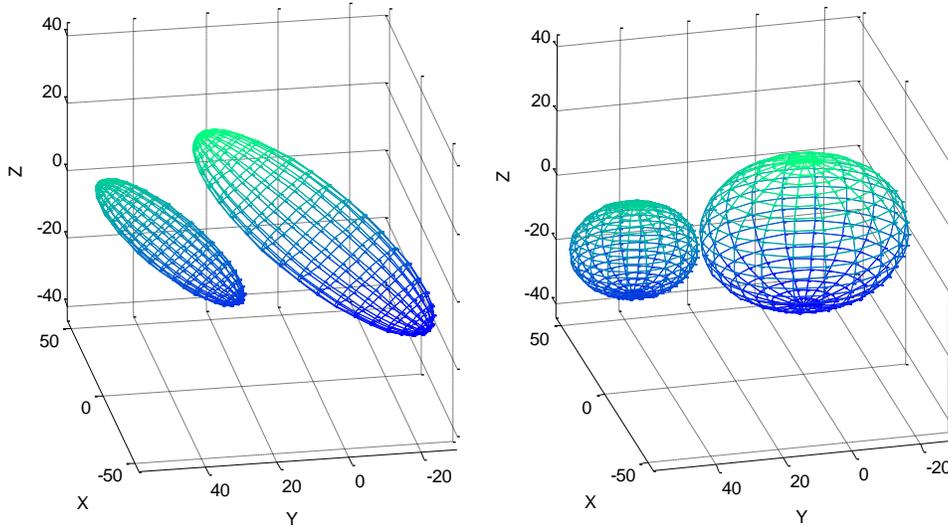
We now present additional 3D error ellipsoids ( $n = 3$ ) corresponding to the specific (symmetric)  $3 \times 3$  error covariance presented in Equation (5.3-1). The error ellipsoids presented in Figure 5.3.1-3 correspond to this error covariance but to two different probability levels – 90% and 50%. The 90% error ellipsoid on the left is significantly larger (more volume) than the 50% error ellipsoid on the right, but has the same shape. Note that with both ellipsoids, we can “see” how uncertainty varies with direction.



**Figure 5.3.1-3:** 90% and 50% error ellipsoids

Figure 5.3.1-4 illustrates the significance of (intra-state vector) correlation ( $\rho$ ) between the various components. The left side of the figure presents the 90% and 50% error ellipsoids again but places them side by side for ease of comparison. The right side of the figure presents the same error ellipsoids but with the off-diagonal elements of the covariance matrix mistakenly ignored, i.e.,  $\rho = 0$  for the various

cross-components. Note the incorrect shape of the error ellipsoids generated without correlations and how this leads to significant misunderstanding of which points may be within the 90% (or 50%) error ellipsoid.



**Figure 5.3.1-4:** 90% and 50% error ellipsoids generated with and without correlations

Finally, as a reminder, we have assumed that the error  $\epsilon X$  has a mean-value of zero, as typically the case. If not, simply modify Equation (5.3-2) for the error ellipsoid from  $\epsilon X^T C_X^{-1} \epsilon X$  to:

$$(\epsilon X - \bar{\epsilon X})^T C_X^{-1} (\epsilon X - \bar{\epsilon X}), \tag{5.3.1-3}$$

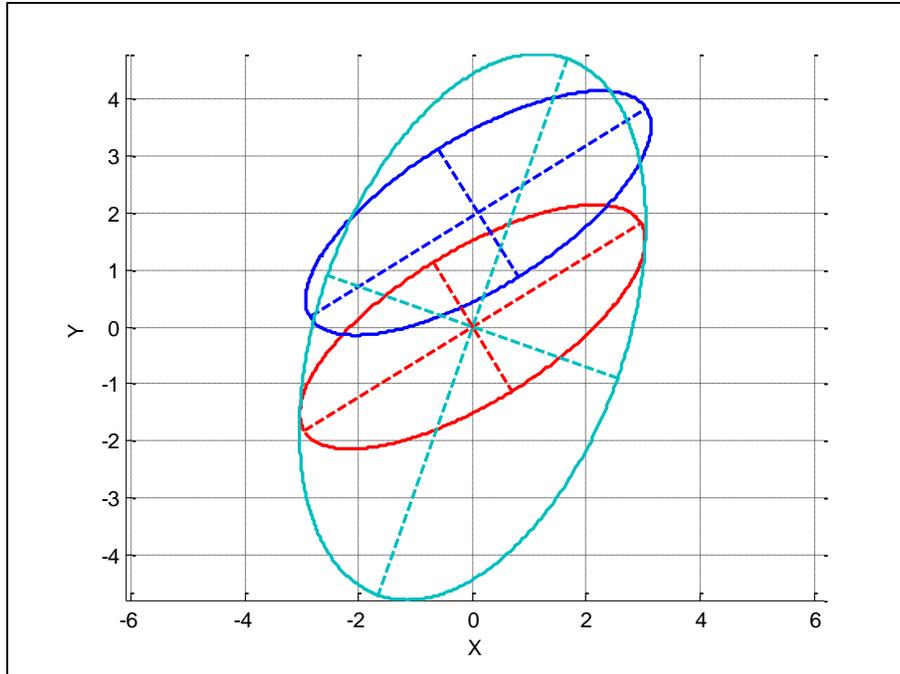
where  $\bar{\epsilon X}$  is the mean-value.

For example, Figure 5.3.1-5 plots three 0.9p ellipses. The first two ellipses have the same error covariance matrix about the mean-value, but the first (red) has a mean-value of zero and the second (blue) has a non-zero mean-value which becomes the ellipse origin. The corresponding predictive statistics for these ellipses are:

$$(1) \bar{\epsilon X} = 0, C_X = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, (2) \bar{\epsilon X} = [0.1 \quad 2]^T, C_X = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \tag{5.3.1-4}$$

The third ellipse (teal) corresponds to a “mean-value zero” approximation of the second’s predictive statistics (see Section 5.2.3):

$$(3) \bar{\epsilon X} = 0, C_X = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 2 \end{bmatrix} [0.1 \quad 2] \tag{5.3.1-5}$$



**Figure 5.3.1-5:** Three related error ellipses: blue differs from red due to a non-zero mean-value of error (blue ellipse origin); teal corresponds to a “mean-value zero approximation” of blue; semi-major and semi-minor axis included with each ellipse

### 5.3.2 Derivation of relationship between probability p & ellipsoidal normalized distance d

The following details how the values in Table 5.3-1 and Table 5.3-2 were generated, and therefore, how to generate additional entries corresponding to different probability levels, if so desired.

Let us assume that  $X$  and  $C_X$  correspond to a multivariate mean-zero Gaussian random variable (and dropping the explicit error notation  $\epsilon X$  for convenience). We will determine the probability that the multi-variate random variable resides within the ellipsoidal boundary in  $R^n$  defined by the quadratic form  $X^T C_X^{-1} X = d^2$ , where  $n=1, 2$ , or  $3$ , and the ellipsoidal “radius”  $d$  is specifiable. In general, the probability equals:

$$p = \frac{1}{(2\pi)^{n/2} \det(C_X)^{1/2}} \iiint_{X^T C_X^{-1} X \leq d^2}^* e^{-1/2(X^T C_X^{-1} X)} dx_1 \dots dx_n, \quad (5.3.2-1)$$

where the notation above specifies integration over the region  $X^T C_X^{-1} X \leq d^2$  in  $R^n$ .

Let us now evaluate the above for the explicit cases  $n=1, 2,$  and  $3,$  and assume that  $X$  and  $C_X$  correspond to representation by an eigenvector basis and eigenvalues ( $\lambda$ ) for simplicity (the above formula is applicable to either representation), i.e.,

$$E\{XX^T\} = C_X = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix}, \text{ and } \lambda_1 = \sigma_x^2, \lambda_2 = \sigma_y^2, \lambda_3 = \sigma_z^2.$$

**n=1:**

$$p = \frac{1}{\sqrt{2\pi}\sigma_x} \int_{\left(\frac{x^2}{\sigma_x^2}\right) \leq d^2}^* e^{-1/2(x^2/\sigma_x^2)} dx = \frac{1}{\sqrt{2\pi}} \int_{x^2 \leq d^2}^* e^{-1/2(x^2)} dx = \frac{2}{\sqrt{\pi}} \int_0^{d/\sqrt{2}} e^{-x^2} dx \equiv \text{erf}(d/\sqrt{2}), \quad (5.3.2-2)$$

where the above series of integrals are equivalent and reflect change of variables.

The integral “erf” is called the “error function”, a well-known function that has no closed form solution, but whose values are tabulated in great detail in many books, can be approximated by numerical integration, and is available in most pseudo-code (e.g., MATLAB) as a function call. Its inverse “erfinv” is also available in most pseudo-code as a function call.

$$\text{Thus, given the desired value } d, \text{ the corresponding } p = \text{erf}\left(\frac{d}{\sqrt{2}}\right); \quad (5.3.2-3)$$

$$\text{or given the desired value } p, \text{ the corresponding } d = \sqrt{2} \text{erfinv}(p).$$

**n=2:**

$$p = \frac{1}{2\pi\sigma_x\sigma_y} \iint_{\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \leq d^2}^* e^{-1/2\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)} dx dy = \frac{1}{2\pi} \iint_{x^2 + y^2 \leq d^2}^* e^{-1/2(x^2 + y^2)} dx dy \quad (5.3.2-4)$$

Switching to polar coordinates,  $x = r\cos(\theta), y = r\sin(\theta), dx dy = r dr d\theta,$  we have:

$$p = \frac{1}{2\pi} \int_0^{2\pi} \int_0^d e^{-1/2(r^2)} r dr d\theta = \frac{1}{2\pi} \int_0^{2\pi} \left(-e^{-\frac{d^2}{2}} + e^0\right) d\theta = \left(1 - e^{-\frac{d^2}{2}}\right). \quad (5.3.2-5)$$

$$\text{Thus, given the desired value } d, \text{ the corresponding } p = \left(1 - e^{-\frac{d^2}{2}}\right); \quad (5.3.2-6)$$

$$\text{or given the desired value } p, \text{ the corresponding } d = \sqrt{-2 \log_e(1 - p)}.$$

**n=3:**

$$p = \frac{1}{(2\pi)^{3/2}\sigma_x\sigma_y\sigma_z} \iiint_{\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} + \frac{z^2}{\sigma_z^2} \leq d^2}^* e^{-1/2\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} + \frac{z^2}{\sigma_z^2}\right)} dx dy dz = \quad (5.3.2-7)$$

$$\frac{1}{(2\pi)^{3/2}} \iiint_{x^2 + y^2 + z^2 \leq d^2}^* e^{-1/2(x^2 + y^2 + z^2)} dx dy dz.$$

Switching to spherical coordinates,

$x = r \sin(\phi) \cos(\theta)$ ,  $y = r \sin(\phi) \sin(\theta)$ ,  $z = r \cos(\theta)$ ,  $dx dy dz = r^2 \sin(\phi) dr d\phi d\theta$ , we have:

$$p = \frac{1}{(2\pi)^{3/2}} \int_0^{2\pi} \int_0^\pi \int_0^d e^{-1/2(r^2)} r^2 \sin(\phi) dr d\phi d\theta. \quad (5.3.2-8)$$

Now we can integrate  $\int_0^d e^{-1/2(r^2)} r^2 dr = 2^{3/2} \int_0^{d/\sqrt{2}} e^{-r^2} r^2 dr$  by parts to get:

$$\sqrt{\frac{\pi}{2}} \operatorname{erf}\left(\frac{d}{\sqrt{2}}\right) - d e^{-d^2/2}. \quad (5.3.2-9)$$

(For integration of the integral  $\int_0^{d/\sqrt{2}} e^{-r^2} r^2 dr$  by parts, represent it as  $\int_a^b u dv$ , where  $u = r$ ,  $dv = e^{-r^2} r dr$ ,  $a = 0$ ,  $b = d/\sqrt{2}$ .)

$$\text{Thus, } p = \frac{1}{(2\pi)^{3/2}} \left( \sqrt{\frac{\pi}{2}} \operatorname{erf}(d/\sqrt{2}) - d e^{-d^2/2} \right) \int_0^{2\pi} \int_0^\pi \sin(\phi) d\phi d\theta, \text{ or}$$

$$p = \operatorname{erf}(d/\sqrt{2}) - \sqrt{2/\pi} d (e^{-d^2/2}). \quad (5.3.2-10)$$

Thus, given the desired value  $d$ , the corresponding  $p$  equals the direct evaluation of Equation (5.3.2-10); or given the desired value  $p$ , the corresponding  $d$  equals the results of an iterative search for  $d$  such that Equation (5.3.2-10) is satisfied to within a small tolerance dictated by desired precision. (5.3.2-11)

### 5.3.3 Additional properties of the Error Ellipsoid

Referring back to the general equation for the error ellipsoid, Equation (5.3-2), the surface of the error ellipsoid corresponds to a constant probability density equal to  $(2\pi)^{-n/2} \det(C_X)^{-1/2} e^{-1/2(d^2)}$ . In addition, the 3d error ellipsoid requires the least volume over all surfaces to capture the specified level of probability it encloses. Similarly, the 2D error ellipse requires the least area over all curves, i.e., it is the “optimal shape”, given that the probability distribution is Gaussian. This can be proven using the Calculus of variations – see reference [29].

Regarding the volume (area) of an error ellipsoid defined by  $\epsilon X^T C_X^{-1} \epsilon X = d^2$ :

$$n = 2 \quad \text{Area} = \operatorname{sqrt}(\det(C_X)) \pi d^2 \quad (2D \text{ ellipse}) \quad (5.3.3-1)$$

$$n = 3 \quad \text{Volume} = \operatorname{sqrt}(\det(C_X)) (4/3) \pi d^3 \quad (3D \text{ ellipsoid}) \quad (5.3.3-2)$$

Note that  $\det(C_X)$  equals the product of the eigenvalues of the error covariance matrix  $C_X$ .

Finally, the error ellipsoid can correspond to either absolute error, as represented by the 3x3 error covariance matrix  $C_X$ , or relative error, as represented by the 3x3 relative error covariance matrix  $relC_X$  (see Section 5.5.4). Once the appropriate error covariance matrix is available, all procedures and interpretations involving the error ellipsoid are the same, other than whether absolute or a relative error is represented.

### 5.3.4 Rendering the Error Ellipsoid

The error ellipsoid is based on Equation (5.3-2) and relative to an original Cartesian coordinate system  $(x,y,z)$ . However the same error ellipsoid can be represented in a more straightforward manner in an eigenvector aligned Cartesian coordinate system  $(x',y',z')$ , where the eigenvectors and their eigenvalues correspond to the original error covariance matrix  $C_X$ . This is illustrated as follows for an error ellipse ( $n=2$ ) for convenience, although easily extendable to an error ellipsoid ( $n=3$ ) in a straightforward manner.

Because we assume a valid  $2 \times 2$  error covariance matrix  $C_X$ , i.e., symmetric and positive definite, there exists a  $2 \times 2$  unitary matrix  $\Phi$  that transforms vectors from the  $(x,y)$  system to the  $(x',y')$  system. The matrix rows of  $\Phi$  consist of the unit eigenvectors of  $C_X$ , and  $\Phi C_X \Phi^T = D$ , where  $D$  is a diagonal  $2 \times 2$  matrix with corresponding eigenvalues as the diagonal elements (maximum eigenvalue assumed in the  $x'$ -direction for specificity). In addition, since  $\Phi$  is unitary,  $\Phi^T = \Phi^{-1}$ . Thus:

$$D^{-1} = (\Phi C_X \Phi^T)^{-1} = \Phi^{T-1} C_X^{-1} \Phi^{-1} = \Phi C_X^{-1} \Phi^T. \quad (5.3.4-1)$$

$$\epsilon X^T C_X^{-1} \epsilon X = d^2 \quad (5.3.4-2)$$

$$\epsilon X^T (\Phi^T \Phi) C_X^{-1} (\Phi^T \Phi) \epsilon X = d^2$$

$$(\Phi \epsilon X)^T (\Phi C_X^{-1} \Phi^T) (\Phi \epsilon X) = d^2$$

$$\epsilon X'^T D^{-1} \epsilon X' = d^2$$

$$\frac{\epsilon x'^2}{\sigma_{eigmax}^2} + \frac{\epsilon y'^2}{\sigma_{eigmin}^2} = d^2.$$

This is illustrated in Figure 5.3.4-1. Note that, as directly implied by Equation (5.3.4-2), the error ellipse semi-major and semi-minor axis correspond to  $d$  multiplied by the square-root of the eigenvalues, i.e.,  $d\sigma_{eigmax}$  and  $d\sigma_{eigmin}$ , respectively.

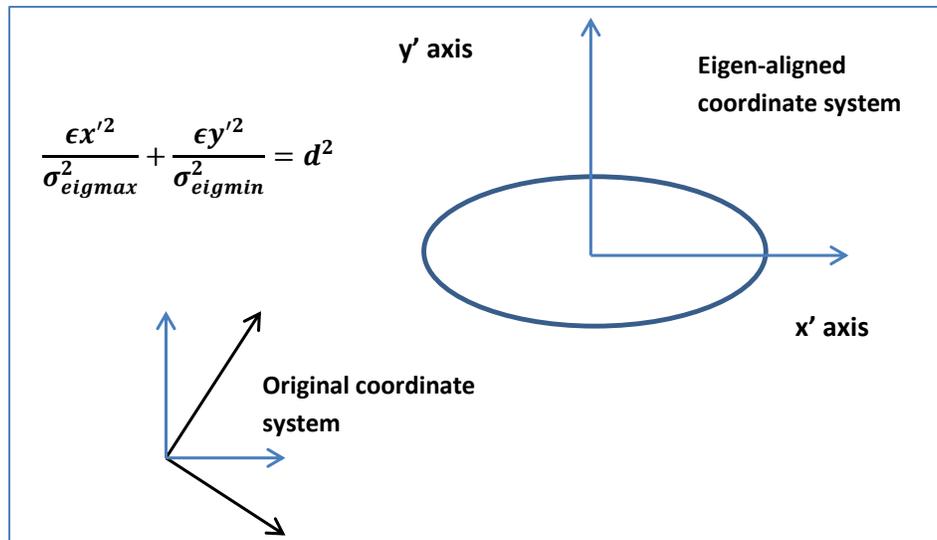


Figure 5.3.4-1: The error ellipse in the eigenvector-aligned coordinate system

See Appendix B for pseudo-code (MATLAB) to render a 2D error ellipse and a 3D error ellipsoid in the original Cartesian coordinate system. The approach is to generate the ellipsoid relative to an eigenvector-aligned coordinate system, and then rotate to the original coordinate system and render (draw). Inputs consist of the error covariance matrix, an optional mean-value, and either the desired probability  $p$  or normalized distance  $d$ .

Finally, an error ellipsoid and corresponding error covariance matrix are equivalent: given the desired probability level, one can be derived solely from the other. Specifically, corresponding to a desired dimension  $n$  and a desired level of probability (and hence, value of  $d$ ) the error ellipsoid is based solely on the  $n \times n$  error covariance matrix (inverse) in its defining formula (Figure 5.3-2), and although seldom performed in practice, the  $n \times n$  error covariance matrix can also be derived via a graph of the error ellipsoid corresponding to dimension  $n$  and desired level of probability (and hence, value of  $d$ ). This is most easily implemented by “reverse engineering” the above ellipsoid rendering procedure, i.e., determine the alignment and length of the error ellipsoid’s principal axes (e.g., semi-major and semi-minor axes), which correspond to the eigenvectors and eigenvalues (scaled by the value  $d$ ) of the error covariance matrix and which completely define it via the corresponding unitary transformation (matrix  $\phi$ ) described earlier in this subsection.

### 5.3.5 Comparison of Covariance Matrices

It is not uncommon in the literature concerning advanced linear algebra, estimation theory, and probability/statistics to see the expressions  $B \geq A$  or  $B > A$  for two error covariance matrices of the same dimension. What does this really mean and why is it so important?

First, by linear algebra convention,  $C \geq 0$  symbolizes a positive semi-definite matrix  $C$ , and  $D > 0$  symbolizes a positive definite matrix  $D$ . In addition, the set of all  $n \times n$  positive definite matrices is a proper subset of the set of all  $n \times n$  positive semi-definite matrices. If the matrix  $D$  is also symmetric, it can be considered a valid error covariance matrix, per subsection 5.2.

In the following equations (5.3.5-1) – (5.3.5-3), we assume that both  $A$  and  $B$  are valid  $n \times n$  error covariance matrices:

**Definitions for  $B \geq A$  and  $B > A$ :** (5.3.5-1)

$B \geq A$  is defined as  $B - A \geq 0$ , i.e.,  $(B - A)$  is a positive semi-definite matrix;

$B > A$  is defined as  $B - A > 0$ , i.e.,  $(B - A)$  is a positive definite matrix.

What are the various implications and properties corresponding to these definitions?

Assume that  $B > A$ :

(5.3.5-2)

(1)  $trace(A) < trace(B)$  and  $A(i, i) < B(i, i)$  for all  $i = 1, \dots, n$ .

Thus, the variance for each error component  $i$  is smaller in  $A$  than in  $B$ . Also, if  $A$  and  $B$  are the solution error covariances for the same but arbitrary state vector  $X$  from Estimators  $a$  and  $b$ , Estimator  $a$  is a “better” estimator than Estimator  $b$ . In fact, if  $B$  corresponds to any other estimator, and if  $A < B$  (or  $A \leq B$ ), Estimator  $a$  is a minimum mean-square estimator by definition.

proof of (1)

i)  $(B - A) > 0$  by assumption

ii)  $Y^T(B - A)Y > 0$  for all  $n \times 1$   $Y$  not equal to zero, by definition of a positive definite matrix

iii) Let  $Y'^T = [0 \dots 1 \dots 0]$  have an entry of 1 in the  $i$ -th component

iv)  $Y'^T(B - A)Y = B(i, i) - A(i, i) > 0$  via (ii)

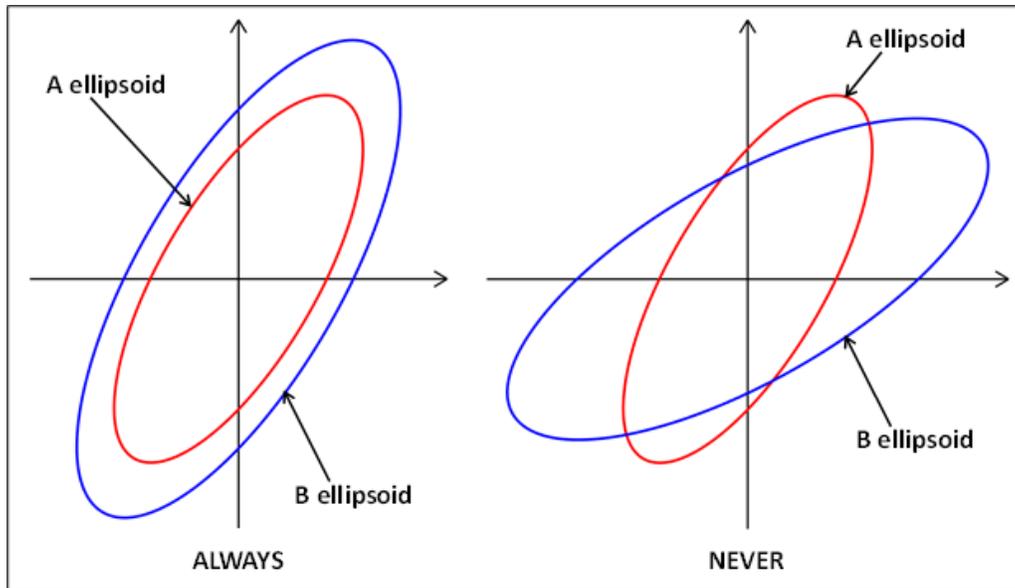
v) therefore,  $\sigma_i^2$  of matrix  $B > \sigma_i^2$  of matrix  $A$

vi) therefore,  $trace(B) > trace(A)$ .

(If  $B \geq A$ ,  $trace(A) \leq trace(B)$  and  $A(i, i) \leq B(i, i)$  for all  $i = 1, \dots, n$ ; proof - simply substitute all  $>$  signs with their  $\geq$  sign counterpart in the proof of (1) above.)

(2)  $A$  has a “better” error ellipsoid than  $B$ .

For example, the left portion of Figure 5.3.5-1 is always applicable, and the right portion is never applicable, where the error ellipsoids  $\epsilon X^T A^{-1} \epsilon X = d^2$  and  $\epsilon X^T B^{-1} \epsilon X = d^2$  are plotted ( $n = 2$ , i.e., an error ellipse for this example). This is intuitive:  $B > A$  implies that the error ellipsoid for  $A$  is always better than the error ellipsoid for  $B$ , regardless where along the ellipsoidal boundary.



**Figure 5.3.5-1:** The error ellipsoid corresponding to error covariance matrix A is better than that corresponding to error covariance matrix B.

(If  $B \geq A$ , A “almost always” has a better ellipsoid than B, since strictly speaking, the two ellipses may share a common boundary at two points ( $n = 2$ ), and two ellipsoids may share a common boundary along an ellipse ( $n = 3$ ). Thus, the left side of Figure 5.3.5-1 is still applicable, although slightly modified, and the right side of the figure remains applicable. This also assumes that  $B \geq A$  in the “strict sense”, i.e.,  $B \neq A$  and  $B > A$  is not true.)

proof of (2)

i)  $B > A$  implies that  $A^{-1} > B^{-1}$

ii)  $\epsilon X^T (A^{-1} - B^{-1}) \epsilon X > 0$  or  $\epsilon X^T A^{-1} \epsilon X > \epsilon X^T B^{-1} \epsilon X$  for all  $\epsilon X$  not equal to zero

iii) If  $\epsilon X'^T A^{-1} \epsilon X' = d^2$  at  $\epsilon X = \epsilon X'$ , then  $\epsilon X'^T B \epsilon X' < d^2$ , consistent with the left side of Figure 5.3.5-1.

iv) Suppose there exists  $\epsilon X = \epsilon X'$  such that  $\epsilon X'^T B^{-1} \epsilon X' > \epsilon X'^T A^{-1} \epsilon X'$  corresponding to the right side of figure - this contradicts (ii).

(If  $B \geq A$ : simply substitute all  $>$  and  $<$  signs with their  $\geq$  and  $\leq$  sign counterparts, respectively, in steps i-iii in the proof of (2) above.)

No longer assume that  $B > A$ :

(5.3.5-3)

(1)  $trace(A) < trace(B)$  and  $A(i, i) < B(i, i)$  for all  $i = 1, \dots, n$ , does not imply  $B > A$ .

proof of 1 by demonstration

i) Let  $n = 2$  and  $A = I_{2 \times 2}$ . Let  $B(1,1) = B(2,2) = 1.1$  and  $B(1,2) = B(2,1) = 0.9$ . Thus,  $trace(A) < trace(B)$  and  $A(i, i) < B(i, i)$  for  $i = 1, \dots, 2$ .

ii) At  $Y'^T = [1 \quad -1]$ ,  $Y'^T A Y' = 2$  and  $Y'^T B Y' = 0.4$ , thus

$Y'^T (B - A) Y' < 0$ , and therefore it is not true that  $B > A$ , i.e., it is not true that  $Y^T (B - A) Y > 0$  for all  $Y \neq 0$ .

iii) At  $\epsilon X'^T = [1 \quad 0]$ ,  $\epsilon X'^T A^{-1} \epsilon X' = 1$ , and at  $\epsilon X''^T = [1/\sqrt{2.75} \quad 0] \cong [0.6 \quad 0]$ ,  $\epsilon X''^T B^{-1} \epsilon X'' = 1$ ; therefore the error ellipse for  $A$  is not better than the error ellipse for  $B$  over the entire boundary of the former.

We can also illustrate this by showing the corresponding standard error ellipses ( $d = 1$ ) in Figure 5.3.5-2 below. The error ellipse for  $A$  is not better than the ellipse for  $B$ ; in fact, the error ellipse for  $B$  contains less area than the error ellipse for  $A$  (1.99 versus 3.14 meters-squared).

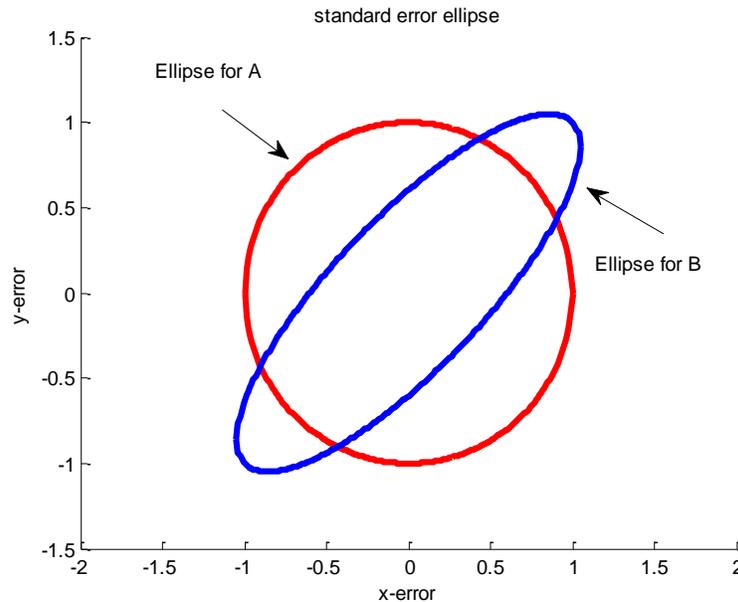


Figure : 5.3.5-2: The error ellipse for A (red) is not better than for B (blue) everywhere

(No longer assume that  $B \geq A$ :  $trace(A) \leq trace(B)$  and  $A(i, i) \leq B(i, i)$  for all  $i = 1, \dots, n$ , does not imply  $B \geq A$ ; the proof of (1) above is easily modified; the above figure remains applicable.)

### 5.3.6 Error ellipsoids: intersection and union

Sometimes we are interested in the “intersection” and “union” of two valid error covariance matrices  $A$  and  $B$  and their corresponding error ellipsoids. The two matrices have the same  $n \times n$  dimension, where  $1 \leq n \leq 3$ . The appropriate definitions follow:

$$C_{A \cap B} \equiv 2(A^{-1} + B^{-1})^{-1}, \text{ equal to the harmonic mean of matrices } A \text{ and } B. \quad (5.3.6-1)$$

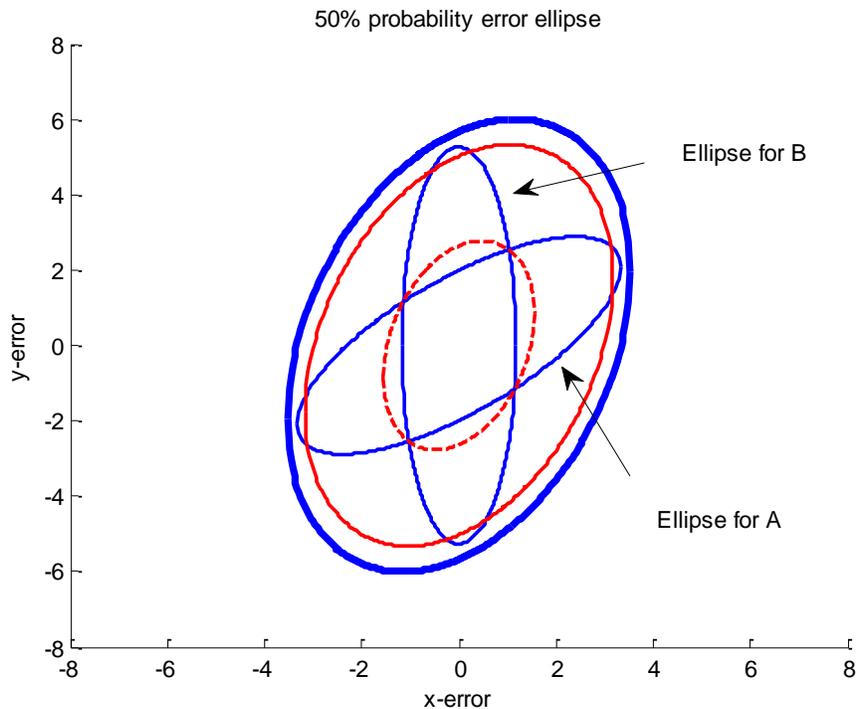
$$C_{A \cup B} \equiv (A + B) - C_{A \cap B} \quad (5.3.6-2)$$

We term the  $n \times n$  error covariance matrix  $C_{A \cap B}$  the “intersection error covariance”; similarly, we term the  $n \times n$  error covariance matrix  $C_{A \cup B}$  the “union error covariance”. These terms follow from the properties of their corresponding error ellipsoids as illustrated in the following example ( $n = 2$ ):

Let error  $\epsilon X_a = [\epsilon x_a \quad \epsilon y_a]^T$  with a mean-value of zero and covariance matrix  $A = \begin{bmatrix} 8 & 5 \\ 5 & 6 \end{bmatrix}$ ;

Let error  $\epsilon X_b = [\epsilon x_b \quad \epsilon y_b]^T$  with a mean-value of zero and covariance matrix  $B = \begin{bmatrix} 1 & 0 \\ 0 & 20 \end{bmatrix}$ .

Figure 5.3.6-1 presents the corresponding 50% error ellipses for error covariance matrices  $A$  (blue),  $B$  (blue),  $(A + B)$  (thick blue),  $C_{A \cap B}$  (red dashes), and  $C_{A \cup B}$  (red). (The probability level for the various ellipses is arbitrary, but must be common.)



**Figure 5.3.6-1:** Various error ellipses corresponding to covariance matrices  $A$  and  $B$  (Example 1)

Note that the error ellipse corresponding to the intersection error covariance matrix  $C_{A \cap B}$  (red dashes) is an ellipsoidal approximation of the intersection of the interiors of the error ellipses corresponding to the error covariance matrices  $A$  (blue) and  $B$  (blue). The error ellipse corresponding to the union error covariance matrix  $C_{A \cup B}$  (red) is an ellipsoidal approximation of the union of the interiors of the error ellipses corresponding to the error covariance matrices  $A$  (blue) and  $B$  (blue). It does not “double

count” their intersection; hence, is smaller than the error ellipse for  $(A + B)$ , i.e., in terms of covariance matrixes per se,  $C_{A \cup B} < (A + B)$ .

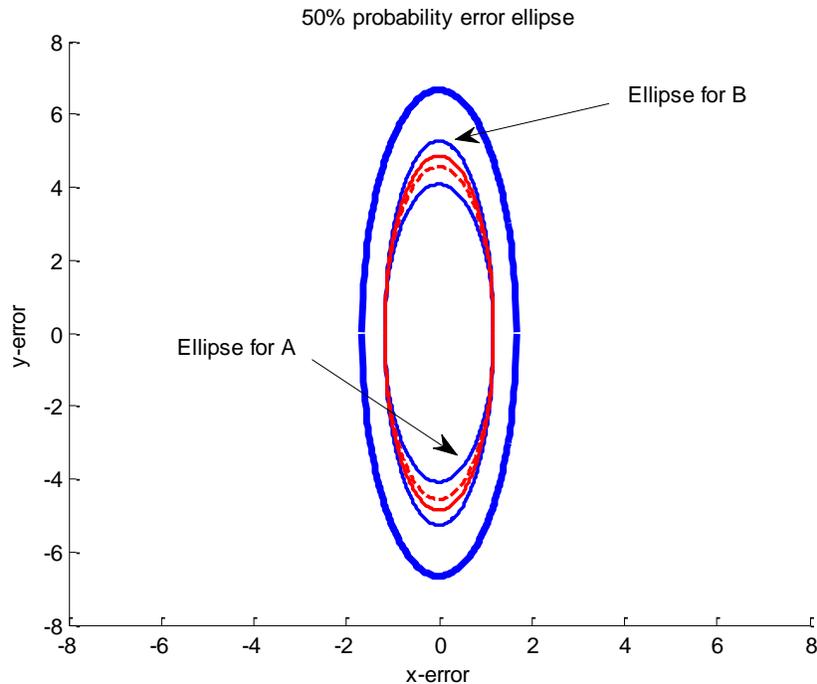
There are various applications where the error covariance matrices  $C_{A \cap B}$  and  $C_{A \cup B}$  and their error ellipsoids are of interest. One such general application corresponds to a 2d error that corresponds to either  $\epsilon X_a$  or  $\epsilon X_b$ . It may be unknown which of these errors is applicable for a particular application, or it may be that a common statistical error model is to be used for both for practicality. Either way, the error is defined as “ $\epsilon X_a$  or  $\epsilon X_b$ ”, not “ $\epsilon X_a$  and (+)  $\epsilon X_b$ ”. The error covariance matrix  $(A + B)$  corresponds to the latter and is too conservative (pessimistic) for the former. The error covariance matrix  $C_{A \cup B}$  is “tailored” to the former. It is virtually the smallest valid error covariance matrix (or equivalent error ellipsoid) that approximately contains both  $A$  and  $B$ , i.e.,  $A \approx \leq C_{A \cup B}$  and  $B \approx \leq C_{A \cup B}$ . Of course,  $C_{A \cup B}$  is an approximate error covariance matrix relative to both  $\epsilon X_a$  and  $\epsilon X_b$  when they are considered individually – a conservative approximation in general, but not too conservative.

This is further illustrated with an additional example similar to the first example except for modification of the error covariance matrix  $A$ :

Let error  $\epsilon X_a = [\epsilon x_a \quad \epsilon y_a]^T$  with a mean-value of zero and covariance matrix  $A = \begin{bmatrix} 1 & 0 \\ 0 & 12 \end{bmatrix}$ ;

Let error  $\epsilon X_b = [\epsilon x_b \quad \epsilon y_b]^T$  with a mean-value of zero and covariance matrix  $B = \begin{bmatrix} 1 & 0 \\ 0 & 20 \end{bmatrix}$ .

Figure 5.3.6-2 presents the corresponding 50% error ellipses for error covariance matrices  $A$  (blue),  $B$  (blue),  $(A + B)$  (thick blue),  $C_{A \cap B}$  (red dashes), and  $C_{A \cup B}$  (red).



**Figure 5.3.6-2:** Various error ellipses corresponding to covariance matrices  $A$  and  $B$  (Example 2)

Note the even larger disparity between the error ellipses for  $(A + B)$  (thick blue) and for  $C_{A \cup B}$  (red) in this example. Further note that the error ellipse for  $C_{A \cup B}$  (red) is somewhat closer to the error ellipse

for  $B$  (larger blue) than the error ellipse for  $A$  (smaller blue). This is appropriate for a reasonable approximation  $C_{A \cup B}$ , as the error ellipse for  $B$  is larger than the error ellipse for  $A$ ; i.e., regarding errors, it is better to be a little more pessimistic than optimistic.

The following proves that the error covariance matrices  $C_{A \cap B}$  and  $C_{A \cup B}$  are valid (positive definite) covariance matrices in general:

$C_{A \cap B}$  is positive definite because the inverse of a positive definite matrix is positive definite itself, and the positive sum of two positive definite matrices is positive definite (see subsections 5.2.1 and 5.2.2). The proof for  $C_{A \cup B}$  is outlined as follows:

$$(A^{-1} + B^{-1}) > A^{-1} \text{ (see subsection 5.3.5 for matrix } > \text{ definition)} \quad (5.3.6-3)$$

$$(A^{-1} + B^{-1})^{-1} < A$$

$$2(A^{-1} + B^{-1})^{-1} < (A + B)$$

$$(A + B) - 2(A^{-1} + B^{-1})^{-1} = (A + B) - C_{A \cap B} = C_{A \cup B} > 0.$$

(In addition, all the above matrices are symmetric as well, also required for an error covariance matrix.)

Finally, Equations (5.3.6-1) and (5.3.6-2) can be extended to more than two covariance matrices, if need be. For example, assume covariance matrices  $A$ ,  $B$ , and  $E$  are relevant. Compute:

$$C_{A \cup B} \equiv D, \text{ followed by } C_{D \cup E}, \text{ or symbolically } C_{((A \cup B) \cup E)}.$$

However, it must be pointed out that this approach is not associative, e.g.,  $C_{((A \cup B) \cup E)} \neq C_{((A \cup (B \cup E)))}$  in general, but typically  $C_{((A \cup B) \cup E)} \cong C_{((A \cup (B \cup E)))}$ .

## 5.4 Predictive Scalar Accuracy Metrics: Linear Error, Circular Error, and Spherical Error

This section of the document assumes that the underlying probability distribution of errors is Gaussian in order to assign probabilities to the predictive scalar accuracy metrics.

Scalar accuracy metrics are used to summarize accuracy and predicted accuracy, and more specifically, regarding the latter, approximate various portions of the corresponding 3d (3x3) error covariance matrix: Linear Error (LE) for  $z$ , Circular Error (CE) for  $x$ - $y$ , and Spherical Error (SE) for  $x$ - $y$ - $z$ . They also correspond to a specific level of probability. In this document, if not specified explicitly, such as CE\_50 for 50%, they are assumed to be at the 90% or  $p = 0.9$  probability level.

The derivation and practical calculations of these scalar metrics as predictive statistics are presented in this section and assume a Gaussian distribution of errors. Errors are also assumed to have a mean-value of zero unless specifically stated otherwise. If non-zero, the underlying data could simply be corrected

by the mean-value, with a resultant mean-value of error set equal to zero. (Note that a non-zero mean-value is sometimes termed a “bias”.)

CE corresponds to horizontal error and is computed from the upper left  $2 \times 2$  portion of the full  $3 \times 3$  error covariance matrix  $C_x$ . CE corresponds to the radius of a circle, centered at the origin, such that there is a 90% probability that the horizontal error resides within the circle, or equivalently, if the circle is centered at a target solution, there is a 90% probability that the true target horizontal location resides within the circle. LE corresponds to a vertical error and is computed from the lower right  $1 \times 1$  portion of the full  $3 \times 3$  error covariance matrix  $C_x$ . There is a 90% probability that the vertical error resides within  $\pm$  the LE value. (Note that we have assumed that the underlying x-y-z coordinate system is a local tangent plane system, i.e., x and y are horizontal components and z the vertical component.)

CE and LE can also be considered approximations to 2D and 1D (90%) error ellipsoids, respectively. Note that a 2D error ellipsoid is an ellipse, and a 1D error ellipsoid is a line. See Section 5.3.3 for a discussion on error ellipsoids, which are equivalent to the error covariance matrix and provide a visual display of the expected magnitude of errors, their directivity, and the interrelationships of their components.

CE and LE are easy to understand, visualize, and are in common use for military applications. The scalar accuracy metric SE, which corresponds to the radius of a 3D sphere, is also used to approximate the corresponding full  $3 \times 3$  error covariance matrix, or equivalently, the 3D (90%) error ellipsoid. Alternatively, the 3D error ellipsoid can be approximated by a CE-LE error cylinder, as described below; however, this requires two scalar metrics (CE and LE) as opposed to just one for SE.

Figures 5.4-1 and 5.4-2 presents examples of CE and a CE-LE cylinder, respectively, that approximate the upper left 2x2 and full 3x3 of the following error covariance matrix:

$$C_X = \begin{bmatrix} \sigma_x^2 & \rho_{xy}\sigma_x\sigma_y & \rho_{xz}\sigma_x\sigma_z \\ \cdot & \sigma_y^2 & \rho_{yz}\sigma_y\sigma_z \\ \cdot & \cdot & \sigma_z^2 \end{bmatrix} = \begin{bmatrix} 10^2 & 0.75 \cdot 10 \cdot 12 & 0.95 \cdot 10 \cdot 9 \\ \cdot & 12^2 & 0.8 \cdot 12 \cdot 9 \\ \cdot & \cdot & 9^2 \end{bmatrix}. \quad (5.4-1)$$

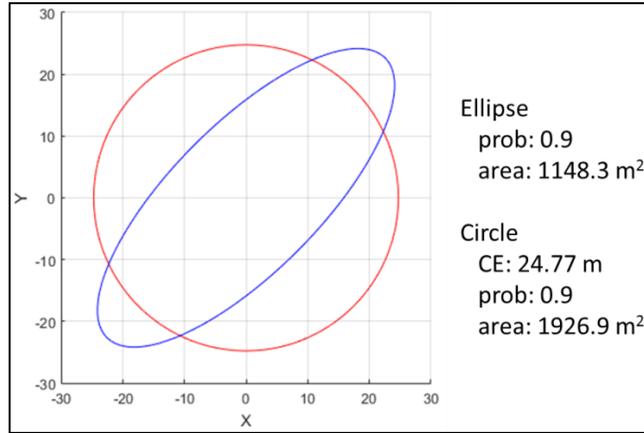


Figure 5.4-1: CE Circle vs Ellipse

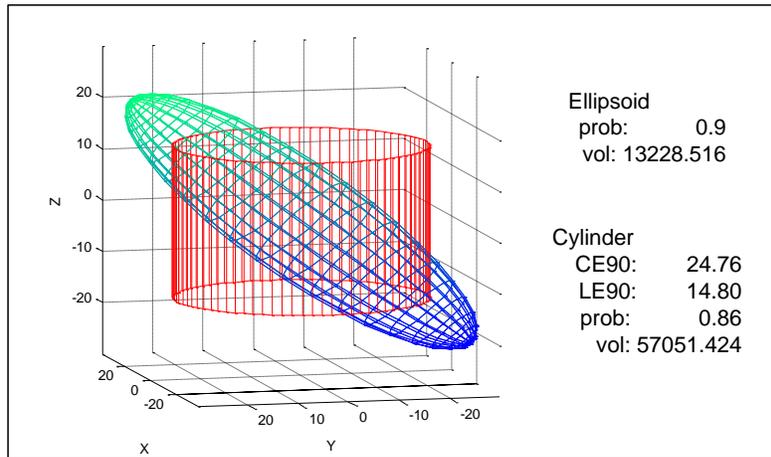


Figure 5.4-2: CE-LE Cylinder vs Ellipsoid

(note the change in x-axis orientation relative to the previous figure)

The top and bottom of the CE-LE cylinder correspond to a circle with radius CE meters. The wall of the cylinder is twice the length LE meters.

The amount of probability enclosed by the CE-LE cylinder is between 81-90%, depending how the vertical errors are correlated with the horizontal errors – if zero correlation, the enclosed probability is 81%, i.e.,  $(0.90^2)$ , if highly (positive or negative) correlated, the enclosed probability approaches 90%.

(In order to derive the actual probability enclosed by the cylinder when vertical errors are correlated with horizontal errors, i.e.,  $C_X(1,3) \neq 0$  and/or  $C_X(2,3) \neq 0$ , the multi-variate Gaussian probability density function is integrated over the three-dimensional region defined by the CE-LE cylinder. That is, Equation (5.1.3-4) is implemented with region  $R$  defined as the specific CE-LE cylinder of interest.)

Of course, LE (alone) approximates the lower right 1x1 of the error covariance matrix specified in Equation (5.4-1). In fact, as opposed to CE and SE, its approximation is exact, since both LE and the 1x1 error covariance matrix one can be derived from the other, given that the error distribution is assumed Gaussian and that the level of probability specified.

The 3D error ellipsoid can also be approximated directly via SE, the radius of the SE spheroid that encloses 90% of the probability. This is depicted in Figure 5.4-3 for the same 3x3 error covariance matrix detailed earlier. Note that the spheroid requires significantly more volume than does the 3D error ellipsoid to enclose the specified level of probability. (And, of course, as discussed in Section 5.3.3, the 3D error ellipsoid requires the least volume over all shapes.) This is to be expected: SE requires only one number, whereas the 3D error ellipsoid (error covariance matrix) requires 6 unique numbers and contains much more information.

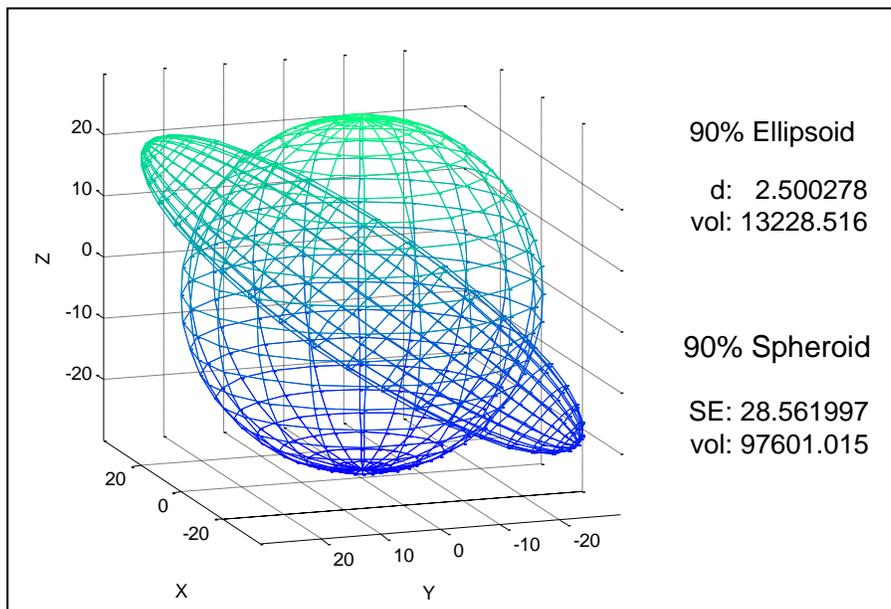


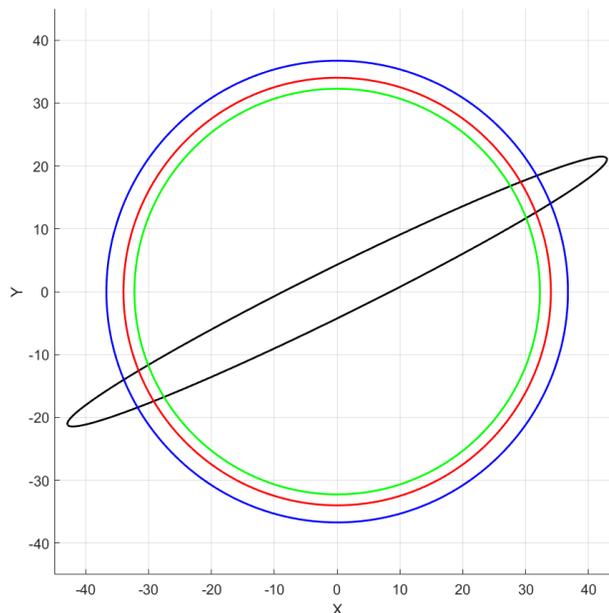
Figure 5.4-3: SE Spheroid vs. Ellipsoid

A desirable feature of scalar accuracy metrics is that they provide a natural representation of accuracy and a convenient summary of predicted accuracy. In fact, by definition, they have a specified probability of error associated with them. They are also used for the categorization of ordinance characteristics. However, scalar accuracy metrics do have limitations as documented in TGD 1 (Overview and Methodologies); thus, predictive scalar accuracy metrics should supplement but not replace the corresponding error covariance matrix.

The following subsections proceed to detail the computation of the predictive scalar accuracy metrics LE, CE, and SE. The error covariance matrix  $C_X$  is assumed to correspond to the relevant portion of the original 3x3 error covariance matrix  $C_X$ , i.e., the lower right 1x1 for LE, the upper left 2x2 for CE, and the full 3x3 for SE.

Sections 5.4.1, 5.4.2, and 5.4.3 correspond to LE, CE, and SE, respectively. Each of these sections includes appropriate derivations, followed by a subsection with the corresponding calculation algorithm. Derivations include those for specific levels of probability:  $p = 0.5, 0.9, 0.95, 0.99, \text{ and } 0.999$ , or in terms of percent:  $XX = 50, 90, 95, 99, \text{ and } 99.9$ ; for example, CE\_95 corresponding to  $XX=95$ .

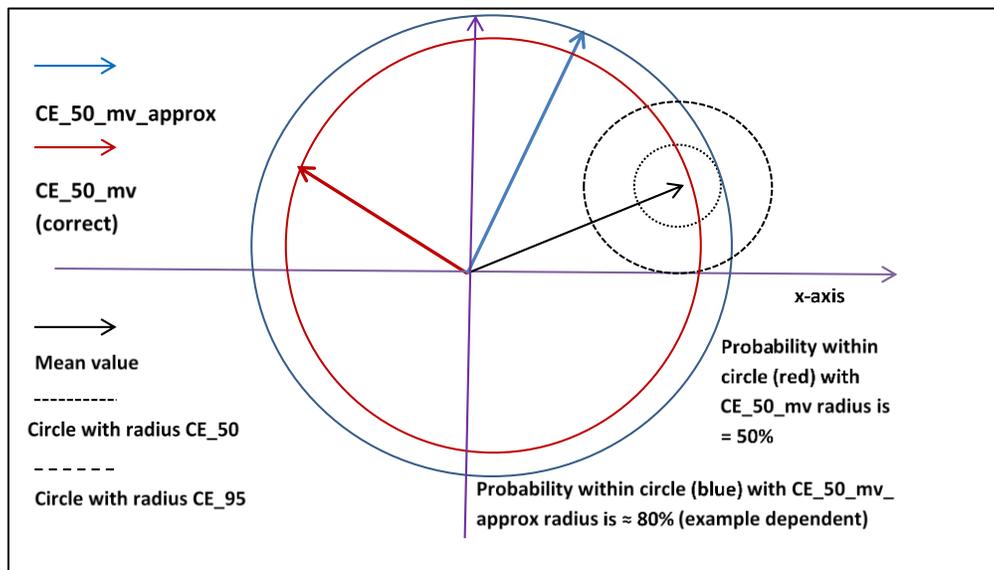
Note that these algorithms provide for high-fidelity approximations to their theoretical, exact calculation counterparts. There are other numerous approximations available, such as the “rms approximation”  $CE_{90}=2.15 \text{ rms}(\sigma_x, \sigma_y)$  and the “average approximation”  $CE_{90}=2.15 \text{ avg}(\sigma_x, \sigma_y)$ , that are even simpler, but low fidelity. They are not documented further, as the high-fidelity “standards” are easily implemented in today’s computer environment with easy to use programming languages (e.g., MATLAB). However, Figure 5.4-4 does present a comparison of various CE\_90 computation methods for a specific error ellipse (black): the high-fidelity baseline method (blue), the rms approximation (red), and the average approximation (green). The two low-fidelity approximations for CE\_90 are approximately 10% too small, i.e., optimistic. As documented later in this chapter, the baseline CE\_90 computation method has a computation (approximation) error on the order of only 0.01 % of the correct CE\_90 value. Finally, note that the square root of the largest eigenvalue and the square root of the smallest eigenvalue of the underlying 2x2 error covariance matrix are the semi-major and semi-minor axis of the ellipse in Figure 5.4-4. The ratio of the semi-minor to semi-major axis is approximately  $r = 0.1$  for this example.



**Figure 5.4-4:** Comparison of CE\_90 computation methods: high-fidelity baseline (blue), “rms approx” (red), “average approx” (green)

As mentioned earlier, the mean-value of predictive errors is almost always assumed zero. However, the following sections also provide a solution when the mean-value is not assumed zero. However, by definition, the corresponding error ellipsoids are still centered at the origin, not the mean-value of error. Thus, for example, given a specific error covariance matrix  $C_x$ , there is an algorithm to compute CE\_50 assuming a mean-value  $\bar{X} = 0$ . If, on the other hand, the specific error covariance is about a non-zero mean-value that is also specified, there is an algorithm to compute CE\_50 as well, let us call this “CE\_50\_mv” here for specificity. (Both algorithms are presented later in this section.)

Note that CE\_50\_mv is the radius of a circle about the origin, not the radius of a circle about the mean-value (location), that encloses 50% probability. Thus, CE\_50\_mv does not simply equal  $CE_{50\_mv\_approx} = |\bar{X}| + CE_{50}$ , which encloses significantly more than 50% probability, as illustrated in Figure 5.4-5:



**Figure 5.4-5:** Accounting for a non-zero mean value (mv) during the computation of CE\_50: the correct way “CE\_50\_mv” (red) versus the low-fidelity approximation “CE\_50\_mv\_approx” (blue)

Finally, a few words regarding notation close out this introduction to scalar accuracy metrics: CE\_XX is also sometimes written as CEXX. In addition, CE\_50 is sometimes referred to as “circular error probable” or “CEP”. Similar comments are applicable to LE and SE.

In the remainder of Section 5.4 of this document, the explicit error notation “ $\epsilon$ ” is dropped for convenience.

**5.4.1 Linear Error (LE)**

The following analytic formulation for LE at the XX % probability level is a straightforward application of the Gaussian probability density function and its properties to the definition of LE provided in Section 5.4.

LE\_XX is defined as that line length L such that:

$$p = \frac{1}{(2\pi)^{1/2}\sigma_z} \int e^{-1/2((z-\bar{z})^2/\sigma_z^2)} dz, \tag{5.4.1-1}$$

integrated over the region  $\sqrt{z^2} \leq L$ , and where probability  $p = XX/100$ , 1d error  $\epsilon X = \epsilon z$  is defined as  $z$  for notational convenience, with mean-value  $\bar{\epsilon X}$  defined as  $\bar{z}$ , and  $1 \times 1$  error covariance matrix  $C_X$  about the mean defined as  $\sigma_z^2$ . Note that if the mean-value is not zero, the length  $L$  is still relative to the origin per the standard definition of LE\_XX.

If we assume that the mean-value of error is zero, and change variables such that  $z/(\sigma_z\sqrt{2}) \rightarrow z^*$ , Equation (5.4.1-1) can be rewritten as:

$$p = \frac{2\sigma_z\sqrt{2}}{(2\pi)^{1/2}\sigma_z} \int_0^{L^*} e^{-z^{*2}} dz^* = \frac{2}{(\pi)^{1/2}} \int_0^{L^*} e^{-z^{*2}} dz^* \equiv \text{erf}(z^*), \tag{5.4.1-2}$$

where  $L^* = L/(\sigma_z\sqrt{2})$ .

Thus, since erf (Error Function) is a well-tabulated function and its inverse available via MATLAB and other programming languages, we have by definition,  $\text{erf}^{-1}(p) = \text{erf}^{-1}(XX/100) = L^*$ ; thus, and accounting for the change of variables:

$$LE_{XX} = \sigma_z\sqrt{2} \times \text{erf}^{-1}\left(\frac{XX}{100}\right) \tag{5.4.1-3}$$

And specifically:

$$LE_{XX} = L(p)\sigma_z, \tag{5.4.1-4}$$

where  $p = XX/100$  and the multiplier  $L(p)$  is listed in Table 5.4.1-1:

**Table 5.4.1-1:** Linear Error (LE) multiplier  $L(p)$  versus probability level  $p$

	Probabilities							
	p=0.5	p=0.6827	p=0.90	p=0.95	p=0.9545	p=0.99	p=0.9973	p=0.999
L(p)	0.6745	1.0000	1.6499	1.9600	2.0000	2.5758	3.0000	3.2905

The light blue entries are the standard probability levels of interest. The violet entries are others of general interest. For example,  $p=0.9973$  is the “three-sigma” level of probability. If the desired probability level is different than any of the above, simply evaluate Equation (5.4.1-3) using the desired value for XX. If the mean-value for error is not equal to zero, solve Equation (5.4.1-1) directly using iteration and numerical integration.

#### 5.4.1.1 Algorithm for Computing LE<sub>XX</sub>

The following are the priority-ordered methods/equations for the computation of LE<sub>XX</sub>:

- (1) Baseline table look-up (mean-value zero, at specific values of p=0.5, 0.9, 0.95, 0.99, and 0.999), see Equation (5.4.1-4).
- (2) Erf (inverse) evaluation (mean-value zero, arbitrary probability level), see Equation (5.4.1-3).
- (3) Integral Equation (arbitrary mean-value and probability level), see Equation (5.4.1-1).

Pseudo-code (MATLAB) for Equation (5.4.1-3) and Equation (5.4.1-1) are presented in Appendix C.

#### 5.4.1.2 Examples of LE<sub>XX</sub> computation

Examples are as follows:

- (1) Assume a desired probability level of 90%, a mean error of zero, and  $C_X \equiv \sigma_z^2 = [9]$  meters-squared. Thus, baseline table interpolation is applicable and the first-ordered choice:

$$LE_{90} = L \times \sigma_z = 1.6499 \times 3 = 4.95 \text{ meters.}$$

- (2) Assume a desired probability level of  $p = 0.70$  (XX=70), a mean error of zero, and  $C_X \equiv \sigma_z^2 = [9]$  meters-squared. Thus, erf (inverse) evaluation is applicable:

$$LE_{70} = 3.1092 \text{ meters.}$$

- (3) Assume a desired probability level of 90%, a mean predictive error equal to  $\bar{X}^T \equiv \bar{z} = [-2]$ , and  $C_X \equiv \sigma_z^2 = [9]$  meters-squared. Thus, the Integral Equation is applicable:

$$LE_{90} = 5.976 \text{ meters.}$$

The solution corresponding to the first and second examples were computed virtually instantaneously, while the solution corresponding to the third example took on the order of 0.02 seconds using non-optimized MATLAB code on a notebook computer. The calculation error was negligible for all.

#### 5.4.2 Circular Error (CE)

The following analytic formulation for CE at the XX % probability level is a straightforward application of the Gaussian (multi-variate) probability density function and its properties to the definition of CE provided in Section 5.4.

CE<sub>XX</sub> is defined as that circular radius  $R$  such that:

$$p = \frac{1}{(2\pi)\det(C_X)^{1/2}} \iint e^{-1/2((X-\bar{X})^T C_X^{-1} (X-\bar{X}))} dx dy, \quad (5.4.2-1)$$

integrated over the region  $\sqrt{x^2 + y^2} \leq R$ , and where probability  $p = XX/100$ , 2d error  $\epsilon X^T = [\epsilon x \ \epsilon y]$  is defined as  $X^T = [x \ y]$  for notational convenience, with mean-value  $\bar{\epsilon X}$  defined as  $\bar{X}^T = [\bar{x} \ \bar{y}]$ , and  $2 \times 2$  error covariance matrix about the mean defined as  $C_X$ . Note that if the mean-value is not zero, the radius  $R$  is still relative to the origin  $[0 \ 0]^T$ , per the standard definition of CE<sub>XX</sub>.

The above integral relationship can be simplified by assuming an eigenvector aligned x-y Cartesian coordinate system. The latter takes advantage of the positive-definite and symmetric properties of the error covariance matrix. In particular, there exists a Cartesian coordinate system aligned with the error covariance matrix eigenvectors such that the error covariance matrix when expressed relative to this system is a diagonal matrix with the eigenvalues down the diagonal. There also exists a transformation matrix  $\Phi$  with (unit) eigenvectors along the rows which transforms vectors in the original x-y coordinate system to corresponding vectors in the eigenvector aligned system. Thus, and taking advantage of circular symmetry (the radius  $R$  applies to either the original or eigenvector-aligned coordinate system), we have the following:

Assume that  $\Phi\bar{X} \rightarrow \bar{X}$ ,  $\Phi C_X \Phi^T \rightarrow C_X = \begin{bmatrix} \sigma_{max}^2 & 0 \\ 0 & \sigma_{min}^2 \end{bmatrix}$ , where the diagonals are the maximum and minimum eigenvalues, with  $\sigma_{max}^2$  assumed associated with the x-axis of the eigenvector aligned Cartesian coordinate system for convenience of notation. Equation (5.4.2-1) becomes the equivalent:

$$p = \frac{1}{(2\pi)\sigma_{max}\sigma_{min}} \iint e^{-1/2\left(\left(\frac{x-\bar{x}}{\sigma_{max}}\right)^2 + \left(\frac{y-\bar{y}}{\sigma_{min}}\right)^2\right)} dx dy, \quad (5.4.2-2)$$

integrated over the region  $\sqrt{x^2 + y^2} \leq R$ .

Note that Equation (5.4.2-2) can be further decomposed, if so desired, as follows:

$$p = \frac{1}{(2\pi)\sigma_{max}\sigma_{min}} \int_{-R}^R e^{-(x-\bar{x})^2/\sigma_{max}^2} \int_{-\sqrt{R^2-x^2}}^{+\sqrt{R^2-x^2}} e^{-(y-\bar{y})^2/\sigma_{min}^2} dy dx. \quad (5.4.2-3)$$

Either of the above Equations (5.4.2-1) or (5.4.2-2) can be solved for iteratively for the radius  $R$ , given the desired probability level  $p = XX/100$ . Note that the right side of the corresponding equation for a given iteration is solved for numerically for the radius  $R$ , with Equation (5.4.2-2) somewhat more numerically stable. Thus, we have:

$$CE_{XX} = R. \quad (5.4.2-4)$$

(Note: if the mean-value is zero and the eigenvalues are equal, the integral in Equation (5.4.2-2) can also be represented in terms of the random variable  $radial = \sqrt{x^2 + y^2}$  and a single integral. The probability distribution of the *radial* random variable is the Rayleigh distribution. If the mean-value is not equal to zero and the eigenvalues are equal, the probability distribution of the *radial* random variable is the Rice distribution which involves a modified Bessel function of the first kind. However, equal eigenvalues are equivalent to a covariance matrix relative to the original and already eigenvector aligned coordinate system that is diagonal with equal variances (standard deviations). This is a significant restriction. Equation (5.4.2-2) in its current form is more general and requires nothing more than a valid covariance matrix. It is also the foundation for the recommended methods that follow.)

Assuming a mean-value of zero and an additional change of variables from the eigenvector-aligned x-y system to a scaled eigenvector aligned  $x^*-y^*$  system corresponding to  $x/\sigma_{max} \rightarrow x^*$  and  $y/\sigma_{min} \rightarrow y^*$ , Equation (5.4.2-2) can also be written as:

$$p = \frac{1}{(2\pi)} \iint e^{-1/2(x^{*2}+y^{*2})} dx^* dy^* \quad (5.4.2-5)$$

integrated over the region  $\sqrt{x^{*2} + r^2 y^{*2}} \leq R/\sigma_{max}$ , where  $r = \sigma_{min}/\sigma_{max}$ .

If we also assume that  $\sigma_{max} = 1$ , it follows that the value  $R = R(p, r)$ , such that the above integral equals the desired level of probability  $p$ , is related to CE\_XX as follows:

$$CE_{XX} = R(p, r)\sigma_{max}, \quad (5.4.2-6)$$

where  $\sigma_{max}$  in Equation (5.4.2-6) is the square root of the actual maximum eigenvalue of  $C_X$ , and  $r = \sigma_{min}/\sigma_{max}$ .

Typically,  $R(p, r)$  is pre-computed for all combination of values of  $r = 0, 0.05, 0.1, \dots 0.95, 1.0$ , i.e., 21 values or table entries, and then interpolated appropriately.

A given entry for the table is pre-computed by solving Equation (5.4.2-2) for  $R$ , with  $\bar{X}$  set to zero, and  $C_X = \begin{bmatrix} \sigma_{max}^2 & 0 \\ 0 & \sigma_{min}^2 \end{bmatrix}$  set to  $\begin{bmatrix} 1 & 0 \\ 0 & r^2 \end{bmatrix}$ . An exception corresponds to values of  $r = 0$ , where the appropriate table value corresponds to limit arguments. Specifically, if  $r = 0$ , the CE\_XX table entry corresponds to LE\_XX, i.e., the expected magnitude of error in the second dimension approaches zero or is a "non-entity".

The following table presents the pre-computed values of  $R(p, r)$  for various probability levels. In particular, columns 2-5 correspond to  $p=0.5, 0.9, 0.95, 0.99$ , and  $0.999$ , respectively, or alternatively, to  $XX=50, 90, 95, 99$ , and  $99.9\%$ , respectively.

**Table 5.4.2-1:** Circular Error (CE) multiplier  $R(p, r)$  versus probability level  $p$  and ratio  $r$ :

Ratio $r$	Probabilities				
	$p=0.5$	$p=0.9$	$p=0.95$	$p=0.99$	$p=0.999$
0.00	0.6745	1.6449	1.9600	2.5758	3.2905
0.05	0.6763	1.6456	1.9606	2.5763	3.2910
0.10	0.6820	1.6479	1.9625	2.5778	3.2921
0.15	0.6916	1.6518	1.9658	2.5803	3.2940
0.20	0.7059	1.6573	1.9704	2.5838	3.2967
0.25	0.7254	1.6646	1.9765	2.5884	3.3003
0.30	0.7499	1.6738	1.9842	2.5942	3.3049
0.35	0.7779	1.6852	1.9937	2.6013	3.3104
0.40	0.8079	1.6992	2.0051	2.6099	3.3172
0.45	0.8389	1.7163	2.0190	2.6203	3.3252
0.50	0.8704	1.7371	2.0359	2.6326	3.3346
0.55	0.9021	1.7621	2.0564	2.6474	3.3459
0.60	0.9337	1.7915	2.0813	2.6653	3.3595
0.65	0.9651	1.8251	2.1111	2.6875	3.3759
0.70	0.9962	1.8625	2.1460	2.7151	3.3965
0.75	1.0271	1.9034	2.1858	2.7492	3.4227
0.80	1.0577	1.9472	2.2303	2.7907	3.4570
0.85	1.0880	1.9936	2.2791	2.8401	3.5018
0.90	1.1181	2.0424	2.3318	2.8974	3.5594
0.95	1.1479	2.0932	2.3881	2.9625	3.6310
1.00	1.1774	2.1460	2.4478	3.0349	3.7169

For an arbitrary error covariance matrix  $C_X$  with corresponding ratio  $r$  ( $0 < r \leq 1$ ) and  $\sigma_{max}$ ,  $CE_{XX}$  is computed as  $CE_{XX} = R^* \sigma_{max}$ , where the normalized radius  $R^*$  is computed as the linear interpolation of  $R(XX/100, r)$  from the corresponding column of Table 5.4.2-1.

**5.4.2.1 Baseline Computation Method: Table Interpolation**

As detailed above, the baseline interpolation method to compute  $CE_{XX}$  assumes a mean-value of zero and fixed probability levels. It is summarized as an algorithm as follows:

- (1) Compute the eigenvalues of  $C_X$ :  $\sigma_{max}^2, \sigma_{min}^2$ , assumed in descending order (5.4.2.1-1)
- (2) Compute  $\sigma_{max}, r = \sigma_{min}/\sigma_{max}$ ; there is no limit on the value of  $r$  other than  $0 < r \leq 1$ .
- (3) Based on the desired probability level  $XX$  (%) and the computed value  $r$ , perform linear interpolation within the appropriate column of Table 5.4.2-1, i.e.,  $R(p = XX/100, r)$  of Section 5.4.2 for a normalized radius value  $R^*$
- (4)  $CE_{XX} = R^* \sigma_{max}$ .

**5.4.2.2 Alternate Computation Method: Monte Carlo Matrix Square Root**

The following alternate approach to the computation of CE\_XX is applicable to arbitrary mean-values and arbitrary probability levels, is computationally accurate and reasonably fast:

(1) Compute 1E6 independent samples of the 2x1 horizontal error:  $s_i = \bar{X} + C_X^{1/2} n_i$ , (5.4.2.2-1)

$i = 1, \dots, 1E6,$

where  $\bar{X}$  and  $C_X$  are the 2x1 mean and the 2x2 error covariance about the mean relative to the original (non-eigenvector aligned) coordinate system,  $n_i$  is a two-element vector with each element the realization of an independent Gaussian or normal  $N(0,1)$  random variable, and where the superscript “1/2” indicates principal matrix square root.  $\bar{X}$  and  $n_i$  are 2x1 vectors, and  $C_X^{1/2}$  is a 2x2 matrix. Also,  $s_i$  is a Gaussian distributed random vector with mean  $\bar{X}$  since it is a linear function of the mean-zero random vector  $n_i$  and added to  $\bar{X}$ .

(2) Order the magnitudes of the error samples  $s_i$  from smallest to largest, and designate  $RE_{XX}$  the XX\_th percent largest, and  $RE_{XX}^*$  the next largest magnitude.

(3)  $CE_{XX} = (RE_{XX} + RE_{XX}^*) / 2$ .

Note that the symmetric  $C_X^{1/2}$  is computed once prior to generating the independent samples, and the samples  $C_X^{1/2} n_i$  are consistent with the error covariance matrix about the mean, i.e.,

$$E\{(s_i - \bar{X})(s_i - \bar{X})^T\} = E\{C_X^{1/2} n_i (C_X^{1/2} n_i)^T\} = C_X^{1/2} E\{n_i n_i^T\} C_X^{1/2} = C_X^{1/2} I_{2x2} C_X^{1/2} = C_X,$$

where  $E\{ \}$  is the expected value operator.

Alternatively, the above can be performed in an equivalent manner relative to the eigenvector-aligned system by computing samples of horizontal error as  $s_i = (\Phi \bar{X} + D \cdot n_i)$ , where “ $\cdot$ ” is the vector dot product,  $D$  a 2x1 vector containing the square-root of the eigenvalues, and  $\Phi$  the 2x2 transformation matrix from the original Cartesian coordinate system to the eigenvector-aligned coordinate system. The speed varies little between the two approaches.

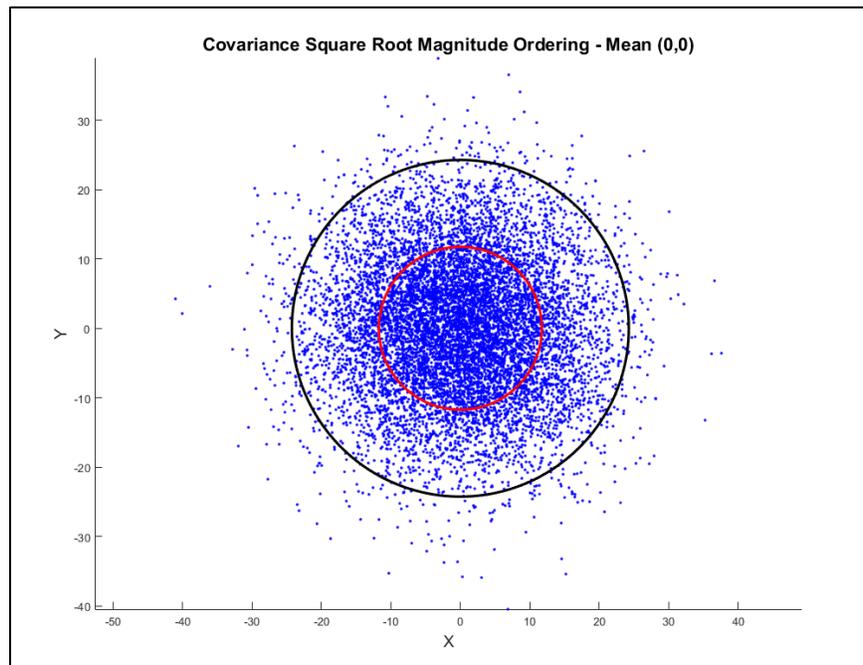
Due to its use of 1E6 random samples, the computational accuracy of the above algorithm is directly associated with statistical significance, and resultant computational error is expected to be on the order of  $1/\sqrt{1E6}$ , or a 0.1% relative error. This assumes reasonable and practical probability values that are within the interval [0.1, 0.999], and that the square-root of the smallest to the largest eigenvalue  $r > 0.0001$ , i.e., applicable to virtually any valid error covariance matrix of interest.

**5.4.2.3 Examples of Monte Carlo Matrix Square Root Method**

The following are examples of the application of Equation/Algorithm (5.4.2.2-1).

The first example assumes a mean-value of zero and a 2x2 diagonal error covariance with 100 meters-squared in each diagonal. The equation was applied twice: once for CE\_50 and once for CE\_95.

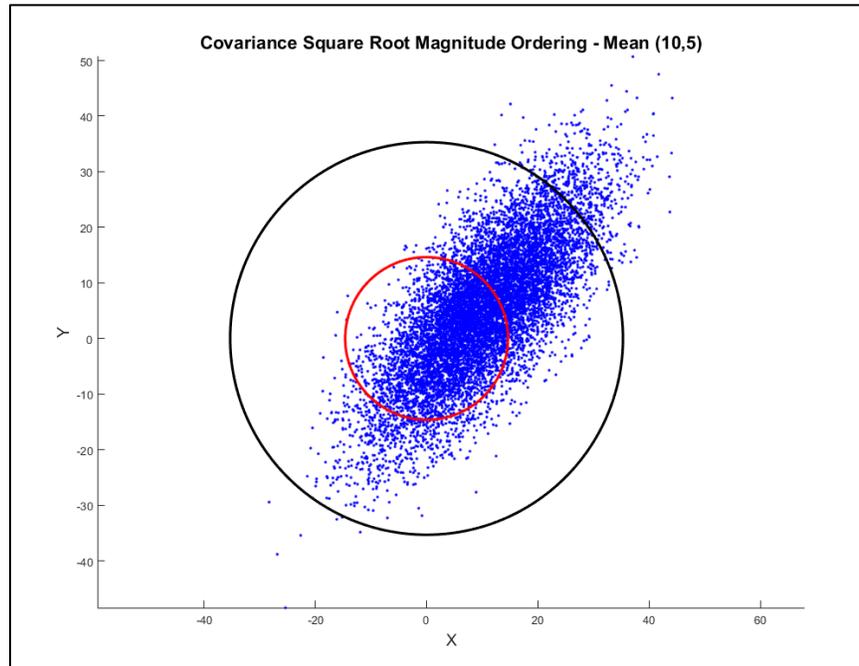
The results are plotted in Figure 5.4.2.3-1, including the first 10,000 of the 1,000,000 independent samples used in the calculation of CE\_50 for context. (The CE\_50 circle in the figure was computed using all 1,000,000 independent samples. The CE\_95 circle was computed similarly, but used a different set of 1,000,000 independent samples for convenience. Both circles are centered at zero by definition.)



**Figure 5.4.2.3-1:** Example 1 - CE\_50 circle (red), CE\_95 circle (black), and 10,000 of 1,000,000 random samples

The next example assumes a mean-value  $\bar{X}^T = [10 \ 5]$  meters, and an error covariance matrix about the mean  $C_X = \begin{bmatrix} 10^2 & 0.75 \times 10 \times 12 \\ & 12^2 \end{bmatrix}$  meters-squared. The equation was applied twice: once for CE\_50 and once for CE\_95.

The results are plotted in Figure 5.4.2.3-2, including the first 10,000 of the 1,000,000 independent samples used in the calculation of CE\_50 for context. (The CE\_50 circle in the figure was computed using all 1,000,000 independent samples. The CE\_95 circle was computed similarly, but used a different set of 1,000,000 independent samples for convenience. Both circles are centered at zero by definition.)



**Figure 5.4.2.3-2:** Example 2 - CE\_50 circle (red), CE\_95 circle (black), and 10,000 of 1,000,000 random samples

Note that sample results (blue points) are not centered about zero and in a non-symmetric fashion due to a mean-value with different non-zero components in the x and y directions. Also note that the actual statistical significance is greater than that implied by the figure, which displays only 1/100\_th the actual number of samples used in the calculation of CE\_50.

**5.4.2.4 CE\_XX Computation Method Selection**

Pseudo-code (MATLAB) for the computation of CE\_XX is presented in Appendix C corresponding to the following operational, priority-ordered methods/equations:

- (1) Baseline table interpolation (mean-value zero,  $p=0.5, 0.9, 0.95, 0.99, \text{ or } 0.999$ ), see Equation (5.4.2.1-1).
- (2) Monte Carlo Matrix Square Root (arbitrary mean-value and probability level), see Equation (5.4.2.2-1).
- (3) Integral Equation (mean-value zero, arbitrary probability level), see Equation (5.4.2-2).

Operationally, Method (1) is the preferred method for the fixed levels of probability previously specified and when the mean-value of predictive error is zero, which is almost always the case. It has small calculation error and is much faster than all other methods.

Operationally, Method (2) is preferred over Method (3) due to no extreme variations in calculation time, although the latter has less calculation error. Method (3) is the preferred method for the generation of new interpolation tables corresponding to probability levels not already fixed.

Note that Method (3)'s Equation (5.4.2-2) can also be used when the mean-value is not zero, but it does not converge as fast as when the mean-value equals zero, or not at all a small percentage of the time; hence, it was left out of the above list, although pseudo-code is also available in Appendix C. The convergence issue and how to detect it is also discussed further in Appendix C.

In general, when a non-zero mean is added, the solution to Equation (5.4.2-2) becomes more difficult and time-consuming. The solution involves an iterative search over numerical evaluation of the integral such that the resultant probability is near the specified amount on the left side of the equation. This becomes more difficult because the candidate CE circle is centered about the origin, not the mean-value about which the error covariance is "centered". (See Figure 5.4.2.3-2 as a corresponding, but sample-based, example.) Further note that in order to improve convergence and throughput corresponding to Equation (5.4.2-2), the square-root of the smallest to largest eigenvalue ratio is assumed  $r \geq 0.02$  if the mean-value is zero, otherwise  $r \geq 0.05$ .

Table 5.4.2.6-1 of Section 5.4.2.6 presents a performance summary for all methods.

**5.4.2.5 Examples of CE<sub>XX</sub> computation**

Examples are as follows:

(1) Assume a desired probability level of 90%,

a mean error of zero, and  $C_X = \begin{bmatrix} 4 & 2 \\ 2 & 3 \end{bmatrix}$  meters-squared. Thus, baseline table interpolation (Equation (5.4.2-6)) is applicable per the ordered priorities of Section 5.4.2.4 and is the first choice:

Eigenvalues equal 5.562 and 1.438 meters-squared

$$\sigma_{eig\_max} = 2.36 \text{ meters}, r = .509$$

$$R^* = 1.74 \text{ (via linear interpolation: } \frac{0.041}{0.05} 1.7371 + \frac{0.009}{0.05} 1.7621 = 1.7416)$$

$$CE_{90} = R^* \sigma_{max} = 4.11 \text{ meters.}$$

(2) Assume a desired probability level of 90%,

a mean-value  $\bar{X}^T = [1 \quad -3]$ , and  $C_X = \begin{bmatrix} 4 & 2 \\ 2 & 3 \end{bmatrix}$  meters-squared. Thus, since the mean-value is not zero, the Monte-Carlo Matrix Square Root method (Equation (5.4.2.2-1)) is applicable:

$$CE_{90} = 5.69 \text{ meters.}$$

(3) Assume a desired probability level of 70% and the smallest possible calculation error,

a mean error of zero, and  $C_X = \begin{bmatrix} 4 & 2 \\ 2 & 3 \end{bmatrix}$  meters-squared. Thus, since p=0.7 is not one of the fixed probability levels previously specified, the Integral Equation with a mean-value of zero (Equation (5.4.2-2)) is applicable:

Eigenvalues equal 5.562 and 1.438 meters-squared

$$\sigma_{eig\_max} = 2.36 \text{ meters}, r = .509$$

$$CE_{70} = 2.81 \text{ meters.}$$

**5.4.2.6 CE\_XX Performance Summary**

A detailed performance evaluation/comparison was made for all the above CE\_XX computation methods. Performance corresponds to both calculation error and throughput, and corresponds to non-optimized MATLAB code on a notebook computer. Performance details are presented in Appendix C and summarized below in Table 5.4.2.6-1:

**Table 5.4.2.6-1:** Performance Summary for CE\_XX Calculations

CE_XX											
Method	mean-value		probabilities		ratio $r >$	execution time (s)		rel error  (%)			% conv
	zero	arbitrary	fixed	arbitrary		mean	max	mean	99th perc	max	
Table Interp	yes	no	yes	no	0	0.0001	0.0002	0.01	0.04	0.1	n/a
M.C. Cov Sqrt	yes	yes	yes	yes	0.0001	0.09	0.15	0.05	0.35	0.6	n/a
Integral Eqn	yes	no	yes	yes	0.02	0.08	1.3	0.005	0.02	0.09	100
Integral Eqn	no	yes	yes	yes	0.05	0.1	1.5	0.005	0.02	0.09*	99.9

Thus, for example, given that the mean-value for the magnitude of relative error for the Monte Carlo covariance matrix square root approach is 0.05% per the above table, if the true CE\_XX equals 5 meters, we would expect the computed value to be the correct value 5.0 +/- 0.0025 meters. Alternatively, given that the 99<sup>th</sup> percentile computation error is 0.35% per the above table, there is an approximate 99% probability that the computed value will be within the interval 5.0 +/- 0.0175 meters.

Note that, as discussed earlier, the Integral Equation approach for an arbitrary probability level and for an arbitrary mean-value can have convergence problems for a small percentage of cases (approximately 0.1% or 1:1000). However, they are almost always detectable as discussed in Appendix C. Also, as indicated by the asterisk in some of the above “max absolute relative error” cell entries, these values can increase up to a value of 40% when convergence is not achieved.

Also, although the Monte Carlo covariance matrix square root approach has somewhat larger calculation error than does the Integral Equation approach, it is still reasonable for an operational environment and it has no extreme calculation times as does the Integral Equation approach. It is also applicable to virtually any error covariance matrix since ratio  $r > 0.0001$ , i.e., always “converges”, even if the corresponding error ellipse has a very large difference in lengths of the semi-major and semi-minor axis.

Finally, the results of Table 5.4.2.6-1 are empirical and based on thousands of simulation cases, each case corresponding to an arbitrarily selected (full) 2x2 error covariance matrix, and if applicable, an arbitrary 2x1 mean-value and an arbitrary probability level within the interval [0.1,0.999]. Method vs. different Method direct comparisons were made, as well as repeatability tests for each specific Method. In general, larger magnitude relative errors were achieved when the probability level was somewhat extreme ( $p < 0.15$  or  $p > 0.95$ ) and/or the ratio  $r$  very small. These characteristics also contributed to infrequent integral equation convergence problems when the mean-value was not zero.

### 5.4.3 Spherical Error (SE)

The definition and derivations/computation for SE is similar to that described above for CE, but extended from two dimensions to three dimensions.

In particular, SE\_XX is defined as that spherical radius R such that:

$$p = \frac{1}{(2\pi)^{3/2} \det(C_X)^{1/2}} \iiint e^{-1/2((X-\bar{X})^T C_X^{-1} (X-\bar{X}))} dx dy dz, \quad (5.4.3-1)$$

integrated over the region  $\sqrt{x^2 + y^2 + z^2} \leq R$ , and where probability  $p = XX/100$ , 3d error  $\varepsilon X^T = [\varepsilon x \ \varepsilon y \ \varepsilon z]$  is defined as  $X^T = [x \ y \ z]$  for notational convenience, with mean-value  $\bar{\varepsilon X}$  defined as  $\bar{X} = [\bar{x} \ \bar{y} \ \bar{z}]$ , and  $3 \times 3$  error covariance matrix  $C_X$  about the mean. Note that if the mean-value is not zero, the radius  $R$  is still relative to the origin  $[0 \ 0 \ 0]^T$ , per the standard definition of SE\_XX.

If the x-y-z coordinate system is assumed eigenvector aligned and the mean-value transformed to that system, we also have the alternate and equivalent formulation:

$$p = \frac{1}{(2\pi)^{3/2} \sigma_{max} \sigma_{mid} \sigma_{min}} \iiint e^{-1/2\left(\left(\frac{x-\bar{x}}{\sigma_{max}}\right)^2 + \left(\frac{y-\bar{y}}{\sigma_{mid}}\right)^2 + \left(\frac{z-\bar{z}}{\sigma_{min}}\right)^2\right)} dx dy dz, \quad (5.4.3-2)$$

integrated over the region  $\sqrt{x^2 + y^2 + z^2} \leq R$ , where the eigenvalues  $\sigma_{max}^2$ ,  $\sigma_{mid}^2$ , and  $\sigma_{min}^2$  (assumed associated with coordinates x, y, and z, respectively, for notational convenience) are the elements of the diagonal error covariance matrix  $C_X$  expressed in the eigenvector aligned system.

Either of the above equations (5.4.3-1) or (5.4.3-2) can be solved for iteratively for  $R$ , given the desired probability level  $p = XX/100$ . Note that the right side of the corresponding equation for a given iteration is solved for numerically. Thus, we have:

$$SE_{XX} = R. \quad (5.4.3-3)$$

Furthermore, assuming a mean-value of zero and an additional change of variables from the eigenvector-aligned x-y-z system to a scaled eigenvector aligned  $x^*-y^*-z^*$  system corresponding to  $x/\sigma_{max} \rightarrow x^*$ ,  $y/\sigma_{mid} \rightarrow y^*$ , and  $z/\sigma_{min} \rightarrow z^*$ , Equation (5.4.3-2) can also be written as:

$$p = \frac{1}{(2\pi)^{3/2}} \iiint e^{-1/2(x^{*2} + y^{*2} + z^{*2})} dx dy dz, \quad (5.4.3-4)$$

integrated over the region  $\sqrt{x^{*2} + r_1^2 y^{*2} + r_2^2 z^{*2}} \leq R/\sigma_{max}$ , where  $r_1 = \sigma_{mid}/\sigma_{max}$  and  $r_2 = \sigma_{min}/\sigma_{max}$ .

NGA.SIG.0026.03\_1.0\_ACCPRED

If we also assume that  $\sigma_{max} = 1$ , it follows that the value  $R = R(p, r_1, r_2)$ , such that the above integral equals the desired level of probability p, is related to SE\_XX as follows:

$$SE_{XX} = R(p, r_1, r_2)\sigma_{max}, \tag{5.4.3-5}$$

where  $\sigma_{max}$  is the square root of the maximum eigenvalue of  $C_X$ ,  $r_1 = \sigma_{eig\_mid}/\sigma_{eig\_max}$ , and  $r_2 = \sigma_{eig\_min}/\sigma_{eig\_max}$ .

Typically  $R(p, r_1, r_2)$  is pre-computed for all combination of values of  $r_1 = 0, 0.05, 0.1, \dots 0.95, 1.0$  and  $r_2 = 0, 0.05, 0.1, \dots 0.95, 1.0$ , i.e., 21x21=441 values or table entries, and then interpolated appropriately.

A given entry for the table is precomputed by, setting  $\bar{X} = 0$  and  $C_X = \begin{bmatrix} \sigma_{max}^2 & 0 & 0 \\ 0 & \sigma_{mid}^2 & 0 \\ 0 & 0 & \sigma_{min}^2 \end{bmatrix}$  to

$\begin{bmatrix} 1 & 0 & 0 \\ 0 & r_1^2 & 0 \\ 0 & 0 & r_2^2 \end{bmatrix}$  and solving Equation (5.4.3-2) for R. An exception corresponds to values of  $r_1 = 0$  or  $r_2 = 0$ , where appropriate table values correspond to limiting arguments. In particular, if  $r_2 = 0$ , SE\_XX table entries correspond to CE\_XX table entries, i.e., the expected magnitude of error in the third dimension approaches zero or is a “non-entity”.

Tables 5.4.3-1 and 5.4.3-2 are precomputed tables of  $R(p = 0.5, r_1, r_2)$ ,  $R(p = 0.9, r_1, r_2)$ ,  $R(p = 0.95, r_1, r_2)$ ,  $R(p = 0.99, r_1, r_2)$ , and  $R(p = 0.999, r_1, r_2)$ . All table entries are presented although each table is symmetric.

**Table 5.4.3-1: Spherical Error (SE) multiplier  $R(p = 0.5, r_1, r_2)$  versus ratios  $r_1$  and  $r_2$**

		$r_1$																				
		0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$r_2$	0.00	0.6745	0.6763	0.6820	0.6916	0.7059	0.7254	0.7499	0.7779	0.8079	0.8389	0.8704	0.9021	0.9337	0.9651	0.9962	1.0271	1.0577	1.0880	1.1181	1.1479	1.1774
	0.05	0.6763	0.6782	0.6838	0.6934	0.7076	0.7271	0.7516	0.7795	0.8094	0.8404	0.8719	0.9035	0.9350	0.9664	0.9975	1.0283	1.0589	1.0891	1.1192	1.1489	1.1784
	0.10	0.6820	0.6838	0.6894	0.6989	0.7130	0.7324	0.7567	0.7844	0.8141	0.8449	0.8762	0.9077	0.9390	0.9703	1.0013	1.0320	1.0625	1.0926	1.1225	1.1522	1.1817
	0.15	0.6916	0.6934	0.6989	0.7084	0.7223	0.7414	0.7654	0.7927	0.8221	0.8526	0.8836	0.9147	0.9459	0.9768	1.0077	1.0381	1.0684	1.0984	1.1282	1.1578	1.1870
	0.20	0.7059	0.7076	0.7130	0.7223	0.7359	0.7546	0.7781	0.8048	0.8336	0.8636	0.8941	0.9248	0.9556	0.9862	1.0167	1.0469	1.0769	1.1067	1.1362	1.1655	1.1947
	0.25	0.7254	0.7271	0.7324	0.7414	0.7546	0.7727	0.7952	0.8211	0.8491	0.8783	0.9081	0.9382	0.9684	0.9986	1.0286	1.0584	1.0881	1.1174	1.1466	1.1756	1.2045
	0.30	0.7499	0.7516	0.7567	0.7654	0.7781	0.7952	0.8167	0.8414	0.8684	0.8966	0.9256	0.9549	0.9844	1.0140	1.0434	1.0728	1.1019	1.1309	1.1597	1.1883	1.2168
	0.35	0.7779	0.7795	0.7844	0.7927	0.8048	0.8211	0.8414	0.8651	0.8909	0.9181	0.9462	0.9748	1.0035	1.0324	1.0612	1.0899	1.1185	1.1470	1.1753	1.2035	1.2315
	0.40	0.8079	0.8094	0.8141	0.8221	0.8336	0.8491	0.8684	0.8909	0.9157	0.9420	0.9692	0.9970	1.0251	1.0533	1.0814	1.1096	1.1376	1.1656	1.1934	1.2211	1.2488
	0.45	0.8389	0.8404	0.8449	0.8526	0.8636	0.8783	0.8966	0.9181	0.9420	0.9675	0.9939	1.0210	1.0484	1.0760	1.1036	1.1313	1.1588	1.1863	1.2137	1.2409	1.2681
	0.50	0.8704	0.8719	0.8762	0.8836	0.8941	0.9081	0.9256	0.9462	0.9692	0.9939	1.0197	1.0462	1.0730	1.1002	1.1273	1.1545	1.1816	1.2086	1.2356	1.2625	1.2893
	0.55	0.9021	0.9035	0.9077	0.9147	0.9248	0.9382	0.9549	0.9748	0.9970	1.0210	1.0462	1.0722	1.0985	1.1251	1.1519	1.1788	1.2055	1.2322	1.2589	1.2854	1.3119
	0.60	0.9337	0.9350	0.9390	0.9459	0.9556	0.9684	0.9844	1.0035	1.0251	1.0484	1.0730	1.0985	1.1245	1.1508	1.1772	1.2037	1.2302	1.2567	1.2830	1.3093	1.3355
	0.65	0.9651	0.9664	0.9703	0.9768	0.9862	0.9986	1.0140	1.0324	1.0533	1.0760	1.1002	1.1251	1.1508	1.1767	1.2029	1.2291	1.2554	1.2817	1.3078	1.3339	1.3599
	0.70	0.9962	0.9975	1.0013	1.0077	1.0167	1.0286	1.0434	1.0612	1.0814	1.1036	1.1273	1.1519	1.1772	1.2029	1.2288	1.2549	1.2810	1.3070	1.3330	1.3590	1.3848
	0.75	1.0271	1.0283	1.0320	1.0381	1.0469	1.0584	1.0728	1.0899	1.1096	1.1313	1.1545	1.1788	1.2037	1.2291	1.2549	1.2807	1.3067	1.3325	1.3585	1.3843	1.4101
	0.80	1.0577	1.0589	1.0625	1.0684	1.0769	1.0881	1.1019	1.1185	1.1376	1.1588	1.1816	1.2055	1.2302	1.2554	1.2810	1.3067	1.3324	1.3582	1.3840	1.4098	1.4355
	0.85	1.0880	1.0891	1.0926	1.0984	1.1067	1.1174	1.1309	1.1470	1.1656	1.1863	1.2086	1.2322	1.2567	1.2817	1.3070	1.3325	1.3582	1.3840	1.4098	1.4356	1.4611
	0.90	1.1181	1.1192	1.1225	1.1282	1.1362	1.1466	1.1597	1.1753	1.1934	1.2137	1.2356	1.2589	1.2830	1.3078	1.3330	1.3585	1.3840	1.4098	1.4355	1.4612	1.4869
	0.95	1.1479	1.1489	1.1522	1.1578	1.1655	1.1756	1.1883	1.2035	1.2211	1.2409	1.2625	1.2854	1.3093	1.3339	1.3590	1.3843	1.4098	1.4356	1.4612	1.4869	1.5125
	1.00	1.1774	1.1784	1.1817	1.1870	1.1947	1.2045	1.2168	1.2315	1.2488	1.2681	1.2893	1.3119	1.3355	1.3599	1.3848	1.4101	1.4355	1.4611	1.4869	1.5125	1.5382

Table 5.4.3-2: Spherical Error (SE) multiplier  $R(p = 0.9, r_1, r_2)$  versus ratios  $r_1$  and  $r_2$

		$r_1$																				
		0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$r_2$	0.00	1.6449	1.6456	1.6479	1.6518	1.6573	1.6646	1.6738	1.6852	1.6992	1.7163	1.7371	1.7621	1.7915	1.8251	1.8625	1.9034	1.9472	1.9936	2.0424	2.0932	2.1460
	0.05	1.6456	1.6464	1.6487	1.6525	1.6581	1.6654	1.6745	1.6860	1.6999	1.7170	1.7378	1.7628	1.7922	1.8258	1.8632	1.9040	1.9478	1.9942	2.0429	2.0938	2.1466
	0.10	1.6479	1.6487	1.6509	1.6548	1.6604	1.6676	1.6769	1.6882	1.7021	1.7192	1.7400	1.7650	1.7944	1.8279	1.8652	1.9060	1.9497	1.9961	2.0448	2.0956	2.1483
	0.15	1.6518	1.6525	1.6548	1.6587	1.6642	1.6714	1.6806	1.6920	1.7059	1.7229	1.7436	1.7686	1.7979	1.8314	1.8687	1.9094	1.9530	1.9993	2.0479	2.0987	2.1512
	0.20	1.6573	1.6581	1.6604	1.6642	1.6697	1.6769	1.6861	1.6974	1.7113	1.7282	1.7489	1.7738	1.8030	1.8364	1.8735	1.9141	1.9576	2.0039	2.0523	2.1029	2.1555
	0.25	1.6646	1.6654	1.6676	1.6714	1.6769	1.6841	1.6932	1.7045	1.7183	1.7352	1.7558	1.7806	1.8097	1.8429	1.8799	1.9204	1.9638	2.0098	2.0581	2.1086	2.1610
	0.30	1.6738	1.6745	1.6769	1.6806	1.6861	1.6932	1.7023	1.7135	1.7273	1.7441	1.7646	1.7892	1.8182	1.8513	1.8881	1.9283	1.9715	2.0173	2.0654	2.1156	2.1678
	0.35	1.6852	1.6860	1.6882	1.6920	1.6974	1.7045	1.7135	1.7247	1.7383	1.7550	1.7755	1.7999	1.8286	1.8614	1.8981	1.9380	1.9809	2.0265	2.0743	2.1243	2.1762
	0.40	1.6992	1.6999	1.7021	1.7059	1.7113	1.7183	1.7273	1.7383	1.7519	1.7685	1.7887	1.8130	1.8414	1.8740	1.9102	1.9498	1.9923	2.0375	2.0850	2.1347	2.1862
	0.45	1.7163	1.7170	1.7192	1.7229	1.7282	1.7352	1.7441	1.7550	1.7685	1.7849	1.8049	1.8289	1.8569	1.8890	1.9248	1.9639	2.0060	2.0506	2.0977	2.1469	2.1981
	0.50	1.7371	1.7378	1.7400	1.7436	1.7489	1.7558	1.7646	1.7755	1.7887	1.8049	1.8245	1.8481	1.8757	1.9071	1.9422	1.9807	2.0221	2.0663	2.1127	2.1614	2.2120
	0.55	1.7621	1.7628	1.7650	1.7686	1.7738	1.7806	1.7892	1.7999	1.8130	1.8289	1.8481	1.8710	1.8979	1.9287	1.9630	2.0007	2.0413	2.0847	2.1304	2.1783	2.2282
	0.60	1.7915	1.7922	1.7944	1.7979	1.8030	1.8097	1.8182	1.8286	1.8414	1.8569	1.8757	1.8979	1.9240	1.9539	1.9873	2.0240	2.0637	2.1061	2.1510	2.1980	2.2472
	0.65	1.8251	1.8258	1.8279	1.8314	1.8364	1.8429	1.8513	1.8614	1.8740	1.8890	1.9071	1.9287	1.9539	1.9827	2.0151	2.0507	2.0894	2.1308	2.1746	2.2207	2.2689
	0.70	1.8625	1.8632	1.8652	1.8687	1.8735	1.8799	1.8881	1.8981	1.9102	1.9248	1.9422	1.9630	1.9873	2.0151	2.0464	2.0809	2.1185	2.1587	2.2015	2.2464	2.2936
	0.75	1.9034	1.9040	1.9060	1.9094	1.9141	1.9204	1.9283	1.9380	1.9498	1.9639	1.9807	2.0007	2.0240	2.0507	2.0809	2.1143	2.1506	2.1898	2.2314	2.2753	2.3214
	0.80	1.9472	1.9478	1.9497	1.9530	1.9576	1.9638	1.9715	1.9809	1.9923	2.0060	2.0221	2.0413	2.0637	2.0894	2.1185	2.1506	2.1858	2.2237	2.2642	2.3070	2.3520
	0.85	1.9936	1.9942	1.9961	1.9993	2.0039	2.0098	2.0173	2.0265	2.0375	2.0506	2.0663	2.0847	2.1061	2.1308	2.1587	2.1898	2.2237	2.2605	2.2998	2.3415	2.3854
	0.90	2.0424	2.0429	2.0448	2.0479	2.0523	2.0581	2.0654	2.0743	2.0850	2.0977	2.1127	2.1304	2.1510	2.1746	2.2015	2.2314	2.2642	2.2998	2.3380	2.3786	2.4213
	0.95	2.0932	2.0938	2.0956	2.0987	2.1029	2.1086	2.1156	2.1243	2.1347	2.1469	2.1614	2.1783	2.1980	2.2207	2.2464	2.2753	2.3070	2.3415	2.3786	2.4180	2.4597
	1.00	2.1460	2.1466	2.1483	2.1512	2.1555	2.1610	2.1678	2.1762	2.1862	2.1981	2.2120	2.2282	2.2472	2.2689	2.2936	2.3214	2.3520	2.3854	2.4213	2.4597	2.5003

Tables 5.4.3-3 – 5.4.3-5 were omitted to save space. All five tables are in Appendix C, as well as in a format suitable for copying, assuming this document is a suitable digital file (e.g., word document).

**5.4.3.1 Baseline Computation Method: Table Interpolation**

Thus, the algorithm for computation of  $SE_{XX}$ , given the usual assumption of a mean error equal to zero, and corresponding to a (full) 3x3 error covariance matrix  $C_X$  expressed relative to an arbitrary Cartesian coordinate system is as follows:

(1) Compute the eigenvalues of  $C_X$ :  $\sigma_{eig,max}^2, \sigma_{eig,mid}^2, \sigma_{eig,min}^2$  (5.4.3.1-1)  
 assumed in descending order

(2) Compute  $\sigma_{eig,max}, r_1 = \sigma_{eig,mid}/\sigma_{eig,max}, r_2 = \sigma_{eig,min}/\sigma_{eig,max}$

(3) Based on the desired probability level XX (%) and the computed values  $r_1, r_2$ , perform bilinear interpolation of the appropriate table  $R(p = XX/100, r_1, r_2)$  of Section 5.4.3 for a normalized radius value  $R^*$

(4)  $SE_{XX} = R^* \sigma_{eig,max}$

**5.4.3.2 Alternate Computation Method: Monte Carlo Matrix Square Root**

The following alternate approach to the computation of SE\_XX is applicable to arbitrary mean-values and arbitrary probability levels, is computationally accurate and reasonably fast:

(1) Compute 1E6 independent samples of 3d error:  $s_i = \bar{X} + C_X^{1/2} n_i, i = 1, \dots, 1E6,$  (5.4.3.2-1)

where  $\bar{X}$  and  $C_X$  are the 3x1 mean and the 3x3 error covariance about the mean relative to the original (non-eigenvector aligned) coordinate system,  $n_i$  is a three-element vector with each element the realization of an independent  $N(0,1)$  random variable, and where the superscript “1/2” indicates principal matrix square root.  $\bar{X}$  and  $n_i$  are 3x1 vectors, and  $C_X^{1/2}$  is a 3x3 matrix.

(2) Order the magnitudes of the 3d error samples from smallest to largest, and designate  $RE_{XX}$  the XX\_th percent largest, and  $RE_{XX}^*$  the next largest magnitude.

(3)  $SE_{XX} = (RE_{XX} + RE_{XX}^*) / 2$ .

Note that the symmetric  $C_X^{1/2}$  is computed once prior to generating the independent samples, and the samples  $C_X^{1/2} n_i$  are consistent with the error covariance matrix about the mean, i.e.,

$$E\{(s_i - \bar{X})(s_i - \bar{X})^T\} = E\{C_X^{1/2} n_i (C_X^{1/2} n_i)^T\} = C_X^{1/2} E\{n_i n_i^T\} C_X^{1/2} = C_X^{1/2} I_{3 \times 3} C_X^{1/2} = C_X,$$

where  $E\{\}$  is the expected value operator.

Alternatively, the above can be performed in an equivalent manner relative to the eigenvector-aligned system by computing samples as  $s_i = (\Phi \bar{X} + D \cdot n_i)$ , where “ $\cdot$ ” is the vector dot product,  $D$  a 3x1 vector containing the square-root of the eigenvalues, and  $\Phi$  the 3x3 transformation matrix from the original Cartesian coordinate system to the eigenvector-aligned coordinate system. The speed varies little between the two approaches.

Due to its use of 1E6 random samples, the computational accuracy of the above algorithm is directly associated with statistical significance, and resultant computational error is expected to be on the order of  $1/\sqrt{1E6}$ , or a 0.1% relative error. This assumes reasonable and practical probability values that are within the interval [0.1, 0.999], and that the square-root of the smallest to the largest eigenvalue  $r > 0.0001$ , i.e., applicable to virtually any valid error covariance matrix.

**5.4.3.3 SE\_XX Computation Method Selection**

Pseudo-code (MATLAB) for the computation of SE\_XX is presented in Appendix C corresponding to the following operationally, priority-ordered methods/equations:

- (1) Baseline table interpolation (mean-value zero,  $p=0.5, 0.9, 0.95, 0.99, \text{ or } 0.999$ ), see Equation (5.4.3.1-1).
- (2) Monte Carlo Matrix Square Root (arbitrary mean-value and probability level), see Equation (5.4.3.2-1).
- (3) Integral Equation (mean-value zero, arbitrary probability level), see Equation (5.4.3-2).

Operationally, Method (1) is the preferred method for the fixed levels of probability previously specified and when the mean-value of predictive error zero, which is almost always the case. It has small calculation error and is much faster than all other methods.

Operationally, Method (2) is preferred over Method (3) due to no extreme variations in calculation time, although the latter has less calculation error. Method (3) is the preferred method for the generation of new interpolation tables corresponding to probability levels not already fixed.

Note that Method (3)'s Equation (5.4.3-2) can also be used when the mean-value is not zero, but it does not converge as fast as when the mean-value equals zero, or not at all a small percentage of the time; hence, it was left out of the above list, although pseudo-code is also available in Appendix C. The convergence issue and how to detect it is also discussed further in Appendix C.

In general, when a non-zero mean is added, the solution to Equation (5.4.3-2) becomes more difficult and time-consuming. The solution involves an iterative search over numerical evaluation of the integral such that the resultant probability is near the specified amount on the left side of the equation. This becomes more difficult because the candidate SE circle is centered about the origin, not the mean-value about which the error covariance is "centered". Further note that in order to improve convergence and throughput corresponding to Equation (5.4.3-2), the square-root of the smallest to largest eigenvalue ratio is assumed  $r \geq 0.02$  if the mean-value is zero, otherwise  $r \geq 0.10$ .

Table 5.4.3.5-1 of Section 5.4.3.5 presents a performance summary for all methods

#### 5.4.3.4 Examples of SE<sub>XX</sub> computation

Examples are as follows:

(1) Assume a desired probability level of 90%, a mean error of zero, and  $C_X = \begin{bmatrix} 4 & -5.4 & 6 \\ -5.4 & 9 & -9 \\ 6 & -9 & 25 \end{bmatrix}$  meters-squared. Thus, baseline table interpolation (Equation (5.4.3.1-1)) is applicable and is the first choice:

Eigenvalues equal 31.2, 6.22, and 0.55 meters-squared

$$\sigma_{eig\_max} = 5.59 \text{ meters}, r_1 = 0.446, r_2 = 0.132$$

$$R^* = 1.72 \text{ ( via bilinear linear interpolation: } \frac{0.018}{0.05} \left( \frac{0.004}{0.05} 1.7021 + \frac{0.046}{0.05} 1.7192 \right) +$$

$$\frac{0.032}{0.05} \left( \frac{0.004}{0.05} 1.7059 + \frac{0.046}{0.05} 1.7229 \right) = 1.7202 )$$

$$SE_{90} = R^* \sigma_{eig\_max} = 9.61 \text{ meters.}$$

(2) Assume a desired probability level of 90%, a mean-value  $\bar{X}^T = [1 \ 0 \ -1]$ , and  $C_X = \begin{bmatrix} 4 & -5.4 & 6 \\ -5.4 & 9 & -9 \\ 6 & -9 & 25 \end{bmatrix}$  meters-squared. Thus, since the mean-value is not zero, the Monte-Carlo Matrix Square Root method (Equation (5.4.3.2-1)) is applicable:

$$SE_{90} = 9.76 \text{ meters.}$$

(3) Assume a desired probability level of 70%, and the lowest possible calculation error,

a mean error of zero, and  $C_X = \begin{bmatrix} 4 & -5.4 & 6 \\ -5.4 & 9 & -9 \\ 6 & -9 & 25 \end{bmatrix}$  meters-squared. Thus, since  $p=0.7$  is not one of the fixed probability levels previously specified, the Integral Equation with a mean-value of zero (Equation (5.4.3-2)) is applicable:

$$SE_{70} = 6.47 \text{ meters.}$$

#### 5.4.3.5 SE\_XX Performance Summary

A detailed performance evaluation/comparison was made for all the above SE\_XX computation methods. Performance corresponds to both calculation error and throughput, and corresponds to non-optimized MATLAB code on a notebook computer. Performance details are presented in Appendix C and summarized below:

**Table 5.4.3.5-1:** Performance Summary for SE\_XX Calculations

SE_XX											
Method	mean-value		probabilities		ratio r >	execution time (s)		rel error  (%)			% conv
	zero	arbitrary	fixed	arbitrary		mean	max	mean	99th perc	max	
Table Interp	yes	no	yes	no	0	0.0002	0.0005	0.02	0.05	0.15	n/a
M.C. Cov Sqrt	yes	yes	yes	yes	0.0001	0.08	0.2	0.05	0.2	0.4	n/a
Integral Eqn	yes	no	yes	yes	0.02	8	20	0.001	0.003	0.011	100
Integral Eqn	no	yes	yes	yes	0.1	9	48	0.001	0.003	0.011*	99.8

Thus, for example, given that the maximum for the magnitude of relative error for the Monte Carlo covariance matrix square root approach is 0.4% per the above table, if the true CE\_XX equals 5 meters, we would expect the computed value to have a value within 5.0 +/- 0.02 meters virtually all of the time.

Note that, as discussed earlier, the Integral Equation approach for an arbitrary probability level and for an arbitrary mean-value can have convergence problems for some cases (approximately 0.2% or 1:500). However, they are almost always detectable as discussed in Appendix C. Also, as indicated by the asterisk in some of the above “max absolute relative error” cell entries, these values can increase up to a value of 40% when convergence is not achieved.

Also, although the Monte Carlo covariance matrix square root approach has somewhat larger calculation error than does the Integral Equation approach, it is still reasonable for an operational environment and

has no large average or maximum calculation times as does the Integral Equation approach. It is also applicable to virtually any error covariance matrix since ratio  $r > 0.0001$ .

Finally, the results of Table 5.4.3.5-1 are empirical and based on thousands of simulation cases, each case corresponding to an arbitrarily selected (full) 3x3 error covariance matrix, and if applicable, an arbitrary 3x1 mean-value and an arbitrary probability level within the interval [0.1,0.999]. Method vs. different Method direct comparisons were made, as well as repeatability tests for each specific Method. In general, larger magnitude relative errors were achieved when the probability level was somewhat extreme ( $p < 0.15$  or  $p > 0.95$ ) and/or the ratio  $r$  very small. These characteristics also contributed to infrequent integral equation convergence problems when the mean-value was not zero.

### 5.5 Multi-state vector Error Covariance Matrix Definition

In many situations, it is more convenient and provides more insight to generalize the error covariance matrix relative to a single state vector to a multi-state vector. For example, a multi-state vector consisting of a collection of individual state vectors, each corresponding to the same stochastic process but at different times. Or perhaps, corresponding to a collection of individual state vectors, all solved for (estimated) simultaneously in one large state vector as part of a batch estimation process. The general case is detailed as follows:

Let  $X_i$  be an  $n_i \times 1$  individual state vector  $i$ . Let the  $n_i \times 1$  random error vector  $\varepsilon X_i$  represent its corresponding error. (Recall that the  $n_i$  components of  $\varepsilon X_i$  are random variables.)

Let  $X = [X_1^T \ \dots \ X_m^T]^T$  be the “stacked”  $n \times 1$  multi-state vector corresponding to the  $m$  individual state vectors, and let  $\varepsilon X = [\varepsilon X_1^T \ \dots \ \varepsilon X_m^T]^T$  represent its corresponding error of the same  $n \times 1$  dimension, where  $n = \sum_{i=1}^m n_i$  and the superscript  $T$  indicates transpose. Let  $C_X$  represent the corresponding  $n \times n$  multi-state vector (symmetric) error covariance matrix:

$$C_X = E\{\varepsilon X \varepsilon X^T\} = E \left\{ \begin{matrix} \varepsilon X_1 \varepsilon X_1^T & \varepsilon X_1 \varepsilon X_2^T & \dots & \varepsilon X_1 \varepsilon X_m^T \\ \varepsilon X_2 \varepsilon X_1^T & \varepsilon X_2 \varepsilon X_2^T & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \varepsilon X_m \varepsilon X_1^T & \varepsilon X_m \varepsilon X_2^T & \dots & \varepsilon X_m \varepsilon X_m^T \end{matrix} \right\} = \begin{bmatrix} C_{X1} & C_{X12} & \dots & C_{X1m} \\ \cdot & C_{X2} & \dots & C_{X2m} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & C_{Xm} \end{bmatrix}. \quad (5.5-1)$$

Note that  $C_{X_i}$  is the  $n_i \times n_i$  error covariance matrix for state vector  $i$ ;  $C_{X_{ik}}$  the  $n_i \times n_k$  error cross-covariance matrix between state vectors  $i$  and  $k$ , and  $E$  is the expected-value operator. The  $\varepsilon X_i$  are random vectors, and the error covariance matrices are descriptive statistics based on assumed underlying probability distributions (not sample statistics). The single dots “ $\cdot$ ” in Equation (5.5-1) indicate symmetric entries (e.g.,  $C_{X_{21}} = C_{X_{12}}^T$ ), and the double dots “ $\dots$ ” indicate “continue the pattern”.  $C_X$  is a symmetric, positive definite matrix (strictly positive eigenvalues), i.e., invertible and a “valid” error covariance matrix.

Equation (5.5-1) assumes errors have a mean-value of zero, as is typically the case for predictive errors; otherwise, the mean-value  $\overline{\varepsilon X_i} \neq 0$  for arbitrary state vector  $i$ , and:

$$C_{X_i} = E\{(\epsilon X_i - \overline{\epsilon X_i})(\epsilon X_i - \overline{\epsilon X_i})^T\} \text{ and } C_{X_{ik}} = E\{(\epsilon X_i - \overline{\epsilon X_i})(\epsilon X_k - \overline{\epsilon X_k})^T\}. \quad (5.5-2)$$

For most applications of interest, the dimensions of  $X_i$  and  $X_k$  are the same, and hence, the dimensions of  $C_{X_i}$  and  $C_{X_{ik}}$ ,  $1 \leq i, k \leq m$ , are the same. In addition, the identity of the components that make up  $X_i$  and  $X_k$  are the same as well; for example, if state vector  $i$  corresponds to ground point  $i$  and  $X_i$  its  $x, y, z$  ground coordinates, state vector  $j$  corresponds to ground point  $j$  and  $X_j$  its  $x, y, z$  ground coordinates in the same coordinate system. This assumption is applicable for the remainder of this guidance document, except where explicitly noted otherwise.

(Note that whether  $C_X$  corresponds to the error in a multi-state or “stacked” state vector or simply the error in one (original) state vector, if not specified specifically in remaining sections of this document, it does not matter. In addition, the dimension of the state vector is generically assumed to be  $n \times 1$  if not specified otherwise.)

### 5.5.1 Details regarding a single state vector

The error covariance matrix for state vector  $i$  contains a statistical measure of the errors in state vector  $X_i$ . More specifically, let the  $n_i \times 1$  state error vector equal:

$$\epsilon X_i = [\epsilon x_{1i} \quad \epsilon x_{2i} \quad \dots \quad \epsilon x_{n_i}]^T. \quad (5.5.1-1)$$

Its corresponding (symmetric)  $n_i \times n_i$  error covariance matrix equals:

$$C_{X_i} = \begin{bmatrix} \sigma_{1i}^2 & \sigma_{1i2i} & \dots & \sigma_{1in_i} \\ \sigma_{2i1i} & \sigma_{2i}^2 & \dots & \sigma_{2in_i} \\ \dots & \dots & \dots & \dots \\ \sigma_{n_i1i} & \sigma_{n_i2i} & \dots & \sigma_{n_i}^2 \end{bmatrix} = \begin{bmatrix} \sigma_{1i}^2 & \sigma_{1i2i} & \dots & \sigma_{1in_i} \\ \cdot & \sigma_{2i}^2 & \dots & \sigma_{2in_i} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \sigma_{n_i}^2 \end{bmatrix}, \text{ and} \quad (5.5.1-2)$$

where, for example, the variance for first-component of state vector  $i$ 's error is  $\sigma_{1i}^2$ , with corresponding standard deviation  $\sigma_{1i}$ . (Note that, as mentioned earlier, all errors are assumed to have a mean value of zero, i.e., the corresponding estimate is unbiased.) The covariance (not the “covariance matrix”) between the first and second component errors, both corresponding to state vector  $i$ , is  $\sigma_{1i2i}$ . The covariance specifies the intra-state vector correlation (statistical similarity) between the first and second component errors. The corresponding intra-state vector correlation coefficient is defined as:

$$-1 < \rho_{1i2i} = \frac{\sigma_{1i2i}}{\sigma_{1i}\sigma_{2i}} < 1. \quad (5.5.1-3)$$

Thus, the covariance can also be written as  $\rho_{1i2i}\sigma_{1i}\sigma_{2i}$ . Also, for two components with the same identities, the intra-state vector correlation coefficient is defined as equal to 1.0, e.g.,  $\frac{\sigma_{1i}^2}{\sigma_{1i}\sigma_{1i}} = 1$ .

### 5.5.2 Details regarding two different state vectors

The  $n_i \times n_k$  (in general, non-symmetric) cross-covariance matrix corresponding to state vector  $i$  and state vector  $k$  specifies the inter-state vector correlation (statistical similarity) between the various error components corresponding to the two different state vectors:

$$\epsilon X_i = [\epsilon X_{1i} \quad \epsilon X_{2i} \quad \dots \quad \epsilon X_{n_i}]^T \quad \epsilon X_k = [\epsilon X_{1k} \quad \epsilon X_{2k} \quad \dots \quad \epsilon X_{n_k}]^T. \quad (5.5.2-1)$$

$$C_{Xik} = \begin{bmatrix} \sigma_{1i1k} & \sigma_{1i2k} & \dots & \sigma_{1in_k} \\ \sigma_{2i1k} & \sigma_{2i2k} & \dots & \sigma_{2in_k} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \sigma_{n_in_k} \end{bmatrix}, \text{ and} \quad (5.5.2-2)$$

where, for example, the corresponding inter-state vector correlation coefficient between the first component error for state vector  $i$  and the second component error for state vector  $k$  is defined as:

$$-1 < \rho_{1i2k} = \frac{\sigma_{1i2k}}{\sigma_{1i}\sigma_{2k}} < 1. \quad (5.5.2-3)$$

When the two components have the same identities, and the inter-state vector correlation coefficient is written as a function of “delta” between state vector applicabilities (e.g. delta time for state vectors from a stochastic process), the function is typically termed an “auto-correlation function”; when they have different identities, a “cross-correlation function”.

### 5.5.3 Applicability of definitions

Although the detailed symbology in Section 5.5.1 and 5.5.2 is unavoidably somewhat complicated, most of the following sections of this document don’t use symbology any “deeper” or complicated than that corresponding to Equation (5.5-1).

Also, sometimes it is convenient to equate individual state vectors with individual “events”, such as individual “collection” times corresponding to a stochastic process. Correspondingly, intra-state vector correlation and inter-state vector correlation are sometimes termed intra-event correlation and inter-event correlation, respectively.

Finally, when convenient, a multi-state vector and its error covariance matrix can also always be considered a (large) single-state vector and error covariance matrix without corresponding detail into individual state vectors. See Equation (5.1.1-4) for definitions and symbology for an assumed (total) state vector dimension of  $n \times 1$ .

### 5.5.4 Generation of the Relative Error Covariance Matrix

Relative error is defined between a pair of state vectors  $i$  and  $k$  as follows:

$$\epsilon X_{ik} \equiv \epsilon X_i - \epsilon X_k \quad (5.5.4-1)$$

Its corresponding relative error covariance matrix, assuming mean-values of zero, is equal to:

$$relC_{Xik} = E\{\epsilon X_{ik} \epsilon X_{ik}^T\} = E\{(\epsilon X_i - \epsilon X_k)(\epsilon X_i - \epsilon X_k)^T\} = \quad (5.5.4.2)$$

$$E\{\epsilon X_i \epsilon X_i^T\} - E\{\epsilon X_i \epsilon X_k^T\} - E\{\epsilon X_k \epsilon X_i^T\} + E\{\epsilon X_k \epsilon X_k^T\} = C_{Xi} - C_{Xik} - C_{Xik}^T + C_{Xk}.$$

Note that if the state vectors  $i$  and  $j$  correspond to 3d locations, the relative error covariance matrix is a  $3 \times 3$  matrix. And in many corresponding applications, errors between the two state vector components are positively correlated; hence,  $C_{Xi} > relC_{Xik}$  and  $C_{Xj} > relC_{Xik}$  (see Section 5.3.5), i.e., the expected magnitude of the  $3 \times 1$  error in the relative difference (“distance”) between the two state vectors is smaller than either’s individual error. This is due to statistical “cancellation” of common errors.

Note that once the relative error covariance matrix is computed (and is positive definite, as expected), a corresponding relative error ellipsoid as well as relative accuracy summaries  $rel\_LE$ ,  $rel\_CE$ , and  $rel\_SE$  can be computed using exactly the same techniques as detailed in Sections 5.3 and 5.4, respectively - simply substitute  $relC_{Xik}$  for  $C_X$ .

## 5.6 Propagation of Multi-State Vector Error Covariance Matrices

In many applications of interest, either a random error vector corresponding to a single stand-alone state vector  $X$ , or a random error vector corresponding to a state vector  $X_i$  within a multi-state vector  $X$ , are “propagated”, i.e., mapped or projected to a random error vector corresponding to a related state vector  $X'$ . The mapping is typically linear between the random error vectors, where it and its properties are as described in Equation (5.2.2-1). In particular, if we assume an  $n \times 1$  mean-zero random error  $\epsilon X$  being mapped to a mean-zero  $m \times 1$  random error  $\epsilon X'$  via an  $m \times n$  matrix  $\Omega$ :

$$\epsilon X' = \Omega \epsilon X \text{ and } C_{X'} \equiv E\{(\epsilon X')(\epsilon X')^T\} = \Omega C_X \Omega^T, \quad (5.6-1)$$

where  $C_{X'}$  is positive definite if  $\Omega$  is full rank (linearly independent rows or columns), and positive semi-definite, if not.

### 5.6.1 Error Covariance Representation in Different Coordinate Systems

One application of the above is the transformation of an error covariance matrix from one coordinate system to another. For example, assume that a MIG solution for geolocation  $X$  and its error covariance matrix  $C_X$  are with respect to an ECF coordinate system, as is common. However, we want to express the corresponding accuracy prediction (error covariance matrix) in an ENU coordinate system, a common and recommended practice as well.

Let  $\epsilon X$  and  $\epsilon X'$  represent the (unknown)  $3 \times 1$  error in ECF and ENU, respectively. Let  $C_X$  and  $C_{X'}$  represent the  $3 \times 3$  error covariance matrix in ECF and ENU, respectively. Let  $\Omega$  represent the  $3 \times 3$  (full rank) ECF-to-ENU transformation matrix, an orthonormal rotation matrix. The mean-value of  $\epsilon X$  is zero because the MIG estimate is unbiased; hence, the mean-value of  $\epsilon X'$  is zero as well:

$$\overline{\epsilon X'} = E\{\epsilon X'\} = E\{\Omega \epsilon X\} = \Omega E\{\epsilon X\} = \Omega \overline{\epsilon X} = 0. \quad (5.6.1-1)$$

And per Equation (5.6.1) and Equation (5.2.2-1), the solution error covariance matrix applicable to an ENU representation of error is a valid error covariance matrix and equal to:

$$C_{X'} \equiv E\{(\epsilon X')(\epsilon X')^T\} = E\{(\Omega \epsilon X)(\Omega \epsilon X)^T\} = \Omega C_X \Omega^T \quad (5.6.1-2)$$

(Note that the actual mapping of geolocation coordinates (not their error) from ECF to ENU can be expressed as  $X' = \Omega(X - X_0) = \Omega X - \Omega X_0$ , where  $X_0$  is the fixed origin of the ENU (local tangent plane) coordinate system expressed in the ECF coordinate system. The term  $\Omega X_0$  has no effect on the mapping of errors or their covariance matrix from the ECF to ENU coordinate system since it is a deterministic constant.)

Many of the general mappings  $\Omega$  of Equation (5.6-1) are based on a first-order Taylor Series expansion, the mathematical derivation detailed in Section 5.6.2.

### 5.6.2 First-order Taylor Series Expansion

Let the  $m \times 1$  state vector  $X'$  be a function of the  $n \times 1$  state vector  $X$ , indicated as  $X' = X'(X)$ .

Perform a first-order Taylor Series expansion about the  $n \times 1$  operating point  $X_0$  using appropriate first-order (vector) partial derivatives:

$$X'(X) = X'(X_0) + \frac{\partial X'(X_0)}{\partial X} (X - X_0) + \text{higher\_order\_terms} \quad (5.6.2-1)$$

$$\epsilon X' \equiv X'(X) - X'(X_0) = \frac{\partial X'(X_0)}{\partial X} (X - X_0) + \text{higher\_order\_terms}$$

$$\epsilon X' \equiv \frac{\partial X'(X_0)}{\partial X} \epsilon X + \text{higher\_order\_terms}$$

$$\epsilon X' \equiv \Omega \epsilon X + \text{higher\_order\_terms}$$

$$\epsilon X' \cong \Omega \epsilon X, \quad (5.6.2-2)$$

and where  $m \times m$   $C_{X'} = E\{(\Omega \epsilon X)(\Omega \epsilon X)^T\} = \Omega C_X \Omega^T$ , and  $m \times n$   $\Omega = \frac{\partial X'(X_0)}{\partial X}$ .

Note that the above process or “linearization” is also the basis for underlying equations in many optimal estimators (see Section 5.8.1 of TGD 1 and TGD 2d in general).

### 5.6.3 Other propagations

There are other propagations that correspond to higher-order Taylor Series expansions as well as Monte Carlo statistical methods. These typically correspond to various classes of estimators, and are discussed somewhat further in Section 5.11 of TGD 1 and TGD 2d in general.

## 5.7 Generic Methods for Generation of the Multi-State Vector Error Covariance Matrix

A valid multi-state vector error covariance matrix  $C_X$  can be generated by essentially three general methods/processes: (1) explicit *a priori* modeling (2) WLS batch estimator, or (3) Kalman filter (or smoother) estimator. For all three methods, the corresponding mean-value of error is typically assumed zero. For the last two methods, this corresponds to unbiased estimators, as typically assumed. These three methods are presented in Sections 5.7.1, 5.7.2, and 5.7.3, respectively.

### 5.7.1 *A priori* modeling

*A priori* modeling typically specifies  $C_X$ , or sub-blocks or parameters that can generate  $C_X$ , for a multi-variate state vector  $X$  that will be available (later) during normal operations.

For example,  $X$  could be a measurement vector that is provided via the manual or automatic measurement of the pixel locations of a set of ground points in a set of images.  $C_X$  is computed for the  $X$  in order to specify its “uncertainty”, or more correctly, its predicted accuracy. Its computation could be based solely on *a priori* modeling from empirical information of past measurement performance, or it can be augmented by the feedback of internal performance metrics from the automatic correlator. For this example,  $C_X$  typically has zero inter-state vector correlation, i.e., measurement errors associated with identifying and measuring pixel locations are usually modeled as uncorrelated between points. (This particular example did not include the effects of sensor support data errors on the measurements, only “mensuration” or direct measurement error.)

As a second example,  $X$  could consist of *a priori* estimates of adjustable parameters  $X_i$  for sensor support data that are about to be adjusted (corrected) simultaneously for  $m$  images in a WLS batch adjustment. (Or more generally, adjustable parameters for  $k$  different sensors, each with  $m_k$  measurements of common object(s) of interest.) The multi-state vector error covariance matrix  $C_X$  is computed for the  $X$  in order to specify its (pre-adjustment) predicted accuracy. It also places “statistical constraints” on the size of the upcoming WLS corrections.

In general, the computation of  $C_X$  is based on the combination of empirical information of past performance as well as system design. For the above example,  $C_X$  typically has significant inter-state vector correlation, which corresponds to an *a priori* model of sensor support data errors as a multi-variate stochastic process.

In general, empirical information used to model predictive statistics (e.g.,  $C_X$ ) can include sample statistics of measured errors relative to “ground-truth” – see Section 5.7.1.2 for further discussion. Regardless the type of empirical information, the following “mechanics” for the actual specification/generation of  $C_X$  are typical:

*A priori* modeling specifies the individual  $C_{X_i}$ , and in some applications, this error covariance may be constant over all  $i = 1, \dots, m$ . The internal structure (elements) of  $C_{X_i}$  specify the intra-state vector correlations. For example, if  $C_{X_i}$  is a diagonal matrix, intra-state vector correlations are zero.

Typically, the cross-covariance  $C_{Xik}$  are not provided directly by the *a priori* model, but are assembled using an *a priori* strictly positive definite correlation function (spdcf),  $\rho(\delta t_{ik})$ , to specify inter-state vector correlation, i.e.,

$$C_{Xik} = \rho(\delta t_{ik}) \cdot C_{Xi}^{1/2} C_{Xk}^{1/2}, \quad (5.7.1-1)$$

where the superscript 1/2 indicates matrix principal square-root, as further detailed in Section 5.8.3 of this document. This form of generation is practical, relatively simple, and ensures a valid  $C_X$ .

Once the appropriate  $C_{Xij}$  are assembled, they are combined with the various  $C_{Xi}$  to form the *a priori*  $C_X$  to go with the provided  $X$ .

Note that this particular form of modeling is termed the “spdcf method”. It is general enough to model errors as stochastic processes or as random fields (see TGD 1). These can be stationary stochastic processes or a subclass of non-stationary stochastic processes, where the  $C_{Xi}$  are allowed to vary over  $i$  (time) but the spdcf remains the same (same concepts for a random field are also applicable).

### 5.7.1.1 Gauss-Markov as an underlying error model

A time sequence of sensor *a priori* metadata errors, as well as many other error processes, may be reasonably modeled as a first order, mean-zero, Gauss-Markov stochastic process. In particular, assuming a scalar error (e.g., sensor position x-component error as a function of time or index  $i$ ) and equal time steps for convenience:

$$\epsilon X_{i+1} = a_x \epsilon X_i + \omega_i, \text{ where } a_x = e^{-\Delta t/T_x}, E\{\epsilon X_i \epsilon X_k\} = e^{-|i-k|\Delta t/T_x} \sigma_x^2, \text{ and} \quad (5.7.1.1-1)$$

$$E\{\omega_i \omega_k\} = (1 - a_x^2) \sigma_x^2 \delta_{ik}.$$

In the above,  $\sigma_x$  and  $T_x$  are the specifiable (predictive) standard deviation (one-sigma) and temporal correlation time constant for the stochastic process  $x_i$ , respectively,  $E\{\}$  the expected value operator,  $\delta_{ik}$  the kronecker delta, and  $\omega_k$  Gaussian white noise with a corresponding standard deviation that is a function of  $\sigma_x$  and  $T_x$ .

Generalizing to three components contained in the multi-variate stochastic process  $X_i$ , assuming “steady-state” operations, and a discrete-time system:

$$\epsilon X_{i+1} = \Phi \epsilon X_i + W_i, \text{ where } \Phi = \begin{bmatrix} a_x & 0 & 0 \\ 0 & a_y & 0 \\ 0 & 0 & a_z \end{bmatrix}, E\{\epsilon X_i \epsilon X_i^T\} = C_{Xi} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \quad (5.7.1.1-2)$$

$$E\{\epsilon X_m \epsilon X_n^T\} = C_{Xik} = \Phi^{|i-k|} C_{Xi}, \text{ and}$$

$$E\{W_i W_i^T\} = \begin{bmatrix} (1 - a_x^2) & 0 & 0 \\ 0 & (1 - a_y^2) & 0 \\ 0 & 0 & (1 - a_z^2) \end{bmatrix} C_{Xi} \delta_{ik} \equiv Q \delta_{ik}.$$

Thus, assuming that  $a = a_x = a_y = a_z$  for simplicity (not required), the above is implemented as an underlying error model for predictive statistics via the spdcf method by equating:

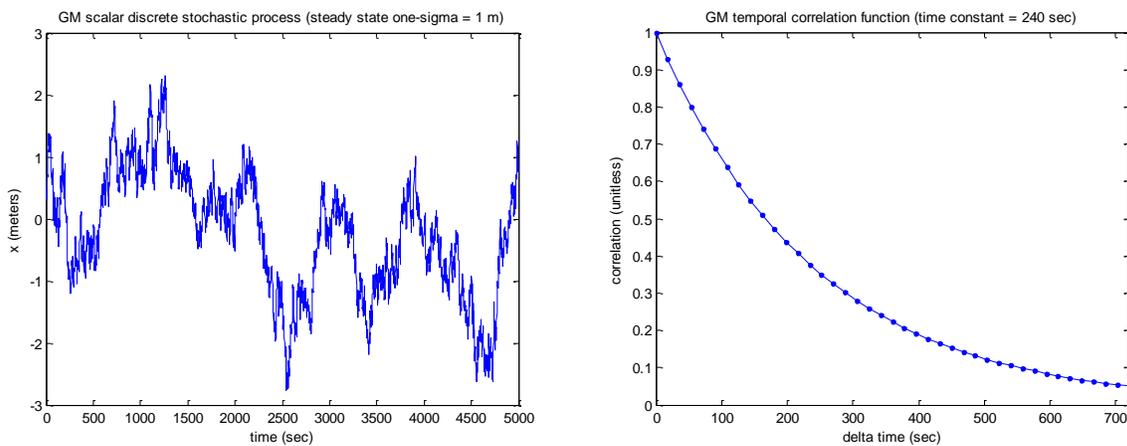
$$C_{Xi} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \text{ for all } i, \text{ and spdcf } \rho(\delta t_{ik}) = a^{|i-k|} = e^{-\Delta t|i-k|/T}. \quad (5.7.1.1-3)$$

Therefore, via Equation (5.7.1-1):

$$C_{Xik} = \rho(\delta t_{ik}) \cdot C_{Xi}^{\frac{1}{2}} C_{Xk}^{\frac{1}{2}} = e^{-\frac{\Delta t|i-k|}{T}} \begin{bmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & \sigma_z \end{bmatrix} \begin{bmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & \sigma_z \end{bmatrix} = \Phi^{|i-k|} C_{Xi}, \quad (5.7.1.1-4)$$

i.e., consistent with the underlying error model as specified by Equation (5.7.1.1-2). Of course, prior to implementation and as part of the *a priori* modeling task, the values for the standard deviations  $\sigma_x, \sigma_y, \sigma_z$  and the value for the time constant  $T$  must be specified.

Figure 5.7.1.1-1 (left) presents a simulation of one realization of x-component error based on implementation of Equation (5.7.1.1-1) (or one component of Equation (5.7.1.1-2)), assuming  $\sigma_x = 1$  meter and time-constant  $T = 240$  seconds. Figure 5.7.1.1-1 (right) presents the corresponding deterministic spdcf, a decaying or damped exponential.



**Figure 5.7.1.1-1:** Gauss-Markov first order process example: realization (left) and corresponding spdcf

References for Gauss-Markov stochastic processes include [20], [15], and [11], and for Gauss-Markov random fields [27], [17], and [19].

The last reference details first-order Gauss-Markov sequential generation equations for 1D, 2D, 3D, and 4D (e.g., 3D spatial and 1D time) random fields, with Figure 5.7.1.1-2 an example corresponding to a 2D scalar random field (the explicit error notation  $\epsilon$  dropped from  $\epsilon z$  for convenience.)

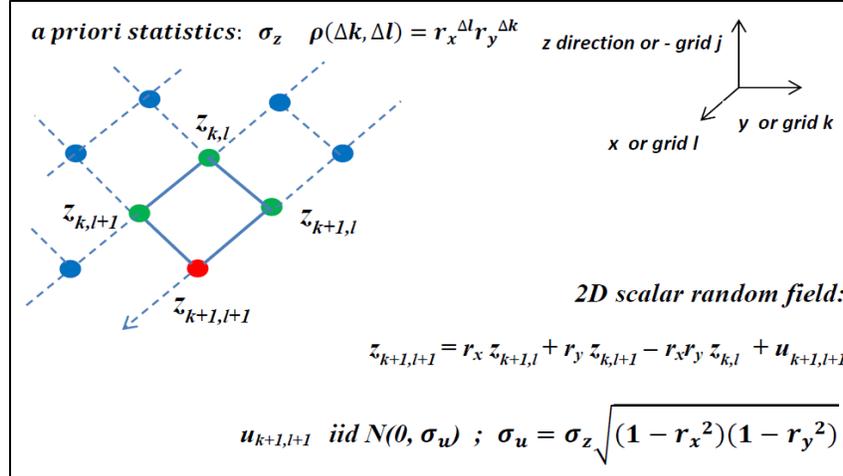


Figure 5.7.1.1-2: Sequential generation of a 2D scalar random field

The (steady state) homogeneous random field corresponds to a scalar  $z_{k,l}$  at horizontal grid location  $k$  and  $l$ . The standard deviation of  $z_{k,l}$  is specifiable as  $\sigma_z$  and the spdcf specifiable as  $\rho(\Delta k, \Delta l) = r_x^{\Delta l} r_y^{\Delta k} = e^{-\Delta l \delta_x / T_x} e^{-\Delta k \delta_y / T_y}$ , where  $\Delta l$  and  $\Delta k$  are the number of grid units between two locations,  $\delta_x$  and  $\delta_y$  are meters/grid unit in the two directions, and  $T_x$  and  $T_y$  specifiable distance constants for the two spatial directions.

Thus, assuming a 2d to 1d index ordering function  $i = o(l, k)$  that maps grid location to individual state vector location within the multi-state vector, we have the following predictive statistics compatible with the spdcf method:

$$\epsilon X_i \equiv z_{k,l}, \tag{5.7.1.1-5}$$

$$C_{Xi} = \sigma_z^2, \text{ spdcf } \rho(\delta d_{ij}) = e^{-\Delta l_{ij} \delta_x / T_x} e^{-\Delta k_{ij} \delta_y / T_y}, \text{ and}$$

$$C_{Xij} = \rho(\delta d_{ij}) \sigma_z \sigma_z = \rho(\delta d_{ij}) \sigma_z^2, \text{ where } \delta d_{ij} \text{ is the 2d spatial separation between } \epsilon X_i \text{ and } \epsilon X_j.$$

### 5.7.1.2 Use of sample statistics

Empirical information used to model errors and their predictive statistics can include that derived from tests using sample statistics of measured errors relative to available “ground truth” or fiducial information. The type of “ground truth” is dependent on the approach available. For example, in the “direct approach”: “true” state vector values are from an independent source and directly correspond to the actual state vectors of interest. In the “indirect approach”: “true” 3d ground coordinates of points are from an independent source and are related to corresponding estimates of their locations that can be generated from the actual state vectors of interest.

In general, the “direct approach” is simpler, more straightforward, and yields better results. The “indirect approach” requires a “reverse mapping” of ground coordinate errors to the actual errors of interest, and is typically iterative in nature. In both approaches, enough samples must be available for reasonable statistical significance. Predictive statistics of “truth” errors should also be available and taken into account regarding the degree-of-fidelity of the derived predictive statistics of interest. Thus,

in the indirect approach, for example, if the actual state vectors of interest only contribute to approximately one meter of derived ground point error, “ground truth” errors need to be on the order of ten centimeters or less, not meters.

In addition, since state vectors of interest are multi-state vectors in general, and can correspond to stochastic processes and random fields, estimates of appropriate spdcf are also required. There is a reasonable amount of research regarding estimation of spdcf (aka variograms, correlogram) in the field of Geostatistics, with references [4], [2], and [30] applicable. For image-based geopositioning, the temporal correlation of a stochastic process corresponding to sensor support data errors affects the horizontal and vertical accuracy of derived ground point locations differently, which can be taken advantage of when estimating the spdcf [11].

In general, the appropriate use of sample-statistics in modeling predictive statistics requires further research, particularly for the indirect approach discussed above. Sample statistics and their relationship to predictive statistics are discussed further in TGD 2b (Sample Statistics).

### 5.7.2 Batch WLS

The following equation presents the general form for the WLS batch estimate of the multi-state vector  $X$  and its multi-state vector error covariance matrix  $C_X$ , given measurements  $M$  with corresponding measurement error covariance matrix  $\Sigma_M$ . If the dimension of  $X$  is  $n$ , the dimension of  $C_X$  is  $nxn$ . The (post-estimate)  $X$  and  $C_X$  are computed automatically by the estimator, assuming it is set-up properly (e.g., sufficient observability):

$$X = C_X B^T W M, \text{ where } C_X = (B^T W B)^{-1}, W = \Sigma_M^{-1}, \text{ and } B = \frac{\partial M}{\partial X}. \quad (5.7.2-1)$$

Furthermore, assuming  $X = [X_1^T \dots X_m^T]^T$ , each  $X_i$  of dimension  $n_i \times 1$ , then  $n = \sum_{i=1}^m n_i$ . If  $C_{X_i}$  is the  $n_i \times n_i$  error covariance for  $X_i$ , then  $C_X = E\{\epsilon X \epsilon X^T\} = \begin{bmatrix} C_{X1} & C_{X12} & \dots & C_{X1m} \\ \cdot & C_{X2} & \dots & C_{X2m} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & C_{Xm} \end{bmatrix}$ , the  $nxn$  error covariance matrix for  $X$ .

$$(5.7.2-2)$$

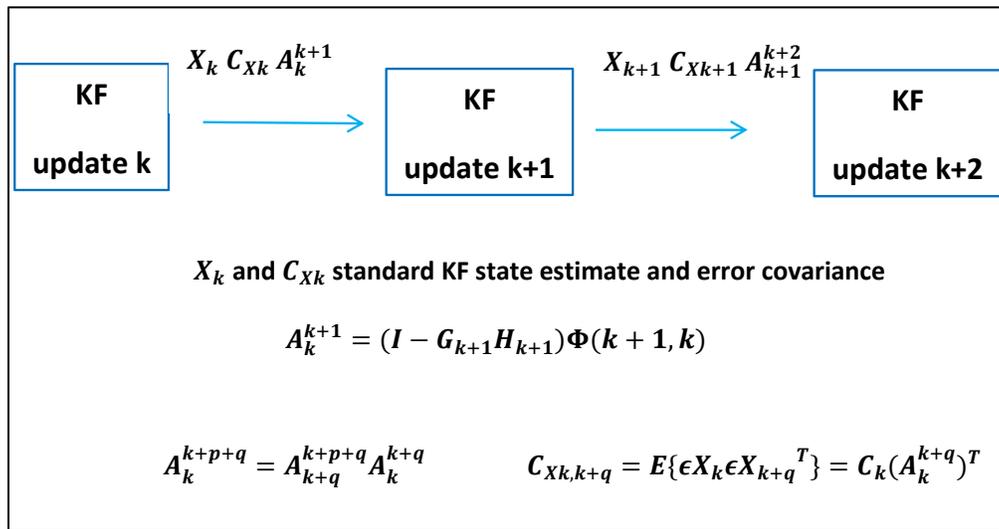
It is required that  $C_X$  is symmetric and positive definite, which implies that all of the  $C_{X_i}$  are symmetric and positive definite, as well. These required conditions on  $C_X$  are guaranteed automatically for a well-formulated WLS solution. Note that the internal structure of the cross-covariance matrices  $C_{X_{ik}}$  can be somewhat complicated, not necessary of the relatively simple form  $C_{X_{ik}} = \rho(\delta t_{ij}) \cdot C_{X_i}^{1/2} C_{X_k}^{1/2}$ , typically used in *a priori* modeling. Their corresponding inter-state vector correlations, and possibly intra-state vector correlations, are usually large (absolute value of correlation coefficients near 1) due to the estimator’s use of a common set of measurements  $M$  for the simultaneous solutions for the various  $X_i$ , i.e.,  $X$ . (Note: the dimension and the identities of the components which make up  $X_i$  and  $X_k$  need not be the same for batch WLS.)

### 5.7.3 Kalman filter or smoother

A standard Kalman filter computes and sequentially outputs a time series of state vectors (estimates) and corresponding error covariance matrices:  $X_1, C_{X1}, X_2, C_{X2}, \dots, X_i, C_{Xi}, \dots, X_m, C_{Xm}$ .

This is not enough to assemble  $C_X$ , i.e., the cross-covariance matrices  $C_{Xik}$  are not included. However, reference [12] presents a rigorous solution. It recommends that the Kalman Filter also compute the “A matrix” at each time or “update” stage, i.e., augment the output  $X_i, C_{Xi}$  at time step  $i$  to  $X_i, C_{Xi}, A_i^{i+1}$ . The latter “A matrix” can be easily computed by the Kalman Filter and is the same dimension as  $C_{Xi}$ .

This process is outlined in Figure 5.7.3-1. Note that the matrices used to generate  $A_i^{i+1}$  are the standard matrices available internally to the Kalman Filter at each update stage  $k$ : the gain matrix ( $G$ ), the partial derivatives of the measurements with respect to the state ( $H$ ), and the state transition matrix ( $\Phi$ ).



**Figure 5.7.3-1:** Augmentation of the standard Kalman Filter output with the “A matrix” - required for “down-stream” cross-covariance generation

Given the augmented Kalman Filter output for a time series  $i = 1, \dots, m$ , the  $m$  different “A matrices” can be used to rigorously (no approximation) assemble any of the possible  $\binom{m}{2}$  number of  $C_{Xik}$ . For example, if  $m = 5000$ , any of the possible  $\cong 12,500,000$  different  $C_{Xik}$ . This is possible due to the transitive property of the “A matrix” as summarized in Figure 5.7.3-1.

See reference [12] for more details regarding the “A matrix”, including derivation of all of its properties. Reference [1] derives similar results for a smoother’s “S matrix”. In addition, reference [16] extends the Kalman Filter results of [12] to include two new features:

- The Kalman filter state (membership) definition can be dynamic, i.e., old components removed and new ones added throughout the time-sequence

- “A matrix” processing can be tailored to computation of the error cross-covariance matrix for a subset of state components that are (only) of interest to “down-stream” applications, which can result in large savings in band-width

### 5.7.3.1 Kalman filter example using the “A matrix” required for cross-covariance

Reference [12] also includes a realistic simulated example of the significant inter-state vector correlation produced by a Kalman Filter (KF), and its impact on a “down-stream” application that estimates 3d ground point locations using the KF-registered image frames (support data) associated with a subset of ten thousand full-motion video frames taken at a 10 hertz rate. KF registration solves for corrections to the support data in real-time.

More specifically, this “down-stream” application solves for the 3d location of a ground point via a WLS estimator using the pixel location of the ground point measured in two of the registered frames which are separated in time (frames) for adequate solution geometry. The measurements of the ground point in the registered frames have two primary sources of error: (1) errors in the explicit measurement of the ground point’s pixel location in each of the two frames, and (2) errors in the previous frame registration (support data) projected to image space for each of the two frames. The latter errors are due to KF solution errors, are dominant, and contain significant inter-state vector correlation. And in order for the WLS application to compute the corresponding multi-state vector error covariance matrix for its input measurements, the corresponding multi-state vector error covariance matrix for the KF solution must be available and used.

The following figures correspond to the above example and present the auto correlation coefficients for KF solution errors for support data attitude (orientation) corrections omega, phi, kappa (Figure 5.7.3.1-1) at frame 7550 with all subsequent frames, and the cross-correlation coefficients for attitude corrections with sensor x-component position corrections at frame 7550 with all subsequent frames (Figure 5.7.3.1-2), i.e., auto-correlation functions and cross-correlation functions, respectively. These correlation coefficients were computed from the  $C_{Xi}$  output by the KF and the  $C_{Xik}$  computed from the  $A_i^{i+1}$  output by the KF. Only the “A matrix” method can rigorously capture the variability of these inter-event correlations, and thus, support optimal WLS “down-stream” 3d ground point estimation.

(Note that the significantly negative-valued cross-correlation function presented in Figure 5.7.3.1-2 can be considered due to negative-valued intra-state vector correlation “damped” by positive-valued inter-state vector correlation that decreases in value with increasing time between Kalman Filter state vector updates.)

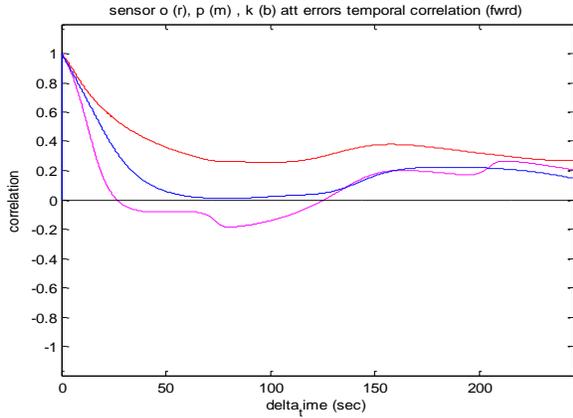


Figure 5.7.3.1-1: KF auto-correlation functions

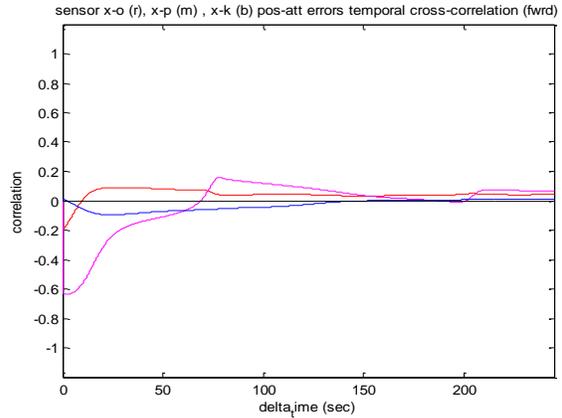


Figure 5.7.3.1-2: KF cross-correlation functions

## 5.8 Generic Methods for Representation/Dissemination of the Multi-State Vector Error Covariance Matrix

There are three generic methods for the representation and dissemination of the multi-state vector error covariance matrix  $C_X$  to “down-stream” applications: (1) direct, (2) “A matrix”, and (3) spdcf.

Note that the dissemination of  $X$  itself is not included in the descriptions below for convenience. Also, as seen below, the assembly example for each method involves only three of the  $m$  individual state vectors  $X_i$ :  $i = 1, 3,$  and  $5$  for specificity. (The appropriate subset of  $C_X$  is also symmetric and positive definite, i.e. valid.) This not only serves for convenience of description, but is typical operationally. For example, if  $C_X$  corresponds to the solution for adjusted image support data in an image bundle adjustment of  $m = 200$  images over a large area of interest, there are typically multiple downstream applications that use different subsets of these adjusted (registered) images in order to accurately extract ground points over their smaller area of interest. However, the bundle adjustment must output the entire  $C_X$  in order to serve all of the applications.

### 5.8.1 Direct method

$$\text{Disseminate: } C_{X1}, C_{X12}, C_{X13}, \dots, C_{X1m}, C_{X2}, C_{X23}, C_{X24}, \dots, C_{X2m}, \dots, C_{Xm}. \quad (5.8.1-1)$$

$$\text{Assembly example for three individual state vectors } i = 1,3,5: C_X = \begin{bmatrix} C_{X1} & C_{X13} & C_{X15} \\ \cdot & C_{X3} & C_{X35} \\ \cdot & \cdot & C_{X5} \end{bmatrix}. \quad (5.8.1-2)$$

The direct method is compatible with WLS generation of  $C_X$ . (In addition, the dimension and the identities of the components which make up the corresponding  $X_i$  and  $X_k$  need not be the same for the direct method.)

### 5.8.2 “A matrix” method

Disseminate:  $C_{X1}, A_1^2, C_{X2}, A_2^3, \dots, C_{Xm-1}, A_{m-1}^m, C_{Xm}, A_m^{m+1}$ . (5.8.2-1)

Assembly example for three individual state vectors  $i = 1,3,5$ :

$$C_X = \begin{bmatrix} C_{X1} & C_{X1}(A_2^3 A_1^2)^T & C_{X1}(A_4^5 A_3^4 A_2^3 A_1^2)^T \\ \cdot & C_{X3} & C_{X3}(A_4^5 A_3^4)^T \\ \cdot & \cdot & C_{X5} \end{bmatrix}. \quad (5.8.2-2)$$

The “A matrix” method is compatible with Kalman Filter (or smoother, with some modifications) generation of  $C_X$ , as discussed in Section 5.7.3.

### 5.8.3 Spdcf method

Disseminate:  $C_{X1}, C_{X2}, \dots, C_{Xm}$ ; and a few parameters defining the scalar-valued spdcf  $\rho(\delta t)$ , where  $\delta t$  can correspond to delta time or delta space, and can be a scalar or multi-dimensional. ( $\delta t_{ik}$  is the delta time or delta distance between applicabilities of individual state vectors, or events,  $i$  and  $k$ ).

Assembly example for three individual state vectors  $i = 1,3,5$ :

$$C_X = \begin{bmatrix} C_{X1} & \rho(\delta t_{13}) \cdot (C_{X1}^{1/2}) (C_{X3}^{1/2}) & \rho(\delta t_{15}) \cdot (C_{X1}^{1/2}) (C_{X5}^{1/2}) \\ \cdot & C_{X3} & \rho(\delta t_{35}) \cdot (C_{X3}^{1/2}) (C_{X5}^{1/2}) \\ \cdot & \cdot & C_{X5} \end{bmatrix}, \quad (5.8.3-1)$$

where the superscript 1/2 indicates principal matrix square root.

The above assembled  $C_X$  is guaranteed valid as proven in [5] and demonstrated in [7] and [8].

Further note that, in general, if  $C_{Xi} = C_{Xk}$ , then  $(C_{Xi}^{1/2}) (C_{Xk}^{1/2}) = C_{Xi}$ . (5.8.3-2)

$$\text{Also, if } C_{Xi} = \begin{bmatrix} \sigma_{1i}^2 & 0 & \dots & 0 \\ 0 & \sigma_{2i}^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_{ni}^2 \end{bmatrix} \text{ and } C_{Xk} = \begin{bmatrix} \sigma_{1k}^2 & 0 & \dots & 0 \\ 0 & \sigma_{2k}^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_{nk}^2 \end{bmatrix}, \text{ then} \quad (5.8.3-3)$$

$$C_{Xik} = \rho(\delta t_{ik}) \begin{bmatrix} \sigma_{1i} \sigma_{1k} & 0 & \dots & 0 \\ 0 & \sigma_{2i} \sigma_{2k} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_{ni} \sigma_{nk} \end{bmatrix} = \begin{bmatrix} \rho(\delta t_{ik}) \sigma_{1i} \sigma_{1k} & 0 & \dots & 0 \\ 0 & \rho(\delta t_{ik}) \sigma_{2i} \sigma_{2k} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \rho(\delta t_{ik}) \sigma_{ni} \sigma_{nk} \end{bmatrix}.$$

The spdcf method is compatible with the *a priori* modeling method for the generation of  $C_X$  (see Section 5.7.1). The specific spdcf is selected based on desired correlation characteristics (see Section 5.8.3.2).

There are no “approximation” errors associated with the spdcf method for representation and dissemination of the multi-variate error covariance matrix, other than those that may exist that directly correspond to the *a priori* modeling itself. Also, it is not uncommon that all of the  $C_{Xi}$  are modeled as diagonal matrices; hence,  $C_{Xik}$  is also diagonal (see Equation (5.8.3-3)), making representation easy.

Also, regarding *a priori* modeling, it is recommended that the following optional constraint be enforced for all relevant  $i, k$  in Equation (5.8.3-1) to insure “realism”:

$$\rho(\delta t_{ik}) \leq \text{sqrt}(\text{trace}(C_{Xi})/\text{trace}(C_{Xk})), \text{ if } \text{trace}(C_{Xk}) > \text{trace}(C_{Xi}),$$

$$\rho(\delta t_{ik}) \leq \text{sqrt}(\text{trace}(C_{Xk})/\text{trace}(C_{Xi})), \text{ if } \text{trace}(C_{Xk}) \leq \text{trace}(C_{Xi}),$$

where  $\text{trace}()$  is the sum of the diagonal elements of the enclosed matrix. (5.8.3-4)

This is done, if need be, by selection of the specific spdcf and/or (minimal) scaling of the desired  $C_{Xi}$  via  $C_{Xi} \rightarrow s \cdot C_{Xi}$ ,  $s > 0$ . Although the multi-state vector error covariance matrix  $C_X$  is theoretically valid without the constraint satisfied, the constraint insures an underlying error process that is statistically consistent with most real-world processes. For example, if inter-event correlation is positive and reasonably high between two events, such as 0.8, their expected magnitude of error should be within 20% of each other. Finally, of course, the constraint is only applicable when the desired  $C_{Xi}$  vary over  $i = 1, \dots, m$ . See [10] for more details on this constraint and “real-world” processes.

The spdcf method can also be used to approximate  $C_X$  (actually the various  $C_{Xik}$ ) generated by either a WLS, Kalman Filter, or some other estimator in order to minimize its size (bandwidth) for dissemination. That is, a suitable spdcf may be fit, subject to the optional constraint (Equation (5.8.3-4)), to the inter-event correlation implied by  $C_X$  prior to dissemination. If used to approximate the results of a KF, the spdcf is typically a function of delta time. If used to approximate the results of a WLS that solves for ground point locations (and possibly image support data adjustments), the spdcf is typically a function of spatial distance. Once the spdcf is generated,  $C_X$  is assembled as in Equation (5.8.3-1). The resultant diagonal blocks  $C_{Xi}$  are exact; however, the fidelity of the resultant cross-blocks  $C_{Xik}$  is application-dependent (additional research is needed to quantify fidelity versus specific applications). Section 5.9.1 presents a specific example.

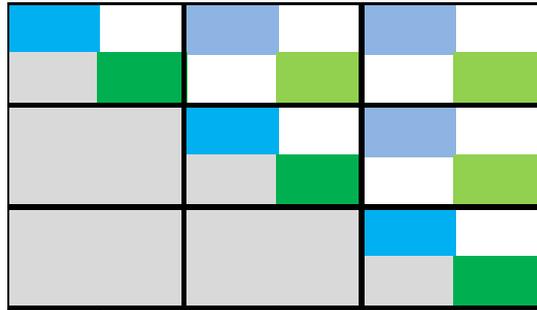
### 5.8.3.1 Correlation subgroups

The above description of the spdcf method assumed one spdcf applicable to all error components which are common to all individual state vectors. Thus, assuming all individual state vector errors  $\epsilon X_i$  were  $nx1$ , the  $nxn$  block-diagonal  $i$  (full, in general) of the multi-state vector error covariance matrix  $C_X$  was equal to  $C_{Xi}$ , and the  $nxn$  cross-block  $i-j$  (full, in general) was equal to  $\rho(\delta t_{ij}) \cdot \left( C_{Xi}^{1/2} \right) \left( C_{Xj}^{1/2} \right)$ , where the scalar correlation value  $\rho(\delta t_{ij})$  multiplied each element of the  $nxn$  matrix  $\left( \left( C_{Xi}^{1/2} \right) \left( C_{Xj}^{1/2} \right) \right)$ .

The above can be generalized to the use of multiple spdcf, one per “correlation subgroup”. The  $C_X$  above had one correlation subgroup that contained all  $nx1$  error components, and thus, all intra-state vector correlations and inter-state vector correlations were allowed, i.e., both  $C_{Xi}$  and  $C_{Xij}$  were full.

However, multiple correlation subgroups can be defined instead, each corresponding to a subset of the  $n$  error components, such that the total number of error components sums to  $n$ .

If subgroup  $k$  has  $n_k$  elements, it corresponds to an  $n_k \times n_k$  block-diagonal (full, in general, and symmetric) in  $C_{X_i}$  and an  $n_k \times n_k$  cross-block (full, in general, not symmetric) in cross-block  $C_{X_{ij}}$ . The elements in two different correlation subgroups are assumed uncorrelated. This is illustrated in Figure 5.8.3.1-1, which assumes three individual state vectors and two correlation subgroups. The colors blue and light blue correspond to blocks and cross-blocks, respectively, for correlation subgroup 1. The colors green and light green correspond to blocks and cross-blocks, respectively, for correlation subgroup 2. The color white indicates no correlation, i.e., cross-blocks of zeros. The color gray indicates error covariance matrix symmetry (intra-block-diagonal symmetry not shown).



**Figure 5.8.3.1-1:** Color-coded assembled multi-state vector error covariance matrix corresponding to three individual state vectors and two correlation subgroups

Thus, when defining correlation subgroups there is always a trade-off: freedom to select a different spdcf per subgroup, but all subgroups must be modelled as uncorrelated with each other. Together, these two attributes insure a valid assembled multi-state vector error covariance matrix.

The above illustrated flexibility and generality using the spdcf method. However, many applications are very straightforward. For example, GPS receiver errors for a typical GPS receiver on a UAV [16] may be reasonably modeled as a first-order Gauss-Markov stochastic process (spdcf is a damped exponential) in a local tangent plane system (ENU) as follows using one correlation subgroup:

$$C_{X_i} = \begin{bmatrix} 2^2 & 0 & 0 \\ 0 & 2^2 & 0 \\ 0 & 0 & 2^2 \end{bmatrix} \text{meters-squared, for all } i, \quad (5.8.3.1-1)$$

$$\text{and } C_{X_{ij}} = e^{-|t_i-t_j|/300} \begin{bmatrix} 2^2 & 0 & 0 \\ 0 & 2^2 & 0 \\ 0 & 0 & 2^2 \end{bmatrix} = \begin{bmatrix} e^{-|t_i-t_j|/300} 2^2 & 0 & 0 \\ 0 & e^{-|t_i-t_j|/300} 2^2 & 0 \\ 0 & 0 & e^{-|t_i-t_j|/300} 2^2 \end{bmatrix} \quad (5.8.3.1-2)$$

meters-squared for all  $i, j$ , and time in seconds.

### 5.8.3.2 Spdcf properties and examples

Strictly positive definite correlation functions (spdcfs) and their applications for the generation of  $C_X$  were described earlier. They provide a practical method to specify inter-state vector correlations, and

the computation of the cross-covariance matrix  $C_{Xik}$  using the spdcf and matrix square roots as indicated in Sections 5.8.3 and 5.8.3.1. The use of spdcfs insures that the resultant  $C_X$  is a valid error covariance matrix for an arbitrary number of individual state vectors (aka “events”). This is not true for other candidate correlation functions, even though they may seem reasonable, i.e., even if their evaluation satisfies the minimal requirements that  $\rho(\delta t = 0) = 1$  and  $-1 < \rho(\delta t \neq 0) < 1$ .

An spdcf that is a function of delta time typically corresponds to a multi-variate stochastic process; such as a time series of individual state vectors (e.g., image support data) with errors  $\epsilon X_i$ . An spdcf that is a function of spatial distance typically corresponds to a multi-variate random field; for example, a spatial location series of individual state vectors (ground points) with errors  $\epsilon X_i$ . (Of course, there can actually be multiple spdcfs, one per correlation subgroup as described in Section 5.8.3.1.)

There are many different families of spdcf, with a specific member of a family specified by the values of a few parameters. Figure 5.8.3.2-1 illustrates members from four different spdcf families: damped exponential, damped cosine, piece-wise linear convex with a non-negative floor, and second order Gauss-Markov. In addition, any convex sum of an arbitrary number of members from an arbitrary number of spdcf families is an spdcf as well. An *a priori* modeling application must select the appropriate family and then specify an appropriate member of that family which reflects the desired correlation characteristics.

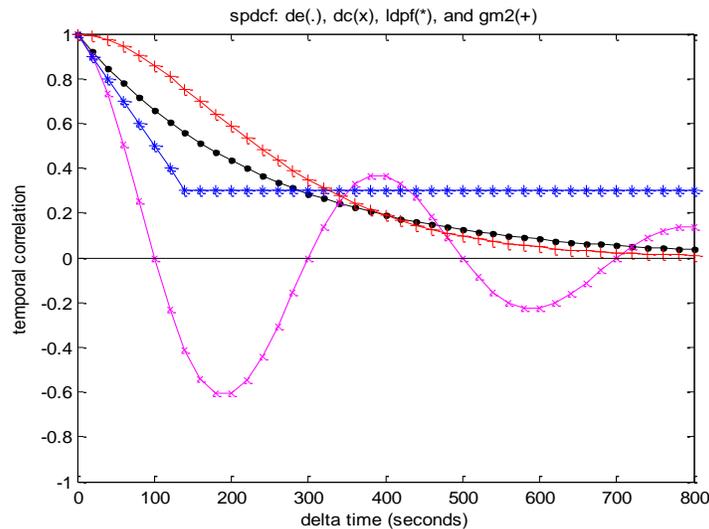


Figure 5.8.3.2-1: Examples of spdcf families

Another spdcf family is the “CSM four parameter” family. (See [18] for more details regarding this spdcf family as well as the Community Sensor Model or CSM.) It is very general, with a specific member specified by the values of four parameters:  $A$ ,  $\alpha$  (“alpha”),  $\beta$  (“beta”), and  $T$ . All of these parameters are unit-less except  $T$ , which has the same units as the independent variable  $\tau$  (or  $\delta t$ ). A specific spdcf member is defined as follows:

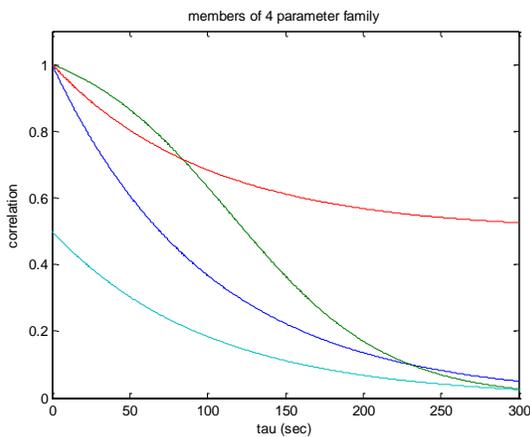
$$\rho(\tau) = A \left[ \alpha + \frac{(1-\alpha)(1+\beta)}{\beta + e^{\tau/T}} \right], \text{ where} \tag{5.8.3.2-1}$$

$$0 < A \leq 1; 0 \leq \alpha < 1; 0 < T; 0 \leq \beta \leq 10.$$

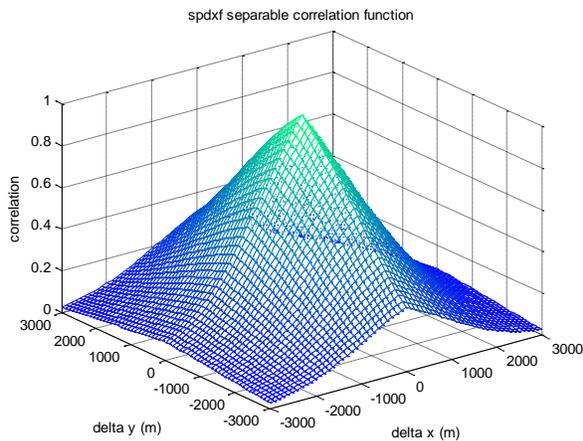
Note that the symbol  $D$  sometimes replaces the symbol  $T$  when dealing with spatial correlation instead of temporal correlation.

In addition,  $\rho(\tau = 0) \equiv 1$ ,  $\rho(\tau = +\epsilon) = A$ , where epsilon is a very small positive number, and  $\rho(\tau \rightarrow +\infty) = A\alpha$ . Note that  $A < 1$  can be considered as corresponding to the inclusion of an uncorrelated random error component, and that  $0 < \alpha$  can be considered as corresponding to the inclusion of a random bias error component. Also, the member  $\{1,0,0,T\}$  corresponds to the familiar damped exponential  $e^{-\tau/T}$ .

Figure 5.8.3.2-2 presents examples of specific members from this family (the units of  $\tau$  in these examples are seconds) with corresponding parameter values:  $\{1,0,0,100\}$  (blue),  $\{1,0,10,50\}$  (green),  $\{1,0.5,0,100\}$  (red), and  $\{0.5,0,0,100\}$  (teal).



A) **Figure 5.8.3.2-2:** Specific members of the “CSM four parameter” family



**Figure 5.8.3.2-3:** Example of a separable spdcf

The spdcf’s independent variable  $\tau$  can be a scalar, such as the absolute value of delta time, or multi-dimensional, such as horizontal two-dimensional distance between two 3d ground points. If the latter, the spdcf can also be separable, e.g., have the form  $\rho(\tau_1, \tau_2) = \rho(\delta x, \delta y) = \rho_x(\delta x) \cdot \rho_y(\delta y)$ . Figure 5.8.3.2-3 presents an example of  $\rho(\delta x, \delta y)$ , where each of its composite functions  $\rho_x(\delta x)$  and  $\rho_y(\delta y)$  are members of the CSM four parameter family. Note the different correlation characteristics in the x

and the y directions. Thus, for example, the multi-state vector error covariance matrix for a field of  $m$  3d ground control points could be represented by  $m$  unique (or the same-valued, if applicable)  $3 \times 3$   $C_{Xi}$ , and eight parameters specifying the two composite functions. That is, using this data,  $C_X$  could be assembled as follows by a down-stream application in order to properly weight the control:

$$C_X = \begin{bmatrix} C_{X1} & \rho_x(\delta x_{12}) \cdot \rho_y(\delta y_{12}) \cdot C_{X1}^{1/2} C_{X2}^{1/2} & \dots & \rho_x(\delta x_{1m}) \cdot \rho_y(\delta y_{1m}) \cdot C_{X1}^{1/2} C_{Xm}^{1/2} \\ \cdot & C_{X2} & \dots & \rho_x(\delta x_{2m}) \cdot \rho_y(\delta y_{2m}) \cdot C_{X2}^{1/2} C_{Xm}^{1/2} \\ & \vdots & \ddots & \vdots \\ & \vdots & & C_{Xm} \end{bmatrix} \quad (5.8.3.2-2)$$

In general, a separable spdcf can be a product of up to  $n$  individual spdcf when distances are expressed in an  $n$ -dimensional metric space, such as the Cartesian Coordinate System  $R^n$ . In addition, this coordinate system need not correspond to the coordinate system in which the individual errors  $\epsilon X_i$  are expressed. Thus, for example, if the  $\epsilon X_i$  correspond to errors in 3d ground locations expressed in a local tangent plane system, spdcf distances can be with respect to a rotated tangent plane.

### 5.8.4 Bandwidth

The following Table 5.8.4-1 summarizes the amount of data or “bandwidth” needed to disseminate the multi-state vector error covariance matrix for the three methods: direct, “A matrix”, and spdcf. All of these methods explicitly disseminate the (upper triangular portion) of the individual  $n \times n$   $C_{Xi}$  for all individual state vectors (aka “events”) of interest. The direct method also explicitly disseminates all of the individual  $n \times n$  cross-covariance matrices  $C_{Xik}$  for all individual state vectors of interest, whereas the other two methods disseminate much less data for  $C_{Xik}$  down-stream assembly. The “A matrix” method requires more data than does the spdcf method, but its assembly of the  $C_{Xik}$  is rigorous (exact) and not an application-dependent approximation as for the spdcf method. Regardless, for a large number of individual state vectors  $m$ , both the “A matrix” and spdcf methods are practical, whereas the direct method may not be. Both of these methods are relatively new and hopefully will facilitate the use of the multi-state vector error covariance matrix  $C_X$  by various applications.

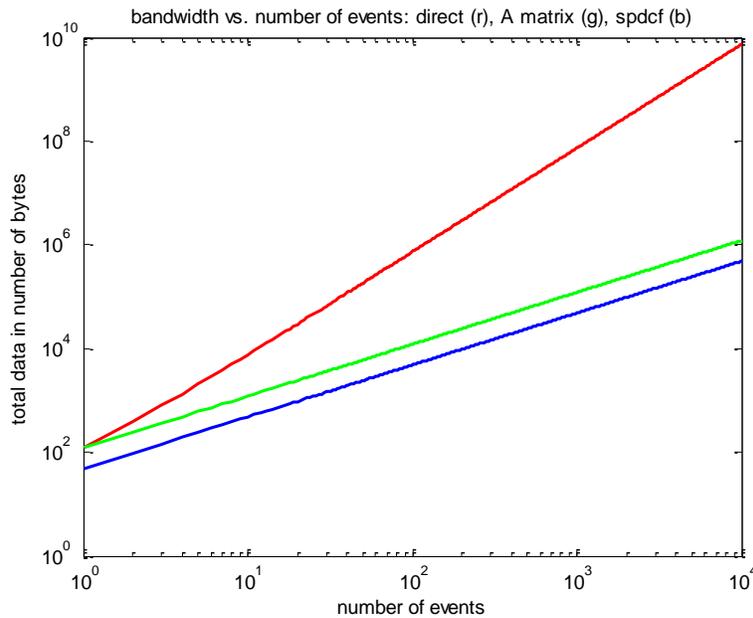
**Table 5.8.4-1:** Bandwidth requirements vs. dissemination method

Method	Single event cov		Cross-cov		average # multi-event cov data elements per event for m events
	# events	Rigorous	# event pairs	Rigorous	
Direct	All	Yes	bandwidth limited	Yes	$(n)(n+1)/2 + (m)(n)(n)$
A matrix	All	Yes	Unlimited	Yes	$(n)(n+1)/2 + (n)(n)$
Spdcf	All	Yes	Unlimited	Approx	$(n)(n+1)/2$

As a specific example of the bandwidth needed for the various methods, Table 5.8.4-1 requirements were converted from (data per event, assuming  $m$  event) to (total data required, summed over all  $m$  events), as a function of number of events  $m$  and presented in Figure 5.8.4-1. The dimension  $n$  was assumed equal to 3 for specificity, and could correspond to individual state vectors  $X_i$  that correspond

to 3d ground point locations, for example. (If the dimension  $n$  is increased, the differences in bandwidth growth would be even more dramatic than currently illustrated in Figure 5.8.4-1.)

In addition, a data element (error covariance element) was assumed to require 8 bytes. (This latter assumption could be reduced if error covariance matrices were normalized, i.e., decomposed to the square-root of their diagonal elements along with all applicable correlation coefficients.).



**Figure 5.8.4-1:** Bandwidth as a function of the number of individual state vectors or “events” for different methods to disseminate the (full) multi-state vector error covariance matrix

## 5.9 Approximation of the Multi-State Vector Error Covariance Matrix

In some applications, a low-bandwidth approximation of a large multi-state vector error covariance matrix may be desirable and feasible. It may correspond to, for example, the solution error covariance matrix of a large batch WLS solution, which typically induces non-trivial intra-state vector correlation and inter-state vector correlation, i.e., a full matrix.

The allowed fidelity (degree of realism – see Section 5.2.4) of the approximation is application-dependent, but the approximation must yield a valid error covariance matrix.

If the corresponding individual state vectors correspond to 3d location of features or ground points, and their collective “footprint” corresponds to a region across horizontal-space (e.g., earth-surface), such an approximation of the multi-state vector error covariance matrix may be preferred over a regional predictive CE/LE summary. The latter consists of a collection of separate regions of CE and LE (average or typical) values, and regions of relative CE and relative LE values for point-pairs within regions and between regions. Such a summary is not invalid as it stands, but if one were to reverse engineer an

approximation of the original multi-state vector error covariance matrix from it for use in valid-added processing, such a covariance matrix would be low fidelity: its general form would be a diagonal matrix for all covariance blocks and covariance cross-blocks, with entries corresponding to x-errors and y-errors equal. More importantly, without proper attention given to the possible contradictory summaries over different regions, it could easily be invalid.

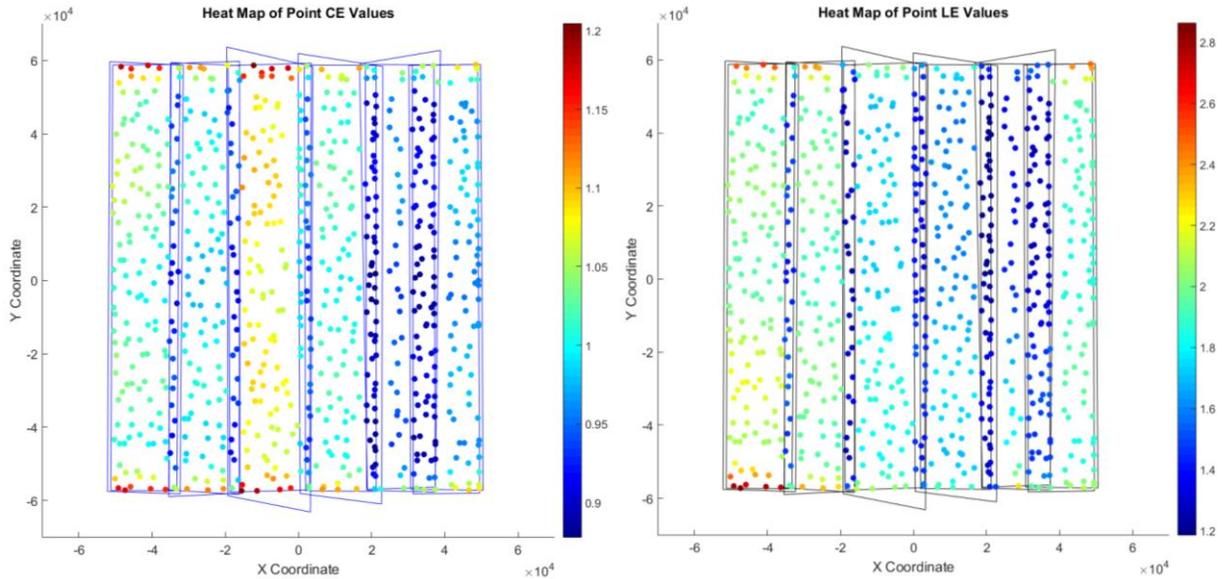
A potential alternate approach for approximation of the original multi-state vector error covariance matrix is based on the spdcf method. Assuming reasonable patterns of correlation between the errors between individual state vectors, results will be both higher fidelity than a predictive CE/LE summary and guaranteed a valid error covariance matrix. The approximation will be low-bandwidth, and after its subsequent generation and dissemination, predictive CE/LE summaries can be generated from it in order to create a “stand-alone” product, if so desired. Section 5.9.1 illustrates one variant of such an approximation and an overview of its computational procedure.

### 5.9.1 Spdcf Method: approximation example

A batch WLS adjustment was performed to correct or “register” 6 large stereo-pairs of WorldView-1 imagery. The adjustment solved for 9 sensor parameter corrections per image, consisting of 3 sensor position offset, 3 sensor attitude offset, and 3 sensor attitude rate corrections. (These parameters affect the ground location to pixel location correspondence.) In addition, 748 tie points were automatically measured between overlapping images (516 2-way tie points and 232 4-way tie points), and each of their corresponding 3d locations were solved for in the adjustment as well. (The tie point image measurements provide “linkage” between the images as well as corresponding ground points.) Thus, the entire state vector for solution consisted of  $108+2244=2352$  elements. In addition, the solution for the 748 tie points and their  $2244 \times 2244$  portion of the  $2352 \times 2352$  full solution (*a posteriori*) error covariance matrix were saved. Default *a priori* error covariance matrices for the sensor parameters for adjustment were used per Digital Globe, including (non-diagonal) error cross-covariance matrices associated with the temporal correlation of sensor parameters for same-pass images generated based on an spdcf, and 0.5 pixel (one-sigma) *a priori* error covariance matrices were used for all (line,sample) image measurements, which were assumed uncorrelated between line and sample coordinates, as well as between measured points.

The above (saved) solution is the  $2244 \times 1$  multi-state vector of interest consisting of 748 individual (but correlated)  $3 \times 1$  (tie point) state vectors, and the corresponding  $2244 \times 2244$  multi-state vector error covariance matrix consisting of  $3 \times 3$  block-diagonals for each tie point and  $3 \times 3$  cross-blocks for each tie point pair. These are of interest to internal or down-stream processing as “derived” ground control, along with their (full) error covariance matrix for appropriate weighting as well as characterization of (absolute and relative) predicted accuracy.

Figure 5.9.1-1 presents corresponding (post-registration) predictive (absolute) CE and LE as they vary across the region, with the tie point footprints included. These values were derived from the block-diagonals of the multi-state vector error covariance matrix.



**Figure 5.9.1-1:** CE (left) and LE (right) for all tie points (2-way and 4-way) from batch registration solution’s multi-state vector error covariance matrix (plus overlay of 12 image footprints making up the 6 stereo pairs)

Furthermore, it was also desirable to approximate the multi-state vector error covariance matrix using the spdcf method, yielding an approximate 1:560 reduction in bandwidth. Only the unique error covariance matrix elements per individual state vector need be saved, as well as a few parameters defining the spdcf(s) used. (In reality, this is a relatively small example in terms of number of images and ground points for ease of illustration; thus, even larger bandwidth reductions are applicable in general.)

The approximation process consisted of first transforming the multi-state vector error covariance matrix to an equivalent error covariance matrix but relative to a local tangent plane system centered at the middle of the footprints. Two correlation subgroups were then identified, one corresponding to x-y horizontal location errors, and the other to vertical location errors. Each correlation subgroup had its own spdcf: a separable spdcf consisting of the product of two spdcf of CSM four-parameter form, one a function of north-south distance (WorldView-1 scan direction in this example) and the other a function of east-west distance.

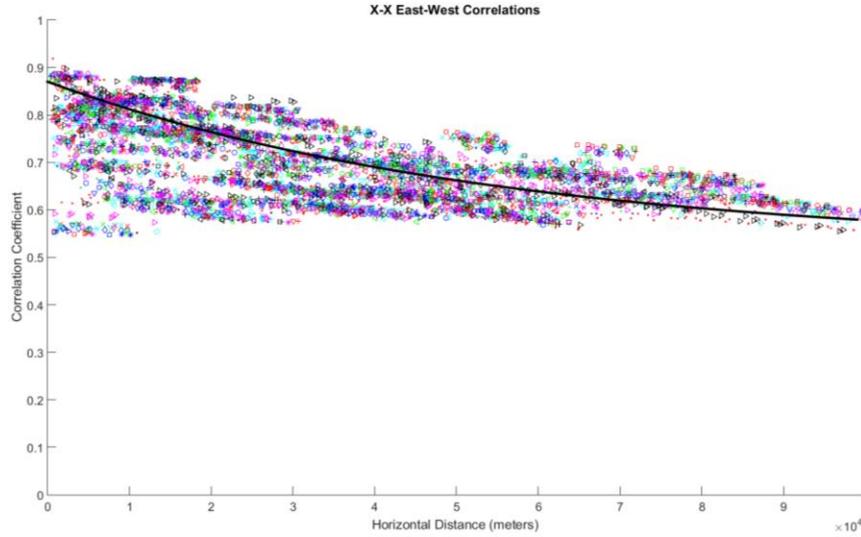
(For an individual image prior to registration, spatial correlation of image location errors due to sensor support data errors is generally high in the scan direction and even higher in the cross-scan direction. However, following registration, and expressed as ground location errors relative to stereo-models, it is generally higher in the scan-direction, as the cross-scan direction crosses stereo models.)

The spdcf were fit to the correlations of each point-pair computed using the original multi-state vector error covariance matrix, described as follows.

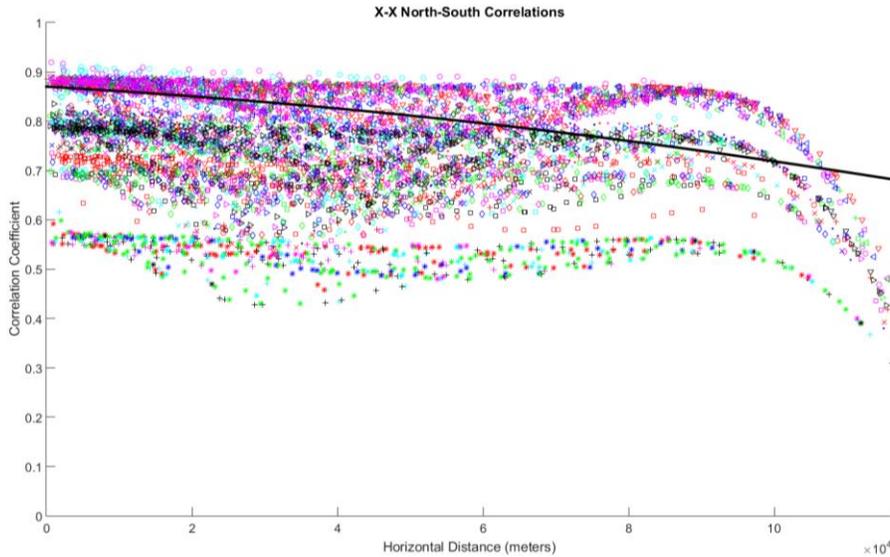
Each spdcf is specified by four parameters  $\{A, \alpha, \beta, D\}$ . A collection of correlation (coefficient) values was computed using all tie point-pairs and the original multi-state vector error covariance matrix. There was a collection of x-x and y-y correlations used for correlation subgroup 1, and a separate collection of z-z correlations for correlation subgroup 2. (Note that x-x correlation corresponds to the correlation coefficient of point  $i$ 's x error with point  $j$ 's x error, for instance.) For a given correlation subgroup, the values of correlation corresponding to minimal distances dictated the  $A$  (initial spdcf) values; those corresponding to very long distance dictated the  $\alpha$  (spdcf floor, as computed by the product of  $A$  times  $\alpha$ ) values, and the general shape of the correlation trend over the range of distances dictated the  $\beta$  (shape) value. The subsequent distance constants  $D$  were solved via a search technique to minimize the corresponding sum of fit residuals squared, holding the  $A$ ,  $\alpha$ ,  $\beta$  values fixed, and with initial approximations for the  $D$  values. (This technique can be augmented to include searching for a subset of the other three parameters, if so desired.) Results are as follows:

SPDCF Parameters XY: EW:	A = 0.93	Alpha = 0.60	Beta = 0	D = 65,000 meters
NS:	A = 0.93	Alpha = 0.35	Beta = 9	D = 65,000 meters
SPDCF Parameters Z: EW:	A = 0.93	Alpha = 0.60	Beta = 0	D = 30,000 meters
NS:	A = 0.93	Alpha = 0.35	Beta = 9	D = 60,000 meters

An example of spdcf fit results is shown next for the (composite) spdcf for correlation group 1 (x and y error), and specifically for x-x correlations versus E-W distance (Figure 5.9.1-2) and x-x correlations versus N-S distance (Figure 5.9.1-3). Note that there are non-negligible fit residuals, not unexpected due to the use of both 2-way and 4-way tie points in the WLS batch registration solution, a relatively small number of images, and no external ground control (measured points with known 3d locations); thus, solution spatial correlation characteristics are not homogenous across the entire area (footprints), and sometimes concentrate in color-coded bands that correspond to the stereo models (footprints).



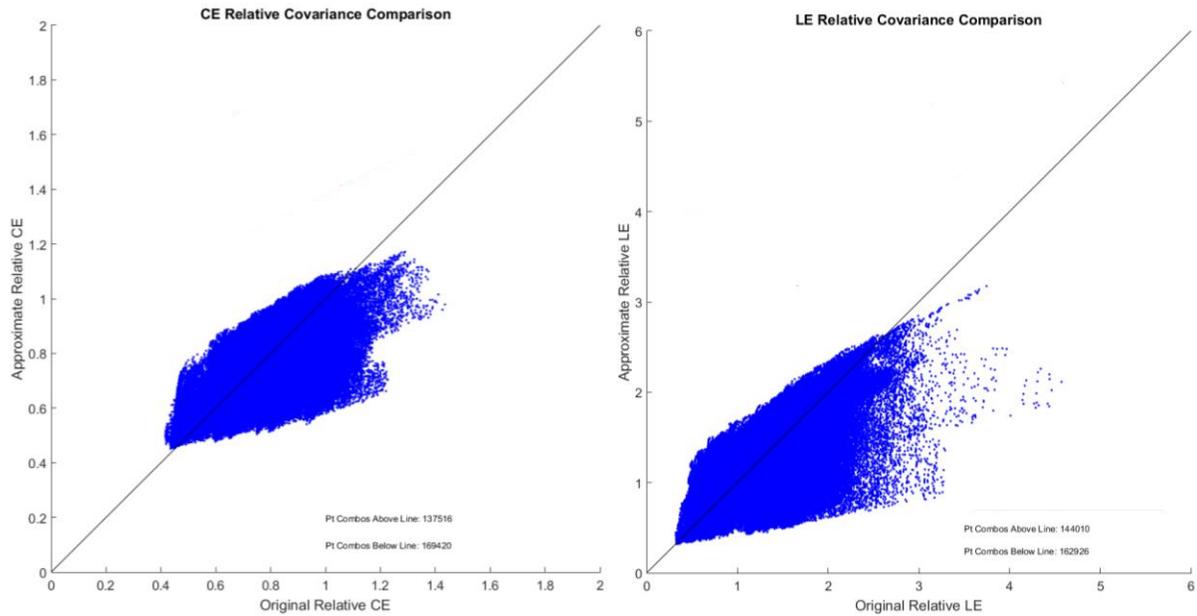
**Figure 5.9.1-2:** spdcf x-x fit results versus E-W distance



**Figure 5.9.1-3:** spdcf x-x fit results versus N-S distance

After generating the spdcfs, the corresponding approximate multi-state vector error covariance matrix was then assembled per Section 5.8 and tested for fidelity. The test involved the predictive absolute CE (abs\_CE) and predictive absolute LE (abs\_LE) computed for each tie point, and the predictive relative CE (rel\_CE) and the predictive relative LE (rel\_LE) computed for each tie point pair, once using the assembled approximate error covariance matrix and once using the original error covariance matrix. (The units for absolute and relative CE and LE are meters; see Section 5.4 and Section 5.5.4 for predictive CE and LE computation details.)

The following presents predicted accuracy summary results graphically, essentially “approximation” versus “original”, where all applicable units are meters. Only rel\_CE and rel\_LE results are shown, as the abs\_CE and abs\_LE results were identical between approximation and original, and thus consistent with Figure 5.9.1-1, as expected.



**Figure 5.9.1-4:** rel\_CE (left) and rel\_LE (right) comparison results using two correlation subgroups; approximation versus original (blue)

In Figure 5.9.1-4 (left), the x-axis corresponds to rel\_CE computed using the original error covariance matrix for a tie point pair, and the corresponding y-axis value (blue dot) corresponds to rel\_CE computed using the approximate error covariance matrix for the same tie point pair. (Thus, a value intersecting the 45 degree black line is a perfect “match”; corresponding comparisons are also applicable to rel\_LE in the figure on the right.)

In general, the approximation does a reasonable job in the computation of rel\_CE and rel\_LE; hence, the approximate multi-state vector error covariance matrix does a reasonable job of capturing the original correlations. And, of course, its corresponding individual 2x2 block-diagonals for x and y, and its 1x1 block-diagonal for z are exact.

(In general, predictive absolute CE and LE are larger than their predictive relative CE and LE counterparts, whether based on the original or approximate error covariance, due to the high correlation of 3d coordinate errors between tie point pairs, induced by the batch WLS registration process.)

Next, the same process described above was repeated, but using only one correlation subgroup, instead of two. It consisted of (x,y,z) coordinate errors. The approximate error covariance matrix will have exactly the same 3x3 error covariance block-diagonals for each individual state vector (tie point) as does the original error covariance matrix; thus, even more fidelity than the process using two correlation subgroups. However, there will be some degradation in spdcf fitting in that all correlations (x-x,y-y,z-z) affect the one composite correlation function. This may degrade somewhat the corresponding rel\_CE and rel\_LE values relative to the original values.

Results are as follows: The spatial correlation of one correlation subgroup (x,y,z) was approximately the same as that for two correlation subgroups described previously. The defining spdcf parameters for one correlation subgroup were as follows:

SPDCF Parameters XYZ: EW: A = 0.93	Alpha = 0.60	Beta = 0	D = 50,000 meters
NS: A = 0.93	Alpha = 0.35	Beta = 9	D = 65,000 meters

In addition, corresponding rel\_CE and rel\_LE results were virtually identical to those of Figure 5.9.1-4 presented earlier for two correlation subgroups – there was no degradation going from two correlation subgroups to one.

Therefore, one correlation subgroup was selected. Correspondingly, the spdcf method to approximate the multi-state vector error covariance matrix of interest yielded “perfect” predicted absolute accuracy results relative to the original multi-state error covariance matrix, and arguably “reasonable” predicted relative accuracy results (correlations) relative to the original multi-state vector error covariance matrix. And, of course, corresponding bandwidth was reduced and the assembled error covariance matrix valid. Also, the amount of band-width reduction and fidelity of predicted relative accuracy results should improve for other examples using a larger number of stereo models.

The following puts the adjective “reasonable” for the above predicted relative accuracy results in perspective: The above one correlation subgroup experiment was repeated, but this time the approximate error covariance matrix had cross-covariance matrices set identically equal to zero, i.e., the spdcf was not used and the 3d location errors were simply assumed uncorrelated for convenience. Comparison of results for rel\_CE and rel\_LE are provided in Figure 5.9.1.5. The blue dots correspond to the earlier results for one correlation subgroup (not explicitly shown, previously), and the red dots to the results when an spdcf was not used. As can be seen in Figure 5.9.1.5, the spdcf is essential.

NGA.SIG.0026.03\_1.0\_ACCPRED

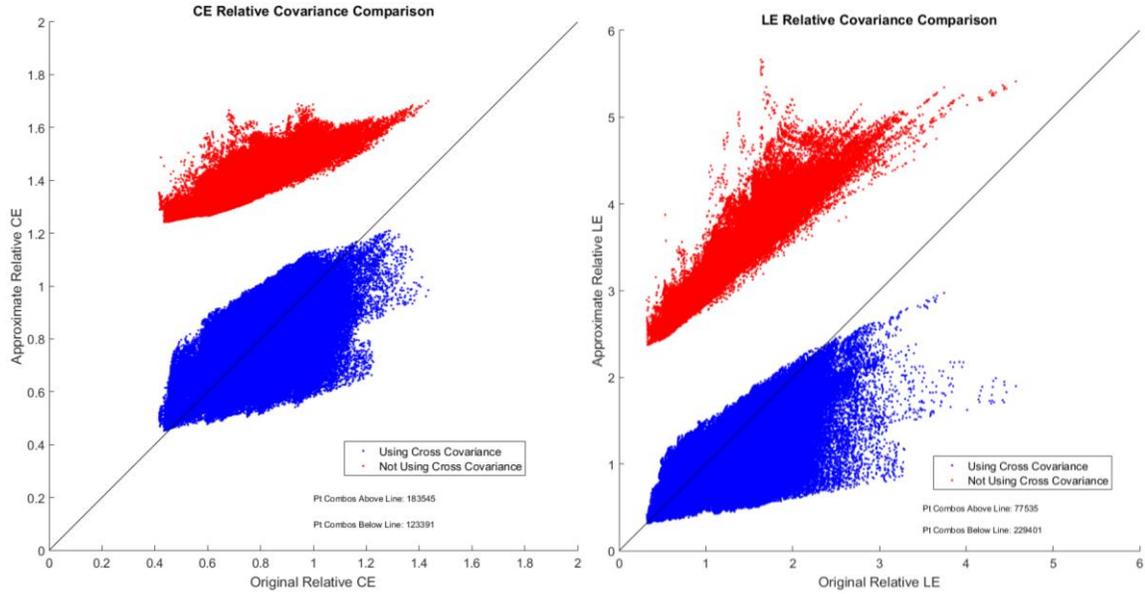


Figure 5.9.1-5: rel\_CE (left) and rel\_LE (right) comparison results using one correlation subgroup; approximation versus original (blue); no spdcf approximation versus original (red)

Finally, the following presents a little more detail regarding the earlier experiment that used an spdcf and one correlation subgroup (Figure 5.9.1-5, blue) for more insight. Three (tie) points were identified, the first (“point 1”) in the East-most stereo block, the second (“point 2”) a few miles away, and the third (“point 3”) far away in the West-most stereo block (see Figure 5.9.1-1). Table 5.9.1-1 details the corresponding original error covariance 3x3 block-diagonals (covariance) for points 1, 2, and 3, and cross-blocks (cross covariance) between points 1-2 and 1-3. It also details the corresponding block-diagonals and cross-blocks from the approximate error covariance. Note that with the spdcf method for approximation, the block-diagonals are identical to the original’s, and the cross-blocks reasonably close, where the approximation’s cross-block is equal to the spdcf value, as a function of E-W and N-S distance between the point pair, times the product of the 3x3 matrix square-roots of the corresponding block-diagonals.

Table 5.9.1-1: Detailed entries in the original multi-state vector error covariance matrix and its approximation corresponding to three tie point locations

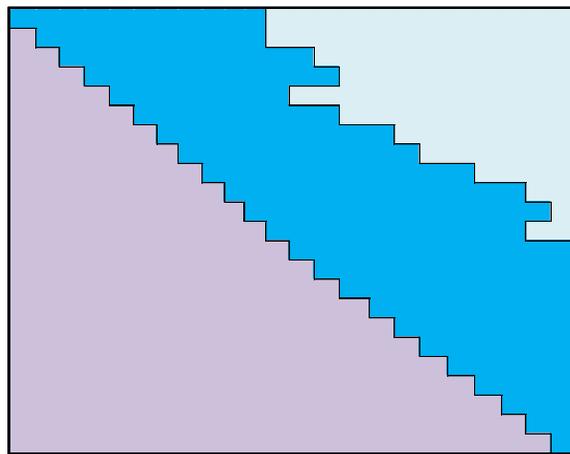
		Point 1 Covariance			Other Point Covariance			Cross Covariance			Approximation Cross Covariance is the product of the SPDCF and the Product of Matrix Square Root			
Close Point (#2)	Original	0.229309	-0.018282	-0.061085	0.236344	-0.019243	-0.061363	0.189307	-0.010358	-0.007122				
		-0.018282	0.263734	0.041796	-0.019243	0.275447	0.038539	-0.010309	0.220159	-0.025033				
		-0.061085	0.041796	1.231333	-0.061363	0.038539	1.258364	-0.007569	-0.024517	0.824708	SPDCF	Product of Matrix Square Root		
	Approx	0.229309	-0.018282	-0.061085	0.236344	-0.019243	-0.061363	0.202324	-0.016284	-0.053343	0.869090	0.232799	-0.018737	-0.061378
		-0.018282	0.263734	0.041796	-0.019243	0.275447	0.038539	-0.016317	0.234239	0.035539		-0.018775	0.269523	0.040892
		-0.061085	0.041796	1.231333	-0.061363	0.038539	1.258364	-0.053081	0.034328	1.081818		-0.061077	0.039499	1.244771
Far Point (#3)	Original	0.229309	-0.018282	-0.061085	0.310256	-0.038443	0.185617	0.103921	0.001154	-0.006354				
		-0.018282	0.263734	0.041796	-0.038443	0.295410	-0.145693	0.006575	0.107450	0.022589				
		-0.061085	0.041796	1.231333	0.185617	-0.145693	1.518881	0.058934	0.007636	0.516280				
	Approx	0.229309	-0.018282	-0.061085	0.310256	-0.038443	0.185617	0.116026	-0.008805	0.001767	0.450997	0.257265	-0.019524	0.003917
		-0.018282	0.263734	0.041796	-0.038443	0.295410	-0.145693	-0.009543	0.123378	-0.005536		-0.021160	0.273567	-0.012275
		-0.061085	0.041796	1.231333	0.185617	-0.145693	1.518881	0.041998	-0.033909	0.610016		0.093122	-0.075186	1.352593

### 5.9.2 Future bandwidth-reduction research

More research/applications of the spdcf method for bandwidth reduction are in order, as the technique appears promising. Other variations include using a representative grid of “anchor points” [26] instead of tie points per se. This is also just one aspect of relevant research regarding bandwidth reduction.

Another research path is the “zeroing out” of cross-covariance blocks when correlations are considered insignificant, e.g., absolute value less than 0.1 for all correlation coefficients in the cross-covariance block. As such, the cross-covariance blocks are no longer explicitly carried as part of the error covariance matrix and understood as containing all zeros. This may also necessitate the addition of minimal positive perturbations to the diagonal elements of the block-diagonal covariance matrices, such that the resultant multi-state vector error covariance matrix is valid. In general, this is a non-trivial task for high-fidelity representation considering all of the statistical interconnections between pairs of individual state vector errors.

Figure 5.9.2-1 is a conceptual example of “block-zero” storage for such a bandwidth limited approximation. This assumes that individual state vectors “closer” together are ordered sequentially and have higher correlations, although such an ordering is not required for reduced bandwidth. Blocks consist of error covariance block-diagonals and cross-covariance blocks, the former always included, i.e., never zeroed.



**Figure 5.9.2-1:** Block-zero storage: non-zero blocks (blue), zero blocks (light blue), transpose not included (light purple) (except for block-diagonals for graphic only)

This type of bandwidth reduction could play a significant role in the generation and storage of very large control point data bases (e.g., contiguous coverage of an entire country or countries), such as those potentially achievable via the Metric Information Network (MIN), described in references [9] and [23].

## 5.10 Overview of References by Section

The following provides an overview of references applicable to the various sections of this document. When multiple references are listed under a category, they are in approximate priority order.

- Introduction (Section 4.1):
  - TGD 1 – “Accuracy and Predicted Accuracy in the NSG: Overview and Methodologies”
  - [14] – the important role of predictive statistics, in particular, the (multi-state vector) error covariance matrix, in a geospatial system
  - [6] - general overview to (image-based) geopositioning and the importance of accuracy predictions
  - [19] - general overview of predicted accuracy and uncertainty relative to GIScience
- Predictive statistics, covariance matrix; definitions and properties (Sections 5.1 and 5.2):
  - [24] - primary probability/statistics reference for this document
  - [22] - primary linear algebra reference for this document
  - [20], [25], [28] - other related references
- Error Ellipsoids (Section 5.3):
  - [14], [6] - general definition and equations for error ellipsoids
  - [29] - a proof that the error ellipsoid contains the maximum probability per volume
  - [22] – definition of matrix  $B > A$  and  $B \geq$  matrix  $A$  and various related inverse and determinant properties
- LE, CE, and SE (Section 5.4):
  - [6] - general definitions and approximation equations
  - [21] - general definitions and alternate approximation equations
- Multi-state vector error covariance matrix (Section 5.5):
  - [14] and [15] – overview and examples
- Generic methods for generation of the multi-state vector error covariance matrix (Section 5.7)
  - [20], [15], and [11] stochastic Gauss-Markov
  - [19], [27], and [17] random field Gauss-Markov
  - [3] – effects of statistical significance and errors corresponding to “ground truth” (sample statistics) in remotely sensed data
  - [14] and [20] - overview, and an overview on estimators, respectively
  - [11] - use of sample statistics from stereo imagery to estimate spdcf
  - [12] and [16] - the “A matrix” for the Kalman Filter
  - [1] – extension of the “A matrix” of [12] to the “S matrix” for smoothers
- Generic methods for representation of the multi-state vector error covariance matrix (Section 5.8)
  - [5], [7], and [18] - the general spdcf and spdcf assembly method
  - [14] – error covariance matrix bandwidth reduction
  - [10] - spdcf assembly method constraint
  - [13] - matrix square roots in general for error covariance applications
  - [8] and [18] - the CSM four parameter spdcf

- [4], [2], and [30] - Geostatistics and its equivalent to the spdcf (variogram and correlogram)
- Approximation of the multi-state vector error covariance matrix (Section 5.9)
  - [26] Anchor points
  - [9] and [23] the Metric Information Network

## 6 Notes

### 6.1 Intended Use

This information and guidance document provides technical guidance to inform the development of geospatial data accuracy characterization for NSG GEOINT collectors, producers and consumers -- accuracy characterization as required to describe the trustworthiness of geolocations for defense and intelligence use and to support practices that acquire, generate, process, exploit, and provide geolocation data and information based on geolocation data. This document is part of a series of complementary documents. TGD 2a provides technical guidance for methods, practices, and algorithms in predictive statistics as of part of a series of information and guidance documents titled Accuracy and Predicted Accuracy in the NSG. Other documents in this series address a more generalized overview of accuracy and predicted accuracy and additional topic specific technical guidance in sample statistics, specification and validation, estimators and quality control, Monte-Carlo simulation, and external data and quality assessment.

## 7 References

- [1] Bercaw, Woodson, "Smoothed Joint Covariance", communication/paper to John Dolloff with permission to distribute, May 2011.
- [2] Christakos, G., Random Field Models in Earth Sciences, Dover, 1992.
- [3] Congalton, R., and Green, K., Assessing the Accuracy of Remotely Sensed Data: Principles and Practices, CRC Press, 2<sup>nd</sup> edition, 2008
- [4] Cressie, N., Statistics for Spatial Data, Wiley, 1993.
- [5] Dolloff, John, et al, "Strictly Positive Definite Correlation Functions", Proc of SPIE, Vol. 6235, 2006.
- [6] Dolloff, John, "Introduction to Photogrammetric-based Geopositioning", April 2010
- [7] Dolloff, John, T., "Assembling a Multi-Imaging Event Error Covariance Matrix", NGA InnoVision white paper, Case # 11-171, July 6 2010.
- [8] Dolloff, John, T., "Parameter ranges for a Valid CSM Decay Model", NGA InnoVision whitepaper, Case # 11-173, July 6 2010
- [9] Dolloff, John, and Settergren, R., "An Assessment of WorldView-1 Positional Accuracy based on Fifty Contiguous Stereo Pairs of Imagery", PE&RS, Vol. 76, No, 8, August 2010.
- [10] Dolloff, J., and Theiss, H., "Correlation of Measurement Errors with Different Uncertainties: Effects on scalar systems, stereo extraction, and the quality assurance of meta-data", NGA InnoVision white paper, Case # 12-075, November 9 2011.

- [11] Dolloff, J.T., and Theiss, H.J., "Temporal Correlation of Metadata Errors for Commercial Satellite Images: Representation and Effects of Stereo Extraction Accuracy", ISPRS XXII Congress, 2012.
- [12] Dolloff, J.T., "Kalman filter outputs for inclusion in video-stream metadata: Accounting for the temporal correlation of errors for optimal target extraction", Proc of SPIE, Vol. 8360, 83600I, 2012.
- [13] Dolloff, J., and Theiss, H., "Matrix Square Roots with Error Covariance Representation Applications", NGA InnoVision whitepaper, Case # 13-073, 02 November 2012.
- [14] Dolloff, J. T., "The full multi-state vector error covariance matrix: Why needed and its practical representation", Proceedings of the SPIE Defense and Security Conference, 2013.
- [15] Dolloff, J., et al, "Optimal Full Motion Video Registration with Rigorous Error Propagation", Proceedings of the SPIE Defense and Security Conference, 2014.
- [16] Dolloff, J. et al, "Full Motion Video Geopositioning Algorithm Integrated Test Bed", Proceedings of the SPIE Defense and Security Conference, 2015.
- [17] Dolloff, J. and Doucette, P., "The Sequential Generation of Gaussian Random Fields for Applications in the Geospatial Sciences", ISPRS International Journal of Geo-Information, 3(2), pp. 817-852, 2014.
- [18] Doucette, P., et al, "Chapter 11.2: Community Sensor Model Concepts," in Manual of Photogrammetry, 6<sup>th</sup> edition (McGlone, J.C., ed.), ASPRS, 2013.
- [19] Doucette, P., Dolloff, J., Lenihan, M., "Geostatistical modeling of uncertainty, simulation, and proposed applications in GIScience", Proceedings of the SPIE Defense and Security Conference, 2015
- [20] Gelb, A., Applied Optimal Estimation, The M.I.T Press, 1974.
- [21] Greenwalt, C.R., and M.E. Schultz, "Principals of Error theory and Cartographic Applications, Feb 1962, ACIC Technical Report #96.
- [22] Horn, R., and Johnson, C., Matrix Analysis, Cambridge University Press, 1985.
- [23] Johnston, C., et al, "An Automated HPC Implementation of the Metric Information Network (MIN)", Proceedings of the ASPRS Annual Conference, 2013.
- [24] Papoulis, A., Probability, Random Variables, and Stochastic Processes, 3<sup>rd</sup> edition, McGraw-Hill, 1991.
- [25] Rao, C.R., Linear Statistical Inference and Its Applications, 2<sup>nd</sup> edition, Wiley, 2002.
- [26] Rodarmel, C., et al, "The Universal Lidar Error Model", PE&RS, Vol.81, No. 7, July 2015.
- [27] Rue, H., and Held, L, Gaussian Markov Random Fields, Chapman and Hall/CRC, 2005.
- [28] Spiegel, R., and Stephens, L., "Statistics", 3<sup>rd</sup> edition, Schaum's Outline Series, 199
- [29] Summers, Daniel, "Circular and Elliptical Error", communication/paper to John Dolloff with permission to distribute, May 30, 2000
- [30] Webster, R., and Oliver, M., Geostatistics for Environmental Scientists, 2<sup>nd</sup> edition, Wiley, 2007.

## Appendix A: Additional Terms and Definitions

There are a number of authoritative guides as well as existing standards within the NSG and Department of Defense for definitions of the identified additional terms used in this technical guidance document. In many cases, the existing definitions provided by these sources are either too general or, in some cases, too narrow or dated by intended purposes contemporary to the document's development and publication. The definitions provided in this document have been expanded and refined to explicitly address details relevant to the current and desired future use of accuracy in the NSG. To acknowledge the basis and/or lineage of certain terms Section 3.1, we reference the following sources considered as either foundational or contributory:

[a] Anderson, James M. and Mikhail, E., *Surveying: Theory and Practice*, 7<sup>th</sup> Edition, WCB/McGraw-Hill, 1998.

[b] DMA-TR-8400.1, DMA Technical Report: Error Theory as Applied to Mapping, Charting, and Geodesy.

[c] Defense Mapping Agency, *Glossary of Mapping, Charting, and Geodetic Terms*, 4<sup>th</sup> Edition, Defense Mapping Agency Hydrographic/Topographic Center, 1981.

[d] ISO TC/211 211n2047, Text for ISO 19111 Geographic Information - Spatial referencing by coordinates, as sent to the ISO Central Secretariat for issuing as FDIS, July 17, 2006.

[e] Joint Publication (JP) 1-02, Department of Defense Dictionary of Military and Associated Terms, November 8, 2010 as amended through January 15, 2016.

[f] MIL-HDBK-850, *Military Handbook: Glossary of Mapping, Charting, and Geodetic Terms*, January 21, 1994.

[g] MIL-STD-2401, Department of Defense Standard Practice; Department of Defense World Geodetic System (WGS), January 11, 1994

[h] MIL-STD-600001, Department of Defense Standard Practice; Mapping, Charting and Geodesy Accuracy, February 26, 1990.

[i] *National System for Geospatial Intelligence* [Brochure] Public Release Case #15-489.

[j] NGA.STND.0046\_1.0, *The Generic Point-cloud Model (GPM): Implementation and Exploitation*, Version 1.0, October 03, 2015.

[k] Oxford Dictionaries ([www.oxforddictionaries.com/us/](http://www.oxforddictionaries.com/us/)) copyright © 2016 by Oxford University Press.

[l] Soler, Tomas and Hothem, L., "Coordinate Systems Used in Geodesy: Basic Definitions and Concepts", *Journal of Surveying Engineering*, Vol. 114, No. 2, May 1988.

**A priori** - Relating to or denoting reasoning or knowledge that proceeds from theoretical deduction rather than from observation or experience. [k]

- For typical NSG accuracy and predicted accuracy applications, *a priori* refers to a mathematical statistical model of errors and/or the corresponding state vector containing those errors prior to its adjustment using additional information.

**A posteriori** - Relating to or denoting reasoning or knowledge that proceeds from observations or experiences to the deduction of probable causes. [k]

- For typical NSG accuracy and predicted accuracy applications, *a posteriori* refers to a refined mathematical statistical model of errors and/or the corresponding state vector containing those errors following its adjustment using additional information.

**Absolute Horizontal Accuracy** - The range of values for the error in an object's horizontal metric geolocation value with respect to a specified geodetic horizontal reference datum, expressed as a radial error at the 90 percent probability level (CE). [b],[f],[j]

- There are two types of absolute horizontal accuracy: *predicted* absolute horizontal accuracy is based on error propagation via a statistical error model; and *measured* absolute horizontal accuracy is an empirically derived metric based on sample statistics.
- The term "horizontal accuracy" is assumed to correspond to "absolute horizontal accuracy".
- The 90% probability level (CE) is the default; 95% and 50% probability levels are optional, i.e., CE\_95 and CE\_50, respectively.

**Absolute Vertical Accuracy** - The range of values for the error in an object's metric elevation value with respect to a vertical reference datum, expressed as a linear error at the 90 percent probability level (LE). [b],[f],[j]

- There are two types of absolute vertical accuracy: *predicted* absolute vertical accuracy is based on error propagation via a statistical error model; and *measured* absolute vertical accuracy is an empirically derived metric based on sample statistics.
- The term "vertical accuracy" is assumed to correspond to "absolute vertical accuracy".
- The 90% probability level (LE) is the default; 95% and 50% probability levels are optional, i.e., LE\_95 and LE\_50, respectively.

**Bias Error** - A category of error; an error that does not vary from one realization (trial or experimental outcome) to the other. When error is represented as a random variable, random vector, stochastic process, or random field, a bias error corresponds to a non-zero mean-value. [f],[j]

- Caution: a given realization of a mean-zero stochastic process with typical temporal correlation and over a reasonable finite time interval appears to have a non-zero sample mean-value; however, when sample statistics are taken over enough multiple (independent) realizations, the sample mean-value approaches zero in accordance with the true mean-value. This characteristic extends to random fields as well.

**CE-LE Error Cylinder** – A 3D cylinder made up of CE and LE such that there is between 81-90% probability that the 3d error resides within.

**Confidence Ellipsoid** - An ellipsoid centered at an estimate of geolocation such that there is a 90% probability (or XX% if specified specifically) that the true geolocation is within the ellipsoidal boundary

(ellipsoid interior). A confidence ellipsoid is typically generated based on an error covariance matrix, an assumed mean-value of error equal to zero, and an assumed multi-variate Gaussian probability distribution of error in up to three spatial dimensions.

**Correlated Error** - A category of errors; errors that are correlated with other errors, and typically represented in the NSG as a random vector, stochastic processes, or random field. A correlated error is independent (uncorrelated) with itself and other errors from one realization (trial or experimental outcome) to the next. However, within a given realization, it is correlated with other errors of interest:

- If a random vector, the various elements (random variables) which make it up are correlated with each other (intra-state vector correlation).
- If a stochastic process, the collection of random vectors which make up the stochastic process are correlated with each other (inter-state vector correlation). That is, the elements of one random vector are correlated with the elements of another random vector, typically the closer the two random vectors in time, the greater the correlation. A similar concept is applicable to random fields.

**Correlated Values** - Values (of random variables) which are related by a statistical interdependence. For two random variables, this interdependence is represented by their covariance and typically expressed as a correlation coefficient – both have non-zero values. This interdependence is relative to deviations about their respective mean-values. [f]

**Covariance** - A measure of the mutual variation of two random variables, where variations (deviations or dispersions) are about their respective mean-values. [b]

**Covariance Function** - The cross-covariance matrix of two random vectors associated with a (same) stochastic process or random field as a function of their corresponding time or spatial locations, respectively. If the stochastic process is (wide sense) stationary or the random field (wide sense) homogeneous, the cross-covariance matrix is a function of delta time or delta position, respectively. When evaluated at delta equal to zero, it equals the common covariance matrix.

**Covariance Matrix** - A symmetric,  $nxn$  positive definite matrix populated with the variances and covariances of the random variables contained within a single, multi-component ( $nx1$ ) state vector or random vector. Note that if row  $i$  ( $1 \leq i \leq n$ ) and all corresponding columns  $j$  ( $1 \leq j \leq n, j \neq i$ ) are zero, random variable  $i$  is uncorrelated with all of the other random variables  $j$ . [b]

**Cross-covariance Matrix** - An  $nxm$  matrix containing the covariance between each pair of elements (random variables) of an  $nx1$  random vector and an  $mx1$  random vector.

**Deterministic Error** - An error that is not random or dependent on “chance” – a “known” value, such as the specific realization of an error of an estimated geolocation as compared to “ground truth”, i.e., their difference, where “ground truth” is assumed error-free.

**Distance Constant** - The (separation) distance value such that the correlation coefficient for spatial correlation expressed as a decaying exponential equals  $e^{-1} \cong 0.37$ .

**Earth Centered Earth Fixed (ECEF) Cartesian Coordinate System** - The Conventional Terrestrial Reference System (CTRS) with the following definition:

- 1) Origin: at the geocenter (center of mass of the earth).
- 2) z-axis: Directed toward the conventional definition of the North Pole, or more precise, towards the conventional terrestrial pole as defined by the International Earth Rotation Service (IERS).
- 3) x-Axis: Passes through the point of zero longitude (approximately on the Greenwich meridian) as defined by the IERS.
- 4) y-axis: forms a right-handed coordinate system with the x- and z-axes. [I]

**Error (augmented definition)** - The difference between the observed or estimated value and its ideal or true value. [f] There are a number of different categories of errors applicable to the NSG: Bias Error, Random Error, and Correlated Error. In general, an error of interest may be a combination of errors from these categories. Their combination is typically represented as either a random variable, random vector, stochastic process, or random field:

- A random variable represents a bias error plus a random error. The former corresponds to the random variable's mean-value, and if equal to zero, the random variable represents random error only, which is uncorrelated from one realization of the random variable to the next realization.
- A random vector, stochastic process, and random field can represent all three categories of error. The random variables that make-up (are elements of) random vectors are uncorrelated from one realization to the next by definition. However, within a given realization, they can also be correlated with each other:
  - For a random vector per se, this correlation is also termed "intra-state vector correlation".
  - For a stochastic process, which consists of a collection of random vectors, random variables in one random vector can also be correlated with random variables in another random vector, this is also termed "inter-state vector" correlation. The same concept is applicable to random fields.

**Error Ellipsoid** - An ellipsoid such that there is a 90% probability (or XX% if specified specifically) that geolocation error is within the ellipsoidal boundary (ellipsoid interior). An error ellipsoid can be generated based on a predictive or sample-based error covariance matrix, centered at an assumed predictive mean-value of error equal to zero or a sample-based mean-value of error not equal to zero, and an assumed multi-variate Gaussian probability distribution of error in up to three dimensions.

**Estimator** - an algorithm/process which estimates the value of an  $n \times 1$  state vector. Its inputs are measurements related to the state vector and may include *a priori* information about the state vector.

- An estimator is usually designed to be an optimal estimator relative to a cost function, such as the sum of weighted *a posteriori* measurement residuals, minimum mean-square solution error, etc.
- Estimators are sequential or batch processes, and an optimal estimator should include both an estimate of the state vector and its predicted accuracy, usually an error covariance matrix, as output. A properly implemented MIG for a target's geolocation is an optimal estimator.

**Gaussian (or Normal) probability distribution** - a specific type of probability distribution for a random variable. The distribution is specified by either a Gaussian probability density function or a Gaussian cumulative distribution function. These in turn are completely characterized by the random variable's mean-value and variance.

- The Gaussian (probability) distribution is a common distribution that approximates many kinds of errors of interest to the NSG, and approximates the distribution for a sum of errors from different (non-Gaussian) distributions as well (Central Limit Theorem). A Gaussian distribution corresponding to an  $n \times 1$  random vector is termed a multi-variate Gaussian distribution.

**Geodetic Coordinate System** - Coordinate system in which position is specified by geodetic latitude, geodetic longitude and (in the three-dimensional case) ellipsoidal height [d].

**Ground Truth** - the reference or (assumed) true value of a geolocation of a measured quantity (e.g. associated with an absolute geolocation, or a relative mensuration).

**Homogeneous** - A descriptor for a random field. A random field is (wide-sense) homogeneous if corresponding (*a priori*) statistics are invariant to spatial location. For example, the mean-value and covariance matrix corresponding to its random vectors are constant, and correlation between two corresponding but arbitrary random vectors in the same realization is a function of spatial distance between them, not the explicit spatial location of each.

**Horizontal Error** - As applied to geospatial measurements and processes, horizontal error is typically observed in the  $x, y$  plane of a local right-handed coordinate system where the  $x, y$  plane is defined as tangent to the defined reference surface at the point of origin. While horizontal error is the  $x$  and  $y$  components of error, it may be generalized by its magnitude or 2D radial error.

**Inter-state Vector Correlation** - The correlation between the errors (random variables) of the elements in two different state vectors.

**Intra-state Vector Correlation** - The correlation between the errors (random variables) of different elements in the same state vector.

**Local Tangent Plane Coordinate System** (Coordinate System/Coordinate Reference System) - A local X,Y,Z right-handed rectangular coordinate system such that the origin is any point selected on a given reference ellipsoid, its XY plane is tangent to the reference ellipsoid at the point of origin, and the Y-axis is typically directed to the North Pole (e.g. an East-North-Up (ENU) system). [a]

**Mean-Value** - The expected value of a random variable. Given a collected sample of measurements, the sample mean-value is the average of the values of the sample measurements. The mean-value of a predictive error is typically assumed zero unless specifically stated otherwise. If correctly modelled, the predictive mean-value should be closely approximated by the sample mean-value taken over a large number of independent and identically distributed samples.

- The concept of mean-value readily extends to random vectors and is the vector of the mean-values of the individual components or random variables making up the random vector. It readily extends to stochastic processes and random fields as well, since they are collections of random vectors. If (wide-sense) stationary or (wide-sense) homogeneous, respectively, their corresponding mean-value is a constant random vector mean-value.

**Metadata** - Higher level or ancillary data describing a collection of data, e.g., the sensor support data corresponding to an image, which specifies corresponding sensor position, attitude, interior orientation parameters, etc.

**Multi-image Geopositioning (MIG)** - An optimal solution for a “target’s” geolocation (state vector) with reliable predicted accuracies based on the (weighted) measurements of the geolocation in one or more images. A batch process which minimizes the sum of weighted *a posteriori* measurement residuals, where the latter may also include measurements equivalent to *a priori* estimates of geolocation. MIG can also correspond to the simultaneous solution for the geolocation of multiple targets. In general, a MIG solution’s predicted accuracies correspond to or are derived from the solution’s *a posteriori* error covariance matrix.

**Multi-state Vector Error Covariance Matrix** - An error covariance matrix corresponding to multiple state vector errors (random error vectors) “stacked” one on top of the other as one large state vector error (random error vector), e.g. to represent the position and attitude errors of multiple images’ adjustable parameter errors that impact the solution and predicted accuracy of a subsequent MIG. The multi-state vector error covariance matrix is sometimes termed the joint covariance matrix for a collection of multiple state vector errors.

**Order Statistics** - Nonparametric statistics performed on a set ordered by ascending magnitude of randomly sampled values. Nonparametric statistics assume no *a priori* information about the underlying probability distribution of a random variable such as its mean-value, variance, or type of probability distribution function. In the NSG, order statistics are used to compute scalar accuracy metrics from independent and identically distributed samples of error.

**Percentile** - If a random variable's probability (or sample) distribution is divided into 100 equal parts, the value of the random variable that corresponds to the percentage of the distribution equal to or below the specified percentile, e.g. the 90<sup>th</sup> percentile indicates the lowest sample value such that it is greater than the values of 90 percent of the samples.

- A more formal definition is as follows: The  $p$  percentile of a random variable  $x$  is defined as the smallest number  $x_p$  such that  $p = \text{prob}\{x \leq x_p\}$ . Thus, the probability distribution function (typically unknown) of the random variable  $x$  evaluated at  $x_p$  is equal to  $p$ .  $x_p$  is a deterministic parameter with typically unknown value.

**Precision** - The closeness to one another of a set of repeated observations of a random variable. [a],[f]

- In terms of accuracy, precision is a measure of the repeatability of the underlying errors. High accuracy implies high precision, but not vice versa. For example, for an error represented as a random variable, high precision implies a small standard deviation, but high accuracy implies both a small standard deviation and a small or zero mean-value (or bias).

**Principal Matrix Square Root** - The principal matrix square root of a valid error covariance matrix is a valid error covariance matrix itself of the same dimension such that when multiplied with itself yields the original error covariance matrix. The calculation of principal matrix square root is based on Singular Value Decomposition.

**Probability density function (pdf)** - A function that defines the probability distribution of a random variable. If continuous, its integral is the (cumulative) probability distribution function.

**Probability distribution** - Identifies the probability of a random variable's values over an applicable range of values. There are many different types of probability distributions: Gaussian or Normal, uniform, exponential, etc.

- In most NSG applications for accuracy and predicted accuracy, the random variable and its probability distributions are assumed continuous.
- The probability distribution is specified by either a probability density function or a (cumulative) probability distribution function; either based on an *a priori* model or sample statistics.

**Probability distribution function (cdf)** - The (cumulative) probability distribution function defines the probability that a random variable's value is less than or equal to a specified number in the interval [0,1].

**Radial Error** - A generalization of two horizontal error components ( $x, y$ ) or three dimensional (horizontal and vertical error components –  $x, y, z$ ) error components to a distance value (magnitude) as measured along the radius of a circle or sphere, respectively.

**Random Error** - A category of error; a measure of deviation from an ideal or true value which results from an accidental and unknown combination of causes and varies from one measurement to the next. Not deterministic. For NSG applications, a random error is typically represented as a random variable, random vector, stationary process, or random field. And more specifically, as deviations about their mean-values, the latter considered biases. [b],[f]

- The random error corresponding to a random variable or the random error corresponding to (the elements of) a random vector are independent (uncorrelated) from one realization to the next, by definition.
- The random error corresponding to (the elements of) a random vector can also be correlated between the various elements for a given realization (intra-state vector correlation); hence this error is also a correlated error.
- The random error corresponding to a stochastic process corresponds to the collection of random errors associated with the collection of random vectors making up the stochastic process. Random error is independent (uncorrelated) from one realization to the next. However, within a specific realization, the individual random error vectors are typically temporally correlated amongst themselves (inter-state vector correlation); hence, random error is also correlated error. This same characteristic extends to random fields.
- The probability distribution for a random variable representing a random error is arbitrary – not necessarily Gaussian.

**Random Error Vector** - An error represented by a  $n \times 1$  random vector, and in the NSG, typically corresponds to the error in a state vector's value. The error itself could correspond to a combination of errors from different error categories: bias error, random error, and/or correlated error. That is, the term "random error vector" does not imply the corresponding category of error is necessarily (only) "random error".

**Random Field** - A random field (RF) is a collection of random vectors (RV), parameterized by an N-dimensional spatial vector  $q$ . In general, two different random vectors from the same realization of the random field are correlated. In the NSG, when error is represented by a random field, its corresponding statistics are specified by a statistical error model. A general descriptor of a given random field is as follows: a (“scalar” or “multi-variate”) (“homogeneous” or “non-homogeneous”) “ND random field”.

- Scalar ( $n=1$ ) or multi-variate ( $n>1$ ) refers to the number of elements  $n$  that each random vector contains and is sometimes described as “(nd)”, e.g. (2d) corresponds to 2 elements (random variables) per random vector.
- Homogeneous or non-homogeneous refers to whether the corresponding statistics are invariant or vary over spatial location  $q$ .
- ND refers to the number of spatial dimensions (number of elements in  $q$ ), e.g. 3D corresponds to 3 spatial dimensions. Each random vector corresponds to a unique value of  $q$ .
- As an example of terminology, “a multi-variate homogeneous 3D random field” or more specifically “a homogeneous 3D random field (2d)” corresponds to a multi-variate homogeneous random field over 3 spatial dimensions ( $q$  is a vector with 3 elements). The random vectors contain 2 elements.
- Spatial dimensions are general. For typical NSG applications, they correspond to some combination of geolocation directions and time. Note that a stochastic process is also a random field with  $N=1$ .
- In general, the collection of random vectors is infinite for a random field; however, only a finite subset are of interest for most applications, i.e., random vectors associated with a finite set of spatial locations.
- For typical NSG applications, the spatial correlation of a random field is specified by one or more strictly positive definite correlation functions (spdcf) contained in the corresponding statistical error model.

**Random Variable** - A variable whose value varies by chance, i.e., non-deterministic. Somewhat more formally, a random variable is a mapping from the space of experimental outcomes to a space of numbers. In the NSG, when error is represented by a random variable (a random vector with one component or element, i.e.,  $n=1$ ), its corresponding statistics are specified by a statistical error model.

- For most NSG applications, the space of experimental outcomes is already a number. For example, the x-component of sensor position can be considered a random variable. Equivalently, it can be defined as the true x-component of sensor position plus x-component of sensor position error, the former a deterministic (typically unknown) value and the latter a random variable.
- A random variable is statistically characterized by its mean-value, variance, and (more completely) its probability density function (pdf). The probability density function (pdf) is typically unknown and not included, but if needed for the calculation of probabilities, assumed Gaussian distributed with the pdf completely characterized by the mean-value and variance.

**Random Vector** - A random vector (RV) is an  $n \times 1$  vector which contains  $n$  random variables as components or elements. In the NSG, when error is represented as a random vector, its corresponding statistics are specified by a statistical error model. The corresponding random vector is also sometimes termed a random error vector.

- The realization of a Random Vector corresponds to a specific value of the vector (components or elements) for a given event such as a trial or experiment. Important descriptive statistics of a RV are its mean (vector) value and the error covariance matrix about the mean, and optionally, a multi-variate probability density function. These statistics can be predictive or sample-based.

**Realization** - For NSG accuracy and predicted accuracy applications, a specific trial or experimental outcome or independent sample involving a random error (category of error).

**Relative Horizontal Accuracy** - The range of values for the error in the difference between two objects' horizontal metric geolocation values with respect to a specified geodetic horizontal reference datum; e.g. expressed as a radial error at the 90 percent probability level (CE90). There are two types of relative horizontal accuracy: predicted relative horizontal accuracy is based on error propagation via a statistical error model(s); and measured relative horizontal accuracy is an empirically derived metric based on sample statistics.

**Relative Vertical Accuracy** - The range of values for the error in the difference between two objects' vertical metric geolocation values with respect to a specified geodetic vertical reference datum; e.g. expressed as a linear error at the 90 percent probability level (LE90). There are two types of relative vertical accuracy: predicted relative vertical accuracy is based on error propagation via a statistical error model(s); and measured relative vertical accuracy is an empirically derived metric based on sample statistics.

**Scalar Accuracy Metrics (augmented definition)** - Convenient one-number summaries of geolocation accuracy and geolocation predicted accuracy expressed as a probability: [b],[f], and [h]

- Linear Error (LE) - LE is an unsigned value that corresponds to the length of a vertical line (segment) such that there is a 90% probability that the absolute value of vertical error resides along the line. If the line is doubled in length and centered at the target solution, there is a 90% probability that the true target vertical location resides along the line. LE\_XX corresponds to LE at the XX % probability level.
- Circular Error (CE) - CE is an unsigned value that corresponds to the radius of a circle such that there is a 90% probability that the horizontal error resides within the circle; or equivalently, if the circle is centered at the target solution, there is a 90% probability the true target horizontal location resides within the circle. CE\_XX corresponds to CE at the XX % probability level.
- Spherical Error (SE) - SE is an unsigned value that corresponds to the radius of a sphere such that there is a 90% probability that 3d error resides within, or equivalently, if the sphere is centered at the target solution, there is a 90% probability that the true target location resides within the sphere. SE\_XX corresponds to SE at the XX % probability level.

For the above scalar accuracy metrics:

- It is assumed that the underlying  $x$ - $y$ - $z$  coordinate system is a local tangent plane system, i.e.,  $x$  and  $y$  are horizontal components and  $z$  the vertical component.
- CE-LE corresponds to the CE-LE error cylinder. There is a probability between 81 to 90 percent that 3d radial error resides within the cylinder. The former value corresponds to uncorrelated horizontal and vertical errors, the latter value to highly correlated horizontal and vertical errors.
- LE\_XX, CE\_XX, and SE\_XX (aka LEXX, CEXX, and SEXX, respectively) are also called XX percentiles for absolute vertical errors, horizontal radial errors, and spherical radial errors, respectively. XX is expressed as an integer greater than zero and less than 100.

**Sensor support data** - See “Metadata”.

**Spatial Correlation** - The correlation between the elements (random variables) of two random vectors at two different spatial locations associated with the same realization of a random field.

**Standard Deviation** - The square root of the variance of a random variable. A measure of deviation or dispersion about the random variable’s mean-value.

**State Vector** - A vector of parameters or variables that describe a system’s state.

**State Vector Error** - A vector of errors corresponding to an estimate of a state vector relative to a (typically unknown) true state vector; a random vector of errors, or random error vector.

**Stationary** - A descriptor for a stochastic process with corresponding (*a priori*) statistics invariant over time. See homogeneous as well for random fields, which if corresponding to one spatial dimension are stochastic processes.

**Stochastic Process** - A stochastic process (SP) is a collection of random vectors (RV), parameterized by a 1D quantity, typically time. For a given realization of the stochastic process, the individual random vectors are correlated with each other. If the random vectors consist of one element or component ( $n=1$ ), the stochastic process is sometimes called a scalar stochastic process, and if greater than one, a multi-variate stochastic process. A stochastic process is also a random field with one spatial (or time) dimension, i.e.,  $N=1$ . In the NSG, when error is represented as a stochastic process, its corresponding statistics are specified by a statistical error model.

**Strictly Positive Definite Correlation Function (spdcf)** - A function which models the statistical correlation between random vectors (random variables), typically applied in the NSG to describe the temporal correlation and/or spatial correlation between various random vectors which are part of a stochastic process or random field, i.e., the spdcf is a function of delta time or delta distance (possibly in each of multiple directions) between random vectors. The proper use of an spdcf ensures assembly of a valid multi-state vector error covariance matrix, i.e., positive definite and symmetric.

**Systematic Error** - An error characteristic or error effect due to errors that are represented by random variables, random vectors, stochastic processes, or random fields. For example, an effect on observations (samples) such that their pattern of magnitude and direction are consistent but not necessarily constant. [f],[j] Such an effect can be associated with:

- Error(s) represented by a stochastic process or random field which appear systematic across time or space, respectively, due to temporal or spatial correlation, respectively.
- The error in a frame image-to-ground sensor model's adjustable parameter for focal length. This error is typically represented by a random variable, with a mean-value of zero and a constant variance, but its effect when projected to the ground appears as a systematic error across ground locations, e.g., it has a scaling effect which increases the closer the ground point to the image footprint's boundary.

**Temporal Correlation** - The correlation between the elements (random variables) of two random vectors at two different times associated with the same realization of a stochastic process.

**Time Constant** - The delta time value such that the correlation coefficient for temporal correlation expressed as a decaying exponential equals  $e^{-1} \cong 0.37$ .

**Uncertainty** - A lack of certainty; limited knowledge; unknown or imperfect information. In terms of NSG applications, more general than the concepts of errors and accuracy, but sometimes used informally as a synonym. Applies to predicted accuracy but not to empirical (sample-based) accuracy.

**Uncorrelated Error** - At an intuitive level, an error that is statically unrelated to all other relevant errors. More precisely, if two random variables represent two uncorrelated errors (about their respective mean-values), their covariance and their corresponding correlation coefficient are zero. A random variable is uncorrelated (with itself) from one realization to the next by definition. This latter property is also true for the random variables making up random vectors, stochastic processes, and random fields. However, these three representations typically include correlated errors within the same realization.

**Uncorrelated Values** - Values (of random variables or errors) which are statistically unrelated. [f] This is represented for two random variables by their covariance with a value of zero.

**Vertical Error** - As applied to geospatial measurements and processes, vertical error is a signed and one dimensional (linear) error value typically observed in the direction of the  $z$ -axis of a local right-handed coordinate system where the  $x, y$  plane is defined as tangent to the defined reference surface at the point of origin and the  $z$ -axis is normal to the  $x, y$  plane and positive in the up direction.

**WGS84 - World Geodetic System 1984** – A documented and formally maintained global coordinate system which allows an unambiguous representation of positional information by providing the basic reference frame (coordinate system), geometric figure for the earth (ellipsoid), earth gravitational model, and means to relate positions on various geodetic datums and systems for DoD operations and applications. [g]

## Appendix B: Pseudo-code for Rendering the Error Ellipsoid

This appendix corresponds to Section 5.3.4 on “Rendering the error ellipsoid”; in particular, corresponding pseudo-code (MATLAB).

### B.1 Plot Error Ellipse

```
function makeEllipse(covar,mCoord,valFlag,prob)
%%%%% Creates ellipse based on provided covariance matrix
%%%%%
%%%%% Inputs: covar: 2x2 covariance matrix
%%%%%           mCoord: mean coordinate value
%%%%%           valFlag: flag indicating if fourth input is probability or
%%%%%                   distance value
%%%%%                   entered value can either be 'p' for probability or
%%%%%                   'd' for distance value
%%%%%           prob: desired probability or distance value for ellipsoid

%%% Checks orientation of mean coordinate vector
if size(mCoord) == [1 2]
    mCoord = transpose(mCoord);
end

%%% Checks entered flag to determine entered value type
if strcmp(valFlag,'d')
    d = prob;
elseif strcmp(valFlag,'p')
    %%% Checks entered probability to determine distance value
    if prob == .5
        d = 1.1774;
    elseif prob == .9
        d = 2.1460;
    elseif prob == .95
        d = 2.4477;
    elseif prob == .99
        d = 3.0349;
    elseif prob == .999
        d = 3.7169;
    else
        fprintf('Entered probability is not one of the options.\n')
        quit
    end
else
    fprintf('Entered flag does not match possible values.\n')
    quit
end

%%% Single Value Decomposition for Eigen values and vectors
[u,s1] = svd(covar);

%%% Scales Eigen values by distance value
ellAxes = d*sqrt(diag(s1));

%%% Rotation angle from Eigen space to XYZ space
```

## NGA.SIG.0026.03\_1.0\_ACCPRED

```
gam = atan2(u(2,1),u(1,1));

%%% Vector of angle values for use in polar coordinate calculation of
%%% points on ellipse
phi = 0:2*pi/500:2*pi;

%%% Calculation of coordinates on ellipse using polar coordinates and
%%% rotating from Eigen space to XYZ space
Xp(1,:) = ellAxes(1)*cos(phi)*cos(gam) -
ellAxes(2)*sin(phi)*sin(gam)+mCoord(1);
Xp(2,:) = ellAxes(1)*cos(phi)*sin(gam)+ellAxes(2)*sin(phi)*cos(gam)+mCoord(2);

%%% Matrix of ellipse axes end points in Eigen space
ellAxes = [ellAxes(1) 0;-ellAxes(1) 0;0 ellAxes(2);0 -ellAxes(2)];

%%% Rotates ellipse axes from Eigen space into XYZ space and translates
%%% axes to mean coordinate
ellAxes = transpose(u*transpose(ellAxes)+mCoord*ones(1,size(ellAxes,1)));

%%% Plots ellipse and axes
figure
clf
hold on
plot(Xp(1,:),Xp(2,:), 'g-', 'LineWidth',2)
plot(ellAxes(1:2,1),ellAxes(1:2,2), 'b-', 'LineWidth',2)
plot(ellAxes(3:4,1),ellAxes(3:4,2), 'b-', 'LineWidth',2)
xlabel('X')
ylabel('Y')
grid on
axis equal
legend('Ellipe', 'Ellipse Axes')
hold off
drawnow
```

## B.2 Plot Error Ellipsoid

```
function makeEllipsoid(covar,mCoord,valFlag,prob)
%%%%% Creates ellipsoid based on provided covariance matrix
%%%%%
%%%%% Inputs: covar: 3x3 covariance matrix
%%%%%           mCoord: mean coordinate value
%%%%%           valFlag: flag indicating if fourth input is probability or
%%%%%                   distance value
%%%%%                   entered value can either be 'p' for probability or
%%%%%                   'd' for distance value
%%%%%           prob: desired probability or distance value for ellipsoid

%%% Checks orientation of mean coordinate vector
if size(mCoord) == [1 3]
    mCoord = transpose(mCoord);
end

%%% Checks entered flag to determine entered value type
```

```

if strcmp(valFlag, 'd')
    d = prob;
elseif strcmp(valFlag, 'p')
    %% Checks entered probability to determine distance value
    if prob == .5
        d = 1.5382;
    elseif prob == .9
        d = 2.5003;
    elseif prob == .95
        d = 2.7955;
    elseif prob == .99
        d = 3.3682;
    elseif prob == .999
        d = 4.0336;
    else
        fprintf('Entered probability is not one of the options.\n')
        quit
    end
else
    fprintf('Entered flag does not match possible values.\n')
    quit
end

%% Single Value Decomposition for Eigen values and vectors
[u,s1] = svd(covar);

%% Scales Eigen values by distance value
ellAxes = d*sqrt(diag(s1));

%% Uses MATLAB function to generate points on ellipsoid in Eigen space
[x,y,z] = ellipsoid(0,0,0,ellAxes(1),ellAxes(2),ellAxes(3),20);

%% Moves ellipsoid coordinates from three n+1 by n+1 matrices into a
%% (n+1)^2 by 3 matrix
xyzEll = zeros(size(x,1)^2,3);
for k = 1:size(x,2)
    xyzEll(k*size(x,1)-(size(x,1)-1):k*size(x,1),1) = x(:,k);
    xyzEll(k*size(x,1)-(size(x,1)-1):k*size(x,1),2) = y(:,k);
    xyzEll(k*size(x,1)-(size(x,1)-1):k*size(x,1),3) = z(:,k);
end

%% Rotates ellipsoid points from Eigen space into XYZ space and translates
%% points to mean coordinate
ApertEllLoc = transpose(u*transpose(xyzEll)+mCoord*ones(1,size(xyzEll,1)));

```

## NGA.SIG.0026.03\_1.0\_ACCPRED

```
%%% Moves ellipsoid coordinates from a (n+1)^2 by 3 matrix into a n+1 by
%%% n+1 by 3 matrix for plotting
ApxyzEllLoc = zeros(size(x,1),size(x,2),3);
for k = 1:size(x,2)
    ApxyzEllLoc(:,k,1)= ApertEllLoc(k*size(x,1)-(size(x,1)-1):k*size(x,1),1);
    ApxyzEllLoc(:,k,2)= ApertEllLoc(k*size(x,1)-(size(x,1)-1):k*size(x,1),2);
    ApxyzEllLoc(:,k,3)= ApertEllLoc(k*size(x,1)-(size(x,1)-1):k*size(x,1),3);
end

%%% Matrix of ellipsoid axes end points in Eigen space
xyzAxes      = [ellAxes(1) 0 0;-ellAxes(1) 0 0;
                0 ellAxes(2) 0;0 -ellAxes(2) 0;
                0 0 ellAxes(3);0 0 -ellAxes(3)];

%%% Rotates ellipsoid axes from Eigen space into XYZ space and translates
%%% axes to mean coordinate
ApertAxeLoc = transpose(u*transpose(xyzAxes)+mCoord*ones(1,size(xyzAxes,1)));

%%% Plots ellipsoid and axes
figure
clf
hold on
plot3(ApertAxeLoc(1:2,1),ApertAxeLoc(1:2,2),ApertAxeLoc(1:2,3),'b-
','LineWidth',2)
surf(ApxyzEllLoc(:,:,1),ApxyzEllLoc(:,:,2),ApxyzEllLoc(:,:,3),'FaceColor',[0
1 0],'EdgeColor',[0 .4 0])
plot3(ApertAxeLoc(3:4,1),ApertAxeLoc(3:4,2),ApertAxeLoc(3:4,3),'b-
','LineWidth',2)
plot3(ApertAxeLoc(5:6,1),ApertAxeLoc(5:6,2),ApertAxeLoc(5:6,3),'b-
','LineWidth',2)
alpha(.3)      % sets transparency of ellipsoid faces
view(-15,40)
xlabel('X')
ylabel('Y')
zlabel('Z')
grid on
axis equal
legend('Ellipsoid Axes','Ellipsoid')
hold off
drawnow
```

## Appendix C: Supplementary Material on Predictive Scalar Accuracy Metrics

This appendix corresponds to:

C.1 Solution Comparisons - the various performance tests addressed in Sections 5.4.2.6 (CE) and 5.4.3.5 (SE);

C.2 Complete Set of SE Interpolation Tables referenced in Section 5.4.3;

C.3 Pseudo-Code for the various LE, CE, and SE computation methods.

### C.1 Solution Comparisons Supporting Sections 5.4.2.6 (CE) and 5.4.3.5 (SE)

Both “Method versus Method Comparison” tests and “Method Repeatability” tests were made. Methods consisted of Table Interpolation (Table\_Interp), Monte Carlo Covariance Square Root (MC\_Cov\_sqrt), and Integral Equation (IE) methods for the computation of CE\_XX or SE\_XX.

Table\_Interp assumed an error mean-value of zero by definition, MC\_Cov\_sqrt usually assumed an arbitrary mean-value, and IE was tested with both a mean-value of zero and an arbitrary mean-value.

Table\_Interp assumed fixed probability levels, and all other methods (unless compared to Table\_Interp) assume arbitrary probability levels, randomly selected (uniform distribution) within the interval [0.1,0.999].

For Method versus Method Comparison tests, the number of independent samples was specified. Thus, if 1000 were specified, there were 1000 pairs of CE\_XX (or SE\_XX) calculated for comparison, each pair generated using an error covariance matrix and mean-value generated randomly for that specific pair, and for a randomly selected (uniform distribution) probability within the interval [0.1,0.999] for that specific pair. (The latter was not applicable if one of the methods was Table\_Interp, which utilizes fixed, specified probability levels XX.)

For Method Repeatability tests, the number of independent samples was specified as above and a corresponding error covariance matrix and mean-value generated randomly for each sample. In addition, the number of perturbations was specified over which sample statistics of the CE\_XX or SE\_XX calculations were computed for that sample. For the MC\_Cov\_sqrt method, perturbations consisted of independent sets of  $1E6$  random vectors  $n_i$  used in the calculation of CE\_XX or SE\_XX. For the IE method, perturbations consisted of different initial estimates of CE\_XX or SE\_XX used to solve the integral equation. The initial estimate for the first perturbation is the standard low-fidelity deterministic estimate as implemented in the pseudo-code. Each subsequent perturbation equaled  $(1 + d) \cdot (\text{first perturbation})$ , where  $d$  is a uniform random number within the interval [-0.2,0.2].

The summary performance results presented in Table 5.4.2.6-1 for CE\_XX and Table 5.4.3.5-1 for SE\_XX are from the integrated results of the combined tests. For a given Method, its repeatability test provides the statistical results of inherent calculation error. All methods have a repeatability test other

than Table\_Interp. Its inherent calculation error is determined by its comparison to IE (mean zero), from which the entries for the interpolation table were generated. In addition, various combinations of Method versus Method comparison tests were also performed, essentially as QA checks on the repeatability tests.

For a given sample in a Method versus Method Comparison test, the primary metric is defined as relative % difference between the two methods:  $100 \cdot (\text{CE\_XX\_Method}_1 - \text{CE\_XX\_Method}_2) / \text{CE\_XX\_Method}_2$ , with a similar metric defined for SE. Sample statistics of this performance metric are then taken over all of the samples.

For a given sample in a Method Repeatability test, the (internal) sample standard deviation about the (internal) sample mean is taken over all corresponding perturbations and the performance metric computed as % of the (internal) sample standard deviation relative to the (internal) sample mean, i.e.,  $100 \cdot (\text{standard deviation} / \text{mean})$ . In addition, for a given sample, the maximum deviation of all perturbations relative to the sample mean is computed relative to the (internal) sample mean, i.e.,  $100 \cdot (\text{max deviation} / \text{mean})$ . Sample statistics for both of these performance metrics are then taken over all samples.

Note that the various tests also provided timing results corresponding to the pseudo-code (MATLAB). All comparison and repeatability tests, along with their timing results, were entered into spread-sheets. Summary results corresponding to these various spread-sheets are plotted below, with related comments.

### C.1.1 Circular Error (CE)

For a given sample, the error covariance matrix and mean-value (if not specified zero) were randomly computed as follows:

The error covariance was randomly generated per sample about the mean-value as:

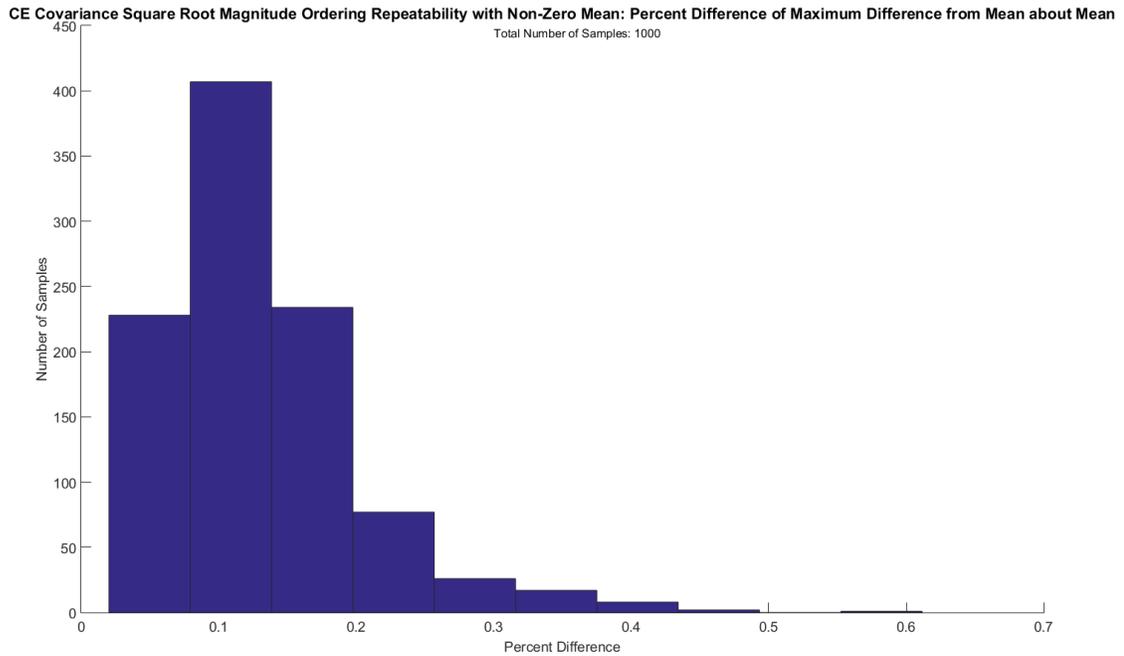
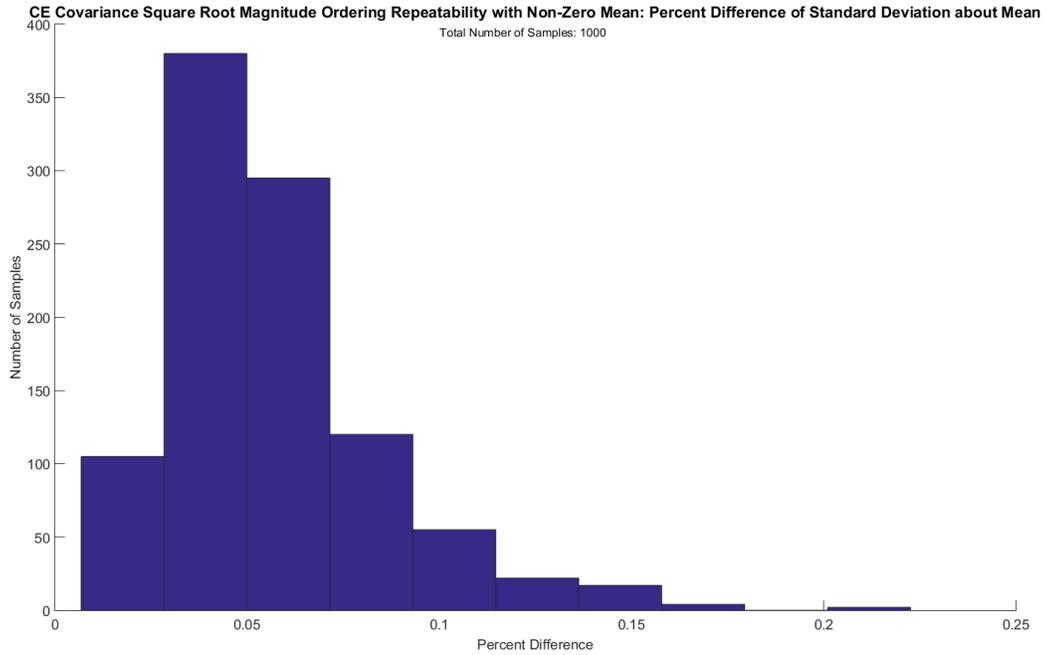
$$\text{Cov}_{random} = 4 \begin{bmatrix} u1 & u3\sqrt{u1 \times u2} \\ u3\sqrt{u1 \times u2} & u2 \end{bmatrix}$$
, where  $u1$  and  $u2$  are random and independent samples from a (0,1) uniform distribution, and  $u3$  a random and independent sample from a (-1,1) uniform distribution. The square root of the smallest to largest eigenvalue was also guaranteed/checked as  $r \geq 0.00001, 0.02, \text{ and } 0.05$  for tests involving MC\_Cov\_sqrt, IE (mean zero), and IE (mean not zero), respectively.

The mean value, if not specified as zero for the test, was randomly generated as  $\bar{X}_{random} = 4 \begin{bmatrix} u4 \\ u5 \end{bmatrix}$ , where  $u4$  and  $u5$  are random and independent samples from a (-1,1) uniform distribution.

**C.1.1.1 CE Method Repeatability Tests**

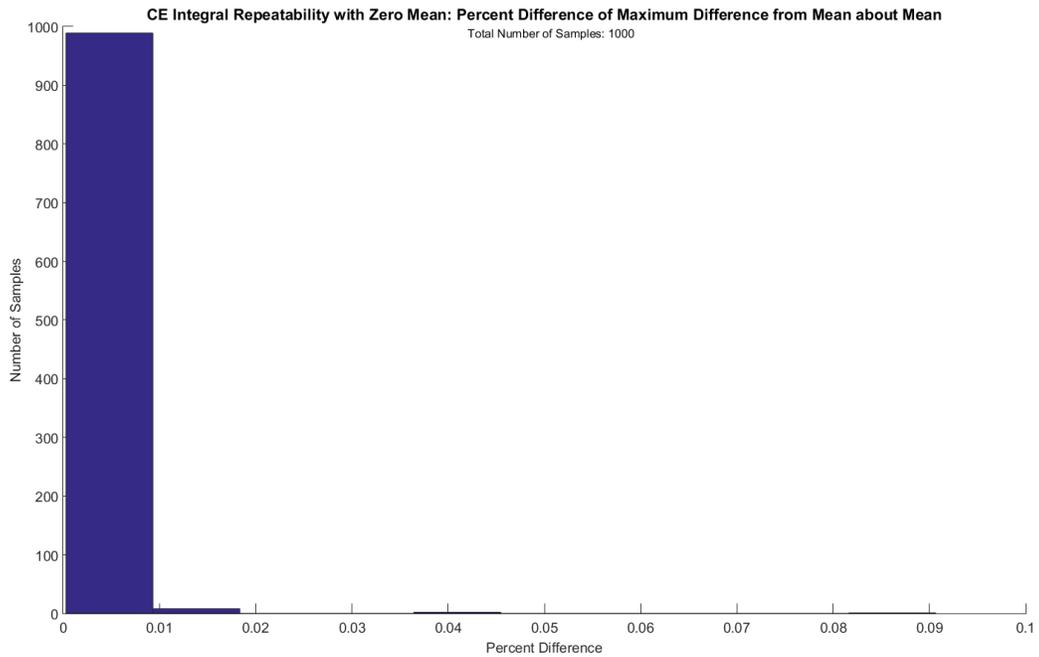
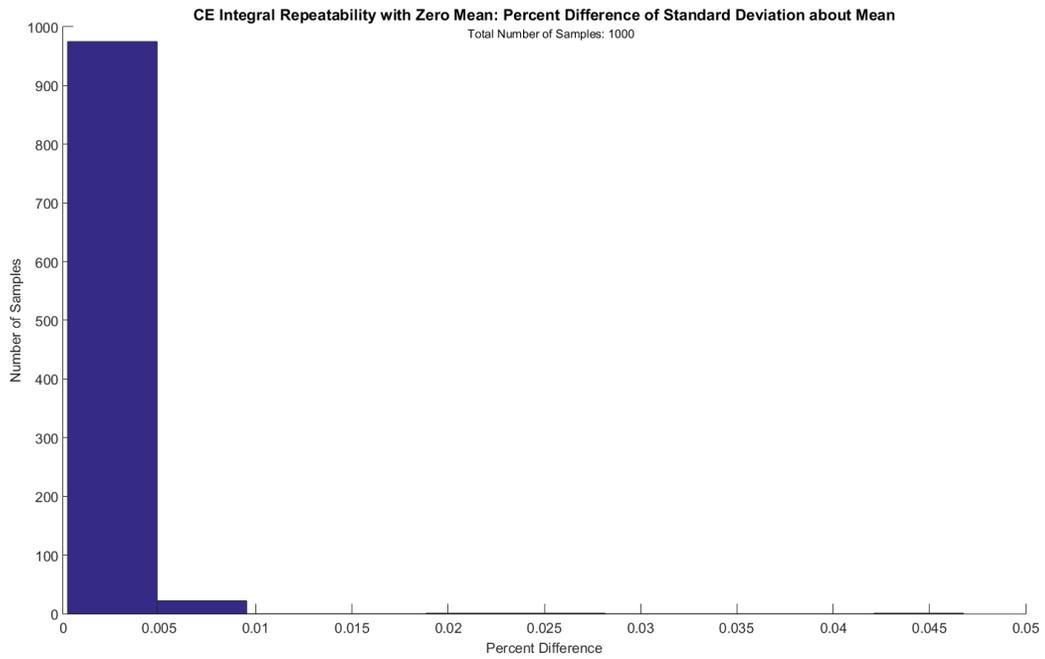
(1) Table\_Interp: not applicable

(2) MC\_Cov\_sqrt:1000 samples and 50 perturbations per sample

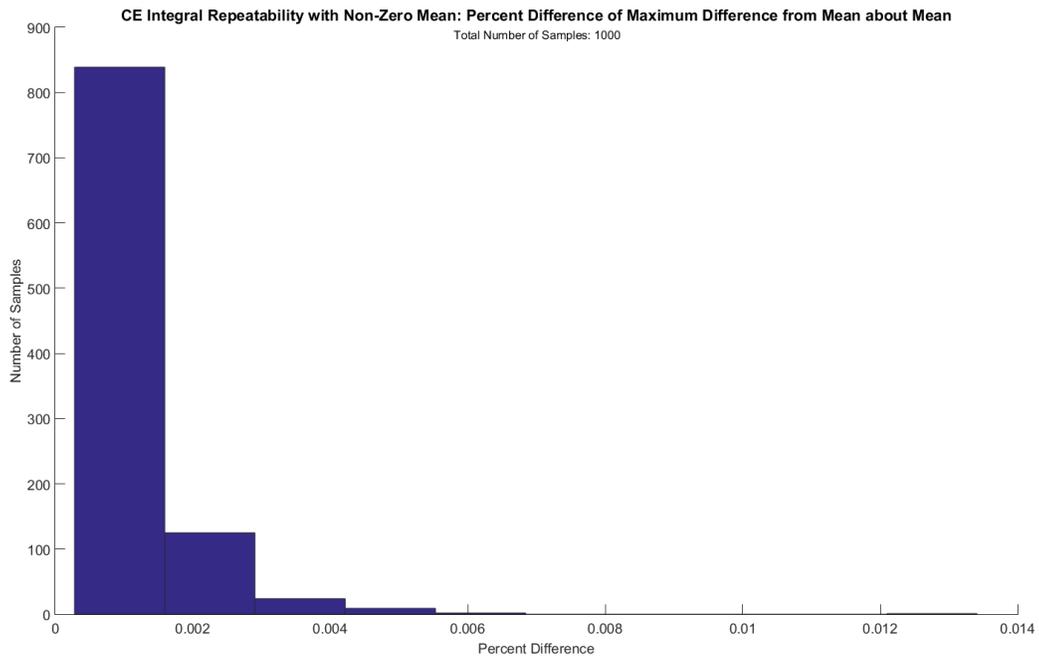
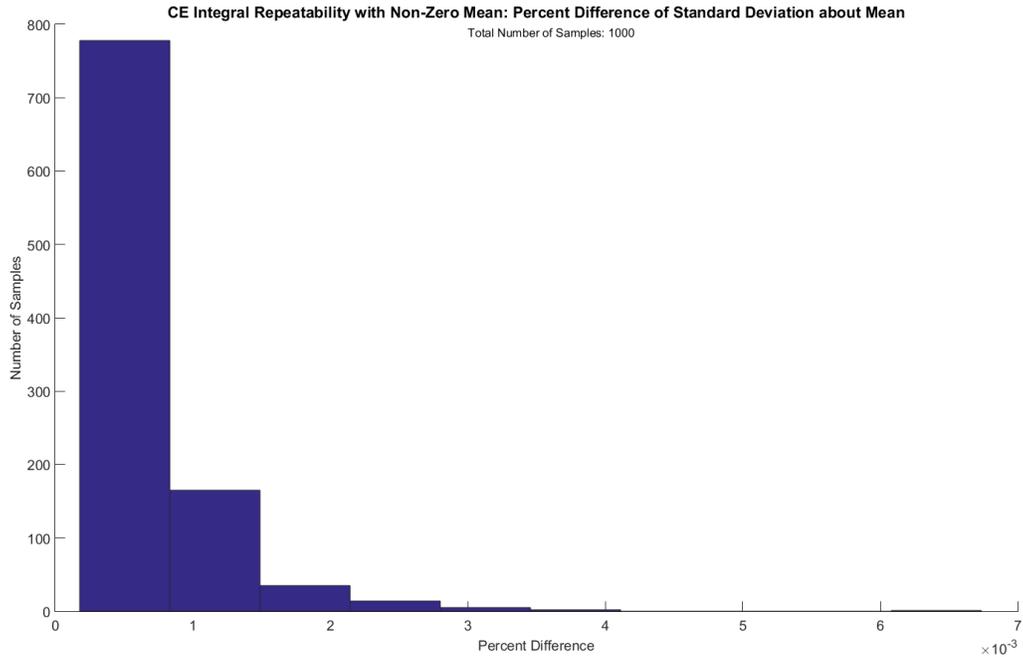


(3)IE (mean zero):1000 samples and 10 perturbations per sample

# NGA.SIG.0026.03\_1.0\_ACCPRED



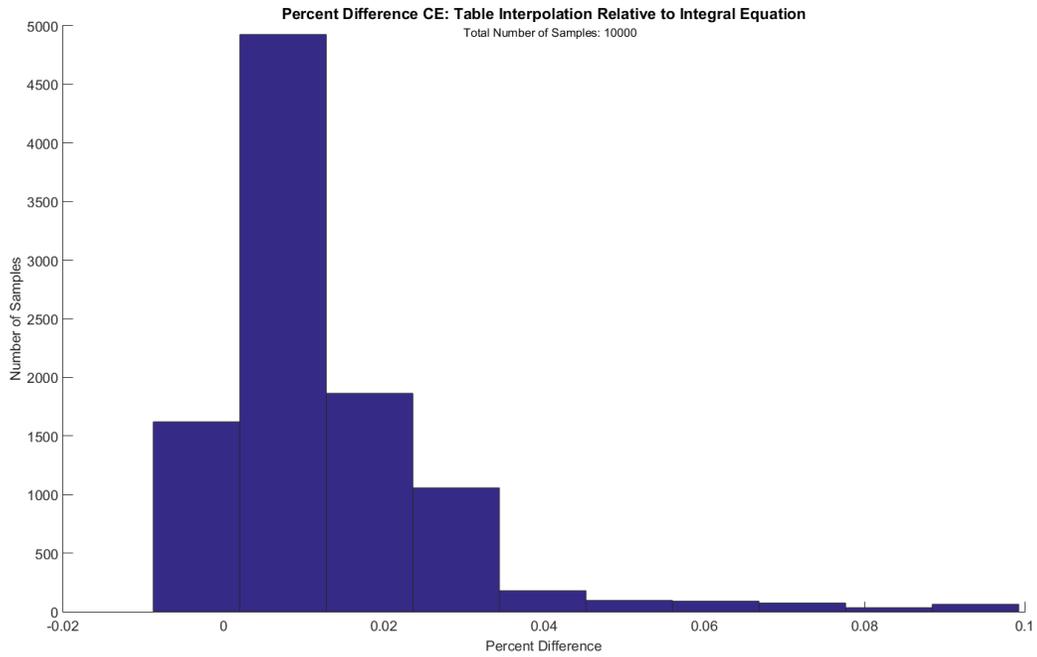
(4) IE (mean value arbitrary):1000 samples and 10 perturbations per sample



As further representation detail, for the above test, the maximum, average, and minimum values randomly selected over the 1000 cases for probability were 0.9980, 0.5520, and 0.1003, respectively; the maximum, average, and minimum values for the ratio  $r$  were 0.9989, 0.4683, and 0.0506, respectively.

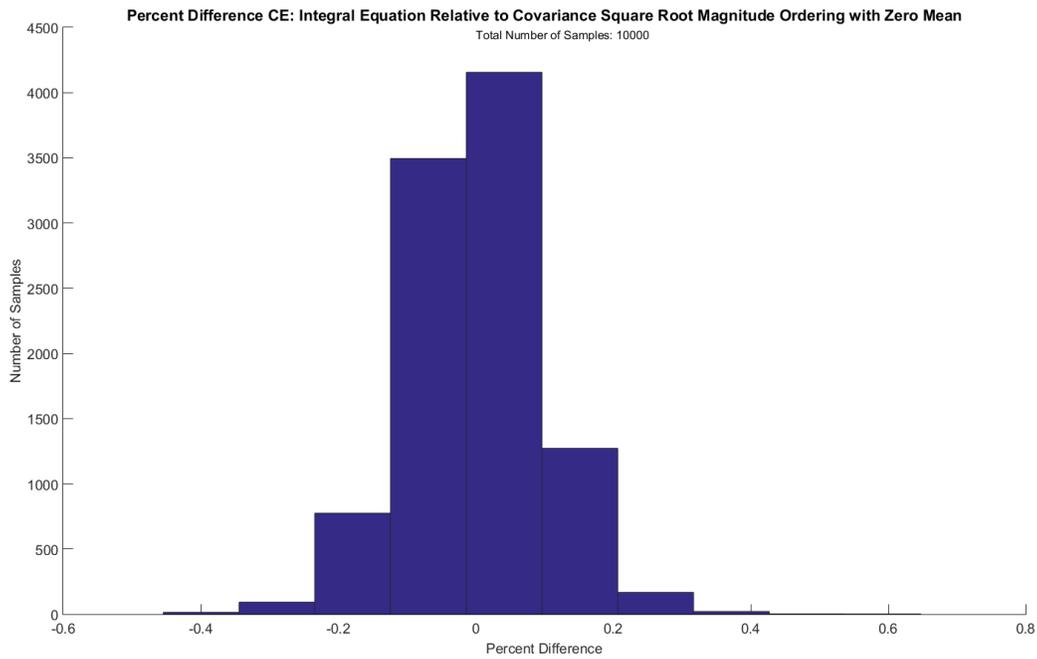
**C.1.1.2 CE Method versus Method Comparison Tests**

(1) Table\_Interp vs. IE (mean zero):10000 samples (2000 for each of 5 fixed probability levels)

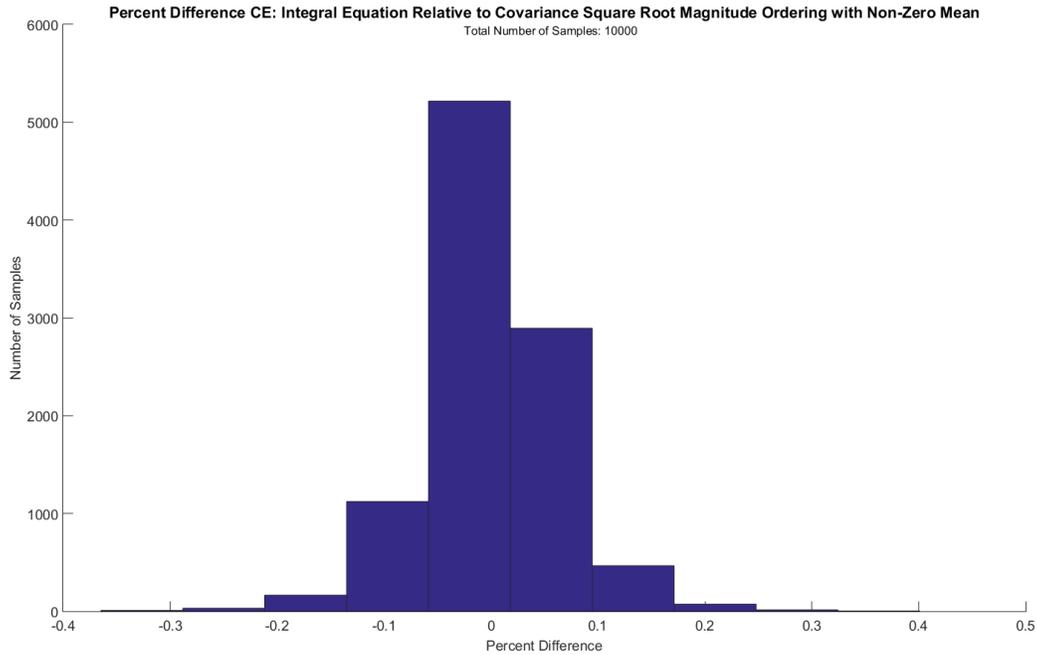


Note: if above prob  $\geq 0.9$ , max difference is 0.05.

(2) MC\_Cov\_sqrt vs. IE (mean zero):1000 samples



(3) MC\_Cov\_sqrt vs. IE (mean value arbitrary):1000 samples



### C.1.2 Spherical Error (SE)

For a given sample, the error covariance matrix and mean-value (if not zero) were randomly computed as follows:

The error covariance was randomly generated per sample about the mean-value as:

$$Cov_{random} = 4 \begin{bmatrix} u1 & u4\sqrt{u1 \times u2} & u5\sqrt{u1 \times u3} \\ . & u2 & u6\sqrt{u2 \times u3} \\ . & . & u3 \end{bmatrix}, \text{ where } u1, u2, \text{ and } u3 \text{ are random and independent}$$

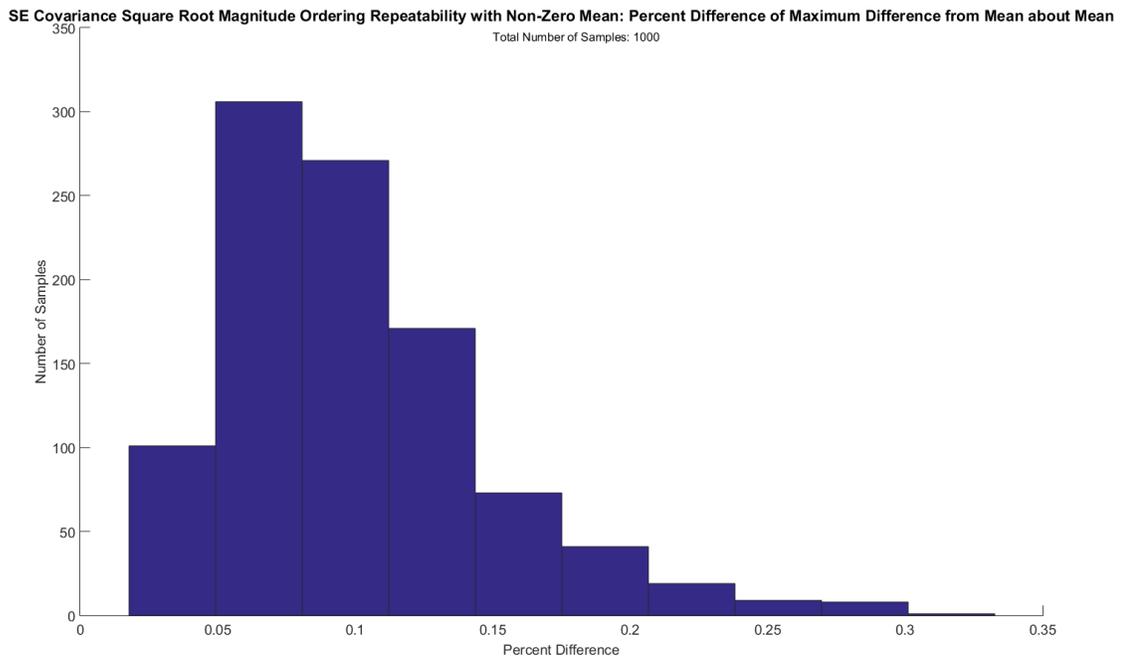
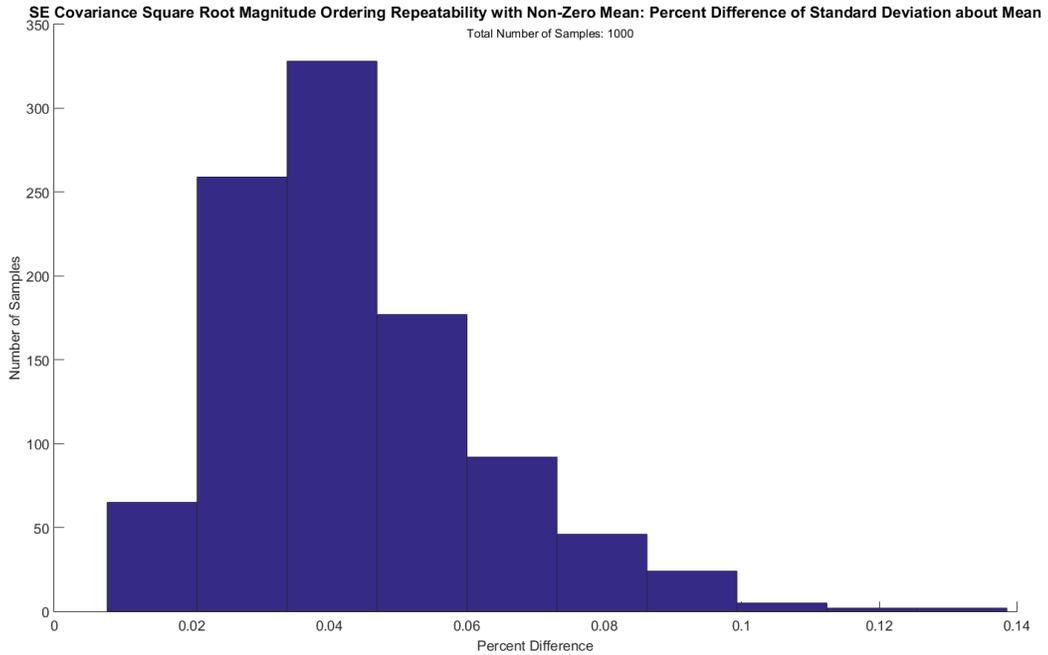
samples from a (0,1) uniform distribution, and u4, u5, and u6 are random and independent sample from a (-1,1) uniform distribution. The generated covariance matrix also checked/guaranteed positive definite. Also, the square root of the smallest to largest eigenvalue was also guaranteed/checked as  $r \geq 0.00001, 0.02, \text{ and } 0.10$  for tests involving MC\_Cov\_sqrt, IE (mean zero), and IE (mean not zero), respectively.

The corresponding mean-value is computed as  $\bar{X}_{random} = 4 \begin{bmatrix} u7 \\ u8 \\ u9 \end{bmatrix}$ , where u7, u8, and u9 are random and independent samples from a (-1,1) uniform distribution.

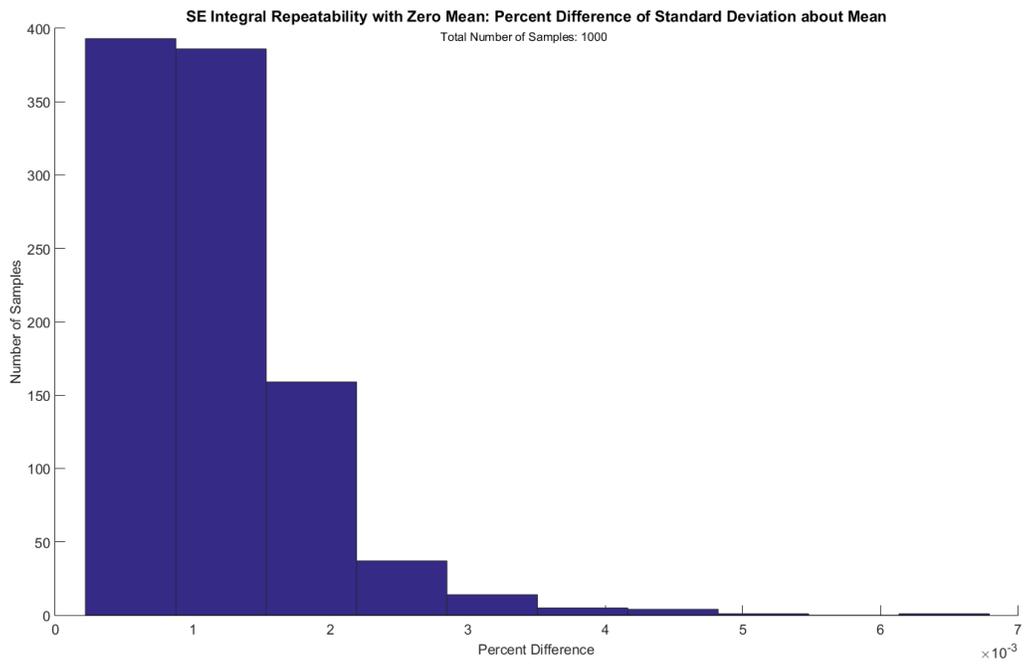
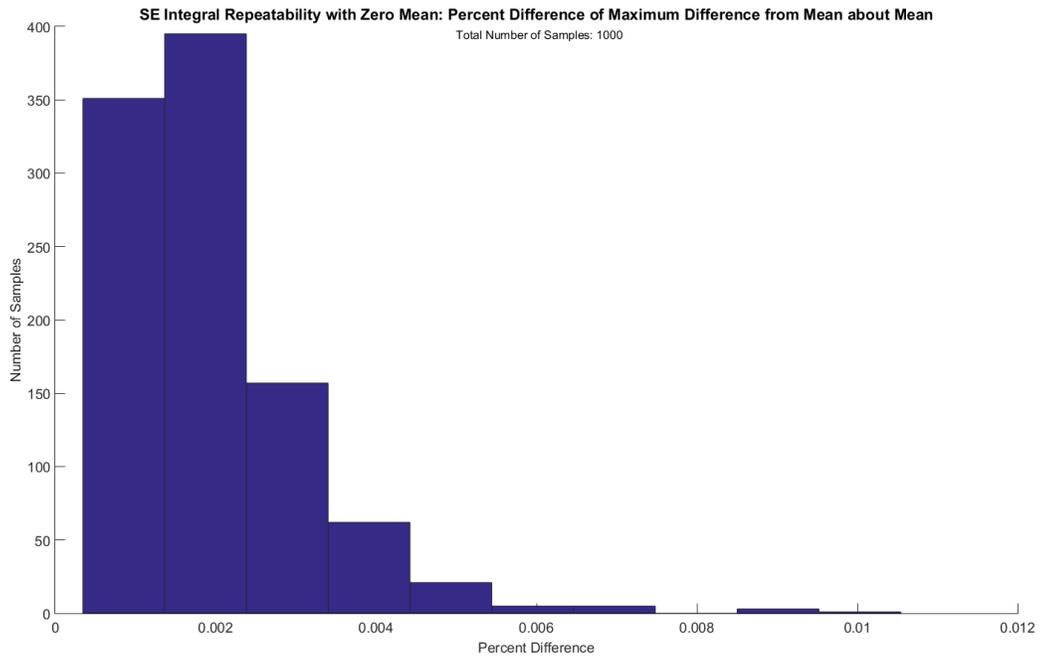
**C.1.2.1 SE Method Repeatability Tests**

(1) Table\_Interp: not applicable

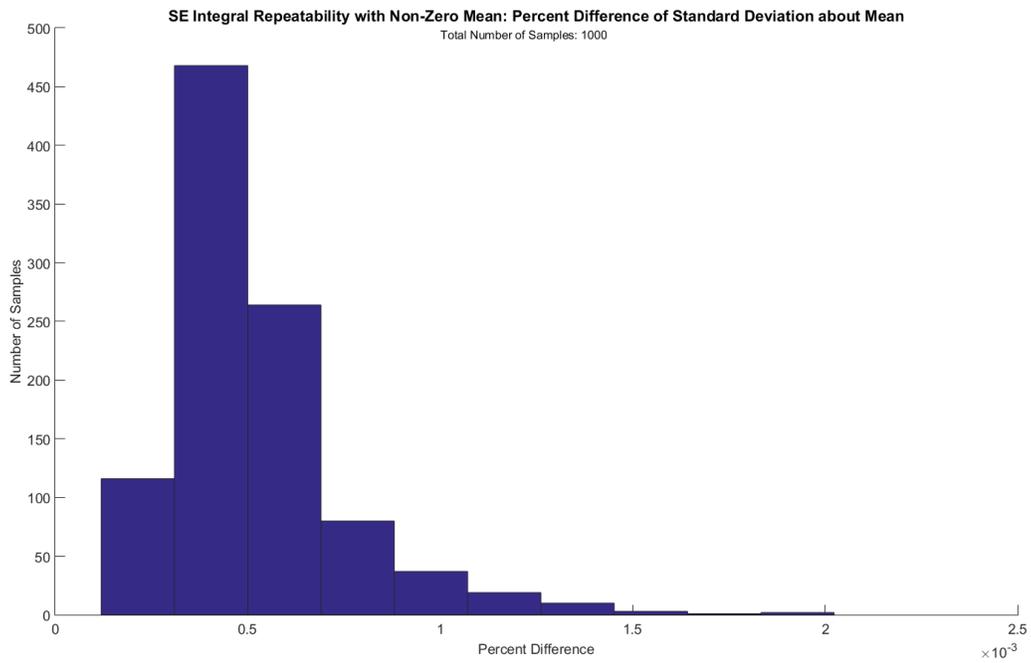
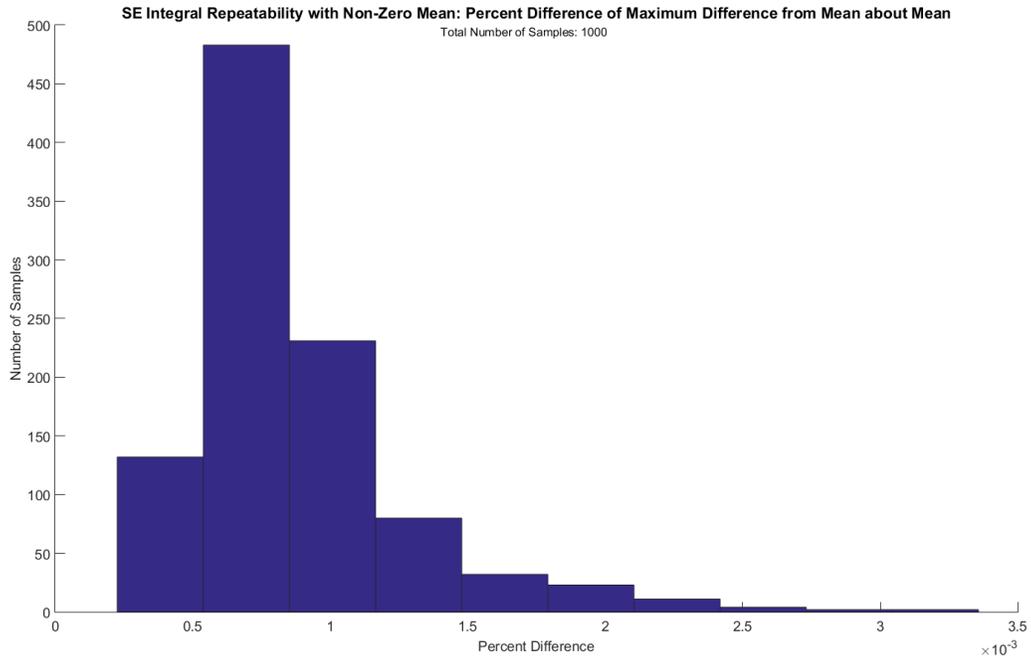
(2) MC\_Cov\_sqrt:1000 samples and 50 perturbations per sample



(3)IE (mean zero):1000 samples and 10 perturbations per sample

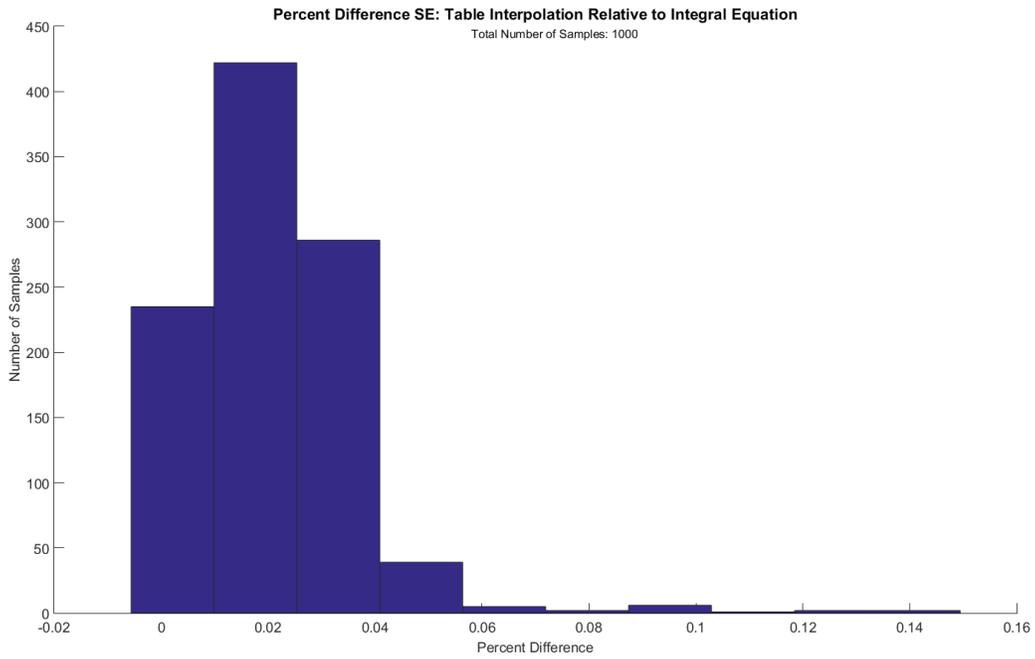


(4) IE (mean value arbitrary):1000 samples and 10 perturbations per sample



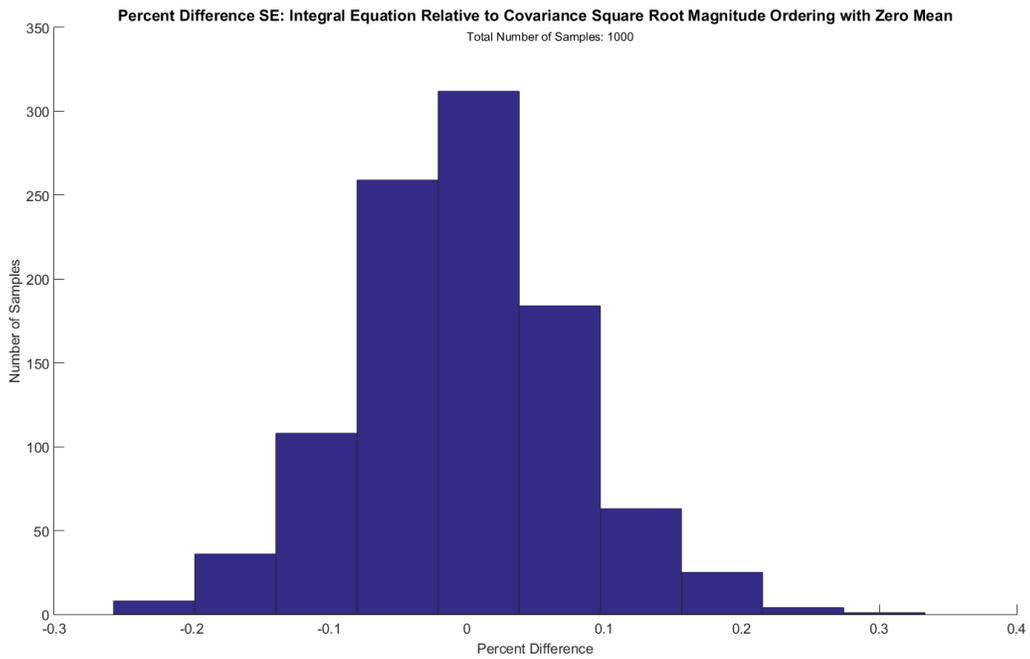
**C.1.2.2 SE Method versus Method Comparison Tests**

(1) Table\_Interp vs. IE (mean zero):1000 samples

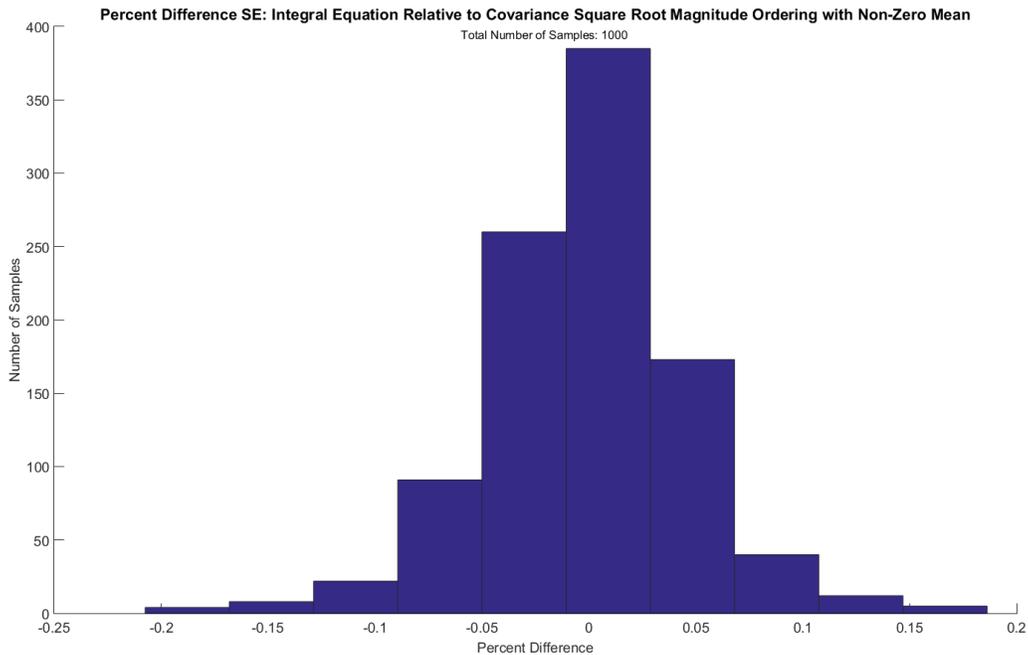


Note if  $p \geq 0.9$ , max is 0.05

(2) MC\_Cov\_sqrt vs. IE (mean zero):1000 samples



(3) MC\_Cov\_sqrt vs. IE (mean value arbitrary):1000 samples



### C.1.3 Summary of Tests

In the above Method Comparison tests, results were consistent with the repeatability tests for both methods involved – the difference was basically consistent with the repeatability test for the method with statistically larger computation errors.

Regarding Method Repeatability tests, there were variations (on the order of a factor of 2 or 3) between Integral Equation repeatability test result subcases: mean value equals zero and mean value arbitrary. This was primarily due to limited statistical significance associated with the number of samples and perturbations. (They were kept to reasonable numbers for reasonable test times.) Thus, when summarizing results in the main body, the “worst” results were reported across mean-zero and mean-not zero variations.

Although not specifically shown in the above test results, during the various comparison test results, it became apparent that the solution for the Integral Equation Method did not always converge. This problem occurred about 0.2 % of the time when the mean-value was arbitrary, and where probability was allowed to vary within the interval [0.1,0.999]. When convergence did not occur, the resultant relative error could be as large as 40%. However, in virtually all cases, the problem could be detected by noting that the number of iterations in the MATLAB code equaled the (specified) maximum and/or the function code equaled more than twice the number of iterations.

**C.1.4 Miscellaneous other Tests**

Two other categories of calculation error tests were performed:  $p=0.999$  and LE Integration. The former held the probability level fixed at the extreme high level, and was performed for both Table\_Int versus IE (mean zero) and MC\_Cov\_sqrt repeatability. Although the number of test samples were limited for convenience, test results were consistent with the previous test results where probability levels varied. The LE Integration test was for arbitrary mean values and provided results as good as for CE Integration and had no convergence issues.

In addition to calculation error performance, timing results for all of the various methods were also tabulated with results summarized in Table 5.4.2.6-1 (CE\_XX) and Table 5.4.3.5-1 (SE\_XX).

**C.2 Complete Set of SE Interpolation Tables**

Assuming an appropriate digital copy of this document, the values in the following five tables can be copied by highlighting the desired cells and copying (right clicking and selecting copy or pressing Ctrl and C simultaneously). Then pasting (right clicking and selecting paste or pressing Ctrl and V simultaneously) where desired.) This allows for direct insertion into appropriate pseudo-code arrays/files.

Table C.2-1: Spherical Error (SE) multiplier  $R(p = 0.5, r_1, r_2)$  versus ratios  $r_1$  and  $r_2$

		$r_1$																				
		0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
L	0.00	0.6745	0.6763	0.6820	0.6916	0.7059	0.7254	0.7499	0.7779	0.8079	0.8389	0.8704	0.9021	0.9337	0.9651	0.9962	1.0271	1.0577	1.0880	1.1181	1.1479	1.1774
	0.05	0.6763	0.6782	0.6838	0.6934	0.7076	0.7271	0.7516	0.7795	0.8094	0.8404	0.8719	0.9035	0.9350	0.9664	0.9975	1.0283	1.0589	1.0891	1.1192	1.1489	1.1784
	0.10	0.6820	0.6838	0.6894	0.6989	0.7130	0.7324	0.7567	0.7844	0.8141	0.8449	0.8762	0.9077	0.9390	0.9703	1.0013	1.0320	1.0625	1.0926	1.1225	1.1522	1.1817
	0.15	0.6916	0.6934	0.6989	0.7084	0.7223	0.7414	0.7654	0.7927	0.8221	0.8526	0.8836	0.9147	0.9459	0.9768	1.0077	1.0381	1.0684	1.0984	1.1282	1.1578	1.1870
	0.20	0.7059	0.7076	0.7130	0.7223	0.7359	0.7546	0.7781	0.8048	0.8336	0.8636	0.8941	0.9248	0.9556	0.9862	1.0167	1.0469	1.0769	1.1067	1.1362	1.1655	1.1947
	0.25	0.7254	0.7271	0.7324	0.7414	0.7546	0.7727	0.7952	0.8211	0.8491	0.8783	0.9081	0.9382	0.9684	0.9986	1.0286	1.0584	1.0881	1.1174	1.1466	1.1756	1.2045
	0.30	0.7499	0.7516	0.7567	0.7654	0.7781	0.7952	0.8167	0.8414	0.8684	0.8966	0.9256	0.9549	0.9844	1.0140	1.0434	1.0728	1.1019	1.1309	1.1597	1.1883	1.2168
	0.35	0.7779	0.7795	0.7844	0.7927	0.8048	0.8211	0.8414	0.8651	0.8909	0.9181	0.9420	0.9675	0.9939	1.0210	1.0484	1.0760	1.1036	1.1313	1.1588	1.1863	1.2137
	0.40	0.8079	0.8094	0.8141	0.8221	0.8336	0.8491	0.8684	0.8909	0.9157	0.9420	0.9692	0.9970	1.0251	1.0533	1.0814	1.1096	1.1376	1.1656	1.1934	1.2211	1.2488
	0.45	0.8389	0.8404	0.8449	0.8526	0.8636	0.8783	0.8966	0.9181	0.9420	0.9675	0.9939	1.0210	1.0484	1.0760	1.1036	1.1313	1.1588	1.1863	1.2137	1.2409	1.2681
	0.50	0.8704	0.8719	0.8762	0.8836	0.8941	0.9081	0.9256	0.9462	0.9692	0.9939	1.0197	1.0462	1.0730	1.1002	1.1273	1.1545	1.1816	1.2086	1.2356	1.2625	1.2893
	0.55	0.9021	0.9035	0.9077	0.9147	0.9248	0.9382	0.9549	0.9748	0.9970	1.0210	1.0462	1.0722	1.0985	1.1251	1.1519	1.1788	1.2055	1.2322	1.2589	1.2854	1.3119
	0.60	0.9337	0.9350	0.9390	0.9459	0.9556	0.9684	0.9844	1.0035	1.0251	1.0484	1.0730	1.0985	1.1245	1.1508	1.1772	1.2037	1.2302	1.2567	1.2830	1.3093	1.3355
	0.65	0.9651	0.9664	0.9703	0.9768	0.9862	0.9986	1.0140	1.0324	1.0533	1.0760	1.1002	1.1251	1.1508	1.1767	1.2029	1.2291	1.2554	1.2817	1.3078	1.3339	1.3599
	0.70	0.9962	0.9975	1.0013	1.0077	1.0167	1.0286	1.0434	1.0612	1.0814	1.1036	1.1273	1.1519	1.1772	1.2029	1.2288	1.2549	1.2810	1.3070	1.3330	1.3590	1.3848
	0.75	1.0271	1.0283	1.0320	1.0381	1.0469	1.0584	1.0728	1.0899	1.1096	1.1313	1.1545	1.1788	1.2037	1.2291	1.2549	1.2807	1.3067	1.3325	1.3585	1.3843	1.4101
	0.80	1.0577	1.0589	1.0625	1.0684	1.0769	1.0881	1.1019	1.1185	1.1376	1.1588	1.1816	1.2055	1.2302	1.2554	1.2810	1.3067	1.3324	1.3582	1.3840	1.4098	1.4355
	0.85	1.0880	1.0891	1.0926	1.0984	1.1067	1.1174	1.1309	1.1470	1.1656	1.1863	1.2086	1.2322	1.2567	1.2817	1.3070	1.3325	1.3582	1.3840	1.4098	1.4356	1.4611
	0.90	1.1181	1.1192	1.1225	1.1282	1.1362	1.1466	1.1597	1.1753	1.1934	1.2137	1.2356	1.2589	1.2830	1.3078	1.3330	1.3585	1.3840	1.4098	1.4355	1.4612	1.4869
	0.95	1.1479	1.1489	1.1522	1.1578	1.1655	1.1756	1.1883	1.2035	1.2211	1.2409	1.2625	1.2854	1.3093	1.3339	1.3590	1.3843	1.4098	1.4356	1.4612	1.4869	1.5125
1.00	1.1774	1.1784	1.1817	1.1870	1.1947	1.2045	1.2168	1.2315	1.2488	1.2681	1.2893	1.3119	1.3355	1.3599	1.3848	1.4101	1.4355	1.4611	1.4869	1.5125	1.5382	

NGA.SIG.0026.03\_1.0\_ACCPRED

Table C.2-2: Spherical Error (SE) multiplier  $R(p = 0.9, r_1, r_2)$  versus ratios  $r_1$  and  $r_2$

		$r_1$																				
		0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$r_2$	0.00	1.6449	1.6456	1.6479	1.6518	1.6573	1.6646	1.6738	1.6852	1.6992	1.7163	1.7371	1.7621	1.7915	1.8251	1.8625	1.9034	1.9472	1.9936	2.0424	2.0932	2.1460
	0.05	1.6456	1.6464	1.6487	1.6525	1.6581	1.6654	1.6745	1.6860	1.6999	1.7170	1.7378	1.7628	1.7922	1.8258	1.8632	1.9040	1.9478	1.9942	2.0429	2.0938	2.1466
	0.10	1.6479	1.6487	1.6509	1.6548	1.6604	1.6676	1.6769	1.6882	1.7021	1.7192	1.7400	1.7650	1.7944	1.8279	1.8652	1.9060	1.9497	1.9961	2.0448	2.0956	2.1483
	0.15	1.6518	1.6525	1.6548	1.6587	1.6642	1.6714	1.6806	1.6920	1.7059	1.7229	1.7436	1.7686	1.7979	1.8314	1.8687	1.9094	1.9530	1.9993	2.0479	2.0987	2.1512
	0.20	1.6573	1.6581	1.6604	1.6642	1.6697	1.6769	1.6861	1.6974	1.7113	1.7282	1.7489	1.7738	1.8030	1.8364	1.8735	1.9141	1.9576	2.0039	2.0523	2.1029	2.1555
	0.25	1.6646	1.6654	1.6676	1.6714	1.6769	1.6841	1.6932	1.7045	1.7183	1.7352	1.7558	1.7806	1.8097	1.8429	1.8799	1.9204	1.9638	2.0098	2.0581	2.1086	2.1610
	0.30	1.6738	1.6745	1.6769	1.6806	1.6861	1.6932	1.7023	1.7135	1.7273	1.7441	1.7646	1.7892	1.8182	1.8513	1.8881	1.9283	1.9715	2.0173	2.0654	2.1156	2.1678
	0.35	1.6852	1.6860	1.6882	1.6920	1.6974	1.7045	1.7135	1.7247	1.7383	1.7550	1.7755	1.7999	1.8286	1.8614	1.8981	1.9380	1.9809	2.0265	2.0743	2.1243	2.1762
	0.40	1.6992	1.6999	1.7021	1.7059	1.7113	1.7183	1.7273	1.7383	1.7519	1.7685	1.7887	1.8130	1.8414	1.8740	1.9102	1.9498	1.9923	2.0375	2.0850	2.1347	2.1862
	0.45	1.7163	1.7170	1.7192	1.7229	1.7282	1.7352	1.7441	1.7550	1.7685	1.7849	1.8049	1.8289	1.8569	1.8890	1.9248	1.9639	2.0060	2.0506	2.0977	2.1469	2.1981
	0.50	1.7371	1.7378	1.7400	1.7436	1.7489	1.7558	1.7646	1.7755	1.7887	1.8049	1.8245	1.8481	1.8757	1.9071	1.9422	1.9807	2.0221	2.0663	2.1127	2.1614	2.2120
	0.55	1.7621	1.7628	1.7650	1.7686	1.7738	1.7806	1.7892	1.7999	1.8130	1.8289	1.8481	1.8710	1.8979	1.9287	1.9630	2.0007	2.0413	2.0847	2.1304	2.1783	2.2282
	0.60	1.7915	1.7922	1.7944	1.7979	1.8030	1.8097	1.8182	1.8286	1.8414	1.8569	1.8757	1.8979	1.9240	1.9539	1.9873	2.0240	2.0637	2.1061	2.1510	2.1980	2.2472
	0.65	1.8251	1.8258	1.8279	1.8314	1.8364	1.8429	1.8513	1.8614	1.8740	1.8890	1.9071	1.9287	1.9539	1.9827	2.0151	2.0507	2.0894	2.1308	2.1746	2.2207	2.2689
	0.70	1.8625	1.8632	1.8652	1.8687	1.8735	1.8799	1.8881	1.8981	1.9102	1.9248	1.9422	1.9630	1.9873	2.0151	2.0464	2.0809	2.1185	2.1587	2.2015	2.2464	2.2936
	0.75	1.9034	1.9040	1.9060	1.9094	1.9141	1.9204	1.9283	1.9380	1.9498	1.9639	1.9807	2.0007	2.0240	2.0507	2.0809	2.1143	2.1506	2.1898	2.2314	2.2753	2.3214
	0.80	1.9472	1.9478	1.9497	1.9530	1.9576	1.9638	1.9715	1.9809	1.9923	2.0060	2.0221	2.0413	2.0637	2.0894	2.1185	2.1506	2.1858	2.2237	2.2642	2.3070	2.3520
	0.85	1.9936	1.9942	1.9961	1.9993	2.0039	2.0098	2.0173	2.0265	2.0375	2.0506	2.0663	2.0847	2.1061	2.1308	2.1587	2.1898	2.2237	2.2605	2.2998	2.3415	2.3854
	0.90	2.0424	2.0429	2.0448	2.0479	2.0523	2.0581	2.0654	2.0743	2.0850	2.0977	2.1127	2.1304	2.1510	2.1746	2.2015	2.2314	2.2642	2.2998	2.3380	2.3786	2.4213
	0.95	2.0932	2.0938	2.0956	2.0987	2.1029	2.1086	2.1156	2.1243	2.1347	2.1469	2.1614	2.1783	2.1980	2.2207	2.2464	2.2753	2.3070	2.3415	2.3786	2.4180	2.4597
1.00	2.1460	2.1466	2.1483	2.1512	2.1555	2.1610	2.1678	2.1762	2.1862	2.1981	2.2120	2.2282	2.2472	2.2689	2.2936	2.3214	2.3520	2.3854	2.4213	2.4597	2.5003	

Table C.2-3: Spherical Error (SE) multiplier  $R(p = 0.95, r_1, r_2)$  versus ratios  $r_1$  and  $r_2$

		$r_1$																				
		0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$r_2$	0.00	1.9600	1.9606	1.9625	1.9658	1.9704	1.9765	1.9842	1.9937	2.0051	2.0190	2.0359	2.0564	2.0813	2.1111	2.1460	2.1858	2.2303	2.2791	2.3318	2.3881	2.4478
	0.05	1.9606	1.9612	1.9632	1.9664	1.9711	1.9771	1.9848	1.9943	2.0058	2.0197	2.0365	2.0570	2.0819	2.1117	2.1466	2.1864	2.2309	2.2796	2.3324	2.3887	2.4482
	0.10	1.9625	1.9632	1.9651	1.9683	1.9729	1.9791	1.9867	1.9962	2.0077	2.0215	2.0383	2.0589	2.0837	2.1135	2.1483	2.1881	2.2325	2.2813	2.3339	2.3902	2.4498
	0.15	1.9658	1.9664	1.9683	1.9716	1.9762	1.9823	1.9899	1.9994	2.0108	2.0247	2.0415	2.0620	2.0868	2.1165	2.1513	2.1910	2.2354	2.2841	2.3367	2.3929	2.4524
	0.20	1.9704	1.9711	1.9729	1.9762	1.9808	1.9868	1.9945	2.0039	2.0153	2.0292	2.0459	2.0664	2.0912	2.1208	2.1555	2.1952	2.2394	2.2880	2.3406	2.3967	2.4561
	0.25	1.9765	1.9771	1.9791	1.9823	1.9868	1.9929	2.0005	2.0099	2.0213	2.0351	2.0518	2.0722	2.0969	2.1265	2.1611	2.2006	2.2448	2.2932	2.3457	2.4016	2.4609
	0.30	1.9842	1.9848	1.9867	1.9899	1.9945	2.0005	2.0081	2.0175	2.0288	2.0425	2.0592	2.0795	2.1041	2.1336	2.1682	2.2075	2.2515	2.2998	2.3520	2.4078	2.4669
	0.35	1.9937	1.9943	1.9962	1.9994	2.0039	2.0099	2.0175	2.0268	2.0381	2.0518	2.0683	2.0885	2.1131	2.1425	2.1767	2.2160	2.2598	2.3079	2.3598	2.4154	2.4743
	0.40	2.0051	2.0058	2.0077	2.0108	2.0153	2.0213	2.0288	2.0381	2.0493	2.0630	2.0795	2.0995	2.1239	2.1531	2.1873	2.2262	2.2697	2.3175	2.3692	2.4246	2.4831
	0.45	2.0190	2.0197	2.0215	2.0247	2.0292	2.0351	2.0425	2.0518	2.0630	2.0764	2.0929	2.1129	2.1371	2.1660	2.1999	2.2385	2.2816	2.3291	2.3804	2.4353	2.4935
	0.50	2.0359	2.0365	2.0383	2.0415	2.0459	2.0518	2.0592	2.0683	2.0795	2.0929	2.1092	2.1290	2.1529	2.1816	2.2150	2.2532	2.2959	2.3429	2.3936	2.4481	2.5058
	0.55	2.0564	2.0570	2.0589	2.0620	2.0664	2.0722	2.0795	2.0885	2.0995	2.1129	2.1290	2.1485	2.1722	2.2004	2.2333	2.2708	2.3129	2.3592	2.4093	2.4631	2.5202
	0.60	2.0813	2.0819	2.0837	2.0868	2.0912	2.0969	2.1041	2.1131	2.1239	2.1371	2.1529	2.1722	2.1953	2.2229	2.2551	2.2919	2.3332	2.3786	2.4279	2.4809	2.5371
	0.65	2.1111	2.1117	2.1135	2.1165	2.1208	2.1265	2.1336	2.1425	2.1531	2.1660	2.1816	2.2004	2.2229	2.2497	2.2810	2.3168	2.3570	2.4014	2.4497	2.5017	2.5570
	0.70	2.1460	2.1466	2.1483	2.1513	2.1555	2.1611	2.1682	2.1767	2.1873	2.1999	2.2150	2.2333	2.2551	2.2810	2.3112	2.3460	2.3850	2.4281	2.4752	2.5259	2.5801
	0.75	2.1858	2.1864	2.1881	2.1910	2.1952	2.2006	2.2075	2.2160	2.2262	2.2385	2.2532	2.2708	2.2919	2.3168	2.3460	2.3794	2.4170	2.4589	2.5046	2.5539	2.6067
	0.80	2.2303	2.2309	2.2325	2.2354	2.2394	2.2448	2.2515	2.2598	2.2697	2.2816	2.2959	2.3129	2.3332	2.3570	2.3850	2.4170	2.4533	2.4937	2.5379	2.5858	2.6371
	0.85	2.2791	2.2796	2.2813	2.2841	2.2880	2.2932	2.2998	2.3079	2.3175	2.3291	2.3429	2.3592	2.3786	2.4014	2.4281	2.4589	2.4937	2.5325	2.5751	2.6214	2.6713
	0.90	2.3318	2.3323	2.3339	2.3367	2.3406	2.3457	2.3520	2.3598	2.3692	2.3804	2.3936	2.4093	2.4279	2.4497	2.4752	2.5046	2.5379	2.5751	2.6162	2.6609	2.7091
	0.95	2.3881	2.3887	2.3902	2.3929	2.3967	2.4016	2.4078	2.4154	2.4246	2.4353	2.4481	2.4631	2.4809	2.5017	2.5259	2.5539	2.5858	2.6214	2.6609	2.7040	2.7506
1.00	2.4478	2.4482	2.4498	2.4524	2.4561	2.4609	2.4669	2.4743	2.4831	2.4935	2.5058	2.5202	2.5371	2.5570	2.5801	2.6067	2.6371	2.6713	2.7091	2.7506	2.7955	

NGA.SIG.0026.03\_1.0\_ACCPRED

Table C.2-4: Spherical Error (SE) multiplier  $R(p = 0.99, r_1, r_2)$  versus ratios  $r_1$  and  $r_2$

		$r_1$																				
		0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$r_2$	0.00	2.5758	2.5763	2.5778	2.5803	2.5838	2.5884	2.5942	2.6013	2.6099	2.6203	2.6326	2.6474	2.6653	2.6875	2.7151	2.7492	2.7907	2.8401	2.8974	2.9625	3.0349
	0.05	2.5763	2.5768	2.5783	2.5808	2.5842	2.5888	2.5947	2.6018	2.6105	2.6207	2.6331	2.6479	2.6658	2.6880	2.7155	2.7496	2.7912	2.8405	2.8979	2.9629	3.0353
	0.10	2.5778	2.5783	2.5798	2.5822	2.5857	2.5903	2.5962	2.6033	2.6118	2.6222	2.6345	2.6493	2.6672	2.6894	2.7169	2.7510	2.7925	2.8418	2.8992	2.9642	3.0365
	0.15	2.5803	2.5808	2.5822	2.5847	2.5882	2.5928	2.5986	2.6057	2.6143	2.6246	2.6369	2.6516	2.6696	2.6917	2.7193	2.7533	2.7948	2.8441	2.9014	2.9664	3.0386
	0.20	2.5838	2.5842	2.5857	2.5882	2.5917	2.5963	2.6021	2.6092	2.6178	2.6281	2.6403	2.6551	2.6730	2.6951	2.7226	2.7566	2.7980	2.8473	2.9045	2.9694	3.0416
	0.25	2.5884	2.5888	2.5903	2.5928	2.5963	2.6008	2.6067	2.6138	2.6224	2.6326	2.6448	2.6595	2.6775	2.6995	2.7270	2.7609	2.8023	2.8515	2.9086	2.9734	3.0454
	0.30	2.5942	2.5947	2.5962	2.5986	2.6021	2.6067	2.6125	2.6195	2.6281	2.6383	2.6505	2.6652	2.6831	2.7051	2.7325	2.7664	2.8077	2.8567	2.9137	2.9784	3.0503
	0.35	2.6013	2.6018	2.6033	2.6057	2.6092	2.6138	2.6195	2.6266	2.6351	2.6453	2.6575	2.6722	2.6900	2.7120	2.7393	2.7731	2.8143	2.8632	2.9200	2.9845	3.0563
	0.40	2.6099	2.6105	2.6118	2.6143	2.6178	2.6224	2.6281	2.6351	2.6436	2.6539	2.6660	2.6806	2.6984	2.7202	2.7475	2.7812	2.8222	2.8710	2.9276	2.9919	3.0635
	0.45	2.6203	2.6207	2.6222	2.6246	2.6281	2.6326	2.6383	2.6453	2.6539	2.6639	2.6761	2.6906	2.7083	2.7302	2.7573	2.7909	2.8317	2.8803	2.9367	3.0007	3.0719
	0.50	2.6326	2.6331	2.6345	2.6369	2.6403	2.6448	2.6505	2.6575	2.6660	2.6761	2.6882	2.7027	2.7203	2.7421	2.7691	2.8024	2.8431	2.8913	2.9474	3.0110	3.0819
	0.55	2.6474	2.6479	2.6493	2.6516	2.6551	2.6595	2.6652	2.6722	2.6806	2.6906	2.7027	2.7171	2.7347	2.7563	2.7831	2.8163	2.8566	2.9045	2.9601	3.0233	3.0937
	0.60	2.6653	2.6658	2.6672	2.6696	2.6730	2.6775	2.6831	2.6900	2.6984	2.7083	2.7203	2.7347	2.7522	2.7736	2.8003	2.8330	2.8729	2.9203	2.9753	3.0378	3.1075
	0.65	2.6875	2.6880	2.6894	2.6917	2.6951	2.6995	2.7051	2.7120	2.7202	2.7302	2.7421	2.7563	2.7736	2.7949	2.8212	2.8535	2.8927	2.9393	2.9935	3.0552	3.1240
	0.70	2.7151	2.7155	2.7169	2.7193	2.7226	2.7270	2.7325	2.7393	2.7475	2.7573	2.7691	2.7831	2.8003	2.8212	2.8469	2.8786	2.9170	2.9625	3.0156	3.0760	3.1436
	0.75	2.7492	2.7496	2.7510	2.7533	2.7566	2.7609	2.7664	2.7731	2.7812	2.7909	2.8024	2.8163	2.8330	2.8535	2.8786	2.9093	2.9465	2.9908	3.0423	3.1012	3.1672
	0.80	2.7907	2.7912	2.7925	2.7948	2.7980	2.8023	2.8077	2.8143	2.8222	2.8317	2.8431	2.8566	2.8729	2.8927	2.9170	2.9465	2.9823	3.0248	3.0746	3.1316	3.1956
	0.85	2.8401	2.8405	2.8418	2.8441	2.8473	2.8515	2.8567	2.8632	2.8710	2.8803	2.8913	2.9045	2.9203	2.9393	2.9625	2.9908	3.0248	3.0655	3.1131	3.1677	3.2294
	0.90	2.8974	2.8978	2.8992	2.9014	2.9045	2.9086	2.9137	2.9200	2.9276	2.9367	2.9474	2.9601	2.9753	2.9935	3.0156	3.0423	3.0746	3.1131	3.1582	3.2102	3.2691
	0.95	2.9625	2.9629	2.9642	2.9664	2.9694	2.9734	2.9784	2.9845	2.9919	3.0007	3.0110	3.0233	3.0378	3.0552	3.0760	3.1012	3.1316	3.1677	3.2102	3.2594	3.3154
1.00	3.0349	3.0353	3.0365	3.0386	3.0416	3.0454	3.0503	3.0563	3.0635	3.0719	3.0819	3.0936	3.1075	3.1240	3.1436	3.1672	3.1956	3.2294	3.2691	3.3154	3.3682	

Table C.2-5: Spherical Error (SE) multiplier  $R(p = 0.999, r_1, r_2)$  versus ratios  $r_1$  and  $r_2$

		$r_1$																				
		0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
$r_2$	0.00	3.2905	3.2910	3.2921	3.2940	3.2967	3.3003	3.3049	3.3104	3.3172	3.3252	3.3346	3.3459	3.3595	3.3759	3.3965	3.4227	3.4570	3.5018	3.5594	3.6310	3.7169
	0.05	3.2910	3.2913	3.2924	3.2944	3.2972	3.3007	3.3053	3.3108	3.3175	3.3255	3.3350	3.3463	3.3599	3.3763	3.3968	3.4231	3.4573	3.5022	3.5598	3.6314	3.7173
	0.10	3.2921	3.2924	3.2936	3.2955	3.2982	3.3019	3.3064	3.3120	3.3182	3.3267	3.3361	3.3475	3.3610	3.3774	3.3979	3.4242	3.4584	3.5032	3.5608	3.6324	3.7183
	0.15	3.2940	3.2944	3.2955	3.2975	3.3002	3.3038	3.3084	3.3139	3.3206	3.3285	3.3380	3.3493	3.3629	3.3793	3.3998	3.4260	3.4603	3.5050	3.5626	3.6341	3.7199
	0.20	3.2967	3.2972	3.2982	3.3002	3.3029	3.3065	3.3111	3.3166	3.3233	3.3313	3.3407	3.3520	3.3655	3.3820	3.4025	3.4287	3.4629	3.5076	3.5651	3.6366	3.7224
	0.25	3.3003	3.3007	3.3019	3.3038	3.3065	3.3101	3.3147	3.3202	3.3269	3.3348	3.3444	3.3556	3.3691	3.3856	3.4060	3.4322	3.4663	3.5111	3.5685	3.6399	3.7256
	0.30	3.3049	3.3052	3.3064	3.3083	3.3111	3.3147	3.3192	3.3247	3.3314	3.3394	3.3488	3.3600	3.3736	3.3900	3.4104	3.4365	3.4707	3.5153	3.5726	3.6440	3.7296
	0.35	3.3104	3.3108	3.3120	3.3139	3.3166	3.3202	3.3247	3.3303	3.3370	3.3449	3.3543	3.3655	3.3790	3.3954	3.4158	3.4419	3.4760	3.5205	3.5778	3.6490	3.7344
	0.40	3.3172	3.3174	3.3187	3.3205	3.3233	3.3269	3.3314	3.3370	3.3436	3.3515	3.3609	3.3721	3.3856	3.4019	3.4223	3.4483	3.4823	3.5268	3.5840	3.6550	3.7403
	0.45	3.3252	3.3254	3.3267	3.3286	3.3313	3.3348	3.3394	3.3449	3.3515	3.3594	3.3688	3.3800	3.3934	3.4097	3.4301	3.4560	3.4900	3.5343	3.5913	3.6622	3.7473
	0.50	3.3346	3.3349	3.3361	3.3380	3.3407	3.3444	3.3488	3.3543	3.3609	3.3688	3.3782	3.3894	3.4027	3.4190	3.4393	3.4652	3.4990	3.5433	3.6001	3.6707	3.7554
	0.55	3.3459	3.3463	3.3474	3.3493	3.3520	3.3556	3.3600	3.3655	3.3721	3.3800	3.3894	3.4004	3.4138	3.4301	3.4503	3.4761	3.5099	3.5539	3.6105	3.6807	3.7651
	0.60	3.3595	3.3598	3.3609	3.3629	3.3655	3.3691	3.3736	3.3790	3.3856	3.3934	3.4027	3.4138	3.4271	3.4433	3.4635	3.4892	3.5228	3.5667	3.6228	3.6927	3.7765
	0.65	3.3759	3.3763	3.3774	3.3793	3.3819	3.3856	3.3900	3.3954	3.4019	3.4097	3.4190	3.4301	3.4433	3.4594	3.4795	3.5052	3.5386	3.5821	3.6378	3.7070	3.7901
	0.70	3.3965	3.3966	3.3979	3.3998	3.4024	3.4060	3.4104	3.4158	3.4223	3.4301	3.4393	3.4503	3.4635	3.4795	3.4995	3.5250	3.5581	3.6011	3.6561	3.7244	3.8064
	0.75	3.4227	3.4230	3.4242	3.4260	3.4286	3.4322	3.4365	3.4419	3.4483	3.4561	3.4652	3.4761	3.4892	3.5052	3.5248	3.5502	3.5827	3.6250	3.6789	3.7458	3.8264
	0.80	3.4570	3.4573	3.4584	3.4602	3.4628	3.4663	3.4706	3.4760	3.4823	3.4900	3.4990	3.5099	3.5228	3.5386	3.5580	3.5827	3.6145	3.6555	3.7078	3.7729	3.8512
	0.85	3.5018	3.5022	3.5032	3.5051	3.5076	3.5111	3.5153	3.5205	3.5268	3.5343	3.5433	3.5539	3.5667	3.5821	3.6008	3.6250	3.6555	3.6948	3.7448	3.8071	3.8826
	0.90	3.5594	3.5597	3.5608	3.5626	3.5651	3.5684	3.5726	3.5778	3.5840	3.5913	3.6001	3.6105	3.6228	3.6377	3.6560	3.6789	3.7078	3.7448	3.7919	3.8507	3.9223
	0.95	3.6310	3.6313	3.6323	3.6341	3.6367	3.6399	3.6440	3.6490	3.6550	3.6622	3.6707	3.6807	3.6927	3.7070	3.7244	3.7458	3.7729	3.8071	3.8507	3.9053	3.9720
1.00	3.7169	3.7173	3.7182	3.7200	3.7224	3.7256	3.7296	3.7344	3.7403	3.7473	3.7554	3.7651	3.7765	3.7901	3.8064	3.8264	3.8512	3.8826	3.9223	3.9720	4.0332	

### C.3 Pseudo-code

The pseudo-code (MATLAB) was run on the following “notebook” computer: Dell Precision T5810 (desktop) with Intel® Xeon® CPU E5-1607 v3 @ 3.10GHz and 16GB of RAM. The Computer has four cores, but programs are single threaded, i.e., no multi-threading was implemented. MATLAB R2015a (Version 8.5.0.197613) was used.

#### C.3.1 Pseudo-code for LE

Algorithm (5.4.1.1-2), i.e., the second algorithm listed in Section 5.4.1.1:

```
function LEcal = LEerf(covar,prob)
% Calculate linear error distance analytically
%
% Inputs: covar - 1x1 variance value
%         prob - probability that linear distance will be calculated to
%
% Outputs: LEcal - calculated linear error distance

% Uses error function inverse to calculate linear error distance
LEcal = sqrt(covar)*sqrt(2)*erfinv(prob);
end % ends LE erf function
```

Algorithm (5.4.1.1-3):

```
function LEcal = LEintegral(covar,mCoord,prob)
% Calculate linear error distance analytically
%
% Inputs: covar - 1x1 variance value
%         mCoord - 1x1 mean coordinate of error
%         prob - probability that linear distance will be calculated to
%
% Outputs: LEcal - calculated linear error distance

% Get initial very crude approximation
if prob > .95 % checks probability value
    LEapprox = 1.96*sqrt(covar); % approximation from variance value
elseif prob <=0.95 && prob > .5 % checks probability value
    LEapprox = 1*sqrt(covar); % approximation from variance value
else % checks probability value
    LEapprox = .5*sqrt(covar); % approximation from variance value
end % ends loop checking variance value
LEapprox = abs(mCoord)+LEapprox; % updates approximation

% Calculate error radius
% function handle for determining distance value
initD = @(D) (LEint(covar,mCoord,D)-prob)^2;
% finds local minimum for solution of linear error distance
LEcal = fminsearch(initD,LEapprox);
end % ends LE analytical function

function LErad = LEint(covar,mCoord,D)
```

## NGA.SIG.0026.03\_1.0\_ACCPRED

```
%%%%%%%% Returns the integral of the bivariate gaussian pdf with Mean mCoord
%%%%%%%% and Covariance covar about the Mean bounded by the Distance D
%%%%%%%%
%%%%%%%% Inputs: covar - 1x1 variance value
%%%%%%%%           mCoord - 1x1 mean coordinate of error
%%%%%%%%           D - linear error distance
%%%%%%%%
%%%%%%%% Outputs: LEint - evaluated integral at current distance D

%%%%%%%% Function handle for the gaussian pdf 1D exponent
expMult = @(z) (-1/2)*((z-mCoord).^2)/covar;

%%%%%%%% Function handle for the gaussian pdf equation to be integrated
guasEq = @(z) 1/(sqrt(2*pi)*sqrt(covar))*exp(expMult(z));

%%%%%%%% Uses MATLAB function to integrate
LErad = integral(guasEq,-D,D);
end % ends LEint function
```

### C.3.2 Pseudo-code for CE

Algorithm (5.4.2.3-1), i.e. the first algorithm listed in Section 5.4.2.3:

```
function CEcal = CEtableV2(covar,prob)
%%%%%%%% Interpolates multiplier from table
%%%%%%%%
%%%%%%%% Inputs: covar - 2x2 covariance matrix
%%%%%%%%           prob - probability at which multiplier will be calculated
%%%%%%%%
%%%%%%%% Outputs: tableVal - interpolated multiplier from table

%%%%%%%% Table of values for CE interpolation
CE = [0.6745  1.6449  1.9600  2.5758  3.2905
      0.6763  1.6456  1.9606  2.5763  3.2910
      0.6820  1.6479  1.9625  2.5778  3.2921
      0.6916  1.6518  1.9658  2.5803  3.2940
      0.7059  1.6573  1.9704  2.5838  3.2967
      0.7254  1.6646  1.9765  2.5884  3.3003
      0.7499  1.6738  1.9842  2.5942  3.3049
      0.7779  1.6852  1.9937  2.6013  3.3104
      0.8079  1.6992  2.0051  2.6099  3.3172
      0.8389  1.7163  2.0190  2.6203  3.3252
      0.8704  1.7371  2.0359  2.6326  3.3346
      0.9021  1.7621  2.0564  2.6474  3.3459
      0.9337  1.7915  2.0813  2.6653  3.3595
      0.9651  1.8251  2.1111  2.6875  3.3759
      0.9962  1.8625  2.1460  2.7151  3.3965
      1.0271  1.9034  2.1858  2.7492  3.4227
      1.0577  1.9472  2.2303  2.7907  3.4570
      1.0880  1.9936  2.2791  2.8401  3.5018
      1.1181  2.0424  2.3318  2.8974  3.5594
      1.1479  2.0932  2.3881  2.9625  3.6310
      1.1774  2.1460  2.4478  3.0349  3.7169];
%%%%%%%% Checks entered probability to determine column of table to use
```

## NGA.SIG.0026.03\_1.0\_ACCPRED

```
if prob == .5 % checks entered probability
    CEuse = CE(:,1); % defines part of table to use
elseif prob == .9 % checks entered probability
    CEuse = CE(:,2); % defines part of table to use
elseif prob == .95 % checks entered probability
    CEuse = CE(:,3); % defines part of table to use
elseif prob == .99 % checks entered probability
    CEuse = CE(:,4); % defines part of table to use
elseif prob == .999 % checks entered probability
    CEuse = CE(:,5); % defines part of table to use
else % checks entered probability
    fprintf('Entered probability is not one of the options for Table
            Interpolation.\n')
    quit % quits program
end % ends loop checking probability

eigVal = sort(eig(covar)); % calcs covariance eigen values
if eigVal(1) <= 0 % checks minimum eigen value
    fprintf('Entered covariance matrix is not positive definite.\n')
    quit % quits program
end
ratio = sqrt(eigVal(1)/eigVal(2)); % ratio of eigen values

tableVal = interp1(0:.05:1,CEuse,ratio); % linear interpolation from table

CEcal = sqrt(eigVal(2))*tableVal; % calcs CE radius
end % ends function
```

Algorithm (5.4.2.3-2):

```
function CEcal = CECovSqrt(covar,mCoord,prob)
%%%%% Calculates circular error radius using covariance square root
%%%%% magnitude ordering
%%%%%
%%%%% Inputs: covar - 2x2 covariance matrix
%%%%%           mCoord - 2x1 mean coordinate of errors
%%%%%           prob - probability that circular radius will be calculated to
%%%%%
%%%%% Outputs: CEcal - calculated circular error radius

%%%%% Check entered probability
if prob < 0.1
    fprintf('Entered probability is smaller than recommended limit.\n')
end

%%%%% Check eigen value ratio
eigVal = sort(eig(covar));
if eigVal(1) <= 0
    fprintf('Entered covariance matrix is not positive definite.\n')
    quit
elseif sqrt(eigVal(1)/eigVal(2)) < 0.0001
    fprintf('Eigen value ratio is smaller than recommended limit.\n')
end
```

## NGA.SIG.0026.03\_1.0\_ACCPRED

```

numSamp = 1e6; % number of samples to be evaluated
X = sqrtm(covar)*randn(2,numSamp); % matrix of random samples
X = X+mCoord*ones(1,numSamp); % translates random samples by mean value
mag = sort(sum(X.^2)); % sorted magnitude of samples from origin
numProb = floor(numSamp*prob); % number of samples for probability
CEcal = sum(sqrt(mag(numProb:numProb+1)))/2; % circular radius
end % ends function

```

### Algorithm (5.4.2.3-3):

```

function [CEcal,output] = CEintegral(covar,mCoord,prob)
% Calculate circular error radius analytically
% Inputs: covar - 2x2 covariance matrix
%          mCoord - 2x1 mean coordinate of errors
%          prob - probability that spherical radius will be calculated to
% Outputs: CEcal - calculated circular error radius

% Convert covariance to eigen value space
[eigVec,covar] = svd(covar);
mCoord = transpose(eigVec)*mCoord;

% Get initial very crude approximation
if prob > .95 % checks probability value
    CEapprox = 2.25*sqrt(mean(diag(covar))); % approximation from mean
elseif prob <=0.95 && prob > .5 % checks probability value
    CEapprox = 1.25*sqrt(mean(diag(covar))); % approximation from mean
else % action based on probability
    CEapprox = .5*sqrt(mean(diag(covar))); % approximation from mean
end % ends loop checking prob
CEapprox = sqrt(sum(mCoord.^2))+CEapprox; % updates approximation

% Calculate error radius
initR = @(R) (CEint(covar,mCoord,R)-prob)^2; % function handle

% Sets maximum number of iterations for search function.
% Number of iterations could be optimized.
if prob < .95 % checks probability
    opt = optimset('MaxIter',20); % option for minimization function
else % checks probability
    opt = optimset('MaxIter',22); % option for minimization function
end % ends loop checking probability
[CEcal(1,1),~,~,output] = fminsearch(initR,CEapprox,opt); % finds radius

```

## NGA.SIG.0026.03\_1.0\_ACCPRED

```

%%%%% Prints warning statements about results based on iteration and
%%%%% function counts.
if output.iterations == 20 && prob < .95
    fprintf('Iteration limit was reached. Result may not be valid.\n')
elseif output.iterations == 22
    fprintf('Iteration limit was reached. Result may not be valid.\n')
elseif output.funcCount > 2*output.iterations
    fprintf('Function count is greater than twice the iteration count. ')
    fprintf('Result most like not valid.\n')
end
end % ends CE integral function

function CERad = CEint(covar,mCoord,R)
%%%%% Returns the integral of the bivariate Gaussian pdf with Mean mCoord
%%%%% and Covariance covar about the Mean bounded by the circle with Radius R
%%%%%
%%%%% Inputs: covar - 2x2 covariance matrix
%%%%%           mCoord - 2x1 mean coordinate of errors
%%%%%           R - radius of circular error
%%%%%
%%%%% Outputs: CEint - evaluated integral at current radius R

detCov = det(covar); % determinate of covariance matrix
invCov = inv(covar); % inverse of covariance matrix

%%%%% Function handle for the Gaussian pdf 2D exponent
expMult = @(x,y) invCov(1,1)*(x-mCoord(1)).^2+...
            2*invCov(1,2)*(x-mCoord(1)).*(y-mCoord(2))+...
            invCov(2,2)*(y-mCoord(2)).^2;

%%%%% Function handle for the Gaussian pdf equation to be integrated
guasEq = @(x,y) 1/((2*pi)*sqrt(detCov))*exp(-expMult(x,y)/2);

%%%%% Function handles for integration limits
ymin = @(x) -sqrt(R^2-x.^2); % function for lower y limit
ymax = @(x) sqrt(R^2-x.^2); % function for upper y limit

%%%%% Uses MATLAB function to integrate across the two variables
CERad = integral2(guasEq,-R,R,ymin,ymax);
end % ends CEint function

```

### C.3.3 Pseudo-code for SE

Algorithm (5.4.3.3-1), i.e., first algorithm listed in Section 5.4.3.3:

```
function SEcal = SEtableV2(covar,prob)
%%%%% Interpolates multiplier from table
%%%%%
%%%%% Inputs: covar - 3x3 covariance matrix
%%%%%          prob - probability at which multiplier will be calculated
%%%%%
%%%%% Outputs: SEcal - calculated SE radius value

%%%%% Tables of values for SE interpolation
SE50    = [];
SE90    = [];
SE95    = [];
SE99    = [];
SE999   = [];

%%%%% Checks entered probability to determine column of table to use
if prob == .5           % checks entered probability
    SEuse = SE50;       % defines part of table to use
elseif prob == .9      % checks entered probability
    SEuse = SE90;       % defines part of table to use
elseif prob == .95     % checks entered probability
    SEuse = SE95;       % defines part of table to use
elseif prob == .99     % checks entered probability
    SEuse = SE99;       % defines part of table to use
elseif prob == .999    % checks entered probability
    SEuse = SE999;      % defines part of table to use
else                    % checks entered probability
    fprintf('Entered probability is not one of the options for Table
Interpolation.\n')
    quit                % quits program
end                    % ends loop checking probability

eigVal = sort(eig(covar)); % calc covariance eigen values
if eigVal(1) <= 0        % checks min eigen value
    fprintf('Entered covariance matrix is not positive definite.\n')
    quit                % quits program
end                    % ends loop checking eigen value
r1      = sqrt(eigVal(2)/eigVal(3)); % ratio of eigen values
r2      = sqrt(eigVal(1)/eigVal(3)); % ratio of eigen values

rcVal   = 0:.05:1;      % table sample points
tabVal  = interp2(rcVal,rcVal,SEuse,r1,r2); % bi-linear interpolation

SEcal   = sqrt(eigVal(3))*tabVal;    % calcs SE radius value
end                    % ends function
```

Algorithm (5.4.3.3-2):

```
function SEcal = SECovSqrt(covar,mCoord,prob)
%%%%% Calculates spherical error radius using covariance square root
%%%%% magnitude ordering
%%%%%
%%%%% Inputs: covar - 3x3 covariance matrix
%%%%%           mCoord - 3x1 mean coordinate of errors
%%%%%           prob - probability that spherical radius will be calculated to
%%%%%
%%%%% Outputs: SEcal - calculated spherical error radius

%%%%% Check entered probability
if prob < 0.1
    fprintf('Entered probability is smaller than allowed limit.\n')
    quit
end

%%%%% Check eigen value ratio
eigVal = sort(eig(covar));
if eigVal(1) <= 0
    fprintf('Entered covariance matrix is not positive definite.\n')
    quit
elseif sqrt(eigVal(1)/eigVal(3)) < 0.0001
    fprintf('Smallest Eigen value ratio is smaller than recommended limit.\n')
end

numSamp = 1e6; % number of samples to be evaluated
X = sqrtm(covar)*randn(3,numSamp); % matrix of random samples
X = X+mCoord*ones(1,numSamp); % translates random samples by mean value
mag = sort(sum(X.^2)); % sorted magnitude of samples from origin
numProb = floor(numSamp*prob); % number of samples for current prob
SEcal = sum(sqrt(mag(numProb:numProb+1)))/2; % spherical radius
end % ends function
```

Algorithm (5.4.3.3-2):

```
function [SEcal,output] = SEintegral(covar,mCoord,prob)
%%%%% Calculates spherical error radius analytically
%%%%%
%%%%% Inputs: covar - 3x3 covariance matrix
%%%%%           mCoord - 3x1 mean coordinate of errors
%%%%%           prob - probability that spherical radius will be calculated to
%%%%%
%%%%% Outputs: SEcal - calculated spherical error radius

%%%%% Convert covariance to eigen value space
[e1,covar] = svd(covar);
mCoord = transpose(e1)*mCoord;
```

## NGA.SIG.0026.03\_1.0\_ACCPRED

```

##### Get initial very crude approximation
if prob > .95                                % checks probability value
    SEapprox = 3*sqrt(mean(diag(covar)));      % approximation
elseif prob <=0.95 && prob > .5             % checks probability value
    SEapprox = 2*sqrt(mean(diag(covar)));      % approximation
else                                          % checks probability value
    SEapprox = sqrt(mean(diag(covar)));        % approximation
end                                           % ends loop
SEapprox = sqrt(sum(mCoord.^2))+SEapprox;     % updates approximation

##### Calculate error radius
initR = @(R) (SEint(covar,mCoord,R)-prob)^2;

##### Sets maximum number of iterations for search function.
##### Number of iterations could be optimized.
if prob < .95                                % checks probability
    opt = optimset('MaxIter',20);            % option for minimization function
else                                          % checks probability
    opt = optimset('MaxIter',22);            % option for minimization function
end                                           % ends loop checking probability

[SEcal(1,1),~,~,output] = fminsearch(initR,SEapprox,opt); % spherical radius

##### Prints warning statements about results based on iteration and
##### function counts.
if output.iterations == 20 && prob < .95
    fprintf('Iteration limit was reached. Result may not be valid.\n')
elseif output.iterations == 22
    fprintf('Iteration limit was reached. Result may not be valid.\n')
elseif output.funcCount > 2*output.iterations
    fprintf('Function count is greater than twice the iteration count. ')
    fprintf('Result most like not valid.\n')
end
end                                           % ends SE analytical function

function SEint = SEint(covar,mCoord,R)
##### Returns the integral of the bivariate gaussian pdf with Mean mCoord
##### and Covariance covar about the Mean bounded by the circle with Radius R
#####
##### Inputs: covar - 3x3 covariance matrix
#####           mCoord - 3x1 mean coordinate of errors
#####           R - radius of spherical error
#####
##### Outputs: SEcal - evaluated integral at current radius R

detCov = det(covar);                          % determinate of covariance matrix
invCov = inv(covar);                          % inverse of covariance matrix

##### Function handle for the gaussian pdf 3D exponent
expMult = @(x,y,z) invCov(1,1)*(x-mCoord(1)).^2+...
    2*invCov(1,2)*(x-mCoord(1)).*(y-mCoord(2))+...
    2*invCov(1,3)*(x-mCoord(1)).*(z-mCoord(3))+...
    invCov(2,2)*(y-mCoord(2)).^2+...
    2*invCov(2,3)*(y-mCoord(2)).*(z-mCoord(3))+...
    invCov(3,3)*(z-mCoord(3)).^2;

```

## NGA.SIG.0026.03\_1.0\_ACCPRED

```
%%%%% Function handle for the gaussian pdf equation to be integrated
guasEq = @(x,y,z) 1/((2*pi)^(3/2)*sqrt(detCov))*exp(-expMult(x,y,z)/2);

%%%%% Function handles for integration limits
ymin    = @(x) -sqrt(R^2-x.^2);           % function for lower y limit
ymax    = @(x)  sqrt(R^2-x.^2);           % function for upper y limit
zmin    = @(x,y) -sqrt(R^2-x.^2-y.^2);   % function for lower z limit
zmax    = @(x,y)  sqrt(R^2-x.^2-y.^2);   % function for upper z limit

%%%%% Uses MATLAB function to integrate across the three variables
SEint   = integral3(guasEq,-R,R,ymin,ymax,zmin,zmax); % preforms integration
end                                           % ends SEint function
```