MEASUREMENT SYSTEM CHARACTERIZATION IN THE PRESENCE OF MEASUREMENT ERRORS

by

Sean A. Commo
B.S.M.E, December 2007, Old Dominion University
M.E., December 2008, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MECHANICAL AND AEROSPACE ENGINEERING

OLD DOMINION UNIVERSITY
August 2012

Approved by:

____

Drew Landman (Director)

____

Peter A. Parker (Member)

____

Colin P. Britcher (Member)

____

Robert L. Ash (Member)

ABSTRACT

MEASUREMENT SYSTEM CHARACTERIZATION IN THE PRESENCE OF MEASUREMENT ERRORS

Sean A. Commo
Old Dominion University, 2012
Director: Dr. Drew Landman

In the calibration of a measurement system, data are collected in order to estimate a mathematical model between one or more factors of interest and a response. Ordinary least squares is a method employed to estimate the regression coefficients in the model. The method assumes that the factors are known without error; yet, it is implicitly known that the factors contain some uncertainty. In the literature, this uncertainty is known as measurement error. The measurement error affects both the estimates of the model coefficients and the prediction, or residual, errors. There are some methods, such as orthogonal least squares, that are employed in situations where measurement errors exist, but these methods do not directly incorporate the magnitude of the measurement errors. This research proposes a new method, known as modified least squares, that combines the principles of least squares with knowledge about the measurement errors. This knowledge is expressed in terms of the variance ratio - the ratio of response error variance to measurement error variance. The variance ratio takes on values between 0 and 1, and for calibration applications, the ratio is typically less than 0.0625. In addition to modified least squares, a new definition of residual errors based on the variance ratio is proposed. Through several simulation studies, it is observed that the new estimator can yield different estimates of the regression coefficients and improve the residual error over ordinary least squares. As a result, modified least squares is shown to be an alternative estimation method in the presence of measurement errors.

DEDICATION

To those who have believed in me and inspired greatness...

Our deepest fear is not that we are inadequate. Our deepest fear is that we are powerful beyond measure. It is our light, not our darkness that most frightens us. We were born to make manifest the glory of God that is within us. And as we let our own light shine, we unconsciously give other people permission to do the same.

- Marianne Williamson

ACKNOWLEDGEMENTS

There are many individuals that have contributed to and supported this research effort that I would like to acknowledge. Without their support and dedication, completing this work would not have been possible.

I would first like to express my gratitude for Dr. Drew Landman, my graduate school advisor. He began working with me six years ago while I was an undergraduate student, and he greatly impacted my decision to pursue graduate studies. Dr. Landman has not only been an exceptional advisor but an invaluable colleague and friend. I look forward to continued successful collaborations in the future with him.

Secondly, I would like to recognize and thank Dr. Peter Parker from NASA. Dr. Parker provided me with my first research opportunity as a masters student calibrating instruments for the Mars Science Laboratory. He was instrumental in my current appointment with NASA for which I am incredibly grateful. He continues to not just be a mentor for me professionally but a benchmark for the kind of individual that I can only hope to be personally and professionally.

I would like to thank Dr. Colin Britcher and Dr. Robert Ash for being willing and able to serve on my dissertation committee. Dr. Britcher and Dr. Ash have supported and provided beneficial feedback to this work. Additionally, I consider myself fortunate to have taken courses during my studies with both as professors. The knowledge I've gained from their courses is something I utilize in my professional career daily.

I would like to acknowledge the current and past management and support staff of the Aeronautics Systems Engineering Branch at NASA Langley Research Center. In particular, I would like to express my sincere appreciation to Mr. Ray Rhew and Mr. Mark Hutchinson for giving me the time and support to focus on my research. Both individuals have continued to make my research a priority in my service to the agency. Furthermore, I am grateful to Mr. Rhew, principal investigator, and Mr. K. Chris Lynn, project manager, of the National Force Measurement Technology Capability (NFMTC) for their continuous support and funding throughout the entirety of this research.

Finally, I would like to thank my wife, Kathy, my parents, Dennis and Laura, and my sister, Jordyn, for their support and love through this difficult endeavor. They have been a source of strength, inspiration, and motivation throughout the years.

NOMENCLATURE

a = Distance Minimized by Ordinary Least Squares

d = Distance Minimized by Modified and Orthogonal Least Squares

E = Expected Value Operator

EFT-1 = Exploration Flight Test-1

F.S. = Full-Scale

k = Number of Factors

ME = Measurement Error

MLS = Modified Least Squares

MSE = Mean Squared Error

n = Number of Design Points

N = Normally Distributed

NIST = National Institute of Standards and Technology

OLS = Ordinary Least Squares

OrthLS = Orthogonal Least Squares

p = Number of Model Terms without the Intercept

S_{xx} = Sum of Squares in x

S_{xy} = Sum of Cross-Products in x and y

S_{yy} = Sum of Squares in y

u = Random Error or Measurement Error in x

Var = Variance Operator

W = Factor with Error

x = Factor without Error

$\bar{x}$ = Sample Mean of x

$\mathbf{x}$ = Model Matrix

y = Observed Response without Error

Y = Observed Response with Error

$\bar{Y}$ = Sample Mean of Y

$\hat{Y}$ = Predicted Response of Y

α = Angle Formed by d and Estimated Model

β = Regression Coefficient

$\hat{\beta}$ = Estimated Regression Coefficient

ϵ = Random Error in y

γ = Variance Ratio

ϕ = Angle Formed by a and Estimated Model

$\hat{\sigma}$ = Estimated Measurement System Accuracy

σ_u^2 = Variance of u

σ_ϵ = Standard Deviation of ϵ

σ_ϵ^2 = Variance of ϵ

Σ = Variance-Covariance Matrix of the Measurement Errors

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Regression analysis is a collection of statistical methods and tools used to estimate mathematical relationships between one or more explanatory variables, or factors, and responses. Regression can be classified into several categories including simple or multiple, linear or nonlinear, and parametric or nonparametric. The earliest published work on regression analysis was done by Legendre and Gauss in the early 19th century although Gauss began formulating his ideas in the late 18th century (Draper and Smith, 1998). Both Legendre and Gauss independently derived what is now known as the method of least squares and applied their methods to orbital mechanics. Additionally, Gauss developed the Gauss-Markov theorem, which revealed an elegant property of the least squares method. The method of least squares minimizes the sum of the squares of the errors. The least squares estimator is the best linear unbiased estimator (BLUE). In other words, the regression coefficients estimated from least squares have minimal variance and are unbiased. While often overlooked, this result is useful when applying the least squares method.

As Legendre and Gauss first recognized, the rationale behind estimating these mathematical relationships ranges from validating physical laws and phenomenon to understanding system behavior and performance. The relationship representing a physical law or system response in terms of k factors is expressed as

$$y = f\left(x_1, \ldots, x_k\right).$$

In k-dimensional space, the function $f(x_1, \ldots, x_k)$ is known as a response surface. The functions that make up the true response surface may be simple or complex. Because the true functions are seldom known, a simpler function over a small region of interest is used to approximate $f(x_1, \ldots, x_k)$. The approximate relationship can be expressed as

$$y = g(x_1, \ldots, x_k) + \epsilon \tag{1}$$

where ϵ is the error. For this research, the form of $g(x_1, \ldots, x_k)$ is limited to a class of linear models based on a Taylor-series expansion in k factors. These functions are often sufficient in approximating the true function since they are extremely flexible. Furthermore, estimating the coefficients in the function from historical or experimental data is accomplished using the method of ordinary least squares (OLS). In the latter case, experiments are designed to collect sufficient information in order to estimate the assumed form of $g(x_1, \ldots, x_k)$.

Second-order Taylor-series response surface functions often work well in real-world applications and therefore are one of the most commonly estimated functions (Myers et al., 2009). The mathematical form of this model is

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \epsilon \tag{2}$$

where k is the number of factors, the β's are the regression coefficients, and ϵ is the error. Based on Equation (2), there are

$$1 + 2k + \frac{k(k-1)}{2} = 1 + p$$

regression coefficients in the model. As a requirement, the experimental design must contain $1 + p$ unique design points. If $\mathbf{x}$ is the model matrix which contains the design points, then the experimental design is said to have a sufficient number of unique design points when $\left(\mathbf{x}^T\mathbf{x}\right)$ is full rank. Additionally, because the function is second-order, the experimental design must also contain 3 unique levels of each of the k factors.

The use of OLS to estimate the regression coefficients carries the following assumptions about the factors, responses, and mathematical model:

- Appropriate model specification. The form of the mathematical model is linear with respect to the regression coefficients, and the model lack-of-fit is not significant.

- Linear independence. The factors in the model matrix, $\mathbf{x}$, are linearly independent or the responses are a linear combination of the factors and regression coefficients. From the experimental design perspective, linear independence is obtained through a sufficient number of unique design points to fit the assumed mathematical model. Linear independence is expressed mathematically when $\mathbf{x}$ is of full column rank or $\left(\mathbf{x}^T\mathbf{x}\right)$ is well-conditioned.

- Independent errors. The errors, ϵ, across all the responses are independent. Mathematically, this is expressed as

$$\operatorname{Cov}\left(\epsilon_i, \epsilon_j\right) = 0 \quad \text{for } i \neq j.$$

  For errors that are not independent, generalized least squares (GLS) should be used.

- Homoscedasticity or constant variance. The variance of the errors, $\operatorname{Var}\left(\epsilon\right)$, across all the responses are equal. This is independent of the settings of the factors.

- Weak exogeneity. The factors are fixed variables and therefore assumed to be known without error. From the regression model, the errors across all the responses have a conditional mean of zero and are independent of the factors.

Other assumptions about normality and identically independent errors are not necessary for OLS, but these assumptions do support additional properties of the estimators.

The error, ϵ, in the response surface function is attributed to one or more of the following:

- The response, y, is observed with error, which may result from systematic biases or random fluctuations. This is the most commonly assumed cause of error in the estimated model.
- The form of the response surface function, $g(x_1, \ldots, x_k)$, is incorrect due to:
  - factors that were excluded but affect the response, or
  - a more complex true function, $f(x_1, \ldots, x_k)$, than the assumed function.
- The factors are not known without error. Typically, any error in the factors is neglected so the OLS can be used to estimate the mathematical model. In the literature, errors in the factors are referred to as errors-in-variables or measurement error (ME). This is the motivation behind the research presented in this dissertation.

The source of MEs can be attributed to one or more causes, including instrument and sampling errors. In practice, all physical experiments contain some degree of ME, but MEs are usually considered negligible in favor of using OLS to estimate a model. Depending on the application, the MEs may be small relative to the other sources of error in the mathematical models and therefore are assumed not to impact the modeling. However, there are a few recommendations for how small the MEs should be in order to neglect them in the regression analysis. Furthermore, statistical software packages that recognize MEs in the modeling are extremely limited, typically limited to the simple linear model or

$$y = \beta_0 + \beta_1 x + \epsilon.$$

At the National Aeronautics and Space Administration (NASA), MEs occur in many complex systems and applications, and the simple linear model is not adequate. For example, consider a wind-tunnel experiment where the factors of interest are angle-of-attack, yaw angle, and Mach number and the responses are aerodynamic forces and moments, and the objective of the experiment is to estimate the mathematical relationships between the factors and responses. It is known implicitly that both the factors and the responses contain some uncertainty; yet, there is no formal methodology employed to estimate the response surface function between the factors and responses which considers both sources of error. This issue is further exemplified when the same test article is tested in different facilities and differences in the modeled relationships cannot be reconciled. Since the ultimate goal is an understanding of the true relationship between the factors and responses, any unaccounted for errors can complicate the ability to achieve this objective.

The research presented in this dissertation proposes a new general methodology for mathematical modeling in the presence of MEs. This methodology is applicable to simple and complex systems and enables direct incorporation of uncertainties in all variables, which is often not considered in practice.

1.2 APPLICATIONS

Both within and outside of NASA, there exist several examples of ME. At NASA, complex measurement systems are designed to collect data for various projects across all the Mission Directorates, including Aeronautics, Exploration, and Atmospheric

Science. In complex, custom-designed measurement systems, system-level calibrations are critical in understanding and capturing the true performance. However, there exists a significant difference between classical instrument calibration and the characterization of a complex measurement system. In laboratory calibration, the factors are considered to be known without error and are traceable to standards from the National Institute of Standards and Technology (NIST) within the United States. In general, many of NASA's measurement systems cannot be calibrated with a system-level standard that is traceable to NIST.

In aeronautics, scaled testing of flight vehicles is performed in wind tunnels to understand aerodynamic performance. The primary measurement system used during these wind-tunnel experiments is a force-balance. A force-balance is a multiple-axis load cell that provides simultaneous, high-precision measurements of aerodynamic forces and moments in up to 6 degrees of freedom. The number of aerodynamic components that can be measured by a force-balance is a function of its mechanical and electrical design. Through a calibration experiment, a mathematical model can be developed between the applied forces and moments, or factors, and the electrical outputs of the strain gauges, or responses. The literature provides several examples of improvements in calibration techniques, including calibration in the presence of additional factors, such as temperature (Lynn et al., 2012; Parker et al., 2001; Parker and DeLoach, 2001). However, current force-balance calibration methods rely on the assumption that the applied forces and moments are known without error. For example, in force-balance calibrations that use gravity-based loads, or dead-weights, the individual weights are calibrated against NIST standards; therefore, each weight is known within a small uncertainty. These uncertainties propagate into errors in the applied forces and moments.

In exploration, research in the development of newer, more robust entry, descent, and landing (EDL) technologies is a major focus. Both landed payload mass and landing accuracy are driven by the EDL architecture for a given mission. The current unmanned exploration mission to Mars, the Mars Science Laboratory (MSL), is nearing the limit of the capabilities of available EDL technologies. While unmanned missions have less stringent requirements for landing, manned missions are infeasible with the current EDL systems (Braun and Manning, 2006). The MSL Entry, Descent, and Landing Instrumentation (MEDLI) was proposed to address some of the challenges associated with the development of newer, more robust EDL technologies (Gazarik et al., 2008). MEDLI is a suite of sensors installed on the forebody heatshield of the MSL entry vehicle. One of the subsystems of MEDLI is the pressure measurement system, which consists of two components: the Mars Entry Atmospheric Data System (MEADS) and the associated electronics. MEADS is a series of through-holes, or pressure ports, in the heatshield that connect via stainless steel tubing to pressure transducers. Power is provided to MEADS by the electronics system. Commo and Parker (2012) discuss a system-level calibration approach that was developed and employed to characterize the performance of the pressure measurement system. Typically for spaceflight instrumentation, component-by-component calibrations are performed with some of the components being calibrated with NIST-traceable standards. However, a system-level, NIST-traceable standard for the pressure measurement system does not exist.

In atmospheric sciences, Earth's upper atmosphere is receiving increased research emphasis. Researchers use satellite observations to gain insight into trends and causes of changes in the Earth's climate. Missions such as the Clouds and Earth's Radiant Energy System (CERES) have been proposed to answer questions about how radiative energy in the atmosphere affects climate change. This includes how increases in quantities of carbon dioxide affect the balance of energy. Satellite missions employ a series of simple instruments and complex measurement systems that are integrated to collect the appropriate data in order to support the mission objectives. These instruments and measurement systems are calibrated on the ground. However, satellites operate in a vacuum environment and are exposed to radiation; both of these environmental factors are known to influence the measurements. In-situ adjustments are made to the instruments and measurement systems in flight to correct for environmental effects. Measurement standards are unavailable for on-orbit satellites, so any in-situ adjustments are relative to a source, which contains some error. Similar to the force-balance example, the uncertainty from the source propagates into measurements made by the sensors on the satellites.

1.3 CLASSIFICATION OF MEASUREMENT ERRORS

Since both fixed and random variables are discussed within this research, it is important to distinguish each mathematically. For consistency, Buonaccorsi's (2010) convention is used to identify these variables. In the case of a fixed variable, a lower case letter (e.g. x) is used while a capital letter (e.g. X) indicates a random variable. Bold letters (e.g. $\mathbf{x}$ and $\mathbf{X}$) are used to indicate a vector or matrix of fixed and random variables, respectively. The *given operator* is also widely used and is designated by "$|$". For example, $|x$ can be interpreted as "given x." When referring to the random variable X, then $|x$ means "given X equal to x." Furthermore, distinguishing the difference between fixed and random factors is also important. If the factor x is taken to be fixed, then the model is identified as a functional model. The random factor X is the structural model.

To demonstrate the difference between functional and structural models, consider an experiment where a factor is applied load. Suppose a 100 lbf. load is to be applied and there are 30 possible configurations to apply the 100 lbf. load. If a configuration is randomly selected for every instance that the load is applied, then the structural model is appropriate. However, if the same configuration is selected for every instance the load is applied, then the functional model is correct. For this application, the focus is on the functional model, but many of the conclusions from the functional model can easily be extended to the structural model (Gleser, 1983). Both functional and structural models are classified as classical ME models, where W is the error-prone measurement of the factor x or X.

1.4 GENERAL FORM OF THE MEASUREMENT ERROR MODEL

The general form of the classical linear model is

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \epsilon,$$

where the βs are the regression coefficients and ϵ is the normally distributed random error of y with a mean of zero and a constant variance of σ_ϵ^2. The subscript p is the number of model terms without the intercept and does not necessarily equal the number of factors in an experiment. For a second-order response surface model based on a Taylor series expansion,

$$p = 2k + \frac{k\,(k-1)}{2}.$$

For example, if $k = 2$, then $p = 5$ and the model is

$$\begin{aligned} y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \epsilon \\ &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \epsilon, \end{aligned}$$

where $\beta_3 = \beta_{12}$, $\beta_4 = \beta_{11}$, $\beta_5 = \beta_{22}$, $x_3 = x_1 x_2$, $x_4 = x_1^2$, and $x_5 = x_2^2$. Using summation notation, the second-order Taylor series model is

$$y = \beta_0 + \sum_{a=1}^{k} \beta_a x_a + \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \beta_{ab} x_a x_b + \sum_{a=1}^{k} \beta_{aa} x_a^2 + \epsilon,$$

where $\sum_{a=1}^{k} \beta_a x_a$ are the linear terms, $\sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \beta_{ab} x_a x_b$ are the two-factor interaction terms, and $\sum_{a=1}^{k} \beta_{aa} x_a^2$ are the second-order terms.

From an experiment, n observations are used to estimate the regression coefficients in the model. At the ith observation, the classical model is

$$y_i = \beta_0 + \beta_1 x_{1_i} + \ldots + \beta_p x_{p_i} + \epsilon_i \qquad\qquad i = 1, 2, \ldots, n.$$

For each observation, the expected mean and variance operators are applied to get

$$E\left(y_i | x_{1_i}, \ldots, x_{p_i}\right) = \beta_0 + \beta_1 x_{1_i} + \ldots + \beta_p x_{p_i} \quad \text{or} \quad E\left(\epsilon_i\right) = 0, \text{ and}$$

$$\text{Var}\left(y_i | x_{1_i}, \ldots, x_{p_i}\right) = \text{Var}\left(\epsilon_i\right) = \sigma_\epsilon^2.$$

As expressed, the variance is assumed to be constant across all observations. Furthermore, since the observations are assumed to be independently and identically distributed, the covariance between two responses, or Cov $\left(y_i, y_j | x_{1_i}, \ldots, x_{p_i}\right)$ where $i \neq j$, is zero. The n equations are expressed as

$$\begin{aligned}
y_1 &= \beta_0 + \beta_1 x_{1_1} + \ldots + \beta_p x_{p_1} + \epsilon_1 \\
y_2 &= \beta_0 + \beta_1 x_{1_2} + \ldots + \beta_p x_{p_2} + \epsilon_2 \\
&\vdots \\
y_n &= \beta_0 + \beta_1 x_{1_n} + \ldots + \beta_p x_{p_n} + \epsilon_n.
\end{aligned}$$

Combining the system of equations in matrix form yields

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{1_1} & \cdots & x_{p_1} \\ 1 & x_{1_2} & \cdots & x_{p_2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1_n} & \cdots & x_{p_n} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

or

$$\mathbf{y} = \mathbf{x}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{y}$ is a $n \times 1$ vector, $\mathbf{x}$ is a $n \times (p+1)$ matrix, $\boldsymbol{\beta}$ is a $(p+1) \times 1$ vector, and $\boldsymbol{\epsilon}$ is a $n \times 1$ vector. The expected mean and covariance, in matrix notation, are

$$E(\mathbf{y}) = E(\mathbf{x}\boldsymbol{\beta}) + E(\boldsymbol{\epsilon}) = \mathbf{x}\boldsymbol{\beta} \quad \text{or} \quad E(\boldsymbol{\epsilon}) = \mathbf{0}, \text{ and}$$

$$\text{Var}(\mathbf{y}) = \text{Var}(\mathbf{x}\boldsymbol{\beta}) + \text{Var}(\boldsymbol{\epsilon}) = \sigma_\epsilon^2 \mathbf{I}.$$

However, the factors are not known without error. Instead, the error-prone value of the factor x is

$$W_i = x_i + u_i$$

where u_i is the random ME associated with x_i and is distributed with a mean of zero and a variance of $\sigma_{u_i}^2$. This form of ME is known as the additive model since $E(W_i|x_i) = x_i$. Therefore, W_i is unbiased for the true value of x_i. For k factors, the errors in the xs are

$$(u_1, \ldots, u_k) \sim \left[\mathbf{0}, \left(\sigma_{u_1}^2, \ldots, \sigma_{u_k}^2\right)\right].$$

Constant variance of the errors in the xs is not necessary, but for most practical applications, this assumption of constant variance is sufficient. The xs in the classical model are replaced with the ME model to get

$$\begin{aligned} y &= \beta_0 + \beta_1 (x_1 + u_1) + \ldots + \beta_p (x_p + u_p) + \epsilon \\ &= \beta_0 + \beta_1 W_1 + \ldots + \beta_p W_p + \epsilon. \end{aligned}$$

Unlike the classical model, the statistical properties of the ME model cannot be generalized. Starting with the first-order model ($p = k$)

$$y = \beta_0 + \sum_{a=1}^{k} \beta_a W_a + \epsilon,$$

the expected value and variance are (Buonaccorsi, 2010)

$$E(y|W_1, \ldots, W_k) = \beta_0 + \sum_{a=1}^{k} \beta_a x_a, \text{ and}$$

$$\mathrm{Var}(y|W_1, \ldots, W_k) = \sum_{a=1}^{k} \beta_a^2 \sigma_{u_a}^2 + \sigma_\epsilon^2.$$

If the MEs are correlated, then the expected value and variance include the covariances between $u_1, \ldots, u_k$. However, independence is assumed between the MEs. Regardless of independence, the variance of the response is inflated due to the ME. For a first-order with interaction model ($p = k + [k(k-1)]/2$),

$$y = \beta_0 + \sum_{a=1}^{k} \beta_a W_a + \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \beta_{ab} W_a W_b + \epsilon,$$

the expected value and variance are (Buonaccorsi, 2010)

$$E(y|W_1, \ldots, W_k) = \beta_0 + \sum_{a=1}^{k} \beta_a x_a + \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \beta_{ab} x_a x_b, \text{ and}$$

$$\mathrm{Var}\left(y|W_1,\ldots,W_k\right)=\sum_{a=1}^{k}\beta_a^2\sigma_{u_a}^2+\sum_{a=1}^{k-1}\sum_{b=a+1}^{k}\left[\beta_{ab}^2\left(x_a^2\sigma_{u_b}^2+x_b^2\sigma_{u_a}^2+\sigma_{u_a}^2\sigma_{u_b}^2\right)\right]+\sigma_\epsilon^2.$$

Similar to the first-order model, the first-order with interaction model is additive since response is unbiased for independent MEs. The variance is slightly larger for the first-order with interaction model since it contains additional model terms. Lastly, for the second-order model $(p=2k+\left[k\left(k-1\right)\right]/2)$

$$y=\beta_0+\sum_{a=1}^{k}\beta_aW_a+\sum_{a=1}^{k-1}\sum_{b=a+1}^{k}\beta_{ab}W_aW_b+\sum_{a=1}^{k}\beta_{aa}W_a^2+\epsilon,$$

the expected value and variance are (Buonaccorsi, 2010)

$$E\left(y|W_1,\ldots,W_k\right)=\beta_0+\sum_{a=1}^{k}\beta_ax_a+\sum_{a=1}^{k-1}\sum_{b=a+1}^{k}\beta_{ab}x_ax_b+\sum_{a=1}^{k}\beta_{aa}\left(x_a^2+\sigma_{u_a}^2\right),\text{ and}$$

$$\begin{aligned}\mathrm{Var}\left(y|W_1,\ldots,W_k\right) &= \sum_{a=1}^{k}\beta_a^2\sigma_{u_a}^2+\sum_{a=1}^{k-1}\sum_{b=a+1}^{k}\left[\beta_{ab}^2\left(x_a^2\sigma_{u_b}^2+x_b^2\sigma_{u_a}^2+\sigma_{u_a}^2\sigma_{u_b}^2\right)\right]\\ &\quad+\sum_{a=1}^{k}\left[4\beta_{aa}^2x_a^2\sigma_{u_a}^2+\sigma_{u_a}^2\right]+\sigma_\epsilon^2.\end{aligned}$$

Although the ME is additive for any single factor, the second-order model is nonadditive since $E\left(y|W_1,\ldots,W_k\right)$ contains some bias. More specifically, the bias in the second-order model is $\sum_{a=1}^{k}\beta_{aa}\sigma_{u_a}^2$. The variance is further inflated by the second-order terms in the model. Therefore, the effect of ME on the statistical properties of the model increases as the order of the model increases. A summary of the bias and variance results are presented in Table 1.

TABLE 1: Summary of Bias and Variance for Classical and Measurement Error Models

		Classical	Measurement Error
First-Order Model	$E(y)$	$\beta_0 + \sum_{a=1}^{k} \beta_a x_a$	$\beta_0 + \sum_{a=1}^{k} \beta_a x_a$
	$\text{Var}(y)$	σ_ϵ^2	$\sigma_\epsilon^2 + \sum_{a=1}^{k} \beta_a^2 \sigma_{u_a}^2$
First-Order w/ Interaction Model	$E(y)$	$\beta_0 + \sum_{a=1}^{k} \beta_a x_a + \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \beta_{ab} x_a x_b$	$\beta_0 + \sum_{a=1}^{k} \beta_a x_a + \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \beta_{ab} x_a x_b$
	$\text{Var}(y)$	σ_ϵ^2	$\sigma_\epsilon^2 + \sum_{a=1}^{k} \beta_a^2 \sigma_{u_a}^2$ $+ \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \left[\beta_{ab}^2 \left(x_a^2 \sigma_{u_b}^2 + x_b^2 \sigma_{u_a}^2 + \sigma_{u_a}^2 \sigma_{u_b}^2\right)\right]$
Second-Order Model	$E(y)$	$\beta_0 + \sum_{a=1}^{k} \beta_a x_a + \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \beta_{ab} x_a x_b$ $+ \sum_{a=1}^{k} \beta_{aa} x_a^2$	$\beta_0 + \sum_{a=1}^{k} \beta_a x_a + \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \beta_{ab} x_a x_b$ $+ \sum_{a=1}^{k} \beta_{aa} \left(x_a^2 + \sigma_{u_a}^2\right)$
	$\text{Var}(y)$	σ_ϵ^2	$\sigma_\epsilon^2 + \sum_{a=1}^{k} \beta_a^2 \sigma_{u_a}^2$ $+ \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \left[\beta_{ab}^2 \left(x_a^2 \sigma_{u_b}^2 + x_b^2 \sigma_{u_a}^2 + \sigma_{u_a}^2 \sigma_{u_b}^2\right)\right]$ $+ \sum_{a=1}^{k} \left[4\beta_{aa}^2 x_a^2 \sigma_{u_a}^2 + \sigma_{u_a}^2\right]$

1.4.1 EXAMPLE: FORCE TRANSDUCER CALIBRATION

Suppose the output of a force transducer, y, is a function of an applied force, F, and temperature, T. The applied forces and temperatures have been calibrated against known standards with the following uncertainties:

$$\begin{aligned} u_F &\sim N\left(0 \text{ lbs.}, 0.01 \text{ lbs.}^2\right), \text{ and} \\ u_T &\sim N\left(0 \text{ K}, 0.01 \text{ K}^2\right). \end{aligned}$$

In physical terms, temperature is known with a precision of $\sqrt{0.01 \text{ K}^2} = 0.1$ degrees Kelvin. If the true model for the force transducer in the presence of ME is

$$y = \beta_0 + \beta_1 (F + u_F) + \beta_2 (T + u_T) + \beta_{12} (F + u_F)(T + u_T) + \epsilon$$

where $\beta_0 = 0$, $\beta_1 = \beta_2 = \beta_{12} = 1$, and $\epsilon \sim N(0, 0.01)$, then the expected mean is

$$\begin{aligned} E(y|F, T, u_F, u_T) &= \beta_0 + \beta_1 F + \beta_2 T + \beta_{12} (F \times T) \\ &= F + T + (F \times T). \end{aligned}$$

Additionally, the variance is

$$\begin{aligned} \text{Var}(y|F, T, u_F, u_T) &= \sigma_\epsilon^2 + \beta_1^2 \sigma_{u_F}^2 + \beta_2^2 \sigma_{u_T}^2 + \beta_{12}^2 \left(F^2 \sigma_{u_T}^2 + T^2 \sigma_{u_F}^2 + \sigma_{u_F}^2 \sigma_{u_T}^2\right) \\ &= 0.01 + 0.01 + 0.01 + \left(0.01T^2 + 0.01F^2 + 0.0001\right) \\ &= 0.0301 + 0.01T^2 + 0.01F^2. \end{aligned}$$

In the classical linear model, the variance of y is 0.01, which comes from ϵ. However, since variance is never less than zero, the variance in the classical model will always be an underestimate of the true, unknown variance.

CHAPTER 2

LITERATURE REVIEW

Adcock (1877, 1878) is credited with developing the first least squares-based method that is applicable to measurement error (ME) models. He proposed a multi-dimensional objective function as an alternative to the one-dimensional objective function that is used in ordinary least squares (OLS) estimation. Adcock considered the normal distance from the estimated line to an observation. Since he utilized least squares principles to estimate the regression coefficients, Adcock minimized the sum of the squared normal distances over n observations. This approach is the basis of orthogonal least squares (OrthLS) regression. It is noted that the line estimated by OrthLS always passes through the centroid of the data. However, OrthLS is limited since it weights equally all the variables in the model. In terms of ME models, it is equivalent to assuming the variability in the factors is equal to the variability in the response. Meanwhile, Pearson (1901) showed that the line estimated by OrthLS is bound by the OLS regressions of Y on x and x on Y in two-dimensional space.

Kummel (1879) expanded upon the work of Adcock by developing a more general solution that is valid when the errors are not assumed to be equal. He is the first person to define the concept of variance ratio, γ; mathematically, the variance ratio is expressed as

$$\gamma = \frac{\sigma_\epsilon^2}{\sigma_u^2}.$$

where σ_ϵ^2 is the variance of the error in the response and σ_u^2 is the variance of the ME. The variance ratio is an important parameter later in this research as it is used to derive a new estimator. Based on prior experimental data and subject-matter expertise, Kummel argues that obtaining an estimate of γ is reasonable in practice. In most calibration equations, γ takes on values between 0 and 0.0625. The interpretation of these values is discussed in the next chapter.

Deming (1931, 1964) continued developing the work of Adcock and Kummel and applied it to the special case of the simple linear model. The method he derived is appropriately named Deming regression. The error-prone observations in a set of experimental data are

$$Y_i = y_i + \epsilon_i$$

and

$$W_i = x_i + u_i$$

where $i = 1, 2, \ldots, n$. The errors ϵ and u are independent random variables and are assumed to be distributed with a mean of zero and a constant variance of σ_ϵ^2 and σ_u^2, respectively, and are related through the variance ratio, γ. If the variances are not constant, the method is invalid. The model coefficients are estimated in Deming's method by minimizing

$$\sum_{i=1}^{n} \left(\frac{\epsilon_i^2}{\sigma_\epsilon^2} + \frac{u_i^2}{\sigma_u^2} \right) = \sum_{i=1}^{n} \left[(Y_i - \beta_0 - \beta_1 x_i)^2 + \gamma \left(W_i - x_i\right)^2 \right]$$

for the model $y = \beta_0 + \beta_1 x$. This expression assumes that the variance ratio is a known value. Any errors in estimating the variance ratio are neglected. Using standard calculus methods, the estimates of the model parameters are

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{W},$$

$$\hat{\beta}_1 = \frac{S_{YY} - \gamma S_{WW} + \sqrt{\left(S_{YY} - \gamma S_{WW}\right)^2 + 4\gamma S_{WY}^2}}{2 S_{WY}}, \text{ and}$$

$$\hat{x}_i = W_i + \frac{\hat{\beta}_1}{\hat{\beta}_1^2 + \gamma} \left(Y_i - \hat{\beta}_0 - \hat{\beta}_1 W_i \right)$$

where

$$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

$$\bar{W} = \frac{1}{n}\sum_{i=1}^{n} W_i$$

$$S_{YY} = \sum_{i=1}^{n}\left(Y_i - \bar{Y}\right)^2,$$

$$S_{WW} = \sum_{i=1}^{n}\left(W_i - \bar{W}\right)^2,$$

and

$$S_{WY} = \sum_{i=1}^{n}\left(W_i - \bar{W}\right)\left(Y_i - \bar{Y}\right).$$

When $\gamma = 1$, the estimates of the model coefficients are equivalent to the OrthLS estimates. The only difference is that when W is unknown, x is substituted into these expressions. It is important to note that for normally distributed errors, the Deming estimates are also maximum likelihood estimates (Casella and Berger, 2002).

Deming regression is considered a special case of total least squares (TLS) (Golub and Loan, 1980), which is used for multivariate ME models. Unlike Deming regression, TLS does not generally have a closed-form solution. The method is a numerical solution to the linear algebraic equation

$$\left(\mathbf{x} + \mathbf{u}\right)\boldsymbol{\beta} = \mathbf{y} + \boldsymbol{\epsilon}$$

where the objective is to minimize $\mathbf{u}$ and $\boldsymbol{\epsilon}$ given $\mathbf{x}$ and $\mathbf{y}$. Singular-value decomposition is one approach to solving for the model coefficients in the equation. However, implementing TLS in practical application, such as calibration, is difficult.

Carroll and Spiegelman (1986) discuss the effect of small MEs on the estimates the regression coefficients and confidence intervals in an instrument characterization experiment. In their discussion, Carroll and Spiegelman only consider the simple linear model. For most calibration applications at NASA LaRC, the MEs are at least four times smaller than the accuracy of the measurement system being characterized. Carroll and Spiegelman acknowledged two criterion used to determine whether ME, or uncertainties in the standard, can be neglected in the analysis. These criterion are:

- the ratio of ME to the true variability in the x's (Draper and Smith, 1998), and
- the ratio of the measured response to the slope, or primary sensitivity (Scheffe, 1973; Mandel, 1984).

They stated that the first criterion is most appropriate for estimating the regression coefficients while the second criterion affects the width of the confidence intervals, but they argue both criterion should be used in combination when analyzing the data.

The ratio, ν, based on Draper and Smith's (1998) criterion is

$$\nu = \frac{\text{ME Variance}}{\text{Variance of } x} = \frac{\sigma_u^2}{S_{xx}}.$$

In the limit as the number of observations, n, approaches infinity, the estimated intercept is approximately

$$\hat{\beta}_0 \sim \beta_0 + \left(\frac{\nu}{\nu+1}\right)\bar{x}\beta_1,$$

and the estimated slope is approximately

$$\hat{\beta}_1 \sim \left(\frac{\nu}{\nu+1}\right)\beta_1.$$

Therefore, for small ν, the estimated coefficients are only affected slightly by ME.

Centering the x's also negates any effect of ME on the estimate of the intercept. The width of the confidence interval without ME is approximately $2t_{\alpha/2,n-p}\sigma_\epsilon$, where $t_{\alpha/2,n-p}$ is the $(1-\alpha/2)$ percentage point of the t-distribution with $n-p$ degrees of freedom. When MEs are ignored, the confidence interval is approximately $2t_{\alpha/2,n-p}\left(\sigma_\epsilon^2+\beta_1^2\sigma_u^2\right)^{1/2}$. As a result, the ratio of confidence interval width when ignoring ME to confidence interval width without ME is

$$\frac{2t_{\alpha/2,n-p}\left(\sigma_\epsilon^2+\beta_1^2\sigma_u^2\right)^{1/2}}{2t_{\alpha/2,n-p}\sigma_\epsilon}=\left(1+\frac{\beta_1^2\sigma_u^2}{\sigma_\epsilon^2}\right)^{\frac{1}{2}}.$$

It is evident that the confidence interval width is increased by $\left(1+\frac{\beta_1^2\sigma_u^2}{\sigma_\epsilon^2}\right)^{1/2}$ when ignoring the ME. This result is also true for prediction intervals.

Carroll and Ruppert (1996) study the effect of modeling error in OrthLS. In any regression, it is desired that the estimated mathematical model adequately characterize the relationship in the experimental data. Standard regression techniques utilize a lack-of-fit (LOF) test to assess model adequacy, but this test is a function of the pure experimental error and the available number of degrees of freedom. In experiments with MEs, replicates are not genuine; instead, data points that are close enough may be considered pseudo-replicates. Therefore, without a LOF-equivalent test for ME models, it is difficult to assess model adequacy. Carroll and Ruppert go on further to suggest that incorrect knowledge of the variance ratio also attributes to LOF in the model. They argue that in practice γ is underestimated, which leads to an overcorrection in the estimated slope in OrthLS. For appropriate use of OrthLS, the variance ratio should be redefined to be

$$\gamma'=\frac{\sigma_q^2+\sigma_\epsilon^2}{\sigma_u^2}$$

where σ_q^2 is the variance of the modeling error and is assumed to be constant. In simulation, the modeling error may be neglected since the appropriate model form is often known. For practical applications, Carroll and Ruppert recommended that OrthLS be used with caution, especially when the appropriate model form is unknown.

CHAPTER 3

SIMPLE LINEAR MODELS

3.1 APPLICATION OF THE SIMPLE LINEAR MODEL

In the literature review, a body of research associated with measurement errors (ME) for the simple linear model was discussed. For some applications, the simple linear model works well. In general, the simplest model that adequately explains the experimental data is desired. It is important to recognize that any estimated model is only an approximation of the true, underlying function. There are many applications that require multi-dimensional, higher-order response surface models, and the theory for the simple linear problem provides the necessary framework on which to expand in order to model more complex situations. Therefore, the discussion begins with the simple linear model in order to formulate a general methodology.

In this research, some of the most common techniques used for MEs in the simple linear model were explored. While additional statistical theory is available, the practical implementation of this theory is limited. As a result, the goal of this work was to develop an engineering solution using some principles from least squares theory. The results are applicable to practical situations including the calibration of simple instruments such as single-axis load cells and pressure transducers.

Consider an instrument where the output, y, is assumed to be linearly related to a single applied factor, x. A calibration experiment is designed to measure the response at known values of x. The resulting set of n design points are used to estimate the linear relationship between x and the observed response Y. The estimated model is then employed on future observations of x to predict Y, as long as the value of x falls within the region spanned by the calibration experiment. However, in the calibration, it is assumed that the applied values of x are known, fixed quantities. In practice, the actual value of x applied is seldom known without error. Since OLS estimation requires that the regression parameters be known without error, it is necessary to utilize practical assumptions with regard to their actual accuracies. The two most common assumptions are:

- The error in x is small relative to the error in the response. In calibration applications, the magnitude of the error in x is typically stipulated.

- The error in x is captured in the error in the response. This results in the model containing a single error term; namely $\epsilon^* = \epsilon + u$.

For ME models, Fuller (1987) demonstrated the consequences of this naive analysis, including induced bias in the estimate of the slope coefficient.

As a motivating, practical example for the simple linear case, consider the NASA Orion Multi-Purpose Crew Vehicle (MPCV), which is currently being designed to transport astronauts beyond low-Earth orbit (LEO). The first flight test of the Orion MPCV will be the Exploration Flight Test 1 (EFT-1) and will be a four-hour orbital test scheduled for late 2014. The flight test will test various spacecraft subsystems and includes several on-board measurement systems to collect flight data. One of these measurement systems is a Flush Air Data System (FADS). The FADS for EFT-1 contains nine pressure ports located in a cruciform pattern across the fore-body heatshield of the aeroshell and the data collected will be inputs to post-flight trajectory reconstruction algorithms that estimate various flight parameters, such as angle-of-attack and Mach number. Each pressure port is connected via tubing to a digital pressure transducer that is mounted on the backside of the heatshield. The uncertainty in estimating the flight parameters is a function of the calibration of each pressure transducer (Commo and Parker, 2012). The transducer output, y, is assumed to be linearly related to the sensed pressure, x. To verify this assumption, a simple calibration experiment was designed to estimate the mathematical relationship between the output and sensed pressure. The pressure levels during the calibration were applied via a reference pressure standard. Reference standards at NASA are calibrated against a NIST-traceable standard. However, reference standards contain a small amount of uncertainty although they are often used in a similar manner as a NIST standard. As a result, the applied pressures are only known to within a certain accuracy. The actual applied pressure in this case is

$$W = x + u$$

where u is the uncertainty in the reference standard as estimated from the NIST-traceable calibration.

Assuming the simple linear model is appropriate for this example, then

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \tag{3}$$

where β_0 is the zero-offset, β_1 is the sensitivity, and ϵ is the normally-distributed random error of y with a mean of zero and a constant variance of σ_ϵ^2. The standard deviation, σ_ϵ, is typically considered to be the accuracy of the instrument in the classical model, but the accuracy is redefined later for the ME model. As expressed, Equation (3) is consistent with the x's being known without error. Under this assumption, OLS is appropriate for estimating the regression coefficients. In the presence of MEs, Equation (3) becomes

$$y_i = \beta_0 + \beta_1 (x_i + u_i) + \epsilon_i \tag{4}$$

where u_i is the error in x_i and is distributed with a mean of zero and a constant variance of σ_u^2. If u was known for every instance of x, then estimation could proceed with OLS estimation using $W = x + u$ as the regressor instead of x. In physical experiments, the value of u is not known and other methods are required to estimate the regression coefficients.

From statistical regression theory, OLS minimizes the squared difference between the observed value of the response, Y, and the predicted observed value of the response from the estimated model, $\hat{Y}$. Mathematically, this is expressed as

$$\left[\hat{\beta}_0, \hat{\beta}_1\right] = \min \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2.$$

If the observed value contains no error, then the minimization is

$$\left[\hat{\beta}_0, \hat{\beta}_1\right] = \min \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2.$$

It is easily seen that any MEs are not considered in this objective function, therefore, the objective function is one-dimensional. For the simple linear model with MEs, a two-dimensional objective function is appropriate. The form of this objective function is determined by the variance ratio, γ. The variance ratio is

$$\gamma = \frac{\sigma_u^2}{\sigma_\epsilon^2}.$$

This ratio is particularly important in determining how well the x's should be known in a calibration experiment. For OLS, γ is assumed to be zero.

One particularly well-known and useful estimation technique in the presence of ME is orthogonal least squares (OrthLS) (Carroll et al., 1985). Most of the literature available on OrthLS is related to the simple linear problem. While OrthLS provides a nice alternative to OLS for ME models, it relies on the assumption of equal error variances of both variables. In other words, the variance ratio is assumed to be one.

In terms of the calibration application, the uncertainty in the calibration reference is on the order of the uncertainty in the response, which is insufficient. It is preferred that the uncertainty in the reference standard be at least four (4) times smaller than the uncertainty in the response of the measurement system. This corresponds to a variance ratio of

$$\gamma = \left(\frac{1}{4}\right)^2 = 0.0625.$$

The variance ratio for typical calibration applications at NASA are 0.0625. Because of this, neither OLS or OrthLS is most appropriate for estimation of the mathematical model. However, there is no known least squares-based technique that can be applied for variance ratios between 0 and 1.

The uncertainties in the responses of the measurement systems at NASA can vary drastically. Highly-accurate instruments, such as the pressure transducers for EFT-1, range from 0.01 to 0.1 percent of the full-scale (F.S.) range. For a 0.01 percent F.S. uncertainty, the calibration reference should have a maximum uncertainty of 0.0025 percent of F.S. Other measurement systems, like force-balances, are transducer-class instruments with slightly larger uncertainties, on the order of 0.10 to 10 percent of F.S. The worse-case is seen in atmospheric science applications where uncertainties can approach 50 percent of F.S. For this research, the focus is limited to a maximum uncertainty of 20 percent of F.S.

For the simple linear model, Figures 1-4 show the spatial dispersion of errors in x and y for several variance ratios. It is obvious that as errors in x become larger, the assumptions required for OLS estimation no longer hold and the estimate of the slope is affected. Therefore, other estimators, such as OrthLS for $\gamma = 1$, are more appropriate and more robust with regards to the MEs. In addition, the implications of the two ME assumptions discussed earlier are revealed. For small MEs relative to the errors in the response (i.e. $\gamma = 0.01$), Figure 2 shows that the estimate of the slope can still be affected. If the ME is combined with the response error to form a single error term, then the dimension of error is reduced from two to one, as in Figure 1.

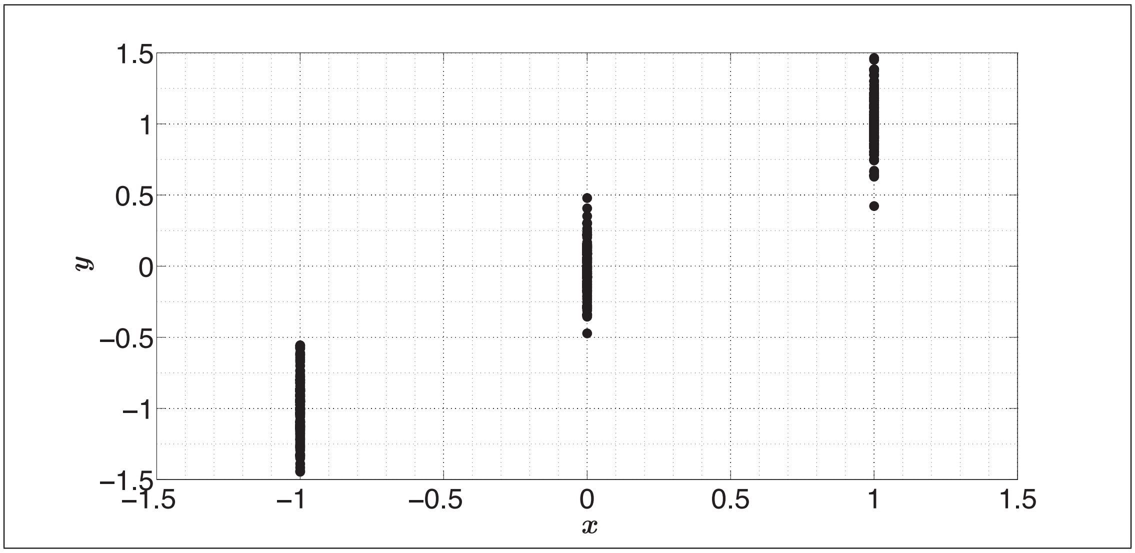

FIG. 1: Spatial Dispersion of Errors for the Simple Linear Model for $\gamma = 0$

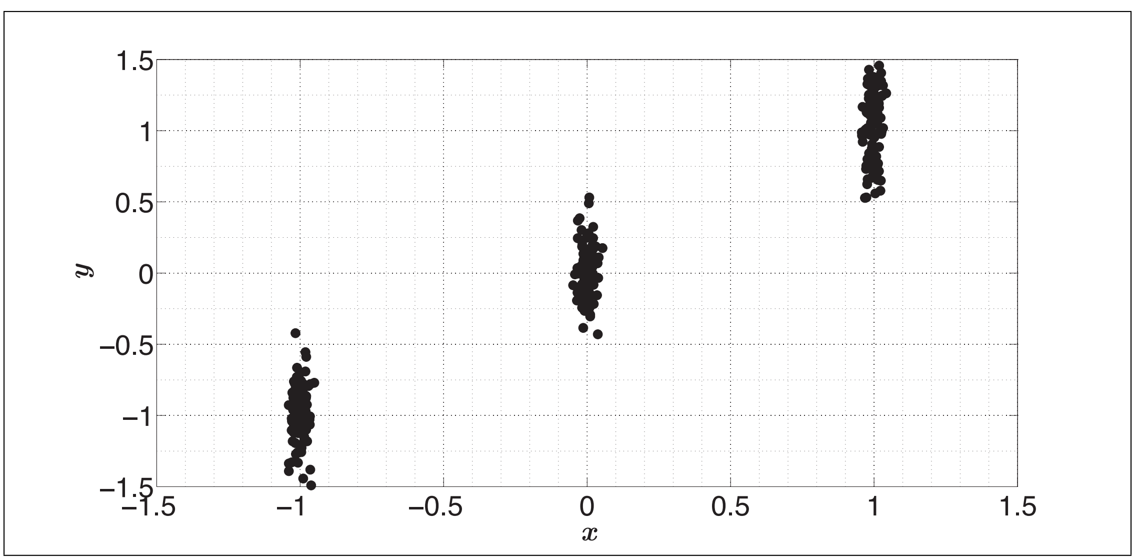

FIG. 2: Spatial Dispersion of Errors for the Simple Linear Model for $\gamma = 0.01$

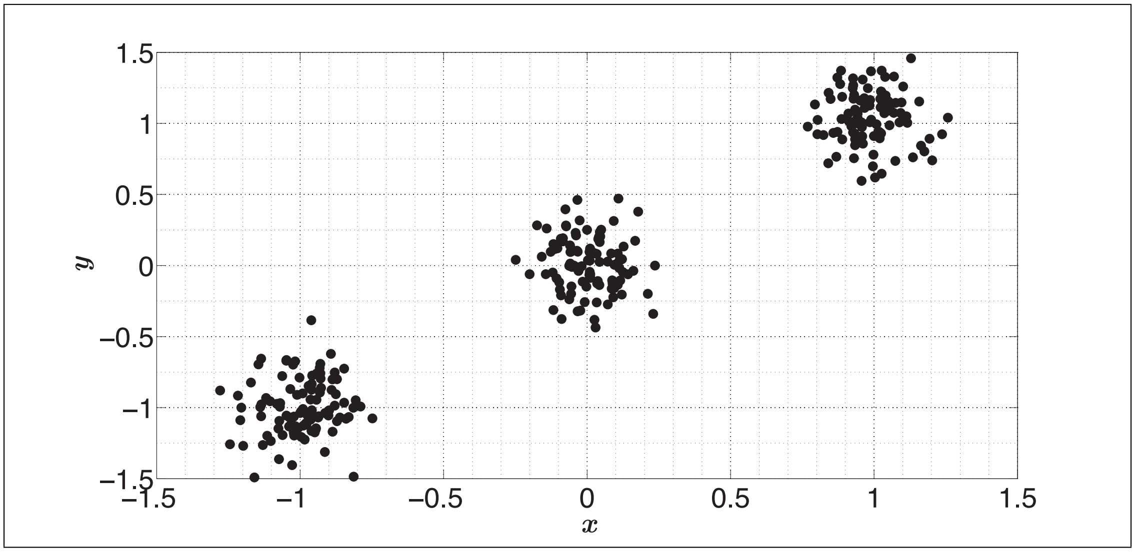

FIG. 3: Spatial Dispersion of Errors for the Simple Linear Model for $\gamma = 0.25$

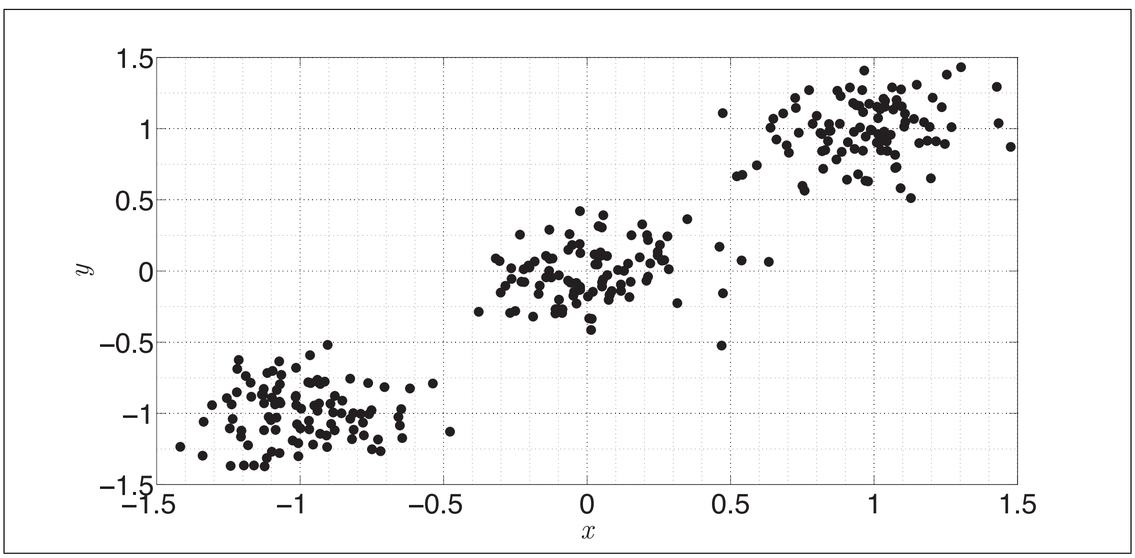

FIG. 4: Spatial Dispersion of Errors for the Simple Linear Model for $\gamma = 1$

3.2 ESTIMATION METHODS FOR THE SIMPLE LINEAR MODEL

For the simple linear measurement system characterization problem, the following three methods of estimating the model coefficients are discussed: OLS, OrthLS, and propose modified least squares (MLS) as a more general approach to address the limitation previously mentioned.

3.2.1 ORDINARY LEAST SQUARES

For a calibration experiment with n design points, the sample means are

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

and

$$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

where the responses are the error-prone values. Using the sample means, the sum of squares are

$$S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

and

$$S_{yy} = \sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2.$$

Similarly, the sum of cross-products is

$$S_{xy} = \sum_{i=1}^{n} (x_i - \bar{x})\left(Y_i - \bar{Y}\right).$$

As stated previously, the minimization criteria for OLS is

$$\left[\hat{\beta}_0, \hat{\beta}_1\right] = \min\left[\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2\right] = \min\left[\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 x_i)^2\right]. \tag{5}$$

The OLS estimators are found by differentiating the objective function in Equation

(5) with respect to β_0 and β_1 and setting each resulting equation to zero. Therefore,

$$\left[\frac{\partial}{\partial \beta_0}\right]_{\hat{\beta}_0,\hat{\beta}_1} = -2\sum_{i=1}^{n}\left(Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i\right) = 0$$

and

$$\left[\frac{\partial}{\partial \beta_1}\right]_{\hat{\beta}_0,\hat{\beta}_1} = -2\sum_{i=1}^{n}\left(Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i\right) x_i = 0.$$

These are known as the least squares normal equations. Solving for $\hat{\beta}_0$ and $\hat{\beta}_1$, the OLS estimators for the simple linear model are

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x} \tag{6}$$

and

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})\left(Y_i - \bar{Y}\right)}{\sum_{x=1}^{n}(x_i - \bar{x})^2} = \frac{S_{xy}}{S_{xx}}. \tag{7}$$

When the xs are centered, $\bar{x} = 0$ and the estimator of the zero-intercept, $\hat{\beta}_0$, is $\bar{Y}$. As mentioned earlier, the Gauss-Markov Theorem states that under the objective function given in Equation (5) and for uncorrelated, homoscedastic errors, the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are best linear unbiased estimates (BLUE). Simply put, under the criteria of minimizing the distance between Y and $\hat{Y}$, the OLS estimates will always yield the smallest sum of squared errors. This concept leads to a new definition of residual error for ME models as a function of the variance ratio.

3.2.2 ORTHOGONAL LEAST SQUARES

When the xs are known without error, the OLS criteria for minimizing the errors in the ys is a logical choice. However, in the ME models, it is more appropriate to use an estimator that considers the error in the xs in addition to the ys. The most commonly-employed method that accounts for errors in both variables for the simple linear model is OrthLS. In OrthLS, the estimators are found by minimizing the sum of the squared orthogonal distances from all the points to the estimated line. See Figure 5 for the geometrical interpretation of the two estimators. Therefore, OrthLS equally weighs the errors in both x and y when deriving the estimators. In terms of the variance ratio, $\gamma = 1$.

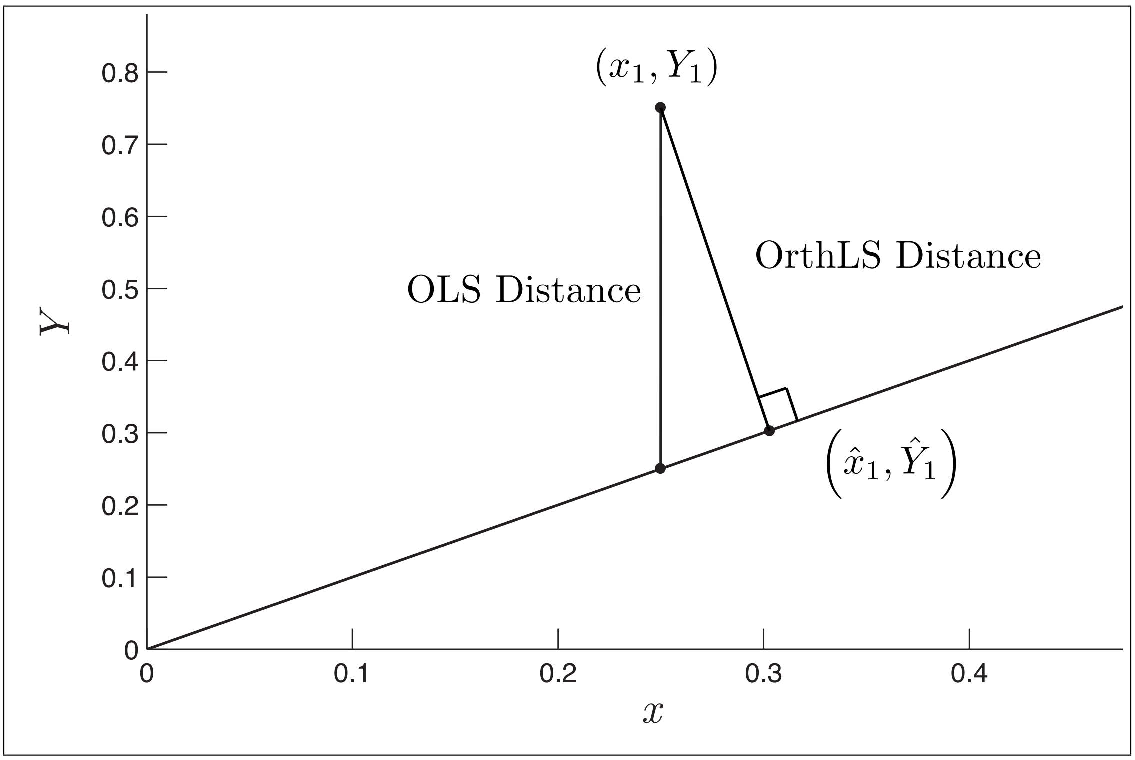

FIG. 5: Distance Minimized by Orthogonal Least Squares

Now consider the data point (x_1, Y_1) in Figure 5. Since u is unknown and an additive ME model is assumed, the best estimate of the unknown, actual level of factor W is x. Therefore, the derivation of the OrthLS estimators proceeds based on x. From algebra, it is shown that the shortest distance from a point to a line is the orthogonal distance. If the line is given by the simple linear model

$$Y = \beta_0 + \beta_1 x,$$

then the corresponding closest point on the line is (Stewart, 2003)

$$\hat{x}_1 = \frac{\beta_1 Y_1 + x_1 - \beta_0 \beta_1}{1 + \beta_1^2}, \tag{8}$$

and

$$\hat{Y}_1 = \beta_0 + \frac{\beta_1}{1 + \beta_1^2} \left(\beta_1 Y_1 + x_1 - \beta_0 \beta_1 \right). \tag{9}$$

The squared orthogonal distance between the point and the line is defined as $(x - \hat{x})^2 + \left(Y - \hat{Y} \right)^2$. Over n design points in an experiment, the objective function is

$$\left[\hat{\beta}_0, \hat{\beta}_1 \right] = \min \sum_{i=1}^{n} \left[(x_i - \hat{x}_i)^2 + \left(Y_i - \hat{Y}_i \right)^2 \right].$$

Substituting Equations (8) and (9) into the objective function

$$\left[\hat{\beta}_0, \hat{\beta}_1 \right] = \min \left[\frac{1}{1 + \beta_1^2} \sum_{i=1}^{n} \left[Y_i - (\beta_0 + \beta_1 x_i) \right]^2 \right]. \tag{10}$$

Differentiating Equation (10) with respect to β_0 and setting the resulting equation to zero yields $\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x}$, which is identical to the OLS estimate of the zero-intercept. Regardless of whether the errors in x are included or excluded, the estimate of the zero-intercept is unaffected in the simple linear model. Substituting this result back into the right-hand side of Equation (10) and setting equal to zero to find the minimum gives

$$\frac{1}{1 + \hat{\beta}_1^2} \sum_{i=1}^{n} \left[\left(Y_i - \bar{Y} \right) - \hat{\beta}_1 \left(x_i - \bar{x} \right) \right]^2 = \frac{1}{1 + \hat{\beta}_1^2} \left[S_{yy} - 2\hat{\beta}_1 S_{xy} + \hat{\beta}_1^2 S_{xx} \right] = 0$$

using the prior definitions of the sample sum of squares and sum of cross-products. The minimum and thus the OrthLS estimate of the sensitivity is

$$\hat{\beta}_1 = \frac{-\left(S_{xx} - S_{yy}\right) + \sqrt{\left(S_{xx} - S_{yy}\right)^2 + 4S_{xy}^2}}{2S_{xy}} \tag{11}$$

which is similar to Deming's estimate of the slope. As in the OLS derivation, we define the residual error as the distance between a point and the estimated line. Hence, the residual error for OrthLS is $(x - \hat{x})^2 + \left(Y - \hat{Y}\right)^2$.

3.2.3 MODIFIED LEAST SQUARES

The estimators for two cases, $\gamma = 0$ and $\gamma = 1$, have been derived. As mentioned previously, for most applications, the variance ratio is between 0 and 1. However, a least squares estimator that can be used generally for these cases would be useful. The geometry of the problem is exploited to help define the new estimator.

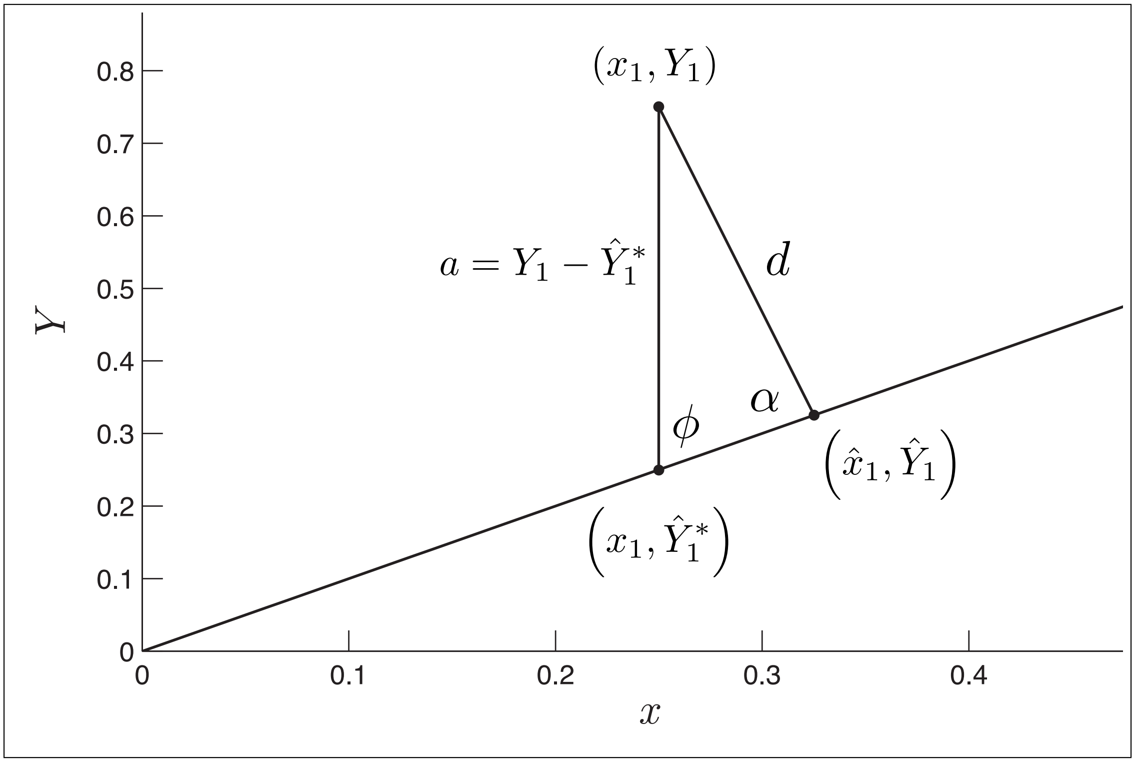

FIG. 6: Distance Minimized by Modified Least Squares

Consider the point (x_1, Y_1) in Figure 6. The vertical line segment between the point and the line is $a = Y_1 - \hat{Y}_1^*$. The angle, ϕ, is made by the line segment of length a and the estimated line as

$$\phi = \frac{\pi}{2} - \tan^{-1}\left(\frac{dy}{dx}\right)$$

where $\dfrac{dy}{dx}$ is the slope of the line or β_1. Next, the line segment d is defined as the distance minimized in MLS. This distance is a function of both the slope of the estimated line and the variance ratio. From d and the estimated line, a second angle, α, is defined as

$$\alpha = \frac{\pi}{2} + (1 - \gamma)\tan^{-1}\left(\frac{dy}{dx}\right)$$

where γ is the variance ratio. The angle α is a first-order approximation based on known angles when $\gamma = 0$ and $\gamma = 1$. From trigonometry, the Law of Sines states

$$\frac{\sin \alpha}{a} = \frac{\sin \phi}{d}.$$

Solving for d yields

$$d = \frac{\sin \phi}{\sin \alpha} a. \tag{12}$$

Using Equation (12), the newly defined objective function for MLS minimizes the sum of the squared lengths of the d_i's or

$$\left[\hat{\beta}_0, \hat{\beta}_1\right] = \min \sum_{i=1}^{n} d_i^2 = \min \sum_{i=1}^{n} \left[\frac{\sin \phi}{\sin \alpha} a\right]^2 \tag{13}$$

Note that as $\alpha \rightarrow \frac{\pi}{2}$, Equation (13) is equivalent to the OrthLS estimator. As in the case of OrthLS, the only additional information required to use Equation (13) is an assumption about the variance ratio. For the calibration applications, the variance ratio varies between 0.0625 and 0.0001.

3.3 SIMULATION STUDY FOR THE SIMPLE LINEAR MODEL

The statistical properties of the OLS estimators are well-known. From the Gauss-Markov theorem, the expected values of $\hat{\beta}_0$ and $\hat{\beta}_1$ are β_0 and β_1, respectively due to the unbiased nature of the OLS estimator (Myers, 1990). Furthermore, the variance of the estimators are

$$\mathrm{Var}\left(\hat{\beta}_0\right) = \sigma_\epsilon^2 \left(\frac{1}{n} + \frac{\bar{x}}{S_{xx}}\right)$$

and

$$\mathrm{Var}\left(\hat{\beta}_1\right) = \frac{\sigma_\epsilon^2}{S_{xx}},$$

and these are minimum variance estimators. To understand the statistical properties of the OrthLS and MLS estimators, a simulation study was conducted to make inferences on the effects of MEs on the prediction capabilities of each method. The prediction capabilities are based on the residual error, which has been redefined to be the variance ratio-weighted distance between any point and the estimated line. This is represented by the distance, d, in Figure 6. The simulation study was designed to be representative of an actual calibration used on simple instruments, such as the pressure transducers for EFT-1 discussed earlier. A calibration experiment is designed with $n = 6$ design points in order to fit the simple linear model

$$y = \beta_0 + \beta_1 x + \epsilon.$$

In the presence of ME, this model becomes

$$y = \beta_0 + \beta_1 (x + u) + \epsilon. \tag{14}$$

The following assumptions were made with regard to Equation (14):

- $\beta_0 = 0$ and $\beta_1 = 1$. Through appropriate centering and scaling of the data, this is always a plausible option in real experiments.

- The errors u and ϵ are independent and identically distributed as normal with means of zero and constant variances of σ_u^2 and σ_ϵ^2, respectively.

- σ_u^2 and σ_ϵ^2 are proportionally related through the variance ratio, γ.

- W_i and Y_i are the error-prone values of x_i and y_i. Therefore, W_i and Y_i are jointly distributed as a bivariate normal distribution

  $$[W_i, Y_i] \sim N\left[\left(x_i, \beta_0 + \beta_1 x_i\right), \Sigma\right]$$

  where $\Sigma = \operatorname{diag}\left(\gamma\sigma_\epsilon^2, \sigma_\epsilon^2\right)$. Both W_i and Y_i are independent random variables.

From these n design points and corresponding responses, the zero-intercept and sensitivity coefficients were estimated, and the mean squared residual errors (MSE) were calculated for each of the estimation methods. The MSE is defined as

$$\text{MSE} = (\text{Bias})^2 + \text{Precision}$$

Next, five (5) confirmation points located at $x = -1, -0.5, 0, 0.5$, and 1 were used to generate confirmation data to further examine the prediction capabilities of each estimated model. For a given calibration design, each method estimates a set of coefficients M times. Each set of estimated coefficients are used L times on the confirmation points and the MSE is calculated.

The simulation study considered three (3) different calibration designs as shown in Table 2. Each design is sufficient for estimating the regression coefficients in the model, and each design has certain advantages and disadvantages over the other designs. Design #1 is a balanced design with replicates throughout the design space.

TABLE 2: Designs Considered for the Simple Linear Simulation Study

Design	Design Points	Comments
Design #1	-1, -1, 0, 0, 1, 1	Replicates throughout Design Space
Design #2	-1, 0, 0, 0, 0, 1	Suspicion of Significant Lack-of-fit
Design #3	-1, -1, -1, 1, 1, 1	Strictly First-order Design

This allowed testing of the constant variance assumption, and for calibration applications, this replication strategy provided insight on the repeatability of the measurement system over the calibration range. Design #2 contains the same number of degrees of freedom for replication as Design #1, but it does not allow testing of constant variance throughout the design space. This design is suited for instances when lack-of-fit is suspected to be significant. Both designs allow for estimation of a second-order model term. Design #3 is a first-order-only calibration design. Replication is dispersed equally at the low and high ends of the range so testing of the constant variance assumption is permissible. However, inferences on the estimation methods are not design-dependent. Table 3 shows the values of γ and σ_ϵ that are used as inputs to the simulation. For variance ratios, the maximum variance ratio studied is $\gamma = 1$, representing errors of equal orders of magnitude. Commonly-assumed

TABLE 3: Variance Ratio and Response Uncertainties Considered for the Simple Linear Simulation Study

Run No.	Variance Ratio, γ	Response Uncertainty, σ_ϵ, % F.S.
1	1 (1:1)	0.2 (20% F.S.)
2	0.25 (2:1)	0.2
3	0.0625 (4:1)	0.2
4	0.01 (10:1)	0.2
5	0.0001 (100:1)	0.2
6	1	0.1 (10% F.S.)
7	0.25	0.1
8	0.0625	0.1
9	0.01	0.1
10	0.0001	0.1
11	1	0.05 (5% F.S.)
12	0.25	0.05
13	0.0625	0.05
14	0.01	0.05
15	0.0001	0.05
16	1	0.01 (1% F.S.)
17	0.25	0.01
18	0.0625	0.01
19	0.01	0.01
20	0.0001	0.01
21	1	0.001 (0.1% F.S.)
22	0.25	0.001
23	0.0625	0.001
24	0.01	0.001
25	0.0001	0.001

ratios for most calibration applications at NASA are also included in the simulation, such at $\gamma = 0.0001, 0.01, 0.0625$, and 0.25. The response uncertainties studied in this simulation are representative of a large population of measurement systems at NASA so the F.S. accuracies range from 0.1 to 20 percent. The simulation study employed a factorial combination of variance ratios and response uncertainties. An outline of the simulation study is provided below:

1. For a given calibration design in Table 2, generate the y_i's from the simple linear model $\beta_0 + \beta_1 x_i$, where $\beta_0 = 0$ and $\beta_1 = 1$.

2. Using (x_i, y_i) for $i = 1, 2, \ldots, 6$, generate (W_i, Y_i) from a bivariate normal distribution. The distribution is $[W_i, Y_i] \sim N\left[(x_i, \beta_0 + \beta_1 x_i), \Sigma\right]$, where $\Sigma = \operatorname{diag}(\gamma\sigma_\epsilon^2, \sigma_\epsilon^2)$ and γ and σ_ϵ are taken from Table 3.

3. From (x_i, Y_i) for $i = 1, 2, \ldots, 6$, estimate $\left(\hat{\beta}_0, \hat{\beta}_1\right)$ for OLS, MLS, and OrthLS. Using $\left(\hat{\beta}_0, \hat{\beta}_1\right)$, compute the predicted value $\hat{y}_i$ for each method and calculate the MSE based on γ and σ_ϵ.

4. For the confirmation points $x_c = [-1.0, -0.5, 0.0, 0.5, 1.0]$, generate the y_{c_i}'s from the simple linear model $\beta_0 + \beta_1 x_{c_i}$, where $\beta_0 = 0$ and $\beta_1 = 1$.

5. Using (x_i, y_i) for $i = 1, 2, \ldots, 5$, generate (W_{c_i}, Y_{c_i}) from a bivariate normal distribution. The distribution is $[W_{c_i}, Y_{c_i}] \sim N\left[(x_i, \beta_0 + \beta_1 x_i), \Sigma\right]$, where $\Sigma = \operatorname{diag}(\gamma\sigma_\epsilon^2, \sigma_\epsilon^2)$ and γ and σ_ϵ are the same values as in Step 2.

6. Using $\left(\hat{\beta}_0, \hat{\beta}_1\right)$, compute the predicted value $\hat{Y}_{c_i}$ for each method and calculate the MSE based on the values γ and σ_ϵ in Step 2.

7. Repeat Steps 5 and 6 for $L = 1000$ times.

8. Repeat Steps 2 through 7 for $M = 100$ times.

9. Repeat Step 8 for $K = 25$ times.

3.3.1 SIMULATION RESULTS

Tables 4 shows the mean estimate of the sensitivity coefficient for the three calibration designs and the three estimation methods. The mean estimates of the zero-intercept are not provided since all three methods yield the same value for a given design, variance ratio, and response uncertainty. From Table 4, the following inferences are made based on the structure of the simulation:

- For a variance ratio of one ($\gamma = 1$), the estimates of $\hat{\beta}_1$ obtained from MLS and OrthLS are equivalent as expected. In the limit as $\gamma \to 0$, the estimate of $\hat{\beta}_1$ from MLS converges towards the OLS estimate.

- For an uncertainty of 1 percent or less ($\sigma_\epsilon \leq 0.01$), the three methods yield numerically-similar estimates of the sensitivity coefficient to four significant figures.

- Larger differences between the estimates are observed when the uncertainty is worse than 5 percent ($\sigma_\epsilon > 0.05$). The magnitude of the difference varies between designs and levels of response uncertainty.

- The difference between the OLS estimate and OrthLS is approximately constant across the levels of variance ratio for a given design and uncertainty. For example, in the first design with an uncertainty of 20 percent, the differences are

  $\gamma = 1 : \Delta = 1.0229 - 1.0031 = 0.0198,$

  $\gamma = 0.25 : \Delta = 1.0177 - 0.9975 = 0.0202,$

  $\gamma = 0.0625 : \Delta = 1.0172 - 0.9967 = 0.0205,$

  $\gamma = 0.01 : \Delta = 1.0173 - 0.9974 = 0.0199,$ and

  $\gamma = 0.0001 : \Delta = 1.0231 - 1.0033 = 0.0198.$

In this example, the differences correspond to a range of approximately 2 percent.

Table 5 presents the variance of the mean estimate of the sensitivity coefficient for the three designs and the three methods. For a given design and level of uncertainty, the variance of the estimates were consistent between the methods and across the levels of variance ratio. Small differences were observed in the magnitude of the variance between the designs for any level of uncertainty, but this could be attributed to the design properties rather than the estimation method. However, from a practical application perspective, using MLS to estimate the sensitivity coefficient resulted in the same magnitude of variability as OLS.

TABLE 4: Mean Estimate of the Sensitivity Coefficient for the Simple Linear Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1			Design #2			Design #3		
			OLS	MLS	OrthLS	OLS	MLS	OrthLS	OLS	MLS	OrthLS
1	1	0.2	1.0031	1.0229	1.0229	1.0075	1.0481	1.0481	1.0026	1.0159	1.0159
2	0.25	0.2	0.9975	1.0075	1.0177	0.9969	1.0172	1.0385	1.0040	1.0107	1.0175
3	0.0625	0.2	0.9967	0.9997	1.0172	1.0008	1.0068	1.0416	1.0001	1.0021	1.0136
4	0.01	0.2	0.9974	0.9979	1.0173	1.0024	1.0034	1.0422	1.0012	1.0015	1.0143
5	0.0001	0.2	1.0033	1.0033	1.0231	0.9971	0.9971	1.0374	0.9993	0.9993	1.0127
6	1	0.1	0.9979	1.0028	1.0028	0.9998	1.0097	1.0097	1.0002	1.0036	1.0036
7	0.25	0.1	1.0004	1.0029	1.0055	0.9989	1.0040	1.0091	0.9996	1.0013	1.0029
8	0.0625	0.1	1.0019	1.0027	1.0070	0.9983	0.9998	1.0083	1.0003	1.0008	1.0035
9	0.01	0.1	1.0004	1.0006	1.0054	1.0000	1.0002	1.0101	0.9998	0.9998	1.0031
10	0.0001	0.1	1.0005	1.0005	1.0055	1.0017	1.0017	1.0121	1.0010	1.0010	1.0044
11	1	0.05	0.9988	1.0000	1.0000	0.9988	1.0013	1.0013	1.0003	1.0011	1.0011
12	0.25	0.05	1.0003	1.0010	1.0016	1.0007	1.0020	1.0032	1.0000	1.0004	1.0008
13	0.0625	0.05	0.9995	0.9997	1.0008	1.0003	1.0006	1.0027	1.0005	1.0007	1.0014
14	0.01	0.05	1.0005	1.0006	1.0018	1.0003	1.0004	1.0028	1.0003	1.0003	1.0011
15	0.0001	0.05	0.9998	0.9998	1.0010	0.9999	0.9999	1.0025	0.9998	0.9998	1.0006
16	1	0.01	1.0000	1.0000	1.0000	1.0001	1.0002	1.0002	1.0000	1.0000	1.0000
17	0.25	0.01	1.0000	1.0000	1.0000	0.9999	0.9999	0.9999	1.0000	1.0000	1.0000
18	0.0625	0.01	0.9999	1.0000	1.0000	0.9999	0.9999	1.0000	1.0000	1.0000	1.0000
19	0.01	0.01	1.0000	1.0000	1.0001	0.9998	0.9998	0.9999	0.9999	0.9999	0.9999
20	0.0001	0.01	1.0000	1.0000	1.0000	0.9998	0.9998	0.9999	0.9999	0.9999	0.9999
21	1	0.001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
22	0.25	0.001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
23	0.0625	0.001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
24	0.01	0.001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
25	0.0001	0.001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

TABLE 5: Variance of the Mean Estimate of the Sensitivity Coefficient for the Simple Linear Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1			Design #2			Design #3		
			OLS	MLS	OrthLS	OLS	MLS	OrthLS	OLS	MLS	OrthLS
1	1	0.2	0.0099	0.0101	0.0101	0.0209	0.0217	0.0217	0.0065	0.0066	0.0066
2	0.25	0.2	0.0096	0.0096	0.0097	0.0212	0.0214	0.0217	0.0067	0.0067	0.0068
3	0.0625	0.2	0.0099	0.0100	0.0101	0.0202	0.0203	0.0208	0.0063	0.0063	0.0064
4	0.01	0.2	0.0099	0.0099	0.0102	0.0195	0.0195	0.0203	0.0068	0.0068	0.0069
5	0.0001	0.2	0.0105	0.0105	0.0106	0.0199	0.0199	0.0205	0.0068	0.0068	0.0068
6	1	0.1	0.0025	0.0025	0.0025	0.0049	0.0049	0.0049	0.0017	0.0017	0.0017
7	0.25	0.1	0.0025	0.0025	0.0025	0.0050	0.0050	0.0051	0.0017	0.0017	0.0017
8	0.0625	0.1	0.0025	0.0025	0.0025	0.0051	0.0051	0.0051	0.0018	0.0018	0.0018
9	0.01	0.1	0.0025	0.0025	0.0025	0.0050	0.0050	0.0051	0.0017	0.0017	0.0017
10	0.0001	0.1	0.0026	0.0026	0.0026	0.0051	0.0051	0.0051	0.0018	0.0018	0.0018
11	1	0.05	0.0006	0.0006	0.0006	0.0012	0.0012	0.0012	0.0004	0.0004	0.0004
12	0.25	0.05	0.0006	0.0006	0.0006	0.0013	0.0013	0.0013	0.0004	0.0004	0.0004
13	0.0625	0.05	0.0006	0.0006	0.0006	0.0013	0.0013	0.0013	0.0004	0.0004	0.0004
14	0.01	0.05	0.0006	0.0006	0.0006	0.0012	0.0012	0.0012	0.0004	0.0004	0.0004
15	0.0001	0.05	0.0006	0.0006	0.0006	0.0013	0.0013	0.0013	0.0004	0.0004	0.0004
16	1	0.01	0.0000	0.0000	0.0000	0.0000	0.0001	0.0001	0.0000	0.0000	0.0000
17	0.25	0.01	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001	0.0000	0.0000	0.0000
18	0.0625	0.01	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001	0.0000	0.0000	0.0000
19	0.01	0.01	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
20	0.0001	0.01	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001	0.0000	0.0000	0.0000
21	1	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
22	0.25	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
23	0.0625	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24	0.01	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0001	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 6 shows the average MSE across all the design and confirmation points for the three designs and the three methods. Recall that the MSE includes both bias in the prediction and precision of the prediction. The root mean squared error (RMSE) is the square root of MSE and is typically used to quote the estimated accuracy of a measurement system after calibration. For example, the 1-σ estimated measurement system accuracy for Design #1, $\gamma = 1$, and $\sigma_\epsilon = 0.2$ based on OLS is

$$\hat{\sigma} = \frac{\sqrt{\text{Mean Squared Error}}}{\text{Full-Scale Range}} = \frac{\sqrt{0.0132}}{1} = 0.115$$

or 11.5 percent of F.S. While Table 6 shows the absolute MSE values, Table 7 reveals the difference between OLS and either MLS or OrthLS. From this table, it is evident that MLS has a smaller MSE than OLS in all the cases studied. Additionally, OrthLS shows a benefit over OLS in most cases when the variance ratio is greater than or equal to 0.25. These differences are expressed in terms of percent improvement in Table 8. Both MLS and OrthLS provide up to a 3 percent reduction in MSE over OLSE, such as in the case of $\gamma = 1$ and $\sigma_\epsilon = 0.2$. As the variance ratio departed from unity, the relative improvement of MLS decreased while OrthLS showed negative impact at small values of γ. Through the use of one-way analysis of variance (ANOVA), the differences observed between the methods were tested for statistical significance. ANOVA compares the means of two or more groups in the presence of the variability within each of the groups (Casella and Berger, 2002). Using a criteria of 0.05, the hypothesis that the MSE between methods are equal is rejected if the p-value from the ANOVA is less than 0.05. ANOVA does not directly indicate which groups are different or by how much the groups differ. The ANOVA is shown in Tables 9-11. Separately, the tables reveal that the difference in MSE is not statistically detectable since the p-value is near one. When comparing the three hypothesis tests simultaneously, the rejection criteria becomes

$$\alpha_{\text{total}} = 1 - (1 - \alpha_{\text{indiv}})^n = 1 - (1 - 0.05)^3 = 0.14$$

where n is the number of simultaneous tests. This value, also known as the family-wise error rate, is the probability of making a Type I error across all the hypothesis tests. However, the conclusion remains that there is no detectable differences in the MSE between the three methods.

Table 12 emphasizes a practical interpretation of the differences in the MSE. The percent improvement in MSE relative to the uncertainty in the response is calculated by

$$\frac{\sqrt{\text{Improvement in Mean Squared Error}}}{\sigma_\epsilon} \times 100.$$

If the improvement in MSE is less than zero, then the percent improvement is zero. These results show that the impact of MLS is greater when comparing the improvement to the uncertainty in the response for the three calibration designs studied. For example, a 0.01 percent improvement in MSE is achieved on a measurement system with a 0.1 percent response uncertainty using the first calibration design and a variance ratio of 0.0625. This difference is considered practically significant. In summary, while the statistical hypothesis tests revealed no difference in the MSE between the methods, the practical benefit of MLS is evident in Table 12. For variance ratios greater than 0.0625, it is recommended that MLS be employed instead of OLS.

TABLE 6: Mean Squared Error for the Simple Linear Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1			Design #2			Design #3		
			OLS	MLS	OrthLS	OLS	MLS	OrthLS	OLS	MLS	OrthLS
1	1	0.2	0.0132	0.0130	0.0130	0.0134	0.0130	0.0130	0.0134	0.0132	0.0132
2	0.25	0.2	0.0196	0.0195	0.0196	0.0200	0.0198	0.0200	0.0195	0.0195	0.0195
3	0.0625	0.2	0.0250	0.0250	0.0252	0.0245	0.0245	0.0251	0.0246	0.0246	0.0248
4	0.01	0.2	0.0262	0.0262	0.0265	0.0259	0.0259	0.0266	0.0260	0.0260	0.0262
5	0.0001	0.2	0.0265	0.0265	0.0269	0.0265	0.0265	0.0274	0.0269	0.0269	0.0271
6	1	0.1	0.0033	0.0033	0.0033	0.0033	0.0033	0.0033	0.0034	0.0034	0.0034
7	0.25	0.1	0.0049	0.0049	0.0049	0.0049	0.0049	0.0049	0.0048	0.0048	0.0048
8	0.0625	0.1	0.0062	0.0062	0.0062	0.0061	0.0061	0.0062	0.006	0.006	0.006
9	0.01	0.1	0.0066	0.0066	0.0066	0.0067	0.0067	0.0067	0.0065	0.0065	0.0065
10	0.0001	0.1	0.0067	0.0067	0.0068	0.0069	0.0069	0.007	0.0068	0.0068	0.0068
11	1	0.05	0.0008	0.0008	0.0008	0.0009	0.0009	0.0009	0.0008	0.0008	0.0008
12	0.25	0.05	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012	0.0012
13	0.0625	0.05	0.0016	0.0016	0.0016	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015
14	0.01	0.05	0.0016	0.0016	0.0016	0.0017	0.0017	0.0017	0.0016	0.0016	0.0016
15	0.0001	0.05	0.0017	0.0017	0.0017	0.0017	0.0017	0.0017	0.0016	0.0016	0.0016
16	1	0.01	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
17	0.25	0.01	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
18	0.0625	0.01	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
19	0.01	0.01	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
20	0.0001	0.01	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
21	1	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
22	0.25	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
23	0.0625	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24	0.01	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0001	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

TABLE 7: Improvement in Mean Squared Error over OLS for the Simple Linear Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1		Design #2		Design #3	
			MLS	OrthLS	MLS	OrthLS	MLS	OrthLS
1	1	0.2	0.0002	0.0002	0.0004	0.0004	0.0001	0.0001
2	0.25	0.2	0.0001	0.0000	0.0001	0.0000	0.0000*	0.0000*
3	0.0625	0.2	0.0000*	-0.0003	0.0000*	-0.0005	0.0000*	-0.0002
4	0.01	0.2	0.0000*	-0.0004	0.0000*	-0.0008	0.0000*	-0.0002
5	0.0001	0.2	0.0000*	-0.0004	0.0000*	-0.0008	0.0000*	-0.0003
6	1	0.1	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*
7	0.25	0.1	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*
8	0.0625	0.1	0.0000*	0.0000	0.0000*	0.0000	0.0000*	0.0000
9	0.01	0.1	0.0000*	0.0000	0.0000*	0.0000	0.0000*	0.0000
10	0.0001	0.1	0.0000*	0.0000	0.0000*	-0.0001	0.0000*	0.0000
11	1	0.05	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*
12	0.25	0.05	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*
13	0.0625	0.05	0.0000*	0.0000	0.0000*	0.0000	0.0000*	0.0000
14	0.01	0.05	0.0000*	0.0000	0.0000*	0.0000	0.0000*	0.0000
15	0.0001	0.05	0.0000*	0.0000	0.0000*	0.0000	0.0000*	0.0000
16	1	0.01	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*
17	0.25	0.01	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*
18	0.0625	0.01	0.0000*	0.0000	0.0000*	0.0000	0.0000*	0.0000
19	0.01	0.01	0.0000*	0.0000	0.0000*	0.0000	0.0000*	0.0000
20	0.0001	0.01	0.0000*	0.0000	0.0000*	0.0000	0.0000*	0.0000
21	1	0.001	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*
22	0.25	0.001	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*	0.0000*
23	0.0625	0.001	0.0000*	0.0000	0.0000*	0.0000	0.0000*	0.0000
24	0.01	0.001	0.0000*	0.0000	0.0000*	0.0000	0.0000*	0.0000
25	0.0001	0.001	0.0000*	0.0000	0.0000*	0.0000	0.0000*	0.0000

Note: Asterisk represents an improvement of less than 0.00005

TABLE 8: Percent Improvement in Mean Squared Error over OLS for the Simple Linear Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1		Design #2		Design #3	
			MLS	OrthLS	MLS	OrthLS	MLS	OrthLS
1	1	0.2	1.46%	1.46%	2.91%	2.91%	0.99%	0.99%
2	0.25	0.2	0.37%	-0.01%	0.75%	-0.04%	0.25%	0.00%*
3	0.0625	0.2	0.03%	-1.08%	0.07%	-2.19%	0.02%	-0.70%
4	0.01	0.2	0.00%*	-1.42%	0.00%*	-2.91%	0.00%*	-0.94%
5	0.0001	0.2	0.00%*	-1.54%	0.00%*	-3.15%	0.00%*	-1.00%
6	1	0.1	0.37%	0.37%	0.72%	0.72%	0.25%	0.25%
7	0.25	0.1	0.10%	0.00%*	0.19%	0.00%*	0.06%	0.00%*
8	0.0625	0.1	0.01%	-0.26%	0.02%	-0.53%	0.01%	-0.17%
9	0.01	0.1	0.00%*	-0.36%	0.00%*	-0.71%	0.00%*	-0.23%
10	0.0001	0.1	0.00%*	-0.37%	0.00%*	-0.77%	0.00%*	-0.25%
11	1	0.05	0.09%	0.09%	0.19%	0.19%	0.06%	0.06%
12	0.25	0.05	0.02%	0.00%*	0.05%	0.00%*	0.02%	0.00%*
13	0.0625	0.05	0.00%*	-0.07%	0.00%*	-0.13%	0.00%*	-0.04%
14	0.01	0.05	0.00%*	-0.09%	0.00%*	-0.18%	0.00%*	-0.06%
15	0.0001	0.05	0.00%*	-0.09%	0.00%*	-0.19%	0.00%*	-0.06%
16	1	0.01	0.00%*	0.00%*	0.01%	0.01%	0.00%*	0.00%*
17	0.25	0.01	0.00%*	0.00%*	0.00%*	0.00%*	0.00%*	0.00%*
18	0.0625	0.01	0.00%*	0.00%	0.00%*	-0.01%	0.00%*	0.00%
19	0.01	0.01	0.00%*	0.00%	0.00%*	-0.01%	0.00%*	0.00%
20	0.0001	0.01	0.00%*	0.00%	0.00%*	-0.01%	0.00%*	0.00%
21	1	0.001	0.00%*	0.00%*	0.00%*	0.00%*	0.00%*	0.00%*
22	0.25	0.001	0.00%*	0.00%*	0.00%*	0.00%*	0.00%*	0.00%*
23	0.0625	0.001	0.00%*	0.00%	0.00%*	0.00%	0.00%*	0.00%
24	0.01	0.001	0.00%*	0.00%	0.00%*	0.00%	0.00%*	0.00%
25	0.0001	0.001	0.00%*	0.00%	0.00%*	0.00%	0.00%*	0.00%

Note: Asterisk represents an improvement of less than 0.005%

TABLE 9: Analysis of Variance of the Mean Squared Error for the Simple Linear Simulation – Design #1

Source	DF	SS	MS	F	P
Method (OLS, MLS, OrthLS)	2	0.0000002	0.0000001	0.00	0.999
Error	447	0.0358240	0.0000801		
Total	449	0.0358242			

TABLE 10: Analysis of Variance of the Mean Squared Error for the Simple Linear Simulation – Design #2

Source	DF	SS	MS	F	P
Method (OLS, MLS, OrthLS)	2	0.0000008	0.0000004	0.00	0.996
Error	447	0.0397981	0.0000890		
Total	449	0.0397989			

TABLE 11: Analysis of Variance of the Mean Squared Error for the Simple Linear Simulation – Design #3

Source	DF	SS	MS	F	P
Method (OLS, MLS, OrthLS)	2	0.0000001	0.0000000	0.00	0.999
Error	447	0.0342407	0.0000766		
Total	449	0.0342408			

TABLE 12: Percent Improvement in Mean Squared Error Relative to Response Uncertainty for the Simple Linear Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1		Design #2		Design #3	
			MLS	OrthLS	MLS	OrthLS	MLS	OrthLS
1	1	0.2	6.94%	6.94%	9.87%	9.87%	5.75%	5.75%
2	0.25	0.2	4.25%	0.00%	6.12%	0.00%	3.50%	0.13%
3	0.0625	0.2	1.46%	0.00%	2.03%	0.00%	1.18%	0.00%
4	0.01	0.2	0.25%	0.00%	0.35%	0.00%	0.20%	0.00%
5	0.0001	0.2	0.00%*	0.00%	0.00%*	0.00%	0.00%*	0.00%
6	1	0.1	3.53%	3.53%	4.90%	4.90%	2.91%	2.91%
7	0.25	0.1	2.17%	0.20%	3.06%	0.18%	1.72%	0.19%
8	0.0625	0.1	0.73%	0.00%	1.03%	0.00%	0.58%	0.00%
9	0.01	0.1	0.13%	0.00%	0.18%	0.00%	0.10%	0.00%
10	0.0001	0.1	0.00%*	0.00%	0.00%*	0.00%	0.00%*	0.00%
11	1	0.05	1.75%	1.75%	2.58%	2.58%	1.43%	1.43%
12	0.25	0.05	1.09%	0.14%	1.52%	0.17%	0.88%	0.11%
13	0.0625	0.05	0.37%	0.00%	0.50%	0.00%	0.29%	0.00%
14	0.01	0.05	0.06%	0.00%	0.09%	0.00%	0.05%	0.00%
15	0.0001	0.05	0.00%*	0.00%	0.00%*	0.00%	0.00%*	0.00%
16	1	0.01	0.36%	0.36%	0.51%	0.51%	0.29%	0.29%
17	0.25	0.01	0.22%	0.03%	0.30%	0.04%	0.17%	0.02%
18	0.0625	0.01	0.07%	0.00%	0.10%	0.00%	0.06%	0.00%
19	0.01	0.01	0.01%	0.00%	0.02%	0.00%	0.01%	0.00%
20	0.0001	0.01	0.00%*	0.00%	0.00%*	0.00%	0.00%*	0.00%
21	1	0.001	0.04%	0.04%	0.05%	0.05%	0.03%	0.03%
22	0.25	0.001	0.02%	0.00%*	0.03%	0.00%*	0.02%	0.00%*
23	0.0625	0.001	0.01%	0.00%	0.01%	0.00%	0.01%	0.00%
24	0.01	0.001	0.00%*	0.00%	0.00%*	0.00%	0.00%*	0.00%
25	0.0001	0.001	0.00%*	0.00%	0.00%*	0.00%	0.00%*	0.00%

Note: Asterisk represents an improvement of less than 0.005%

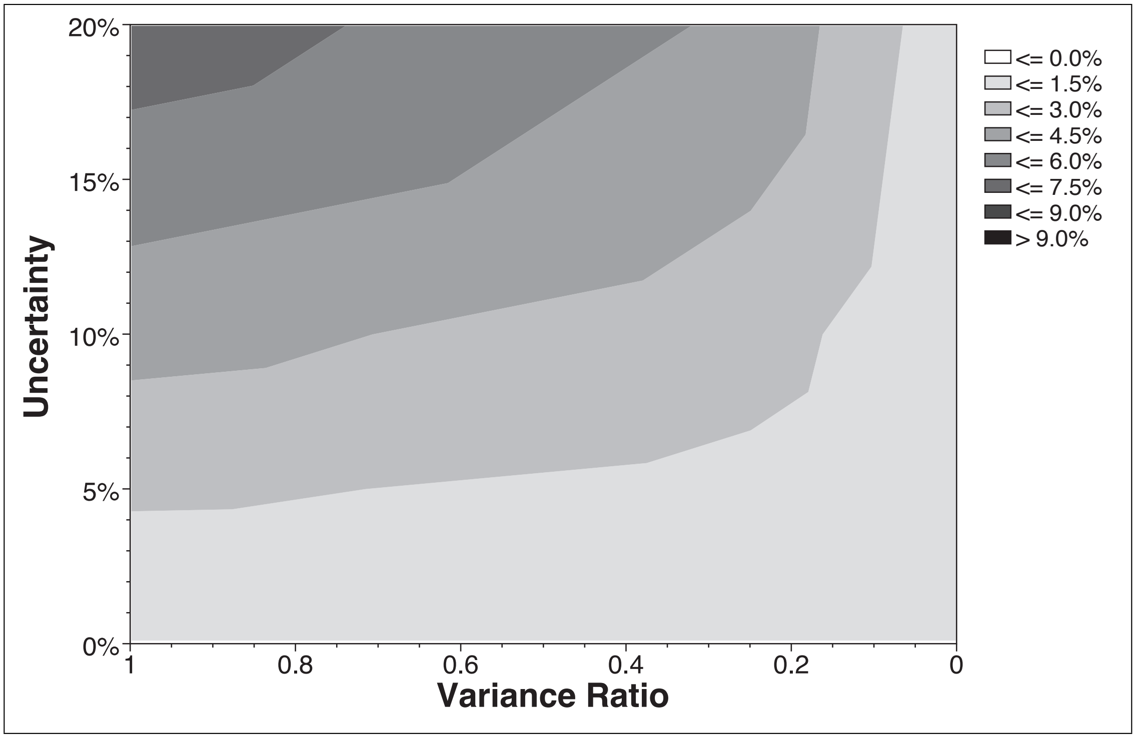

FIG. 7: Percent Improvement in Mean Squared Error of MLS for the Simple Linear Simulation – Design #1

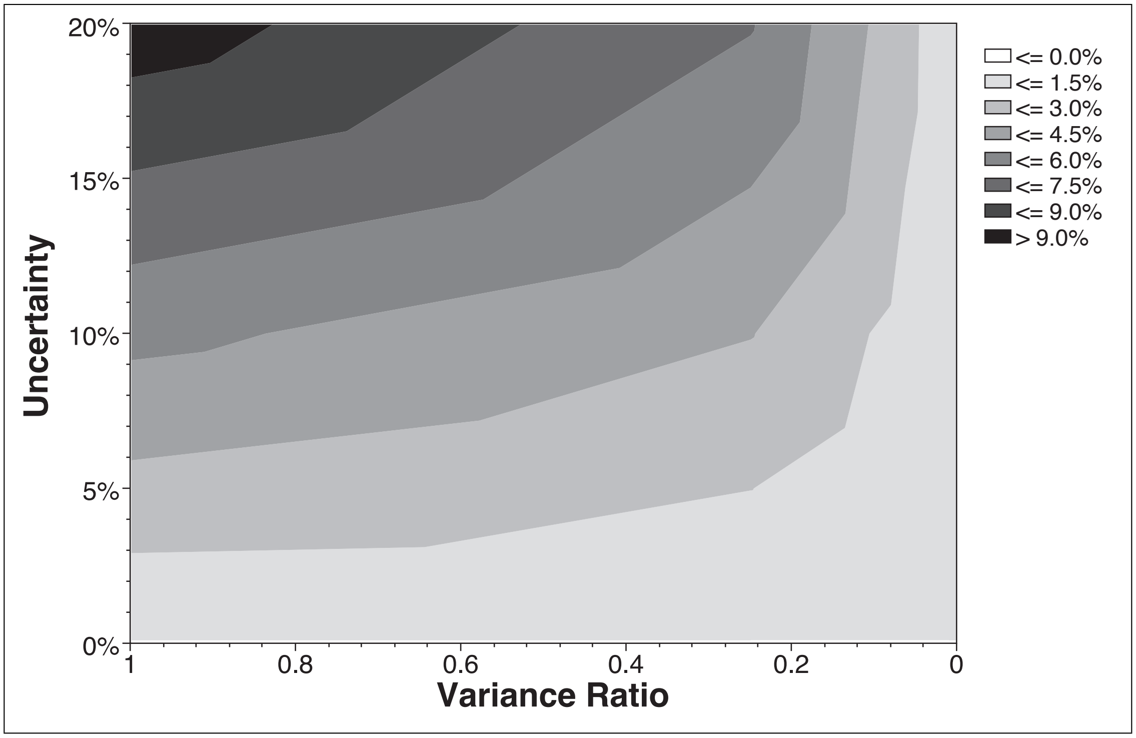

FIG. 8: Percent Improvement in Mean Squared Error of MLS for the Simple Linear Simulation – Design #2

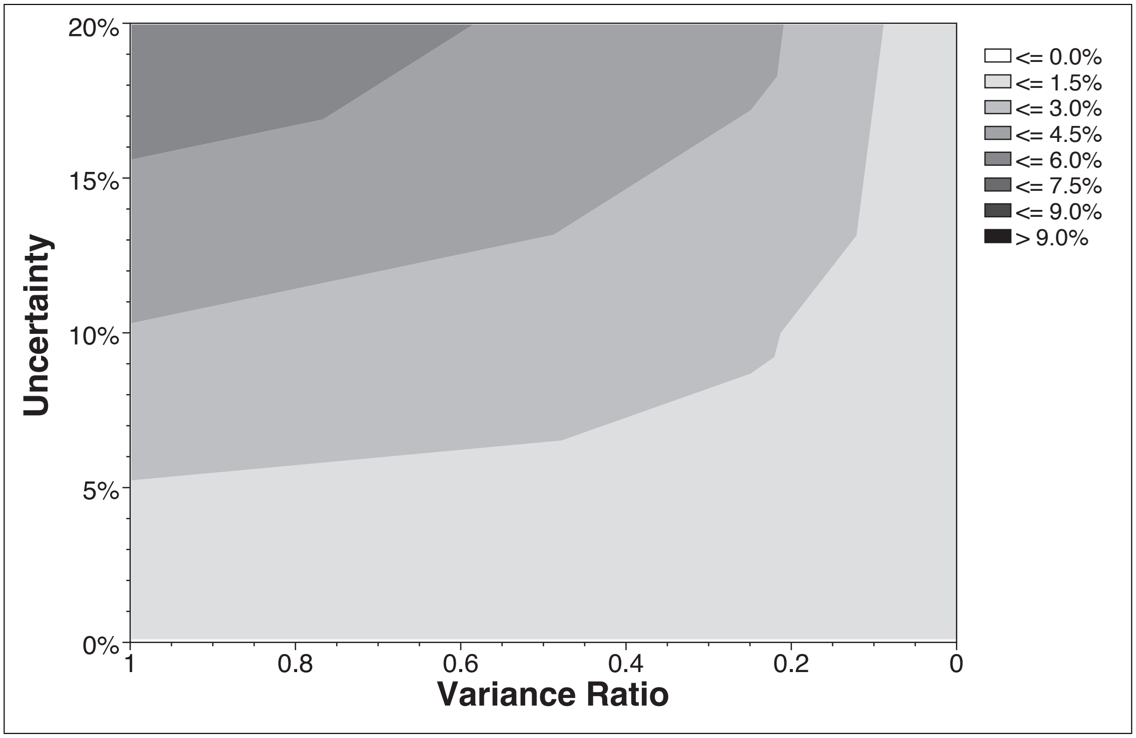

FIG. 9: Percent Improvement in Mean Squared Error of MLS for the Simple Linear Simulation – Design #3

CHAPTER 4

MULTI-DIMENSIONAL, HIGHER-ORDER MODELS

4.1 APPLICATION OF THE SIMPLE POLYNOMIAL MODEL

In the previous chapter, the simple linear model was discussed in detail, and modified least squares (MLS) was proposed as an alternative method to ordinary least squares (OLS) for estimating the regression coefficients in the presence of measurement errors. The statistical properties of the estimator and the prediction capabilities of MLS were studied through simulation along with OLS and orthogonal least squares (OrthLS). Results showed that the numerical differences in estimating the sensitivity coefficient were small between the three methods. Furthermore, differences in the mean squared error (MSE) were also small between the methods, and through the use of analysis of variance (ANOVA), the differences were deemed statistically undetectable. However, the practical implications of the different methods were pronounced. For the three designs considered, MLS showed a substantial improvement in MSE over OLS relative to the uncertainty in the response. As an example, MLS had a 0.06 to 0.51 percent reduction for an response uncertainty of 0.01, depending on the level of the variance ratio. Based on these results, it was recommended to employ MLS for simple linear models when the variance ratio was larger than 0.0625.

While the simplest form of the estimated model is desired, many applications are more complex and include additional factors or higher-order model terms. For now, the focus is limited to single-factor, polynomial models. The multiple-factor case is discussed shortly. The single-factor, polynomial model is identified as the simple polynomial model within this research. For OLS estimation of a simple polynomial model, the theory is readily available and is briefly reviewed in this chapter (Myers, 1990). Methods comparable to OrthLS for simple polynomial models are underdeveloped within the literature. Other estimation techniques, such as those based on the Method of Moments, for the simple polynomial model are limited in practical application and available in popular ME references (Fuller, 1987; Buonaccorsi, 2010). The methods of OrthLS and MLS discussed in the previous chapter provide the necessary framework to expand upon for simple polynomial models. Developing these methods for other model forms is important to understanding the implications of using naive estimation methods, such as OLS.

In this research, the focus is on second-order polynomial models, which are used in several calibration applications. Similar to the simple linear problem, the goal is to develop an engineering solution that can be easily applied to practical situations where the estimated model is a polynomial. Consider again the calibration of the pressure transducers for EFT-1. Due to certain properties of the sensing element inside the transducer, the applied pressure is suspected to have a nonlinear effect on the response. The magnitude of this nonlinear effect is small relative to the primary sensitivity of the transducer but nonetheless is significant in affecting the response. Therefore, the simple linear model may not be sufficient in capturing the functional behavior between x and y. A set of n design points are used to estimate the regression coefficients in the model, and the assumed appropriate form of the model is now

$$y_i = \beta_0 + \beta_1 x_i + \beta_{11} x_i^2 + \epsilon_i \tag{15}$$

where β_0 is the zero-offset, β_1 is the sensitivity, β_{11} is the nonlinear effect, and ϵ is the normally-distributed random error of y with a mean of zero and a constant variance of σ_ϵ^2. The applied pressures are only known to a certain accuracy and as a result, the actual applied pressure is

$$W = x + u$$

where u is the uncertainty in the reference standard as estimated from the NIST-traceable calibration. With MEs, Equation (15) becomes

$$y_i = \beta_0 + \beta_1 (x_i + u_i) + \beta_{11} (x_i + u_i)^2 + \epsilon_i \tag{16}$$

where u_i is distributed with a mean of zero and constant variance of σ_u^2. Since the value of u at a value of x is unknown, the variance ratio is used to develop both MLS and OrthLS as alternative estimation methods to OLS in the presence of MEs.

When comparing the effect of MEs on the simple linear and quadratic models, there are a few important points to remember. While the assumed form of the ME model is additive, the expected value and variance of the response are affected by the order of the model, as discussed earlier in Chapter 1. For convenience, the results are summarized for the simple linear and quadratic models below in Table 13. In the simple linear model, the expected value for classical and measurement errors are similar while the variance is inflated due to the ME. Conversely, both the expected value and variance of y are affected by the ME in the quadratic model. The expected value of y is biased by $\beta_{11}\sigma_u^2$ and the variance of y is further inflated by $\sigma_u^2 (1 + 4\beta_{11}^2 x_1^2)$.

TABLE 13: Comparison of Bias and Variance of Simple Linear and Quadratic Models

		Classical Error	Measurement Error
Simple Linear Model	$E(y)$	$\beta_0 + \beta_1 x_1$	$\beta_0 + \beta_1 x_1$
	$\mathrm{Var}(y)$	σ_ϵ^2	$\sigma_\epsilon^2 + \beta_1^2 \sigma_u^2$
Simple Quadratic Model	$E(y)$	$\beta_0 + \beta_1 x_1 + \beta_{11} x_1^2$	$\beta_0 + \beta_1 x_1 + \beta_{11} (x_1^2 + \sigma_u^2)$
	$\mathrm{Var}(y)$	σ_ϵ^2	$\sigma_\epsilon^2 + \sigma_u^2 (1 + \beta_1^2 + 4\beta_{11}^2 x_1^2)$

The objective function within each method is a geometric interpretation. For instance, OLS minimizes the sum of the squared differences between the observed response, Y_i, and the predicted response, $\hat{Y}_i$, over n design points. Geometrically, this is equivalent to the Y-distance between a point and the estimated line. Alternatively, OrthLS minimized the sum of the squared orthogonal distances between n design points and the estimated line. MLS minimizes the sum of the squared variance-weighted distances and is bound at the extremes by OLS and OrthLS. One advantage of OLS in practical applications is that the minimization yields a closed-form set of equations that are readily solvable. The other two methods lend themselves to numerical solutions. Because of this, OLS is simpler to use in practice and most basic software packages contain OLS regression.

For the simple quadratic model given by Equation (15), Figures 10-13 demonstrate the effect of variance ratio on the distribution of errors in x and Y. In Figure 10 where $\gamma = 0$, it is obvious that the errors are again one-dimensional and because of this, OLS is a sufficient estimator of the model. Additionally, while the curvature is small, it is still observed in Figure 10. As $\gamma \rightarrow 1$, it is seen that both the slope and curvature are affected by the errors in x and OLS is no longer a sufficient estimator since the errors are two-dimensional. In the simple linear case, only the slope was affected by any MEs. Furthermore, the errors in x can also disguise any departures from linearity, as seen in Figure 13.

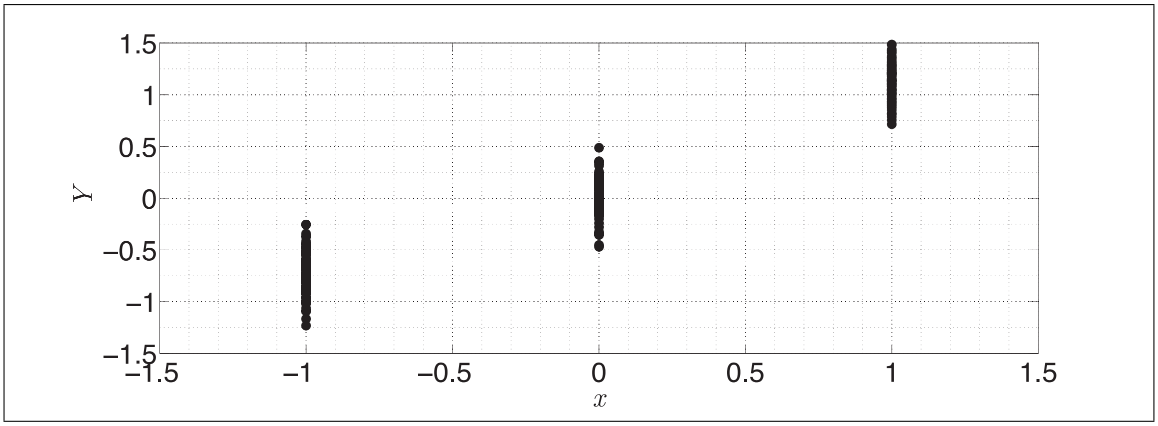

FIG. 10: Spatial Dispersion of Errors for the Simple Quadratic Relationship for $\gamma = 0$

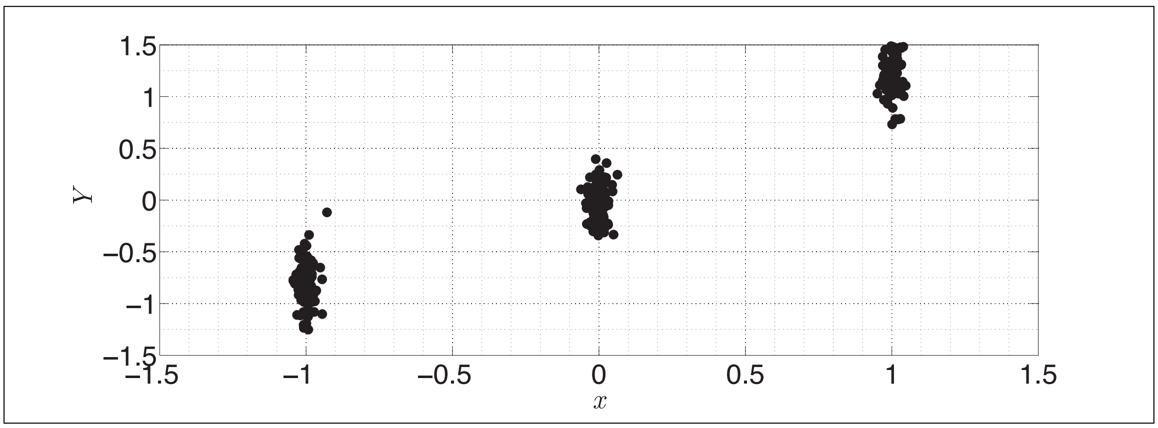

FIG. 11: Spatial Dispersion of Errors for the Simple Quadratic Relationship for $\gamma = 0.01$

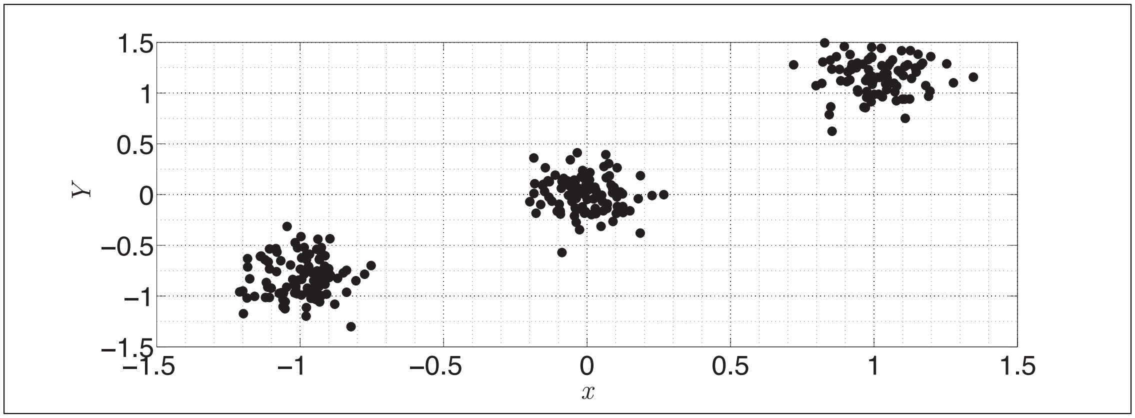

FIG. 12: Spatial Dispersion of Errors for the Simple Quadratic Relationship for $\gamma = 0.25$

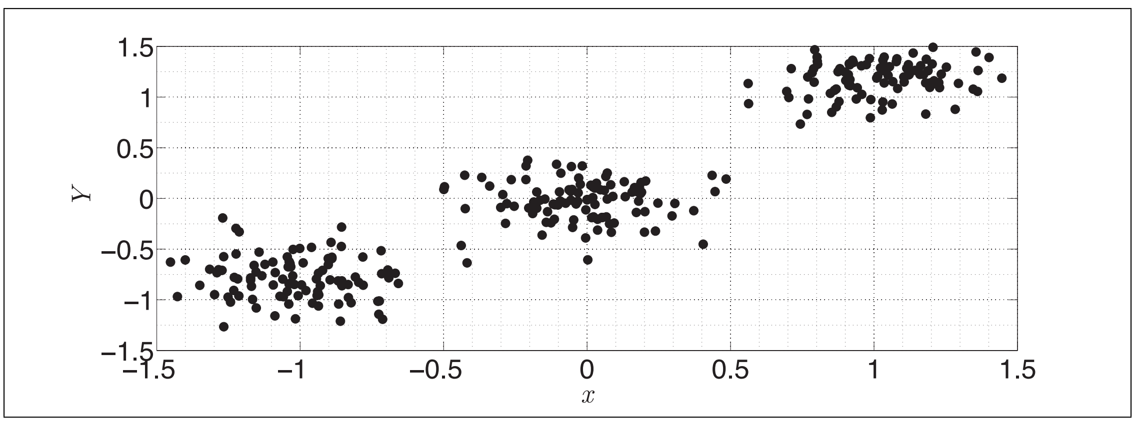

FIG. 13: Spatial Dispersion of Errors for the Simple Quadratic Relationship for $\gamma = 1$

4.2 ESTIMATION METHODS FOR THE SIMPLE QUADRATIC MODEL

For the simple quadratic measurement system characterization problem, the methods of OLS, MLS, and OrthLS are again discussed. As a note, the OLS estimator derived shortly is applicable to both polynomial and multi-dimensional response surface models.

4.2.1 ORDINARY LEAST SQUARES

Consider the calibration of a measurement system with k factors of interest and n design points are executed during the calibration. These n design points are sufficient in estimating a model of the general form

$$Y_i = \beta_0 + \beta_1 x_{1_i} + \ldots + \beta_p x_{p_i} \tag{17}$$

where $Y_i = y_i + \epsilon_i$ and p is the order of the equation. Equation (17) can be expressed more generally in matrix form as

$$\mathbf{Y} = \mathbf{x}\boldsymbol{\beta}$$

where $\mathbf{Y}$ is a $(n \times 1)$ vector of responses with error, $\mathbf{x}$ is a $[n \times (p+1)]$ expanded model matrix, and $\boldsymbol{\beta}$ is a $[(p+1) \times 1]$ vector of regression coefficients. In matrix form, OLS minimizes

$$\hat{\boldsymbol{\beta}} = \min \left[\left(\mathbf{Y} - \hat{\mathbf{Y}} \right)' \left(\mathbf{Y} - \hat{\mathbf{Y}} \right) \right] = \min \left[\left(\mathbf{Y} - \mathbf{x}\hat{\boldsymbol{\beta}} \right)' \left(\mathbf{Y} - \mathbf{x}\hat{\boldsymbol{\beta}} \right) \right].$$

The minimum is found by differentiating with respect to $\boldsymbol{\beta}$ and setting the resulting equation to zero or

$$\frac{\partial}{\partial \hat{\boldsymbol{\beta}}} \left[\left(\mathbf{Y} - \mathbf{x}\hat{\boldsymbol{\beta}} \right)' \left(\mathbf{Y} - \mathbf{x}\hat{\boldsymbol{\beta}} \right) \right] = 0.$$

Performing the indicated differentiation yields

$$-2\mathbf{x}'\mathbf{Y} + 2\left(\mathbf{x}'\mathbf{x}\right)\hat{\boldsymbol{\beta}} = 0.$$

Therefore, the least squares normal equations are

$$\left(\mathbf{x}'\mathbf{x}\right)\hat{\boldsymbol{\beta}} = \mathbf{x}'\mathbf{Y}.$$

Finally, solving for $\hat{\boldsymbol{\beta}}$ gives

$$\hat{\boldsymbol{\beta}} = (\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'\mathbf{Y} \tag{18}$$

where $\mathbf{x}$ must be of full column rank. The matrix $(\mathbf{x}'\mathbf{x})^{-1}$ is commonly known as the variance-covariance matrix of the estimated regression coefficients (Myers, 1990). The elements of this matrix are important to the statistical properties of the estimated coefficients. Furthermore, the failure of $\mathbf{x}$ to be of full column rank leads to a matrix that cannot be inverted; namely, $(\mathbf{x}'\mathbf{x})$.

4.2.2 ORTHOGONAL AND MODIFIED LEAST SQUARES

At this point, the OLS estimator derived in the previous section addresses the case when the variance ratio is zero. Next, a more general estimator is derived that can be employed when the variance ratio is greater than zero. Based on the results of the simple linear model, the new estimator is expected to perform similarly. The approach used to derive the MLS method in the simple linear model is expanded upon to derive both MLS and OrthLS for the simple quadratic model. The only distinction between the two methods is that the variance ratio is set equal to one for OrthLS.

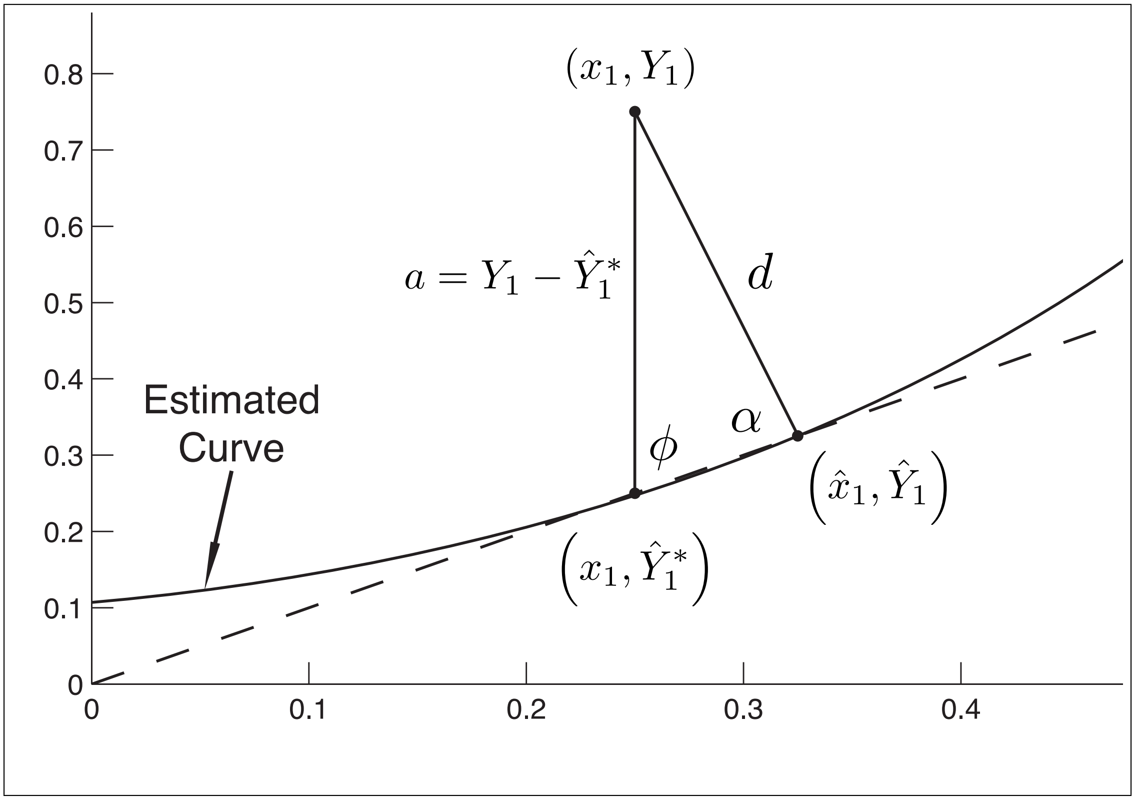

FIG. 14: Distance Minimized by Modified and Orthogonal Least Squares for the Simple Polynomial Model

Consider the point (x_1, Y_1) in Figure 14. The solid curve in the figure represents the estimated curve, which is of the form

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_{11} x^2.$$

As mentioned previously, there are restrictions on how large $\hat{\beta}_{11}$ can be relative to $\hat{\beta}_1$. In normalized units, $\hat{\beta}_{11}$ is restricted to values of -0.2 to 0.2. This assumption is typically valid in measurement system applications. The vertical line segment, a, represents the one-dimensional distance between the data point and the estimated curve, which is the distance minimized in OLS. The point $\left(x_1, \hat{Y}_1^*\right)$ is located where the line segment a intersects the estimated curve. At the point $\left(x_1, \hat{Y}_1^*\right)$, the derivative of the estimated curve is evaluated. For a simple quadratic model, the derivative is

$$\frac{dy\left(x_1\right)}{dx} = \hat{\beta}_1 + 2\hat{\beta}_{11}x_1,$$

which is the local slope of the tangent line. This slope determines the angle, ϕ, given by

$$\phi = \frac{\pi}{2} - \tan^{-1}\left(\frac{dy\left(x_1\right)}{dx}\right) = \frac{\pi}{2} - \tan^{-1}\left(\hat{\beta}_1 + 2\hat{\beta}_{11}x_1\right).$$

It is clear that ϕ in the quadratic model is a function of the location in the design space. From the line segment d and the slope of the tangent line, the angle α is defined as

$$\alpha = \frac{\pi}{2} + (1-\gamma)\tan^{-1}\left(\frac{dy\left(x_1\right)}{dx}\right)$$

which is the same first-order approximation used in the simple linear case. The distinction between MLS and OrthLS is due to the angle α being a function of the variance ratio γ. For OrthLS, $\gamma = 1$; thus, $\alpha = \frac{\pi}{2}$. The value of α in MLS varies between $\frac{\pi}{2}$ and $\frac{\pi}{2} + \tan^{-1}\left(\hat{\beta}_1 + 2\hat{\beta}_{11}x_1\right)$. Exploiting the geometry of the problem, the Law of Sines yields

$$d = \frac{\sin\phi}{\sin\alpha}a$$

where a can be written as $Y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1 - \hat{\beta}_{11}x_1^2$. The MLS estimates are obtained by minimizing the squared length of d or

$$\left[\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_{11}\right] = \min\sum_{i=1}^{n} d_i^2 = \min\sum_{i=1}^{n}\left[\frac{\sin\phi_i}{\sin\alpha_i}a_i\right]^2. \tag{19}$$

For the OrthLS estimates, Equation (19) simplifies to

$$\left[\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_{11}\right] = \min\sum_{i=1}^{n} d_i^2 = \min\sum_{i=1}^{n}\left[\left(\sin\phi_i\right)a_i\right]^2 \tag{20}$$

since $\sin\frac{\pi}{2} = 1$. While it is possible to solve this equation analytically, the solutions to Equations (19) and (20) are obtained numerically. The OLS estimates are used as an initial guess for the algorithm, and since the initial guess is generally close, convergence of the solution occurs rather quickly.

4.3 SIMULATION STUDY FOR THE SIMPLE QUADRATIC MODEL

In the simple quadratic model, the ϵs are assumed to be independently and identically distributed as normal with a mean of zero and a constant variance of σ_ϵ^2. Under these conditions, the Gauss-Markov theorem holds, and the statistical properties of the OLS estimators are well-known. Specifically, the estimated coefficients are unbiased, or $E\left(\hat{\boldsymbol{\beta}}\right) = \boldsymbol{\beta}$, and the coefficients have minimal variance. It is shown that

$$\mathrm{Var}\left(\hat{\boldsymbol{\beta}}\right) = \sigma_\epsilon^2 \left(\mathbf{x}'\mathbf{x}\right)^{-1}$$

where $\left(\mathbf{x}'\mathbf{x}\right)^{-1}$ is known as the variance-covariance matrix of the β's. The variances of the estimated coefficients are located along the diagonal with the covariances appearing on the off-diagonal (Myers, 1990). To understand the statistical properties of the OrthLS and MLS estimators, a simulation was conducted to make inferences on the effects of MEs on the prediction capabilities of each method for the simple quadratic model. The prediction capabilities are again based on a variance ratio-weighted distance between any data point and the estimate curve. A representative calibration experiment is designed with $n = 6$ design points in order to fit the simple quadratic model

$$y = \beta_0 + \beta_1 x + \beta_{11} x_i^2 + \epsilon.$$

In the presence of ME, this model becomes

$$y = \beta_0 + \beta_1 \left(x + u\right) + \beta_{11} \left(x_i + u_i\right)^2 + \epsilon. \tag{21}$$

The following assumptions are made about Equation (21):

- $\beta_0 = 0$, $\beta_1 = 1$, and $\beta_{11} = 0.2$. These values are representative of real measurement systems, where the sensitivity is the dominant effect. The curvature is limited to 20 percent of F.S. effect.

- The errors u and ϵ are independent and identically distributed as normal with means of zero and constant variances of σ_u^2 and σ_ϵ^2, respectively.

- σ_u^2 and σ_ϵ^2 are proportionally related through the variance ratio, γ.

- W_i is the error-prone value of x_i. Therefore, W_i and Y_i are jointly distributed as a bivariate normal distribution

$$[W_i, Y_i] \sim N\left[\left(x_i, \beta_0 + \beta_1 x_i + \beta_{11} x_i^2\right), \Sigma\right]$$

  where $\Sigma = \text{diag}\left(\gamma\sigma_\epsilon^2, \sigma_\epsilon^2\right)$. Both W_i and Y_i are independent random variables.

The structure of the simulation is similar to the simulation executed for the simple linear model. From the n design points and corresponding responses, the three model coefficients are estimated and the MSE is calculated for each of the estimation methods. Five (5) confirmation points located at $x = -2/3, -1/3, 0, 1/3$, and $2/3$ are used to generate confirmation data to further examine the prediction capabilities of each estimated model. For a given calibration design, each method estimates a set of coefficients M times. Each set of estimated coefficients are used L times with the confirmation points and the MSE is calculated across $(M \times L)$ observations.

The simulation study considered two of the three designs that were used for the simple linear simulation and are shown in Table 14. As mentioned previously, Design #3 used in the simple linear simulation does not allow estimation of the curvature since there are only two unique levels of x. The features of each design are briefly reviewed for convenience. In Design #1, the replicates are evenly dispersed

TABLE 14: Designs Considered for the Simple Quadratic Simulation Study

Design	Design Points	Comments
Design #1	-1, -1, 0, 0, 1, 1	Replicates throughout Design Space
Design #2	-1, 0, 0, 0, 0, 1	Suspicion of Significant Lack-of-fit

among the unique levels, which provides a more appropriate metric for assessing the constant variance assumption. Design #2 assumes constant variance and simply uses the estimate of repeatability at the center (i.e. $x = 0$) to quote the repeatability throughout the design space. Table 3 shows the values of variance ratio and response uncertainty that are used as inputs to the simulation. The simulation employs a factorial combination of the variance ratio and response uncertainty.

4.3.1 SIMULATION RESULTS

Table 15 shows the mean estimates of the model coefficients for the two calibration designs and the three estimation methods. Unlike the simple linear simulation results, all three model coefficients are provided. From Table 15, the following inferences are made based on the structure of the simulation:

- The zero-intercept estimate varies between the three estimation methods depending on the level of variance ratio and response uncertainty. In the simple linear simulation, all three methods yielded the same estimate of the zero-intercept. The impact of the method on the estimates of the coefficients is greater for a quadratic model than a linear model.

- For a variance ratio of one ($\gamma = 1$), the estimates of all three model coefficients obtained from MLS and OrthLS are equivalent as expected. In the limit as $\gamma \to 0$, it is evident that the estimate of $\hat{\beta}_1$ from MLS converges towards the OLS estimate.

- For an uncertainty of 1 percent or better ($\sigma_\epsilon \leq 0.01$), the three methods yield numerically-similar estimates of the sensitivity coefficient to four significant figures.

- Larger differences between the estimates are observed when the uncertainty is larger than 5 percent ($\sigma_\epsilon > 0.05$). The magnitude of the difference varies between designs and levels of uncertainty.

- For any level of variance ratio or uncertainty,

$$\left(\hat{\beta}_1\right)_{\text{OLS}} \leq \left(\hat{\beta}_1\right)_{\text{MLS}} \leq \left(\hat{\beta}_1\right)_{\text{OrthLS}}.$$

  However, the table reveals that a similar conclusion cannot be made for the other two coefficients.

Table 16 presents the variance of the mean estimates of the model coefficients for the two designs and three methods. Based on the results, it is clear that the variances are a function of the calibration design. In Design #1, the replicated points are evenly dispersed throughout the design space, which benefits the sensitivity estimate. Therefore, a more precise (less variable) estimate is obtained. Design #2 is more centrally weighted in its replication strategy, which favors the zero-intercept estimate. Comparing the results for Design #2, it is evident that the design yields a more precise estimate of the zero-intercept. Between-method variability is small and from a practical standpoint, the variability in the MLS estimates is approximately equivalent to the variability in the OLS estimates.

TABLE 15: Mean Estimates of the Model Coefficients for the Simple Quadratic Simulation

Run No	Variance Ratio, γ	Uncertainty, σ_ϵ	Model Coefficient	Design #1 OLS	Design #1 MLS	Design #1 OrthLS	Design #2 OLS	Design #2 MLS	Design #2 OrthLS
1	1	0.2	$\hat{\beta}_0$	-0.0143	-0.0068	-0.0068	0.0038	0.0035	0.0035
			$\hat{\beta}_1$	1.0122	1.0283	1.0283	1.0149	1.0485	1.0485
			$\hat{\beta}_{11}$	0.2200	0.2140	0.2140	0.1788	0.1914	0.1914
2	0.25	0.2	$\hat{\beta}_0$	0.0143	0.0127	0.0200	-0.0163	-0.0163	-0.0167
			$\hat{\beta}_1$	0.9929	1.0006	1.0093	1.0039	1.0207	1.0433
			$\hat{\beta}_{11}$	0.1951	0.1988	0.1918	0.2040	0.2073	0.2208
3	0.0625	0.2	$\hat{\beta}_0$	0.0114	0.0095	0.0116	0.0153	0.0153	0.0146
			$\hat{\beta}_1$	0.9897	0.9918	1.0051	0.9887	0.9936	1.0284
			$\hat{\beta}_{11}$	0.1929	0.1959	0.1981	0.1709	0.1712	0.1888
4	0.01	0.2	$\hat{\beta}_0$	0.0331	0.0329	0.0378	0.0228	0.0228	0.0226
			$\hat{\beta}_1$	0.9958	0.9962	1.0112	1.0031	1.0037	1.0323
			$\hat{\beta}_{11}$	0.1653	0.1657	0.1622	0.1958	0.1958	0.2068
5	0.0001	0.2	$\hat{\beta}_0$	-0.0203	-0.0203	-0.0136	0.0049	0.0049	0.0043
			$\hat{\beta}_1$	1.0068	1.0068	1.0212	1.0003	1.0003	1.0394
			$\hat{\beta}_{11}$	0.2235	0.2235	0.2188	0.1959	0.1959	0.2132
6	1	0.1	$\hat{\beta}_0$	0.0060	0.0099	0.0099	-0.0144	-0.0145	-0.0145
			$\hat{\beta}_1$	1.0080	1.0115	1.0115	1.0032	1.0110	1.0110
			$\hat{\beta}_{11}$	0.1904	0.1857	0.1857	0.2247	0.2281	0.2281
7	0.25	0.1	$\hat{\beta}_0$	-0.0025	-0.0022	0.0004	0.0069	0.0069	0.0069
			$\hat{\beta}_1$	1.0006	1.0024	1.0040	0.9984	1.0024	1.0069
			$\hat{\beta}_{11}$	0.2073	0.2072	0.2042	0.1780	0.1787	0.1809
8	0.0625	0.1	$\hat{\beta}_0$	-0.0088	-0.0088	-0.0054	0.0082	0.0082	0.0082
			$\hat{\beta}_1$	1.0004	1.0010	1.0038	1.0029	1.0041	1.0116
			$\hat{\beta}_{11}$	0.2041	0.2041	0.2000	0.1944	0.1945	0.1977
9	0.01	0.1	$\hat{\beta}_0$	0.0013	0.0013	0.0044	0.0003	0.0003	0.0003

TABLE 15 – Continued

Run No	Variance Ratio, γ	Uncertainty, σ_ϵ	Model Coefficient	Design #1 OLS	Design #1 MLS	Design #1 OrthLS	Design #2 OLS	Design #2 MLS	Design #2 OrthLS
			$\hat{\beta}_1$	1.0068	1.0069	1.0110	0.9970	0.9972	1.0046
			$\hat{\beta}_{11}$	0.2063	0.2064	0.2031	0.2005	0.2005	0.2032
10	0.0001	0.1	$\hat{\beta}_0$	0.0076	0.0076	0.0108	0.0081	0.0081	0.0081
			$\hat{\beta}_1$	1.0082	1.0082	1.0113	0.9944	0.9944	1.0027
			$\hat{\beta}_{11}$	0.1896	0.1896	0.1858	0.1785	0.1785	0.1813
11	1	0.05	$\hat{\beta}_0$	0.0016	0.0024	0.0024	0.0033	0.0033	0.0033
			$\hat{\beta}_1$	0.9997	1.0006	1.0006	0.9959	0.9976	0.9976
			$\hat{\beta}_{11}$	0.1990	0.1982	0.1982	0.1937	0.1943	0.1943
12	0.25	0.05	$\hat{\beta}_0$	-0.0051	-0.0049	-0.0041	0.0016	0.0016	0.0016
			$\hat{\beta}_1$	1.0062	1.0067	1.0071	0.9963	0.9972	0.9981
			$\hat{\beta}_{11}$	0.2016	0.2016	0.2006	0.1970	0.1972	0.1977
13	0.0625	0.05	$\hat{\beta}_0$	0.0030	0.0029	0.0035	0.0006	0.0006	0.0006
			$\hat{\beta}_1$	0.9992	0.9994	1.0000	0.9963	0.9965	0.9982
			$\hat{\beta}_{11}$	0.1964	0.1965	0.1960	0.1952	0.1952	0.1959
14	0.01	0.05	$\hat{\beta}_0$	0.0013	0.0013	0.0020	0.0023	0.0023	0.0023
			$\hat{\beta}_1$	1.0012	1.0012	1.0020	0.9984	0.9985	1.0008
			$\hat{\beta}_{11}$	0.1997	0.1997	0.1988	0.1919	0.1919	0.1928
15	0.0001	0.05	$\hat{\beta}_0$	-0.0064	-0.0064	-0.0056	-0.0014	-0.0014	-0.0014
			$\hat{\beta}_1$	1.0008	1.0008	1.0016	1.0021	1.0021	1.0041
			$\hat{\beta}_{11}$	0.2062	0.2062	0.2054	0.2071	0.2071	0.2079
16	1	0.01	$\hat{\beta}_0$	-0.0014	-0.0013	-0.0013	-0.0003	-0.0003	-0.0003
			$\hat{\beta}_1$	1.0000	1.0001	1.0001	1.0003	1.0004	1.0004
			$\hat{\beta}_{11}$	0.2015	0.2015	0.2015	0.2008	0.2009	0.2009
17	0.25	0.01	$\hat{\beta}_0$	-0.0009	-0.0009	-0.0009	0.0011	0.0011	0.0011
			$\hat{\beta}_1$	0.9994	0.9994	0.9995	0.9993	0.9994	0.9994
			$\hat{\beta}_{11}$	0.2010	0.2010	0.2010	0.1995	0.1996	0.1996

TABLE 15 – Continued

Run No	Variance Ratio, γ	Uncertainty, σ_ϵ	Model Coefficient	Design #1 OLS	Design #1 MLS	Design #1 OrthLS	Design #2 OLS	Design #2 MLS	Design #2 OrthLS
18	0.0625	0.01	$\hat{\beta}_0$	0.0003	0.0003	0.0003	-0.0004	-0.0004	-0.0004
			$\hat{\beta}_1$	1.0008	1.0008	1.0008	0.9999	1.0000	1.0000
			$\hat{\beta}_{11}$	0.1998	0.1998	0.1998	0.2019	0.2019	0.2019
19	0.01	0.01	$\hat{\beta}_0$	-0.0008	-0.0008	-0.0008	-0.0002	-0.0002	-0.0002
			$\hat{\beta}_1$	0.9996	0.9996	0.9996	1.0009	1.0009	1.0010
			$\hat{\beta}_{11}$	0.2016	0.2016	0.2016	0.2003	0.2003	0.2003
20	0.0001	0.01	$\hat{\beta}_0$	-0.0009	-0.0009	-0.0009	0.0005	0.0005	0.0005
			$\hat{\beta}_1$	0.9985	0.9985	0.9985	0.9993	0.9993	0.9994
			$\hat{\beta}_{11}$	0.2002	0.2002	0.2002	0.1994	0.1994	0.1994
21	1	0.001	$\hat{\beta}_0$	-0.0001	-0.0001	-0.0001	-0.0001	-0.0001	-0.0001
			$\hat{\beta}_1$	1.0001	1.0001	1.0001	1.0000	1.0000	1.0000
			$\hat{\beta}_{11}$	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
22	0.25	0.001	$\hat{\beta}_0$	0.0000	0.0000	0.0000	-0.0001	-0.0001	-0.0001
			$\hat{\beta}_1$	1.0000	1.0000	1.0000	0.9999	0.9999	0.9999
			$\hat{\beta}_{11}$	0.2001	0.2001	0.2001	0.2000	0.2000	0.2000
23	0.0625	0.001	$\hat{\beta}_0$	-0.0001	-0.0001	-0.0001	0.0001	0.0001	0.0001
			$\hat{\beta}_1$	0.9999	0.9999	0.9999	1.0000	1.0000	1.0000
			$\hat{\beta}_{11}$	0.2001	0.2001	0.2001	0.1998	0.1998	0.1998
24	0.01	0.001	$\hat{\beta}_0$	-0.0001	-0.0001	-0.0001	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	1.0000	1.0000	1.0000	1.0001	1.0001	1.0001
			$\hat{\beta}_{11}$	0.2001	0.2001	0.2001	0.2001	0.2001	0.2001
25	0.0001	0.001	$\hat{\beta}_0$	0.0001	0.0001	0.0001	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
			$\hat{\beta}_{11}$	0.1998	0.1998	0.1998	0.1999	0.1999	0.1999

TABLE 16: Variance of the Mean Estimates of the Model Coefficients for the Simple Quadratic Simulation

Run No	Variance Ratio, γ	Uncertainty, σ_ϵ	Model Coefficient	Design #1 OLS	Design #1 MLS	Design #1 OrthLS	Design #2 OLS	Design #2 MLS	Design #2 OrthLS
1	1	0.2	$\hat{\beta}_0$	0.0162	0.0177	0.0177	0.0094	0.0094	0.0094
			$\hat{\beta}_1$	0.0087	0.0088	0.0088	0.0159	0.0167	0.0167
			$\hat{\beta}_{11}$	0.0247	0.0295	0.0295	0.0286	0.0325	0.0325
2	0.25	0.2	$\hat{\beta}_0$	0.0212	0.0220	0.0222	0.0095	0.0095	0.0096
			$\hat{\beta}_1$	0.0088	0.0090	0.0094	0.0204	0.0208	0.0215
			$\hat{\beta}_{11}$	0.0292	0.0309	0.0324	0.0372	0.0385	0.0442
3	0.0625	0.2	$\hat{\beta}_0$	0.0193	0.0194	0.0197	0.0097	0.0097	0.0097
			$\hat{\beta}_1$	0.0081	0.0081	0.0080	0.0207	0.0207	0.0212
			$\hat{\beta}_{11}$	0.0292	0.0295	0.0331	0.0255	0.0256	0.0318
4	0.01	0.2	$\hat{\beta}_0$	0.0211	0.0211	0.0203	0.0119	0.0119	0.0119
			$\hat{\beta}_1$	0.0089	0.0089	0.0092	0.0188	0.0188	0.0183
			$\hat{\beta}_{11}$	0.0288	0.0289	0.0297	0.0266	0.0266	0.0298
5	0.0001	0.2	$\hat{\beta}_0$	0.0173	0.0173	0.0185	0.0082	0.0082	0.0082
			$\hat{\beta}_1$	0.0103	0.0103	0.0107	0.0188	0.0188	0.0203
			$\hat{\beta}_{11}$	0.0291	0.0291	0.0342	0.0250	0.0250	0.0294
6	1	0.1	$\hat{\beta}_0$	0.0049	0.0048	0.0048	0.0024	0.0024	0.0024
			$\hat{\beta}_1$	0.0027	0.0027	0.0027	0.0046	0.0047	0.0047
			$\hat{\beta}_{11}$	0.0069	0.0071	0.0071	0.0073	0.0076	0.0076
7	0.25	0.1	$\hat{\beta}_0$	0.0044	0.0044	0.0044	0.0026	0.0026	0.0026
			$\hat{\beta}_1$	0.0022	0.0023	0.0023	0.0043	0.0043	0.0044
			$\hat{\beta}_{11}$	0.0064	0.0064	0.0064	0.0066	0.0067	0.0068
8	0.0625	0.1	$\hat{\beta}_0$	0.0039	0.0039	0.0041	0.0019	0.0019	0.0019
			$\hat{\beta}_1$	0.0024	0.0024	0.0024	0.0042	0.0042	0.0042
			$\hat{\beta}_{11}$	0.0067	0.0067	0.0070	0.0080	0.0080	0.0083
9	0.01	0.1	$\hat{\beta}_0$	0.0053	0.0053	0.0053	0.0026	0.0026	0.0026

TABLE 16 – Continued

Run No	Variance Ratio, γ	Uncertainty, σ_ϵ	Model Coefficient	Design #1 OLS	Design #1 MLS	Design #1 OrthLS	Design #2 OLS	Design #2 MLS	Design #2 OrthLS
			$\hat{\beta}_1$	0.0026	0.0026	0.0025	0.0058	0.0058	0.0059
			$\hat{\beta}_{11}$	0.0070	0.0071	0.0073	0.0074	0.0074	0.0076
10	0.0001	0.1	$\hat{\beta}_0$	0.0053	0.0053	0.0052	0.0025	0.0025	0.0025
			$\hat{\beta}_1$	0.0027	0.0027	0.0026	0.0059	0.0059	0.0059
			$\hat{\beta}_{11}$	0.0069	0.0069	0.0070	0.0094	0.0094	0.0098
11	1	0.05	$\hat{\beta}_0$	0.0013	0.0013	0.0013	0.0007	0.0007	0.0007
			$\hat{\beta}_1$	0.0006	0.0006	0.0006	0.0013	0.0013	0.0013
			$\hat{\beta}_{11}$	0.0019	0.0020	0.0020	0.0021	0.0021	0.0021
12	0.25	0.05	$\hat{\beta}_0$	0.0012	0.0012	0.0012	0.0006	0.0006	0.0006
			$\hat{\beta}_1$	0.0005	0.0005	0.0005	0.0014	0.0014	0.0014
			$\hat{\beta}_{11}$	0.0016	0.0016	0.0016	0.0023	0.0023	0.0023
13	0.0625	0.05	$\hat{\beta}_0$	0.0013	0.0013	0.0013	0.0006	0.0006	0.0006
			$\hat{\beta}_1$	0.0005	0.0005	0.0005	0.0011	0.0011	0.0011
			$\hat{\beta}_{11}$	0.0019	0.0019	0.0019	0.0014	0.0014	0.0014
14	0.01	0.05	$\hat{\beta}_0$	0.0011	0.0011	0.0011	0.0006	0.0006	0.0006
			$\hat{\beta}_1$	0.0006	0.0006	0.0006	0.0013	0.0013	0.0013
			$\hat{\beta}_{11}$	0.0015	0.0015	0.0016	0.0019	0.0019	0.0019
15	0.0001	0.05	$\hat{\beta}_0$	0.0012	0.0012	0.0011	0.0006	0.0006	0.0006
			$\hat{\beta}_1$	0.0007	0.0007	0.0006	0.0011	0.0011	0.0011
			$\hat{\beta}_{11}$	0.0019	0.0019	0.0019	0.0019	0.0019	0.0019
16	1	0.01	$\hat{\beta}_0$	0.0001	0.0001	0.0001	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001
			$\hat{\beta}_{11}$	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
17	0.25	0.01	$\hat{\beta}_0$	0.0001	0.0001	0.0001	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_{11}$	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001

TABLE 16 – Continued

Run No	Variance Ratio, γ	Uncertainty, σ_ϵ	Model Coefficient	Design #1 OLS	Design #1 MLS	Design #1 OrthLS	Design #2 OLS	Design #2 MLS	Design #2 OrthLS
18	0.0625	0.01	$\hat{\beta}_0$	0.0001	0.0001	0.0001	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001
			$\hat{\beta}_{11}$	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
19	0.01	0.01	$\hat{\beta}_0$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_{11}$	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
20	0.0001	0.01	$\hat{\beta}_0$	0.0001	0.0001	0.0001	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001
			$\hat{\beta}_{11}$	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
21	1	0.001	$\hat{\beta}_0$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_{11}$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
22	0.25	0.001	$\hat{\beta}_0$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_{11}$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
23	0.0625	0.001	$\hat{\beta}_0$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_{11}$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24	0.01	0.001	$\hat{\beta}_0$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_{11}$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0001	0.001	$\hat{\beta}_0$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_1$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
			$\hat{\beta}_{11}$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 17 shows the average MSE across all the design and confirmation points for the two designs and the three methods. Based on the values shown, the 1-σ estimated accuracy, in percent, is calculated using

$$\hat{\sigma} = \frac{\sqrt{\text{Mean Squared Error}}}{\text{Full-Scale Range}} \times 100.$$

For Runs 11 through 15, the estimated 1-σ accuracy ranges from 2.6 to 3.6 percent of F.S., which is smaller than the response error of 5 percent. While Table 17 shows the absolute MSE values, Table 18 gives the difference between OLS and either MLS or OrthLS. From this table, it is once again observed that MLS has a smaller MSE than OLS for every combination of variance ratio and response uncertainty studied. Meanwhile, OrthLS only shows improvement when the variance ratio is one. When the variance ratio is one, OrthLS is equivalent to MLS by definition. The improvement in MSE is expressed in terms of percent in Table 19. There appears to be additional benefit in using MLS for the simple quadratic model when MEs are present. However, there also appears to be a larger penalty for using OrthLS as the variance ratio approaches zero. An analysis of variance (ANOVA) is performed to test whether differences in MSE are statistically detectable. From the results of the ANOVA shown in Tables 20 and 21, it is inferred that there is no detectable difference in MSE between the three methods for this simulation. This conclusion is also valid when looking at the hypothesis tests simultaneously.

Table 22 relates the improvement in MSE to the known response uncertainty, σ_ϵ. For cases where OLS had a smaller MSE, the percent improvement is expressed as zero in the table. The results in the table show that MLS has a significant impact on the reduction in MSE for most of the cases tested during the simulation. For example, a 0.14 percent improvement in MSE is achieved on a 1 percent uncertainty measurement system using the first calibration design and a variance ratio of 0.0625.

This translates to an overall reduction of 14 percent, which is considered practically significant. Based on the results from the simple quadratic simulation, it is recommended that MLS be employed instead of OLS when the variance ratio is greater than 0.0625. Below a variance ratio of 0.0625, the practical benefit of MLS is small.

TABLE 17: Mean Squared Error for the Simple Quadratic Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1			Design #2		
			OLS	MLS	OrthLS	OLS	MLS	OrthLS
1	1	0.2	0.0113	0.0108	0.0108	0.0096	0.0093	0.0093
2	0.25	0.2	0.0160	0.0158	0.0161	0.0155	0.0153	0.0156
3	0.0625	0.2	0.0186	0.0186	0.0194	0.0198	0.0198	0.0208
4	0.01	0.2	0.0206	0.0206	0.0214	0.0165	0.0165	0.0171
5	0.0001	0.2	0.0191	0.0191	0.0203	0.0219	0.0219	0.0233
6	1	0.1	0.0028	0.0027	0.0027	0.0023	0.0023	0.0023
7	0.25	0.1	0.0036	0.0036	0.0036	0.0038	0.0038	0.0038
8	0.0625	0.1	0.0044	0.0044	0.0045	0.0048	0.0048	0.0049
9	0.01	0.1	0.0060	0.0060	0.0060	0.0045	0.0045	0.0046
10	0.0001	0.1	0.0045	0.0045	0.0046	0.0051	0.0051	0.0051
11	1	0.05	0.0007	0.0007	0.0007	0.0005	0.0005	0.0005
12	0.25	0.05	0.0010	0.0010	0.0010	0.0008	0.0008	0.0008
13	0.0625	0.05	0.0010	0.0010	0.0010	0.0011	0.0011	0.0011
14	0.01	0.05	0.0012	0.0012	0.0012	0.0015	0.0015	0.0015
15	0.0001	0.05	0.0013	0.0013	0.0013	0.0013	0.0013	0.0013
16	1	0.01	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
17	0.25	0.01	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
18	0.0625	0.01	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
19	0.01	0.01	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001
20	0.0001	0.01	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001
21	1	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
22	0.25	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
23	0.0625	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24	0.01	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0001	0.001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

TABLE 18: Improvement in Mean Squared Error over OLS for the Simple Quadratic Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1 MLS	Design #1 OrthLS	Design #2 MLS	Design #2 OrthLS
1	1	0.2	0.0004	0.0004	0.0002	0.0002
2	0.25	0.2	0.0002	0.0000	0.0001	-0.0001
3	0.0625	0.2	0.0000*	-0.0008	0.0000*	-0.0010
4	0.01	0.2	0.0000*	-0.0008	0.0000*	-0.0006
5	0.0001	0.2	0.0000*	-0.0012	0.0000*	-0.0014
6	1	0.1	0.0000*	0.0000*	0.0000*	0.0000*
7	0.25	0.1	0.0000*	0.0000	0.0000*	0.0000
8	0.0625	0.1	0.0000*	0.0000	0.0000*	0.0000
9	0.01	0.1	0.0000*	-0.0001	0.0000*	0.0000
10	0.0001	0.1	0.0000*	-0.0001	0.0000*	0.0000
11	1	0.05	0.0000*	0.0000*	0.0000*	0.0000*
12	0.25	0.05	0.0000*	0.0000	0.0000*	0.0000
13	0.0625	0.05	0.0000*	0.0000	0.0000*	0.0000
14	0.01	0.05	0.0000*	0.0000	0.0000*	0.0000
15	0.0001	0.05	0.0000*	0.0000	0.0000*	0.0000
16	1	0.01	0.0000*	0.0000*	0.0000*	0.0000*
17	0.25	0.01	0.0000*	0.0000	0.0000*	0.0000
18	0.0625	0.01	0.0000*	0.0000	0.0000*	0.0000
19	0.01	0.01	0.0000*	0.0000	0.0000*	0.0000
20	0.0001	0.01	0.0000*	0.0000	0.0000*	0.0000
21	1	0.001	0.0000*	0.0000*	0.0000*	0.0000*
22	0.25	0.001	0.0000*	0.0000	0.0000*	0.0000
23	0.0625	0.001	0.0000*	0.0000	0.0000*	0.0000
24	0.01	0.001	0.0000*	0.0000	0.0000*	0.0000
25	0.0001	0.001	0.0000*	0.0000	0.0000*	0.0000

Note: Asterisk represents an improvement of less than 0.00005

TABLE 19: Percent Improvement in Mean Squared Error over OLS for the Simple Quadratic Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1		Design #2	
			MLS	OrthLS	MLS	OrthLS
1	1	0.2	3.95%	3.95%	2.59%	2.59%
2	0.25	0.2	1.35%	-0.30%	0.66%	-0.83%
3	0.0625	0.2	0.09%	-4.44%	0.06%	-5.22%
4	0.01	0.2	0.00%*	-3.73%	0.00%*	-3.34%
5	0.0001	0.2	0.00%*	-6.09%	0.00%*	-6.17%
6	1	0.1	0.97%	0.97%	0.67%	0.67%
7	0.25	0.1	0.20%	-0.14%	0.15%	-0.05%
8	0.0625	0.1	0.03%	-1.10%	0.01%	-0.63%
9	0.01	0.1	0.00%*	-1.47%	0.00%*	-0.75%
10	0.0001	0.1	0.00%*	-1.47%	0.00%*	-0.78%
11	1	0.05	0.31%	0.31%	0.14%	0.14%
12	0.25	0.05	0.08%	-0.05%	0.04%	-0.01%
13	0.0625	0.05	0.00%*	-0.17%	0.00%*	-0.15%
14	0.01	0.05	0.00%*	-0.33%	0.00%*	-0.22%
15	0.0001	0.05	0.00%*	-0.30%	0.00%*	-0.21%
16	1	0.01	0.02%	0.02%	0.00%*	0.00%*
17	0.25	0.01	0.00%*	0.00%	0.00%*	0.00%
18	0.0625	0.01	0.00%*	-0.01%	0.00%*	0.00%
19	0.01	0.01	0.00%*	-0.01%	0.00%*	-0.01%
20	0.0001	0.01	0.00%*	-0.01%	0.00%*	-0.01%
21	1	0.001	0.00%*	0.00%*	0.00%*	0.00%*
22	0.25	0.001	0.00%*	0.00%	0.00%*	0.00%
23	0.0625	0.001	0.00%*	0.00%	0.00%*	0.00%
24	0.01	0.001	0.00%*	0.00%	0.00%*	0.00%
25	0.0001	0.001	0.00%*	0.00%	0.00%*	0.00%

Note: Asterisk represents an improvement of less than 0.005%

TABLE 20: Analysis of Variance of the Mean Squared Error for the Simple Quadratic Simulation – Design #1

Source	DF	SS	MS	F	P
Method (OLS, MLS, OrthLS)	2	0.0000014	0.0000007	0.02	0.985
Error	447	0.0209744	0.0000469		
Total	449	0.0209759			

TABLE 21: Analysis of Variance of the Mean Squared Error for the Simple Quadratic Simulation – Design #2

Source	DF	SS	MS	F	P
Method (OLS, MLS, OrthLS)	2	0.0000016	0.0000008	0.01	0.990
Error	447	0.0339253	0.0000759		
Total	449	0.0339269			

TABLE 22: Percent Improvement in Mean Squared Error Relative to Response Uncertainty for the Simple Quadratic Simulation

Run No	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1 MLS	Design #1 OrthLS	Design #2 MLS	Design #2 OrthLS
1	1	0.2	10.56%	10.56%	7.87%	7.87%
2	0.25	0.2	7.35%	0.00%	5.05%	0.00%
3	0.0625	0.2	2.09%	0.00%	1.73%	0.00%
4	0.01	0.2	0.41%	0.00%	0.23%	0.00%
5	0.0001	0.2	0.00%*	0.00%	0.00%*	0.00%
6	1	0.1	5.19%	5.19%	3.92%	3.92%
7	0.25	0.1	2.71%	0.00%	2.43%	0.00%
8	0.0625	0.1	1.05%	0.00%	0.82%	0.00%
9	0.01	0.1	0.25%	0.00%	0.13%	0.00%
10	0.0001	0.1	0.00%*	0.00%	0.00%*	0.00%
11	1	0.05	2.96%	2.96%	1.72%	1.72%
12	0.25	0.05	1.75%	0.00%	1.09%	0.00%
13	0.0625	0.05	0.47%	0.00%	0.40%	0.00%
14	0.01	0.05	0.10%	0.00%	0.08%	0.00%
15	0.0001	0.05	0.00%*	0.00%	0.00%*	0.00%
16	1	0.01	0.77%	0.77%	0.45%	0.45%
17	0.25	0.01	0.29%	0.00%	0.21%	0.00%
18	0.0625	0.01	0.14%	0.00%	0.07%	0.00%
19	0.01	0.01	0.02%	0.00%	0.02%	0.00%
20	0.0001	0.01	0.00%*	0.00%	0.00%*	0.00%
21	1	0.001	0.05%	0.05%	0.04%	0.04%
22	0.25	0.001	0.03%	0.00%	0.02%	0.00%
23	0.0625	0.001	0.00%*	0.00%	0.00%*	0.00%
24	0.01	0.001	0.00%*	0.00%	0.00%*	0.00%
25	0.0001	0.001	0.00%*	0.00%	0.00%*	0.00%

Note: Asterisk represents an improvement of less than 0.005%

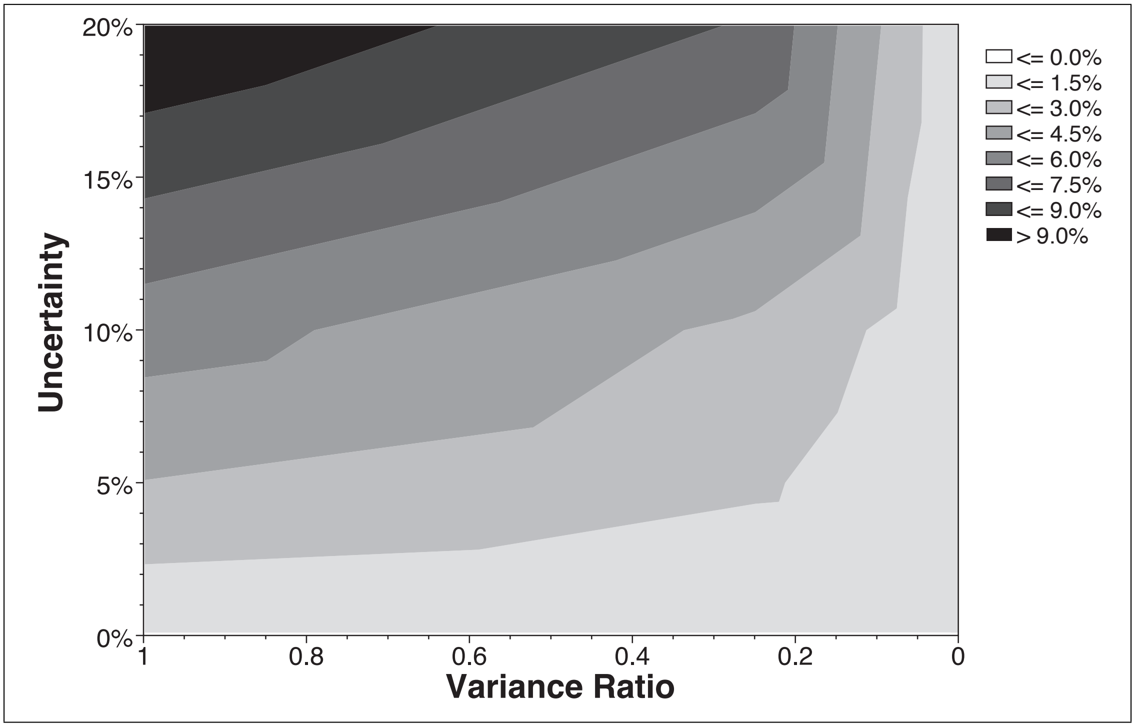

FIG. 15: Percent Improvement in Mean Squared Error of MLS for the Simple Quadratic Simulation – Design #1

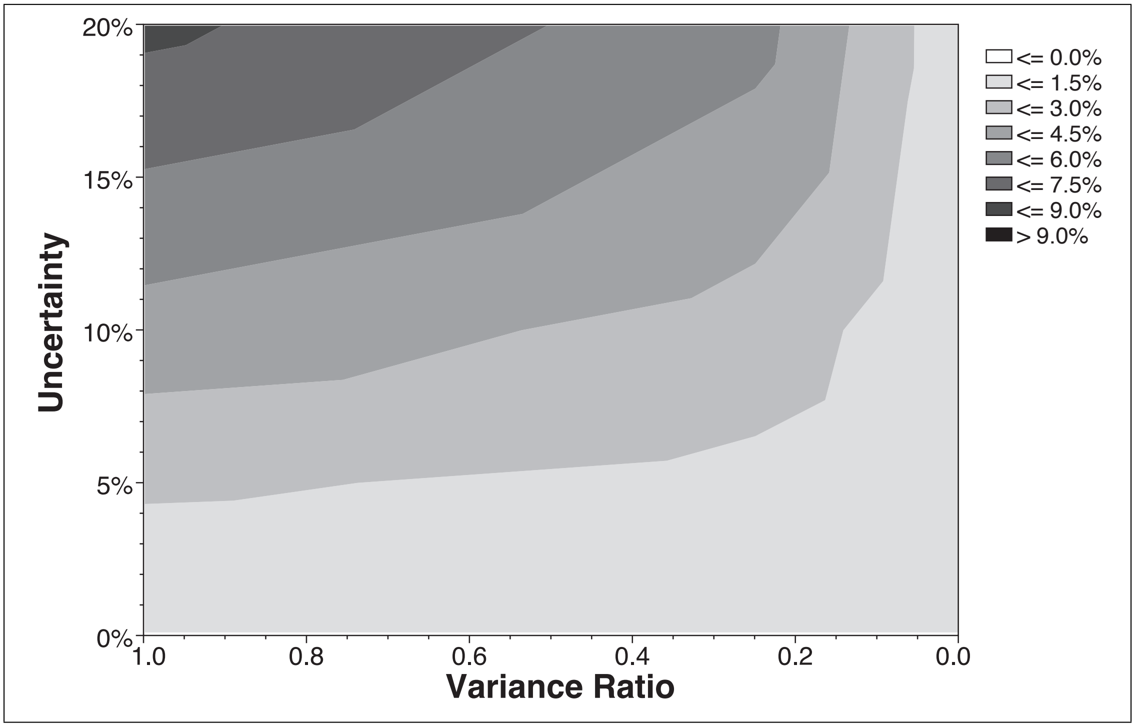

FIG. 16: Percent Improvement in Mean Squared Error of MLS for the Simple Quadratic Simulation – Design #2

4.4 APPLICATION OF THE MULTI-DIMENSIONAL, HIGHER-ORDER RESPONSE SURFACE MODEL

Recall that an estimated model is an approximation of the true, unknown model and is highlighted frequently throughout this research. The models discussed to this point are examples of the simplest possible forms. In the context of measurement systems, the true behavior is often not characterized by a simple linear or quadratic model. This is primarily due to the fact that more than one factor often affects the response of the system. Unless prior knowledge exists to suggest that one of these simple models are sufficient, a more robust, flexible model is recommended. In calibration applications at NASA LaRC, the second-order Taylor-series model is used for modeling the functional relationship of a complex measurement system. The appeal of the Taylor-series model is that it is flexible, the model coefficients are estimated through standard regression techniques, and the model is easily expanded to include any number of factors of interest. Furthermore, experiments are strategically designed to ensure certain favorable properties of the estimated model. Therefore, the second-order Taylor-series model is the final model considered within this research.

At NASA LaRC, most force-balances utilize the second-order Taylor-series model to approximate the relationship between the applied loads and the measured output of the strain gauges. For a standard six-component force-balance, the model is

$$y_i = \beta_0 + \sum_{a=1}^{6} \beta_a x_{a_i} + \sum_{a=1}^{5} \sum_{b=a+1}^{6} \beta_{ab} x_{a_i} x_{b_i} + \sum_{a=1}^{6} \beta_{aa} x_{a_i}^2 + \epsilon_i.$$

This model contains 6 linear, 15 two-factor interaction, and 6 pure quadratic terms. Each of the terms in the model represents some physical or electrical characteristic of the force-balance. For example, the first-order terms are attributed to machining or gauging errors. The two-factor interactions are typically associated with the magnitude of the deflections present while the force-balance is under load (Guarino, 1964).

The NTF-113A/B/C are a family of single-piece force-balances used for testing in the NASA LaRC's National Transonic Facility (NTF). With a full-scale normal force capacity of 6500 lbf., the NTF-113 family of force-balances are capable of measuring the six aerodynamic forces and moments simultaneously. These force-balances are gauged in the standard force configuration, which only resolves axial force and rolling moment directly (AIAA, 2003). Two sets of normal force and side force gauges, one located in the forward cage section and the other located in the aft cage section, are used to compute the remaining four forces and moments. The typical accuracies for each component for this family of force-balances is on the order of 0.1 to 0.2 percent of full-scale, with axial force typically having the largest amount of uncertainty.

The NTF-113 family of force-balances are calibrated using manual stand systems. Gravity-based loads, or deadweights, are applied to the force-balance through an intricate system of pulleys and cables. These deadweights are not known without error but are traceable to a NIST standard. Since the applied loads contain some uncertainty and because no force-balance standard exists, there are several sources of ME that are introduced into the calibration. Within this research, it is assumed that the MEs are equal across all the factors that contain MEs. However, it is possible in practice for the MEs to be unequal. The effect of the MEs on these complex model forms is studied throughout and the results are discussed shortly.

4.5 ESTIMATION METHODS FOR THE MULTI-DIMENSIONAL, HIGHER-ORDER RESPONSE SURFACE MODEL

Because the OLS estimator derived earlier for the simple quadratic model is also valid for multi-dimensional response surface models, it is not discussed here. However, the OrthLS and MLS require small modifications to the derivations in order to apply them to these more complex models.

4.5.1 ORTHOGONAL AND MODIFIED LEAST SQUARES

Recall that in the derivation of the OrthLS and MLS methods that the Law of Sines was used to define the vector d that was to be minimized. The Law of Sines required knowledge of the angles ϕ and α, which were given by

$$\phi = \frac{\pi}{2} - \tan^{-1}\left(\frac{dy\left(x_1\right)}{dx}\right)$$

and

$$\alpha = \frac{\pi}{2} + (1-\gamma)\tan^{-1}\left(\frac{dy\left(x_1\right)}{dx}\right).$$

Both ϕ and α contain derivatives that need to be evaluated. For the simple linear model, these derivatives were constants. Conversely, the derivatives for the simple quadratic model were a function of the location in the design space. In both cases, however, the geometry was relatively simple. The derivative can be interpreted as the rate-of-change in two-dimensional space. The corollary to this in multi-dimensional space is the gradient, which provides both the magnitude and direction of the rate-of-change of a surface. If $f(x_1, \ldots, x_k)$ is the response surface function, then the gradient is defined as

$$\nabla f = \left[\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_k}\right].$$

The magnitude of the gradient

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2 + \ldots + \left(\frac{\partial f}{\partial x_k}\right)^2}$$

replaces the derivates in ϕ and α to give

$$\phi = \frac{\pi}{2} - \tan^{-1}\left(\sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2 + \ldots + \left(\frac{\partial f}{\partial x_k}\right)^2}\right)$$

and

$$\alpha = \frac{\pi}{2} + (1-\gamma)\tan^{-1}\left(\sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2 + \ldots + \left(\frac{\partial f}{\partial x_k}\right)^2}\right).$$

The rest of the procedure for OrthLS and MLS estimation remains the same as previously derived.

4.6 SIMULATION STUDY FOR THE SIX-COMPONENT FORCE-BALANCE

For the six-component force-balance simulation, only one experimental design is considered. The design points in Table 23 are based on a spherical central composite design (CCD) (Myers et al., 2009). This design is part of a larger class of second-order designs which are sufficient in estimating all the coefficients in the regression model and are typically more efficient to execute than a three-level factorial design. At NASA LaRC, a variant of the CCD is used for force-balance calibrations on the Single-Vector System (Parker et al., 2001). A full-factorial, spherical CCD in 6 factors contains 77 unique design points, but the simulation employs a half-fraction of the two-level, six-factor factorial design. With a half-fraction of the factorial design, the CCD contains 45 unique points. In practice, fractional factorial designs are used as a part of screening experiments or in cases where experimental resources are limited. However, there are some consequences to fractionating a design, which are typically discussed in terms of design resolution. When estimating the coefficients in the second-order Taylor-series model, the fraction must be resolution V or higher. The half-fraction of the two-level, six-factor factorial design is a resolution VI, which means that there is no confounding between any of the terms in the model. Designs with a resolution III or IV have confounding between linear and two-factor interaction terms (Montgomery, 2009).

TABLE 23: Spherical Central Composite Design for the Six-Component Force-Balance Simulation Study

Design Point	x_1	x_2	x_3	x_4	x_5	x_6
1	-0.6390	-0.6390	-0.6390	-0.6390	-0.6390	-0.6390
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	-0.6390	-0.6390	0.6390	0.6390	0.6390	0.6390
4	0.0000	-1.0000	0.0000	0.0000	0.0000	0.0000
5	0.6390	0.6390	0.6390	-0.6390	-0.6390	0.6390
6	-0.6390	0.6390	-0.6390	0.6390	0.6390	0.6390
7	0.6390	0.6390	0.6390	0.6390	0.6390	0.6390
8	0.6390	-0.6390	-0.6390	0.6390	-0.6390	-0.6390
9	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
10	-0.6390	-0.6390	0.6390	0.6390	-0.6390	-0.6390
11	0.6390	-0.6390	0.6390	-0.6390	-0.6390	-0.6390
12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

TABLE 23 – Continued

Design Point	x_1	x_2	x_3	x_4	x_5	x_6
13	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.6390	0.6390	-0.6390	-0.6390	0.6390	0.6390
15	0.6390	-0.6390	-0.6390	-0.6390	-0.6390	0.6390
16	0.6390	-0.6390	0.6390	0.6390	-0.6390	0.6390
17	0.0000	0.0000	0.0000	0.0000	0.0000	-1.0000
18	-0.6390	0.6390	0.6390	0.6390	0.6390	-0.6390
19	-1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
21	-0.6390	-0.6390	-0.6390	0.6390	-0.6390	0.6390
22	0.6390	0.6390	-0.6390	-0.6390	-0.6390	-0.6390
23	0.6390	0.6390	0.6390	-0.6390	0.6390	-0.6390
24	-0.6390	0.6390	0.6390	0.6390	-0.6390	0.6390
25	-0.6390	0.6390	-0.6390	0.6390	-0.6390	-0.6390
26	0.0000	0.0000	0.0000	-1.0000	0.0000	0.0000
27	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
28	-0.6390	0.6390	0.6390	-0.6390	-0.6390	-0.6390
29	0.6390	-0.6390	-0.6390	-0.6390	0.6390	-0.6390
30	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
31	-0.6390	-0.6390	-0.6390	-0.6390	0.6390	0.6390
32	0.6390	0.6390	-0.6390	0.6390	0.6390	-0.6390
33	0.6390	-0.6390	0.6390	0.6390	0.6390	-0.6390
34	0.6390	0.6390	-0.6390	0.6390	-0.6390	0.6390
35	0.6390	-0.6390	0.6390	-0.6390	0.6390	0.6390
36	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
37	-0.6390	-0.6390	0.6390	0.6390	0.6390	-0.6390
38	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
39	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
40	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
41	0.6390	-0.6390	-0.6390	0.6390	0.6390	0.6390
42	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
43	0.0000	0.0000	0.0000	0.0000	-1.0000	0.0000
44	-0.6390	0.6390	0.6390	-0.6390	0.6390	0.6390
45	0.6390	0.6390	0.6390	0.6390	-0.6390	-0.6390
46	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
47	-0.6390	0.6390	-0.6390	-0.6390	0.6390	-0.6390
48	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
49	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
50	-0.6390	-0.6390	0.6390	-0.6390	-0.6390	0.6390
51	-0.6390	-0.6390	-0.6390	0.6390	0.6390	-0.6390
52	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
53	-0.6390	0.6390	-0.6390	-0.6390	-0.6390	0.6390
54	0.0000	0.0000	-1.0000	0.0000	0.0000	0.0000

The CCD is used $M = 100$ times to simulate 100 calibrations of the force-balance. Each calibration produces a new set of estimated coefficients for each method and from these coefficients, the MSE based on the design points alone is calculated. For a given set of estimated coefficients, each set are applied $L = 1000$ times to the confirmation points shown in Table 24. The 34 confirmation points are based on a computer algorithm that distributes the points evenly throughout the six-dimensional design space. These points are used to generate confirmation data to further examine the prediction capabilities of each estimated model and calculate an overall MSE. In total, for one given set of coefficients, $34 \times 1000 = 34000$ confirmation points are applied. This simulation can be interpreted practically as the number of data points collected on a force-balance over the period of one calibration cycle.

TABLE 24: Confirmation Points for the Six-Component Force-Balance Simulation Study

Confirmation Point	x_1	x_2	x_3	x_4	x_5	x_6
1	-0.1380	-0.6000	-0.4140	0.2640	-0.1380	-0.6000
2	-0.6000	-0.6000	-0.6000	0.6000	0.6000	0.3660
3	-0.6000	-0.6000	0.3720	-0.6000	-0.6000	-0.6000
4	-0.1260	0.6000	-0.0960	-0.6000	-0.2280	-0.2700
5	-0.1260	0.6000	-0.0960	-0.6000	-0.2280	-0.2700
6	-0.6000	-0.6000	0.6000	0.6000	0.6000	-0.6000
7	-0.6000	0.6000	0.6000	-0.6000	-0.6000	0.6000
8	-0.5280	0.6000	-0.3480	0.2880	-0.0060	0.6000
9	0.6000	-0.6000	-0.6000	-0.6000	-0.2580	0.6000
10	-0.6000	0.6000	-0.6000	0.6000	-0.6000	-0.6000
11	0.6000	0.2280	-0.5520	0.6000	0.6000	0.6000
12	-0.5400	-0.4080	0.0720	-0.4740	0.4620	0.6000
13	-0.4500	-0.1200	-0.6000	-0.2880	-0.6000	0.1680
14	-0.6000	-0.1240	0.1500	0.6000	-0.2640	-0.0480
15	-0.1860	0.1740	0.6000	0.0000	-0.3060	-0.6000
16	-0.6000	-0.1240	0.1500	0.6000	-0.2640	-0.0480
17	0.5040	-0.5580	-0.0942	-0.0780	0.6000	-0.0840
18	0.6000	0.6000	0.6000	-0.6000	0.6000	-0.6000
19	0.6000	-0.1200	-0.6000	-0.6000	-0.6000	-0.6000
20	0.6000	0.5280	-0.6000	0.2340	0.0809	-0.2160
21	0.6000	0.6000	0.1620	0.6000	-0.2270	-0.6000

TABLE 24 – Continued

Confirmation Point	x_1	x_2	x_3	x_4	x_5	x_6
22	0.0308	0.2520	-0.0715	0.6000	0.6000	-0.5100
23	0.1800	-0.2700	0.6000	-0.6000	0.1020	0.0360
24	-0.1260	-0.6000	0.6000	0.2390	-0.6000	0.6000
25	0.6000	0.2160	0.1780	-0.2280	-0.6000	0.4560
26	0.1800	-0.2700	0.6000	-0.6000	0.1020	0.0360
27	0.2880	0.5400	-0.6000	-0.6000	0.6000	0.6000
28	0.6000	-0.6000	-0.3240	0.0420	-0.6000	-0.0180
29	0.5040	-0.5580	-0.0942	-0.0780	0.6000	-0.0840
30	0.6000	-0.6000	0.6000	0.6000	-0.6000	-0.5340
31	0.3660	-0.1800	-0.6000	0.6000	-0.6000	0.6000
32	-0.1860	0.1740	0.6000	0.0000	-0.3060	-0.6000
33	-0.5760	0.6000	0.5820	-0.0660	0.6000	0.0720
34	0.2400	0.6000	0.6000	0.6000	-0.6000	0.2400
35	-0.6000	0.5400	-0.6000	-0.4660	0.1140	0.2500
36	-0.6000	0.0420	-0.6000	-0.6000	0.6000	-0.6000
37	0.6000	0.6000	0.1260	-0.0900	0.3360	0.3410
38	0.6000	-0.0360	0.6000	0.6000	0.4020	0.6000

If the assumed form of the model is

$$y_i = \beta_0 + \sum_{a=1}^{6} \beta_a x_{a_i} + \sum_{a=1}^{5} \sum_{b=a+1}^{6} \beta_{ab} x_{a_i} x_{b_i} + \sum_{a=1}^{6} \beta_{aa} x_{a_i}^2 + \epsilon_i,$$

then in the presence of ME, it becomes

$$\begin{aligned} y_i = \beta_0 + \sum_{a=1}^{6} \beta_a \left(x_{a_i} + u_{a_i}\right) \quad &+ \sum_{a=1}^{5} \sum_{b=a+1}^{6} \beta_{ab} \left(x_{a_i} + u_{a_i}\right)\left(x_{b_i} + u_{b_i}\right) \\ &+ \sum_{a=1}^{6} \beta_{aa} \left(x_{a_i} + u_{a_i}\right)^2 + \epsilon_i. \end{aligned} \tag{22}$$

The following assumptions are made about Equation (22)

- The βs are given in Table 25, which are based on an actual set of coefficients for the axial force component of the NTF-113C force-balance.

- $u_1 = u_2 = u_3 = u_4 = u_5 = u_6 = u$. The six MEs are assumed to be equal and are represented by u. The errors u and ϵ are independently and identically distributed as normal with means of zero and constant variances of σ_u^2 and σ_ϵ^2, respectively.

- σ_u^2 and σ_ϵ^2 are proportionally related through the variance ratio, γ.
- $(W_{1_i}, \ldots, W_{6_i})$ are the error-prone values of $(x_{1_i}, \ldots, x_{x_i})$. Therefore, $(W_{1_i}, \ldots, W_{6_i})$ and Y_i are jointly distributed as a multivariate normal distribution

$$[W_{1_i}, \ldots, W_{6_i}, Y_i] \\ \sim N\left[\left(x_{1_i}, \ldots, x_{6_i}, \beta_0 + \sum_{a=1}^{6} \beta_a x_{a_i} + \sum_{a=1}^{5} \sum_{b=a+1}^{6} \beta_{ab} x_{a_i} x_{b_i} + \sum_{a=1}^{6} \beta_{aa} x_{a_i}^2\right), \Sigma\right]$$

where $\Sigma = \mathrm{diag}\,(\gamma\sigma_\epsilon^2, \gamma\sigma_\epsilon^2, \gamma\sigma_\epsilon^2, \gamma\sigma_\epsilon^2, \gamma\sigma_\epsilon^2, \gamma\sigma_\epsilon^2, \sigma_\epsilon^2)$. Both $(W_{1_i}, \ldots, W_{6_i})$ and Y_i are independent random variables.

TABLE 25: Model Coefficients for the Six-Component Force-Balance Simulation Study

First-Order Terms		Interaction Terms		Quadratic Terms	
β_0	0.0000	β_{12}	0.0321	β_{11}	-0.1015
β_1	0.1334	β_{13}	0.0934	β_{22}	0.0007
β_2	1.0000	β_{14}	0.0014	β_{33}	0.0387
β_3	-0.0095	β_{15}	-0.0027	β_{44}	0.0624
β_4	0.0257	β_{16}	0.0023	β_{55}	-0.0025
β_5	-0.0029	β_{23}	-0.0025	β_{66}	-0.0299
β_6	0.0006	β_{24}	0.0000		
		β_{25}	-0.0003		
		β_{26}	0.0009		
		β_{34}	0.0000		
		β_{35}	-0.0013		
		β_{36}	-0.0001		
		β_{45}	-0.0683		
		β_{46}	-0.0105		
		β_{56}	0.0306		

4.6.1 SIMULATION RESULTS

Table 26 shows the mean estimates of the model coefficients for two combinations of variance ratio and response uncertainty: $\gamma = 0.25$, $\sigma_\epsilon = 0.1$ and $\gamma = 0.0625$, $\sigma_\epsilon = 0.001$. In the first case, larger differences in the estimates are observed in the linear and quadratic terms. For example, the primary sensitivity, $\hat{\beta}_2$, ranges from 0.9920 for OLS to 1.0284 for OrthLS. This corresponds to a 3.6 percent difference over the F.S. range. Smaller effects, such as $\hat{\beta}_{11}$, vary as much as 2.5 percent of F.S. It is also interesting to note the trends in the estimates for the first case are not consistent. Simply put,

$$\hat{\beta}_{i_{\mathrm{OLS}}} \leq \hat{\beta}_{i_{\mathrm{MLS}}} \leq \hat{\beta}_{i_{\mathrm{OrthLS}}}$$

does not hold for all 28 estimates. The second case of $\gamma = 0.0625$ and $\sigma_\epsilon = 0.001$ shown in the table is representative of a NTF-113 calibration. Unlike the first case, these estimates are numerically equivalent to four significant figures across the three methods. Table 27 shows the variance of the coefficient estimates for the three methods. Between the methods, the variances are similar for a given coefficient. Furthermore, variances within like coefficients are also similar. For example, the variance of the linear coefficients is approximately 0.0006. These variances are again attributed to the design rather than the estimation method; thus, the variability in the MLS estimates are equal to the OLS estimates.

TABLE 26: Mean Estimates of the Model Coefficients for the Six-Component Force-Balance Simulation

Model Term	$\gamma = 0.25$, $\sigma_\epsilon = 0.1$ OLS	MLS	OrthLS	$\gamma = 0.0625$, $\sigma_\epsilon = 0.001$ OLS	MLS	OrthLS
$\hat{\beta}_0$	0.0069	0.0071	0.0071	0.0000	0.0000	0.0000
$\hat{\beta}_1$	0.1343	0.1364	0.1388	1.0000	1.0000	1.0000
$\hat{\beta}_2$	0.9920	1.0087	1.0284	-0.0801	-0.0801	-0.0801
$\hat{\beta}_3$	-0.0020	-0.0019	-0.0018	0.0900	0.0900	0.0900
$\hat{\beta}_4$	0.0144	0.0148	0.0152	-0.0300	-0.0300	-0.0300
$\hat{\beta}_5$	-0.0104	-0.0108	-0.0112	-0.0500	-0.0500	-0.0500
$\hat{\beta}_6$	-0.0037	-0.0038	-0.0038	0.3500	0.3500	0.3500
$\hat{\beta}_{12}$	0.0364	0.0380	0.0397	0.0500	0.0500	0.0500
$\hat{\beta}_{13}$	0.0918	0.0926	0.0926	0.0000	0.0000	0.0000
$\hat{\beta}_{14}$	0.0203	0.0212	0.0221	-0.0100	-0.0100	-0.0100
$\hat{\beta}_{15}$	0.0075	0.0084	0.0095	-0.0201	-0.0201	-0.0201
$\hat{\beta}_{16}$	0.0052	0.0043	0.0029	0.0699	0.0699	0.0699
$\hat{\beta}_{23}$	-0.0029	-0.0028	-0.0023	-0.0500	-0.0500	-0.0501
$\hat{\beta}_{24}$	0.0190	0.0184	0.0180	-0.0099	-0.0099	-0.0099
$\hat{\beta}_{25}$	0.0007	0.0012	0.0020	0.0000	0.0000	0.0000
$\hat{\beta}_{26}$	0.0020	0.0021	0.0023	0.0300	0.0300	0.0300
$\hat{\beta}_{34}$	0.0009	0.0003	0.0000	0.0000	0.0000	0.0000
$\hat{\beta}_{35}$	0.0027	0.0027	0.0029	-0.0199	-0.0199	-0.0199
$\hat{\beta}_{36}$	0.0110	0.0105	0.0099	-0.1000	-0.1000	-0.1000
$\hat{\beta}_{45}$	-0.0671	-0.0685	-0.0697	-0.0300	-0.0300	-0.0300
$\hat{\beta}_{46}$	-0.0119	-0.0126	-0.0130	-0.4500	-0.4500	-0.4499
$\hat{\beta}_{56}$	0.0399	0.0399	0.0401	0.0200	0.0200	0.0200
$\hat{\beta}_{11}$	-0.1151	-0.1265	-0.1401	0.1200	0.1200	0.1199
$\hat{\beta}_{22}$	-0.0003	0.0015	0.0041	-0.0100	-0.0100	-0.0100
$\hat{\beta}_{33}$	0.0432	0.0473	0.0512	0.0399	0.0399	0.0400
$\hat{\beta}_{44}$	0.0709	0.0785	0.0884	-0.0498	-0.0498	-0.0498
$\hat{\beta}_{55}$	-0.0141	-0.0143	-0.0151	0.0100	0.0100	0.0100
$\hat{\beta}_{66}$	-0.0357	-0.0382	-0.0404	-0.0401	-0.0401	-0.0401

TABLE 27: Variance of the Mean Estimates of the Model Coefficients for the Six-Component Force-Balance Simulation

Model Term	$\gamma = 0.25$, $\sigma_\epsilon = 0.1$			$\gamma = 0.0625$, $\sigma_\epsilon = 0.001$		
	OLS	MLS	OrthLS	OLS	MLS	OrthLS
$\hat{\beta}_0$	0.0006	0.0006	0.0006	0.0000	0.0000	0.0000
$\hat{\beta}_1$	0.0006	0.0007	0.0007	0.0000	0.0000	0.0000
$\hat{\beta}_2$	0.0006	0.0006	0.0006	0.0000	0.0000	0.0000
$\hat{\beta}_3$	0.0005	0.0006	0.0006	0.0000	0.0000	0.0000
$\hat{\beta}_4$	0.0006	0.0006	0.0006	0.0000	0.0000	0.0000
$\hat{\beta}_5$	0.0005	0.0005	0.0005	0.0000	0.0000	0.0000
$\hat{\beta}_6$	0.0007	0.0007	0.0007	0.0000	0.0000	0.0000
$\hat{\beta}_{12}$	0.0015	0.0015	0.0016	0.0000	0.0000	0.0000
$\hat{\beta}_{13}$	0.0018	0.0018	0.0019	0.0000	0.0000	0.0000
$\hat{\beta}_{14}$	0.0015	0.0015	0.0016	0.0000	0.0000	0.0000
$\hat{\beta}_{15}$	0.0020	0.0020	0.0020	0.0000	0.0000	0.0000
$\hat{\beta}_{16}$	0.0019	0.0020	0.0020	0.0000	0.0000	0.0000
$\hat{\beta}_{23}$	0.0017	0.0017	0.0017	0.0000	0.0000	0.0000
$\hat{\beta}_{24}$	0.0017	0.0017	0.0018	0.0000	0.0000	0.0000
$\hat{\beta}_{25}$	0.0019	0.0019	0.0020	0.0000	0.0000	0.0000
$\hat{\beta}_{26}$	0.0018	0.0018	0.0018	0.0000	0.0000	0.0000
$\hat{\beta}_{34}$	0.0013	0.0013	0.0013	0.0000	0.0000	0.0000
$\hat{\beta}_{35}$	0.0019	0.0019	0.0020	0.0000	0.0000	0.0000
$\hat{\beta}_{36}$	0.0018	0.0019	0.0019	0.0000	0.0000	0.0000
$\hat{\beta}_{45}$	0.0019	0.0019	0.0019	0.0000	0.0000	0.0000
$\hat{\beta}_{46}$	0.0019	0.0019	0.0019	0.0000	0.0000	0.0000
$\hat{\beta}_{56}$	0.0016	0.0016	0.0016	0.0000	0.0000	0.0000
$\hat{\beta}_{11}$	0.0049	0.0050	0.0052	0.0000	0.0000	0.0000
$\hat{\beta}_{22}$	0.0051	0.0054	0.0063	0.0000	0.0000	0.0000
$\hat{\beta}_{33}$	0.0043	0.0045	0.0048	0.0000	0.0000	0.0000
$\hat{\beta}_{44}$	0.0035	0.0036	0.0038	0.0000	0.0000	0.0000
$\hat{\beta}_{55}$	0.0039	0.0041	0.0042	0.0000	0.0000	0.0000
$\hat{\beta}_{66}$	0.0049	0.0052	0.0055	0.0000	0.0000	0.0000

Table 28 shows the average MSE across all the design and confirmation points for each of the three methods. The case of $\gamma = 0.0625$ and $\sigma_\epsilon = 0.001$ is closest to the actual calibration of a NTF-113 force-balance. Based on the results from this case, the quote accuracy is

$$\hat{\sigma} = \frac{\sqrt{\text{MSE}}}{\text{Full-Scale Range}} = \frac{\sqrt{4.5190 \times 10^{-7}}}{1} = 0.0007$$

or 0.07 percent of F.S. Table 29 shows the difference in MSE between OLS and either MLS or OrthLS. While the MSE for MLS is smaller in all the cases studied, the absolute difference is once again smaller than 5.0×10^{-5}. In terms of percent difference, as shown in Table 30, there are more noticeable improvements in MSE for variance ratios greater than 0.25 and uncertainties larger than 0.1. From the ANOVA in Table 31, there still remains strong evidence ($p > 0.05$) that the three methods do not provide a detectable difference in MSE.

Lastly, Table 32 reiterates the practical benefits of the improvement in MSE attained by the method of MLS. For instances where the OLS had a smaller MSE, the percent improvement is expressed as zero. The improvement in MSE by using MLS estimation is once again observed in the results. For example, 0.2 percent improvement in MSE is achieved on a measurement system with a 0.1 percent uncertainty in the response with a variance ratio of 0.025. This is a substantial difference over the improvement in the same combination of uncertainty and variance ratio in the simple linear study, where the percent improvement was 0.02 percent. Therefore, in complex model forms, such as the Taylor-series model, the impact of MLS is much greater. It is once again concluded that the largest benefit of MLS is in situations where the variance ratio is greater than 0.0625. However, the data do suggest that the practical benefit of MLS at lower variance ratios may become more pronounced in higher-order models.

TABLE 28: Mean Squared Error over OLS for the Six-Component Force-Balance Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	OLS	MLS	OrthLS
1	1	0.2	0.0096	0.0090	0.0090
2	0.25	0.2	0.0136	0.0134	0.0137
3	0.0625	0.2	0.0169	0.0169	0.0179
4	0.01	0.2	0.0185	0.0185	0.0198
5	0.0001	0.2	0.0202	0.0202	0.0219
6	1	0.1	0.0024	0.0024	0.0024
7	0.25	0.1	0.0034	0.0034	0.0034
8	0.0625	0.1	0.0045	0.0045	0.0045
9	0.01	0.1	0.0048	0.0048	0.0048
10	0.0001	0.1	0.0050	0.0050	0.0051
11	1	0.05	0.0006	0.0006	0.0006
12	0.25	0.05	0.0008	0.0008	0.0008
13	0.0625	0.05	0.0011	0.0011	0.0011
14	0.01	0.05	0.0012	0.0012	0.0012
15	0.0001	0.05	0.0012	0.0012	0.0012
16	1	0.01	0.0000	0.0000	0.0000
17	0.25	0.01	0.0000	0.0000	0.0000
18	0.0625	0.01	0.0000	0.0000	0.0000
19	0.01	0.01	0.0000	0.0000	0.0000
20	0.0001	0.01	0.0000	0.0000	0.0000
21	1	0.001	0.0000	0.0000	0.0000
22	0.25	0.001	0.0000	0.0000	0.0000
23	0.0625	0.001	0.0000	0.0000	0.0000
24	0.01	0.001	0.0000	0.0000	0.0000
25	0.0001	0.001	0.0000	0.0000	0.0000

TABLE 29: Improvement in Mean Squared Error over OLS for the Six-Component Force-Balance Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	MLS	OrthLS
1	1	0.2	0.0005	0.0005
2	0.25	0.2	0.0002	-0.0001
3	0.0625	0.2	0.0000*	-0.0010
4	0.01	0.2	0.0000*	-0.0013
5	0.0001	0.2	0.0000*	-0.0017
6	1	0.1	0.0000*	0.0000*
7	0.25	0.1	0.0000*	0.0000
8	0.0625	0.1	0.0000*	0.0000
9	0.01	0.1	0.0000*	-0.0001
10	0.0001	0.1	0.0000*	-0.0001
11	1	0.05	0.0000*	0.0000*
12	0.25	0.05	0.0000*	0.0000
13	0.0625	0.05	0.0000*	0.0000
14	0.01	0.05	0.0000*	0.0000
15	0.0001	0.05	0.0000*	0.0000
16	1	0.01	0.0000*	0.0000*
17	0.25	0.01	0.0000*	0.0000*
18	0.0625	0.01	0.0000*	0.0000
19	0.01	0.01	0.0000*	0.0000
20	0.0001	0.01	0.0000*	0.0000
21	1	0.001	0.0000*	0.0000*
22	0.25	0.001	0.0000*	0.0000*
23	0.0625	0.001	0.0000*	0.0000
24	0.01	0.001	0.0000*	0.0000
25	0.0001	0.001	0.0000*	0.0000

Note: Asterisk represents an improvement of less than 0.00005

TABLE 30: Percent Improvement in Mean Squared Error over OLS for the Six-Component Force-Balance Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1	
			MLS	OrthLS
1	1	0.2	5.48%	5.48%
2	0.25	0.2	1.21%	-0.58%
3	0.0625	0.2	0.11%	-6.13%
4	0.01	0.2	0.00%*	-7.01%
5	0.0001	0.2	0.00%*	-8.46%
6	1	0.1	1.09%	1.09%
7	0.25	0.1	0.27%	-0.07%
8	0.0625	0.1	0.02%	-0.81%
9	0.01	0.1	0.00%*	-1.08%
10	0.0001	0.1	0.00%*	-1.28%
11	1	0.05	0.25%	0.25%
12	0.25	0.05	0.07%	0.00%
13	0.0625	0.05	0.00%*	-0.21%
14	0.01	0.05	0.00%*	-0.27%
15	0.0001	0.05	0.00%*	-0.27%
16	1	0.01	0.01%	0.01%
17	0.25	0.01	0.00%*	0.00%*
18	0.0625	0.01	0.00%*	-0.01%
19	0.01	0.01	0.00%*	-0.01%
20	0.0001	0.01	0.00%*	-0.01%
21	1	0.001	0.00%*	0.00%*
22	0.25	0.001	0.00%*	0.00%*
23	0.0625	0.001	0.00%*	0.00%
24	0.01	0.001	0.00%*	0.00%
25	0.0001	0.001	0.00%*	0.00%

Note: Asterisk represents an improvement of less than 0.005%

TABLE 31: Analysis of Variance of the Mean Squared Error for the Six-Component Force-Balance Simulation

Source	DF	SS	MS	F	P
Method (OLS, MLS, OrthLS)	2	0.0000253	0.0000126	0.24	0.790
Error	4047	0.2165413	0.0000535		
Total	4049	0.2165666			

TABLE 32: Percent Improvement in Mean Squared Error Relative to Response Uncertainty for the Six-Component Force-Balance Simulation

Run No.	Variance Ratio, γ	Uncertainty, σ_ϵ	Design #1	
			MLS	OrthLS
1	1	0.2	11.45%	11.45%
2	0.25	0.2	6.41%	0.00%
3	0.0625	0.2	2.16%	0.00%
4	0.01	0.2	0.37%	0.00%
5	0.0001	0.2	0.00%*	0.00%
6	1	0.1	5.16%	5.16%
7	0.25	0.1	3.06%	0.00%
8	0.0625	0.1	1.04%	0.00%
9	0.01	0.1	0.18%	0.00%
10	0.0001	0.1	0.00%*	0.00%
11	1	0.05	2.46%	2.46%
12	0.25	0.05	1.57%	0.00%
13	0.0625	0.05	0.52%	0.00%
14	0.01	0.05	0.09%	0.00%
15	0.0001	0.05	0.00%*	0.00%
16	1	0.01	0.60%	0.60%
17	0.25	0.01	0.35%	0.05%
18	0.0625	0.01	0.12%	0.00%
19	0.01	0.01	0.02%	0.00%
20	0.0001	0.01	0.00%*	0.00%
21	1	0.001	0.32%	0.32%
22	0.25	0.001	0.20%	0.02%
23	0.0625	0.001	0.07%	0.00%
24	0.01	0.001	0.01%	0.00%
25	0.0001	0.001	0.00%*	0.00%

Note: Asterisk represents an improvement of less than 0.005%

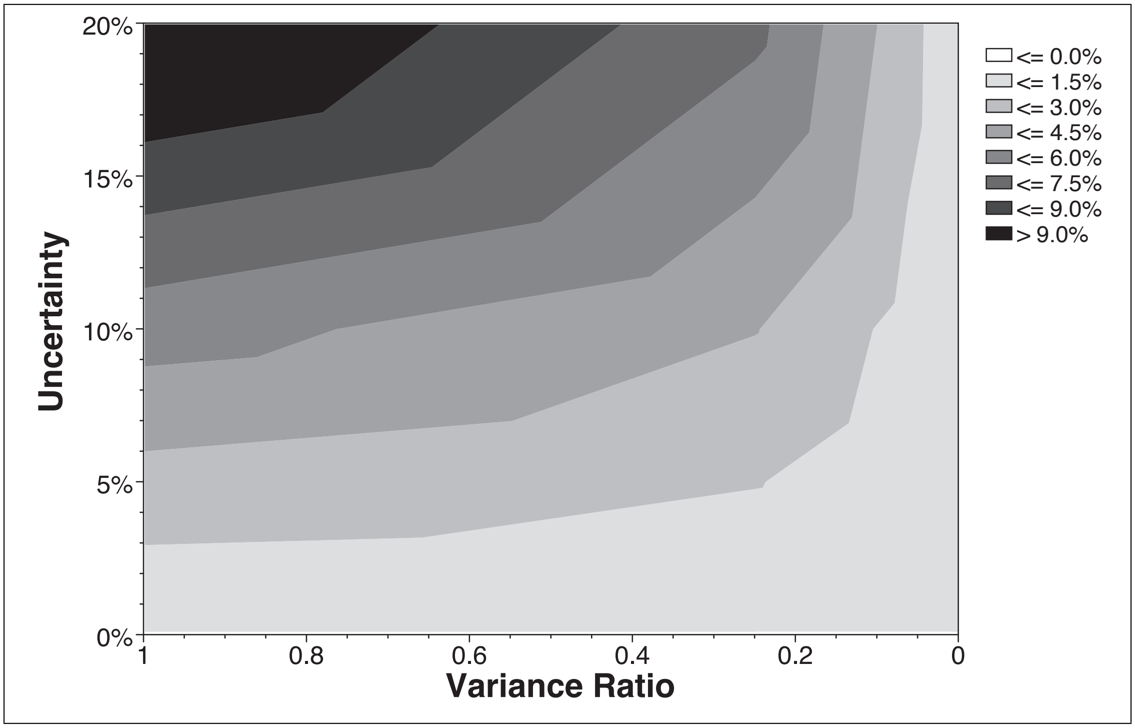

FIG. 17: Percent Improvement in Mean Squared Error of MLS for the Six-Component Force-Balance Simulation

CHAPTER 5

CONCLUSIONS

Measurement errors (ME) are a source of variability in a calibration experiment that are often ignored due to the complexities associated with the regression analysis. Standard regression techniques, such as ordinary least squares (OLS), are not appropriate in situations that contain MEs since these methods assume that the factors are known without error. In practice, it is more appropriate to assume that the factors are known within a certain level of uncertainty. Methods, such as those proposed by Adcock (1878), Kummel (1879), and Deming (1931, 1964), are available for instances where MEs are present in the data, but these methods are under-developed or limited in calibration applications at NASA. For example, there are no known methods that are capable of estimating the model for a 6-component force-balance while accounting for any MEs. Furthermore, it is reasonable to assume that a variance ratio of 1 is unlikely in calibration, which is the basis of using a method like orthogonal least squares (OrthLS). Therefore, the method of modified least squares (MLS) derived within this research is a method that accounts for any MEs and is employable for simple or complex mathematical models.

Based on the scope of the simulations, the following are contributions of the MLS method:

- New definition of residual error. In OLS, this is defined as the difference in the observed and predicted values of the response, or $Y_i - \hat{Y}_i$. However, the residual error should include other sources of variability if present, such as MEs. For MLS, the residual error is defined as a variance ratio-weighted distance between the observed and predicted data point. Geometrically, this is represented by the length d, which is the distance that is minimized in MLS. In terms of quoted accuracy of a measurement system, the former definition of residual error is an underestimate of the true accuracy.

- Smaller mean squared error (MSE). The MSE for MLS is smaller than the MSE for OLS for all the cases studied, and depending on the variance ratio and accuracy of the measurement system, the improvement in MSE ranges from 0 to 5.5 percent. The improvements are attributed to small differences in the estimates of the model coefficients. However, the practical impact of MLS is seen when the MSE is compared with the accuracy of the measurement system. Even in cases where the accuracy is high (i.e. $\sigma_\epsilon \leq 0.01$), the improvements in MSE are meaningful.

The pressure transducers for EFT-1 are an example of a simple linear instrument with a stated accuracy of 0.1 percent of full-scale (F.S.). With the NIST-traceable uncertainty in the reference standard, the variance ratio for this type of calibration is on the order of 0.0625. From the results of the simulation, the practical improvement in MSE is small, approximately 0.01 percent. The accuracies of the six components on the NTF-113 family of force-balances range from 0.1 percent to 0.5 percent of F.S. Considering the same value of variance ratio of 0.0625, the results indicate a more substantial improvement in MSE by employing MLS. The improvement for the 6-component is at least 0.07 percent.

For multiple-component calibrations, the method of MLS is derived assuming that the MEs are equal across the k factors. In practice, it is unlikely that the variances are equal. Unequal errors translate into multiple variance ratios instead of a single variance ratio, as used in the 6-component force-balance simulation. By allowing multiple variance ratios, the individual factors are more appropriately weighted based on the respective uncertainties. As a result, the potential impact of MLS could be greater when considering individual variance ratios.

Model reduction is a technique that is frequently used in calibration applications to remove terms from the estimated model that do not detectably influence the response. A statistical hypothesis test is used to evaluate whether a term in the model is influential to a specified significance level. The test is based on Student's t-distribution (Casella and Berger, 2002). The model term is considered statistically significant if

$$\left| \frac{\hat{\beta}_i}{\text{s.e.}\left(\hat{\beta}_i\right)} \right| > t_{n-2,0.025}$$

where s.e. $\left(\hat{\beta}_i\right)$ is the standard error if the β_i estimate, which is given by

$$\text{s.e.}\left(\hat{\beta}_i\right) = \sqrt{\frac{\text{Var}\left(\hat{\beta}_i\right)}{n}},$$

and $t_{n-2,0.025}$ is the t-statistic with $n-2$ degrees of freedom and a 95 percent confidence level. While model reduction is not discussed within the research, it is suspected that the terms deemed significant in OLS would also be significant in MLS and OrthLS. The same is suspected of the terms that are determined not to be significant. This rationale is based on the results that for any given estimated model coefficient, the variance of the estimate is nearly equivalent across the three methods.

Finally, the simulations conducted within this research only represent a small sample of all possible simulations that are possible. The simulations are representative of typical calibrations and life-cycles of measurement systems at NASA. However, to understand the general statistical properties of the MLS estimators, one of the following need to be completed:

- Analytical derivation of the properties. The estimators are ratios of random variables, which complicates the derivation process. The delta method can be used to obtain an asymptotic approximation of the ratios (Parker et al., 2010).

- Larger-scale simulation. In order to make general inferences based on numerical results, the simulation should consider a large number of additional calibration designs, variance ratios, and measurement system accuracies. Furthermore, the number of times that the coefficients are estimated in the simulation should be expanded since 100 is considered a small number in estimating variance.

From a practical standpoint, a larger simulation is the logical choice in proceeding with understanding the statistical properties of the estimators.

BIBLIOGRAPHY

Adcock, R. (1877). Note on the method of least squares. *The Analyst*, 4(6):183–184.

Adcock, R. (1878). A Problem in Least Squares. *The Analyst*, 5(2):53–54.

AIAA (2003). Recommended practice: Calibration and use of internal strain gage balances with application to wind tunnel testing. Technical Report R-091-2003, AIAA.

Braun, R. D. and Manning, R. M. (2006). Mars Exploration Entry, Descent, and Landing Challenges. *Journal of Spacecraft and Rockets*, 44:310–323.

Buonaccorsi, J. P. (2010). *Measurement Error: Models, Methods, and Applications.* Taylor & Francis Group, Boca Raton, FL.

Carroll, R., Gallo, P., and Gleser, L. (1985). Comparison of Least Squares and Errors-in-Variables Regression. *Journal of the American Statistical Association*, 80:929–932.

Carroll, R. and Ruppert, D. (1996). The use and misuse of orthogonal regression in linear errors-in-variables models. *The American Statistician*, 50(1):1–6.

Carroll, R. J. and Spiegelman, C. H. (1986). The effect of ignoring small measurement errors in precision instrument calibration. *Journal of Quality Technology*, 18(3):170–173.

Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Duxbury Thomson Learning, Pacific Grove, CA, 2nd edition.

Commo, S. A. and Parker, P. A. (2012). Statistical Engineering Perspective on Planetary Entry, Descent, and Landing Research. *Quality Engineering*, 24(2):306–316.

Deming, W. (1964). *Statistical Adjustment of Data*. Dover Books on Mathematics Series. Dover Publications.

Deming, W. E. (1931). The Application of Least Squares. *Philosophical Magazine Series 7*, 11(68):146–158.

Draper, N. R. and Smith, H. (1998). *Applied Regression Analysis*. John Wiley & Sons, Inc, New York, NY, 3rd edition.

Fuller, W. A. (1987). *Measurement Error Models*. John Wiley & Sons, Inc, New York, NY.

Gazarik, M. J., Little, A., Cheatwood, F. N., Wright, M. J., Herath, J. A., Martinez, E. R., Munk, M. M., Novak, F. J., and Wright, H. S. (2008). Overview of the MEDLI Project. In *2008 IEEE Aerospace Conference*, Big Sky, Montana.

Gleser, L. (1983). Functional, Structural, and Ultrastructural Errors-in-Variables Models. In *Proceedings of the Business and Economic Statistics Section*, pages 57–66, Alexandria, VA. American Statistical Association.

Golub, G. H. and Loan, C. F. V. (1980). An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893.

Guarino, J. (1964). Calibration and evaluation of multi-component strain gage balances. In *NASA Interlaboratory Force Measurements Meeting*, Jet Propulsion Laboratory, Pasadena, CA.

Kummel, C. (1879). Reduction of Observed Equations which Contain More than One Observed Quantity. *The Analyst*, 6:97–105.

Lynn, K. C., Commo, S. A., and Parker, P. A. (2012). Wind-Tunnel Balance Characterization for Hypersonic Research Applications. *Journal of Aircraft*, 49(2).

Mandel, J. (1984). Fitting straight lines when both variables are subject to error. *Journal of Quality Technology*, 16:1–13.

Montgomery, D. (2009). *Design and Analysis of Experiments*. John Wiley & Sons, New York, 7th edition.

Myers, R. H. (1990). *Classical and Modern Regression with Application*. Duxbury Press, Belmont, CA, 2nd edition.

Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M. (2009). *Response Surface Methodology*. John Wiley & Sons, 3rd edition.

Parker, P. A. and DeLoach, R. (2001). Response surface methods for force balance calibration modeling. In *IEEE 19th International Confress on Instrumentation in Aerospace Simulation Facilities*, Cleveland, Ohio.

Parker, P. A., Morton, M., Draper, N., and Line, W. (2001). A Single-Vector Force Calibration Method Featuring the Modern Design of Experiments. In *39th AIAA Aerospace Sciences Meeting & Exhibit*, Reno, Nevada. American Institute from Aeronautics and Astronautics.

Parker, P. A., Vining, G. G., Wilson, S. R., III, J. L. S., and Johnson, N. G. (2010). The Prediction Properties of the Classical and Inverse Regression for the Simple Linear Calibration Problem. *Journal of Quality Technology*, 42(4):1–16.

Pearson, K. (1901). *On Lines and Planes of Closest Fit to Systems of Points in Space*. University College.

Scheffe, H. (1973). A statistical theory of calibration. *Annals of Statistics*, 1:1–37.

Stewart, J. (2003). *Calculus: Early Transcendentals*. Thomson/Brooks/Cole, 5th edition.

APPENDIX A

SIMPLE LINEAR SIMULATION CODE

```
function [dist, ccdist, bmean, bvar] = simplinsim5(n,N,std,gamma)
% n: number of calibrations
% N: number of uses per calibration
% std: measurement system accuracy
% gamma: variance ratio (delta-to-epsilon)

warning off all;
options = optimset('Algorithm','levenberg-marquardt', ...
     'TolFun',1e-12,'TolX',1e-12,'Display','off');

tic;

%% Design Points

x1 = [-1; -1; 0; 0; 1; 1];
x2 = [-1; 0; 0; 0; 0; 1];
x3 = [-1; -1; -1; 1; 1; 1];

x1m = [1 -1; 1 -1; 1 0; 1 0; 1 1; 1 1];
x2m = [1 -1; 1 0; 1 0; 1 0; 1 0; 1 1];
x3m = [1 -1; 1 -1; 1 -1; 1 1; 1 1; 1 1];

%% Confirmation Points

xconf = [-1; -0.5; 0; 0.5; 1];

%% True Model and Responses
% b0 = 0, b1 = 1

y1 = x1;
y2 = x2;
y3 = x3;

yconf = xconf;
```

```
%% Error in x and y --- Sampled from Bivariate Normal Distribution

sigma = [gamma*(std^2) 0; 0 (std^2)];

for i = 1:n

    for j = 1:6
        cal(i).des1(j,:) = mvnrnd([x1(j,1) y1(j,1)],sigma);
        yobs_des1(j,i) = cal(i).des1(j,2);

        cal(i).des2(j,:) = mvnrnd([x2(j,1) y2(j,1)],sigma);
        yobs_des2(j,i) = cal(i).des2(j,2);

        cal(i).des3(j,:) = mvnrnd([x3(j,1) y3(j,1)],sigma);
        yobs_des3(j,i) = cal(i).des3(j,2);
    end

    %% Ordinary Least Squares Estimation

    bord1(:,i) = regress(yobs_des1(:,i),x1m);
    bord2(:,i) = regress(yobs_des2(:,i),x2m);
    bord3(:,i) = regress(yobs_des3(:,i),x3m);

    %% Ordinary Least Squares Distance

    for j = 1:6
        phi_ord1(j,i) = (pi/2) - atan(bord1(2,i));
        phi_ord2(j,i) = (pi/2) - atan(bord2(2,i));
        phi_ord3(j,i) = (pi/2) - atan(bord3(2,i));

        alpha_ord1(j,i) = (pi/2) + (1-gamma)*atan(bord1(2,i));
        alpha_ord2(j,i) = (pi/2) + (1-gamma)*atan(bord2(2,i));
        alpha_ord3(j,i) = (pi/2) + (1-gamma)*atan(bord3(2,i));

        dist_ord1(j,i) = ((sin(phi_ord1(j,i))/ ...
           sin(alpha_ord1(j,i))) ...
           (yobs_des1(j,i) - bord1(1,i) - bord1(2,i)*x1(j,1)));
        dist_ord2(j,i) = ((sin(phi_ord2(j,i))/ ...
           sin(alpha_ord2(j,i)))* ...
           (yobs_des2(j,i) - bord2(1,i) - bord2(2,i)*x2(j,1)));
        dist_ord3(j,i) = ((sin(phi_ord3(j,i))/ ...
           sin(alpha_ord3(j,i)))* ...
           (yobs_des3(j,i) - bord3(1,i) - bord3(2,i)*x3(j,1)));
    end
```

```
%% Orthogonal Least Squares Estimation

borth1(:,i) = lsqnonlin(@(borth1)(sin((pi/2) - ...
   atan(borth1(2)))/sin(pi/2))*(yobs_des1(1:6,i) - ...
   borth1(1) - borth1(2)*x1(1:6,1)),bord1(:,i),[],[],options);
borth2(:,i) = lsqnonlin(@(borth2)(sin((pi/2) - ...
   atan(borth2(2)))/sin(pi/2))*(yobs_des2(1:6,i) - ...
   borth2(1) - borth2(2)*x2(1:6,1)),bord2(:,i),[],[],options);
borth3(:,i) = lsqnonlin(@(borth3)(sin((pi/2) - ...
   atan(borth3(2)))/sin(pi/2))*(yobs_des3(1:6,i) - ...
   borth3(1) - borth3(2)*x3(1:6,1)),bord3(:,i),[],[],options);

for j = 1:6
    phi_orth1(j,i) = (pi/2) - atan(borth1(2,i));
    phi_orth2(j,i) = (pi/2) - atan(borth2(2,i));
    phi_orth3(j,i) = (pi/2) - atan(borth3(2,i));

    alpha_orth1(j,i) = (pi/2) + (1-gamma)*atan(borth1(2,i));
    alpha_orth2(j,i) = (pi/2) + (1-gamma)*atan(borth2(2,i));
    alpha_orth3(j,i) = (pi/2) + (1-gamma)*atan(borth3(2,i));

    dist_orth1(j,i) = ((sin(phi_orth1(j,i))/ ...
       sin(alpha_orth1(j,i)))*(yobs_des1(j,i) - ...
       borth1(1,i) - borth1(2,i)*x1(j,1)));
    dist_orth2(j,i) = ((sin(phi_orth2(j,i))/ ...
       sin(alpha_orth2(j,i)))*(yobs_des2(j,i) - ...
       borth2(1,i) - borth2(2,i)*x2(j,1)));
    dist_orth3(j,i) = ((sin(phi_orth3(j,i))/ ...
       sin(alpha_orth3(j,i)))*(yobs_des3(j,i) - ...
       borth3(1,i) - borth3(2,i)*x3(j,1)));
end

%% Modified Least Squares

bmod1(:,i) = lsqnonlin(@(bmod1)(sin((pi/2) - atan(bmod1(2)))/ ...
   sin((pi/2) + (1-gamma)*atan(bmod1(2))))* ...
   (yobs_des1(1:6,i) - bmod1(1) - bmod1(2)*x1(1:6,1)), ...
   bord1(:,i),[],[],options);
bmod2(:,i) = lsqnonlin(@(bmod2)(sin((pi/2) - atan(bmod2(2)))/ ...
   sin((pi/2) + (1-gamma)*atan(bmod2(2))))* ...
   (yobs_des2(1:6,i) - bmod2(1) - bmod2(2)*x2(1:6,1)), ...
   bord2(:,i),[],[],options);
bmod3(:,i) = lsqnonlin(@(bmod3)(sin((pi/2) - atan(bmod3(2)))/ ...
```

```
        sin((pi/2) + (1-gamma)*atan(bmod3(2))))* ...
        (yobs_des3(1:6,i) - bmod3(1) - bmod3(2)*x3(1:6,1)), ...
        bord3(:,i),[],[],options);

    for j = 1:6
        phi_mod1(j,i) = (pi/2) - atan(bmod1(2,i));
        phi_mod2(j,i) = (pi/2) - atan(bmod2(2,i));
        phi_mod3(j,i) = (pi/2) - atan(bmod3(2,i));

        alpha_mod1(j,i) = (pi/2) + (1-gamma)*atan(bmod1(2,i));
        alpha_mod2(j,i) = (pi/2) + (1-gamma)*atan(bmod2(2,i));
        alpha_mod3(j,i) = (pi/2) + (1-gamma)*atan(bmod3(2,i));

        dist_mod1(j,i) = ((sin(phi_mod1(j,i))/ ...
           sin(alpha_mod1(j,i)))*(yobs_des1(j,i) - ...
           bmod1(1,i) - bmod1(2,i)*x1(j,1)));
        dist_mod2(j,i) = ((sin(phi_mod2(j,i))/ ...
           sin(alpha_mod2(j,i)))*(yobs_des2(j,i) - ...
           bmod2(1,i) - bmod2(2,i)*x2(j,1)));
        dist_mod3(j,i) = ((sin(phi_mod3(j,i))/ ...
           sin(alpha_mod3(j,i)))*(yobs_des3(j,i) - ...
           bmod3(1,i) - bmod3(2,i)*x3(j,1)));
    end

end

for j = 1:6
    dist_ord1(j,n+1) = 0;
    dist_ord2(j,n+1) = 0;
    dist_ord3(j,n+1) = 0;

    dist_mod1(j,n+1) = 0;
    dist_mod2(j,n+1) = 0;
    dist_mod3(j,n+1) = 0;

    dist_orth1(j,n+1) = 0;
    dist_orth2(j,n+1) = 0;
    dist_orth3(j,n+1) = 0;

    dist_ord1(j,n+2) = mean(dist_ord1(j,1:n));
    dist_ord2(j,n+2) = mean(dist_ord2(j,1:n));
    dist_ord3(j,n+2) = mean(dist_ord3(j,1:n));

    dist_mod1(j,n+2) = mean(dist_mod1(j,1:n));
```

```
    dist_mod2(j,n+2) = mean(dist_mod2(j,1:n));
    dist_mod3(j,n+2) = mean(dist_mod3(j,1:n));

    dist_orth1(j,n+2) = mean(dist_orth1(j,1:n));
    dist_orth2(j,n+2) = mean(dist_orth2(j,1:n));
    dist_orth3(j,n+2) = mean(dist_orth3(j,1:n));

    dist_ord1(j,n+3) = var(dist_ord1(j,1:n));
    dist_ord2(j,n+3) = var(dist_ord2(j,1:n));
    dist_ord3(j,n+3) = var(dist_ord3(j,1:n));

    dist_mod1(j,n+3) = var(dist_mod1(j,1:n));
    dist_mod2(j,n+3) = var(dist_mod2(j,1:n));
    dist_mod3(j,n+3) = var(dist_mod3(j,1:n));

    dist_orth1(j,n+3) = var(dist_orth1(j,1:n));
    dist_orth2(j,n+3) = var(dist_orth2(j,1:n));
    dist_orth3(j,n+3) = var(dist_orth3(j,1:n));

    dist_ord1(j,n+4) = dist_ord1(j,n+2)^2 + dist_ord1(j,n+3);
    dist_ord2(j,n+4) = dist_ord2(j,n+2)^2 + dist_ord2(j,n+3);
    dist_ord3(j,n+4) = dist_ord3(j,n+2)^2 + dist_ord3(j,n+3);

    dist_mod1(j,n+4) = dist_mod1(j,n+2)^2 + dist_mod1(j,n+3);
    dist_mod2(j,n+4) = dist_mod2(j,n+2)^2 + dist_mod2(j,n+3);
    dist_mod3(j,n+4) = dist_mod3(j,n+2)^2 + dist_mod3(j,n+3);

    dist_orth1(j,n+4) = dist_orth1(j,n+2)^2 + dist_orth1(j,n+3);
    dist_orth2(j,n+4) = dist_orth2(j,n+2)^2 + dist_orth2(j,n+3);
    dist_orth3(j,n+4) = dist_orth3(j,n+2)^2 + dist_orth3(j,n+3);
end

dist1 = [dist_ord1(1:6,n+2:n+4); 0 0 0;  ...
   dist_mod1(1:6,n+2:n+4); 0 0 0; ...
   dist_orth1(1:6,n+2:n+4)];
dist2 = [dist_ord2(1:6,n+2:n+4); 0 0 0; ...
   dist_mod2(1:6,n+2:n+4); 0 0 0; ...
   dist_orth2(1:6,n+2:n+4)];
dist3 = [dist_ord3(1:6,n+2:n+4); 0 0 0; ...
   dist_mod3(1:6,n+2:n+4); 0 0 0; ...
   dist_orth3(1:6,n+2:n+4)];

dist = [dist1 zeros(20,1) dist2 zeros(20,1) dist3];
```

```
%% Confirmation Point Estimation

for i = 1:n

    for k = 1:N

        for j = 1:5
            cal(i,k).conf(j,:) = ...
               mvnrnd([xconf(j,1) yconf(j,1)], sigma);
            yobs(i).conf(j,k) = cal(i,k).conf(j,2);

            cdist(i).ord1(j,k) = ((sin(phi_ord1(j,i))/ ...
               sin(alpha_ord1(j,i)))*(yobs(i).conf(j,k) - ...
               bord1(1,i) - bord1(2,i)*xconf(j,1)));
            cdist(i).ord2(j,k) = ((sin(phi_ord2(j,i))/ ...
               sin(alpha_ord2(j,i)))*(yobs(i).conf(j,k) - ...
               bord2(1,i) - bord2(2,i)*xconf(j,1)));
            cdist(i).ord3(j,k) = ((sin(phi_ord3(j,i))/ ...
               sin(alpha_ord3(j,i)))*(yobs(i).conf(j,k) - ...
               bord3(1,i) - bord3(2,i)*xconf(j,1)));

            cdist(i).mod1(j,k) = ((sin(phi_mod1(j,i))/ ...
               sin(alpha_mod1(j,i)))*(yobs(i).conf(j,k) - ...
               bmod1(1,i) - bmod1(2,i)*xconf(j,1)));
            cdist(i).mod2(j,k) = ((sin(phi_mod2(j,i))/ ...
               sin(alpha_mod2(j,i)))*(yobs(i).conf(j,k) - ...
               bmod2(1,i) - bmod2(2,i)*xconf(j,1)));
            cdist(i).mod3(j,k) = ((sin(phi_mod3(j,i))/ ...
               sin(alpha_mod3(j,i)))*(yobs(i).conf(j,k) - ...
               bmod3(1,i) - bmod3(2,i)*xconf(j,1)));

            cdist(i).orth1(j,k) = ((sin(phi_orth1(j,i))/ ...
               sin(alpha_orth1(j,i)))*(yobs(i).conf(j,k) - ...
               borth1(1,i) - borth1(2,i)*xconf(j,1)));
            cdist(i).orth2(j,k) = ((sin(phi_orth2(j,i))/ ...
               sin(alpha_orth2(j,i)))*(yobs(i).conf(j,k) - ...
               borth2(1,i) - borth2(2,i)*xconf(j,1)));
            cdist(i).orth3(j,k) = ((sin(phi_orth3(j,i))/ ...
               sin(alpha_orth3(j,i)))*(yobs(i).conf(j,k) - ...
               borth3(1,i) - borth3(2,i)*xconf(j,1)));
        end

    end
```

```
    for j = 1:5
        cdist(i).ord1(j,N+1) = 0;
        cdist(i).ord2(j,N+1) = 0;
        cdist(i).ord3(j,N+1) = 0;

        cdist(i).mod1(j,N+1) = 0;
        cdist(i).mod2(j,N+1) = 0;
        cdist(i).mod3(j,N+1) = 0;

        cdist(i).orth1(j,N+1) = 0;
        cdist(i).orth2(j,N+1) = 0;
        cdist(i).orth3(j,N+1) = 0;

        cdist(i).ord1(j,N+2) = mean(cdist(i).ord1(j,1:N));
        cdist(i).ord2(j,N+2) = mean(cdist(i).ord2(j,1:N));
        cdist(i).ord3(j,N+2) = mean(cdist(i).ord3(j,1:N));

        cdist(i).mod1(j,N+2) = mean(cdist(i).mod1(j,1:N));
        cdist(i).mod2(j,N+2) = mean(cdist(i).mod2(j,1:N));
        cdist(i).mod3(j,N+2) = mean(cdist(i).mod3(j,1:N));

        cdist(i).orth1(j,N+2) = mean(cdist(i).orth1(j,1:N));
        cdist(i).orth2(j,N+2) = mean(cdist(i).orth2(j,1:N));
        cdist(i).orth3(j,N+2) = mean(cdist(i).orth3(j,1:N));

    end

    for j = 1:5
        ccdist_ord1(j,i) = cdist(i).ord1(j,N+2);
        ccdist_ord2(j,i) = cdist(i).ord2(j,N+2);
        ccdist_ord3(j,i) = cdist(i).ord3(j,N+2);

        ccdist_mod1(j,i) = cdist(i).mod1(j,N+2);
        ccdist_mod2(j,i) = cdist(i).mod2(j,N+2);
        ccdist_mod3(j,i) = cdist(i).mod3(j,N+2);

        ccdist_orth1(j,i) = cdist(i).orth1(j,N+2);
        ccdist_orth2(j,i) = cdist(i).orth2(j,N+2);
        ccdist_orth3(j,i) = cdist(i).orth3(j,N+2);
    end

end

for j = 1:5
```

```
ccdist_ord1(j,n+1) = 0;
ccdist_ord2(j,n+1) = 0;
ccdist_ord3(j,n+1) = 0;

ccdist_mod1(j,n+1) = 0;
ccdist_mod2(j,n+1) = 0;
ccdist_mod3(j,n+1) = 0;

ccdist_orth1(j,n+1) = 0;
ccdist_orth2(j,n+1) = 0;
ccdist_orth3(j,n+1) = 0;

ccdist_ord1(j,n+2) = mean(ccdist_ord1(j,1:n));
ccdist_ord2(j,n+2) = mean(ccdist_ord2(j,1:n));
ccdist_ord3(j,n+2) = mean(ccdist_ord3(j,1:n));

ccdist_mod1(j,n+2) = mean(ccdist_mod1(j,1:n));
ccdist_mod2(j,n+2) = mean(ccdist_mod2(j,1:n));
ccdist_mod3(j,n+2) = mean(ccdist_mod3(j,1:n));

ccdist_orth1(j,n+2) = mean(ccdist_orth1(j,1:n));
ccdist_orth2(j,n+2) = mean(ccdist_orth2(j,1:n));
ccdist_orth3(j,n+2) = mean(ccdist_orth3(j,1:n));

ccdist_ord1(j,n+3) = var(ccdist_ord1(j,1:n));
ccdist_ord2(j,n+3) = var(ccdist_ord2(j,1:n));
ccdist_ord3(j,n+3) = var(ccdist_ord3(j,1:n));

ccdist_mod1(j,n+3) = var(ccdist_mod1(j,1:n));
ccdist_mod2(j,n+3) = var(ccdist_mod2(j,1:n));
ccdist_mod3(j,n+3) = var(ccdist_mod3(j,1:n));

ccdist_orth1(j,n+3) = var(ccdist_orth1(j,1:n));
ccdist_orth2(j,n+3) = var(ccdist_orth2(j,1:n));
ccdist_orth3(j,n+3) = var(ccdist_orth3(j,1:n));

ccdist_ord1(j,n+4) = ccdist_ord1(j,n+2)^2 + ...
   ccdist_ord1(j,n+3);
ccdist_ord2(j,n+4) = ccdist_ord2(j,n+2)^2 + ...
   ccdist_ord2(j,n+3);
ccdist_ord3(j,n+4) = ccdist_ord3(j,n+2)^2 + ...
   ccdist_ord3(j,n+3);

ccdist_mod1(j,n+4) = ccdist_mod1(j,n+2)^2 + ...
```

```
            ccdist_mod1(j,n+3);
        ccdist_mod2(j,n+4) = ccdist_mod2(j,n+2)^2 + ...
            ccdist_mod2(j,n+3);
        ccdist_mod3(j,n+4) = ccdist_mod3(j,n+2)^2 + ...
            ccdist_mod3(j,n+3);

        ccdist_orth1(j,n+4) = ccdist_orth1(j,n+2)^2 + ...
            ccdist_orth1(j,n+3);
        ccdist_orth2(j,n+4) = ccdist_orth2(j,n+2)^2 + ...
            ccdist_orth2(j,n+3);
        ccdist_orth3(j,n+4) = ccdist_orth3(j,n+2)^2 + ...
            ccdist_orth3(j,n+3);
end

ccdist1 = [ccdist_ord1(1:5,n+2:n+4); 0 0 0; ...
   ccdist_mod1(1:5,n+2:n+4); 0 0 0; ...
   ccdist_orth1(1:5,n+2:n+4)];
ccdist2 = [ccdist_ord2(1:5,n+2:n+4); 0 0 0; ...
   ccdist_mod2(1:5,n+2:n+4); 0 0 0; ...
   ccdist_orth2(1:5,n+2:n+4)];
ccdist3 = [ccdist_ord3(1:5,n+2:n+4); 0 0 0; ...
   ccdist_mod3(1:5,n+2:n+4); 0 0 0; ...
   ccdist_orth3(1:5,n+2:n+4)];

ccdist = [ccdist1 zeros(17,1) ccdist2 zeros(17,1) ccdist3];

%% Mean and Variance of Coefficients over n Calibrations for each
%% Estimator and Design

bmean = [mean(bord1(1,:)) mean(bmod1(1,:)) mean(borth1(1,:)) 0 ...
   mean(bord2(1,:)) mean(bmod2(1,:)) mean(borth2(1,:)) 0 ...
   mean(bord3(1,:)) mean(bmod3(1,:)) mean(borth3(1,:)); ...
   mean(bord1(2,:)) mean(bmod1(2,:)) mean(borth1(2,:)) 0 ...
   mean(bord2(2,:)) mean(bmod2(2,:)) mean(borth2(2,:)) 0 ...
   mean(bord3(2,:)) mean(bmod3(2,:)) mean(borth3(2,:));];
bvar = [var(bord1(1,:)) var(bmod1(1,:)) var(borth1(1,:)) 0 ...
   var(bord2(1,:)) var(bmod2(1,:)) var(borth2(1,:)) 0 ...
   var(bord3(1,:)) var(bmod3(1,:)) var(borth3(1,:)); ...
   var(bord1(2,:)) var(bmod1(2,:)) var(borth1(2,:)) 0 ...
   var(bord2(2,:)) var(bmod2(2,:)) var(borth2(2,:)) 0 ...
   var(bord3(2,:)) var(bmod3(2,:)) var(borth3(2,:));];

t = toc;
tmin = t/60;
```

```
disp(['Elapsed time is ', num2str(tmin), ' minutes.']);
```

APPENDIX B

SIMPLE QUADRATIC SIMULATION CODE

```
function [dist, ccdist, bmean, bvar] = simpquadsim5(n,N,std,gamma)
% n: number of calibrations
% N: number of uses per calibration
% std: measurement system accuracy
% gamma: variance ratio (delta-to-epsilon)

warning off all;
options = optimset('Algorithm','levenberg-marquardt',...
   'TolFun',1e-12,'TolX',1e-12,'Display','off');

tic;

%% Design Points

x1 = [-1; -1; 0; 0; 1; 1];
x2 = [-1; 0; 0; 0; 0; 1];

x1m = [1 -1 1; 1 -1 1; 1 0 0; 1 0 0; 1 1 1; 1 1 1];
x2m = [1 -1 1; 1 0 0; 1 0 0; 1 0 0; 1 0 0; 1 1 1];

%% Confirmation Points

xconf = [-1; -0.5; 0; 0.5; 1];

%% True Model and Responses
% b0 = 0, b1 = 1, b11 = 0.2

y1 = x1 + 0.2*(x1.^2);
y2 = x2 + 0.2*(x2.^2);

yconf = xconf + 0.2*(xconf.^2);

%% Error in x and y --- Sampled from Bivariate Normal Distribution

sigma = [gamma*(std^2) 0; 0 (std^2)];
```

```
for i = 1:n

    for j = 1:6
        cal(i).des1(j,:) = mvnrnd([x1(j,1) y1(j,1)],sigma);
        yobs_des1(j,i) = cal(i).des1(j,2);

        cal(i).des2(j,:) = mvnrnd([x2(j,1) y2(j,1)],sigma);
        yobs_des2(j,i) = cal(i).des2(j,2);

    end

    %% Ordinary Least Squares Estimation

    bord1(:,i) = regress(yobs_des1(:,i),x1m);
    bord2(:,i) = regress(yobs_des2(:,i),x2m);

    %% Ordinary Least Squares Distance

    for j = 1:6
        phi_ord1(j,i) = (pi/2) - atan(bord1(2,i));
        phi_ord2(j,i) = (pi/2) - atan(bord2(2,i));

        alpha_ord1(j,i) = (pi/2) + (1-gamma)*atan(bord1(2,i));
        alpha_ord2(j,i) = (pi/2) + (1-gamma)*atan(bord2(2,i));

        dist_ord1(j,i) = ((sin(phi_ord1(j,i))/...
           sin(alpha_ord1(j,i)))*(yobs_des1(j,i) - bord1(1,i) - ...
           bord1(2,i)*x1(j,1) - bord1(3,i)*x1(j,1)*x1(j,1)));
        dist_ord2(j,i) = ((sin(phi_ord2(j,i))/...
           sin(alpha_ord2(j,i)))*(yobs_des2(j,i) - bord2(1,i) - ...
           bord2(2,i)*x2(j,1) - bord2(3,i)*x2(j,1)*x2(j,1)));
    end

    %% Orthogonal Least Squares Estimation

    borth1(:,i) = lsqnonlin(@(borth1)(sin((pi/2) - ...
       atan(borth1(2)))/sin(pi/2))*(yobs_des1(1:6,i) - ...
       borth1(1) - borth1(2)*x1(1:6,1) - borth1(3)*...
       x1(1:6,1).^2),bord1(:,i),[],[],options);
    borth2(:,i) = lsqnonlin(@(borth2)(sin((pi/2) - ...
       atan(borth2(2)))/sin(pi/2))*(yobs_des2(1:6,i) - ...
       borth2(1) - borth2(2)*x2(1:6,1) - borth2(3)*...
       x2(1:6,1).^2),bord2(:,i),[],[],options);
```

```
for j = 1:6
    phi_orth1(j,i) = (pi/2) - atan(borth1(2,i));
    phi_orth2(j,i) = (pi/2) - atan(borth2(2,i));

    alpha_orth1(j,i) = (pi/2) + (1-gamma)*atan(borth1(2,i));
    alpha_orth2(j,i) = (pi/2) + (1-gamma)*atan(borth2(2,i));

    dist_orth1(j,i) = ((sin(phi_orth1(j,i))/...
    sin(alpha_orth1(j,i)))*(yobs_des1(j,i) - borth1(1,i) - ...
    borth1(2,i)*x1(j,1) - borth1(3)*x1(j,1)^2));
    dist_orth2(j,i) = ((sin(phi_orth2(j,i))/...
       sin(alpha_orth2(j,i)))*(yobs_des2(j,i) - borth2(1,i) - ...
       borth2(2,i)*x2(j,1) - borth2(3)*x2(j,1)^2));
end

dist_orth1(7,i) = 0;
dist_orth2(7,i) = 0;

dist_orth1(8,i) = sum(dist_orth1(1:6,i));
dist_orth2(8,i) = sum(dist_orth2(1:6,i));

%% Modified Least Squares

bmod1(:,i) = lsqnonlin(@(bmod1)(sin((pi/2) - ...
   atan(bmod1(2)))/sin((pi/2) + (1-gamma)*atan(bmod1(2))))*...
   (yobs_des1(1:6,i) - bmod1(1) - bmod1(2)*x1(1:6,1) - ...
   bmod1(3)*x1(1:6,1).^2),bord1(:,i),[],[],options);
bmod2(:,i) = lsqnonlin(@(bmod2)(sin((pi/2) - ...
   atan(bmod2(2)))/sin((pi/2) + (1-gamma)*atan(bmod2(2))))*...
   (yobs_des2(1:6,i) - bmod2(1) - bmod2(2)*x2(1:6,1) - ...
   bmod2(3)*x2(1:6,1).^2),bord2(:,i),[],[],options);

for j = 1:6
    phi_mod1(j,i) = (pi/2) - atan(bmod1(2,i));
    phi_mod2(j,i) = (pi/2) - atan(bmod2(2,i));

    alpha_mod1(j,i) = (pi/2) + (1-gamma)*atan(bmod1(2,i));
    alpha_mod2(j,i) = (pi/2) + (1-gamma)*atan(bmod2(2,i));

    dist_mod1(j,i) = ((sin(phi_mod1(j,i))/...
       sin(alpha_mod1(j,i)))*(yobs_des1(j,i) - bmod1(1,i) - ...
       bmod1(2,i)*x1(j,1) - bmod1(3)*x1(j,1)^2));
    dist_mod2(j,i) = ((sin(phi_mod2(j,i))/...
```

```
            sin(alpha_mod2(j,i)))*(yobs_des2(j,i) - bmod2(1,i) - ...
            bmod2(2,i)*x2(j,1) - bmod2(3)*x2(j,1)^2));
    end

end

for j = 1:6
    dist_ord1(j,n+1) = 0;
    dist_ord2(j,n+1) = 0;

    dist_mod1(j,n+1) = 0;
    dist_mod2(j,n+1) = 0;

    dist_orth1(j,n+1) = 0;
    dist_orth2(j,n+1) = 0;

    dist_ord1(j,n+2) = mean(dist_ord1(j,1:n));
    dist_ord2(j,n+2) = mean(dist_ord2(j,1:n));

    dist_mod1(j,n+2) = mean(dist_mod1(j,1:n));
    dist_mod2(j,n+2) = mean(dist_mod2(j,1:n));

    dist_orth1(j,n+2) = mean(dist_orth1(j,1:n));
    dist_orth2(j,n+2) = mean(dist_orth2(j,1:n));

    dist_ord1(j,n+3) = var(dist_ord1(j,1:n));
    dist_ord2(j,n+3) = var(dist_ord2(j,1:n));

    dist_mod1(j,n+3) = var(dist_mod1(j,1:n));
    dist_mod2(j,n+3) = var(dist_mod2(j,1:n));

    dist_orth1(j,n+3) = var(dist_orth1(j,1:n));
    dist_orth2(j,n+3) = var(dist_orth2(j,1:n));

    dist_ord1(j,n+4) = dist_ord1(j,n+2)^2 + dist_ord1(j,n+3);
    dist_ord2(j,n+4) = dist_ord2(j,n+2)^2 + dist_ord2(j,n+3);

    dist_mod1(j,n+4) = dist_mod1(j,n+2)^2 + dist_mod1(j,n+3);
    dist_mod2(j,n+4) = dist_mod2(j,n+2)^2 + dist_mod2(j,n+3);

    dist_orth1(j,n+4) = dist_orth1(j,n+2)^2 + dist_orth1(j,n+3);
    dist_orth2(j,n+4) = dist_orth2(j,n+2)^2 + dist_orth2(j,n+3);
end
```

```
dist1 = [dist_ord1(1:6,n+2:n+4); 0 0 0; ...
   dist_mod1(1:6,n+2:n+4); 0 0 0; ...
   dist_orth1(1:6,n+2:n+4)];
dist2 = [dist_ord2(1:6,n+2:n+4); 0 0 0; ...
   dist_mod2(1:6,n+2:n+4); 0 0 0; ...
   dist_orth2(1:6,n+2:n+4)];

dist = [dist1 zeros(20,1) dist2];

for i = 1:n

    for k = 1:N

        for j = 1:5
            cal(i,k).conf(j,:) = ...
               mvnrnd([xconf(j,1) yconf(j,1)], sigma);
            yobs(i).conf(j,k) = cal(i,k).conf(j,2);

            cdist(i).ord1(j,k) = ((sin(phi_ord1(j,i))/...
               sin(alpha_ord1(j,i)))*(yobs(i).conf(j,k) - ...
               bord1(1,i) - bord1(2,i)*xconf(j,1) - ...
               bord1(3,i)*xconf(j,1)^2));
            cdist(i).ord2(j,k) = ((sin(phi_ord2(j,i))/...
               sin(alpha_ord2(j,i)))*(yobs(i).conf(j,k) - ...
               bord2(1,i) - bord2(2,i)*xconf(j,1) - ...
               bord2(3,i)*xconf(j,1)^2));

            cdist(i).mod1(j,k) = ((sin(phi_mod1(j,i))/...
               sin(alpha_mod1(j,i)))*(yobs(i).conf(j,k) - ...
               bmod1(1,i) - bmod1(2,i)*xconf(j,1) - ...
               bmod1(3,i)*xconf(j,1)^2));
            cdist(i).mod2(j,k) = ((sin(phi_mod2(j,i))/...
               sin(alpha_mod2(j,i)))*(yobs(i).conf(j,k) - ...
               bmod2(1,i) - bmod2(2,i)*xconf(j,1) - ...
               bmod2(3,i)*xconf(j,1)^2));

            cdist(i).orth1(j,k) = ((sin(phi_orth1(j,i))/...
               sin(alpha_orth1(j,i)))*(yobs(i).conf(j,k) - ...
               borth1(1,i) - borth1(2,i)*xconf(j,1) - ...
               borth1(3,i)*xconf(j,1)^2));
            cdist(i).orth2(j,k) = ((sin(phi_orth2(j,i))/...
               sin(alpha_orth2(j,i)))*(yobs(i).conf(j,k) - ...
               borth2(1,i) - borth2(2,i)*xconf(j,1) - ...
               borth2(3,i)*xconf(j,1)^2));
```

```
    end

end

for j = 1:5
    cdist(i).ord1(j,N+1) = 0;
    cdist(i).ord2(j,N+1) = 0;

    cdist(i).mod1(j,N+1) = 0;
    cdist(i).mod2(j,N+1) = 0;

    cdist(i).orth1(j,N+1) = 0;
    cdist(i).orth2(j,N+1) = 0;

    cdist(i).ord1(j,N+2) = mean(cdist(i).ord1(j,1:N));
    cdist(i).ord2(j,N+2) = mean(cdist(i).ord2(j,1:N));

    cdist(i).mod1(j,N+2) = mean(cdist(i).mod1(j,1:N));
    cdist(i).mod2(j,N+2) = mean(cdist(i).mod2(j,1:N));

    cdist(i).orth1(j,N+2) = mean(cdist(i).orth1(j,1:N));
    cdist(i).orth2(j,N+2) = mean(cdist(i).orth2(j,1:N));

    cdist(i).ord1(j,N+3) = var(cdist(i).ord1(j,1:N));
    cdist(i).ord2(j,N+3) = var(cdist(i).ord2(j,1:N));

    cdist(i).mod1(j,N+3) = var(cdist(i).mod1(j,1:N));
    cdist(i).mod2(j,N+3) = var(cdist(i).mod2(j,1:N));

    cdist(i).orth1(j,N+3) = var(cdist(i).orth1(j,1:N));
    cdist(i).orth2(j,N+3) = var(cdist(i).orth2(j,1:N));

end

for j = 1:5
    ccdist_ord1(j,i) = cdist(i).ord1(j,N+2);
    ccdist_ord2(j,i) = cdist(i).ord2(j,N+2);

    ccdist_mod1(j,i) = cdist(i).mod1(j,N+2);
    ccdist_mod2(j,i) = cdist(i).mod2(j,N+2);

    ccdist_orth1(j,i) = cdist(i).orth1(j,N+2);
    ccdist_orth2(j,i) = cdist(i).orth2(j,N+2);
end
```

```
end

for j = 1:5
    ccdist_ord1(j,n+1) = 0;
    ccdist_ord2(j,n+1) = 0;

    ccdist_mod1(j,n+1) = 0;
    ccdist_mod2(j,n+1) = 0;

    ccdist_orth1(j,n+1) = 0;
    ccdist_orth2(j,n+1) = 0;

    ccdist_ord1(j,n+2) = mean(ccdist_ord1(j,1:n));
    ccdist_ord2(j,n+2) = mean(ccdist_ord2(j,1:n));

    ccdist_mod1(j,n+2) = mean(ccdist_mod1(j,1:n));
    ccdist_mod2(j,n+2) = mean(ccdist_mod2(j,1:n));

    ccdist_orth1(j,n+2) = mean(ccdist_orth1(j,1:n));
    ccdist_orth2(j,n+2) = mean(ccdist_orth2(j,1:n));

    ccdist_ord1(j,n+3) = var(ccdist_ord1(j,1:n));
    ccdist_ord2(j,n+3) = var(ccdist_ord2(j,1:n));

    ccdist_mod1(j,n+3) = var(ccdist_mod1(j,1:n));
    ccdist_mod2(j,n+3) = var(ccdist_mod2(j,1:n));

    ccdist_orth1(j,n+3) = var(ccdist_orth1(j,1:n));
    ccdist_orth2(j,n+3) = var(ccdist_orth2(j,1:n));

    ccdist_ord1(j,n+4) = ccdist_ord1(j,n+2)^2 + ccdist_ord1(j,n+3);
    ccdist_ord2(j,n+4) = ccdist_ord2(j,n+2)^2 + ccdist_ord2(j,n+3);

    ccdist_mod1(j,n+4) = ccdist_mod1(j,n+2)^2 + ccdist_mod1(j,n+3);
    ccdist_mod2(j,n+4) = ccdist_mod2(j,n+2)^2 + ccdist_mod2(j,n+3);

    ccdist_orth1(j,n+4) = ccdist_orth1(j,n+2)^2 + ccdist_orth1(j,n+3);
    ccdist_orth2(j,n+4) = ccdist_orth2(j,n+2)^2 + ccdist_orth2(j,n+3);
end

ccdist1 = [ccdist_ord1(1:5,n+2:n+4); 0 0 0; ...
   ccdist_mod1(1:5,n+2:n+4); 0 0 0; ...
   ccdist_orth1(1:5,n+2:n+4)];
```

```
ccdist2 = [ccdist_ord2(1:5,n+2:n+4); 0 0 0; ...
   ccdist_mod2(1:5,n+2:n+4); 0 0 0; ...
   ccdist_orth2(1:5,n+2:n+4)];

ccdist = [ccdist1 zeros(17,1) ccdist2];

%% Mean and Variance of Coefficients over n Calibrations for
%% each Estimator and Design

bmean = [mean(bord1(1,:)) mean(bmod1(1,:)) mean(borth1(1,:)) 0 ...
   mean(bord2(1,:)) mean(bmod2(1,:)) mean(borth2(1,:)); ...
   mean(bord1(2,:)) mean(bmod1(2,:)) mean(borth1(2,:)) 0 ...
   mean(bord2(2,:)) mean(bmod2(2,:)) mean(borth2(2,:));];
bvar = [var(bord1(1,:)) var(bmod1(1,:)) var(borth1(1,:)) 0 ...
   var(bord2(1,:)) var(bmod2(1,:)) var(borth2(1,:)); ...
   var(bord1(2,:)) var(bmod1(2,:)) var(borth1(2,:)) 0 ...
   var(bord2(2,:)) var(bmod2(2,:)) var(borth2(2,:));];

t = toc;
tmin = t/60;
disp(['Elapsed time is ', num2str(tmin), ' minutes.']);
```

APPENDIX C

6-COMPONENT FORCE-BALANCE SIMULATION CODE

```
function [dist, ccdist, bmean, bvar] = multsim5(n,N,std,gamma)
% n: number of calibrations
% N: number of uses per calibration
% std: measurement system accuracy
% gamma: variance ratio (delta-to-epsilon)

warning off all;
options = optimset('Algorithm','levenberg-marquardt',...
   'TolFun',1e-12,'TolX',1e-12,'Display','off');

tic;

%% Design Points - Central Composite Design

x1 = [-0.639 -0.639 -0.639 -0.639 -0.639 -0.639; ...
0.000 0.000 0.000 0.000 0.000 0.000; ...
-0.639 -0.639 0.639 0.639 0.639 0.639; ...
0.000 -1.00 0.000 0.000 0.000 0.000; ...
0.639 0.639 0.639 -0.639 -0.639 0.639; ...
-0.639 0.639 -0.639 0.639 0.639 0.639; ...
0.639 0.639 0.639 0.639 0.639 0.639; ...
0.639 -0.639 -0.639 0.639 -0.639 -0.639; ...
0.000 0.000 0.000 0.000 0.000 1.00; ...
-0.639 -0.639 0.639 0.639 -0.639 -0.639; ...
0.639 -0.639 0.639 -0.639 -0.639 -0.639; ...
0.000 0.000 0.000 0.000 0.000 0.000; ...
1.00 0.000 0.000 0.000 0.000 0.000; ...
0.639 0.639 -0.639 -0.639 0.639 0.639; ...
0.639 -0.639 -0.639 -0.639 -0.639 0.639; ...
0.639 -0.639 0.639 0.639 -0.639 0.639; ...
0.000 0.000 0.000 0.000 0.000 -1.00; ...
-0.639 0.639 0.639 0.639 0.639 -0.639; ...
```

```
-1.00 0.000 0.000 0.000 0.000 0.000; ...
0.000 0.000 0.000 0.000 0.000 0.000; ...
-0.639 -0.639 -0.639 0.639 -0.639 0.639; ...
0.639 0.639 -0.639 -0.639 -0.639 -0.639; ...
0.639 0.639 0.639 -0.639 0.639 -0.639; ...
-0.639 0.639 0.639 0.639 -0.639 0.639; ...
-0.639 0.639 -0.639 0.639 -0.639 -0.639; ...
0.000 0.000 0.000 -1.00 0.000 0.000; ...
0.000 0.000 0.000 0.000 0.000 0.000; ...
-0.639 0.639 0.639 -0.639 -0.639 -0.639; ...
0.639 -0.639 -0.639 -0.639 0.639 -0.639; ...
0.000 1.00 0.000 0.000 0.000 0.000; ...
-0.639 -0.639 -0.639 -0.639 0.639 0.639; ...
0.639 0.639 -0.639 0.639 0.639 -0.639; ...
0.639 -0.639 0.639 0.639 0.639 -0.639; ...
0.639 0.639 -0.639 0.639 -0.639 0.639; ...
0.639 -0.639 0.639 -0.639 0.639 0.639; ...
0.000 0.000 0.000 0.000 0.000 0.000; ...
-0.639 -0.639 0.639 -0.639 0.639 -0.639; ...
0.000 0.000 0.000 0.000 0.000 0.000; ...
0.000 0.000 0.000 0.000 1.00 0.000; ...
0.000 0.000 0.000 1.00 0.000 0.000; ...
0.639 -0.639 -0.639 0.639 0.639 0.639; ...
0.000 0.000 0.000 0.000 0.000 0.000; ...
0.000 0.000 0.000 0.000 -1.00 0.000; ...
-0.639 0.639 0.639 -0.639 0.639 0.639; ...
0.639 0.639 0.639 0.639 -0.639 -0.639; ...
0.000 0.000 0.000 0.000 0.000 0.000; ...
-0.639 0.639 -0.639 -0.639 0.639 -0.639; ...
0.000 0.000 1.00 0.000 0.000 0.000; ...
0.000 0.000 0.000 0.000 0.000 0.000; ...
-0.639 -0.639 0.639 -0.639 -0.639 0.639; ...
-0.639 -0.639 -0.639 0.639 0.639 -0.639; ...
0.000 0.000 0.000 0.000 0.000 0.000; ...
-0.639 0.639 -0.639 -0.639 -0.639 0.639; ...
0.000 0.000 -1.00 0.000 0.000 0.000];

x1m = x2fx(x1,'quadratic');

%% Confirmation Points

xconf = [-0.138 -0.600 -0.414 0.264 -0.138 -0.600; ...
-0.600 -0.600 -0.600 0.600 0.600 0.366; ...
-0.600 -0.600 0.372 -0.600 -0.600 -0.600; ...
```

```
-0.126 0.600 -0.0960 -0.600 -0.228 -0.270; ...
-0.126 0.600 -0.0960 -0.600 -0.228 -0.270; ...
-0.600 -0.600 0.600 0.600 0.600 -0.600; ...
-0.600 0.600 0.600 -0.600 -0.600 0.600; ...
-0.528 0.600 -0.348 0.288 -0.00600 0.600; ...
0.600 -0.600 -0.600 -0.600 -0.258 0.600; ...
-0.600 0.600 -0.600 0.600 -0.600 -0.600; ...
0.600 0.228 -0.552 0.600 0.600 0.600; ...
-0.540 -0.408 0.0720 -0.474 0.462 0.600; ...
-0.450 -0.120 -0.600 -0.288 -0.600 0.168; ...
-0.600 -0.124 0.150 0.600 -0.264 -0.0480; ...
-0.186 0.174 0.600 0.000 -0.306 -0.600; ...
-0.600 -0.124 0.150 0.600 -0.264 -0.0480; ...
0.504 -0.558 -0.0942 -0.0780 0.600 -0.0840; ...
0.600 0.600 0.600 -0.600 0.600 -0.600; ...
0.600 -0.120 -0.600 -0.600 -0.600 -0.600; ...
0.600 0.528 -0.600 0.234 0.0809 -0.216; ...
0.600 0.600 0.162 0.600 -0.227 -0.600; ...
0.0308 0.252 -0.0715 0.600 0.600 -0.510; ...
0.180 -0.270 0.600 -0.600 0.102 0.0360; ...
-0.126 -0.600 0.600 0.239 -0.600 0.600; ...
0.600 0.216 0.178 -0.228 -0.600 0.456; ...
0.180 -0.270 0.600 -0.600 0.102 0.0360; ...
0.288 0.540 -0.600 -0.600 0.600 0.600; ...
0.600 -0.600 -0.324 0.0420 -0.600 -0.0180; ...
0.504 -0.558 -0.0942 -0.0780 0.600 -0.0840; ...
0.600 -0.600 0.600 0.600 -0.600 -0.534; ...
0.366 -0.180 -0.600 0.600 -0.600 0.600; ...
-0.186 0.174 0.600 0.000 -0.306 -0.600; ...
-0.576 0.600 0.582 -0.0660 0.600 0.0720; ...
0.240 0.600 0.600 0.600 -0.600 0.240; ...
-0.600 0.540 -0.600 -0.466 0.114 0.250; ...
-0.600 0.0420 -0.600 -0.600 0.600 -0.600; ...
0.600 0.600 0.126 -0.0900 0.336 0.341; ...
0.600 -0.0360 0.600 0.600 0.402 0.600];

xconfm = x2fx(xconf,'quadratic');

%% True Model and Responses
% Beta's Based on NTF-113C Calibration - Axial Force Component

beta = [0 0.1334 1 -0.0095 0.0257 -0.0029 0.0006 0.0321 ...
   0.0934 0.0014 -0.0027 0.0023 -0.0025 0 -0.0003 0.0009 0 ...
   -0.0013 -0.0001 -0.0683 -0.0105 0.0306 -0.1015 0.0007 0.0387 ...
```

```
   0.0624 -0.0025 -0.0299]';

y1 = x1m*beta;

yconf = xconfm*beta;

%% Error in x and y --- Sampled from Multi-variate
%% Normal Distribution

sigma = [gamma*(std^2) 0 0 0 0 0 0; 0 gamma*(std^2) 0 0 0 0 0; ...
    0 0 gamma*(std^2) 0 0 0 0; 0 0 0 gamma*(std^2) 0 0 0; ...
    0 0 0 0 gamma*(std^2) 0 0; 0 0 0 0 0 gamma*(std^2) 0; ...
    0 0 0 0 0 0 std^2];

for i = 1:n

    for j = 1:length(x1(:,1))
        cal(i).des1(j,:) = mvnrnd([x1(j,1) x1(j,2) x1(j,3) ...
           x1(j,4) x1(j,5) x1(j,6) y1(j,1)],sigma);
        yobs_des1(j,i) = cal(i).des1(j,7);

    end

    %% Ordinary Least Squares Estimation

    bord1(:,i) = regress(yobs_des1(:,i),x1m);

    %% Ordinary Least Squares Distance

    for j = 1:length(x1(:,1))
        phi_ord1(j,i) = (pi/2) - atan(sqrt( ...
            (bord1(2,i) + bord1(8,i)*x1(j,2) + ...
            bord1(9,i)*x1(j,3) + bord1(10,i)*x1(j,4) + ...
            bord1(11,i)*x1(j,5) + bord1(12,i)*x1(j,6) + ...
            2*bord1(23,i)*x1(j,1))^2 + (bord1(3,i) + ...
            bord1(8,i)*x1(j,1) + bord1(13,i)*x1(j,3) + ...
            bord1(14,i)*x1(j,4) + bord1(15,i)*x1(j,5) + ...
            bord1(16,i)*x1(j,6) + 2*bord1(24,i)*x1(j,2))^2 + ...
            (bord1(4,i) + bord1(9,i)*x1(j,1) + ...
            bord1(13,i)*x1(j,2) + bord1(17,i)*x1(j,4) + ...
            bord1(18,i)*x1(j,5) + bord1(19,i)*x1(j,6) + ...
            2*bord1(25,i)*x1(j,3))^2 + (bord1(5,i) + ...
            bord1(10,i)*x1(j,1) + bord1(14,i)*x1(j,2) + ...
            bord1(17,i)*x1(j,3) + bord1(20,i)*x1(j,5) + ...
```

```
        bord1(21,i)*x1(j,6) + 2*bord1(26,i)*x1(j,4))^2 + ...
        (bord1(6,i) + bord1(11,i)*x1(j,1) + ...
        bord1(15,i)*x1(j,2) + bord1(18,i)*x1(j,3) + ...
        bord1(20,i)*x1(j,4) + bord1(22,i)*x1(j,6) + ...
        2*bord1(27,i)*x1(j,5))^2 + (bord1(7,i) + ...
        ord1(12,i)*x1(j,1) + bord1(16,i)*x1(j,2) + ...
        bord1(19,i)*x1(j,3) + bord1(21,i)*x1(j,4) + ...
        bord1(22,i)*x1(j,5) + 2*bord1(28,i)*x1(j,6))^2));

    alpha_ord1(j,i) = (pi/2) + (1-gamma)*atan(sqrt( ...
        (bord1(2,i) + bord1(8,i)*x1(j,2) + ...
        bord1(9,i)*x1(j,3) + bord1(10,i)*x1(j,4) + ...
        bord1(11,i)*x1(j,5) + bord1(12,i)*x1(j,6) + ...
        2*bord1(23,i)*x1(j,1))^2 + (bord1(3,i) + ...
        bord1(8,i)*x1(j,1) + bord1(13,i)*x1(j,3) + ...
        bord1(14,i)*x1(j,4) + bord1(15,i)*x1(j,5) + ...
        bord1(16,i)*x1(j,6) + 2*bord1(24,i)*x1(j,2))^2 + ...
        (bord1(4,i) + bord1(9,i)*x1(j,1) + ...
        bord1(13,i)*x1(j,2) + bord1(17,i)*x1(j,4) + ...
        bord1(18,i)*x1(j,5) + bord1(19,i)*x1(j,6) + ...
        2*bord1(25,i)*x1(j,3))^2 + (bord1(5,i) + ...
        bord1(10,i)*x1(j,1) + bord1(14,i)*x1(j,2) + ...
        bord1(17,i)*x1(j,3) + bord1(20,i)*x1(j,5) + ...
        bord1(21,i)*x1(j,6) + 2*bord1(26,i)*x1(j,4))^2 + ...
        (bord1(6,i) + bord1(11,i)*x1(j,1) + ...
        bord1(15,i)*x1(j,2) + bord1(18,i)*x1(j,3) + ...
        bord1(20,i)*x1(j,4) + bord1(22,i)*x1(j,6) + ...
        2*bord1(27,i)*x1(j,5))^2 + (bord1(7,i) + ...
        bord1(12,i)*x1(j,1) + bord1(16,i)*x1(j,2) + ...
        bord1(19,i)*x1(j,3) + bord1(21,i)*x1(j,4) + ...
        bord1(22,i)*x1(j,5) + 2*bord1(28,i)*x1(j,6))^2));

    ratio_ord1(j,i) = sin(phi_ord1(j,i))/sin(alpha_ord1(j,i));

end

res_ord1(1:j,i) = (yobs_des1(:,i) - x1m*bord1(:,i));

for j = 1:length(x1(:,1))

    dist_ord1(j,i) = ratio_ord1(j,i)*res_ord1(j,i);

end
```

```
%% Orthogonal Least Squares Estimation

borth1(:,i) = lsqnonlin(@(borth1)(sin((pi/2) - atan(sqrt( ...
   (borth1(2) + borth1(8)*x1(1:j,2) + borth1(9)*x1(1:j,3) + ...
   borth1(10)*x1(1:j,4) + borth1(11)*x1(1:j,5) + ...
   borth1(12)*x1(1:j,6) + 2*borth1(23)*x1(1:j,1)).^2 + ...
   (borth1(3) + borth1(8)*x1(1:j,1) + borth1(13)*x1(1:j,3) + ...
   borth1(14)*x1(1:j,4) + borth1(15)*x1(1:j,5) + ...
   borth1(16)*x1(1:j,6) + 2*borth1(24)*x1(1:j,2)).^2 + ...
   (borth1(4) + borth1(9)*x1(1:j,1) + borth1(13)*x1(1:j,2) + ...
   borth1(17)*x1(1:j,4) + borth1(18)*x1(1:j,5) + ...
   borth1(19)*x1(1:j,6) + 2*borth1(25)*x1(1:j,3)).^2 + ...
   (borth1(5) + borth1(10)*x1(1:j,1) + borth1(14)*x1(1:j,2) + ...
   borth1(17)*x1(1:j,3) + borth1(20)*x1(1:j,5) + ...
   borth1(21)*x1(1:j,6) + 2*borth1(26)*x1(1:j,4)).^2 + ...
   (borth1(6) + borth1(11)*x1(1:j,1) + borth1(15)*x1(1:j,2) + ...
   borth1(18)*x1(1:j,3) + borth1(20)*x1(1:j,4) + ...
   borth1(22)*x1(1:j,6) + 2*borth1(27)*x1(1:j,5)).^2 + ...
   (borth1(7) + borth1(12)*x1(1:j,1) + borth1(16)*x1(1:j,2) + ...
   borth1(19)*x1(1:j,3) + borth1(21)*x1(1:j,4) + ...
   borth1(22)*x1(1:j,5) + 2*borth1(28)*x1(1:j,6)).^2))./...
   sin(pi/2)).*(yobs_des1(1:j,i) - x1m*borth1(1:28)),...
   bord1(:,i),[],[],options);

for j = 1:length(x1(:,1))
    phi_orth1(j,i) = (pi/2) - atan(sqrt( ...
       (borth1(2,i) + borth1(8,i)*x1(j,2) + ...
       borth1(9,i)*x1(j,3) + borth1(10,i)*x1(j,4) + ...
       borth1(11,i)*x1(j,5) + borth1(12,i)*x1(j,6) + ...
       2*borth1(23,i)*x1(j,1))^2 + (borth1(3,i) + ...
       borth1(8,i)*x1(j,1) + borth1(13,i)*x1(j,3) + ...
       borth1(14,i)*x1(j,4) + borth1(15,i)*x1(j,5) + ...
       borth1(16,i)*x1(j,6) + 2*borth1(24,i)*x1(j,2))^2 + ...
       (borth1(4,i) + borth1(9,i)*x1(j,1) + ...
       borth1(13,i)*x1(j,2) + borth1(17,i)*x1(j,4) + ...
       borth1(18,i)*x1(j,5) + borth1(19,i)*x1(j,6) + ...
       2*borth1(25,i)*x1(j,3))^2 + (borth1(5,i) + ...
       borth1(10,i)*x1(j,1) + borth1(14,i)*x1(j,2) + ...
       borth1(17,i)*x1(j,3) + borth1(20,i)*x1(j,5) + ...
       borth1(21,i)*x1(j,6) + 2*borth1(26,i)*x1(j,4))^2 + ...
       (borth1(6,i) + borth1(11,i)*x1(j,1) + ...
       borth1(15,i)*x1(j,2) + borth1(18,i)*x1(j,3) + ...
       borth1(20,i)*x1(j,4) + borth1(22,i)*x1(j,6) + ...
       2*borth1(27,i)*x1(j,5))^2 + (borth1(7,i) + ...
```

```
        borth1(12,i)*x1(j,1) + borth1(16,i)*x1(j,2) + ...
        borth1(19,i)*x1(j,3) + borth1(21,i)*x1(j,4) + ...
        borth1(22,i)*x1(j,5) + 2*borth1(28,i)*x1(j,6))^2));

    alpha_orth1(j,i) = (pi/2) + (1-gamma)*atan(sqrt( ...
        (borth1(2,i) + borth1(8,i)*x1(j,2) + ...
        borth1(9,i)*x1(j,3) + borth1(10,i)*x1(j,4) + ...
        borth1(11,i)*x1(j,5) + borth1(12,i)*x1(j,6) + ...
        2*borth1(23,i)*x1(j,1))^2 + (borth1(3,i) + ...
        borth1(8,i)*x1(j,1) + borth1(13,i)*x1(j,3) + ...
        borth1(14,i)*x1(j,4) + borth1(15,i)*x1(j,5) + ...
        borth1(16,i)*x1(j,6) + 2*borth1(24,i)*x1(j,2))^2 + ...
        (borth1(4,i) + borth1(9,i)*x1(j,1) + ...
        borth1(13,i)*x1(j,2) + borth1(17,i)*x1(j,4) + ...
        borth1(18,i)*x1(j,5) + borth1(19,i)*x1(j,6) + ...
        2*borth1(25,i)*x1(j,3))^2 + (borth1(5,i) + ...
        borth1(10,i)*x1(j,1) + borth1(14,i)*x1(j,2) + ...
        borth1(17,i)*x1(j,3) + borth1(20,i)*x1(j,5) + ...
        borth1(21,i)*x1(j,6) + 2*borth1(26,i)*x1(j,4))^2 + ...
        (borth1(6,i) + borth1(11,i)*x1(j,1) + ...
        borth1(15,i)*x1(j,2) + borth1(18,i)*x1(j,3) + ...
        borth1(20,i)*x1(j,4) + borth1(22,i)*x1(j,6) + ...
        2*borth1(27,i)*x1(j,5))^2 + (borth1(7,i) + ...
        borth1(12,i)*x1(j,1) + borth1(16,i)*x1(j,2) + ...
        borth1(19,i)*x1(j,3) + borth1(21,i)*x1(j,4) + ...
        borth1(22,i)*x1(j,5) + 2*borth1(28,i)*x1(j,6))^2));

    ratio_orth1(j,i) = sin(phi_orth1(j,i))/...
        sin(alpha_orth1(j,i));

end

res_orth1(1:j,i) = (yobs_des1(:,i) - x1m*borth1(:,i));

for j = 1:length(x1(:,1))

    dist_orth1(j,i) = ratio_orth1(j,i)*res_orth1(j,i);

end

%% Modified Least Squares

bmod1(:,i) = lsqnonlin(@(bmod1)(sin((pi/2) - atan(sqrt( ...
    (bmod1(2) + bmod1(8)*x1(1:j,2) + bmod1(9)*x1(1:j,3) + ...
```

```
    bmod1(10)*x1(1:j,4) + bmod1(11)*x1(1:j,5) + ...
    bmod1(12)*x1(1:j,6) + 2*bmod1(23)*x1(1:j,1)).^2 + ...
    (bmod1(3) + bmod1(8)*x1(1:j,1) + bmod1(13)*x1(1:j,3) + ...
    bmod1(14)*x1(1:j,4) + bmod1(15)*x1(1:j,5) + ...
    bmod1(16)*x1(1:j,6) + 2*bmod1(24)*x1(1:j,2)).^2 + ...
    (bmod1(4) + bmod1(9)*x1(1:j,1) + bmod1(13)*x1(1:j,2) + ...
    bmod1(17)*x1(1:j,4) + bmod1(18)*x1(1:j,5) + ...
    bmod1(19)*x1(1:j,6) + 2*bmod1(25)*x1(1:j,3)).^2 + ...
    (bmod1(5) + bmod1(10)*x1(1:j,1) + bmod1(14)*x1(1:j,2) + ...
    bmod1(17)*x1(1:j,3) + bmod1(20)*x1(1:j,5) + ...
    bmod1(21)*x1(1:j,6) + 2*bmod1(26)*x1(1:j,4)).^2 + ...
    (bmod1(6) + bmod1(11)*x1(1:j,1) + bmod1(15)*x1(1:j,2) + ...
    bmod1(18)*x1(1:j,3) + bmod1(20)*x1(1:j,4) + ...
    bmod1(22)*x1(1:j,6) + 2*bmod1(27)*x1(1:j,5)).^2 + ...
    (bmod1(7) + bmod1(12)*x1(1:j,1) + bmod1(16)*x1(1:j,2) + ...
    bmod1(19)*x1(1:j,3) + bmod1(21)*x1(1:j,4) + ...
    bmod1(22)*x1(1:j,5) + 2*bmod1(28)*x1(1:j,6)).^2 ...
    ))./sin((pi/2) + (1-gamma)*atan(sqrt( ...
    (bmod1(2) + bmod1(8)*x1(1:j,2) + bmod1(9)*x1(1:j,3) + ...
    bmod1(10)*x1(1:j,4) + bmod1(11)*x1(1:j,5) + ...
    bmod1(12)*x1(1:j,6) + 2*bmod1(23)*x1(1:j,1)).^2 + ...
    (bmod1(3) + bmod1(8)*x1(1:j,1) + bmod1(13)*x1(1:j,3) + ...
    bmod1(14)*x1(1:j,4) + bmod1(15)*x1(1:j,5) + ...
    bmod1(16)*x1(1:j,6) + 2*bmod1(24)*x1(1:j,2)).^2 + ...
    (bmod1(4) + bmod1(9)*x1(1:j,1) + bmod1(13)*x1(1:j,2) + ...
    bmod1(17)*x1(1:j,4) + bmod1(18)*x1(1:j,5) + ...
    bmod1(19)*x1(1:j,6) + 2*bmod1(25)*x1(1:j,3)).^2 + ...
    (bmod1(5) + bmod1(10)*x1(1:j,1) + bmod1(14)*x1(1:j,2) + ...
    bmod1(17)*x1(1:j,3) + bmod1(20)*x1(1:j,5) + ...
    bmod1(21)*x1(1:j,6) + 2*bmod1(26)*x1(1:j,4)).^2 + ...
    (bmod1(6) + bmod1(11)*x1(1:j,1) + bmod1(15)*x1(1:j,2) + ...
    bmod1(18)*x1(1:j,3) + bmod1(20)*x1(1:j,4) + ...
    bmod1(22)*x1(1:j,6) + 2*bmod1(27)*x1(1:j,5)).^2 + ...
    (bmod1(7) + bmod1(12)*x1(1:j,1) + bmod1(16)*x1(1:j,2) + ...
    bmod1(19)*x1(1:j,3) + bmod1(21)*x1(1:j,4) + ...
    bmod1(22)*x1(1:j,5) + 2*bmod1(28)*x1(1:j,6)).^2 ...
    )))).*(yobs_des1(1:j,i) - x1m*bmod1(1:28)),...
    bord1(:,i),[],[],options);

for j = 1:length(x1(:,1))
    phi_mod1(j,i) = (pi/2) - atan(sqrt( ...
        (bmod1(2,i) + bmod1(8,i)*x1(j,2) + ...
        bmod1(9,i)*x1(j,3) + bmod1(10,i)*x1(j,4) + ...
        bmod1(11,i)*x1(j,5) + bmod1(12,i)*x1(j,6) + ...
```

```
   2*bmod1(23,i)*x1(j,1))^2 + (bmod1(3,i) + ...
   bmod1(8,i)*x1(j,1) + bmod1(13,i)*x1(j,3) + ...
   bmod1(14,i)*x1(j,4) + bmod1(15,i)*x1(j,5) + ...
   bmod1(16,i)*x1(j,6) + 2*bmod1(24,i)*x1(j,2))^2 + ...
   (bmod1(4,i) + bmod1(9,i)*x1(j,1) + ...
   bmod1(13,i)*x1(j,2) + bmod1(17,i)*x1(j,4) + ...
   bmod1(18,i)*x1(j,5) + bmod1(19,i)*x1(j,6) + ...
   2*bmod1(25,i)*x1(j,3))^2 + (bmod1(5,i) + ...
   bmod1(10,i)*x1(j,1) + bmod1(14,i)*x1(j,2) + ...
   bmod1(17,i)*x1(j,3) + bmod1(20,i)*x1(j,5) + ...
   bmod1(21,i)*x1(j,6) + 2*bmod1(26,i)*x1(j,4))^2 + ...
   (bmod1(6,i) + bmod1(11,i)*x1(j,1) + ...
   bmod1(15,i)*x1(j,2) + bmod1(18,i)*x1(j,3) + ...
   bmod1(20,i)*x1(j,4) + bmod1(22,i)*x1(j,6) + ...
   2*bmod1(27,i)*x1(j,5))^2 + (bmod1(7,i) + ...
   bmod1(12,i)*x1(j,1) + bmod1(16,i)*x1(j,2) + ...
   bmod1(19,i)*x1(j,3) + bmod1(21,i)*x1(j,4) + ...
   bmod1(22,i)*x1(j,5) + 2*bmod1(28,i)*x1(j,6))^2));

alpha_mod1(j,i) = (pi/2) + (1-gamma)*atan(sqrt( ...
   (bmod1(2,i) + bmod1(8,i)*x1(j,2) + ...
   bmod1(9,i)*x1(j,3) + bmod1(10,i)*x1(j,4) + ...
   bmod1(11,i)*x1(j,5) + bmod1(12,i)*x1(j,6) + ...
   2*bmod1(23,i)*x1(j,1))^2 + (bmod1(3,i) + ...
   bmod1(8,i)*x1(j,1) + bmod1(13,i)*x1(j,3) + ...
   bmod1(14,i)*x1(j,4) + bmod1(15,i)*x1(j,5) + ...
   bmod1(16,i)*x1(j,6) + 2*bmod1(24,i)*x1(j,2))^2 + ...
   (bmod1(4,i) + bmod1(9,i)*x1(j,1) + ...
   bmod1(13,i)*x1(j,2) + bmod1(17,i)*x1(j,4) + ...
   bmod1(18,i)*x1(j,5) + bmod1(19,i)*x1(j,6) + ...
   2*bmod1(25,i)*x1(j,3))^2 + (bmod1(5,i) + ...
   bmod1(10,i)*x1(j,1) + bmod1(14,i)*x1(j,2) + ...
   bmod1(17,i)*x1(j,3) + bmod1(20,i)*x1(j,5) + ...
   bmod1(21,i)*x1(j,6) + 2*bmod1(26,i)*x1(j,4))^2 + ...
   (bmod1(6,i) + bmod1(11,i)*x1(j,1) + ...
   bmod1(15,i)*x1(j,2) + bmod1(18,i)*x1(j,3) + ...
   bmod1(20,i)*x1(j,4) + bmod1(22,i)*x1(j,6) + ...
   2*bmod1(27,i)*x1(j,5))^2 + (bmod1(7,i) + ...
   bmod1(12,i)*x1(j,1) + bmod1(16,i)*x1(j,2) + ...
   bmod1(19,i)*x1(j,3) + bmod1(21,i)*x1(j,4) + ...
   bmod1(22,i)*x1(j,5) + 2*bmod1(28,i)*x1(j,6))^2));

ratio_mod1(j,i) = sin(phi_mod1(j,i))/sin(alpha_mod1(j,i));
```

```
    end

    res_mod1(1:j,i) = (yobs_des1(:,i) - x1m*bmod1(:,i));

    for j = 1:length(x1(:,1))

        dist_mod1(j,i) = ratio_mod1(j,i)*res_mod1(j,i);

    end

end

for j = 1:length(x1(:,1))
    dist_ord1(j,n+1) = 0;

    dist_mod1(j,n+1) = 0;

    dist_orth1(j,n+1) = 0;

    dist_ord1(j,n+2) = mean(dist_ord1(j,1:n));

    dist_mod1(j,n+2) = mean(dist_mod1(j,1:n));

    dist_orth1(j,n+2) = mean(dist_orth1(j,1:n));

    dist_ord1(j,n+3) = var(dist_ord1(j,1:n));

    dist_mod1(j,n+3) = var(dist_mod1(j,1:n));

    dist_orth1(j,n+3) = var(dist_orth1(j,1:n));

    dist_ord1(j,n+4) = dist_ord1(j,n+2)^2 + dist_ord1(j,n+3);

    dist_mod1(j,n+4) = dist_mod1(j,n+2)^2 + dist_mod1(j,n+3);

    dist_orth1(j,n+4) = dist_orth1(j,n+2)^2 + dist_orth1(j,n+3);
end

dist1 = [dist_ord1(1:j,n+2:n+4); 0 0 0; ...
   dist_mod1(1:j,n+2:n+4); 0 0 0; ...
   dist_orth1(1:j,n+2:n+4)];

dist = [dist1];
```

```
for i = 1:n

    for k = 1:N

        for j = 1:length(xconf(:,1))
            cal(i,k).conf(j,:) = ...
               mvnrnd([xconf(j,1) xconf(j,2) xconf(j,3) xconf(j,4) ...
               xconf(j,5) xconf(j,6) yconf(j,1)], sigma);
            yobs(i).conf(j,k) = cal(i,k).conf(j,7);

            cphi_ord1(j,i) = (pi/2) - atan(sqrt( ...
               (bord1(2,i) + bord1(8,i)*xconf(j,2) + ...
               bord1(9,i)*xconf(j,3) + bord1(10,i)*xconf(j,4) + ...
               bord1(11,i)*xconf(j,5) + bord1(12,i)*xconf(j,6) + ...
               2*bord1(23,i)*xconf(j,1))^2 + (bord1(3,i) + ...
               bord1(8,i)*xconf(j,1) + bord1(13,i)*xconf(j,3) + ...
               bord1(14,i)*xconf(j,4) + bord1(15,i)*xconf(j,5) + ...
               bord1(16,i)*xconf(j,6) + ...
               2*bord1(24,i)*xconf(j,2))^2 + (bord1(4,i) + ...
               bord1(9,i)*xconf(j,1) + bord1(13,i)*xconf(j,2) + ...
               bord1(17,i)*xconf(j,4) + bord1(18,i)*xconf(j,5) + ...
               bord1(19,i)*xconf(j,6) + ...
               2*bord1(25,i)*xconf(j,3))^2 + (bord1(5,i) + ...
               bord1(10,i)*xconf(j,1) + bord1(14,i)*xconf(j,2) + ...
               bord1(17,i)*xconf(j,3) + bord1(20,i)*xconf(j,5) + ...
               bord1(21,i)*xconf(j,6) + ...
               2*bord1(26,i)*xconf(j,4))^2 + (bord1(6,i) + ...
               bord1(11,i)*xconf(j,1) + bord1(15,i)*xconf(j,2) + ...
               bord1(18,i)*xconf(j,3) + bord1(20,i)*xconf(j,4) + ...
               bord1(22,i)*xconf(j,6) + ...
               2*bord1(27,i)*xconf(j,5))^2 + (bord1(7,i) + ...
               bord1(12,i)*xconf(j,1) + bord1(16,i)*xconf(j,2) + ...
               bord1(19,i)*xconf(j,3) + bord1(21,i)*xconf(j,4) + ...
               bord1(22,i)*xconf(j,5) + 2*bord1(28,i)*xconf(j,6))^2));

            cphi_orth1(j,i) = (pi/2) - atan(sqrt( ...
               (borth1(2,i) + borth1(8,i)*xconf(j,2) + ...
               borth1(9,i)*xconf(j,3) + borth1(10,i)*xconf(j,4) + ...
               borth1(11,i)*xconf(j,5) + borth1(12,i)*xconf(j,6) + ...
               2*borth1(23,i)*xconf(j,1))^2 + (borth1(3,i) + ...
               borth1(8,i)*xconf(j,1) + borth1(13,i)*xconf(j,3) + ...
               borth1(14,i)*xconf(j,4) + borth1(15,i)*xconf(j,5) + ...
               borth1(16,i)*xconf(j,6) + ...
               2*borth1(24,i)*xconf(j,2))^2 + (borth1(4,i) + ...
```

```
   borth1(9,i)*xconf(j,1) + borth1(13,i)*xconf(j,2) + ...
   borth1(17,i)*xconf(j,4) + borth1(18,i)*xconf(j,5) + ...
   borth1(19,i)*xconf(j,6) + ...
   2*borth1(25,i)*xconf(j,3))^2 + (borth1(5,i) + ...
   borth1(10,i)*xconf(j,1) + borth1(14,i)*xconf(j,2) + ...
   borth1(17,i)*xconf(j,3) + borth1(20,i)*xconf(j,5) + ...
   borth1(21,i)*xconf(j,6) + ...
   2*borth1(26,i)*xconf(j,4))^2 + (borth1(6,i) + ...
   borth1(11,i)*xconf(j,1) + borth1(15,i)*xconf(j,2) + ...
   borth1(18,i)*xconf(j,3) + borth1(20,i)*xconf(j,4) + ...
   borth1(22,i)*xconf(j,6) + ...
   2*borth1(27,i)*xconf(j,5))^2 + (borth1(7,i) + ...
   borth1(12,i)*xconf(j,1) + borth1(16,i)*xconf(j,2) + ...
   borth1(19,i)*xconf(j,3) + borth1(21,i)*xconf(j,4) + ...
   borth1(22,i)*xconf(j,5) + ...
   2*borth1(28,i)*xconf(j,6))^2));

cphi_mod1(j,i) = (pi/2) - atan(sqrt( ...
   (bmod1(2,i) + bmod1(8,i)*xconf(j,2) + ...
   bmod1(9,i)*xconf(j,3) + bmod1(10,i)*xconf(j,4) + ...
   bmod1(11,i)*xconf(j,5) + bmod1(12,i)*xconf(j,6) + ...
   2*bmod1(23,i)*xconf(j,1))^2 + (bmod1(3,i) + ...
   bmod1(8,i)*xconf(j,1) + bmod1(13,i)*xconf(j,3) + ...
   bmod1(14,i)*xconf(j,4) + bmod1(15,i)*xconf(j,5) + ...
   bmod1(16,i)*xconf(j,6) + ...
   2*bmod1(24,i)*xconf(j,2))^2 + (bmod1(4,i) + ...
   bmod1(9,i)*xconf(j,1) + bmod1(13,i)*xconf(j,2) + ...
   bmod1(17,i)*xconf(j,4) + bmod1(18,i)*xconf(j,5) + ...
   bmod1(19,i)*xconf(j,6) + ...
   2*bmod1(25,i)*xconf(j,3))^2 + (bmod1(5,i) + ...
   bmod1(10,i)*xconf(j,1) + bmod1(14,i)*xconf(j,2) + ...
   bmod1(17,i)*xconf(j,3) + bmod1(20,i)*xconf(j,5) + ...
   bmod1(21,i)*xconf(j,6) + ...
   2*bmod1(26,i)*xconf(j,4))^2 + (bmod1(6,i) + ...
   bmod1(11,i)*xconf(j,1) + bmod1(15,i)*xconf(j,2) + ...
   bmod1(18,i)*xconf(j,3) + bmod1(20,i)*xconf(j,4) + ...
   bmod1(22,i)*xconf(j,6) + ...
   2*bmod1(27,i)*xconf(j,5))^2 + (bmod1(7,i) + ...
   bmod1(12,i)*xconf(j,1) + bmod1(16,i)*xconf(j,2) + ...
   bmod1(19,i)*xconf(j,3) + bmod1(21,i)*xconf(j,4) + ...
   bmod1(22,i)*xconf(j,5) + ...
   2*bmod1(28,i)*xconf(j,6))^2));

calpha_ord1(j,i) = (pi/2) - (gamma-1)*(atan(sqrt( ...
```

```
   (bord1(2,i) + bord1(8,i)*xconf(j,2) + ...
   bord1(9,i)*xconf(j,3) + bord1(10,i)*xconf(j,4) + ...
   bord1(11,i)*xconf(j,5) + bord1(12,i)*xconf(j,6) + ...
   2*bord1(23,i)*xconf(j,1))^2 + (bord1(3,i) + ...
   bord1(8,i)*xconf(j,1) + bord1(13,i)*xconf(j,3) + ...
   bord1(14,i)*xconf(j,4) + bord1(15,i)*xconf(j,5) + ...
   bord1(16,i)*xconf(j,6) + ...
   2*bord1(24,i)*xconf(j,2))^2 + (bord1(4,i) + ...
   bord1(9,i)*xconf(j,1) + bord1(13,i)*xconf(j,2) + ...
   bord1(17,i)*xconf(j,4) + bord1(18,i)*xconf(j,5) + ...
   bord1(19,i)*xconf(j,6) + ...
   2*bord1(25,i)*xconf(j,3))^2 + (bord1(5,i) + ...
   bord1(10,i)*xconf(j,1) + bord1(14,i)*xconf(j,2) + ...
   bord1(17,i)*xconf(j,3) + bord1(20,i)*xconf(j,5) + ...
   bord1(21,i)*xconf(j,6) + ...
   2*bord1(26,i)*xconf(j,4))^2 + (bord1(6,i) + ...
   bord1(11,i)*xconf(j,1) + bord1(15,i)*xconf(j,2) + ...
   bord1(18,i)*xconf(j,3) + bord1(20,i)*xconf(j,4) + ...
   bord1(22,i)*xconf(j,6) + ...
   2*bord1(27,i)*xconf(j,5))^2 + (bord1(7,i) + ...
   bord1(12,i)*xconf(j,1) + bord1(16,i)*xconf(j,2) + ...
   bord1(19,i)*xconf(j,3) + bord1(21,i)*xconf(j,4) + ...
   bord1(22,i)*xconf(j,5) + ...
   2*bord1(28,i)*xconf(j,6))^2)));

calpha_mod1(j,i) = (pi/2) - (gamma-1)*(atan(sqrt( ...
   (bmod1(2,i) + bmod1(8,i)*xconf(j,2) + ...
   bmod1(9,i)*xconf(j,3) + bmod1(10,i)*xconf(j,4) + ...
   bmod1(11,i)*xconf(j,5) + bmod1(12,i)*xconf(j,6) + ...
   2*bmod1(23,i)*xconf(j,1))^2 + (bmod1(3,i) + ...
   bmod1(8,i)*xconf(j,1) + bmod1(13,i)*xconf(j,3) + ...
   bmod1(14,i)*xconf(j,4) + ...
   bmod1(15,i)*xconf(j,5) + bmod1(16,i)*xconf(j,6) + ...
   2*bmod1(24,i)*xconf(j,2))^2 + (bmod1(4,i) + ...
   bmod1(9,i)*xconf(j,1) + bmod1(13,i)*xconf(j,2) + ...
   bmod1(17,i)*xconf(j,4) + bmod1(18,i)*xconf(j,5) + ...
   bmod1(19,i)*xconf(j,6) + ...
   2*bmod1(25,i)*xconf(j,3))^2 + (bmod1(5,i) + ...
   bmod1(10,i)*xconf(j,1) + bmod1(14,i)*xconf(j,2) + ...
   bmod1(17,i)*xconf(j,3) + bmod1(20,i)*xconf(j,5) + ...
   bmod1(21,i)*xconf(j,6) + ...
   2*bmod1(26,i)*xconf(j,4))^2 + (bmod1(6,i) + ...
   bmod1(11,i)*xconf(j,1) + bmod1(15,i)*xconf(j,2) + ...
   bmod1(18,i)*xconf(j,3) + bmod1(20,i)*xconf(j,4) + ...
```

```
      bmod1(22,i)*xconf(j,6) + ...
      2*bmod1(27,i)*xconf(j,5))^2 + (bmod1(7,i) + ...
      bmod1(12,i)*xconf(j,1) + bmod1(16,i)*xconf(j,2) + ...
      bmod1(19,i)*xconf(j,3) + bmod1(21,i)*xconf(j,4) + ...
      bmod1(22,i)*xconf(j,5) + ...
      2*bmod1(28,i)*xconf(j,6))^2)));

   calpha_orth1(j,i) = (pi/2) - (gamma-1)*(atan(sqrt( ...
      (borth1(2,i) + borth1(8,i)*xconf(j,2) + ...
      borth1(9,i)*xconf(j,3) + borth1(10,i)*xconf(j,4) + ...
      borth1(11,i)*xconf(j,5) + borth1(12,i)*xconf(j,6) + ...
      2*borth1(23,i)*xconf(j,1))^2 + (borth1(3,i) + ...
      borth1(8,i)*xconf(j,1) + borth1(13,i)*xconf(j,3) + ...
      borth1(14,i)*xconf(j,4) + borth1(15,i)*xconf(j,5) + ...
      borth1(16,i)*xconf(j,6) + ...
      2*borth1(24,i)*xconf(j,2))^2 + (borth1(4,i) + ...
      borth1(9,i)*xconf(j,1) + borth1(13,i)*xconf(j,2) + ...
      borth1(17,i)*xconf(j,4) + borth1(18,i)*xconf(j,5) + ...
      borth1(19,i)*xconf(j,6) + ...
      2*borth1(25,i)*xconf(j,3))^2 + (borth1(5,i) + ...
      borth1(10,i)*xconf(j,1) + borth1(14,i)*xconf(j,2) + ...
      borth1(17,i)*xconf(j,3) + borth1(20,i)*xconf(j,5) + ...
      borth1(21,i)*xconf(j,6) + ...
      2*borth1(26,i)*xconf(j,4))^2 + (borth1(6,i) + ...
      borth1(11,i)*xconf(j,1) + borth1(15,i)*xconf(j,2) + ...
      borth1(18,i)*xconf(j,3) + borth1(20,i)*xconf(j,4) + ...
      borth1(22,i)*xconf(j,6) + ...
      2*borth1(27,i)*xconf(j,5))^2 + (borth1(7,i) + ...
      borth1(12,i)*xconf(j,1) + borth1(16,i)*xconf(j,2) + ...
      borth1(19,i)*xconf(j,3) + borth1(21,i)*xconf(j,4) + ...
      borth1(22,i)*xconf(j,5) + ...
      2*borth1(28,i)*xconf(j,6))^2)));

   cratio_ord1(j,k) = sin(cphi_ord1(j,i))/...
      sin(calpha_ord1(j,i));

   cratio_mod1(j,k) = sin(cphi_mod1(j,i))/...
      sin(calpha_mod1(j,i));

   cratio_orth1(j,k) = sin(cphi_orth1(j,i))/...
      sin(calpha_orth1(j,i));

end
```

```
    cres(i).ord1(1:j,k) = (yobs(i).conf(:,k) - ...
       xconfm*bord1(:,i));

    cres(i).mod1(1:j,k) = (yobs(i).conf(:,k) - ...
       xconfm*bmod1(:,i));

    cres(i).orth1(1:j,k) = (yobs(i).conf(:,k) - ...
       xconfm*borth1(:,i));

    for j = 1:length(xconf(:,1))

        cdist(i).ord1(j,i) = ...
           ratio_ord1(j,i)*cres(i).ord1(j,k);

        cdist(i).mod1(j,i) = ...
           ratio_mod1(j,i)*cres(i).mod1(j,k);

        cdist(i).orth1(j,i) = ...
           ratio_orth1(j,i)*cres(i).orth1(j,k);

    end

end

for j = 1:38
    cdist(i).ord1(j,N+1) = 0;

    cdist(i).mod1(j,N+1) = 0;

    cdist(i).orth1(j,N+1) = 0;

    cdist(i).ord1(j,N+2) = mean(cdist(i).ord1(j,1:N));

    cdist(i).mod1(j,N+2) = mean(cdist(i).mod1(j,1:N));

    cdist(i).orth1(j,N+2) = mean(cdist(i).orth1(j,1:N));

    cdist(i).ord1(j,N+3) = var(cdist(i).ord1(j,1:N));

    cdist(i).mod1(j,N+3) = var(cdist(i).mod1(j,1:N));

    cdist(i).orth1(j,N+3) = var(cdist(i).orth1(j,1:N));

end
```

```
    for j = 1:38
        ccdist_ord1(j,i) = cdist(i).ord1(j,N+2);

        ccdist_mod1(j,i) = cdist(i).mod1(j,N+2);

        ccdist_orth1(j,i) = cdist(i).orth1(j,N+2);
    end

end

for j = 1:38
    ccdist_ord1(j,n+1) = 0;

    ccdist_mod1(j,n+1) = 0;

    ccdist_orth1(j,n+1) = 0;

    ccdist_ord1(j,n+2) = mean(ccdist_ord1(j,1:n));

    ccdist_mod1(j,n+2) = mean(ccdist_mod1(j,1:n));

    ccdist_orth1(j,n+2) = mean(ccdist_orth1(j,1:n));

    ccdist_ord1(j,n+3) = var(ccdist_ord1(j,1:n));

    ccdist_mod1(j,n+3) = var(ccdist_mod1(j,1:n));

    ccdist_orth1(j,n+3) = var(ccdist_orth1(j,1:n));

    ccdist_ord1(j,n+4) = ccdist_ord1(j,n+2)^2 + ccdist_ord1(j,n+3);

    ccdist_mod1(j,n+4) = ccdist_mod1(j,n+2)^2 + ccdist_mod1(j,n+3);

    ccdist_orth1(j,n+4) = ccdist_orth1(j,n+2)^2 + ccdist_orth1(j,n+3);
end

ccdist1 = [ccdist_ord1(1:j,n+2:n+4); 0 0 0; ...
   ccdist_mod1(1:j,n+2:n+4); 0 0 0; ...
   ccdist_orth1(1:j,n+2:n+4)];

ccdist = [ccdist1];

%% Mean and Variance of Coefficients over n Calibrations for
```

```
%% each Estimator and Design

bmean = [mean(bord1(1,:)) mean(bmod1(1,:)) mean(borth1(1,:)); ...
    mean(bord1(2,:)) mean(bmod1(2,:)) mean(borth1(2,:)); ...
    mean(bord1(3,:)) mean(bmod1(3,:)) mean(borth1(3,:)); ...
    mean(bord1(4,:)) mean(bmod1(4,:)) mean(borth1(4,:)); ...
    mean(bord1(5,:)) mean(bmod1(5,:)) mean(borth1(5,:)); ...
    mean(bord1(6,:)) mean(bmod1(6,:)) mean(borth1(6,:)); ...
    mean(bord1(7,:)) mean(bmod1(7,:)) mean(borth1(7,:)); ...
    mean(bord1(8,:)) mean(bmod1(8,:)) mean(borth1(8,:)); ...
    mean(bord1(9,:)) mean(bmod1(9,:)) mean(borth1(9,:)); ...
    mean(bord1(10,:)) mean(bmod1(10,:)) mean(borth1(10,:)); ...
    mean(bord1(11,:)) mean(bmod1(11,:)) mean(borth1(11,:)); ...
    mean(bord1(12,:)) mean(bmod1(12,:)) mean(borth1(12,:)); ...
    mean(bord1(13,:)) mean(bmod1(13,:)) mean(borth1(13,:)); ...
    mean(bord1(14,:)) mean(bmod1(14,:)) mean(borth1(14,:)); ...
    mean(bord1(15,:)) mean(bmod1(15,:)) mean(borth1(15,:)); ...
    mean(bord1(16,:)) mean(bmod1(16,:)) mean(borth1(16,:)); ...
    mean(bord1(17,:)) mean(bmod1(17,:)) mean(borth1(17,:)); ...
    mean(bord1(18,:)) mean(bmod1(18,:)) mean(borth1(18,:)); ...
    mean(bord1(19,:)) mean(bmod1(19,:)) mean(borth1(19,:)); ...
    mean(bord1(20,:)) mean(bmod1(20,:)) mean(borth1(20,:)); ...
    mean(bord1(21,:)) mean(bmod1(21,:)) mean(borth1(21,:)); ...
    mean(bord1(22,:)) mean(bmod1(22,:)) mean(borth1(22,:)); ...
    mean(bord1(23,:)) mean(bmod1(23,:)) mean(borth1(23,:)); ...
    mean(bord1(24,:)) mean(bmod1(24,:)) mean(borth1(24,:)); ...
    mean(bord1(25,:)) mean(bmod1(25,:)) mean(borth1(25,:)); ...
    mean(bord1(26,:)) mean(bmod1(26,:)) mean(borth1(26,:)); ...
    mean(bord1(27,:)) mean(bmod1(27,:)) mean(borth1(27,:)); ...
    mean(bord1(28,:)) mean(bmod1(28,:)) mean(borth1(28,:))];

bvar = [var(bord1(1,:)) var(bmod1(1,:)) var(borth1(1,:)); ...
    var(bord1(2,:)) var(bmod1(2,:)) var(borth1(2,:)); ...
    var(bord1(3,:)) var(bmod1(3,:)) var(borth1(3,:)); ...
    var(bord1(4,:)) var(bmod1(4,:)) var(borth1(4,:)); ...
    var(bord1(5,:)) var(bmod1(5,:)) var(borth1(5,:)); ...
    var(bord1(6,:)) var(bmod1(6,:)) var(borth1(6,:)); ...
    var(bord1(7,:)) var(bmod1(7,:)) var(borth1(7,:)); ...
    var(bord1(8,:)) var(bmod1(8,:)) var(borth1(8,:)); ...
    var(bord1(9,:)) var(bmod1(9,:)) var(borth1(9,:)); ...
    var(bord1(10,:)) var(bmod1(10,:)) var(borth1(10,:)); ...
    var(bord1(11,:)) var(bmod1(11,:)) var(borth1(11,:)); ...
    var(bord1(12,:)) var(bmod1(12,:)) var(borth1(12,:)); ...
    var(bord1(13,:)) var(bmod1(13,:)) var(borth1(13,:)); ...
```

```
    var(bord1(14,:)) var(bmod1(14,:)) var(borth1(14,:)); ...
    var(bord1(15,:)) var(bmod1(15,:)) var(borth1(15,:)); ...
    var(bord1(16,:)) var(bmod1(16,:)) var(borth1(16,:)); ...
    var(bord1(17,:)) var(bmod1(17,:)) var(borth1(17,:)); ...
    var(bord1(18,:)) var(bmod1(18,:)) var(borth1(18,:)); ...
    var(bord1(19,:)) var(bmod1(19,:)) var(borth1(19,:)); ...
    var(bord1(20,:)) var(bmod1(20,:)) var(borth1(20,:)); ...
    var(bord1(21,:)) var(bmod1(21,:)) var(borth1(21,:)); ...
    var(bord1(22,:)) var(bmod1(22,:)) var(borth1(22,:)); ...
    var(bord1(23,:)) var(bmod1(23,:)) var(borth1(23,:)); ...
    var(bord1(24,:)) var(bmod1(24,:)) var(borth1(24,:)); ...
    var(bord1(25,:)) var(bmod1(25,:)) var(borth1(25,:)); ...
    var(bord1(26,:)) var(bmod1(26,:)) var(borth1(26,:)); ...
    var(bord1(27,:)) var(bmod1(27,:)) var(borth1(27,:)); ...
    var(bord1(28,:)) var(bmod1(28,:)) var(borth1(28,:))];

t = toc;
tmin = t/60;
disp(['Elapsed time is ', num2str(tmin), ' minutes.']);
```

VITA

Sean A. Commo
Department of Mechanical and Aerospace Engineering
Old Dominion University
Norfolk, VA 23529

Sean began his work with NASA in 2008 as a graduate research assistant under Dr. Peter Parker working on calibration of pressure measurement systems for space applications. In 2009, he joined NASA under the Cooperative Education Program and received a permanent appointment in 2010 while continuing research in the area of planetary entry, descent, and landing. In addition, he has worked on projects in force measurement systems and aeronautics research and he is a member of the NASA Langley Statistical Engineering Team. Prior to his work with NASA, Sean spent two years working at the ODU-operated Langley Full-Scale Tunnel. He received a Bachelors degree in Mechanical Engineering and Masters degree in Aerospace Engineering from Old Dominion University. Currently, he has published two journal articles and been invited to speak at conferences within the aerospace and statistics community. His research interest areas include design of experiments, response surface methodology, and measurement system characterization methods and techniques.

Typeset using LaTeX.