Data Driven Confidence Regions for Cost Estimating Relationships

Christopher Jarvis
USAF AFLCMC EGOL/FZC
christopher.jarvis.3@us.af.mil

May 2, 2017

Abstract

Typically, cost estimating models contain many uncertain parameters which drive the model output results. To characterize the model output distributions, Monte Carlo methods are employed, which require accurate description of model parameter distributions and dependencies. In this paper we review the techniques to compute the parameter confidence regions for linear and nonlinear regression methods that can be used for this purpose. Finally, comments are made regarding the current practical implementations and limitations.

1 Introduction

When building models in cost estimating as in many other scientific fields, an analyst begins with observed data and an assumed model form that governs the relationship of the independent and dependent variables. These models contain parameters that often represent specific physical properties and the model form of equations can be dictated by underlying dynamical processes. The desire is to determine the value of the parameters associated with the assumed model that provide the best fit in some measure between the model predictions and the observed data.

Regression analysis, which is a ubiquitous tool for parameter estimation, identifies the parameters that minimize the sum of squared errors between the observed and predicted data points. In theory, there exist population parameters for the model type that determine all possible observations up to the random error terms. Since the observed data is only a subset of the total population, the computed parameters are reflective of the sample and are not necessarily the true population parameters. Thus, given a set of parameters the secondary task is to quantify the uncertainty associated with the parameters computed from the sample data. Furthermore, in cost estimating applications, a model that encompasses all costs for a program may contain many uncertain parameters. The uncertainty of the model outputs can be assessed using a Monte Carlo simulation to numerically sample the inputs and capture the output distributions. The Monte Carlo process and the accuracy of the results relies on quantification of the parameter distributions and dependencies between them. Typically, distributions for all parameters are one dimensional distributions and some measure of correlation is used to force pairwise relationships between parameter inputs. This process may not accurately capture the underlying multidimensional distribution of parameters and could consequently produce incorrect output distributions and ultimately misleading results.

The purpose of this paper is to review the methods for finding the parameter confidence regions for linear and nonlinear regression models. The methods will be applied to equations typical in cost

estimating applications. To support that goal, in section 2, an overview of nonlinear regression is presented. Joint confidence region computations are developed in section 3. In section 4, the cost estimating relationship models are introduced as well as numerical results. Finally, in section 5, the results of this paper, and some potential ideas for future work are summarized.

2 Review of Nonlinear Regression

To facilitate the development of the later equations a review of nonlinear regression is presented using a matrix approach. For additional introduction and results see [1, 2]. The general form for all models considered here is

$$y_i = f(\mathbf{x}_i, \boldsymbol{\beta}) + \varepsilon_i \tag{1}$$

where $\mathbf{x}_i$ is a vector of variables from $\mathbb{R}^N$, $\boldsymbol{\beta} = (\beta_0, ..., \beta_{p-1})^T$ is a vector of parameters from $\mathbb{R}^p$, and $f(\mathbf{x}_i, \boldsymbol{\beta})$ is the function relating the input variables to the output data y_i using the parameters $\boldsymbol{\beta}$ and a random error term ε_i.

When the function $f(\mathbf{x}_i, \boldsymbol{\beta})$ is linear in $\boldsymbol{\beta}$, the model is called linear. In some cases, the function is not linear in $\boldsymbol{\beta}$, but can be made linear through a nonlinear transformation. Applying a logarithm to both the input variables and possibly the output data is one common transformation. When the function is linear or transformably linear in the parameters, the Ordinary Least Squares (OLS) regression to find the parameter values. When the function $f(\mathbf{x}_i, \boldsymbol{\beta})$ is nonlinear in $\boldsymbol{\beta}$ or has no linearizing transformation an approach other than OLS must be used to solve for the parameter vector $\boldsymbol{\beta}$. The most common approach of Nonlinear Regression (NLR) involves iterated linearization. The linearization process as the name suggests creates a linear model that under the right circumstances closely approximates the nonlinear model around the specified point. The process is described below.

To find a least squares solution, recall the error equation is given by the difference between the observed data y_i and the model predictions $\hat{y}_i$, referred to as the fitted data, which are based on the parameter values. Hence, the error equation can be thought of a function of the independent variables and the parameter values given by

$$\begin{aligned} \varepsilon_i(\mathbf{x}_i, \boldsymbol{\beta}) &= y_i - \hat{y}_i \\ &= y_i - f(\mathbf{x}_i, \boldsymbol{\beta}) \end{aligned}$$

and the total sum of squared errors is

$$S(\mathbf{x}_i, \boldsymbol{\beta}) = \sum_{i=1}^{N} (\varepsilon_i(\mathbf{x}_i, \boldsymbol{\beta}))^2 \tag{2}$$

$$= \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i, \boldsymbol{\beta}))^2 . \tag{3}$$

Assuming that $f(\mathbf{x}_i, \boldsymbol{\beta})$ is sufficiently regular, we can expand $f(\mathbf{x}_i, \boldsymbol{\beta})$ about a point $\boldsymbol{\beta}^{(k)}$ in the parameter space using a Taylor series expansion as

$$f(\mathbf{x}_i, \boldsymbol{\beta}) = f\left(\mathbf{x}_i, \boldsymbol{\beta}^{(k)}\right) + \sum_{j=0}^{p-1} \left[\frac{\partial f(\mathbf{x}_i, \boldsymbol{\beta})}{\partial \beta_j}\right]_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(k)}} \left(\beta_j - \beta_j^{(k)}\right) + LTE$$

where LTE is the local truncation error associated with the first order Taylor Series approximation. The term $\boldsymbol{\beta}^{(k)}$ in this context represents an iterative approximation for the true parameters. Using

the Taylor series approximation above and neglecting the higher order terms, $f(\mathbf{x}_i, \boldsymbol{\beta})$ can be locally approximated as a linear function in $\boldsymbol{\beta}$ and (1) can be written using matrix operations as

$$\begin{aligned} y_i &\approx f\left(\mathbf{x}_i, \boldsymbol{\beta}^{(k)}\right) + \sum_{j=0}^{p-1} \left[\frac{\partial f(\mathbf{x}_i, \boldsymbol{\beta})}{\partial \beta_j}\right]_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(k)}} \left(\beta_j - \beta_j^{(k)}\right) + \varepsilon_i \\ &= f\left(\mathbf{x}_i, \boldsymbol{\beta}^{(k)}\right) + \left[\frac{\partial f(\mathbf{x}_i,\boldsymbol{\beta})}{\partial \beta_0} \quad \frac{\partial f(\mathbf{x}_i,\boldsymbol{\beta})}{\partial \beta_1} \quad \cdots \quad \frac{\partial f(\mathbf{x}_i,\boldsymbol{\beta})}{\partial \beta_{p-1}}\right]_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(k)}} \begin{bmatrix} \beta_0 - \beta_0^{(k)} \\ \beta_1 - \beta_1^{(k)} \\ \vdots \\ \beta_{p-1} - \beta_{p-1}^{(k)} \end{bmatrix} + \varepsilon_i. \end{aligned} \tag{4}$$

Stacking the equations for each of the data points y_i we obtain the matrix system equation given by

$$\mathbf{Y} = \mathbf{F}^{(k)} + \mathbf{D}^{(k)}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}\right) \tag{5}$$

or equivalently

$$\mathbf{D}^{(k)}\left(\boldsymbol{\beta} - \boldsymbol{\beta}^{(k)}\right) = \left(\mathbf{Y} - \mathbf{F}^{(k)}\right) \tag{6}$$

where the error terms are included in the data vector $\mathbf{Y}$ and the model approximation $\mathbf{F}$ and the matrix $\mathbf{D}$ of partial derivatives called the Jacobian are given by

$$\mathbf{F}^{(k)} = \begin{bmatrix} f\left(\mathbf{x}_1, \boldsymbol{\beta}^{(k)}\right) \\ f\left(\mathbf{x}_2, \boldsymbol{\beta}^{(k)}\right) \\ \vdots \\ f\left(\mathbf{x}_N, \boldsymbol{\beta}^{(k)}\right) \end{bmatrix} \qquad \mathbf{D}^{(k)} = \begin{bmatrix} \frac{\partial f(\mathbf{x}_1,\boldsymbol{\beta})}{\partial \beta_k} & \frac{\partial f(\mathbf{x}_1,\boldsymbol{\beta})}{\partial \beta_1} & \cdots & \frac{\partial f(\mathbf{x}_1,\boldsymbol{\beta})}{\partial \beta_{p-1}} \\ \frac{\partial f(\mathbf{x}_2,\boldsymbol{\beta})}{\partial \beta_k} & \frac{\partial f(\mathbf{x}_2,\boldsymbol{\beta})}{\partial \beta_1} & \cdots & \frac{\partial f(\mathbf{x}_2,\boldsymbol{\beta})}{\partial \beta_{p-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(\mathbf{x}_N,\boldsymbol{\beta})}{\partial \beta_k} & \frac{\partial f(\mathbf{x}_N,\boldsymbol{\beta})}{\partial \beta_1} & \cdots & \frac{\partial f(\mathbf{x}_N,\boldsymbol{\beta})}{\partial \beta_{p-1}} \end{bmatrix}_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(k)}}.$$

When the number of data points is not equal to the number of parameters i.e. $N \neq p$, then the $\mathbf{D}$ matrix is non-square, and hence not invertible. When there are less data points than parameters ($N < p$), the system (5) does not have a unique parameter solution but rather an infinite number of solutions through a $p - N$ dimensional subspace of $\mathbb{R}^p$. When there are more data points than parameters ($N > p$), the system is overdetermined and has either a single solution (in the case of perfect data) or does not have a solution. When there are equal numbers of distinct data points and parameters the system containing the error is solved exactly. The least squares solution minimizes the sum of squared errors and gives the exact solution, if it exists. In practical applications, for this reason, it is typical to have more data points than parameters with the hope that the least squares solution will provide a better approximation to the true population parameters by reducing the influence of the error.

A mathematical generalization of the matrix inverse is called the pseudoinverse. Given an $[n \times m]$ matrix $\mathbf{A}$, the pseudoinverse of $\mathbf{A}$ is defined by

$$\mathbf{A}^{\dagger} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$$

where $\mathbf{A}^T$ is the transpose of the matrix. Then, for the overdetermined system

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{z}$$

the least squares solution using the pseudoinverse is given by

$$\boldsymbol{x} = \mathbf{A}^{\dagger}\boldsymbol{z}. \tag{7}$$

To understand how the pseudoinverse gives the least squares solution note the residual (error) term for any vector $\boldsymbol{x}$ is given by

$$\boldsymbol{\varepsilon} = \boldsymbol{z} - \mathbf{A}\boldsymbol{x}$$

and the sum of squared errors is

$$\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = (\boldsymbol{z} - \mathbf{A}\boldsymbol{x})^T (\boldsymbol{z} - \mathbf{A}\boldsymbol{x}) .$$

The least squares solution minimizes the sum of squared errors equation yielding a zero derivative,

$$\begin{aligned} 0 &= \frac{\partial}{\partial \boldsymbol{x}} \left[\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}\right] \\ &= -2\mathbf{A}^T (\boldsymbol{z} - \mathbf{A}\boldsymbol{x}) . \end{aligned} \tag{8}$$

Rearranging (8) we have

$$\mathbf{A}^T\mathbf{A}\boldsymbol{x} = \mathbf{A}^T\boldsymbol{z}$$

which is the matrix form of the normal equations. The term on the left side $\mathbf{A}^T\mathbf{A}$ is a square matrix and for at least m unique data points, is invertible. The solution is given by

$$\boldsymbol{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1} \mathbf{A}^T\boldsymbol{z}$$

which agrees with (7).

Using the pseudoinverse, the system linearized about the point $\boldsymbol{\beta}^{(k)}$ in (6) can be solved for a correction vector

$$\boldsymbol{\delta}^{(k)} = \left[\mathbf{D}^{(k)}\right]^{\dagger} \left(\mathbf{Y} - \mathbf{F}^{(k)}\right) . \tag{9}$$

Therefore, starting from an initial guess $\boldsymbol{\beta}^{(k)}$, we find a new estimate $\boldsymbol{\beta}^{(k+1)}$ by computing

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \boldsymbol{\delta}^{(k)}. \tag{10}$$

This process is known as the Gauss-Newton step and is repeated until the termination criteria are met. A slight but common modification adds a scaling parameter $\gamma \in (0, 1]$ to the correction vector so that (10) becomes

$$\begin{aligned} \boldsymbol{\beta}^{(k+1)} &= \boldsymbol{\beta}^{(k)} + \gamma\boldsymbol{\delta}^{(k)} \\ &= \boldsymbol{\beta}^{(k)} + \gamma \left(\left[\mathbf{D}^{(k)}\right]^{\dagger} \left(\mathbf{Y} - \mathbf{F}^{(k)}\right) \right) . \end{aligned} \tag{11}$$

The correction vector is computed from the linearization around the point $\boldsymbol{\beta}^{(k)}$ and using a full correction step can overshoot true solution or produce convergence issues. The scaling parameter allows for partial steps in the right direction and can increase the stability of the overall optimization process, potentially at the cost of additional iterations required to reach a given tolerance.

When the problem is linear in the parameters the Jacobian matrix is constant and the parameter vector can be found after one full step iteration. Here, the matrix $\mathbf{D}$ is the traditional data (or design) matrix referred to in literature as $\mathbf{X}$ [1]. Consequently, we have derived the well known result that the maximum likelihood unbiased estimator $\boldsymbol{b}$ of the true parameter vector $\boldsymbol{\beta}$ is given by

$$\boldsymbol{b} = \mathbf{D}^{\dagger}\mathbf{Y}. \tag{12}$$

Additionally we reference [1] that for the linear case

$$\boldsymbol{b} \sim N\left(\boldsymbol{\beta}, \sigma^2 \left(\mathbf{D}^T\mathbf{D}\right)^{-1}\right) . \tag{13}$$

3 Parameter Confidence Regions for Regression Problems

From (13) we have that the parameter estimates themselves are random variables with a distribution which depend on the error terms. What may not be as obvious is that the estimates also depend on the model and data sample points. Specifically, for the inverse of $\mathbf{D}^T\mathbf{D}$ to exist we require that $\mathbf{D}$ have full column rank. Even if $\mathbf{D}^T\mathbf{D}$ is non-singular, the condition number may be large enough to yield untrustworthy results when computing the population parameter estimates. Ultimately, we are interested not only the parameter values, but their uncertainty characterization which is influenced by the data error and sampling.

For a single parameter of interest, confidence intervals are typical to describe the uncertainty of the parameter estimate. These can be seen in a parameter margin of error, e.g. $\pm 3\%$ or in a range, e.g. $\mu \in [7.4, 7.6]$. When considering the multiple parameters a one dimensional interval is no longer appropriate. The range of reasonable values is a multidimensional set, for linear problems an ellipse for 2 parameters or a hyperellipsoid for more than 2 parameters. Simply stated, for a given confidence coefficient α the confidence region for a parameter θ is the set $CR \in \mathbb{R}^p$ such that

$$Pr\left(\theta \in CR\right) < \left(1 - \alpha\right). \tag{14}$$

As stated above, this set CR depends on the model characteristics and particular sample. The true population parameter vector is a constant and for any confidence region, is either contained in the region or not. So the confidence region must be interpreted as a statement regarding the probability that the process and sample can reliably create a region that would contain the true population parameter with probability $(1 - \alpha)$. For a single sample this can be restated to say that the confidence region is the set of all values such that if any were the true population parameter, it would not be statistically different at the α level of confidence [3].

More formally, the confidence region is defined [1, 2] as the set of all $\tilde{\boldsymbol{\beta}}$ such that

$$S\left(\mathbf{x}_i, \tilde{\boldsymbol{\beta}}\right) \leq S\left(\mathbf{x}_i, \boldsymbol{b}\right)\left(1 + \frac{p}{N-p} F\left(p, N-p, 1-\alpha\right)\right) \tag{15}$$

where $S\left(\mathbf{x}_i, \tilde{\boldsymbol{\beta}}\right)$ is defined as in (2) and $F\left(p, N-p, 1-\alpha\right)$ is the Fisher distribution. After some rearranging (15) can be expressed as

$$\left(S\left(\mathbf{x}_i, \tilde{\boldsymbol{\beta}}\right) - S\left(\mathbf{x}_i, \boldsymbol{b}\right)\right) \leq p s^2 F\left(p, N-p, 1-\alpha\right) \tag{16}$$

where we have only recognized the term $s^2 = S\left(\mathbf{x}_i, \boldsymbol{b}\right)/(N-p)$ as the mean squared error of the estimate. From these definitions, determining the boundaries of the confidence region may require the evaluation of the model at a potentially significant number of points in the parameter space. In the case of a linear model, by taking advantage of the constant Jacobian we can determine the confidence region after solving the linear regression without the need for sampling. Given the parameter estimate $\boldsymbol{b}$ from the linear regression, the evaluated model vector $\mathbf{F}\left(\mathbf{x}, \boldsymbol{b}\right)$ and the matrix $\mathbf{D}$, using the Taylor series expansion, the model evaluated at any other parameter vector $\tilde{\boldsymbol{\beta}}$ is

$$\mathbf{F}\left(\mathbf{x}, \tilde{\boldsymbol{\beta}}\right) = \mathbf{F}\left(\mathbf{x}, \boldsymbol{b}\right) + \mathbf{D}\left(\tilde{\boldsymbol{\beta}} - \boldsymbol{b}\right).$$

From this it can be shown that

$$S\left(\mathbf{x}_i, \tilde{\boldsymbol{\beta}}\right) - S\left(\mathbf{x}_i, \boldsymbol{b}\right) = \left(\tilde{\boldsymbol{\beta}} - \boldsymbol{b}\right)^T \mathbf{D}^T\mathbf{D}\left(\tilde{\boldsymbol{\beta}} - \boldsymbol{b}\right)$$

and combining with (16) we obtain the result

$$\left(\tilde{\boldsymbol{\beta}} - \boldsymbol{b}\right)^T \mathbf{D}^T \mathbf{D} \left(\tilde{\boldsymbol{\beta}} - \boldsymbol{b}\right) \leq p s^2 F\left(p, N-p, 1-\alpha\right). \tag{17}$$

From this definition, given the regression estimate $\boldsymbol{b}$, the Jacobian matrix and s^2, the complete confidence region can be computed without the need for evaluating the model at additional points. The linear approximation to the confidence region for nonlinear problems can be found using (17) where the Jacobian matrix $\mathbf{D}$ is fixed after evaluating at $\boldsymbol{\beta} = \boldsymbol{b}$. If the nonlinear problem is only "slightly" nonlinear, i.e. the Jacobian doesn't change significantly over the parameter space, then the linear approximation confidence regions may provide a good approximation of the true confidence region.

In the next section, the confidence regions and linear approximations will be compared for equations that are common in cost estimating. If the linear approximations are good approximations to the true parameter regions then they could be suitable as input distributions for Monte Carlo simulations.

4 Cost Estimating Relationships

In this section some typical Cost Estimating Relationship (CER) models are described. Using some sample data the joint confidence regions are computed.

4.1 Learning Curves

4.1.1 Model Description

The theory of learning, which explains the cost and quantity trends observed, is primarily attributed to two researchers, T.P. Wright [4] and J. R. Crawford. The basic learning model assumes that as the number of units produced increases, the cost (in hours or dollars) to produce those units decreases. As a simple example, consider the reduction in time required to manufacture a unit as the worker becomes familiar with the instructions. As the worker produces more and gains experience, the labor time per unit decreases and the unit cost decreases.

According to both Wright and Crawford, as the total number of units doubles the cost decreases by a fixed percentage. While Wright and Crawford both developed the same model equation their interpretations of the types of cost has spawned two learning theories, Unit and Cumulative Average (CUMAV). Here we will focus only on the Unit theory. In this theory the mathematical formulation for a learning curve based model has the form

$$y = T_1 \, x^{\log_2(LC)} \tag{18}$$

where T_1 is defined as the cost of a theoretical first unit and LC is the percentage reduction that occurs when the number of units doubles and is referred to as the learning curve slope. This formulation is a two parameter model depending on the T_1 and LC values to uniquely define the predictions for all production units x. Representative slopes can range from 0.75 for manual labor intensive or complex manufacturing processes to 0.95 for simple or automated processes.

In many cases, units are procured as a lot and the costs are not tracked on an individual basis but rather only the total cost of the lot is known. In this case the average cost for a production lot i with first unit F_i and last unit L_i is given by

$$\frac{1}{L_i - (F_i - 1)} \sum_{k=F_i}^{L_i} y_k = \bar{y}_i = \frac{T_1}{L_i - (F_i - 1)} \sum_{k=F_i}^{L_i} x_k^{\log_2(LC)}. \tag{19}$$

In order to avoid the necessary computation of the sum, a potentially non-integer production unit $\tilde{x}_i$, called the lot midpoint for lot i, is defined such that the unit cost of the lot midpoint unit is the same as the average unit cost of the lot. Once $\tilde{x}_i$ is found, then the average unit cost of a lot is given as

$$\frac{1}{L_i - (F_i - 1)} \sum_{k=F_i}^{L_i} y_k = \bar{y}_i = T_1 \, \tilde{x}_i^{\log_2(LC)}. \tag{20}$$

To derive what the $\tilde{x}_i$ value should be note, one approximation of the summation can be derived using calculus

$$\begin{aligned} \sum_{k=F_i}^{L_i} x_k^{\log_2(LC)} &\approx \int_{F_i - \frac{1}{2}}^{L_i + \frac{1}{2}} x^{\log_2(LC)} dx^* \\ &= \left. \frac{x^{\log_2(LC)+1}}{\log_2(LC) + 1} \right|_{F_i - \frac{1}{2}}^{L_i + \frac{1}{2}} \\ &= \frac{\left(L_i + \frac{1}{2}\right)^{\log_2(LC)+1} - \left(F_i - \frac{1}{2}\right)^{\log_2(LC)+1}}{\log_2(LC) + 1}. \end{aligned} \tag{21}$$

Combining the above with the average unit cost equation we obtain

$$\begin{aligned} \frac{1}{L_i - (F_i - 1)} \sum_{k=F_i}^{L_i} y_k &= \frac{T_1}{L_i - (F_i - 1)} \sum_{i=F_i}^{L_i} x_k^{\log_2(LC)} \\ &\approx \frac{T_1}{L_i - (F_i - 1)} \left(\frac{\left(L_i + \frac{1}{2}\right)^{\log_2(LC)+1} - \left(F_i - \frac{1}{2}\right)^{\log_2(LC)+1}}{\log_2(LC) + 1} \right) \\ &= T_1 \left(\frac{\left(L_i + \frac{1}{2}\right)^{\log_2(LC)+1} - \left(F_i - \frac{1}{2}\right)^{\log_2(LC)+1}}{(L_i - (F_i - 1))\left(\log_2(LC) + 1\right)} \right) \end{aligned}$$

which yields an approximation of $\tilde{x}_i$ given by

$$\tilde{x}_i = \left(\frac{\left(L_i + \frac{1}{2}\right)^{\log_2(LC)+1} - \left(F_i - \frac{1}{2}\right)^{\log_2(LC)+1}}{(L_i - (F_i - 1))\left(\log_2(LC) + 1\right)} \right)^{\frac{1}{\log_2(LC)}}. \tag{22}$$

From this definition it is evident that the lot midpoint values depend on the slope of the learning curve and the lot size. A simpler heuristic approximation for the lot midpoint is given by

$$\hat{\tilde{x}}_i = \frac{F_i + L_i + 2\sqrt{F_i \, L_i}}{4} \tag{23}$$

which is the average of the algebraic and geometric lot average unit numbers. This approximation does not depend on the slope of the learning curve and can be used as an initial estimate for an iterative solver.

*In [5] a formulation is presented that relies on inclusion of correction terms to the stated approximation to provide an arbitrary level of accuracy. Additionally, by using the integration bounds F_i and $L_i + 1$ instead, we can arrive at an equivalent lot midpoint formulation for the CUMAV equations.

By substituting the (21) directly into (19), the average production lot cost model equation is given by

$$\begin{aligned}\bar{y}_i &= f\left(F_i, L_i; T_1, LC\right) + \varepsilon_i \\ &= T_1 \left(\frac{(L_i + 0.5)^{(\log_2 LC + 1)} - (F_i - 0.5)^{(\log_2 LC + 1)}}{(L_i - F_i + 1)\left(\log_2 LC + 1\right)} \right) + \varepsilon_i. \end{aligned} \tag{24}$$

This model is nonlinear with two variables and two parameters. On closer examination, the lot average unit cost equation (20) using the lot midpoint only has one input variable and by applying logarithmic transformation we obtain a form of the basic equation that is linear in the parameters

$$\ln \bar{y}_i = \ln T_1 + b \ln \tilde{x}_i \tag{25}$$

where $b = \log_2 (LC)$ is called the learning curve exponent associated with a particular learning curve slope. This model can be solved using OLS in Log-Log space to find an estimate of the $\ln T_1, b$ parameters and consequently, the unit space parameters T_1, LC.

4.1.2 Numerical Experiment

In this section we test the various approaches to solve the model and compare both the parameter estimates and their associated confidence regions. For all the tests a representative data set is used, shown in Table 1 and a plot of the data in Figure 1 using the heuristic lot midpoint $\hat{\tilde{x}}_i$.

Lot	1	2	3	4	5	6	7
First Unit	1	11	24	45	67	91	115
Last Unit	10	23	44	66	90	114	134
Average Unit Cost	138.39	121.78	100.00	86.78	70.71	69.84	70.51

Table 1: Learning Curve Data

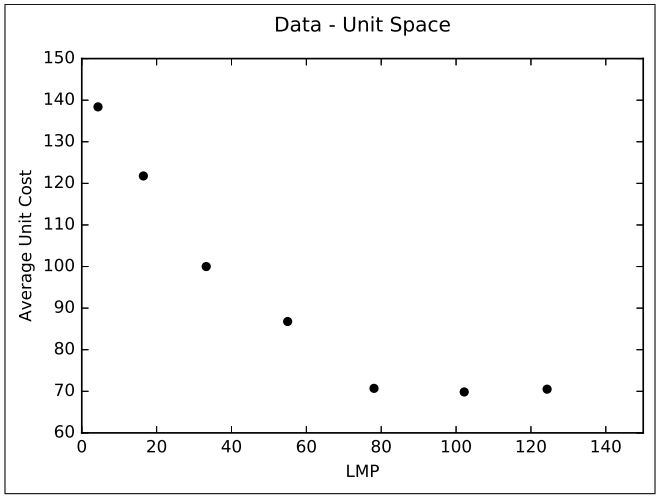

Figure 1: Learning Curve Data

First, we consider the logarithmic transformation of the model since it can be solved using OLS. This is a common approach whenever a model is transformably linear. Using a single step and the

lot midpoint heuristic approximation in (23) we obtain the solution parameters

$$\ln T_1 = 5.3415$$
$$b = -0.2278$$

which yields a unit space standard error of estimate SE of 9.6737. The unit space standard error of the estimate is given since the results of transformed linear and nonlinear regression models will be compared.

Using the parameter estimates and covariance matrix, the marginal confidence intervals and joint confidence regions can be computed and are shown in Figure 2. The marginal confidence

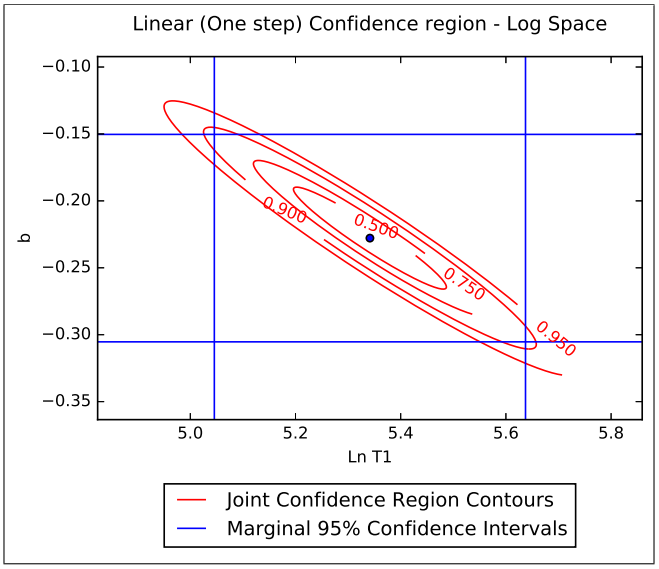

Figure 2: Linear (One-step) Confidence Region in Log Space

intervals consider the parameters independently and determine the extent of each interval using a specified confidence coefficient, the t distribution and the standard error of the parameter estimator. The intersection of the marginal confidence intervals for each parameter may not represent the set of all "reasonable" parameter values as indicated by the joint confidence region. As an example, any point in the lower left or upper right of the intersection of marginal distribution intervals that is outside the ellipses is outside of the 95% joint confidence region for the parameter pairs. These points should not be considered as a reasonable pair of parameters at the 95% level of significance. The joint confidence region in this figure has been computed using (17). As stated previously, for linear problems (17) yields the exact ellipsiodal confidence regions and are equivalent to (15) [2]. When modeling linear problems, the marginal distributions combined with the correlation information can more faithfully represent the joint confidence regions. Recall that the (Pearson) correlation is a normalized form of the covariance information and that the joint confidence regions for linear problems can be directly computed from the covariance matrix. In fact, the ellipse angles and axis lengths are determined by the eigenvectors and eigenvalues of the covariance matrix [6].

Next, we solve the iterated OLS problem and use the previous iteration LC to compute new lot midpoints based on (22). The resulting parameters are

$$\ln T_1 = 5.3350$$
$$b = -0.2262$$

with a unit space standard error of estimate SE of 9.5692. The marginal confidence intervals and joint confidence regions are shown in Figure 3. In Figure 4, joint confidence regions of both

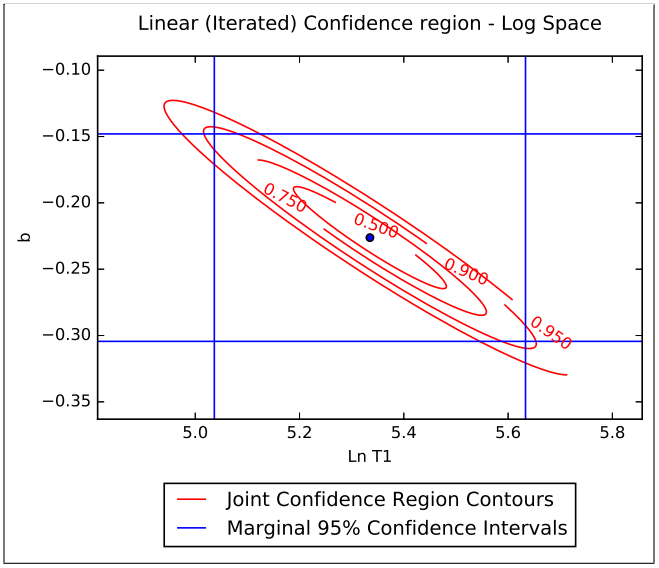

Figure 3: Linear (Iterated) Confidence Region in Log Space

the single step and iterated linear models are shown. For this data set, since there is very little

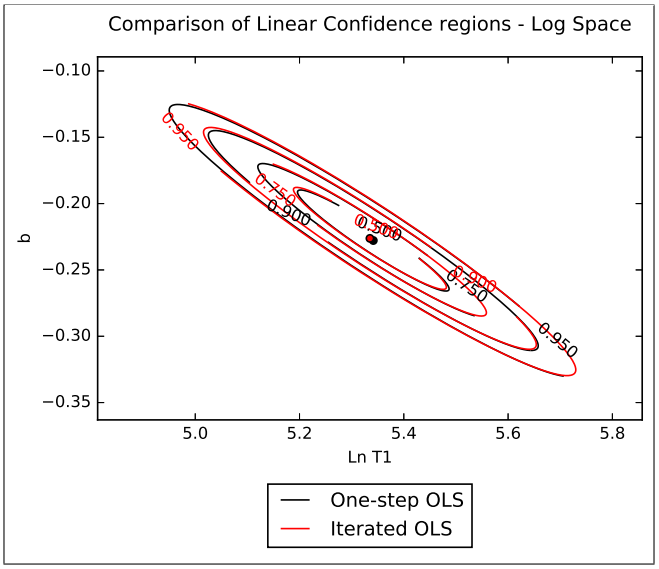

Figure 4: Comparison of Linear Confidence Regions in Log Space

difference between the two sets of solutions and their corresponding confidence regions it may be sufficient to use the lot midpoint heuristic.

Once the solution has been found in the transformed data space, the unit space parameters and regions are found by applying the inverse transformation. The resulting parameter values and confidence regions are shown in Table 2 and Figure 5 respectively. After the inverse transformation there is still little difference between the two parameter values and confidence regions. The notable distinction now is that since the transformation is nonlinear, the confidence regions for both models

	One-step OLS	Iterated OLS
T_1	208.82	207.47
LC	0.8539	0.8549
Unit space SE	9.6737	9.5692

Table 2: Comparison of Transformed OLS parameters in Unit Space

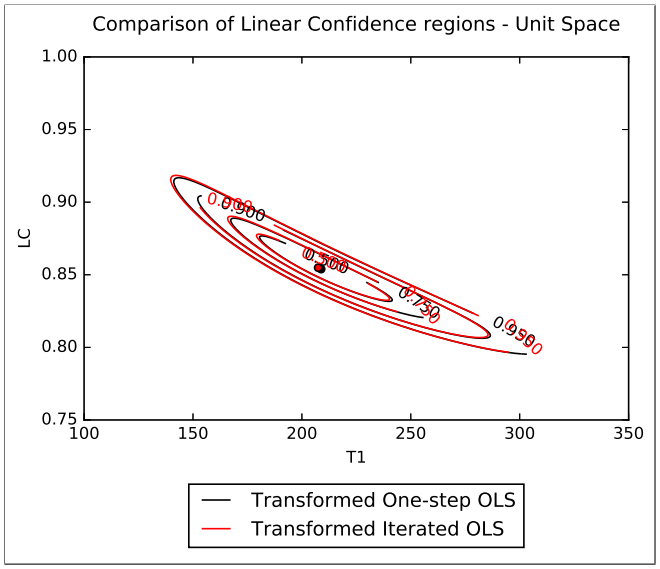

Figure 5: Comparison of Linear Confidence Regions in Unit Space

that were ellipsoidal in the regression space are now distorted ellipses after mapping back into the original data space.

The parameters are also computed directly in unit space using nonlinear regression to solve (24) and the resulting parameters are shown in Table 3. The approximate joint confidence regions are computed using the linear approximation given in (17) and the true joint confidence regions are computed using the model evaluation method given in (16). The resulting regions are plotted with the iterated OLS confidence region mapped into the unit space from Figure 6. It is evident from the figure that the linear approximation from the nonlinear regression final iteration does a pretty good job of matching the true joint confidence region obtained using the model evaluation method. Thus, for this problem and this data set the Jacobian of the nonlinear system does not change much over the parameter space. It is also obvious that the confidence regions obtained from the transformed OLS model mapped back into unit space are different than true joint confidence region.

	One-step OLS	Iterated OLS	Nonlinear Regression
T_1	208.82	207.47	195.79
LC	0.8539	0.8549	0.8569
Unit space SE	9.6737	9.5692	8.1010

Table 3: Comparison of Unit Space Parameter values by Solution Technique

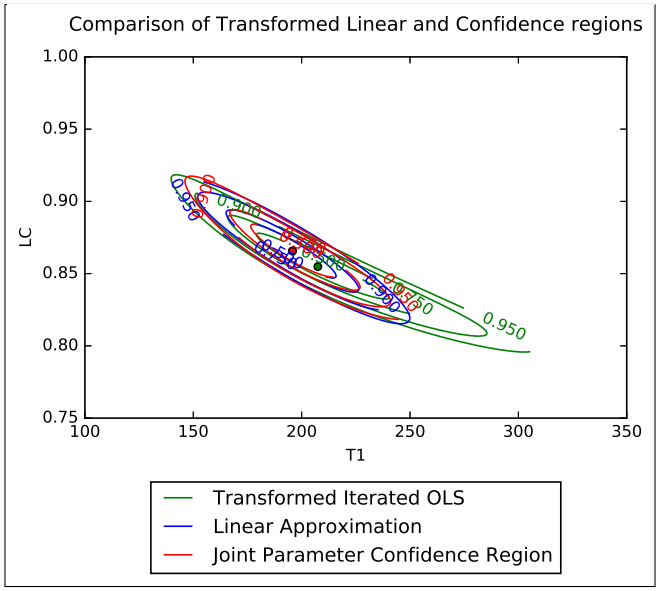

Figure 6: Comparison of Confidence Regions in Unit Space

A discussion here regarding error assumptions is warranted. The basic definition of any model should also define the assumption regarding the types of errors that affect the data. Specifically, much of standard OLS theory is based on the assumption of a linear additive error term, i.e. (1) where $f(\mathbf{x}_i, \boldsymbol{\beta})$ is linear in the parameters. When applying any nonlinear transformation to the data, the error assumption is changed according to the transformation as well. In most cases, for convenience, the error is assumed to be linear in the transformed space, i.e. where the regression is performed. When comparing models that solve for the parameters in different spaces there is inevitably different assumptions regarding the error terms. This contributes to the differences in parameter estimates when comparing the OLS methods to the unit space nonlinear regression methods. Likewise, each confidence region uses the term s^2 which is the mean squared error of its respective estimate in the regression space to determine the bounds of the confidence regions. The differences in the error term assumption and the resulting s^2 term drives the differences between the transformed OLS confidence regions and the unit space confidence regions. In a practical application with this data set, the use of the joint confidence region from a Transformed OLS model would include values that have a higher T_1 value and a steeper learning curve slope than the data suggests, potentially overstating the costs.

In this example we have compared the unit space results of both parameter estimates and joint confidence regions of transformably linear regression problems to their corresponding nonlinear regression results. The data suggests that though the parameter estimates themselves may be close, the confidence regions can be more severely impacted by the transformation applied and the resulting error assumption implications.

4.2 Weight Curves

Another common cost estimating relationship relates the cost of an item to its weight. Here we present two nonlinear equations, each have only one independent variable and two parameters to illustrate the differences in joint confidence regions of the parameters.

4.2.1 Model Descriptions

First, a model that is similar to the learning curve model is given by

$$\text{Model 1}: \; y = \theta_1 x^{\theta_2}. \tag{26}$$

In contrast to the learning curve equation which had a negative learning exponent b, in weight based CERs the growth exponent θ_2 is usually positive indicating that the cost increases as weight increases. If the growth exponent is greater than one, the CER is superlinear, i.e. the cost grows faster than proportionally to a weight increase. While this may be accurate for some data sets, it is not assumed to be typical.

The second model employed is given by

$$\text{Model 2}: \; y = \theta_1 \left(1 - e^{-\theta_2 x}\right). \tag{27}$$

In this model we note that the independent variable and a parameter appear as part of an exponent.

4.2.2 Numerical Experiment

In this section, the two models are solved using the representative data shown in Table 4. This data is from [7], without the observation relative importance weighting factor.

Data point	Cost $K	Weight (lbs)	Data point	Cost $K	Weight (lbs)
Obs 1	3,106.64	77.05	Obs 8	19,796.80	332.50
Obs 2	29,166.32	1,236.77	Obs 9	7,526.40	269.42
Obs 3	4,820.48	232.14	Obs 10	6,002.24	123.84
Obs 4	34,111.22	863.36	Obs 11	11,668.48	316.15
Obs 5	6,387.04	224.40	Obs 12	6,329.12	59.77
Obs 6	20,871.60	720.44	Obs 13	4,683.20	59.17
Obs 7	28,621.92	959.33	Obs 14	21,068.72	369.12

Table 4: Weight CER Data

The parameters for each model type are computed directly in unit space using nonlinear regression. The resulting parameters and standard errors are shown in Table 5. The resulting fitted data from models are shown in Figure 7. As stated, these two different model types were chosen to illustrate that two models with similar fit statistics can have vastly different underlying parameter joint confidence regions.

	Model 1	Model 2
θ_1	233.91	40,021.07
θ_2	0.6991	0.0013
SE	4,525.4	4,273.5

Table 5: Weight CER Model Results

The joint confidence regions are computed using both the linear approximation and the model evaluation method with the results shown in Figure 8 and Figure 9.

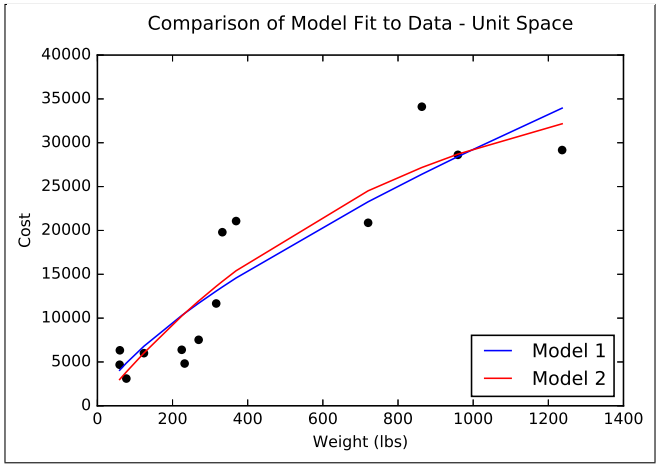

Figure 7: Comparison of Model fits

From the figures we see that the confidence regions are quite different between the two model both in scale and shape. Also, the true parameter confidence regions are quite different than their corresponding linear approximations. For continuous nonlinear models, the linear approximation may only be valid in a small neighborhood of the computed parameter values as is shown in the close up Figure 10 and Figure 11.

In this example we have shown that two models with relatively equal fit statistics can have substantially different parameter confidence regions in size and scale. Furthermore, for both models, the joint confidence regions resulting from the linear approximation method provide fairly poor approximations to the true joint parameter confidence regions. The concern here is that for these models using the marginal distributions and Pearson correlation would include many values that are outside the true confidence regions. The ultimate result is any uncertainty analysis output would be incorrect and potentially misleading to any decision maker relying on the analysis.

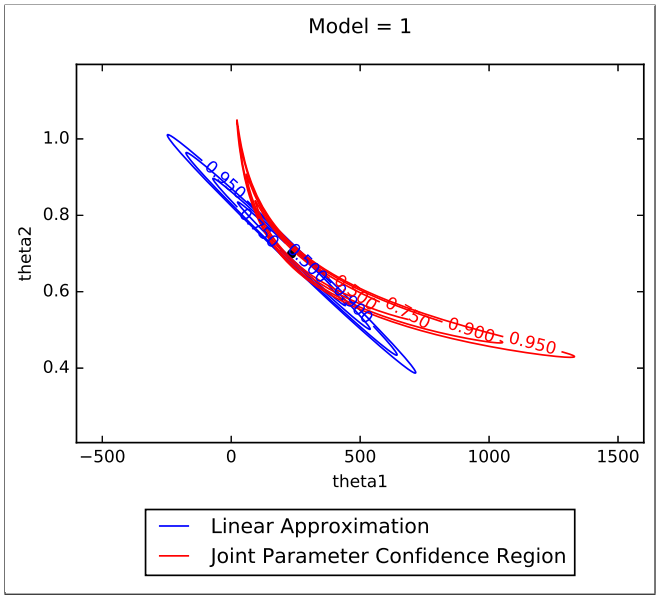

Figure 8: Comparison of Confidence Regions for Model 1

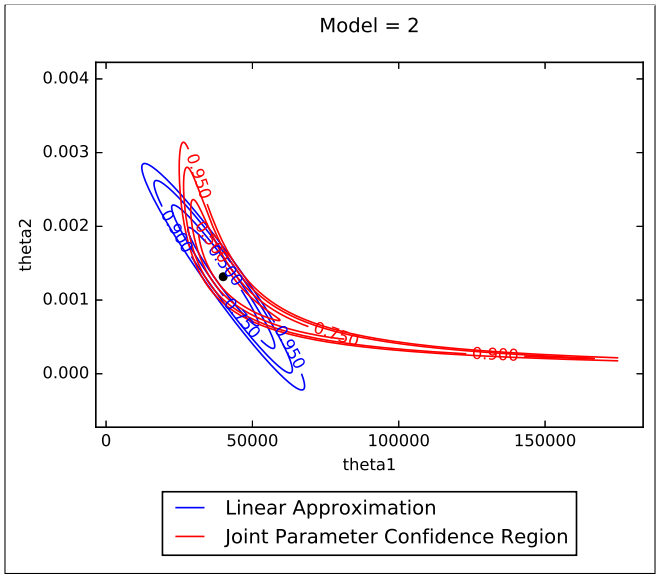

Figure 9: Comparison of Confidence Regions for Model 2

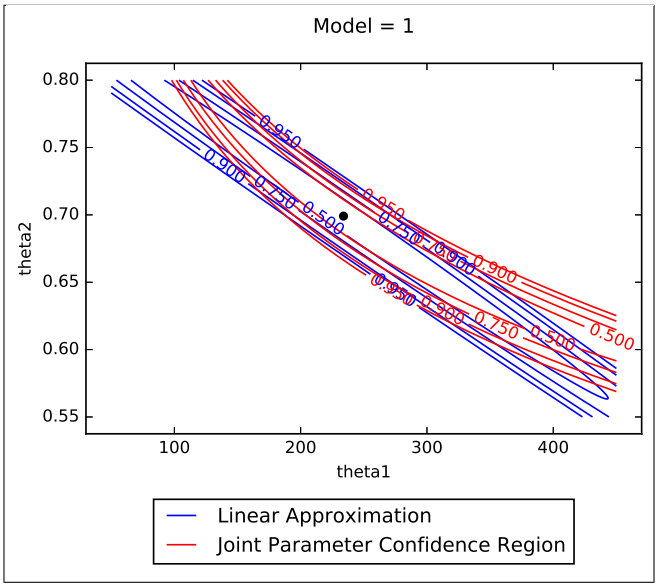

Figure 10: Comparison of Confidence Regions for Model 1

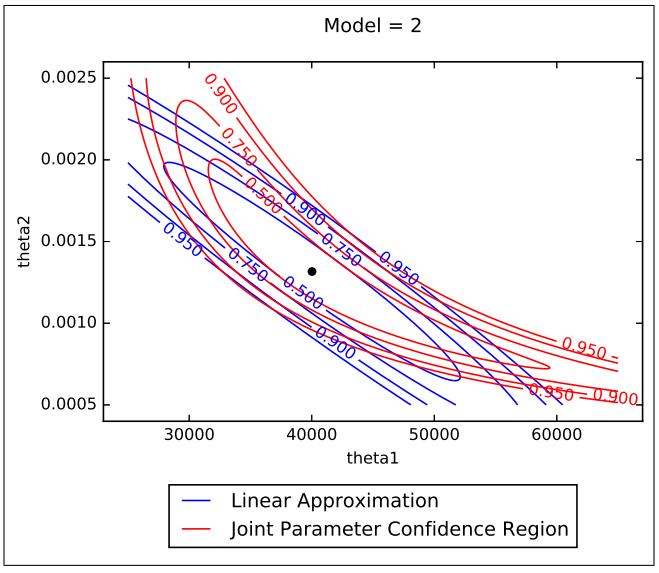

Figure 11: Comparison of Confidence Regions for Model 2

5 Conclusions

5.1 Summary of Results

In this paper, the joint confidence region computation methods have been reviewed and applied to common nonlinear cost estimating problems. It has been done to highlight several key ideas and limitations within our current capabilities.

First, it was shown that the product of marginal confidence intervals may not provide a reasonable representation for a joint confidence region. The joint confidence regions capture additional parameter dependence information and can be significantly different than the product of marginal confidence intervals, which treat parameters as independent. Next, it was shown that once mapped back to the original data space, confidence regions resulting from a linearizing transformation may not provide an accurate representation of the true parameter confidence region. The assumptions regarding the error terms are modified through the transformation and introduce differences both in the parameter estimates and associated confidence regions. Once the confidence regions are mapped back into the original space the confidence regions are no longer ellipsoidal.

While many Monte Carlo applications allow for non-independent variables, typically correlation is used to generate appropriate random samples of the marginal distributions instead of sampling directly the multivariate parameter distribution. The Pearson correlation coefficient only measures the linear relationship between two variables which, as we have seen from the examples, for nonlinear models the parameter relationships are likely nonlinear as well. To allow for non-ellipsoidal shapes or combinations of different types of distributions to be sampled dependently, many software packages (including Crystal Ball and @Risk) utilize a rank correlation measure. Unfortunately, there are easily constructed examples where over the range of variables the rank correlation value is zero yet the variables are strongly dependent. So while the traditional approach may still provide good results, there is a chance that for any problem that any sampling technique based on correlation alone, may be flawed and could produce misleading results.

5.2 Future Work

In spite of the vast research in parameter identification and uncertainty quantification, there are still many open areas of research. A logical next step to this paper is to identify other methods to sample from joint distributions in a way that preserves more if not all of the dependence information between parameters. While there are many methods that do currently exist such as multivariate inverse transform sampling, Copulas and Markov chain Monte Carlo methods, most are heavily data driven and would require more robust software and programming packages such as R or Python over the traditional spreadsheet based models.

Additionally, in the figures presented in this paper the $0.50, 0.75, 0.90$ and 0.95 confidence levels have been shown. There is a clear need for rules of thumb and/or policy regarding the acceptable levels of parameter uncertainty that should be accounted for as part of a cost estimate uncertainty analysis. These levels and their impacts consequently influence the distribution of the model outputs and any decisions made using the data.

In light of all the obstacles to proper analysis, there is occasionally, a temptation to simplify the model form so that easier theory and results may be applied. However, doing so may degrade the ability to make inferences about the reality of the situation, physical properties or processes that are being analyzed. Finally, as cost estimators and stewards of limited resources, we should always be concerned about supplying the decision makers with the right amount of useful data and analysis to make the best decision.

References

[1] Norman R Draper and Harry Smith. *Applied Regression Analysis*. John Wiley & Sons, 1966.

[2] Douglas M Bates and Donald G Watts. *Nonlinear Regression Analysis and Its Applications*. John Wiley & Sons, 1988.

[3] David Roxbee Cox and David Victor Hinkley. *Theoretical Statistics*. CRC Press, 1979.

[4] Theodore P. Wright. Factors Affecting the Cost of Airplanes. *Journal of Aerospace Science*, 1936.

[5] David A. Lee. The Cost Analyst's Companion. *Logistics Management Institute, McLean, VA*, 1997.

[6] C. J. Wild G. A. F. Seber. *Nonlinear Regression*. Wiley, 1989.

[7] Shu-Ping Hu. Develop press for nonlinear equations. 2016.