Abstract

AKHAVAN TABATABAEI, RAHA. AN INTERACTIVE FRAMEWORK FOR THE PARAMETER DESIGN PROBLEM. (Under direction of Dr. Yahya Fathi.)

We propose an interactive framework for solving the Parameter Design Problem. In this context we consider a response function $Y = h(x_1,...,x_n)$, where $x_1,...,x_n$ are random variables with known probability distribution functions, and h is a continuous and differentiable function. Given the distribution functions of x_1 through x_n and the function h, we use a family of Johnson distributions to approximate the probability density function of Y by using its first four moments as input. This density function can be displayed graphically and compared with the given specification limit of Y. Using this approach we develop a computer program that would allow the user to modify the set points of $x_1,...,x_n$ manually, and immediately observe the impact of this adjustment on the probability density function of Y. The user can then interactively search for and determine a satisfactory set of values for these set points. We also present two case studies and solve them by this method. The framework that we propose also provides a platform for employing other techniques such as nonlinear programming or statistical design of experiment in order to assist the user in determining a satisfactory solution for the parameter design problem.

An Interactive Framework for The Parameter Design Problem

By

Raha Akhavan Tabatabaei

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the
Degree of Master of Science

Industrial Engineering

Raleigh

2005

APPROVED BY:

Chair of Advisory Committee

Biography

Raha Akhavan Tabatabaei is a graduate student with co-major in Industrial Engineering and Operations Research at North Carolina State University, where she will receive her Master of Science degree in December 2005. She received her bachelor's degree in Industrial Engineering from Sharif University of Technology in January of 2001. She is from Tehran, Iran.

Table of Contents

Page

List of Tables

List of Figures

Page

Chapter 2

Chapter 4

Chapter 5

Chapter 1

Introduction

In this Chapter we define the Parameter Design Problem and review the literature on this subject. Then we propose an interactive model to solve the Parameter Design Problem.

1.1 Definition of the Problem

Let $x = (x_1,....,x_n)$ be a vector of design variables (components) for a manufacturing product or a manufacturing process and let $Y = h(x_1....x_n)$ represent an output characteristic of interest. We assume that $x_1,....,x_n$ are random variables with known probability distributions. Since the output characteristic, *Y*, is a function of a set of random variables, *Y* itself is a random variable whose probability density function and its moments are dependent on probability density functions of $x_1,....,x_n$ and the form of the response function, *h*.

Let $\mu_1,....,\mu_n$ denote the mean values (set points) of $x_1,....,x_n$. We further assume that $\mu_1,....,\mu_n$ are controllable parameters (i.e., their values can be determined by the design engineer). In the design phase values of $\mu_1,....,\mu_n$ are determined such that the nominal value of *Y* is equal to its target value, τ. During the production phase, existence of noise can cause values of $x_1....x_n$ to deviate from their set points $\mu_1,....,\mu_n$, hence causing *Y* to deviate form its target value, τ.

Parameter Design Problem focuses on determining $\mu_1,....,\mu_n$ such that not only nominal value of *Y* is kept on target, but also the variation or noise in *Y* is minimized.

1.2 Literature Review

Initially, the Parameter Design Problem was addressed by G. Taguchi. Since then, many researchers have proposed various methodologies for solving this problem. The most notable work among all is one that is originally proposed by Taguchi and Wu [13] which suggests a model to maximize a performance measure known as 'signal to noise ratio'. Taguchi and Wu's method is later discussed in Leon et al. [12] and Box [1].

This methodology is based on using an orthogonal array in experimental design to maximize signal to noise ratio. Fathi [6] states the advantages of this method in being modest in computational requirements and suitable for the situations where the analytical form of the response function between the design variables and the output characteristic *Y* is not available. The major disadvantage is also stated as terminating at a suboptimal solution.

Box and Fung [2] proposed a constrained nonlinear programming (NLP) model to maximize the signal to noise ratio, and showed that when applicable, this model could result in better solutions. Fathi [7] proposed a constrained nonlinear programming model that minimizes the variance of *Y*. In this method variance of *Y* is approximated by successive quadratic approximations of the response function. This is only applicable to cases where the response function is known or can be well approximated.

Another approach proposed by Fathi [6] approximates the variance of *Y* by using a linear approximation. He uses Taylor series expansion to approximate *h* up to its linear term. From there, mean and variance of *h* are calculated about the point ($\mu_1 .... \mu_n$). Then he uses this approximation as the objective function of a NLP model. He constrains the problem by forcing nominal value of *Y* to be on target, and $\mu_1 .... \mu_n$ satisfy any other technical requirement (constraint). This methodology is referred to as Parameter Design Model with Linear Approximation of Variance or PDM-LAV.

1.3 The Proposed Model

In this thesis we propose a different approach for solving the Parameter Design Problem. This is an interactive approach in which the design engineer can manually adjust the values of the controllable parameters $\mu_1 .... \mu_n$ and see the effect of this adjustment on the probability density function of *Y*, instantaneously.
In this model, we take advantage of a method of approximating the probability density function of a variate by an empirical distribution. In particular we approximate probability distribution of *Y* by a family of Johnson distributions.

To determine the parameters of the corresponding Johnson distribution, we use the first four moments of the variate *Y*. To this end we approximate these first four moments of the variate either by using the Taylor series expansion of the function $h(x_1,....,x_n)$ about the set points $\mu_1,....,\mu_n$, or (as appropriate) by using a Monte-Carlo simulation model.

Once the first four moments of *Y* are evaluated, we identify the Johnson family that it belongs to, and then calculate the distribution parameters of that family. Once the appropriate Johnson family and its parameters are estimated, we form the probability density function (pdf) for *Y* and plot its graph. Now a design engineer can visually observe the shape of pdf of *Y* at the given set points for $\mu_1,....,\mu_n$ and compare it with the target value of *Y* and its specification limits. He or she can then try different settings for $\mu_1,....,\mu_n$ as appropriate. For each setting that is tried, the pdf of *Y* is graphed and hence the design engineer can see the effect of this new setting on the pdf of *Y.* This process continues until a desirable set of values for $\mu_1,....,\mu_n$ is determined.

1.4 Organization of the Thesis

The empirical distributions and methods of approximation are discussed in Chapter 2. The methods for approximating the first four moments of *Y* are discussed in Chapter 3. Chapter 4 explains the details of our methodology and the software that is developed for this purpose. In Chapter 5 we present two case studies involving the Parameter Design Model, using this methodology.

The software that we develop can also serve as the basis for a framework in which we may include different approaches in the Parameter Design Problem., i.e. the statistical design of experiments or the nonlinear programming model. In Chapter 6, the features that can be added to the existing model and also other methodologies that can be included in the framework to enhance its capacity and facilitate the search for an optimal combination of set points, are discussed. In this Chapter we also discuss our findings and avenues for further research.

Chapter 2

Approximation by Empirical Distributions

In this Chapter we discuss methodologies of fitting an empirical distribution to a variate and in particular we focus on the Johnson distributions.

2.1 Background

In the context of the PDP, for a general form of the response function $Y = h(x_1,...,x_n)$, it is usually difficult to determine the exact form of the pdf for *Y*, even if the exact form of the pdf for $x_1,...,x_n$ is known. In this case we use an approximation of the pdf of *Y* in our analysis.

One method for approximating the pdf of *Y* is fitting a well-known distribution such as normal, Gamma, or lognormal. The most common of these methods is the use of normal distribution, since it follows from the central limit theorem that normal distribution provides a reasonable representation of many physical phenomena. Gamma and lognormal distributions are also used for variables which are bounded at one end, and Beta distribution is used for those that are bounded at both ends.

Although these models cover a wide variety of shapes, they do not provide the desirable degree of generality required. In Figure 2-1, the regions in the (β_1, β_2) plane that are covered by the well-known distributions of lognormal, normal, Beta, Gamma and student t are shown. In this figure, $\beta_1 = \left(\frac{\mu_3}{(\sigma^2)^{3/2}} \right)^2$ represents the square of the standardized measure of skewness and $\beta_2 = \frac{\mu_4}{(\sigma^2)^2}$ represents the standardized measure of peakedness. In these expressions μ_3 and μ_4 are the third and fourth moments of *Y* about its mean, and σ^2 is its variance.

In this figure, the distributions which do not have any shape parameter, such as normal, exponential, and uniform, are represented as single points, whereas others which involve different shapes of curves and have one or two shape parameters are represented as lines or regions on the (β_1, β_2) plane.

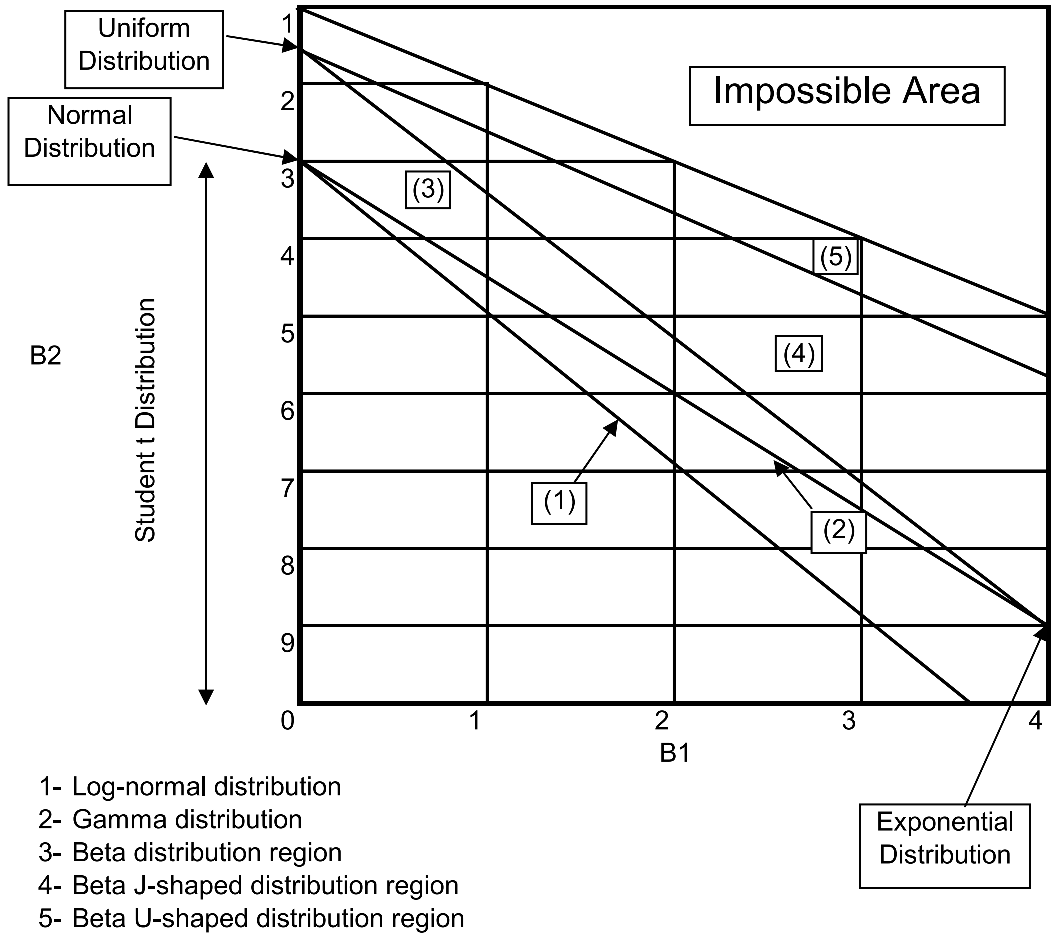


Figure 2-1: (β_1, β_2) plane for well-known distributions.

While this figure can be used for examining the closeness of a given distribution to a well-known distribution (i.e., uniform, normal, etc.), little can be done about the large area that is not occupied by any of these well-known distributions. This justifies the use of empirical distributions.

A more general technique for approximating the pdf of a random variable would be to use either Johnson or Pearson distributions. In this section we discuss Johnson distribution in some detail and use this distribution throughout the thesis. For discussion on Pearson distribution one can refer to [8] and [10].

2.2 Johnson Distribution

Johnson [11] suggested transforming a standard normal variate as the basis for empirical distributions. One advantage of this transformation is that we can use a table of areas under a standard normal distribution for estimating the percentiles of the fitted distribution.

The general form of this transformation is [8] :

$$z = \gamma + \eta\tau(x;\varepsilon,\lambda), \quad \eta > 0, -\infty < \gamma < \infty, \lambda > 0, -\infty < \varepsilon < \infty, \tag{2-1}$$

where x is the variable to be fitted by a Johnson distribution.

In this general form, τ is an arbitrary function, $\gamma, \eta, \varepsilon, \lambda$ are the four parameters of Johnson distribution and z is a standard normal variate.

Johnson has proposed the following three families for the function τ :

(a) $$\tau_1(x;\varepsilon,\lambda) = \ln(\frac{x-\varepsilon}{\lambda}), \qquad x \geq \varepsilon, \tag{2-2}$$

(b) $$\tau_2(x;\varepsilon,\lambda) = \ln(\frac{x-\varepsilon}{\lambda+\varepsilon-x}), \qquad \varepsilon \leq x \leq \varepsilon+\lambda \tag{2-3}$$

(c) $$\tau_3(x;\varepsilon,\lambda) = \sinh^{-1}\ln(\frac{x-\varepsilon}{\lambda}), \qquad -\infty < x < \infty \tag{2-4}$$

Hahn and Shapiro [8] find the probability density functions for the three families of Johnson distributions using methods of variable transformation, given that z is a normal variate.

1-Family that corresponds to τ_1 also known as *Johnson S_L family:*

$$f_1(x) = \frac{\eta}{(x-\varepsilon)\sqrt{2\pi}} \exp\{-\frac{1}{2}[\gamma + \eta\ln(\frac{x-\varepsilon}{\lambda})]^2\}, \tag{2-5}$$

$x \geq \varepsilon, \eta > 0, -\infty < \gamma < \infty, \lambda > 0, -\infty < \varepsilon < \infty.$

Setting $\gamma^* = \gamma - \eta\ln\lambda$ gives,

$$f_1(x) = \frac{\eta}{(x-\varepsilon)\sqrt{2\pi}} \exp\{-\frac{1}{2}\eta^2[\frac{\gamma^*}{\eta} + \ln(x-\varepsilon)]^2\}, \tag{2-6}$$

$x \geq \varepsilon, \eta > 0, -\infty < \gamma^* < \infty, -\infty < \varepsilon < \infty.$

This is in fact a generalized form of the lognormal distribution as described in page 99 in [8]. In most of the literature as well as in the model that we develop in this thesis, equation (2-6) is used as the pdf of S_L family.

2- Family that corresponds to τ_2 also known as *Johnson* S_B *family:*

$$f_2(x) = \frac{\eta}{\sqrt{2\pi}} \frac{\lambda}{(x-\varepsilon)(\lambda - x + \varepsilon)} \exp\{-\frac{1}{2}[\gamma + \eta \ln(\frac{x-\varepsilon}{\lambda - x + \varepsilon})]^2\}, \qquad (2\text{-}7)$$

$\varepsilon \le x \le \lambda + \varepsilon, \eta > 0, -\infty < \gamma < \infty, \lambda > 0, -\infty < \varepsilon < \infty.$

3-Family that corresponds to τ_3 also known as *Johnson* S_U *family:*

$$f_3(x) = \frac{\eta}{\sqrt{2\pi}} \frac{1}{\sqrt{(x-\varepsilon)^2 + \lambda^2}} \exp[-\frac{1}{2}(\gamma + \eta \ln\{(\frac{x-\varepsilon}{\lambda}) + [(\frac{x-\varepsilon}{\lambda})^2 + 1]^{1/2}\})^2], \qquad (2\text{-}8)$$

$-\infty < x < \infty, \eta > 0, -\infty < \gamma < \infty, \lambda > 0, -\infty < \varepsilon < \infty.$

Hahn and Shapiro [8] also show the (β_1, β_2) plane for these Johnson families.

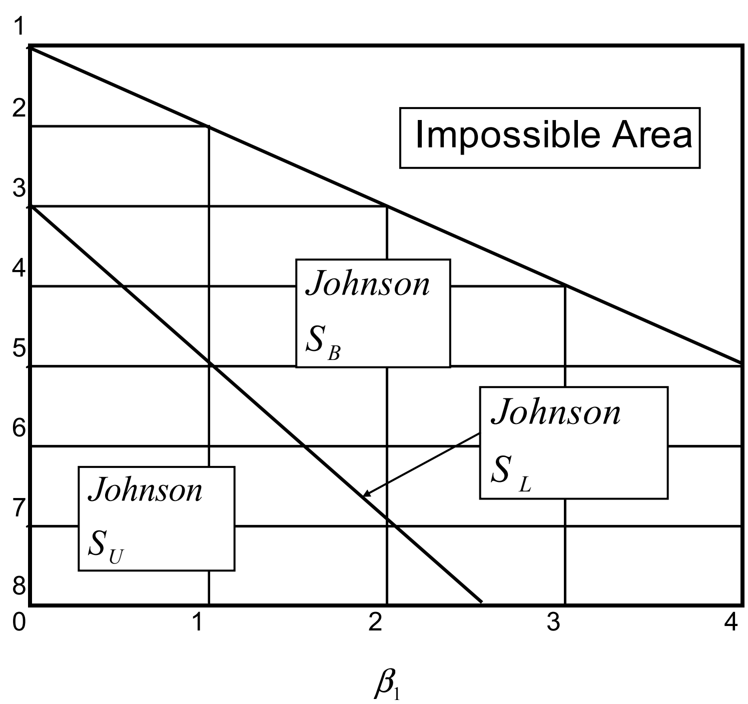


Figure 2-2: (β_1, β_2) plane for Johnson distributions.

Note that the Johnson S_L and S_B families both have two shape parameters γ and η, one location parameter ε, and one scale parameter λ.

From the defined range of the random variate x in each family, we can see that S_U, S_L and S_B distributions are defined for unbounded variates, variates bounded at one end, and variates bounded from both above and below, respectively. However these limitations need not be strictly followed when the distributions are used for approximation. This is similar to approximating a bounded variate with normal distribution which is unbounded by nature. Figures 2-3, 2-4, and 2-5 show some examples of curves from the S_U, S_L and S_B families, respectively. It can be observed that these families of distributions give a wide range of shapes; hence they can be used in a variety of situations.

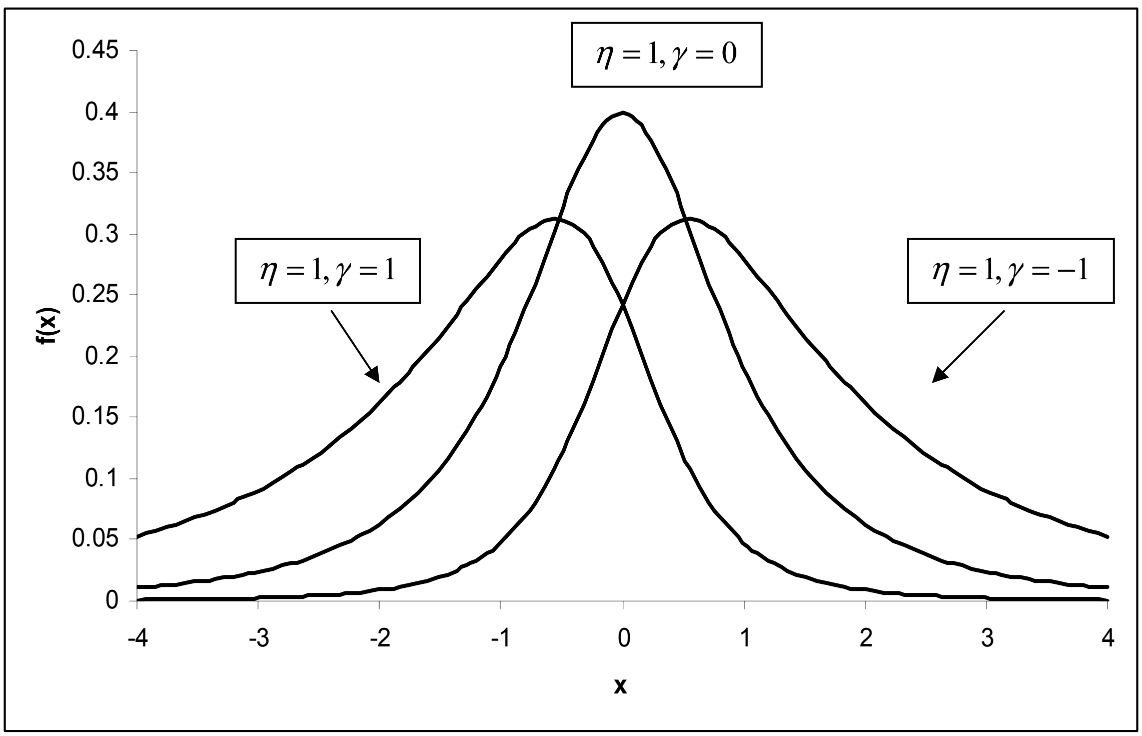


Figure 2-3: Johnson S_U distributions with $\varepsilon = 0, \lambda = 1$ and various values of the parameters η and γ.

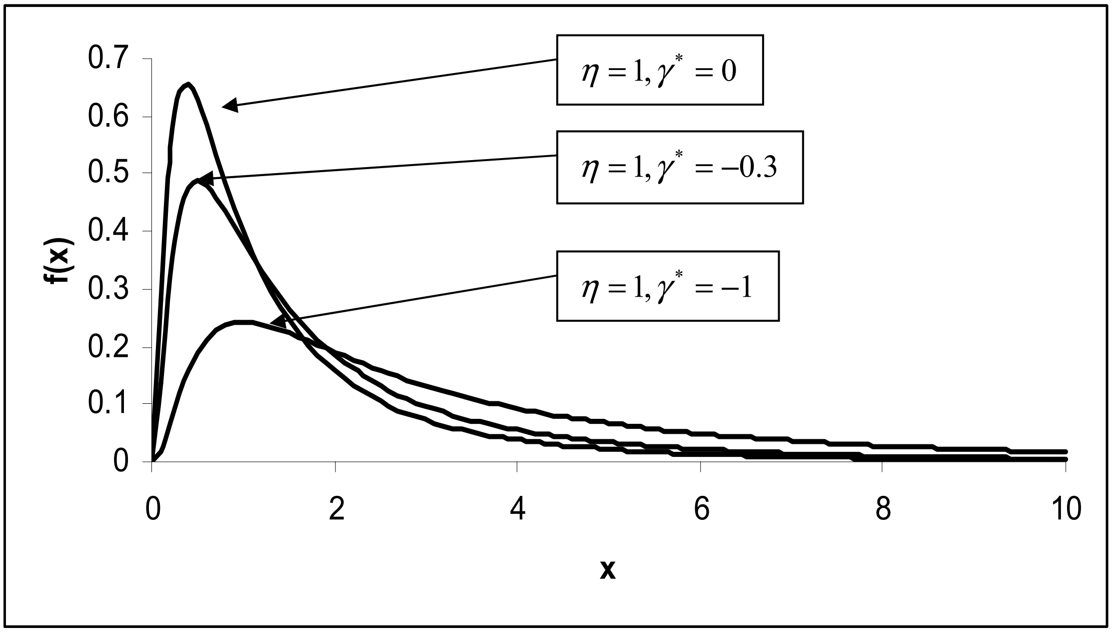


Figure 2-4: Johnson S_L distributions with $\varepsilon = 0$ and various values of the parameters η and γ^*.

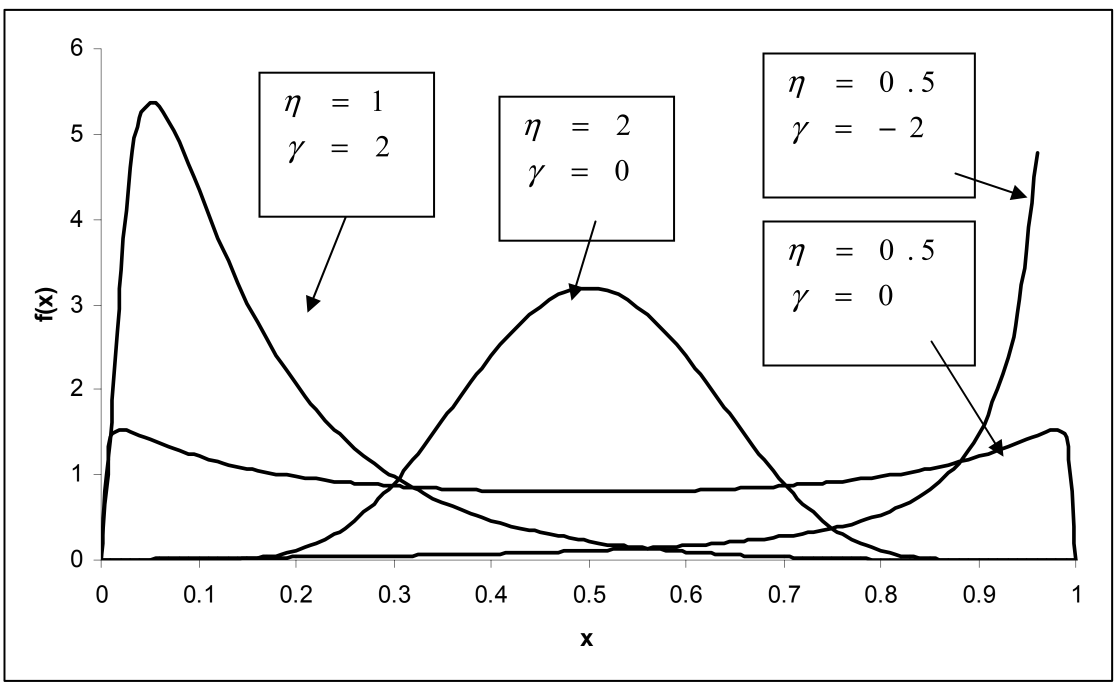


Figure 2-5: Johnson S_B distributions with $\varepsilon = 0, \lambda = 1$ and various values of the parameters η and γ.

2.3 Approximation by Johnson Distribution

In order to approximate a variate, *Y*, using Johnson distributions, we must first determine its corresponding family of Johnson distribution and then determine the parameters of that family. To this end we need to determine the values of β_1 and β_2, and identify the point (β_1, β_2) in Figure 2-2. If the (β_1, β_2) point falls close to the log-normal line, then S_L is chosen. If it falls above or below this line, then S_B or S_U should be chosen, respectively. The range of the regions in Figure 2-2 can be extended by use of these parametric equations

$$\beta_1 = (\omega - 1)(\omega + 2)^2, \tag{2-9}$$

$$\beta_2 = \omega^4 + 2\omega^3 + 3\omega^2 - 3. \tag{2-10}$$

Additional points can be obtained by choosing values for ω and solving for the corresponding values of β_1 and β_2, as we discuss later.

To determine parameters of the fitted Johnson distribution (i.e. $\gamma, \eta, \lambda, \varepsilon$), two different approaches are used in the literature. One method is based on using a relatively large random sample. This method is discussed in [8] in details. The second approach, which is used in this thesis, is based on using the first four moments of the random variate *Y*. This method is proposed by Hill, Hill and Holder in [9], where they provide the details of this algorithm and a corresponding computer program. This program takes the mean, the standard deviation, the skewness and the peakedness of a variate as its input. With this input, the algorithm finds the family of Johnson distribution that fits the variate and then calculates the parameters of the fitted distribution. This method is numerical in nature and it is based on the works by Elderton and Johnson [4] and Draper [3]. The complete script of the computer program can be found in Appendix 1. Here, we briefly review the basic ideas of this algorithm.

Let μ, σ, μ_3 and μ_4 denote the mean, standard deviation, third moment about the mean and fourth moment about the mean of the variate, respectively. Hill et al. [9] further separates the regions in Figure 2-2 by introducing two new families of distributions. One family corresponds to the point $(\beta_1 = 0, \beta_2 = 3)$ which represents the normal pdf. The second family corresponds to the line segment separating the two regions of

"Johnson S_B" and the "Impossible Area". This line segment is referred to as Johnson S_T. The relationship between β_1 and β_2 for this line segment is $\beta_2 = \beta_1 + 1$.

The first step in the algorithm is to determine the family of Johnson distribution that fits the variate. Let β_1^* denote the given value for square of skewness and β_2^* denote the given value for peakedness of the variate. If $\beta_1^* = 0$ and $\beta_2^* = 3$, then normal pdf is selected. If $\beta_1^* + 1 = \beta_2^*$, then S_T is selected. Otherwise, we solve equation (2-9) with β_1^* to determine the corresponding value of ω. We denote this value by ω^*. Then we substitute ω^* into equation (2-10) and solve for β_2. Let $\hat{\beta}_2$ denote the corresponding value of β_2. If $\beta_2^* = \hat{\beta}_2$, then we select S_L; if $\beta_2^* > \hat{\beta}_2$, we select S_B; and if $\beta_2^* < \hat{\beta}_2$, we select S_U.

At this point the algorithm starts to determine the parameters of the selected family of Johnson distribution.

1- S_L Curves:

With ω^* having been evaluated as above, let

$$\eta = (\ln \omega^*)^{-\frac{1}{2}}, \tag{2-11}$$

$$\gamma^* = \frac{1}{2}\eta \ln\{\omega^*(\omega^* - 1)/\sigma^2, \tag{2-12}$$

$$\varepsilon = \pm\mu - \exp\{(1/2\eta - \gamma^*)/\eta\}, \tag{2-13}$$

In equation (2-13) the sign of μ is determined in each case to be the same as the sign of μ_3 .For example, if the third moment of the variate, μ_3, is negative then (2-13) becomes $\varepsilon = -\mu - \exp\{(1/2\eta - \gamma^*)/\eta\}$, otherwise, it becomes $\varepsilon = \mu - \exp\{(1/2\eta - \gamma^*)/\eta\}$.

2- S_U Curves:

In estimating the parameters of a S_U curve, two cases are identified depending on the value of β_1^*. If $\beta_1^* = 0$, the corresponding curve is symmetrical, and its associated parameters are as follow.

$$\omega = \{(2\beta_2^* - 2)^{\frac{1}{2}} - 1\}^{\frac{1}{2}} \tag{2-14}$$

$$\eta = (\ln\omega)^{-\frac{1}{2}} \tag{2-15}$$

$$\gamma = 0\,. \tag{2-16}$$

If $\beta_1^* > 0$ then, the corresponding curve is not symmetrical and the associated values of parameters ω, η and γ are found by an iterative procedure which is discussed in [4]. This iterative procedure is described in the computer program listing in Appendix 1.

In both of these cases, the values of ε and λ are then found from the following equations:

$$\mu_2 = \frac{1}{2}\lambda^2(\omega - 1)\{\omega\cosh(2\gamma/\eta) + 1\} \tag{2-17}$$

$$\mu_1 = \varepsilon - \lambda\omega^{\frac{1}{2}}\sinh(\gamma/\eta)\,. \tag{2-18}$$

3- S_B Curves:

In order to determine the value of η for this family, we consider its boundary values. When the point (β_1^*, β_2^*) is approaching the S_T boundary, η approaches zero; when the point is approaching the S_L boundary, η tends to be the same value as for a S_L curve [equation (2-11)]. To estimate η, it is first set equal to the value from interpolation between these two boundaries. Then Hill et al. [9] use a series of formulas based on work of Draper [3] to evaluate the rest of parameters based on estimated value of η, and then improve these values iteratively. These series of formulas are also presented in the computer program listing in Appendix 1.

4-Normal Curves:

Parameter η is set equal to $1/\sigma$, and γ is set equal to μ/σ; λ and ε are set to 0.

5- S_T Curve:

For this family let,

$$\varepsilon = \mu - \frac{\sigma\sqrt{1/2 + 1/2\sqrt{1 - \frac{4}{\beta_1 + 4}}}}{\sqrt{1/2 - 1/2\sqrt{1 - \frac{4}{\beta_1 + 4}}}}, \quad (2\text{-}19)$$

$$\lambda = \varepsilon + \sigma\sqrt{\beta_1 + 4}, \quad (2\text{-}20)$$

$$\eta = \sqrt{1/2 + 1/2\sqrt{1 - \frac{4}{\beta_1 + 4}}}, \quad (2\text{-}21)$$

and γ be set to 0.

2.4 Examples of Johnson Approximation

In this section we present several numerical examples. In these examples, we compare the probability distribution function of a number of variates that have known probability distributions (e.g., Beta and Gamma, etc.), with their approximations by Johnson distribution, in order to validate this algorithm.

Example 1

In this example we fit Johnson distribution to Beta and Gamma distributions with various parameter values. To start the algorithm we need the first four moments of the variate. The first four moments of Beta and Gamma variates are given in [8] with respect to the parameters of each distribution. We take some cases of these variates with different values for their parameters, and use the values of their moments as input to the algorithm. After the algorithm determines the family type and evaluates the parameters of the corresponding family, we insert the parameters in equations (2-6), (2-7) or (2-8) as appropriate. Then we plot the pdf and compare it with the actual pdf of the corresponding distribution as appropriate. Comparing Figures 2-1 and 2-2 shows that both Beta and Gamma curves are in Johnson S_B region. Several examples of comparing the Johnson distribution and the actual probability density functions of Gamma and Beta are shown here.

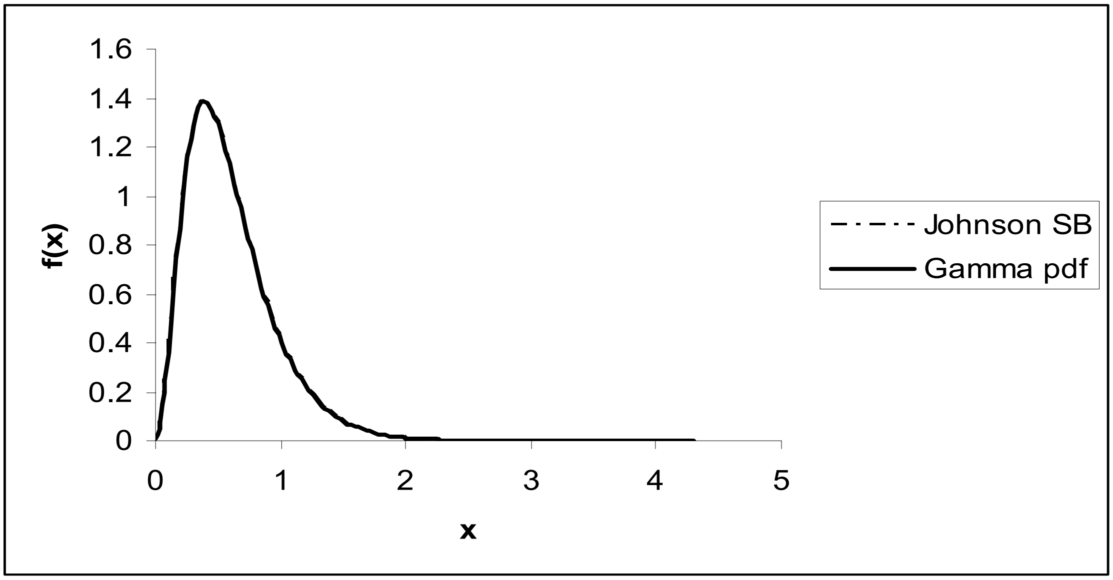


Figure 2-6: Fitting Johnson distribution to Gamma with shape parameter =3 and scale parameter =5

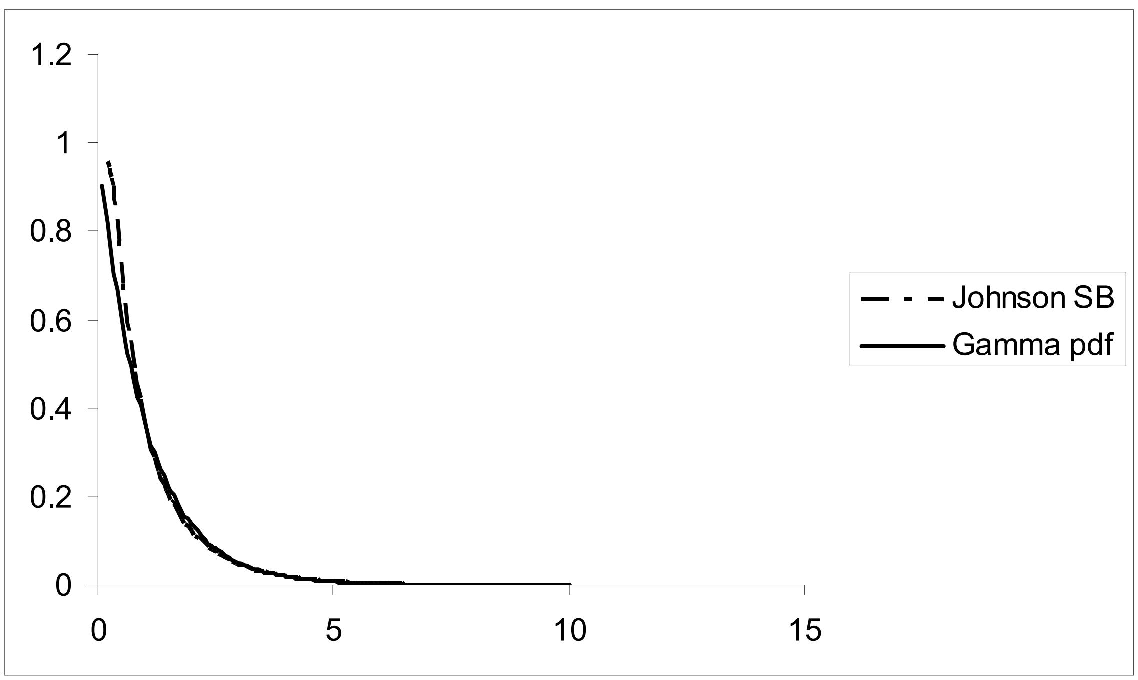


Figure 2-7: Fitting Johnson distribution to Gamma with shape parameter =1 and scale parameter =1 (Exponential)

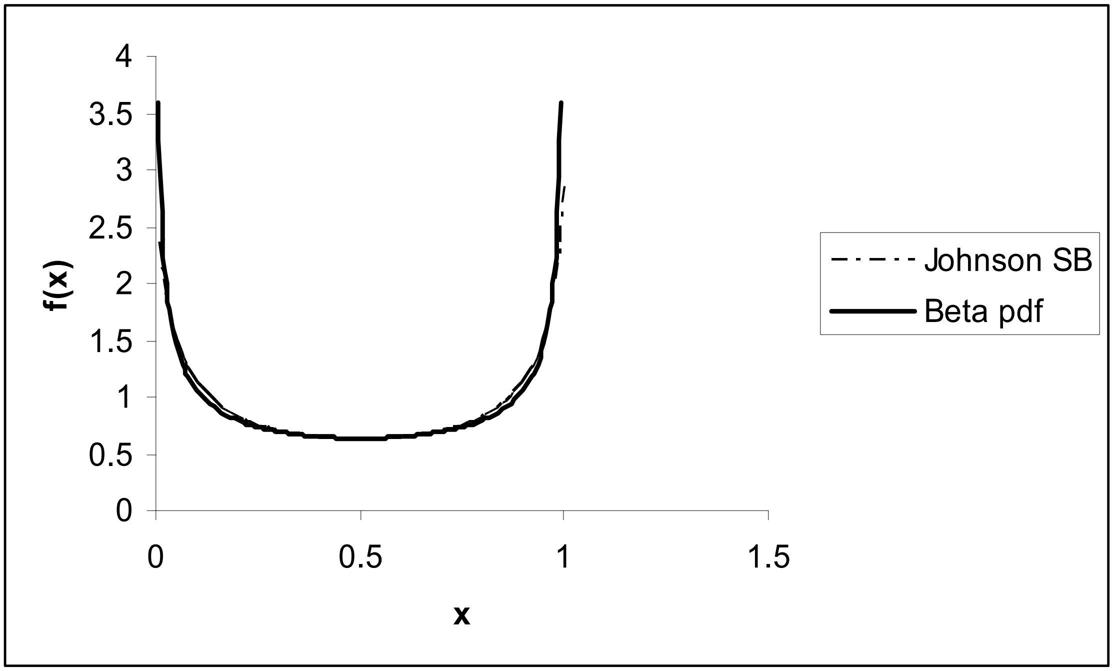


Figure 2-8: Fitting Johnson distribution to Beta with α =.5 and β =.5

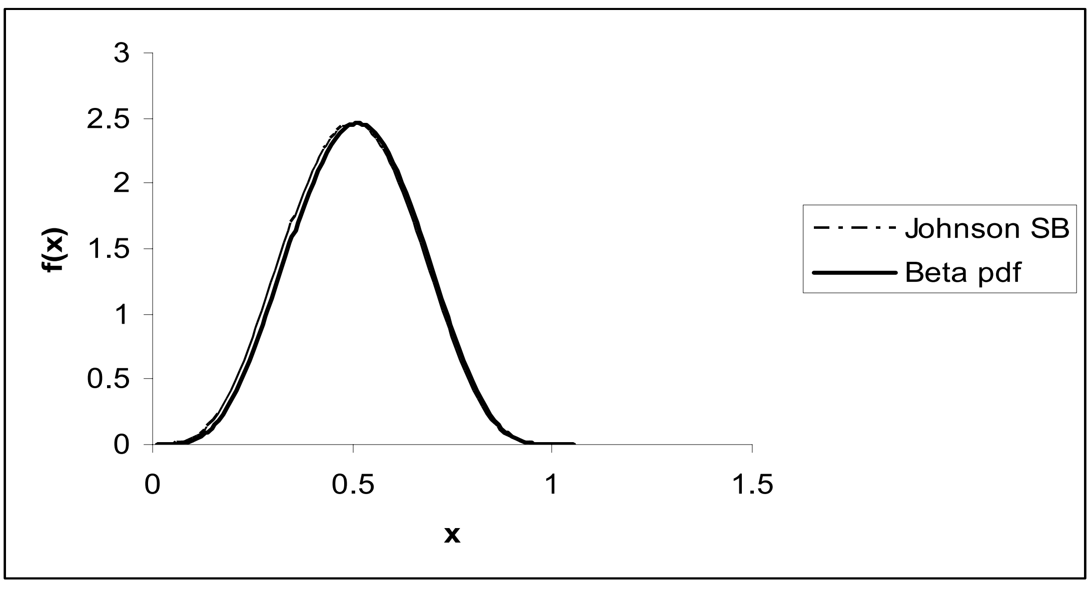


Figure 2-9: Fitting Johnson distribution to Beta with α =5 and β =5

We observe that the fitted Johnson distribution is reasonably close to the actual probability density function of the known distribution in each of these cases.

Example 2

In this example we solve a case that is originally presented in [8]. In this case the corresponding (β_1, β_2) point is in the S_U region. We find the parameters of Johnson S_U family by the method that we discussed earlier and compare them with the results in [8].

In this example measurements of the coefficient of friction for a metal were obtained on 250 samples. The mean, standard deviation, the skewness and the peakedness are estimated from the sample and are presented in [8] as 0.0345 for mean, 0.0098 for variance, 0.87 for skewness and 4.92 for peakedness, respectively. Plotting the values of $\beta_1 = (0.87)^2 = 0.7569$ and $\beta_2 = 4.92$ in Figure 2-2 suggests the use of a Johnson S_U distribution. Table 2-1 shows the values of the parameters that are found by each method.

Parameters of S_U	Hahn and Shapiro	Hill, Hill and Hull
γ	**-1.783**	**-1.785**
η	**2.433**	**2.446**
λ	**0.0169**	**0.0169**
ε	**0.0198**	**0.0198**

Table 2-1: Coefficient of Friction Example

Figure 2-10 shows the fitted Johnson distributions to the parameters of Table 2-1. The two probability density functions closely cover each other.

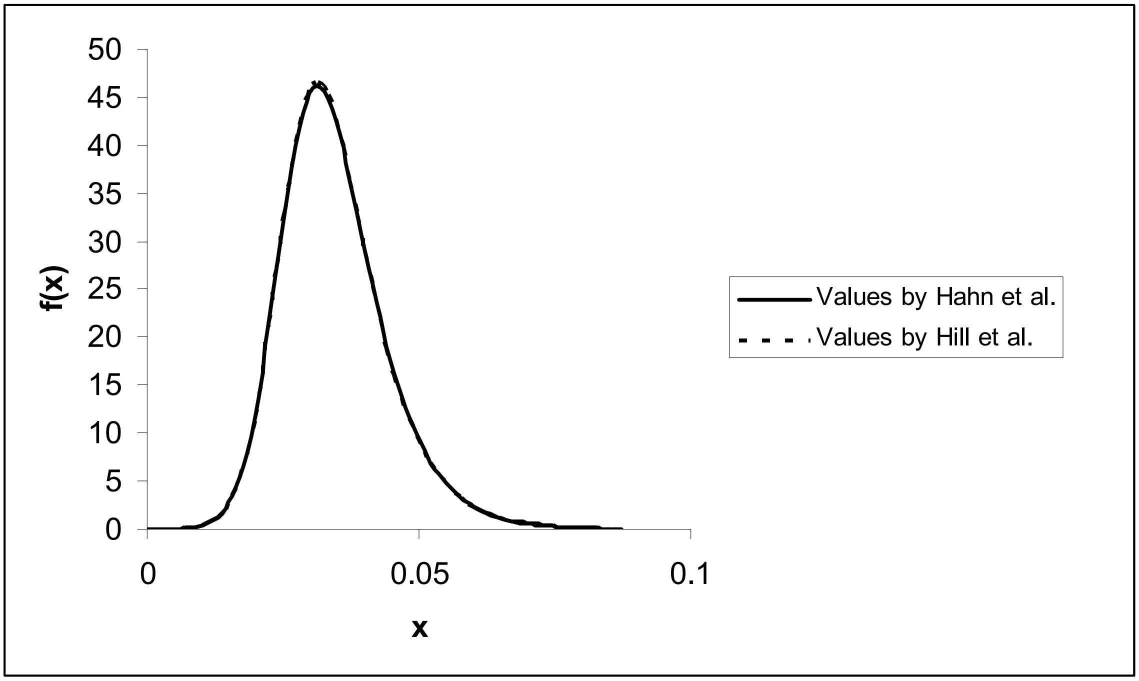


Figure 2-10: Comparison of methods

We consider this approximation method to be reasonably close for the purpose of the interactive model that we propose in Chapter 5.

Chapter 3

Evaluating the First Four Moments of the Response Function

In this Chapter we discuss two methods for determining the moments of *Y*, where $Y = h(x_1,...,x_n)$ as described in Chapter 1. We assume that we know the probability density function for each component variable $x_1,...,x_n$. We further assume that the function *h* is given in closed form and that it is continuous and differentiable at all points. Both methods that are discussed here are approximation methods, since determining the exact values of the moments of *Y* can be computationally difficult, if at all possible. One of these two methods is based on Monte-Carlo simulation and the other method is based on the Taylor series expansion of the function *h*.

3.1 Monte-Carlo Simulation

In this method with help of a computer program, capable of generating random numbers, we generate a value for each of the components $x_1,...,x_n$ and determine the corresponding value of *Y*. If we repeat this process *k* times, we can create a collection of *k* randomly generated values for *Y* (i.e., a random sample of *Y*).

Based on this random sample, we can estimate the first four moments of *Y* by a series of formulas that are adopted from [8].

1- Estimating mean:

$$\bar{y} = \frac{\sum_{i=1}^{k} y_i}{k} \tag{3-1}$$

2- Estimating variance:

$$m_2 = \hat{\sigma}^2 = \frac{\sum_{i=1}^{k}(y_i - \bar{y})^2}{k} = \frac{k\sum_{i=1}^{k} y_i^2 - (\sum_{i=1}^{k} y_i)^2}{k(k-1)} \tag{3-2}$$

3-Estimating the third moment:

$$m_3 = \frac{\sum_{i=1}^{k}(y_i - \bar{y})^3}{k} = \frac{\sum_{i=1}^{k} y_i^3}{k} - 3\frac{\sum_{i=1}^{k} y_i^2}{k}\frac{\sum_{i=1}^{k} y_i}{k} + 2\left(\frac{\sum_{i=1}^{k} y_i}{k}\right)^3 \tag{3-3}$$

4-Estimating the fourth moment:

$$m_4 = \frac{\sum_{i=1}^{k}(y_i - \bar{y})^4}{k} = \frac{\sum_{i=1}^{k} y_i^4}{k} - 4\frac{\sum_{i=1}^{k} y_i}{k}\frac{\sum_{i=1}^{k} y_i^3}{k} + 6\left(\frac{\sum_{i=1}^{k} y_i}{k}\right)^2\frac{\sum_{i=1}^{k} y_i^2}{k} - 3\left(\frac{\sum_{i=1}^{k} y_i}{k}\right)^4 \tag{3-4}$$

Then the skewness of Y is estimated as

$$\sqrt{b_1} = \frac{m_3}{(m_2)^{3/2}}, \tag{3-5}$$

and the peakedness of Y is estimated as

$$b_2 = \frac{m_4}{(m_2)^2}. \tag{3-6}$$

Equations (3-1) and (3-2), i.e., $\bar{y}$ and m_2, can be used as estimates for mean and variance of Y respectively. Equations (3-5) and (3-6), i.e., $\sqrt{b_1}$ and b_2, can be used to estimate $\sqrt{\beta_1}$ and β_2, respectively.

3.2 Taylor Series Approximation

Tukey [14] develops a collection of formulas for approximating the first four moments of the random variate Y. These formulas are based on the Taylor series expansion of the function h about the point $\mu_1,...,\mu_n$, using its first few terms. The complete forms of these formulas are given in [14], and we employ a simplified version of these formulas as given below. In these formulas the following notations are used for the moments of the components ($x_i, i = 1,2,...,n$): mean of $x_i = \mu_i$, variance of $x_i = \sigma_i^2$, skewness of $x_i = \gamma_i$, and peakedness of $x_i = \Gamma_i$. Also μ_Y denotes the mean of *Y*, σ_Y^2 denotes the variance of *Y*, ϕ_Y denotes the third central moment of *Y* about its mean, and φ_Y denotes the fourth central moment of *Y* about its mean.

$$\mu_Y \approx h(\mu_1, \mu_2, ..., \mu_n). \tag{3-7}$$

$$\sigma_Y^2 \approx \sum_i h_i^2 \sigma_i^2 \tag{3-8}$$

$$\phi_Y \approx \sum_i h_i^3 \gamma_i \sigma_i^3 \tag{3-9}$$

$$\varphi_Y \approx \sum_i h_i^4 (\Gamma_i - 3)\sigma_i^4 + 3\sigma_Y^4 \tag{3-10}$$

In these formulas all the summations are form 1 to *n*, and $h_i = \left.\dfrac{\partial h}{\partial x_i}\right|_{\mu,\mu_2,...,\mu_n}$.

Formulas for more accurate approximation of these parameters are presented in [14]. These formulas contain higher order terms of the Taylor series approximation.

3.3 Numerical Examples

In this section we present two numerical examples and observe the accuracy of these two methods for determining the moments of *Y*. In example 1, we only use the Monte-Carlo simulation method since the corresponding function is linear, and in example 2 we use both methods.

Example 1

We consider the following function: $y = x_1 + x_2 + x_3$, and we assume $x_1 \sim N(2,1^2)$, $x_2 \sim N(3,1^2)$, and $x_3 \sim N(5,2^2)$.

Of course in this example it is easy to determine the exact form of the probability distribution function of *y* and its parameters. It is clear that *y* has a normal distribution with the following parameters: mean of *y* is $\mu_y =$ 2+3+5=10, variance of *y* is $\sigma_y^2 = 1^2 + 1^2 + 2^2 = 6$. Skewness and peakedness of *y* are 0 and 3, respectively since *y* is a normal random variate.

In order to evaluate these parameters using the Monte-Carlo simulation method, we randomly generate 250,000 values for each component. Then we determine the corresponding values for the function y. We can then estimate mean, variance, skweness and peakedness of y using equations (3-1) through (3-6). Table 3-1 shows the results from simulating this function versus the values that are calculated analytically.

Parameter	Values form Simulation	Actual Values
Mean	10.0090	10
Variance	6.0034	6
Skewness	0.0018	0
Peakedness	2.9953	3

Table 3-1: Comparing the results form simulation with actual values

As it is observed, the results are very close to the actual moments. The 95% confidence interval for mean in this example is [9.9993, 10.0186] and for variance is [5.9703, 6.0369].

Example 2

We consider the function $y = x^2$ and assume that $x \sim N(\mu, \sigma^2)$. In this example we determine the moments of y using both approximation methods with different sets of values for μ and σ^2. Results are shown in Table 3-2.

	$\mu = 0, \sigma = 1$		$\mu = 10, \sigma = 2$		$\mu = 5, \sigma = 3$	
	Taylor Series Approx.	Simulation	Taylor Series Approx.	Simulation	Taylor Series Approx.	Simulation
Mean	1.00	1.00	104.00	104.11	34.00	34.60
Variance	0.00	1.98	1,600.00	1,639.20	900.00	1,086.82
Third Moment	0.00	7.50	0.00	39,314.00	0.00	56,757.00
Fourth Moment	0.00	51.53	7,680,000	9,313,200.00	2430,000	7,446,421.00

Table 3-2: Comparing Taylor approximation method with Monte-Carlo simulation

Comparison shows that Taylor series approximation method approximates mean and variance of the variate relatively well. However, it is observed that the approximated values for the third and the fourth moments of y are quite inaccurate. In this example we deliberately choose a simple function ($y = x^2$) to show that Taylor series approximation method based on formulas (3-7) through (3-10) gives poor approximation values for the third and the fourth moments, even for such an uncomplicated function. We address this issue further in Chapter 6.

Chapter 4

Methodology

In this Chapter we propose a methodology for developing an interactive model for the Parameter Design Problem, as defined in Chapter 1. We also discuss a computer program that we have developed for this interactive model and present a numerical example.

4.1 The Model

In the context of the Parameter Design Problem as we defined earlier, we need to determine a set of values for the controllable parameters, $\mu_1,...,\mu_n$ so that the corresponding value of the response variable, *Y*, is at or close to its target value τ , with minimum variation. To this end we propose an interactive model and a corresponding computer program, in which the user is able to graphically observe the probability distribution of *Y* for a given set of values for $\mu_1,...,\mu_n$. We then allow the user to change the values of one or more of the parameters (i.e., $\mu_1,...,\mu_n$) and immediately observe the impact of this change on the graph of the corresponding probability distribution of *Y*. In this model the user can make a subjective assessment of the distribution of *Y* by comparing it with its target value and its specification limits. Objective measurements such as the corresponding values of the "percent defective" or the "expected loss function" can also be estimated to assist the user in evaluating the quality of the solution. In this manner the user can employ this interactive model to determine an appropriate set of values for $\mu_1,...,\mu_n$.

Obviously, in order to implement this approach, we need to determine the form of the probability distribution function of *Y* for a given response function $Y = h(x_1,...,x_n)$ and for the given distribution functions of $x_1,...,x_n$. In general, determining the exact form of the distribution function of *Y* can be difficult, if at all possible. Hence, we propose to approximate this function using an appropriate Johnson distribution function. To this end we take the following two steps.

Step 1-Approximating the First Four Moments of *Y*

In Chapter 3 we discuss two methods for approximating the moments of a distribution, i.e., Monte-Carlo simulation and Taylor series approximation. Monte-Carlo simulation is more accurate but more time consuming. Taylor series approximation is less accurate in this case but obviously much faster. In this model we approximate the first four moments of the random variate, *Y*, with one of these two methods, as appropriate. See Remark 1 below.

Step 2- Fitting Johnson Distribution

After approximating the first four moments of *Y*, we can use the numerical methods of Hill et al. [9] to find the corresponding Johnson family, as we discuss in Chapter 2. Once the parameters of the Johnson distribution are known, we can form the pdf of the response function, using equations (2-6) through (2-8), and plot it. This pdf can be used to calculate different measures of variation such as estimating the "fraction defective" or the "expected loss function". Evaluation of these measures can be saved for further comparison with other feasible combinations of set points. We can also provide assistance to the user for determining an appropriate set of values for $\mu_1,...,\mu_n$. To this end we can employ a non-linear programming model as discusses in [7] or use other techniques.

Remark 1

Ideally we would like to use Taylor series method to approximate the first four moments of *Y* for each value of $\mu_1,...,\mu_n$, since it is much faster than the Monte-Carlo simulation method. But in practice we note that this method is not sufficiently precise for approximating the third and the fourth moments of *Y*, although it is relatively more accurate for approximating the first two moments of *Y* (i.e., mean and variance). Hence, we employ the Monte-Carlo simulation method to approximate the third and the fourth moments. However, we also note that in some situations (depending on the form of the response function), the third and the fourth moments of *Y* are less sensitive to changes in $\mu_1,...,\mu_n$. Thus we employ a mixed strategy in which we initially use Monte-Carlo simulation to approximate all four moments of *Y*. Then during the interactive phase, while changing the values of $\mu_1,...,\mu_n$, we assume that the values of the third and the fourth moments of Y remain unchanged; we use Taylor series method to approximate the first two moments of Y during the interactive phase and carry out the analysis accordingly. When the interactive phase is complete, once again we employ a Monte-Carlo simulation model to approximate the third and the fourth moments. If their values at this stage are significantly different from their values at the start of the interactive phase, we update their values and repeat the interactive phase. If, on the other hand, their terminal values are reasonably close to their original values, we terminate the procedure.

Remark 2

As a terminating criterion we use the corresponding value of the fraction defective as estimated by using the fitted Johnson distribution. First we evaluate the fraction defective at the original set points, and refer to it as p_0. In the interactive phase we try to reduce the value of the fraction defective by changing the values of the set points. Then we evaluate the fraction defective at the end of the interactive phase via the fitted Johnson distribution, using its mean and variance as calculated by Taylor series approximation and its skewness and peakedness as the initial values obtained via Monte-Carlo simulation at the start of the interactive phase. We refer to this value as p_1.

Now we employ a Monte-Carlo simulation model to re-evaluate the moments of the response at the current set points. Then we calculate the fraction defective with these updated values of the moments. We refer to this value as p_2. If p_1 and p_2 are sufficiently close to each other we terminate the algorithm. Otherwise we repeat the interactive phase using the updated values of the third and the fourth moments.

In the numerical examples of the next Chapter, if the value of p_2 is within 5% of the value of p_1, we terminate the procedure, otherwise we repeat the interactive phase. Figure 4-1 shows a flowchart of this procedure.

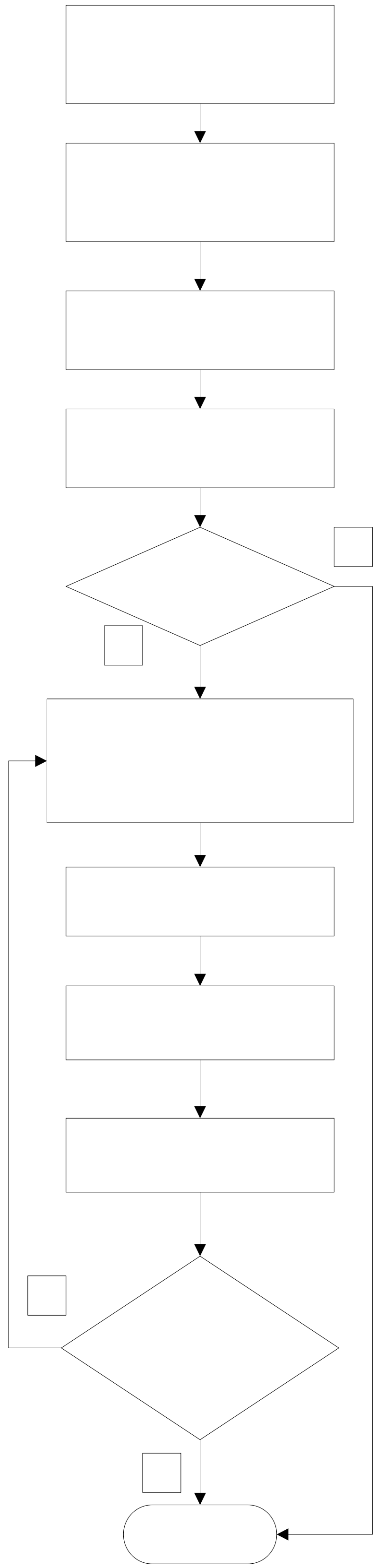


Figure 4-1: Flowchart of the program

4.2 Computer Program

We have developed a computer program to implement this interactive approach for the Parameter Design Problem. Microsoft Excel with VBA in its background is chosen to develop this model. There are two main reasons for this choice: the high graphical capability of MS Excel to draw plots and charts, and availability of the software on almost every desktop. Since in this method the graphs for the pdf should be plotted every time that the set points are changed, visual features of MS Excel charts make it relatively easy to carry out this approach. Appearance of charts can be edited to the user's taste and other features can be added as the user wishes. Another reason that makes MS Excel appealing in this context is the fact that it is available on almost any machine in work environments and the users have easy access to the software. Therefore, potential users do not need to install a new software package on their machines.

4.2.1 Modules of the Program

This program consists of three main procedures and some subroutines that are invoked during the implementation of these procedures. The first procedure approximates the moments of the response function; the second procedure fits a Johnson distribution to this function based on the value of its moments; and the third procedure plots the pdf and the specification limits. Figure 4-2 shows a schematic view of this software system. In this figure, sections that are connected to the main program via solid lines are already developed and they are operational. Sections that are connected via dash lines are stipulated for future development.

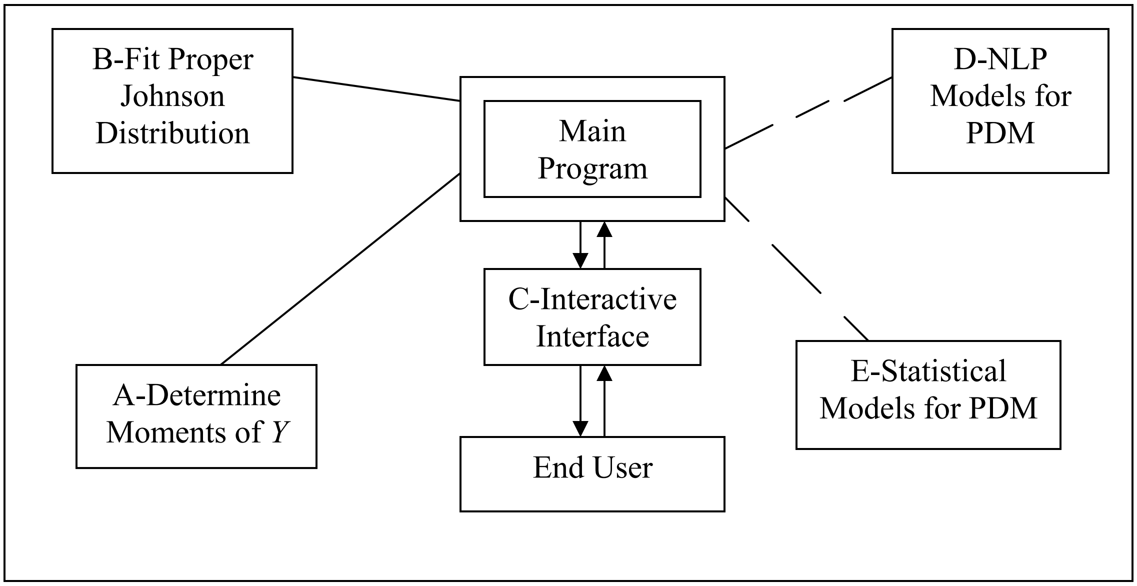


Figure 4-2: Schematic view of the program

Procedure A that is designed to approximate the first four moments of the response function, takes the closed form relationship between the response function *Y* and the components as well as the first four moments of the components as input. This program is designed such that a response function with up to ten arguments can be given as input. Once the input is entered in the designated worksheet (this worksheet is called "Moments of Y"), a macro which is called "FunctionMom" will approximate the first four moments of *Y* as well as its skewness and peakedness using the Taylor series approximation method. For the Taylor series approximation method we use the formulas of Chapter 3, i.e., (3-7) through (3-10).

To evaluate the partial derivatives in these formulas, we use the symmetric difference quotient formula for numerical approximation of first derivative which is

$$f'(x) = \frac{f(x+\delta) - f(x-\delta)}{2\delta} \qquad (4\text{-}5).$$

Here, δ represents incremental changes in variable *x*, and Evans [5] recommends using $\delta = \sigma$ in the context of the above formulas.

From this point, the next part of the methodology, i.e., fitting Johnson distribution, begins and all of the calculations for this part are done by a macro that is called "JOHNSONTYPES". This macro copies the mean, variance, skewness and peakedness of the response, from the Table in the worksheet "Moments of Y", to a designated Table in another worksheet that is called “Johnson”. First the macro checks if there are any flaws in the input to this step. To this end location of (β_1, β_2) is checked against Figure 2-2 to determine if the point is in the impossible area. If this point is in the impossible area, message “IMPOSSIBLE AREA” will appear as an indication of system failure. Otherwise, the standard deviation is checked for non-negativity. If both of the tests are passed, then a message will appear to show that the input is successfully entered.

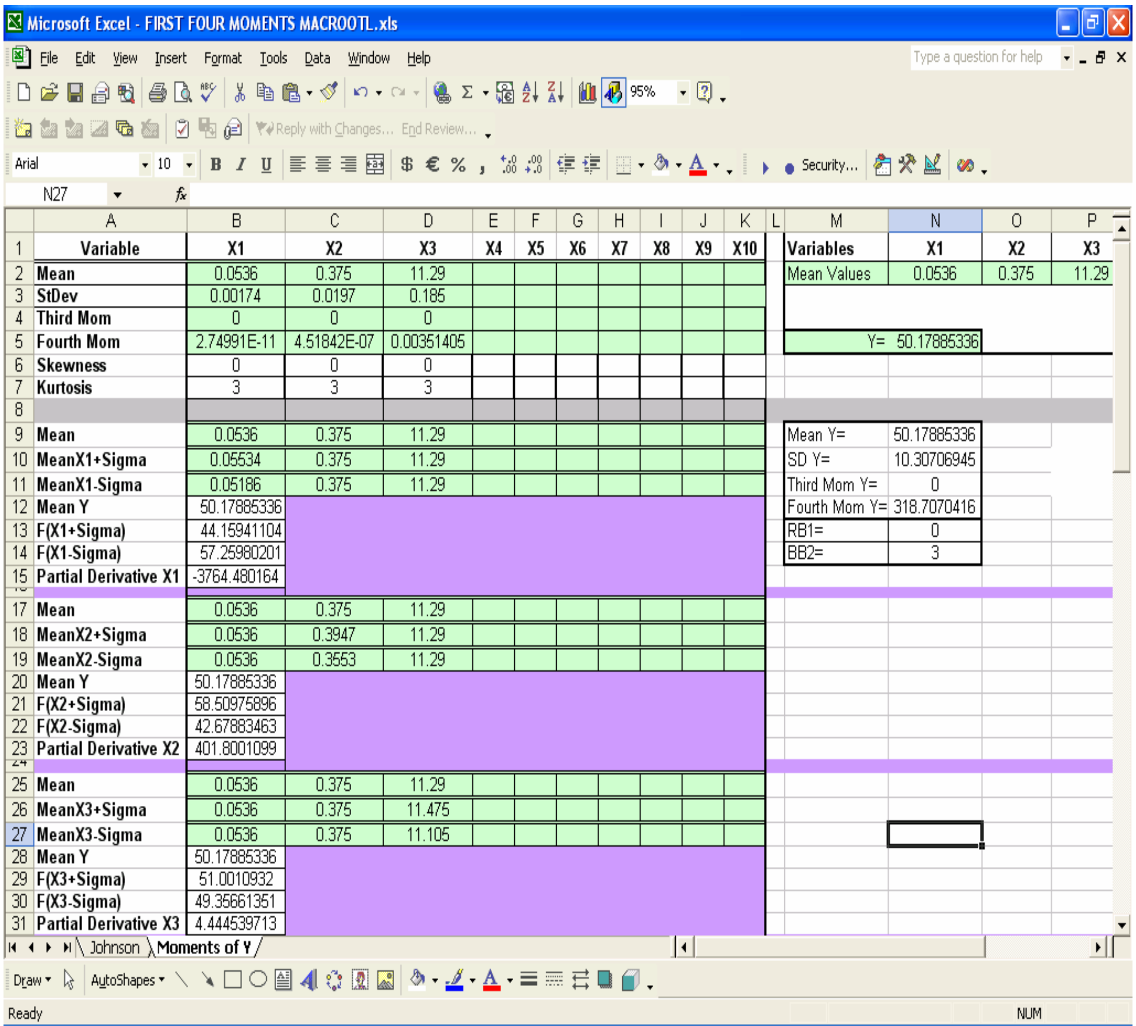
In this case, the program proceeds to determine the type of the Johnson family by methods that are discussed in Chapter 2. There will be five possible outputs from this stage, namely $S_U, S_B, S_L, S_T,$ and *Normal.* Once the type is determined, the macro begins its procedure to approximate the parameters of the respective type. If the Johnson families S_U or S_B are chosen, the program calls the corresponding subroutines for S_U or S_B respectively. In the case of S_L, *Normal*, and S_T families, the corresponding procedures are in the main body of the macro “JOHNSONTYPES”. After estimating the parameters of the Johnson distribution as appropriate, the parameters are inserted in a designated table in the worksheet “Johnson”.

At this time the last main macro is called. This macro plots the probability distribution function of the response, *Y,* as well as the specification limits on a graph.

Now the user can start trying different set points for the components. He or she can see the effect of the change on the distribution of the response for each combination of the set points. The complete script of this program is presented in Appendix 2. Figures (4-3) and (4-4) show a view of worksheets “Moments of Y” and “Johnson” for a numerical example. We include these snapshots of the screen in order to offer a view of the interactive environment, and the numerical values are not meant to correspond to any specific example that we describe here.

In Figure 4-3, the user enters the values for the mean, the standard deviation, the third moment and the fourth moment for each component in rows 2 through 5 of columns B through K. In the specific example of this schematic view, the response, *Y*, is a function of three components, x_1, x_2 and x_3. In cell N5, the user enters the closed form relationship between the response function and the components. Then the program evaluated the partial derivatives with respect to each component, through the methods described earlier.

Cells A15, A23 and A31 show the values of the partial derivatives with respect to x_1, x_2 and x_3 for this example, respectively. Then the estimated values of the mean, the standard deviation the third moment and the fourth moment of Y are calculated in cells N9 through N12, via formulas of Chapter 3. At the end, the skewness and the peakedness of Y are evaluated in cells N13 and N14 from the values of its moments.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Variable	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10		Variables	X1	X2	X3
2	Mean	0.0536	0.375	11.29									Mean Values	0.0536	0.375	11.29
3	StDev	0.00174	0.0197	0.185												
4	Third Mom	0	0	0												
5	Fourth Mom	2.74991E-11	4.51842E-07	0.00351405									Y=	50.17885336		
6	Skewness	0	0	0												
7	Kurtosis	3	3	3												
8																
9	Mean	0.0536	0.375	11.29									Mean Y=	50.17885336		
10	MeanX1+Sigma	0.05534	0.375	11.29									SD Y=	10.30706945		
11	MeanX1-Sigma	0.05186	0.375	11.29									Third Mom Y=	0		
12	Mean Y	50.17885336											Fourth Mom Y=	318.7070416		
13	F(X1+Sigma)	44.15941104											RB1=	0		
14	F(X1-Sigma)	57.25980201											BB2=	3		
15	Partial Derivative X1	-3764.480164														
17	Mean	0.0536	0.375	11.29												
18	MeanX2+Sigma	0.0536	0.3947	11.29												
19	MeanX2-Sigma	0.0536	0.3553	11.29												
20	Mean Y	50.17885336														
21	F(X2+Sigma)	58.50975896														
22	F(X2-Sigma)	42.67883463														
23	Partial Derivative X2	401.8001099														
25	Mean	0.0536	0.375	11.29												
26	MeanX3+Sigma	0.0536	0.375	11.475												
27	MeanX3-Sigma	0.0536	0.375	11.105												
28	Mean Y	50.17885336														
29	F(X3+Sigma)	51.0010932														
30	F(X3-Sigma)	49.35661351														
31	Partial Derivative X3	4.444539713														

Figure 4-3: View of “Moments of Y” worksheet.

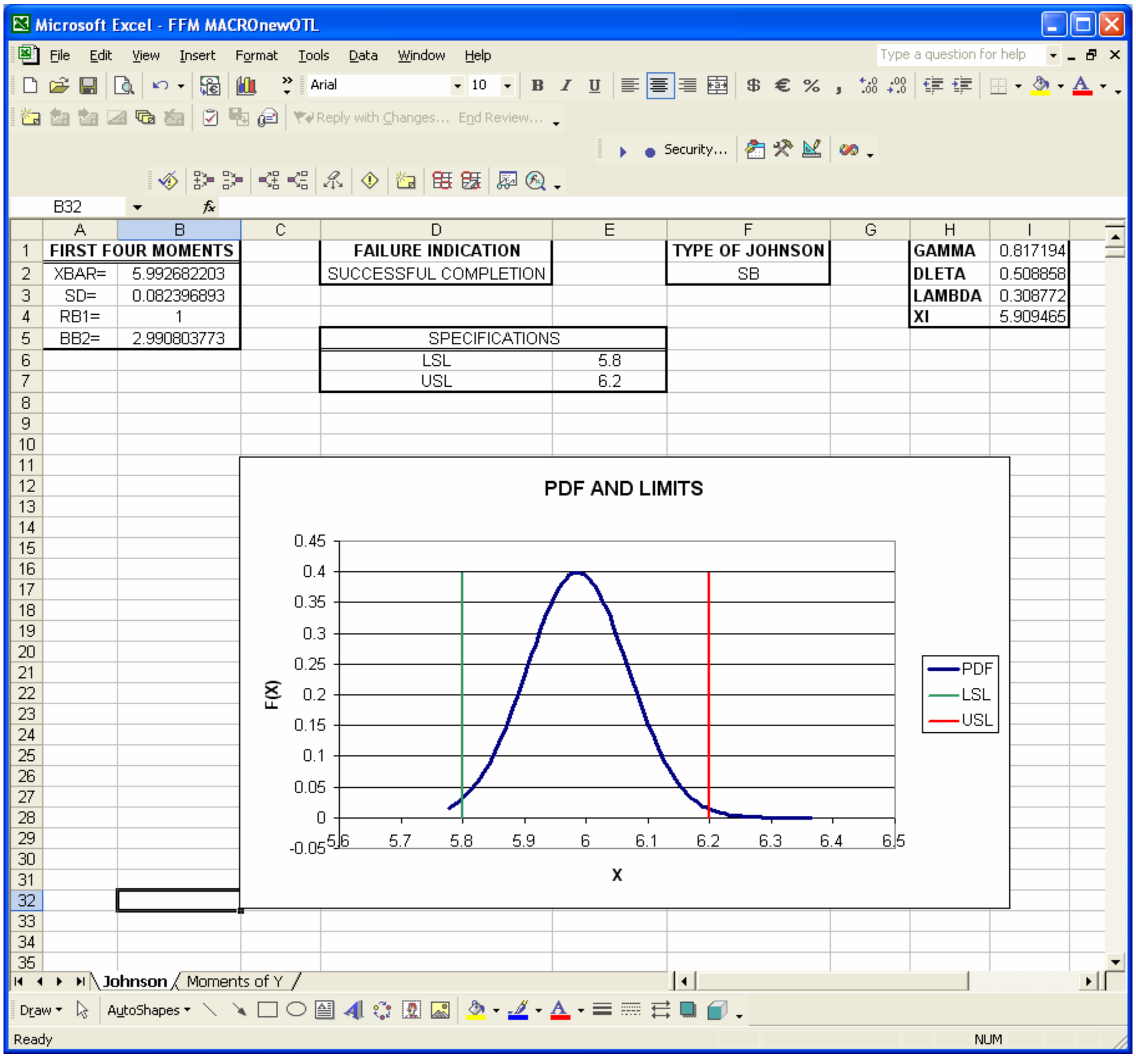


Figure 4-4: View of "Johnson" worksheet

4.3 A Numerical Example

Consider an output characteristic, Y, which is a function of three components X_1, X_2 and X_3. The response function is given as $Y = (X_1^2 + 3X_2^2)X_3$. Suppose X_1, X_2 and X_3 are normally distributed random variables with respective means μ_1, μ_2 and μ_3, and standard deviations $\sigma_1 = 0.1, \sigma_2 = 0.1$ and $\sigma_3 = 0.01$. Specification limits for Y are, 3,255 $\pm$ 150. The following are the technical constraints on μ_1, μ_2 and μ_3:

$$2 \le \mu_1 \le 10$$
$$2 \le \mu_2 \le 10$$
$$10 \le \mu_3 \le 20$$

We start with the following combination of set points: $\mu_1 = 5, \mu_2 = 8$ and $\mu_3 = 15$. Table 4-1 shows the approximated values of the mean, variance, skewness and peakedness of *Y* at these set points by Monte Carlo simulation.

Mean	3,255.6
Standard Deviation	73.479
Skewness	0.036404
Peakedness	2.9941

Table 4-1: Results of MC simulation at the initial set points

The program approximates the distribution of this function to be Johnson S_L. The fraction defective at this combination is approximately $p_0 = 3.45\%$. Figure 4-5 shows the graph of probability density function of *Y*.

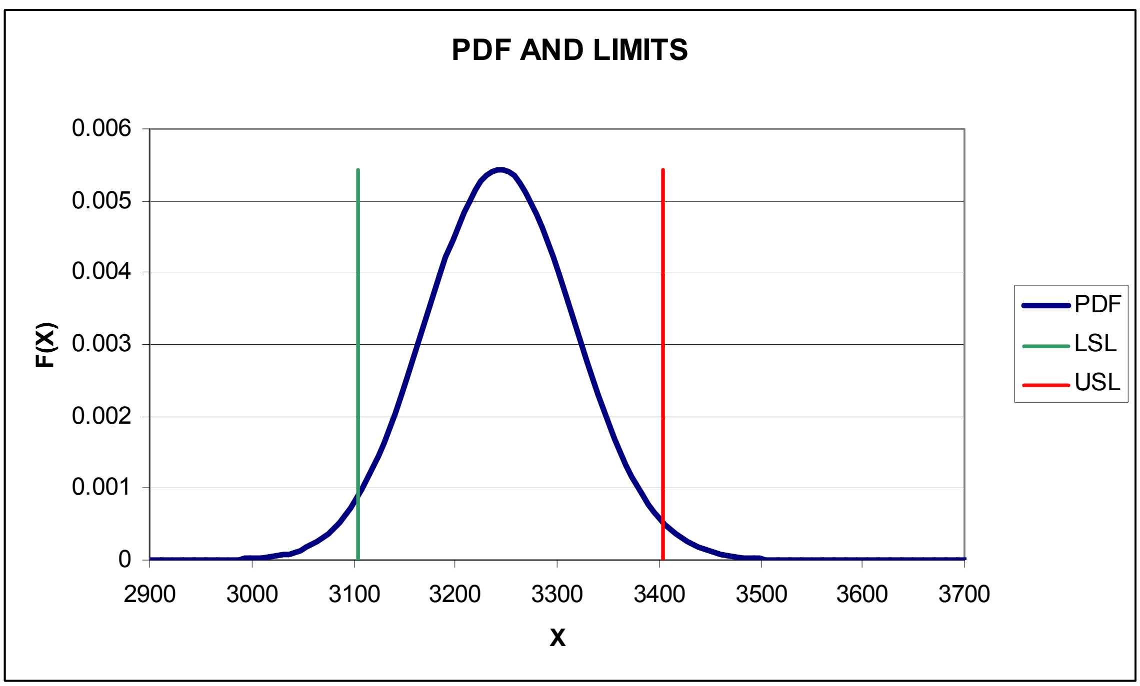


Figure 4-5: Pdf of *Y* with starting set points

We now start the interactive phase of the program and after trying a number of combinations of set points for μ_1, μ_2, and μ_3, we find the combination $\mu_1 = 7.34$, $\mu_2 = 8.0935$ and $\mu_3 = 13$ to be a better combination than the first one. For this combination Table 4-2 shows the approximated values of the mean, variance, skewness and peakedness of *Y* at the new set points. Note that the mean and variance are calculated by Taylor series method and skewness and peakedness are from the Monte-Carlo simulation at the starting combination (i.e., at the start of the interactive phase).

Mean	3,255.0677
Standard Deviation	65.9983
Skewness	0.036404
Peakedness	2.9941

Table 4-2: Results of MC simulation at the new set points

The program approximates the distribution of this function as Johnson S_L too. The corresponding value of the fraction defective for this combination is $p_1 = 2.32\%$. Figure 4-6 shows the graph of probability density function of *Y*.

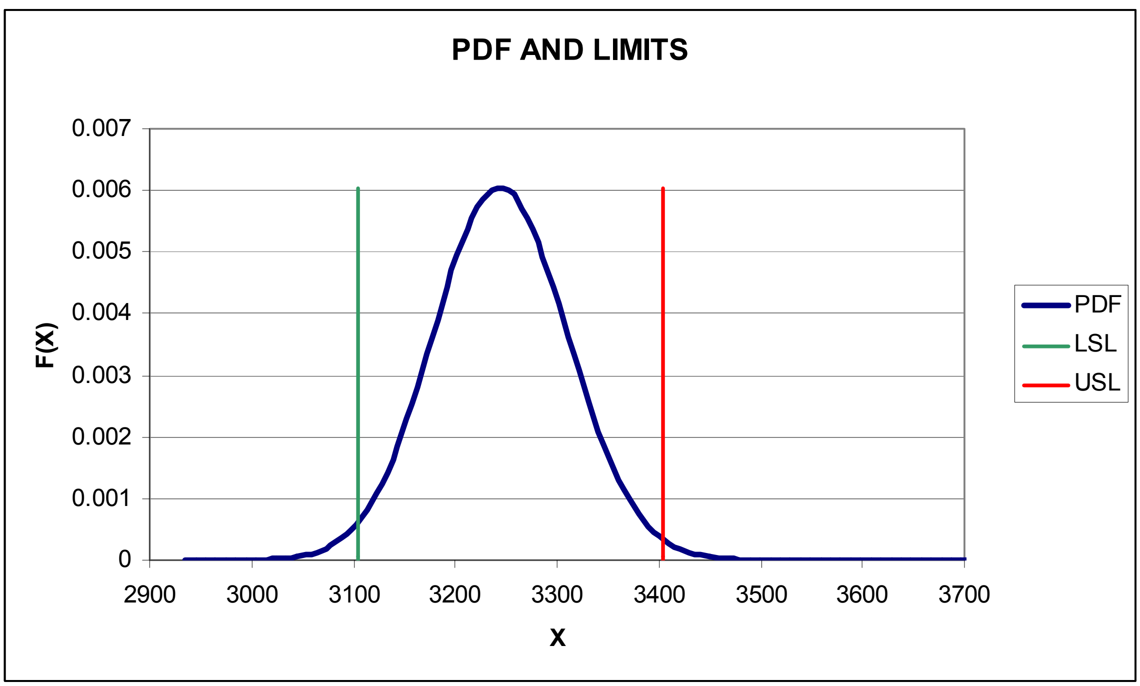


Figure 4-6: Pdf of *Y* with new set points

Now we simulate the response function with this combination of set points by Monte-Carlo simulation, and compare its first four moments with the values of Table 4-2. Table 4-3 shows the results from simulation.

Mean	3255.03
Standard Deviation	65.921
Skewness	0.032912
Peakedness	2.9932

Table 4-3: Results of MC simulation at the final stage

The fraction defective here is p_2 =2.32%. Figure 4-7 shows the probability density function of this combination with moments estimated by Monte-Carlo simulation, which of course closely resembles Figure 4-6.

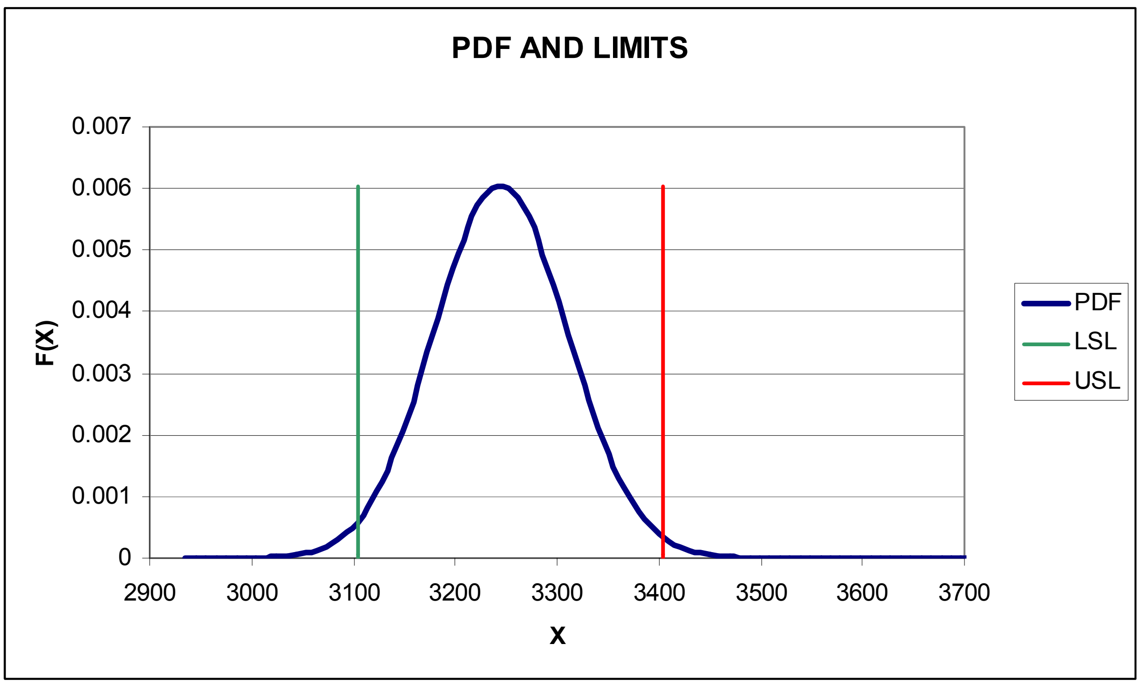


Figure 4-7: Pdf of *Y* with new set points, moments estimated by MC simulation

As it is observed, the values for the fraction defective for this new set point from both methods are equal. So, we accept the combination $\mu_1 = 7.34, \mu_2 = 8.0935$ and $\mu_3 = 13$ as the final answer. This combination keeps the mean (nominal) value of the response function almost on target and reduces its variance and the fraction defective.

Chapter 5

Case Studies

In this Chapter we present two case studies to illustrate the interactive model and the corresponding software, for the Parameter Design Problem. Throughout this Chapter all the assumptions of Chapter 1 for Parameter Design Problem hold. In each case we start with the original set points of the components and then find other set points that decrease the variation of the response function while keeping its nominal value on target. The final results are then verified with Monte-Carlo Simulation.

5.1 Deflection of a Coil Spring

5.1.1 Statement of the Problem

We have adopted this example from [6] and applied some changes to it. A detailed description of the problem can be found in [6]. Let δ denote the deflection along the axis of the spring. As discussed in [6] deflection, δ , is a function of three components, wire diameter, *d*, coil diameter, *D*, and number of active coils, *N*. The relationship is given as:

$$\delta = \frac{D^3 N}{143750 d^4}$$

We assume that each component is a normally distributed random variable with given values of its mean and standard deviation as below:

$$d \sim N(.0517, .00174^2),$$

$$D \sim N(0.357, .0197^2),$$

$$N \sim (11.29, .185^2).$$

The technical requirements on the set points of the components are:

$$0.05 \le \mu_d \le 0.20$$
$$0.25 \le \mu_D \le 1.30$$
$$2 \le \mu_N \le 15$$

We further assume that the set points of each component can be modified (i.e., it is a controllable parameter). The target value for δ is 0.50 in. and the upper and lower specification limits are given as 0.60 in. and 0.40 in. respectively.

5.1.2 Distribution of δ at the Original Set Points

We use the given values for the mean and the standard deviation of the components, and values 0 and 3 for their skewness and peakedness (the components are all normal variates). A Monte-Carlo Simulation of this function with the original set points gives the following results in Table 5-1.

Mean	0.51037
Standard Deviation	0.10995
Skewness	0.5929
Peakedness	3.631

Table 5-1: Results from Monte-Carlo Simulation at the initial set points

When these values are fed to the procedure for Johnson fitting, the result is an S_L distribution. The corresponding value of the fraction defective is $p_0 = 34.64\%$. Figure 5-1 shows the plot for this pdf. In this plot the target value of δ (0.5 in) and its specification limits (LSL=0.4 and USL=0.6) are clearly marked.

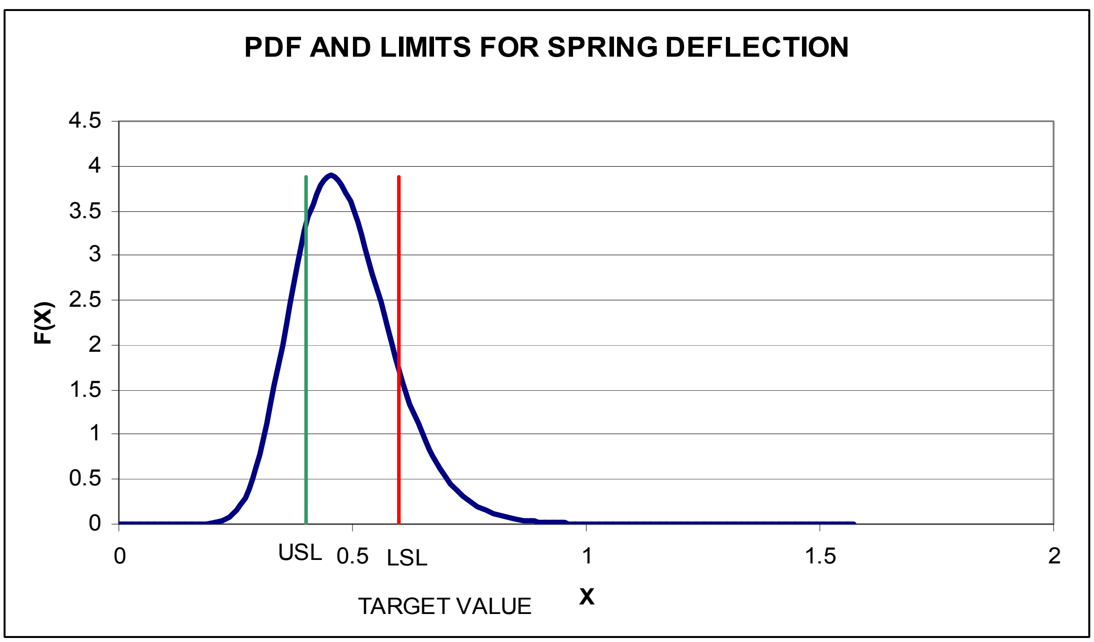


Figure 5-1: Spring Deflection case with original set points.

From Figure 5-1 we observe that the original set points for the components make the mean of the response function slightly off target (i.e., not exactly equal to 0.5 in). We also observe that there is a relatively high probability for deflection of the spring to violate its specification limits, as reflected in the corresponding value of the fraction defective. Histogram of the data set from Monte-Carlo simulation is shown in Figure 5-2 for comparison.

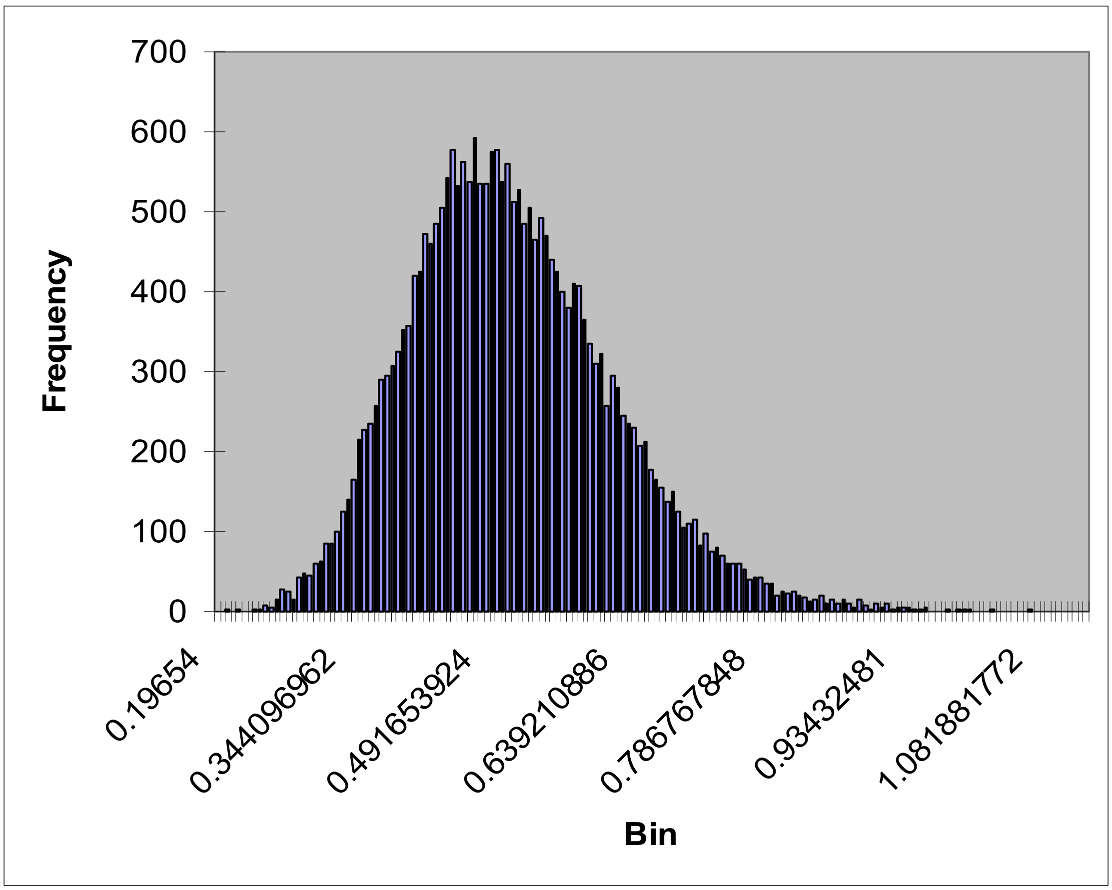


Figure 5-2: Histogram at the original set points.

5.1.3 Interactive Stage

At this stage we modify the values of each controllable parameter, μ_d, μ_D, and μ_N, and observe the impact of this modification on the pdf of δ. To choose new values for these set points such that the nominal value of δ stays on target (0.5 in), we can use the partial derivatives of the function with respect to each component (which are evaluated at the current set points), as a guide for increasing or decreasing the set points of each component. If the partial derivative with respect to a specific component is positive, it implies that increasing the value of that component results in an increase in the mean value of *Y* and if the partial derivative is negative, increasing the value of that component results in a decrease in the mean value of *Y*.

Table 5-3 shows some of the different combinations of set points that we tried, in decreasing order of their respective values of fraction defective. In this table we only present those combinations of the set points that satisfy the technical constraints and result in $\mu_\delta \approx 0.5$. We tried many other combinations of values for μ_d, μ_D, and μ_N in the context of designing the parameters interactively.

Set Points			Response				Johnson Type	Fraction Defective
μ_d	μ_D	μ_N	μ_δ	σ_δ	Skewness	Peakedness		
0.05640	0.4826	6.460	0.49919	0.08818	0.5929	3.631	S_L	24.61%
0.05642	0.4826	6.467	0.49903	0.08814	0.5929	3.631	S_L	24.60%
0.06081	0.6200	4.110	0.49832	0.07772	0.5929	3.631	S_L	18.73%
0.06500	0.7474	3.067	0.49901	0.07307	0.5929	3.631	S_L	15.99%
0.07598	0.8796	3.514	0.49917	0.06259	0.5929	3.631	S_L	9.99%
0.13660	1.3000	11.360	0.49865	0.03502	0.5929	3.631	S_L	0.80%

Table 5-2: Improvement of fraction defective for spring deflection

Some of the combinations in this table are suggested in Table 2 of [6]. Note that for all of the combinations in Table 5-2, we use the same skewness and peakedness that we found for the original combination, via Monte-Carlo simulation presented in Table 5-1 (i.e., skewness = 0.5929 and peakedness = 3.631). Figure 5-3 shows the probability density function of deflection at the best combination of set points, i.e., $\mu_d = 0.1366$, $\mu_D = 1.3$, and $\mu_N = 11.36$, with the skewness and peakedness form Monte-Carlo simulation of the original combination. The value of the fraction defective at this combination of set points is $p_1 = 0.80\%$.

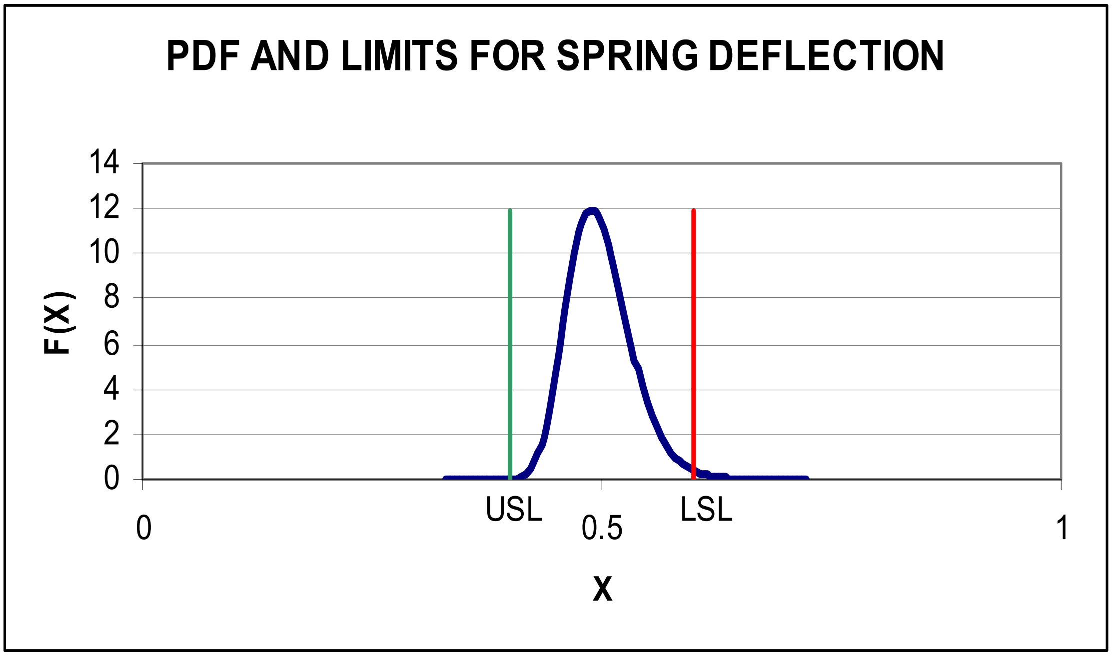


Figure 5-3: Best case with skewness and peakedness of the starting point

Now as it is explained in Chapter 4, we calculate the moments of the response function at this best combination, by Monte Carlo simulation. Table 5-3 shows the values from Monte-Carlo simulation.

Mean	0.49986
Standard Deviation	0.035071
Skewness	0.21865
Peakedness	3.0732

Table 5-3: Results from MC simulation for the best case in Table 5-2

The program fits a Johnson S_B distribution using the moments in Table 5-3. The fraction defective for this distribution is $p_2 = 0.48$ %. Figure 5-4 shows the pdf of this distribution.

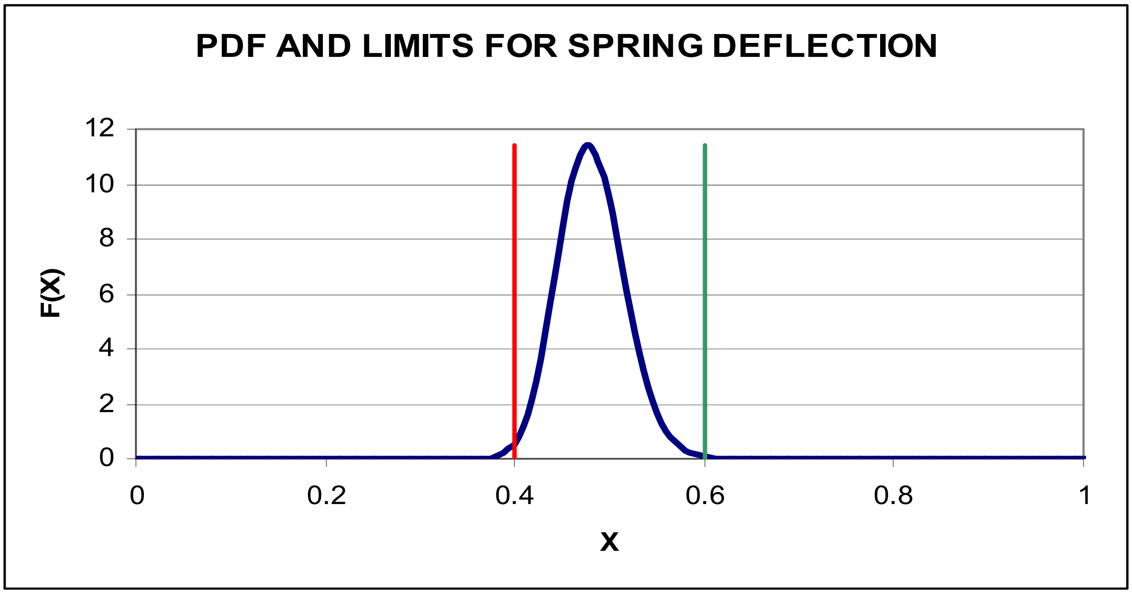


Figure 5-4: Best Case Found by Interactive Model with moments from Monte-Carlo simulation

The fraction defective p_2= 0.48%, calculated with moments from Monte-Carlo simulation, is not sufficiently close to $p_1 = 0.80\%$ (more than 5% which is our limiting value for termination as described earlier). Comparison between Figures 5-3 and 5-4 shows a difference in the shape of these two probability density functions as well. This discrepancy comes from the difference in skewness and peakedness used in each case. Therefore, we take this best case combination as a new starting point and repeat the interactive phase. We keep the values for skewness and peakedness that are found by Monte-Carlo simulation, i.e., skewness= 0.21865 and peakedness= 3.0732, for all of the combinations that we try, and we calculate their mean and variance by Taylor series approximation. Table 5-4 shows some of these trial combinations.

Set Points			Response				Johnson Type	Fraction Defective
μ_d	μ_D	μ_N	μ_δ	σ_δ	Skewness	Peakedness		
0.1360	1.34	10.3	0.50395	0.03524	0.21865	3.0723	S_B	0.59%
0.1365	1.35	10.2	0.50288	0.03502	0.21865	3.0723	S_B	0.52%
0.1365	1.40	9.1	0.50036	0.03466	0.21865	3.0723	S_B	0.43%

Table 5-4: Improvement of fraction defective in spring deflection with a new starting point

As it is shown in this table, the last combination results in the least fraction defective compared to other combinations that we have tried. The corresponding value of fraction defective at this combination is $p_1 = 0.43\%$. Figure 5-5 shows the probability density function for this combination.

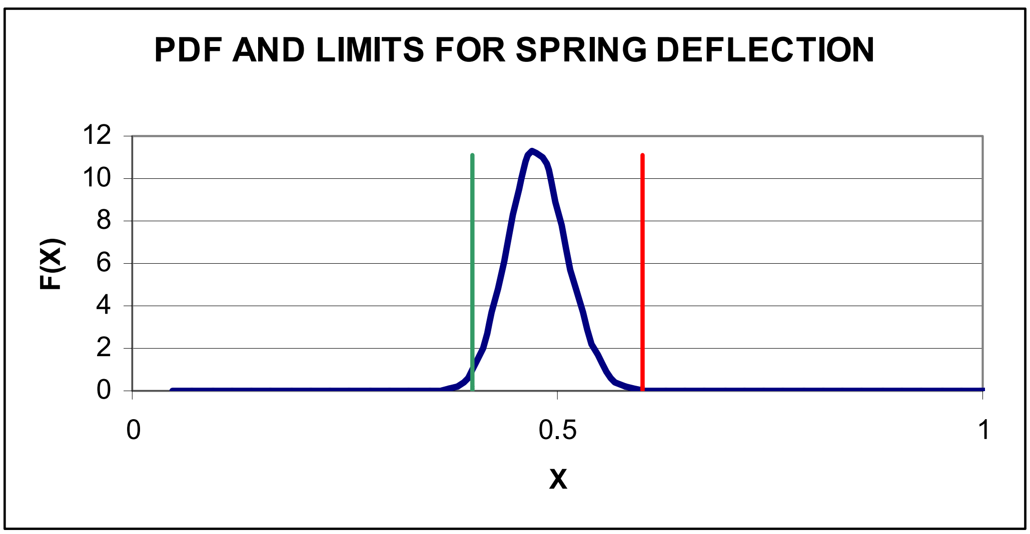


Figure 5-5: Best case with skewness and peakedness of the starting point

Monte-Carlo simulation for this combination gives the following results:

Mean	0.50146
Standard Deviation	0.034773
Skewness	0.21883
Peakedness	3.0719

It is observed that the skewness and peakedness from the Monte-Carlo simulation are relatively close to the skewness and peakedness of the new starting point. The corresponding value of the fraction defective is p_2 = 0.45%. Since this value is within 5% of the value of p_1, we can stop here and keep this combination as an acceptable solution.

5.2 An Output Transformerless Pull-Push Circuit

5.2.1 Statement of the Problem

This example deals with an Output Transformerless (OTL) Pull-Push Circuit. A graphical representation of this circuit is given in Figure 5-6. We have adopted this example form [6] with slight modifications.

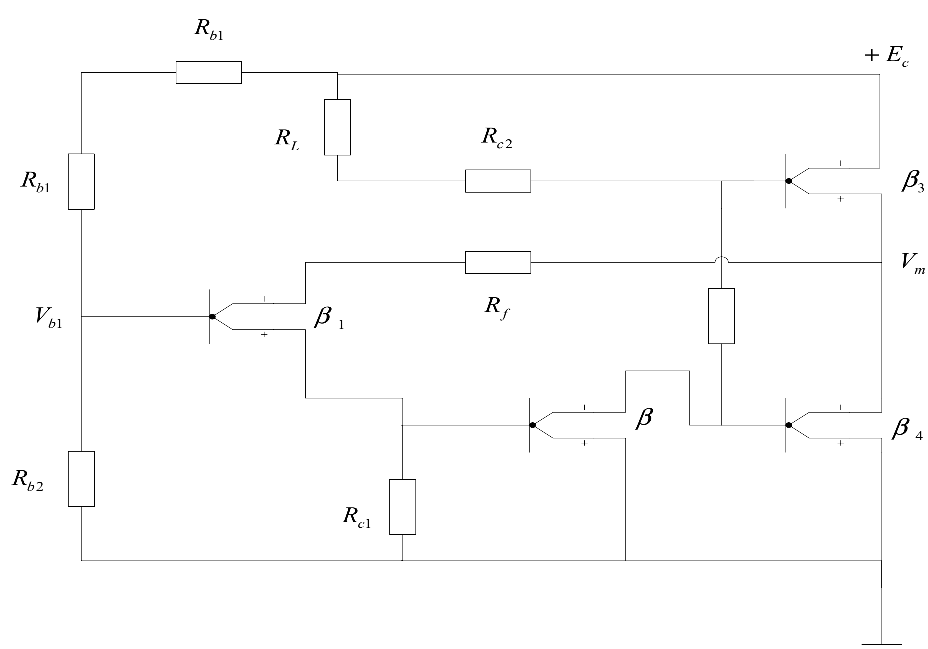


Figure 5-6: Graphical representation of the OTL pull-push circuit

The performance characteristic of interest is V_m and the response function is given by

$$V_m = (V_{b1} + V_{be1})\frac{\beta R_\circ}{\beta R_\circ + R_f} + (E_c - V_{be3})\frac{R_f}{\beta R_\circ + R_f} + \frac{V_{be2} R_f \beta R_\circ}{(\beta R_\circ + R_f) R_{c1}}$$

where

$$V_{b1} = \frac{E_c R_{b2}}{R_{b1} + R_{b2}} \qquad \text{and} \qquad R_\circ = R_{c2} + R_L$$

with $R_L = 9 Ohms$, $V_{be1} = V_{be3} = 0.65\ Volts$, $V_{be2} = 0.74\ Volts$, and $E_c = 12\ Volts$. $R_{b2}, R_{b1}, R_f, R_{c2}$, and R_{c1} are resistances, and β is the current gain of the transistor. The means of these components are denoted by μ_1 through μ_6 and standard deviations are denoted by σ_1 through σ_6, respectively. Parameters μ_1 through μ_6 are controllable. All the components are assumed to be normally distributed. The technical requirements of μ_1 through μ_6 are as below:

$$25000 \le \mu_1 \le 70000$$

$$50000 \le \mu_2 \le 150000$$

$$649.4 \le \mu_3 \le 2053.5$$

$$237.1 \le \mu_4 \le 749.9$$

$$1271.1 \le \mu_5 \le 2260.3$$

$$73 \le \mu_6 \le 280$$

Target value for V_m is *6 Volts*. We further assume that the specification limits for V_m are 6 ± 0.2.

The original set points and the standard deviations of the components are given as below.

$$\mu_1 = 48930, \mu_2 = 117100, \mu_3 = 2054, \mu_4 = 237.1, \mu_5 = 1271, \text{ and } \mu_6 = 73$$

$$\sigma_i = 0.02\mu_i \quad \text{for } i = 1,...,5, \text{ and } \sigma_6 = 0.15\mu_6.$$

5.2.2 Variation with the Original Set Points

With the original set points (mean values) and standard deviations as above and values 0 and 3 for the skewness and peakedness of the components (resulting form their normality), Monte-Carlo simulation gives the values for the mean, standard deviation, skweness and peakedness of V_m as in Table 5-5.

Mean	6.0073
Standard Deviation	0.11652
Skewness	0.4208
Peakedness	3.6557

Table 5-5: Results from Monte-Carlo Simulation for the original set points

When the distribution of V_m is approximated with a Johnson distribution through our model, the resulting pdf is Johnson S_U. Figure 5-7 shows this pdf.

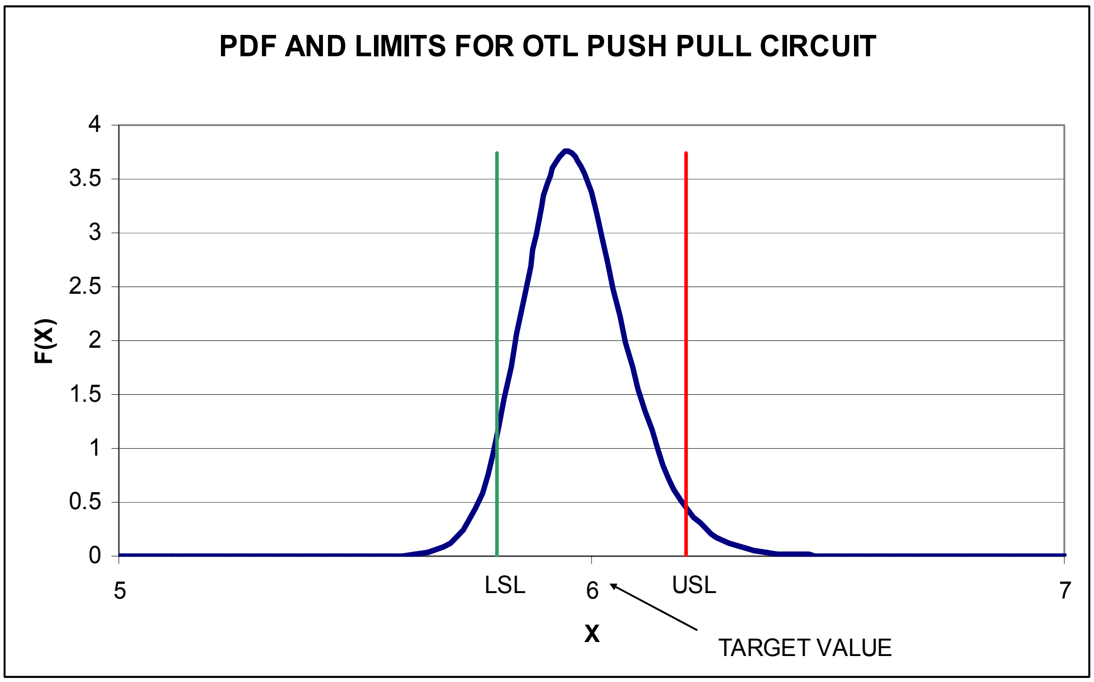


Figure 5-7: OTL Circuit with original set points

We observe that the mean of V_m is very close to its target value (i.e., $\mu_{V_m} \approx 6$) but the probability that V_m violates its specification limits is considerably high. The corresponding value of the fraction defective is p_0 = 8.27%. The resulting histogram is shown in Figure 5-8.

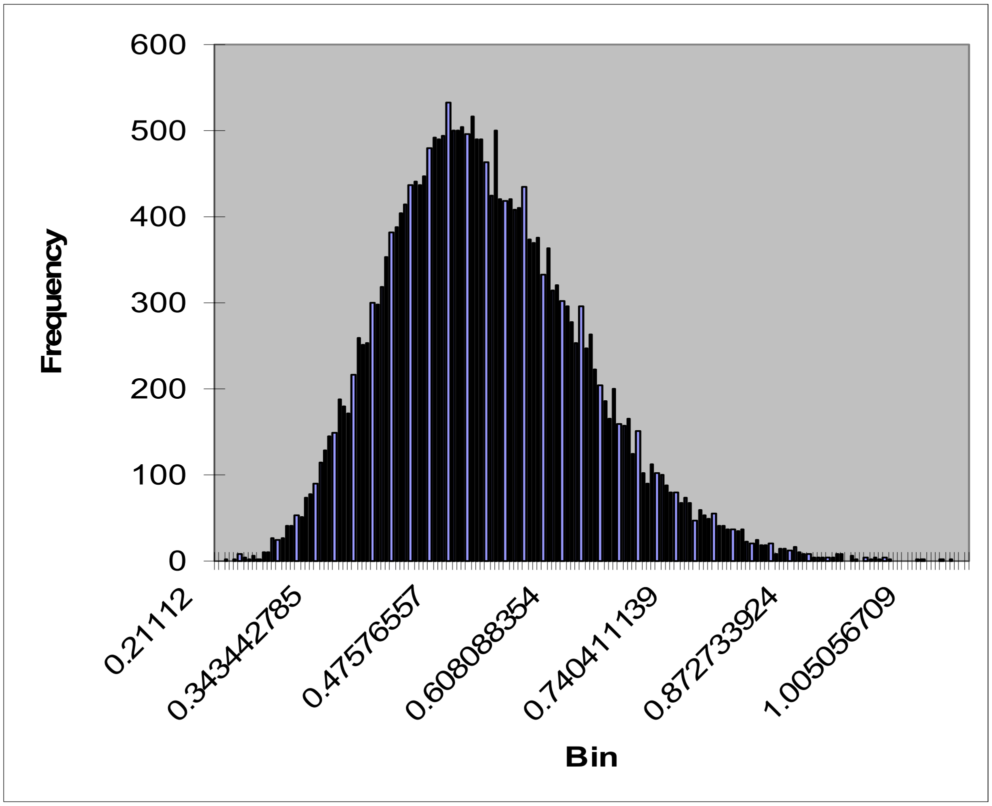


Figure 5-8: Histogram at the original set points.

5.2.3 Interactive Stage

At this stage we try several combinations of set points. Some of these combinations are taken form Table 4 in [6]. Table 5-6 shows those combinations that result in reasonably lower values of fraction defective, compared to the original case. These combinations are shown in decreasing order of their respective fraction defective values. Note that we have used the skewness and peakedness from the Monte-Carlo simulation of the starting combination.

Set Points						Response				Johnson Types	Fraction Defective
μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	Mean	Standard Deviation	Skewness	Peakedness		
56500	87000	1044	500	1275	280	6.0197	0.08255	0.4208	3.6557	S_U	2.54 %
55000	85000	1044	500	1275	280	6.0093	0.08249	0.4208	3.6557	S_U	2.17 %
55835	75000	649.4	749.9	2260.3	280	6.0000	0.08306	0.4208	3.6557	S_U	2.00 %
56590	87010	1044	746.2	1302	280	5.9987	0.0825	0.4208	3.6557	S_U	1.92 %
58750	100000	1539.9	237.1	1467.8	280	5.9871	0.0828	0.4208	3.6557	S_U	1.76 %

Table 5-6: Improvement in fraction defective

In this table, the best combination of set points appears in the last row, i.e., $\mu_1 = 58750$, $\mu_2 = 100000, \mu_3 = 1539.9, \mu_4 = 237.1, \mu_5 = 1467.8$, and $\mu_6 = 280$. The corresponding value of the fraction defective is p_1 = 1.76%.

Now we conduct a Monte-Carlo simulation with the set points of the best case combination. Table 5-7 shows the results and Figure 5-9 shows the fitted Johnson S_B distribution.

Mean	6.011
Standard Deviation	0.08267
Skewness	0.02016
Peakedness	2.9879

Table 5-7: Results from Monte-Carlo Simulation for the best case set points

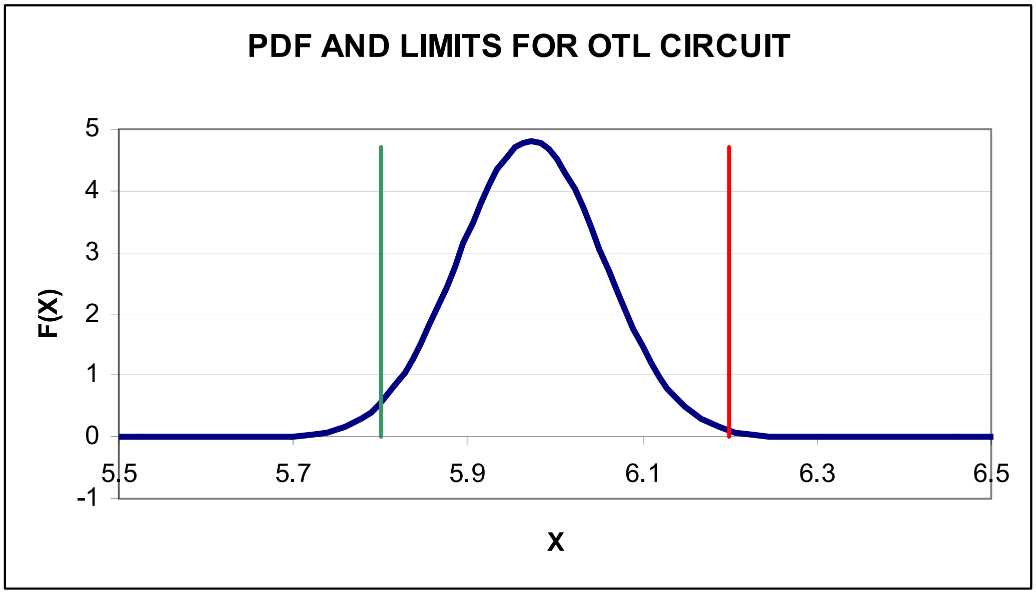


Figure 5-9: Fitted Johnson distribution of the best case set points.

The corresponding values of the fraction defective for this distribution is p_2 = 1.65%. Since the value p_2 is not within 5% of p_1, we resume the interactive stage with updated values of the skewness and peakedness as 0.02016 and 2.9879, respectively. We use this combination as a new starting point and resume trying different combinations. Table 5-8 shows the results of this trial.

Set Points						Response				Johnson Types	Fraction Defective
μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	Mean	Standard Deviation	Skewness	Peakedness		
55500	85500	1044	500	1275	280	6.0184	0.08255	0.02016	2.9879	S_B	1.78%
55000	85500	1044	500	1275	280	5.9927	0.08239	0.02016	2.9879	S_B	1.54%
55500	85000	1044	600	1350	280	5.9953	0.08246	0.02016	2.9879	S_B	1.52%
40000	62000	1044	600	1270	280	5.9969	0.08247	0.02016	2.9879	S_B	1.51%

Table 5-8: Improving variation with a new starting combination

In Table 5-8, the combinations are sorted in decreasing order of their fraction defective. As before, the skewness and peakedness from the Monte-Carlo simulation of the new starting combination (i.e., from Table 5-7) are used as the skewness and the peakedness at all combinations. The combination with the least fraction defective is the combination at the last row of Table 5-8, with value of $p_1 = 1.51\%$

Now we conduct a Monte-Carlo simulation with the last combination in this table. The results are shown in Table 5-9.

Mean	5.994
Standard Deviation	0.08237
Skewness	0.01889
Peakedness	3.003

Table 5-9: Results from Monte-Carlo simulation of the best case

We can observe that the values of skewness and peakedness by Monte-Carlo simulation are reasonably close to the values in Table 5-7 for the best case combination. The corresponding value of the fraction defective is p_2= 1.55%. Since this value is very close to the value of p_1 = 1.51% (within 5% of p_1), we accept the corresponding combination as the combination that results in the least fraction defective. The set points for this combination are $\mu_1 = 40000, \mu_2 = 62000, \mu_3 = 1044, \mu_4 = 600, \mu_5 = 1270, \mu_6 = 280$.

Chapter 6

Summary of Findings and Avenues for Further Research

In this Chapter we discuss our findings from solving Parameter Design Problems with this interactive framework. We also discuss avenues for further research.

6.1. Summary of the Proposed Model

In the Parameter Design Problem, as we define in Chapter 1, we want to determine a set of values for $\mu_1,...,\mu_n$, such that the mean of *Y* remains at (or close to) its target value τ and its variation is minimized. To solve this problem we propose a method by which the user can manually change the values of $\mu_1,...,\mu_n$ and observe the effect of this change on the variation of *Y*. To build this model, we approximate the probability distribution of *Y* by a Johnson distribution as we discuss in Chapter 2. To find the required inputs (the first four moments of Y) for fitting a Johnson distribution, we use Taylor series approximation method or Monte-Carlo simulation as appropriate and as we explain in Chapter 3. We have developed a computer program that we can use to solve the Parameter Design Problems by this method. This program is discussed in Chapter 4. In Chapter 5 we solve two case studies using this model. In this Chapter we discuss our findings, and offer several suggestions that can help improving this model.

6.2 Findings

As it is shown in Chapter 5 the results form this model conform to the results of other models for solving Parameter Design Problems e.g., non-linear programming. However, with this model the user can visually observe the probability distribution function of *Y* at each combination of set points for $\mu_1,...,\mu_n$ and compare different combinations with each other. We no longer have to assume that *Y* has a normal probability distribution function. We can also calculate the value of the fraction defective more accurately.

In solving different cases with this model, we find that it will be very useful if we have a guide for changing our set points. For example, it is useful to know that increasing which set points lead to increase in mean of *Y*. In our case studies in Chapter 5, we used the partial derivatives of the function h with respect to each variable, $x_1,...,x_n$ for this purpose.

We also find that by adding some new features to this model, we can have an interactive framework for solving Parameter Design Problems. In the next section we offer our suggestions for these new features.

6.3 Avenues for Further Research

This model is the building block of an interactive framework that a designer can use to implement different methods of solving the Parameter Design Problem. Other features may be easily added to this framework in order to facilitate and enhance its capacity as described below.

Calculating Measures of Variation

We can add a module to this model that calculates measures of variation such as "expected loss". Performance measures such as C_P or C_{PK} can also be estimated for each combination. Estimating these measures will help the designer in comparing different combinations form several points of views.

A Systemic Method for Changing the Set Points

As we discussed earlier, it is very helpful if we have a guide for changing the set points and finding new feasible combinations. We can add a module to the existing model that suggests new combinations to the user. This feature can be designed such that set points are changed systematically. Then the user can compare the different combinations that are recommended by the program, by observing the corresponding probability distribution functions on the screen.

A Nonlinear Programming Module

A useful feature to be added to this framework is a module that employs an appropriate optimization technique for finding an optimal set of values for $\mu_1,...,\mu_n$. The objective function of this model can be the loss function, the fraction defective, or any other measure that is appropriate for the user. If the respective model is a standard nonlinear programming model, then we can potentially use Excel Solver to analyze the model.

Statistical Models

A module for solving the Parameter Design Problem using statistical methods can also be added to this framework. In this part we can use the techniques of design of experiments to find an optimal set of values for the design variables of a Parameter Design Problem.

Finding a More Accurate Way of Approximating the Moments

One avenue for further research can be on the subject of approximating the first four moments of the variate Y, by a more accurate method. Such a method helps to enhance the overall performance of the interactive algorithm.

As a first step towards this goal, for several examples we tried using the extended versions of the formulas of Chapter 3 for estimating the moments of Y [i.e., formulas (3-7) through (3-10)] as given in [5]. However, the approximated values that we obtained for the third and the fourth moments in this manner were still not sufficiently accurate for the purpose of this algorithm.

References

[1] Box, G.E.P., "Signal-to-noise ratios, performance criteria and transformations" (with discussion), Technometrics 30 (1988) 1-17.

[2] Box, G.E.P., and Fung, C.A., "Studies in quality improvement: Minimizing the transmitted variation by parameter design", Report No.8, Center for Quality and Productivity Improvement, University of Wisconsin, Madison, WI, 1986.

[3] Draper, J., "Properties of distributions resulting from certain simple transformations of the normal distribution" M.Sc. thesis, University of London.

[4] Elderton, W.P., and Johnson, N.L., *Systems of Frequency Curves*, Cambridge University Press, 1969.

[5] Evans, David H., "Statistical Tolerancing: The State of the Art: Part II. Methods for Estimating Moments", *Journal of Quality Technology* 7 (1975).

[6] Fathi, Y., "A linear approximation model for the parameter design problem", *European Journal of Operational Research* 97 (1997) 561-570.

[7] Fathi, Y., "A nonlinear programming approach to the parameter design problem", *European Journal of Operational Research* 53 (1991) 371-381.

[8] Hahn, G.J., and Shapiro, S.S., *Statistical Models in Engineering,* Wiley, New York, 1967.

[9] Hill, I.D., Hill, R., and Holder, R.L., "Fitting Johnson Curves by Moments", *Applied Statistics*, 25 (1976) 180-189.

[10] Johnson, N.L., Nixon, E., and Amos, D.E., "Table of percentage points of Pearson curves, for given $\sqrt{\beta_1}$ and β_2, expressed in standard measure", *Biometrika* 50 (1963) 459-498.

[11] Johnson, N.L., "Systems of Frequency Curves Generated by Methods of Translation" *Biometrika* 36 (1949) 149-176.

[12] Leon, R., Shoemaker, A.C., and Kackar, R.N., "Performance measures independent of adjustment" (with discussion), *Technometrics* 29 (1987) 253-285.

[13] Taguchi, G., and Wu, Y., "Introduction to off-line quality control", Central Japan Quality Control Association, 1980.

[14] Tukey, John W., "Propagation of Errors, Fluctuations and Tolerances, No. 1: Basic Generalized Formulas", Technical Report No.10, Princeton University, Princeton, NJ, 1957.

Appendix 1

Script of the Computer Program Developed by Hill et al. [9]

Acknowledgements

We are indebted to J. Draper for permission to make use of his M.Sc. thesis, and to the referee for many useful comments.

References

Draper, J. (1951). Properties of distributions resulting from certain simple transformations of the normal distribution. M.Sc. thesis, University of London.

—— (1952). Properties of distributions resulting from certain simple transformations of the normal distribution. *Biometrika*, **39**, 290–301.

Elderton, W. P. and Johnson, N. L. (1969). *Systems of Frequency Curves*. Cambridge: University Press.

Johnson, N. L. (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, **36**, 149–176.

Ord, J. K. (1972). *Families of Frequency Distributions*. London: Griffin.

Pearson, E. S. and Hartley, H. O. (1972). *Biometrika Tables for Statisticians*, Vol. 2. Cambridge: University Press.


```
      SUBROUTINE JNSN(XBAR, SD, RB1, BB2, ITYPE, GAMMA, DELTA, XLAM, XI,
     *  IFAULT)
C
C         ALGORITHM AS 99  APPL. STATIST. (1976) VOL.25, NO.2
C
C         FINDS TYPE AND PARAMETERS OF A JOHNSON CURVE
C         WITH GIVEN FIRST FOUR MOMENTS
C
      REAL XBAR, SD, RB1, BB2, GAMMA, DELTA, XLAM, XI, TOL, B1, B2, Y,
     *  X, U, W, ZERO, ONE, TWO, THREE, FOUR, HALF, QUART
      LOGICAL FAULT
      DATA TOL /0.01/
      DATA ZERO, ONE, TWO, THREE, FOUR, HALF, QUART
     *    /0.0, 1.0, 2.0, 3.0, 4.0, 0.5, 0.25/
      IFAULT = 1
      IF (SD .LT. ZERO) RETURN
      IFAULT = 0
      XI = ZERO
      XLAM = ZERO
      GAMMA = ZERO
      DELTA = ZERO
      IF (SD .GT. ZERO) GOTO 10
      ITYPE = 5
      XI = XBAR
      RETURN
   10 B1 = RB1 * RB1
      B2 = BB2
      FAULT = .FALSE.
C
C         TEST WHETHER LOGNORMAL (OR NORMAL) REQUESTED
C
      IF (B2 .GE. ZERO) GOTO 30
   20 IF (ABS(RB1) .LE. TOL) GOTO 70
      GOTO 80
C
C         TEST FOR POSITION RELATIVE TO BOUNDARY LINE
C
   30 IF (B2 .GT. B1 + TOL + ONE) GOTO 60
      IF (B2 .LT. B1 + ONE) GOTO 50
C
C         ST DISTRIBUTION
C
   40 ITYPE = 5
      Y = HALF + HALF * SQRT(ONE - FOUR / (B1 + FOUR))
      IF (RB1 .GT. ZERO) Y = ONE - Y
      X = SD / SQRT(Y * (ONE - Y))
      XI = XBAR - Y * X
      XLAM = XI + X
      DELTA = Y
      RETURN
   50 IFAULT = 2
      RETURN
```

```
   60 IF (ABS(RB1) .GT. TOL .OR. ABS(B2 - THREE) .GT. TOL) GOTO 80
C
C         NORMAL DISTRIBUTION
C
   70 ITYPE = 4
      DELTA = ONE / SD
      GAMMA = -XBAR / SD
      RETURN
C
C         TEST FOR POSITION RELATIVE TO LOGNORMAL LINE
C
   80 U = ONE / THREE
      X = HALF * B1 + ONE
      Y = RB1 * SQRT(QUART * B1 + ONE)
      W = (X + Y) ** U + (X - Y) ** U - ONE
      U = W * W * (THREE + W * (TWO + W)) - THREE
      IF (B2 .LT. ZERO .OR. FAULT) B2 = U
      X = U - B2
      IF (ABS(X) .GT. TOL) GOTO 90
C
C         LOGNORMAL (SL) DISTRIBUTION
C
      ITYPE = 1
      XLAM = SIGN(ONE, RB1)
      U = XLAM * XBAR
      X = ONE / SQRT(ALOG(W))
      DELTA = X
      Y = HALF * X * ALOG(W * (W - ONE) / (SD * SD))
      GAMMA = Y
      XI = U - EXP((HALF / X - Y) / X)
      RETURN
C
C         SB OR SU DISTRIBUTION
C
   90 IF (X .GT. ZERO) GOTO 100
      ITYPE = 2
      CALL SUFIT(XBAR, SD, RB1, B2, GAMMA, DELTA, XLAM, XI)
      RETURN
  100 ITYPE = 3
      CALL SBFIT(XBAR, SD, RB1, B2, GAMMA, DELTA, XLAM, XI, FAULT)
      IF (.NOT.FAULT) RETURN
C
C         FAILURE - TRY TO FIT APPROXIMATE RESULT
C
      IFAULT = 3
      IF (B2 .GT. B1 + TWO) GOTO 20
      GOTO 40
      END
C
      SUBROUTINE SUFIT(XBAR, SD, RB1, B2, GAMMA, DELTA, XLAM, XI)
C
C         ALGORITHM AS 99.1  APPL. STATIST. (1976) VOL.25, NO.2
C
C         FINDS PARAMETERS OF JOHNSON SU CURVE WITH
C         GIVEN FIRST FOUR MOMENTS
C
      REAL XBAR, SD, RB1, B2, GAMMA, DELTA, XLAM, XI, TOL, B1, B3, W, Y,
     *  W1, WM1, Z, V, A, B, X, ZERO, ONE, TWO, THREE, FOUR, SIX,
     *  SEVEN, EIGHT, NINE, TEN, HALF, ONE5, TWO8
      DATA TOL /0.01/
      DATA ZERO, ONE, TWO, THREE, FOUR, SIX, SEVEN, EIGHT, NINE, TEN,
     *  HALF, ONE5, TWO8 /0.0, 1.0, 2.0, 3.0, 4.0, 6.0, 7.0, 8.0, 9.0,
     *  10.0, 0.5, 1.5, 2.8/
      B1 = RB1 * RB1
      B3 = B2 - THREE
C
C         W IS FIRST ESTIMATE OF EXP(DELTA ** (-2))
C
      W = SQRT(SQRT(TWO * B2 - TWO8 * B1 - TWO) - ONE)
      IF (ABS(RB1) .GT. TOL) GOTO 10
```

```
C          SYMMETRICAL CASE - RESULTS ARE KNOWN
C
      Y = ZERO
      GOTO 20
C
C          JOHNSON ITERATION (USING Y FOR HIS M)
C
   10 W1 = W + ONE
      WM1 = W - ONE
      Z = W1 * B3
      V = W * (SIX + W * (THREE + W))
      A = EIGHT * (WM1 * (THREE + W * (SEVEN + V)) - Z)
      B = 16.0 * (WM1 * (SIX + V) - B3)
      Y = (SQRT(A * A - TWO * B * (WM1 * (THREE + W *
     *  (NINE + W * (TEN + V))) - TWO * W1 * Z)) - A) / B
      Z = Y * WM1 * (FOUR * (W + TWO) * Y + THREE * W1 * W1) ** 2 /
     *  (TWO * (TWO * Y + W1) ** 3)
      V = W * W
      W = SQRT(SQRT(ONE - TWO * (ONE5 - B2 + (B1 *
     *  (B2 - ONE5 - V * (ONE + HALF * V))) / Z)) - ONE)
      IF (ABS(B1 - Z) .GT. TOL) GOTO 10
C
C          END OF ITERATION
C
      Y = Y / W
      Y = ALOG(SQRT(Y) + SQRT(Y + ONE))
      IF (RB1 .GT. ZERO) Y = -Y
   20 X = SQRT(ONE / ALOG(W))
      DELTA = X
      GAMMA = Y * X
      Y = EXP(Y)
      Z = Y * Y
      X = SD / SQRT(HALF * (W - ONE) * (HALF * W * (Z + ONE / Z) + ONE))
      XLAM = X
      XI = (HALF * SQRT(W) * (Y - ONE / Y)) * X + XBAR
      RETURN
      END
C
      SUBROUTINE SBFIT(XBAR, SIGMA, RTB1, B2, GAMMA, DELTA, XLAM, XI,
     *  FAULT)
C
C          ALGORITHM AS 99.2  APPL. STATIST. (1976) VOL.25, NO.2
C
C          FINDS PARAMETERS OF JOHNSON SB CURVE WITH
C          GIVEN FIRST FOUR MOMENTS
C
      REAL HMU(6), DERIV(4), DD(4), XBAR, SIGMA, RTB1, B2, GAMMA, DELTA,
     *  XLAM, XI, TT, TOL, RB1, B1, E, U, X, Y, W, F, D, G, S, H2, T,
     *  H2A, H2B, H3, H4, RBET, BET2, ZERO, ONE, TWO, THREE, FOUR, SIX,
     *  HALF, QUART, ONE5
      LOGICAL NEG, FAULT
      DATA TT, TOL, LIMIT /1.0E-4, 0.01, 50/
      DATA ZERO, ONE, TWO, THREE, FOUR, SIX, HALF, QUART, ONE5
     *  /0.0, 1.0, 2.0, 3.0, 4.0, 6.0, 0.5, 0.25, 1.5/
      RB1 = ABS(RTB1)
      B1 = RB1 * RB1
      NEG = RTB1 .LT. ZERO
C
C          GET D AS FIRST ESTIMATE OF DELTA
C
      E = B1 + ONE
      U = ONE / THREE
      X = HALF * B1 + ONE
      Y = RB1 * SQRT(QUART * B1 + ONE)
      W = (X + Y) ** U + (X - Y) ** U - ONE
      F = W * W * (THREE + W * (TWO + W)) - THREE
      E = (B2 - E) / (F - E)
      IF (ABS(RB1) .GT. TOL) GOTO 5
      F = TWO
      GOTO 20
    5 D = ONE / SQRT(ALOG(W))
```

```
      IF (D .LT. 0.04) GOTO 10
      F = TWO - 8.5245 / (D * (D * (D - 2.163) + 11.346))
      GOTO 20
   10 F = 1.25 * D
   20 F = E * F + ONE
      IF (F .LT. 1.8) GOTO 25
      D = (0.626 * F - 0.408) * (THREE - F) ** (-0.479)
      GOTO 30
   25 D = 0.8 * (F - ONE)
C
C        GET G AS FIRST ESTIMATE OF GAMMA
C
   30 G = ZERO
      IF (B1 .LT. TT) GOTO 70
      IF (D .GT. ONE) GOTO 40
      G = (0.7466 * D ** 1.7973 + 0.5955) * B1 ** 0.485
      GOTO 70
   40 IF (D .LE. 2.5) GOTO 50
      U = 0.0124
      Y = 0.5291
      GOTO 60
   50 U = 0.0623
      Y = 0.4043
   60 G = B1 ** (U * D + Y) * (0.9281 + D * (1.0614 * D - 0.7077))
   70 M = 0
C
C        MAIN ITERATION STARTS HERE
C
   80 M = M + 1
      FAULT = M .GT. LIMIT
      IF (FAULT) RETURN
C
C        GET FIRST SIX MOMENTS FOR LATEST G AND D VALUES
C
      CALL MOM(G, D, HMU, FAULT)
      IF (FAULT) RETURN
      S = HMU(1) * HMU(1)
      H2 = HMU(2) - S
      FAULT = H2 .LE. ZERO
      IF (FAULT) RETURN
      T = SQRT(H2)
      H2A = T * H2
      H2B = H2 * H2
      H3 = HMU(3) - HMU(1) * (THREE * HMU(2) - TWO * S)
      RBET = H3 / H2A
      H4 = HMU(4) - HMU(1) * (FOUR * HMU(3) - HMU(1) *
     *  (SIX * HMU(2) - THREE * S))
      BET2 = H4 / H2B
      W = G * D
      U = D * D
C
C        GET DERIVATIVES
C
      DO 120 J = 1, 2
      DO 110 K = 1, 4
      T = K
      IF (J .EQ. 1) GOTO 90
      S = ((W - T) * (HMU(K) - HMU(K + 1)) + (T + ONE) *
     *  (HMU(K + 1) - HMU(K + 2))) / U
      GOTO 100
   90 S = HMU(K + 1) - HMU(K)
  100 DD(K) = T * S / D
  110 CONTINUE
      T = TWO * HMU(1) * DD(1)
      S = HMU(1) * DD(2)
      Y = DD(2) - T
      DERIV(J) = (DD(3) - THREE * (S + HMU(2) * DD(1) - T * HMU(1))
     *  - ONE5 * H3 * Y / H2) / H2A
      DERIV(J + 2) = (DD(4) - FOUR * (DD(3) * HMU(1) + DD(1) * HMU(3))
     *  + SIX * (HMU(2) * T + HMU(1) * (S - T * HMU(1)))
     *  - TWO * H4 * Y / H2) / H2B
```

```
  120 CONTINUE
      T = ONE / (DERIV(1) * DERIV(4) - DERIV(2) * DERIV(3))
      U = (DERIV(4) * (RBET - RB1) - DERIV(2) * (BET2 - B2)) * T
      Y = (DERIV(1) * (BET2 - B2) - DERIV(3) * (RBET - RB1)) * T
C
C         FORM NEW ESTIMATES OF G AND D
C
      G = G - U
      IF (B1 .EQ. ZERO .OR. G .LT. ZERO) G = ZERO
      D = D - Y
      IF (ABS(U) .GT. TT .OR. ABS(Y) .GT. TT) GOTO 80
C
C         END OF ITERATION
C
      DELTA = D
      XLAM = SIGMA / SQRT(H2)
      IF (NEG) GOTO 130
      GAMMA = G
      GOTO 140
  130 GAMMA = -G
      HMU(1) = ONE - HMU(1)
  140 XI = XBAR - XLAM * HMU(1)
      RETURN
      END
C
      SUBROUTINE MOM(G, D, A, FAULT)
C
C         ALGORITHM AS 99.3  APPL. STATIST. (1976) VOL.25, NO.2
C
C         EVALUATES FIRST SIX MOMENTS OF A JOHNSON
C         SB DISTRIBUTION, USING GOODWIN METHOD
C
      REAL A(6), B(6), C(6), G, D, ZZ, VV, RTTWO, RRTPI, W, E, R, H, T,
     *  U, Y, X, V, F, Z, S, P, Q, AA, AB, EXPA, EXPB, ZERO, ONE, TWO,
     *  THREE
      LOGICAL L, FAULT
      DATA ZZ, VV, LIMIT /1.0E-5, 1.0E-8, 500/
C
C         RTTWO IS SQRT(2.0)
C         RRTPI IS RECIPROCAL OF SQRT(PI)
C         EXPA IS A VALUE SUCH THAT EXP(EXPA) DOES NOT QUITE
C           CAUSE OVERFLOW
C         EXPB IS A VALUE SUCH THAT 1.0 + EXP(-EXPB) MAY BE
C           TAKEN TO BE 1.0
C
      DATA RTTWO, RRTPI, EXPA, EXPB
     *  /1.414213562, 0.5641895835, 80.0, 23.7/
      DATA ZERO, ONE, TWO, THREE /0.0, 1.0, 2.0, 3.0/
      FAULT = .FALSE.
      DO 10 I = 1, 6
   10 C(I) = ZERO
      W = G / D
C
C         TRIAL VALUE OF H
C
      IF (W .GT. EXPA) GOTO 140
      E = EXP(W) + ONE
      R = RTTWO / D
      H = 0.75
      IF (D .LT. THREE) H = 0.25 * D
      K = 1
      GOTO 40
C
C         START OF OUTER LOOP
C
   20 K = K + 1
      IF (K .GT. LIMIT) GOTO 140
      DO 30 I = 1, 6
   30 C(I) = A(I)
C
C         NO CONVERGENCE YET - TRY SMALLER H
```

```
      H = 0.5 * H
   40 T = W
      U = T
      Y = H * H
      X = TWO * Y
      A(1) = ONE / E
      DO 50 I = 2, 6
   50 A(I) = A(I - 1) / E
      V = Y
      F = R * H
      M = 0
C
C         START OF INNER LOOP
C         TO EVALUATE INFINITE SERIES
C
   60 M = M + 1
      IF (M .GT. LIMIT) GOTO 140
      DO 70 I = 1, 6
   70 B(I) = A(I)
      U = U - F
      Z = ONE
      IF (U .GT. -EXPB) Z = EXP(U) + Z
      T = T + F
      L = T .GT. EXPB
      IF (.NOT.L) S = EXP(T) + ONE
      P = EXP(-V)
      Q = P
      DO 90 I = 1, 6
      AA = A(I)
      P = P / Z
      AB = AA
      AA = AA + P
      IF (AA .EQ. AB) GOTO 100
      IF (L) GOTO 80
      Q = Q / S
      AB = AA
      AA = AA + Q
      L = AA .EQ. AB
   80 A(I) = AA
   90 CONTINUE
  100 Y = Y + X
      V = V + Y
      DO 110 I = 1, 6
      IF (A(I) .EQ. ZERO) GOTO 140
      IF (ABS((A(I) - B(I)) / A(I)) .GT. VV) GOTO 60
  110 CONTINUE
C
C         END OF INNER LOOP
C
      V = RRTPI * H
      DO 120 I = 1, 6
  120 A(I) = V * A(I)
      DO 130 I = 1, 6
      IF (A(I) .EQ. ZERO) GOTO 140
      IF (ABS((A(I) - C(I)) / A(I)) .GT. ZZ) GOTO 20
  130 CONTINUE
C
C         END OF OUTER LOOP
C
      RETURN
  140 FAULT = .TRUE.
      RETURN
      END
```

Appendix 2

Script of the Computer Codes
In Microsoft Excel VBA

Subroutine “FunctionMOM”

```
Sub FunctionMOM()

'This macro gets the first four moments of up to ten variables and the functional
relationship between these variables and the variate Y as input and calculates the first
four moments of Y based on formulas of Tukey [14]up to their first terms

    Sheets("Moments of Y").Select
    'Calculate skewness and Curtosis for variables
    Range("B6").Select
    ActiveCell.FormulaR1C1 = "=R[-2]C/R[-3]C^3"
    Range("B7").Select
    ActiveCell.FormulaR1C1 = "=R[-2]C/R[-4]C^4"
    Range("B6:B7").Select
    Selection.AutoFill Destination:=Range("B6:K7"), Type:=xlFillDefault
    Range("B6:K7").Select

    'Copy means of variables into 10 subtables for paritial derivatives
    Range("B2:K2").Select
    Selection.Copy
    Range("B9:K11").Select
    ActiveSheet.Paste
    Range("B17:K19").Select
    ActiveSheet.Paste
    Range("B25:K27").Select
    ActiveSheet.Paste
    Range("B33:K35").Select
    ActiveSheet.Paste
    Range("B41:K43").Select
    ActiveSheet.Paste
```

```
Range("B49:K51").Select
ActiveSheet.Paste
Range("B57:K59").Select
ActiveSheet.Paste
Range("B65:K67").Select
ActiveSheet.Paste
Range("B73:K75").Select
ActiveSheet.Paste
Range("B81:K83").Select
ActiveSheet.Paste

'Copy formula for relationship of Y and design vriables into "Mean of Y" cells
Range("N5").Select
Selection.Copy
Range("B12").Select
Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, _
    SkipBlanks:=False, Transpose:=False
Range("B20").Select
Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, _
    SkipBlanks:=False, Transpose:=False
Range("B28").Select
Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, _
    SkipBlanks:=False, Transpose:=False
Range("B36").Select
Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, _
    SkipBlanks:=False, Transpose:=False
Range("B44").Select
Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, _
    SkipBlanks:=False, Transpose:=False
Range("B52").Select
Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, _
```

```
    SkipBlanks:=False, Transpose:=False
Range("B60").Select
Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, _
    SkipBlanks:=False, Transpose:=False
Range("B68").Select
Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, _
    SkipBlanks:=False, Transpose:=False
Range("B76").Select
Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, _
    SkipBlanks:=False, Transpose:=False
Range("B84").Select
Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, _
    SkipBlanks:=False, Transpose:=False

'Make Mean X+/-Sigma for all the variables
Range("B10").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+R[-7]C"
Range("B11").Select
ActiveCell.FormulaR1C1 = "=R[-2]C-R[-8]C"
Range("C18").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+R[-15]C"
Range("C19").Select
ActiveCell.FormulaR1C1 = "=R[-2]C-R[-16]C"
Range("D26").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+R[-23]C"
Range("D27").Select
ActiveCell.FormulaR1C1 = "=R[-2]C-R[-24]C"
Range("E34").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+R[-31]C"
Range("E35").Select
ActiveCell.FormulaR1C1 = "=R[-2]C-R[-32]C"
```

```
Range("F42").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+R[-39]C"
Range("F43").Select
ActiveCell.FormulaR1C1 = "=R[-2]C-R[-40]C"
Range("G50").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+R[-47]C"
Range("G51").Select
ActiveCell.FormulaR1C1 = "=R[-2]C-R[-48]C"
Range("H58").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+R[-55]C"
Range("H59").Select
ActiveCell.FormulaR1C1 = "=R[-2]C-R[-56]C"
Range("I66").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+R[-63]C"
Range("I67").Select
ActiveCell.FormulaR1C1 = "=R[-2]C-R[-64]C"
Range("J74").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+R[-71]C"
Range("J75").Select
ActiveCell.FormulaR1C1 = "=R[-2]C-R[-72]C"
Range("K82").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+R[-79]C"
Range("K83").Select
ActiveCell.FormulaR1C1 = "=R[-2]C-R[-81]C"

'Make F(X+/-Sigma)for all the variables
Range("B12").Select
Selection.AutoFill Destination:=Range("B12:B14"), Type:=xlFillDefault
Range("B12:B14").Select
Range("B20").Select
Selection.AutoFill Destination:=Range("B20:B22"), Type:=xlFillDefault
```

```
Range("B20:B22").Select
Range("B28").Select
Selection.AutoFill Destination:=Range("B28:B30"), Type:=xlFillDefault
Range("B28:B30").Select
Range("B36").Select
Selection.AutoFill Destination:=Range("B36:B38"), Type:=xlFillDefault
Range("B36:B38").Select
Range("B44").Select
Selection.AutoFill Destination:=Range("B44:B46"), Type:=xlFillDefault
Range("B44:B46").Select
Range("B52").Select
Selection.AutoFill Destination:=Range("B52:B54"), Type:=xlFillDefault
Range("B52:B54").Select
Range("B60").Select
Selection.AutoFill Destination:=Range("B60:B62"), Type:=xlFillDefault
Range("B60:B62").Select
Range("B68").Select
Selection.AutoFill Destination:=Range("B68:B70"), Type:=xlFillDefault
Range("B68:B70").Select
Range("B76").Select
Selection.AutoFill Destination:=Range("B76:B78"), Type:=xlFillDefault
Range("B76:B78").Select
Range("B84").Select
Selection.AutoFill Destination:=Range("B84:B86"), Type:=xlFillDefault
Range("B84:B86").Select

'Calculate partial derivative with respect to each variable
Range("B15").Select
ActiveCell.FormulaR1C1 = "=(R[-2]C-R[-1]C)/(2*R[-12]C)"
Range("B23").Select
ActiveCell.FormulaR1C1 = "=(R[-2]C-R[-1]C)/(2*R[-20]C[1])"
```

```
Range("B31").Select
ActiveCell.FormulaR1C1 = "=(R[-2]C-R[-1]C)/(2*R[-28]C[2])"
Range("B39").Select
ActiveCell.FormulaR1C1 = "=(R[-2]C-R[-1]C)/(2*R[-36]C[3])"
Range("B47").Select
ActiveCell.FormulaR1C1 = "=(R[-2]C-R[-1]C)/(2*R[-44]C[4])"
Range("B55").Select
ActiveCell.FormulaR1C1 = "=(R[-2]C-R[-1]C)/(2*R[-52]C[5])"
Range("B63").Select
ActiveCell.FormulaR1C1 = "=(R[-2]C-R[-1]C)/(2*R[-60]C[6])"
Range("B71").Select
ActiveCell.FormulaR1C1 = "=(R[-2]C-R[-1]C)/(2*R[-68]C[7])"
Range("B79").Select
ActiveCell.FormulaR1C1 = "=(R[-2]C-R[-1]C)/(2*R[-76]C[8])"
Range("B87").Select
ActiveCell.FormulaR1C1 = "=(R[-2]C-R[-1]C)/(2*R[-84]C[9])"

'Copy the results into "Johnson worksheet"
Range("N9:N10").Select
Selection.Copy
Sheets("Johnson").Select
Range("B2:B3").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
    :=False, Transpose:=False
'Sheets("Moments of Y").Select
'Range("N13:N14").Select
'Application.CutCopyMode = False
'Selection.Copy
'Sheets("Johnson").Select
'Range("B4:B5").Select
'Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
```

```
        :=False, Transpose:=False


    'Run macros for determinimg the family and parameters of Johnson and graph the pdf
    JOHNSONTYPES
    GRAPHS

End Sub
```

Subroutine "JOHNSONTYPES"

'MAIN PART OF THIS MACRO IS ORIGINALLY PROPOSED BY HILL, HILL, AND HOLDER "FITTING JOHNSON CURVES BY MOMENTS" APPLIED STATISTICS, 25 (1976) 180-189

```
Sub JOHNSONTYPES()
'This macro determines the family of Johnson distribution that fits a variate given its first four moments, it also determines the corresponding parameters of the distribution

 Const TOL = 0.01
'DEFINE VARIABLES

    Dim X As Double
    Dim Y As Double
    Dim U As Double
    Dim W As Double
    Dim XBAR As Double
    Dim SD As Double
    Dim RB1 As Double
    Dim B1 As Double
    Dim B2 As Double
    Dim BB2 As Double
    Dim LAMBDA As Double
    Dim DELTA As Double
    Dim GAMMA As Double
    Dim XI As Double
    Dim XLAM As Double
    Dim B3 As Double

    XBAR = Range("B2")
```

```
    SD = Range("B3")
    RB1 = Range("B4")
    B2 = Range("B5")
    B1 = RB1 ^ 2

    Worksheets("Johnson").Range("D2:F2").ClearContents
    Worksheets("Johnson").Range("I1:I4").ClearContents
    Worksheets("Johnson").Range("A11:A19").ClearContents

'CHECK FAILURES
    Range("D2").Select
    If B2 < B1 + 1 Then
        ActiveCell.FormulaR1C1 = "IMPOSSIBLE AREA"
    ElseIf Range("B3") < 0 Then
        ActiveCell.FormulaR1C1 = "NEGATIVE STANDARD DEVIATION"
    Else: ActiveCell.FormulaR1C1 = "SUCCESSFUL COMPLETION"
    End If

    Columns("D:D").EntireColumn.AutoFit

  If Range("D2") = "SUCCESSFUL COMPLETION" Then

      'DETERMINE WHICH TYPE

      'TEST WHETHER LOGNORMAL OR NORMAL REQUESTED
      If B2 >= 0 Then
        GoTo 30
      End If
20      If Abs(RB1) <= TOL Then
          GoTo 70
      Else: GoTo 80
```

```
End If

'TESST FOR POSITION RELATIVE TO BOUNDARY LINE
30    If B2 > B1 + TOL + 1 Then
        GoTo 60
      End If

'CALCULATE PARAMETERS OF ST DISTRIBUTION
40    Range("F2").Select
      ActiveCell.FormulaR1C1 = "ST"
      Y = 0.5 + 0.5 * Sqr(1 - 4 / (B1 + 4))
      If RB1 > 0 Then
        Y = 1 - Y
      End If
      X = SD / Sqr(Y * (1 - Y))
      XI = XBAR - Y * X
      XLAM = XI + X
      DELTA = Y

'INSERT PARAMETERS INTO CELLS I1:I4
 Range("I1").Select
 ActiveCell.FormulaR1C1 = 0
 Range("I2").Select
 ActiveCell.FormulaR1C1 = DELTA
 Range("I3").Select
 ActiveCell.FormulaR1C1 = XLAM
 Range("I4").Select
 ActiveCell.FormulaR1C1 = XI

 Exit Sub
```

```
60      If Abs(RB1) > TOL Or Abs(B2 - 3) > TOL Then
          GoTo 80
        End If

        'CALCULATE PARAMETERS OF NORMAL DISTRIBUTION AND INSERT
INTO CELLS
70     Range("F2").Select
       ActiveCell.FormulaR1C1 = "NORMAL"
       DELTA = 1 / SD
       GAMMA = -XBAR / SD
       Range("I1").Select
       ActiveCell.FormulaR1C1 = GAMMA
       Range("I2").Select
       ActiveCell.FormulaR1C1 = DELTA

       Exit Sub

       'TEST FOR POSITION RELATEVE TO LOGNORMAL LINE

80      U = 1 / 3
        X = 0.5 * B1 + 1
        Y = RB1 * Sqr(0.25 * B1 + 1)
        W = (X + Y) ^ U + (X - Y) ^ U - 1
        U = W ^ 2 * (3 + W * (2 + W)) - 3
        If B2 < 0 Then
           B2 = U
        End If
        X = U - B2

        If Abs(X) > TOL Then
          GoTo 90
```

```
        End If

    'COMPUTE THE PARAMETERS OF THE DISTRIBUTIONS OTHER THAN ST
AND NORMAL

    'SL
    Range("F2").Select
    ActiveCell.FormulaR1C1 = "SL"

        XLAM = Sgn(RB1)
        LAMBDA = XLAM
        U = XLAM * XBAR
        X = 1 / Sqr(Log(W))
        DELTA = X
        Y = 0.5 * X * Log(W * (W - 1) / (SD ^ 2))
        GAMMA = Y
        XI = U - Exp((0.5 / X - Y) / X)
        Range("I1").Select
        ActiveCell.FormulaR1C1 = GAMMA
        Range("I2").Select
        ActiveCell.FormulaR1C1 = DELTA
        Range("I3").Select
        ActiveCell.FormulaR1C1 = LAMBDA
        Range("I4").Select
        ActiveCell.FormulaR1C1 = XI

        Exit Sub

    'SB OR SU DISTRIBUTION

90      If X > 0 Then
```

```
      GoTo 100
    Else
      Range("F2").Select
      ActiveCell.FormulaR1C1 = "SU"
      SUFIT
      Exit Sub
    End If

100   Range("F2").Select
    ActiveCell.FormulaR1C1 = "SB"
    SBFIT

  End If

  ' GRAPHS

  End Sub
```

Subroutine “SUFIT”

```
'MAIN PART OF THIS MACRO IS ORIGINALLY PROPOSED BY HILL, HILL,
AND HOLDER "FITTING JOHNSON CURVES BY MOMENTS" APPLIED
STATISTICS, 25 (1976) 180-189
Sub SUFIT()

'EXTRA
Const TOL = 0.01
'DEFINE VARIABLES

   XBAR = Range("B2")
   SD = Range("B3")
   RB1 = Range("B4")
   B2 = Range("B5")
   B1 = RB1 ^ 2

If Range("F2") = "SU" Then

         Dim W1, WM1 As Double
         Dim z As Double
         Dim V As Double
         Dim A As Double
         Dim B As Double
         Dim XV As Double


      B3 = B2 - 3
      'W IS FIRSST ESTIMATE OF EXP(DELTA^(-2))
      W = Sqr(Sqr(2 * B2 - 2.8 * B1 - 2) - 1)
```

```
If Abs(RB1) > TOL Then
   GoTo 10
Else
   Y = 0
   GoTo 20
End If

Do
10       W1 = W + 1
   WM1 = W - 1
   z = W1 * B3
   V = W * (6 + W * (3 + W))
   A = 8 * (WM1 * (3 + W * (7 + V)) - z)
   B = 16 * (WM1 * (6 + V) - B3)
   'ADD EXTRA VARIABLES TO REDUCE COMPLEXITY
   XV = (3 + W * (9 + W * (10 + V)))
   Y = (Sqr(A * A - 2 * B * (WM1 * XV - 2 * W1 * z)) - A) / B
   XV = (4 * (W + 2) * Y + 3 * W1 ^ 2) ^ 2
   z = Y * WM1 * XV / (2 * (2 * Y + W1) ^ 3)
   V = W ^ 2
   XV = (B1 * (B2 - 1.5 - V * (1 + 0.5 * V)))
   W = Sqr(Sqr(1 - 2 * (1.5 - B2 + XV / z)) - 1)

Loop Until Abs(B1 - z) <= TOL

'END OF ITERATION

 Y = Y / W
 Y = Log(Sqr(Y) + Sqr(Y + 1))
   If RB1 > 0 Then
      Y = -Y
```

```
        End If

20    X = Sqr(1 / Log(W))
    DELTA = X
    GAMMA = Y * X
    Y = Exp(Y)
    z = Y ^ 2
    XV = (0.5 * W * (z + 1 / z) + 1)
    X = SD / Sqr(0.5 * (W - 1) * XV)
    XLAM = X
    LAMBDA = XLAM
    XI = (0.5 * Sqr(W) * (Y - 1 / Y)) * X + XBAR

     Range("I1").Select
     ActiveCell.FormulaR1C1 = GAMMA
     Range("I2").Select
     ActiveCell.FormulaR1C1 = DELTA
     Range("I3").Select
     ActiveCell.FormulaR1C1 = LAMBDA
     Range("I4").Select
     ActiveCell.FormulaR1C1 = XI

End If
End Sub
```

Subroutine "SBFIT"

```
'MAIN PART OF THIS MACRO IS ORIGINALLY PROPOSED BY HILL, HILL,
AND HOLDER "FITTING JOHNSON CURVES BY MOMENTS" APPLIED
STATISTICS, 25 (1976) 180-189
Sub SBFIT()

If Range("F2") = "SB" Then
  'EXTRA
  Const TOL = 0.01
  'DEFINE VARIABLES

  XBAR = Range("B2")
  SD = Range("B3")
  RB1 = Range("B4")
  B2 = Range("B5")
  B1 = RB1 ^ 2

    Const TT = 1# * 10 ^ (-4)
    Const LIMIT = 50

    Dim RTB1 As Double
    Dim E, F As Double
    Dim D As Variant
    Dim G As Variant
    Dim M As Double
    Dim HMU(6), DERIV(4), DD(4), GAMMA, DELTA, XLAM, XI, U, X, Y, W, S,
H2, T, H2A, H2B, H3, H4, RBET, BET2 As Double
    Dim J, K As Long
    Dim NEG As Long
    Dim FAULT As Integer
```

```
RTB1 = RB1
RB1 = Abs(RTB1)

If RTB1 < 0 Then
  NEG = 1
 Else
  NEG = 0
End If

'GET D AS FIRST ESTIMATE OF DELTA

E = B1 + 1
U = 1 / 3
X = 0.5 * B1 + 1
Y = RB1 * Sqr(0.25 * B1 + 1)
W = (X + Y) ^ U + (X - Y) ^ U - 1
F = W * W * (3 + W * (2 + W)) - 3
E = (B2 - E) / (F - E)

  If Abs(RB1) > TOL Then
    GoTo 5
  End If
   F = 2
  GoTo 20
5      D = 1 / Sqr(Log(W))
  If D < 0.64 Then
    GoTo 10
  End If
  F = 2 - 8.5254 / (D * (D * (D - 2.163) + 11.346))
```

```
        GoTo 20
10       F = 1.25 * D
20       F = E * F + 1

        If F < 1.8 Then
            GoTo 25
        End If
        D = (0.625 * F - 0.408) * (3 - F) ^ (-0.479)
        GoTo 30
25       D = 0.8 * (F - 1)

      'GET G AS FIRST ESTIMATE OF GAMMA

30      G = 0
        If B1 < TT Then
            GoTo 70
        End If

        If D > 1 Then
           GoTo 40
        End If
        G = (0.7466 * D ^ 1.7973 + 0.5955) * B1 ^ 0.485
        GoTo 70
40       If D <= 2.5 Then
           GoTo 50
        End If
        U = 0.0124
        Y = 0.5291
        GoTo 60
50       U = 0.0623
        Y = 0.4043
```

```
60      G = B1 ^ (U * D + Y) * (0.9281 + D * (1.0614 * D - 0.7077))
70      M = 0

        'MAIN ITERATION STARTS HERE

80       M = M + 1
        If M > LIMIT Then
           FAULT = 1
         Else: FAULT = 0
        End If

        'GET FIRST SIX MOMENTS FOR LATEST G AND D VALUES
        Range("A11").Select
        ActiveCell.FormulaR1C1 = G
        Range("A12").Select
        ActiveCell.FormulaR1C1 = D

        MOMSUB

        HMU(1) = Range("A13")
        HMU(2) = Range("A14")
        HMU(3) = Range("A15")
        HMU(4) = Range("A16")
        HMU(5) = Range("A17")
        HMU(6) = Range("A18")
        FAULT = Range("A19")

        If FAULT = 1 Then Exit Sub

        S = HMU(1) ^ 2
        H2 = HMU(2) - S
```

```
    If H2 <= 0 Then
      FAULT = 1
    Else: FAULT = 0
    End If

    If FAULT = 1 Then Exit Sub

    T = Sqr(H2)
    H2A = T * H2
    H2B = H2 ^ 2
    H3 = HMU(3) - HMU(1) * (3 * HMU(2) - 2 * S)
    RBET = H3 / H2A
    H4 = HMU(4) - HMU(1) * (4 * HMU(3) - HMU(1) * (6 * HMU(2) - 3 * S))
    BET2 = H4 / H2B
    W = G * D
    U = D * D

    'GET DERIVATIVES
  For J = 1 To 2
    For K = 1 To 4
      T = K
      If J = 1 Then GoTo 90

      S = ((W - T) * (HMU(K) - HMU(K + 1)) + (T + 1) * (HMU(K + 1) - HMU(K +
2))) / U
      GoTo 100
90        S = HMU(K + 1) - HMU(K)
100       DD(K) = T * S / D
110     Next K

    T = 2 * HMU(1) * DD(1)
```

```
        S = HMU(1) * DD(2)
        Y = DD(2) - T
        DERIV(J) = (DD(3) - 3 * (S + HMU(2) * DD(1) - T * HMU(1)) - 1.5 * H3 * Y /
H2) / H2A
        DERIV(J + 2) = (DD(4) - 4 * (DD(3) * HMU(1) + DD(1) * HMU(3)) + 6 *
(HMU(2) * T + HMU(1) * (S - T * HMU(1))) - 2 * H4 * Y / H2) / H2B
120  Next J

        T = ONE / (DERIV(1) * DERIV(4) - DERIV(2) * DERIV(3))
        U = (DERIV(4) * (RBET - RB1) - DERIV(2) * (BET2 - B2)) * T
        Y = (DERIV(1) * (BET2 - B2) - DERIV(3) * (RBET - RB1)) * T

        'FORM NEW ESTIMATES OF G AND D

        G = G - U
        If B1 = 0 Or G < 0 Then
           G = 0
        End If
        D = D - Y
        If Abs(U) > TT Or Abs(Y) > TT Then
          GoTo 80
        End If

        'END OF ITERATION
        DELTA = D
        XLAM = SD / Sqr(H2)
        LAMBDA = XLAM
        If NEG = 1 Then
          GoTo 130
        End If
        GAMMA = G
```

```
        GoTo 140
130     GAMMA = -G
        HMU(1) = 1 - HMU(1)
140     XI = XBAR - XLAM * HMU(1)

         Range("I1").Select
        ActiveCell.FormulaR1C1 = GAMMA
        Range("I2").Select
        ActiveCell.FormulaR1C1 = DELTA
        Range("I3").Select
        ActiveCell.FormulaR1C1 = LAMBDA
        Range("I4").Select
        ActiveCell.FormulaR1C1 = XI


End If

End Sub
```

Subroutine "MOMSUB"

```
'THIS MACRO IS ORIGINALLY PROPOSED BY HILL, HILL, AND HOLDER
"FITTING JOHNSON CURVES BY MOMENTS" APPLIED STATISTICS, 25 (1976)
180-189
Sub MOMSUB()

    Const zz = 1# * 10 ^ -5
    Const VV = 1# * 10 ^ -8
    Const LIMIT = 500
    Const RTTWO = 1.414213562    'SQRT(2)
    Const RRTPI = 0.5641895835   'RECIPROCAL OF SQRT(PI)
    Const EXPA = 80#             'A VALUE SUCH THAT EXP(EXPA) DOES NOT QUIT
CAUSE OVERFLOW
    Const EXPB = 23.7            'A VALUE SUCH THAT 1.0+EXP(-EXPB)MAYBE
TAKEN TO BE 1.0

    Dim G, D, W, E, R, H, T, UY, X, V, F, z, S, P, Q, AA, AB, HMU(6) As Variant


    Dim A(6) As Double
    Dim B(6) As Double
    Dim C(6) As Double
    Dim i, L As Long
    Dim FAULT As Integer


    G = Range("A11")
    D = Range("A12")
```

```
   For i = 1 To 6
      HMU(i) = 0
   Next i

   For i = 1 To 6
10    C(i) = 0
   Next i

     W = G / D

     'TRIAL VALUE OF H

     If W > EXPA Then
       GoTo 140
     End If


       E = Exp(W) + 1
       R = RTTWO / D
       H = 0.75
          If D < 3 Then H = 0.25 * D

          K = 1
          GoTo 40

              'START OF OUTER LOOP

20               K = K + 1
              If K > LIMIT Then GoTo 140
```

```
            For i = 1 To 6
30              C(i) = A(i)
            Next i

            'NO CONVERGENCE YET- TRY SMALLER H

        H = 0.5 * H

40        T = W
        U = T
        Y = H ^ 2
        X = 2 * Y
        A(1) = 1 / E
        For i = 2 To 6
50          A(i) = A(i - 1) / E
        Next i
        V = Y
        F = R * H
        M = 0

        'START OF INNER LOOP TO EVALUATE INFINITE SERIES

60        M = M + 1

          If M > LIMIT Then GoTo 140

        For i = 1 To 6
70          B(i) = A(i)
        Next i
        U = U - F
        z = 1
```

```
  If U > -EXPB Then
      z = Exp(U) + z
  End If

T = T + F

  If T > EXPB Then
      L = 1
    Else: L = 0
  End If

  If L = 0 Then S = Exp(T) + 1

P = Exp(-V)
Q = P
For i = 1 To 6
  AA = A(i)
  P = P / z
  AB = AA
  AA = AA + P
      If AA = AB Then GoTo 100

      If L = 1 Then GoTo 80

  Q = Q / S
  AB = AA
  AA = AA + Q
  If AA = AB Then L = 1

80      A(i) = AA
```

```
90     Next i

100     Y = Y + X
      V = V + Y

      For i = 1 To 6
         If A(i) = 0 Then GoTo 140

         If Abs(A(i) - B(i)) / A(i) > VV Then GoTo 60

110      Next i

         'END OF INNER LOOP
         V = RRTPI * H

         For i = 1 To 6
120            A(i) = V * A(i)
         Next i

      For i = 1 To 6
         If A(i) = 0 Then GoTo 140

         If Abs((A(i) - C(i)) / A(i)) > zz Then GoTo 20
      Next i

     'END OF OUTER LOOP

     For i = 1 To 6
      HMU(i) = A(i)
     Next i
```

```
    Range("A13").Select
    ActiveCell.FormulaR1C1 = HMU(1)
    Range("A14").Select
    ActiveCell.FormulaR1C1 = HMU(2)
    Range("A15").Select
    ActiveCell.FormulaR1C1 = HMU(3)
    Range("A16").Select
    ActiveCell.FormulaR1C1 = HMU(4)
    Range("A17").Select
    ActiveCell.FormulaR1C1 = HMU(5)
    Range("A18").Select
    ActiveCell.FormulaR1C1 = HMU(6)
    Range("A19").Select
    ActiveCell.FormulaR1C1 = FAULT

   Exit Sub

140 FAULT = 1
'WRTIE THE OUTPUT TO EXCEL
  End Sub
```

Subroutine "GRAPHS"

```
Sub GRAPHS()
'This macro draws the graph of the density funciton of the fitted Johnson distribution

Dim GAMMA, DELTA, LAMBDA, XI, X, LOLIM, UPLIM, XBAR, SD, STP,
        LOLIMSTP, USL, LSL, MAXY As Variant
Dim TEMPRANGEX, TEMPRANGEY, USLRANGE, LSLRANGE, SLYRANGE As
        Range
Dim i As Double

  'Enter the parameters of Johnson distribution, mean, variance and spec limits from
        worksheet "Johnson" as input
   GAMMA = Range("I1")
   DELTA = Range("I2")
   LAMBDA = Range("I3")
   XI = Range("I4")
   XBAR = Range("B2")
   SD = Range("B3")
   USL = Range("E6")
   LSL = Range("E7")

Application.ScreenUpdating = False
'If the variate is in the impossible area terminate
If Range("D2") = "IMPOSSIBLE AREA" Then
   GoTo 10
End If
'Generate data for graph of SL function
  If Range("F2") = "SL" Then
  'find the startind point of the graph by comparing XI and zero
    Range("G12").Select
```

```
  ActiveCell.FormulaR1C1 = "=MAX(R[-8]C[2],0)"

  'Find the upper and lowr limits for x axis
  LOLIM = XBAR - 5 * SD
  UPLIM = XBAR + 10 * SD
  'Insert the density function in cell B10 and copy to cells b10 through B110
  Range("B10").Select
  ActiveCell.FormulaR1C1 = "=(R2C9 / (Sqrt(2 * Pi()) * (RC[-1] - R4C9))) * Exp(-0.5
       * R2C9 ^ 2 * (R1C9 / R2C9 + LN(RC[-1] - R4C9)) ^ 2)"
  Range("B10").Select
  Selection.AutoFill Destination:=Range("B10:B110") ' & Trim(Str(10 + (UPLIM -
       LOLIM) * 10))), Type:=xlFillDefault
  'Insert values of the x axis in column A
  STP = (UPLIM - LOLIM) / 100
  Range("A10").Select
  LOLIMSTP = LOLIM
  For i = 0 To 100
    ActiveCell.Offset(i, 0).Value = LOLIMSTP + STP
    LOLIMSTP = LOLIMSTP + STP
  Next i

End If

'Generate data for graph of normal function
If Range("F2") = "NORMAL" Then

'Find the upper and lowr limits for x axis
  LOLIM = XBAR - 4 * SD
  UPLIM = XBAR + 4 * SD
  'Insert formula for the density function of normal in cells B10 through B110
  Range("B10").Select
```

```
  ActiveCell.FormulaR1C1 = "=(1/SQRT(2*PI()))*EXP(-((RC[-1]-
       R2C)^2/(2*R3C^2)))"
  Range("B10").Select
  Selection.AutoFill Destination:=Range("B10:B110") ' & Trim(Str((UPLIM - LOLIM)
       * 100 + 10))), Type:=xlFillDefault
  'Insert values for the x axis in column A
  STP = 8 * SD / 100
  Range("A10").Select
  LOLIMSTP = LOLIM
  For i = 0 To 100
   ActiveCell.Offset(i, 0).Value = LOLIMSTP + STP
   LOLIMSTP = LOLIMSTP + STP
  Next i
End If

'Generate data for graph of SB function
If Range("F2") = "SB" Or Range("F2") = "ST" Then

'Find the upper and the lower bound for x
  LOLIM = XI
  UPLIM = XI + LAMBDA

  'Insert density function in cells B10 through B110
  Range("B10").Select
  ActiveCell.FormulaR1C1 = "=(R2C9/SQRT(2*PI()))*(R3C9/((RC[-1]-
       R4C9)*(R3C9-RC[-1]+R4C9)))*EXP(-.5*(R1C9+R2C9*LN((RC[-1]-
       R4C9)/(R3C9-RC[-1]+R4C9)))^2)"

  Range("B10").Select
  Selection.AutoFill Destination:=Range("B10:B110") ' & Trim(Str(10 + (UPLIM -
       LOLIM) * 10))), Type:=xlFillDefault
```

```
  'Insert values of x in column A
  STP = LAMBDA / 100
  Range("A10").Select
  LOLIMSTP = LOLIM

  For i = 1 To 100
    Range("A10") = LOLIM
    ActiveCell.Offset(i, 0).Value = LOLIMSTP + STP
    LOLIMSTP = LOLIMSTP + STP
  Next i

End If

'Generate data for graph of SU function
If Range("F2") = "SU" Then

'Find upper and lower bounds for x
  LOLIM = XI - 4 * LAMBDA
  UPLIM = XI + 4 * LAMBDA

  'Insert density function in cells B10 through B110
  Range("B10").Select
  ActiveCell.FormulaR1C1 = "=(R2C9/SQRT(2*PI()))*(1/SQRT((RC[-1]-
       R4C9)^2+R3C9^2))*EXP(-.5*(R1C9+R2C9*LN((RC[-1]-R4C9)/R3C9+(((RC[-
       1]-R4C9)/R3C9)^2+1)^.5))^2)"

  Range("B10").Select
  Selection.AutoFill Destination:=Range("B10:B110") ' & Trim(Str(10 + (UPLIM -
       LOLIM) * 10))), Type:=xlFillDefault
```

```
  'Insert values of x in column A
  STP = 8 * LAMBDA / 100
  Range("A10").Select
  LOLIMSTP = LOLIM

  For i = 1 To 100
    Range("A10") = LOLIM
    ActiveCell.Offset(i, 0).Value = LOLIMSTP + STP
    LOLIMSTP = LOLIMSTP + STP
  Next i

End If

 'Generate data for marking the specification limits
  Range("G11").Select
  ActiveCell.FormulaR1C1 = "=MAX(RC[-5]:R[98]C[-5])"
  MAXY = Range("G11")

  Range("E10").Select
  Range("E10").FormulaR1C1 = 0
  STP = MAXY / 100
  For i = 1 To 100
   ActiveCell.Offset(i, 0).Value = STP * i
  Next i

  For i = 0 To 100
   ActiveCell.Offset(i, -2).Value = USL
   ActiveCell.Offset(i, -1).Value = LSL
  Next i
Application.ScreenUpdating = True
```

```
'DRAW CHART BASED ON 100 DATA POINTS
'SET THE RANGES FOR THE DATA SOURCES OF X, F(X), USL AND LSL
Set TEMPRANGEX = Range("A10:A110") '& Trim(Str((UPLIM - LOLIM) * 10 +
    10)))
Set TEMPRANGEY = Range("B11:B110") '& Trim(Str((UPLIM - LOLIM) * 10 +
    10)))
Set USLRANGE = Range("C10:C110")
Set LSLRANGE = Range("D10:D110")
Set SLYRANGE = Range("E10:E110")

'ADD DATA TO CHART
Range("A10").Select
Charts.Add
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.SetSourceData Source:=Sheets("Johnson").Range("E2")
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection(1).XValues = TEMPRANGEX
ActiveChart.SeriesCollection(1).Values = TEMPRANGEY
'NAME THE DATA SERIES
ActiveChart.SeriesCollection(1).Name = "=""PDF"""
ActiveChart.SeriesCollection(2).XValues = USLRANGE
ActiveChart.SeriesCollection(2).Values = SLYRANGE
ActiveChart.SeriesCollection(2).Name = "=""LSL"""
ActiveChart.SeriesCollection(3).XValues = LSLRANGE
ActiveChart.SeriesCollection(3).Values = SLYRANGE
ActiveChart.SeriesCollection(3).Name = "=""USL"""

'DEFINE NAME AND LOCATION OF CHART AND NAME THE AXES
ActiveChart.Location Where:=xlLocationAsObject, Name:="Johnson"
```

```
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "PDF AND LIMITS"
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "X"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "F(X)"
    End With

'CHANGE COLOR AND STYLE OF CHART
ActiveChart.PlotArea.Select
    With Selection.Border
        .ColorIndex = 16
        .Weight = xlThin
        .LineStyle = xlContinuous
    End With
    With Selection.Interior
        .ColorIndex = 2
        .PatternColorIndex = 1
        .Pattern = xlSolid
    End With
    ActiveChart.ChartArea.Select
    ActiveChart.SeriesCollection(2).Select
    With Selection.Border
        .ColorIndex = 57
        .Weight = xlMedium
        .LineStyle = xlContinuous
    End With
    With Selection
        .MarkerBackgroundColorIndex = xlNone
        .MarkerForegroundColorIndex = xlNone
```

```
    .MarkerStyle = xlNone
    .Smooth = True
    .MarkerSize = 3
    .Shadow = False
End With
ActiveChart.PlotArea.Select
ActiveChart.SeriesCollection(3).Select
With Selection.Border
    .ColorIndex = 57
    .Weight = xlMedium
    .LineStyle = xlContinuous
End With
With Selection
    .MarkerBackgroundColorIndex = xlNone
    .MarkerForegroundColorIndex = xlNone
    .MarkerStyle = xlNone
    .Smooth = True
    .MarkerSize = 3
    .Shadow = False
End With
ActiveChart.SeriesCollection(1).Select
With Selection.Border
    .ColorIndex = 57
    .Weight = xlThick
    .LineStyle = xlContinuous
End With
With Selection
    .MarkerBackgroundColorIndex = xlNone
    .MarkerForegroundColorIndex = xlNone
    .MarkerStyle = xlNone
    .Smooth = True
```

```
    .MarkerSize = 3
    .Shadow = False
End With
ActiveChart.ChartArea.Select
ActiveChart.SeriesCollection(3).Select
With Selection.Border
    .ColorIndex = 3
    .Weight = xlMedium
    .LineStyle = xlContinuous
End With
With Selection
    .MarkerBackgroundColorIndex = xlNone
    .MarkerForegroundColorIndex = xlNone
    .MarkerStyle = xlNone
    .Smooth = True
    .MarkerSize = 3
    .Shadow = False
End With
ActiveChart.ChartArea.Select
ActiveChart.SeriesCollection(2).Select
With Selection.Border
    .ColorIndex = 50
    .Weight = xlMedium
    .LineStyle = xlContinuous
End With
With Selection
    .MarkerBackgroundColorIndex = xlNone
    .MarkerForegroundColorIndex = xlNone
    .MarkerStyle = xlNone
    .Smooth = True
    .MarkerSize = 3
```

```
        .Shadow = False
    End With
    ActiveChart.ChartArea.Select

10 End Sub
```