Fall 2007

Robust Mechanism synthesis with random and interval variables

Pavan Kumar Venigella

ROBUST MECHANISM SYNTHESIS

WITH RANDOM AND INTERVAL

VARIABLES

by

PAVAN KUMAR VENIGELLA

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

2007

Approved by

______________________________ ______________________________

Dr. Xiaoping Du, Advisor Dr. Ashok Midha

____

Dr. K. Chandrashekhara

ABSTRACT

Robust mechanism synthesis ensures that the performance of a mechanism is not sensitive to uncertainties in the mechanism and its environment. The uncertainties include the dimension variations, installation errors, random input motion, and various external forces. Robust mechanism synthesis is used to minimize the impact of these uncertainties on the mechanism performance. Robust mechanism synthesis has been performed by either a probabilistic approach or a worst case approach. The former approach describes uncertain parameters as random variables while the latter approach treats uncertain parameters as interval variables.

In this work, methods are developed for robustness assessment and robust mechanism synthesis when both random and interval variables exists. The average mean value, average standard deviation and the difference between the maximum and minimum standard deviations are used to measure the robustness of the mechanism performance. The robustness is evaluated by double loop Monte Carlo simulation. In the synthesis process, the average of mean performance, the average standard deviation of the performance, and the difference between the maximum and minimum standard deviations of the performance are minimized simultaneously. The feasibility robustness of the mechanism is also maintained with a desired probability level in the worst case of the interval variables. The synthesis problems of a crank slider mechanism and a four bar mechanism are used to demonstrate the effectiveness of the proposed methods.

ACKNOWLEDGMENTS

I would like to thank Dr. Xiaoping Du for his constant support and encouragement towards my Masters Degree Program at UMR. His course laid the foundation for my knowledge and has been the greatest motivation factor towards my research. I am grateful to him for giving me an opportunity to work with him. He has always given me time for discussions from which I have learned the approach to analyze, understand and solve complicated problems. I am thankful to him for his excellent mentorship in terms of research guidance, academic advising and moral support.

I would like to thank my committee members Dr. Ashok Midha and Dr. K Chandrashekhara for their timely support and invaluable help. I would also like to thank them for reviewing my thesis and offering important feedback and suggestions.

I would like to thank Dr. Ashok Midha and Mr. Alan Pilch for giving me an opportunity to work as a graduate teaching assistant for the senior design course. I would like to thank National Science Foundation and Department of Mechanical Engineering for their financial support.

I would like to express special thanks to my parents Mr. Krishna Venigella and Mrs. Lakshmi Rajyam Venigella for their blessings and good wishes, and to my brother Mr. Chandan Venigella for his undying love and support all the way through. I would also like to thank all my friends and especially my roommates at Rolla for their unflinching moral support during my stay here.

I would like to thank everyone who helped me accomplish my tasks and achieve my goals towards completing my graduate degree program.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

NOMENCLATURE

Symbol	Description
$\boldsymbol{X}$	Vector of Random Variables
$\boldsymbol{Y}$	Vector of Interval Variables
$\boldsymbol{d}$	Vector of Design Variables
Z	Response Variable
$\bar{Y}$	Midpoint of Y
δY	Width of Interval Y
σ_Z	Standard Deviation of Z
μ_Z	Mean Value of Z
$\bar{Z}$	Midpoint of Z
δZ	Width of Interval Z
$\bar{\mu}_Z$	Average of Mean Values of Z
$\bar{\sigma}_Z$	Average of Standard Deviations of Z
$\delta\sigma_Z$	Standard Deviation Difference of Z
N	Number of Samples of Random Variables
N_i	Number of Intervals

1. INTRODUCTION

Robust design [1,2,3,4] is a powerful design method for achieving high quality and productivity. By assessing variations (uncertainties) that a product experiences during design, manufacture, and operation, robust design ensures that a product perform its intended function regardless of the variations [5]. Due to various uncertainties, the actual product performance will always deviate from the desired or designed values [6]. If the uncertainties are not properly handled during the design process, the product performance will exhibit large variations and therefore deteriorate the product quality and reliability [7].

The fundamental principle of robust design is to improve product quality or stabilize the product performance by minimizing the effects of variations without eliminating their causes [8,9,10,11]. Therefore, robust design can achieve high quality by just changing design variables at the design stage without using tight tolerances [12].

1.1. UNCERTAINTY

Uncertainty is the difference between the model prediction and actuality [13,14]. Uncertainty can be also viewed as the deviation of an observed or calculated value from the true value. Uncertainty could occur in many ways in a system. Uncertainty could occur in the parameters of a mathematical model of a system or could be in the sequence of possible events in a discrete event system.

Uncertainty is generally distinguished as aleatory uncertainty and epistemic uncertainty [13,14].

1.1.1. Aleatory Uncertainty. Aleatory uncertainty also termed as objective or stochastic uncertainty, describes the inherent variation associated with the physical system or the environment under consideration. Sources of aleatory uncertainty are from a complex physical phenomenon, which includes temperature variations, material properties, dimensions of a product caused by manufacturing imprecision, environmental conditions, etc. Since uncertainty is resulted from natural variability, it will be very expensive and time consuming to reduce the uncertainty or sources of uncertainty. It is impossible to nullify aleatory uncertainty. Aleatory uncertainty is usually modeled by the probability theory [13,14].

1.1.2. Epistemic Uncertainty. Epistemic uncertainty is described mainly as lack of knowledge or information in any phase or operation of a design process. Epistemic uncertainty derives from some level of ignorance or incomplete information about a physical system or environment. This definition stresses on the key aspect that the fundamental cause of epistemic uncertainty is incomplete information or incomplete knowledge of some characteristic of a system or the environment. This indicates that epistemic uncertainty can be reduced by gaining knowledge or information of a system or environment.

Some of the sources of epistemic uncertainty are insufficient or no experimental data available and limited understanding of physical processes. Epistemic uncertainty can be modeled by probability or "non probability" theories.

For any particular physical system of interest which is mathematically modeled, uncertainty can be conveniently classified into parameter uncertainty and model structure uncertainty. Parameter uncertainty can be aleatory (due to inherent variation) or epistemic (due to limited information) in the physical system or environment in assessing the parameter characteristics. For example, if the length of a shaft, one of the parameters in a system varies around its nominal value within its specified tolerance with a normal distribution, the parameter uncertainty with the length is aleatory. Stochastic parameters in a specified mathematical model can be aleatory in nature. Due to variant operational environment, the external force *F* as shown in Figure 1.1 is an uncertain variable. If there are sufficient data available, *F* can be described mathematically with a random distribution. In this case, the parameter uncertainty is aleatory. However, if the data available is insufficient, *F* may not be precisely modeled by a random distribution. Then *F* has epistemic parameter uncertainty.

Figure 1.1. A Cantilever Beam

Model structure uncertainty is epistemic in nature because it is the uncertainty in the model structure itself. Model structure uncertainty is a special type of epistemic uncertainty as it concerns actual structural changes, or selection of one model among a class of models. This type of uncertainty comes from lack of knowledge, simplification and assumptions in the model building process.

To conclude, uncertainty is the variation of model prediction from actuality. Uncertainty can be classified as parameter uncertainty which can be aleatory or epistemic in nature and model structure uncertainty as shown in Figure 1.2. Model structure uncertainty is totally epistemic in nature. To control model structure uncertainty, the designer must select a model which fulfills the design requirements even under variations in the system or environment.

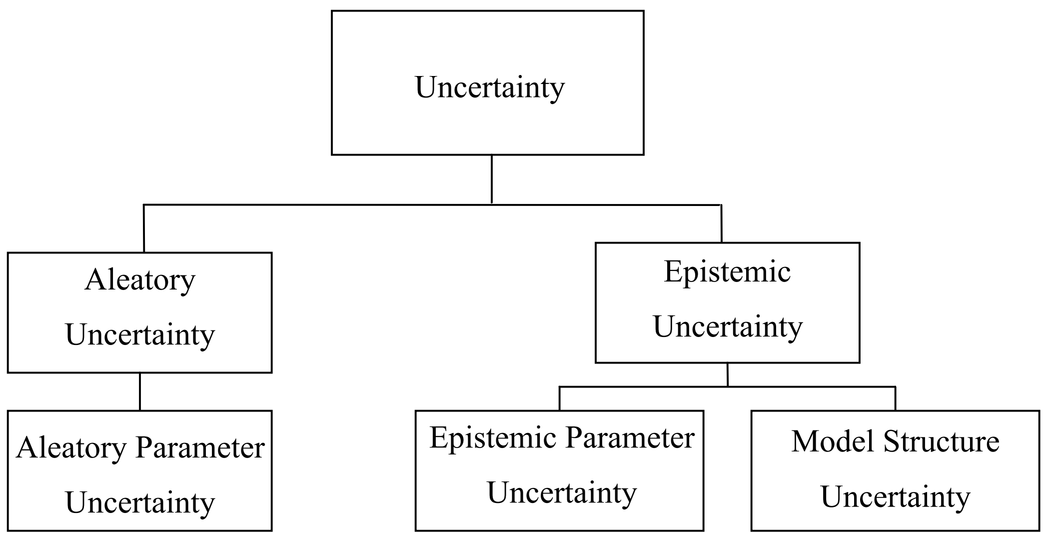

Figure 1.2. Classification of Uncertainty

1.2. UNCERTAINTY IN MECHANISMS

As in other design problems, there are many uncertainties in a mechanism synthesis problem [15,16,17,18,19,20]. For example, the dimension of a link in a mechanism is always random no matter how small its tolerance is. This kind of uncertainty is due to manufacturing imprecision. Installation errors also exist. The input motion, such as the angular velocity of a motor, is not deterministic. The external forces may be stochastic. All these uncertainties result in variations in the mechanism performance. To deal with the uncertainties, robust design has been introduced in mechanism synthesis [21,22].

There are two different methods for robust mechanism synthesis. They are probabilistic mechanism synthesis [9] and interval mechanism synthesis [23]. In the probabilistic robust mechanism synthesis, all the uncertain variables are treated as random variables. The robustness for the objective function is achieved by minimizing its standard deviation. Some constraints are maintained at desired probability levels. In the interval robust mechanism synthesis, uncertain variables are assumed within intervals. The reason of using intervals is due to limited information about the uncertain variables. Without adequate information, it is difficult to obtain the distribution. In other circumstances, uncertainties may not be due to randomness. Therefore, intervals are used to model uncertainties. The robustness is achieved by minimizing the range (width) of the objective function. Some of the constraints are maintained on their worse bounds.

In many applications, both random variables and intervals may exist. For example, for a new mechanism design, the installation errors and operation condition may not be known in advance. Intervals are usually used for the associated parameters. Since it is well known that dimensions are normally distributed, uncertainties associated with dimensions are modeled by normal distributions. In this case, both random and interval variables are present. The treatment of the mixture of such a mixture in reliability-based design has been reported recently [24,25,26]. A reliability based design is proposed to deal with the uncertain variables characterized by the mixture of probability distributions and interval variables. The reliability is computed at the worst case combination of interval variables.

In this work, one investigates possible ways for robust mechanism synthesis when both random and interval variables are involved. The tasks include how to define robustness with random and interval variables, how to evaluate robustness, and how to achieve robustness for mechanism synthesis under an optimization framework.

1.3. OVERVIEW OF THESIS

The rest of the thesis is organized as follows:

In Section 2, some background information of random variables and interval variables is presented, which lays the foundation of the proposed work.

In Section 3, robustness is assessed with only random variables and with only interval variables. A robustness assessment method is developed by the double loop Monte Carlo Simulation, when both random variables and interval variables are present.

In Section 4, the robustness assessment is integrated with the nonlinear optimization to achieve the mechanism robustness.

In Section 5, the validation of proposed methodology is done with two examples. The first example is the design of a slider crank mechanism and the second example is the design of a four bar mechanism.

In Section 6, concludes are made and future research directions are given.

2. UNCERTAINTY MODELING

As mentioned earlier, uncertainty could occur in many ways in a system. There is a need for a precise method of quantifying uncertainty. By quantifying uncertainty precisely, robust design ensures that a product performs its intended function regardless of the uncertainties. In recent years, a number of approaches have been proposed in the literature to the better representation of uncertainty [14,27,28,29,30,31]. The uncertainties associated with the mechanisms can be modeled using probabilistic or interval methods. In probabilistic approach, uncertainty is treated as random variable following a specific probability distribution [23,24,32,33]. In interval approach, uncertainty is denoted by a simple range [23,34]. If the uncertainty in the parameter is aleatory in nature, probability approach can be used to model the uncertainty. If the parameter uncertainty is epistemic in nature, interval approach can be used to model the uncertainty.

Current robust design methodologies treat the uncertainties as either aleatory or epistemic in nature [9,21,23]. But in reality the uncertainties can occur as both aleatory and epistemic in nature [24,32]. In other words, random and interval variables can be mixed [24,25]. If such a problem arises, one option is to consider all the variables as random by assigning probability distributions to interval variables. The other option is to consider all the variables as interval variables by converting random variables to intervals. Both the methods may lead to a misleading result as the uncertainties are not modeled accurately. In this work, uncertainty is treated as a mixture of random and interval variables. In this section, an introduction to random variables and interval variables is given. Some formulations for random and interval variables are shown, which are used in later sections.

2.1. RANDOM VARIABLE

Formally, a random variable is defined as a function where a real value is assigned to every possible outcome for an experiment or an engineering system. The random variable can also be described as a variable whose values are numerical outcomes of a random phenomenon. A probability distribution is assigned to a random variable. The common examples of a random variable are length of a shaft, time taken for completing a project and the life of an electronic component. In this thesis, an uppercase letter denotes a random variable, a lower case letter denotes an observation of a random variable, and bold letter denotes a vector. Random variables are used to model aleatory uncertainty. As learned from the previous section, aleatory uncertainty is an uncertainty where sufficient data are available.

2.1.1. Probability Density Function (PDF). Consider a random variable X, its probability distribution is the measure of probability of X on its range. Because of the physical phenomenon or data patterns, different variables may follow different probability distributions. A probability density function (PDF) fully describes a continuous random variable by defining the probability of its occurrence. The PDF of a random variable X is denoted using $f_X(x)$. The PDF of X over an interval $[x, x+\delta x]$, can be expressed as

$$P(x \le X \le x+\delta x) = f_X(x)\delta x \tag{2.1}$$

PDF of X over a finite interval $[a,b]$ can also be determined as

$$P(a \le X \le b) = \int_a^b f(x)\, dx \tag{2.2}$$

which is the area underneath the curve $f(x)$ from $x = a$ to $x = b$ as shown in Figure 2.1.

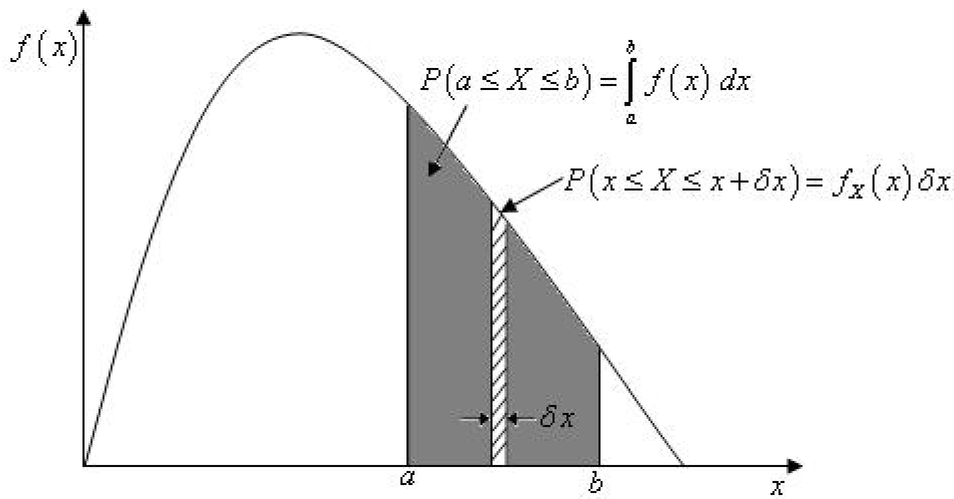

Figure 2.1. Probability Density Function

2.1.2. Cumulative Distribution Function (CDF). CDF of a random variable X represents the probability that the random variable X takes on a value less than or equal to a constant x. CDF is denoted by $F(x)$ which is given by,

$$F(x) = P(X \le x) = \int_{-\infty}^{x} f(x)\, dx \tag{2.3}$$

The CDF is the area underneath the PDF curve in the range of $(-\infty, x]$ as is shown in the Figure 2.2.

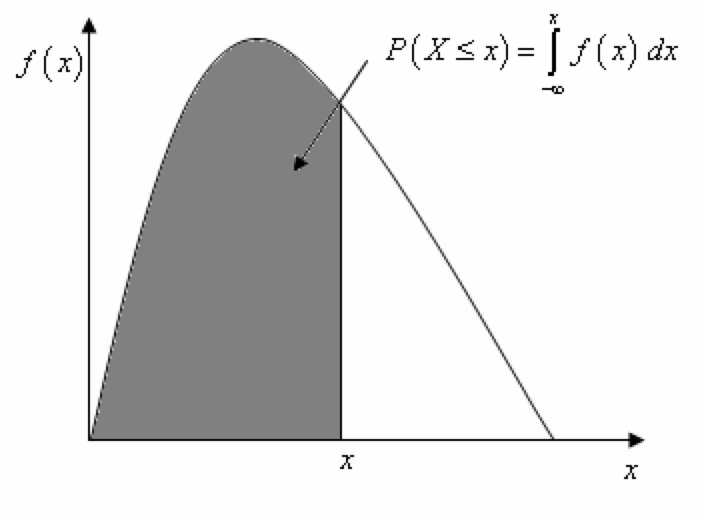

Figure 2.2. Cumulative Distribution Function

2.1.3. Normal Distribution. Normal distribution has the shape of the classic bell curve as shown in Figure 2.3. Any random variable with a normal distribution has a mean μ_x and standard deviation σ_x. The standard deviation is smaller for data deviating less from the mean value and larger for more dispersed data set.

The PDF of a normal distribution is expressed as

$$f_X(x) = \frac{1}{\sigma_x\sqrt{2\pi}} \exp\left[\frac{-1}{2}\left(\frac{x-\mu_x}{\sigma_x}\right)^2\right], -\infty < x < +\infty \tag{2.4}$$

The CDF of a normal distribution is expressed as

$$F(x) = \frac{1}{\sigma_x\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left[\frac{-1}{2}\left(\frac{x-\mu_x}{\sigma_x}\right)^2\right] dx, -\infty < x < +\infty \tag{2.5}$$

Examples of normal distribution include dimensions of a product, measurement errors, intensity of light and financial indicators such as stock values (or) commodity prices.

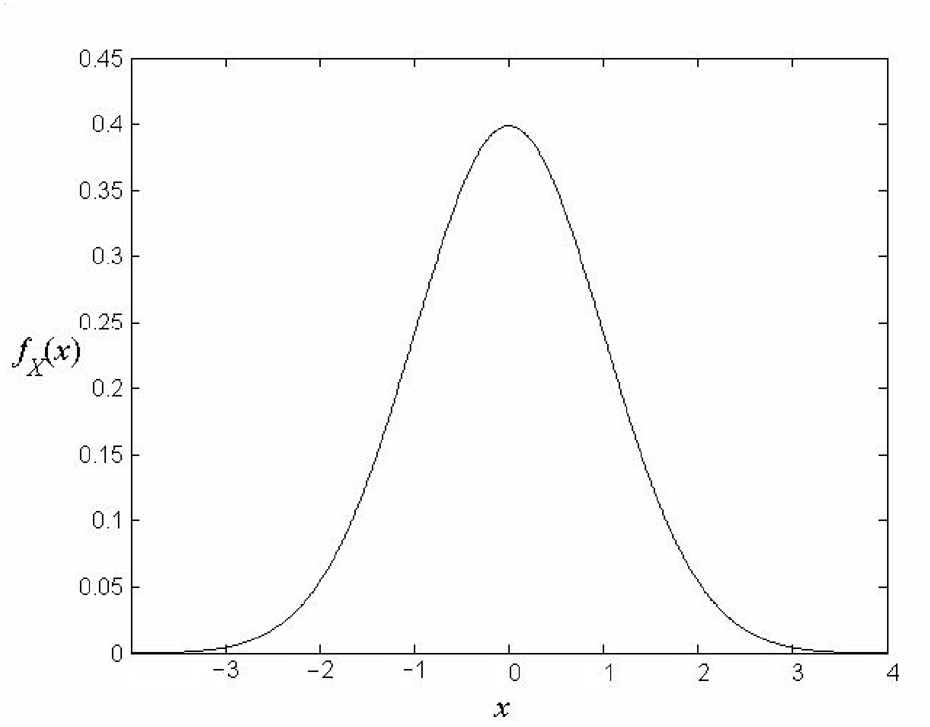

Figure 2.3. PDF of Normal Distribution

2.2. INTERVAL VARIABLE

In real life engineering systems, situations may arise where only limited information of a variable can be obtained. The only information known is the range in which the design variable lies. In such cases, treating them as random variables by assuming a distribution may lead to misleading results. So, they are treated as interval variables. Some of the situations where the design variables can be treated as interval variables are given below.

i. Suppose a new system is designed. Complete information on some of the quantities in the system may not be known. The only information known is the possible ranges of the quantity [35].

ii. The time of failure of a component. If a component fails in between two inspections, then it can be said that the time of failure of the component is in between those two inspections. The range of time is known, which can be treated as an interval variable.

iii. Measurement from a device can be with in an interval [24]. When the measurement from a device is between two adjacent landmarks, the only information available is that the reading is in a range.

iv. Intervals are used in many engineering formulations. For example, the coefficient of friction of a material can be with in a range. No information is available on how it is distributed within the range [24].

v. With the advancement of computers, most of the engineering analysis such as finite element analysis and kinematic analysis is done using computers. The error induced in the result will be with in a range [24].

Let a design variable Y be an interval variable as shown in Figure 2.4. Let Y_L and Y_U be the lower bound and upper bounds of Y. Then Y can be defined as [36]

$$Y = [Y_L, Y_U] = \{y \in \mathbb{R} \mid Y_L \leq y \leq Y_U\} \tag{2.6}$$

where $Y_L, Y_U \in \mathbb{R}$ and $Y_L \leq Y_U$.

Now, some simple arithmetic operations that can be done on Y are presented. The midpoint of interval $\overline{Y}$ is given by,

$$\overline{Y} = \frac{1}{2}(Y_L + Y_U) \tag{2.7}$$

The width of interval δY is given by,

$$\delta Y = Y_U - Y_L \tag{2.8}$$

The radius of interval $r(Y)$ is given by,

$$r(Y) = \frac{1}{2}(Y_U - Y_L) \tag{2.9}$$

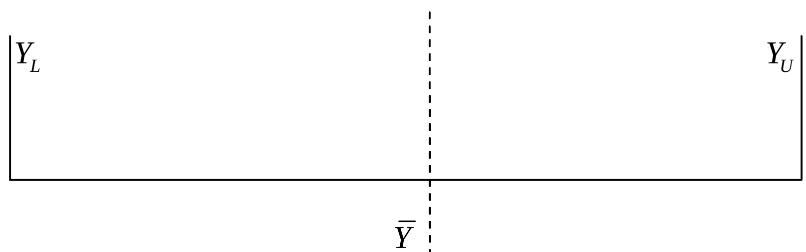

Figure 2.4. Interval Variable Y

In this work, uncertainty is to be modeled when both random and interval variables exist at the same time. The complexity of the problem increases with the mixture of random and interval variables. In the next section, the proposed method of robustness assessment in such situations is shown. Then, the robust mechanism synthesis with mixture of random and interval variables is introduced.

3. ROBUSTNESS ASSESSMENT

Every mechanism is subjected to uncertainties [21]. Uncertainties can be in the form of dimensional tolerances in the links, clearances in the joints and so on. The output of the mechanism is affected due to the uncertainties. Probabilistic, fuzzy or interval methods are generally used to model the uncertainties in an engineering system [34]. The probabilistic method describes the uncertain parameter as a random variable following a specific probability distribution [23,24,29,33,37]. If the information about the probability distribution is not available, interval approach or fuzzy theory can be used [38]. In interval approach, the uncertainty of the parameter is denoted by a simple range [23,24,34]. In fuzzy theory, the desirability of using different values within the range is described by using a membership function to the range [23]. The interval approach can be conveniently used when there is no sufficient information available about the probability distribution of the uncertain variable. Current literature [14,24,25] states that, in many engineering applications such as mechanisms the uncertain variables can be in the form random variables and interval variables at the same time.

If the variation in the output caused by uncertainties is ignored, nonrobust designs can result [39]. Taguchi introduced the concept of robust design [9,12,32,40]. Robust design tries to achieve a minimum variation in the output by controlling the parameters causing the variation [40]. The main objective of robust design is to “optimize the mean performance” and “minimize the performance variation due to uncertainties” [21,41]. The former can be achieved by finding the relation between the mean performance and the design variables. Here the challenge lies in precisely quantifying the performance variation due to the uncertainties, which is known as robustness assessment. Robustness assessment can be easily done if all the design variables are treated as random variables or interval variables. But in practical applications both random variables and interval variables exist at the same time.

In this section, the existing methods for measuring and evaluating robustness with only random variables and with only interval variables are reviewed. The idea is extended to the situation where both random and interval variables are involved.

3.1. ROBUSTNESS ASSESSMENT WITH ONLY RANDOM VARIABLES

Mathematically, robustness is measured by the mean and variance (or standard deviation) of the performance [9,29,32,37]. Let a random variable Z be a response variable that represents a performance in mechanism synthesis as shown in Figure 3.1 and be in the form of

$$Z = g(\boldsymbol{X}), \tag{3.1}$$

where $\boldsymbol{X} = \left(X_1, X_2, \cdots, X_{n_X} \right)$ is the vector consisting of n_X random variables.

In this work, all the random variables in $\boldsymbol{X} = \left(X_1, X_2, \cdots, X_{n_X} \right)$ are assumed to be independent. The methods discussed in this paper are also applicable to correlated random variables. The elements of $\boldsymbol{X}$ can be design variables (e.g. dimensions of a mechanism) that can be controllable by a designer or noise factors that are uncontrollable (e.g. external forces).

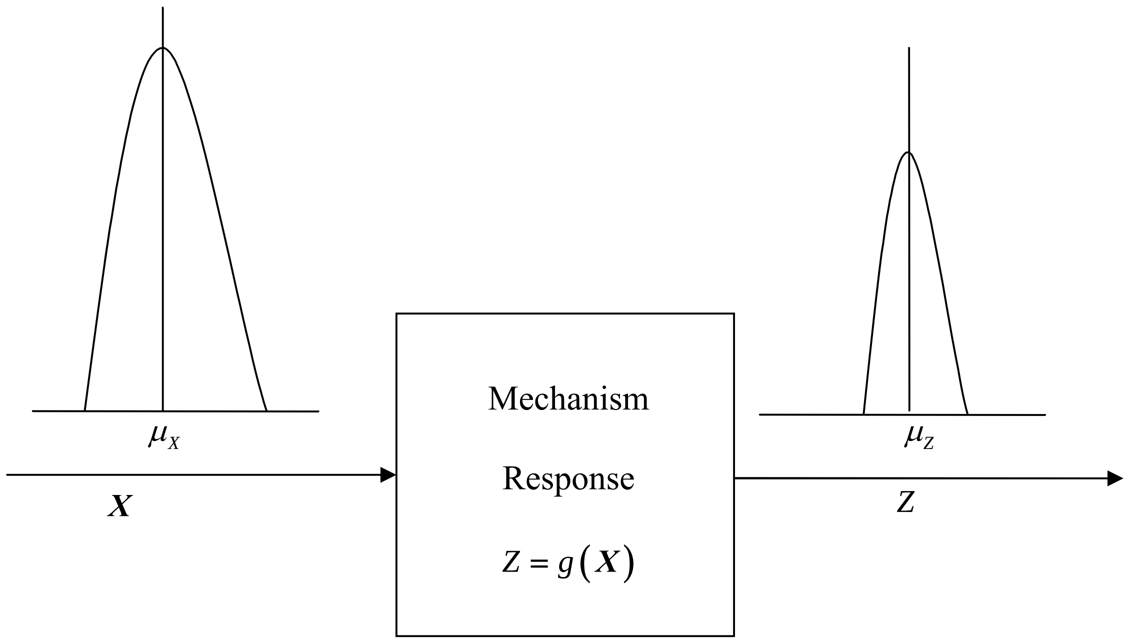

Figure 3.1. Robustness Assessment with only Random Variables

Theoretically, the variance σ_Z^2 of Z is calculated by

$$\sigma_Z^2 = E\left[\left(Z-\mu_Z\right)^2\right] = \int_{-\infty}^{\infty}\left[g(\mathbf{x})-\mu_Z\right]^2 f_{\mathbf{X}}(\mathbf{x})d\mathbf{x}, \tag{3.2}$$

where μ_Z is the mean of Z, which is computed by

$$\mu_Z = E[Z] = \int_{-\infty}^{\infty} g(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x})d\mathbf{x}. \tag{3.3}$$

In the above equations, $f_{\mathbf{X}}$ is the joint probability density function (PDF) of the random variables $\boldsymbol{X}$. Due to high dimensionality, analytical solutions to both of the above equations are difficult to obtain. Many approximation methods [37] and Monte Carlo simulation (MCS) [37] have been proposed. MCS is the simplest method and results accurate estimations. In this work we use MCS.

3.2. MONTE CARLO SIMULATION

MCS is useful to observe the dynamic behavior of a system with variation in the set of inputs. MCS is a powerful analysis tool that generates random numbers based on the probability density function (PDF) of the random variables $\boldsymbol{X}$ and simulates the behavior of a response variable Z when the data is insufficient to make decisions.

The outline of MCS is depicted in Figure 3.2. MCS contains three steps:

1. Sampling on random input variables according to their distributions.
2. Evaluating response variable Z at each sample.
3. Analyzing the response variable Z.

The response variable Z can be evaluated from equation 3.1. The estimate of mean and variance of Z is calculated from the samples of Z obtained from MCS. The equations are:

$$\sigma_Z^2 \cong \frac{1}{N-1}\sum_{i=1}^{N}\left[g(\mathbf{x}_i)-\mu_Z\right]^2 , \tag{3.4}$$

where the mean μ_z is estimated by

$$\mu_Z \cong \frac{1}{N}\sum_{i=1}^{N} g(\mathbf{x}_i) \tag{3.5}$$

$\mathbf{x}_i$ are the samples of random vector $\boldsymbol{X}$, which are drawn from the distributions of $\boldsymbol{X}$. N is the number of samples (simulations).

MCS is an iterative method that has an inherent error involved. The accuracy of MCS depends on the number of simulations N. A large number of simulations must be performed to achieve an accurate estimate. With the increase in the number of simulations, MCS demands a lot of computational power.

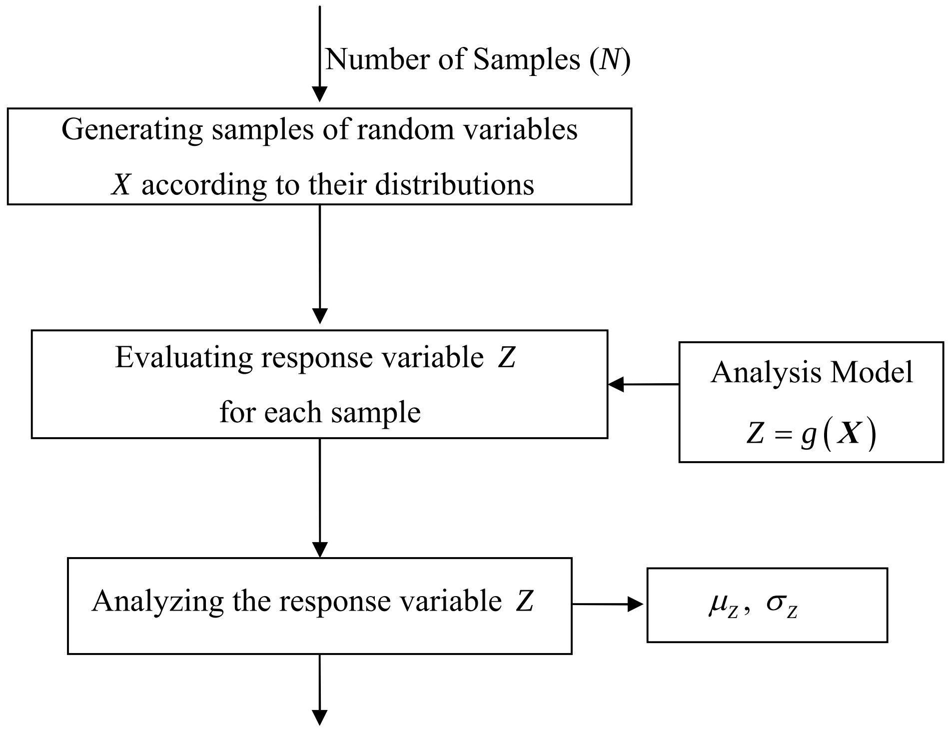

Figure 3.2. Monte Carlo Simulation

The robustness of a system is assessed by the standard deviation σ_Z obtained from MCS. For a robust system, a low standard deviation σ_Z value with a mean value μ_Z equal to the desired value is to be achieved. Design optimization techniques are used to achieve this, which are mentioned in Section 4. Consider two designs design A and design B as shown in Figure 3.3. Consider the two designs subjected to similar conditions. Both the designs met the primary requirement of mean value μ_Z, which is equal to the desired value. From Figure 3.3, it is evident that standard deviations σ_Z of both the designs are different. σ_{Z_A} (standard deviation of Design A) is less than

σ_{Z_B} (standard deviation of Design B), which suggests that Design A is more robust compared to design B.

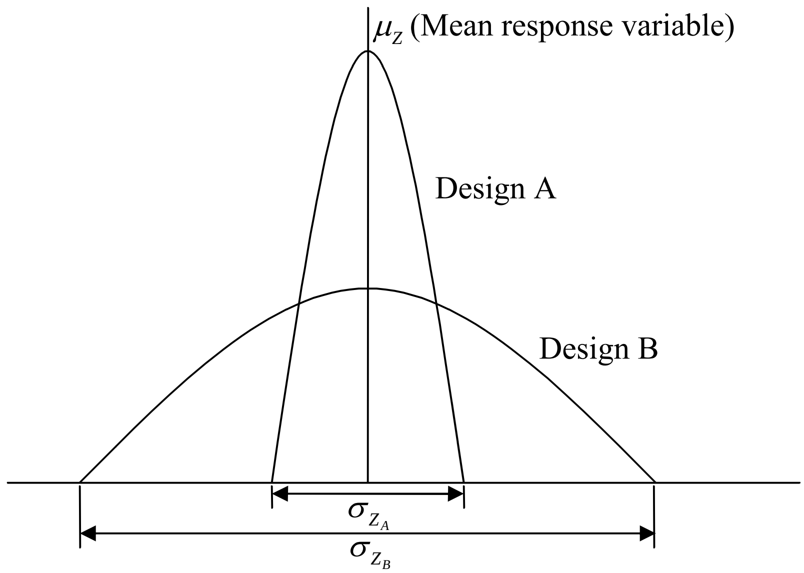

Figure 3.3. Robustness Comparison Between Design A and Design B

3.3. ROBUSTNESS ASSESSMENT WITH ONLY INTERVAL VARIABLES

Mathematically, robustness is measured by the width of the interval of the performance [23,34,36]. Let a variable Z be a response variable that represents a performance of a system as shown in Figure 3.4 and be in the form of

$$Z = g(\boldsymbol{Y}), \tag{3.6}$$

Where, $\boldsymbol{Y} = \left(Y_1, Y_2, \cdots, Y_{n_Y}\right)$ is the vector consisting of n_Y interval variables. In this work, all the interval variables in $\boldsymbol{Y} = \left(Y_1, Y_2, \cdots, Y_{n_Y}\right)$ are assumed to be independent. The elements of $\boldsymbol{Y}$ can be design variables that can be controllable by a designer or noise factors that are uncontrollable. From the concepts discussed in Section 2, the midpoint $\left(\bar{Z}\right)$ and width of the interval $\left(\delta Z\right)$ of Z is calculated by

$$\bar{Z} = \frac{1}{2}\left(Z_U + Z_L\right), \tag{3.7}$$

$$\delta Z = Z_U - Z_L, \tag{3.8}$$

where Z_U and Z_L represents the upper bound and lower bound of Z.

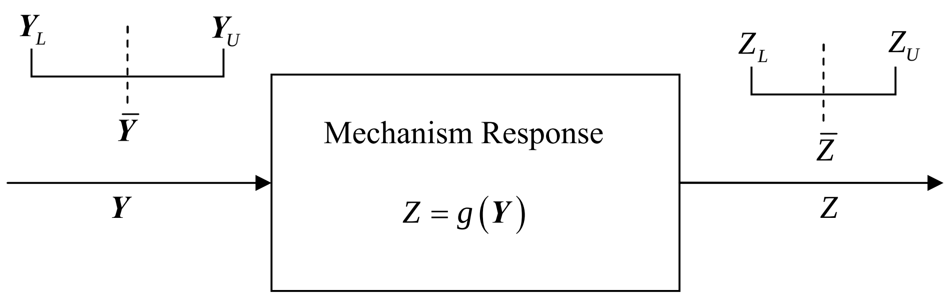

Figure 3.4. Robustness Assessment with only Interval Variables

The robustness of the system is assessed by the width of the interval δZ. δZ is to be as low as possible while $\bar{Z}$ equal to the desired value. Consider two designs as shown in Figure 3.5 subjected to similar conditions. The mid points for both the designs are equal to $\bar{Z}$, and satisfy the primary requirement. Now, comparing the width of the interval for the two designs, δZ_A is greater than δZ_B. That is, if the input variable $\boldsymbol{Y}$ is subjected to an uncertainty of δY. By using Design A the uncertainty in the output response would be δZ_A, and by using Design B the uncertainty in the output response would be δZ_B. As δZ_B is less than δZ_A, Design B is said to be more robust than Design A.

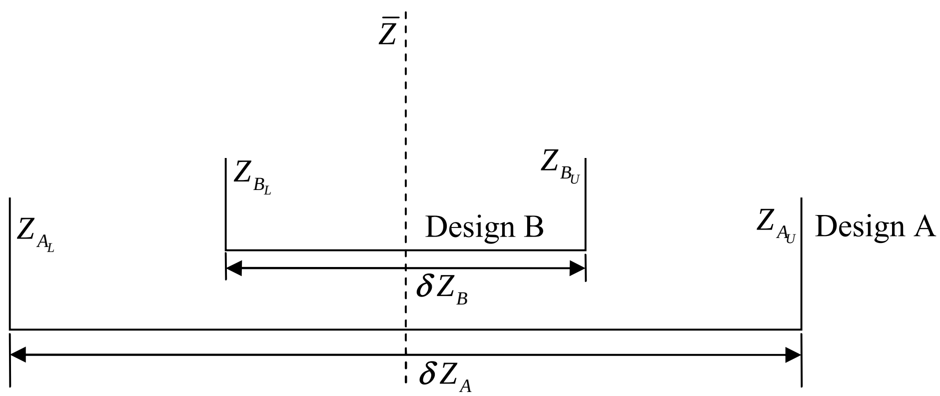

Figure 3.5. Robustness Assessment Between Design A and Design B

3.4. ROBUSTNESS ASSESSMENT WITH BOTH RANDOM AND INTERVAL VARIABLES

In current robustness assessment, uncertainties are usually treated as random variables or as interval variables [32]. However, in many practical engineering applications both random variables and interval variables exist at the same time. When the distributions of the design variables are precisely known, the design variables can be treated as random variables. If the evaluation of the probabilistic characters of a design variable is prohibitively expensive or may not be precisely known, the design variables can be treated as interval variables. In thesis work, a methodology of evaluating the robustness when both random and interval variables exist at the same time is proposed.

When both random variables $\boldsymbol{X} = \left(X_1, X_2, \cdots, X_{n_X}\right)$ and interval variables $\boldsymbol{Y} = \left(Y_1, Y_2, \cdots, Y_{n_Y}\right)$ exist, the model becomes

$$Z = g(\boldsymbol{X}, \boldsymbol{Y}) . \tag{3.9}$$

With the existence of intervals, the mean and standard deviation of Z will also be intervals. Figure 3.6 explains more about the existence of both random and interval variables in the design model.

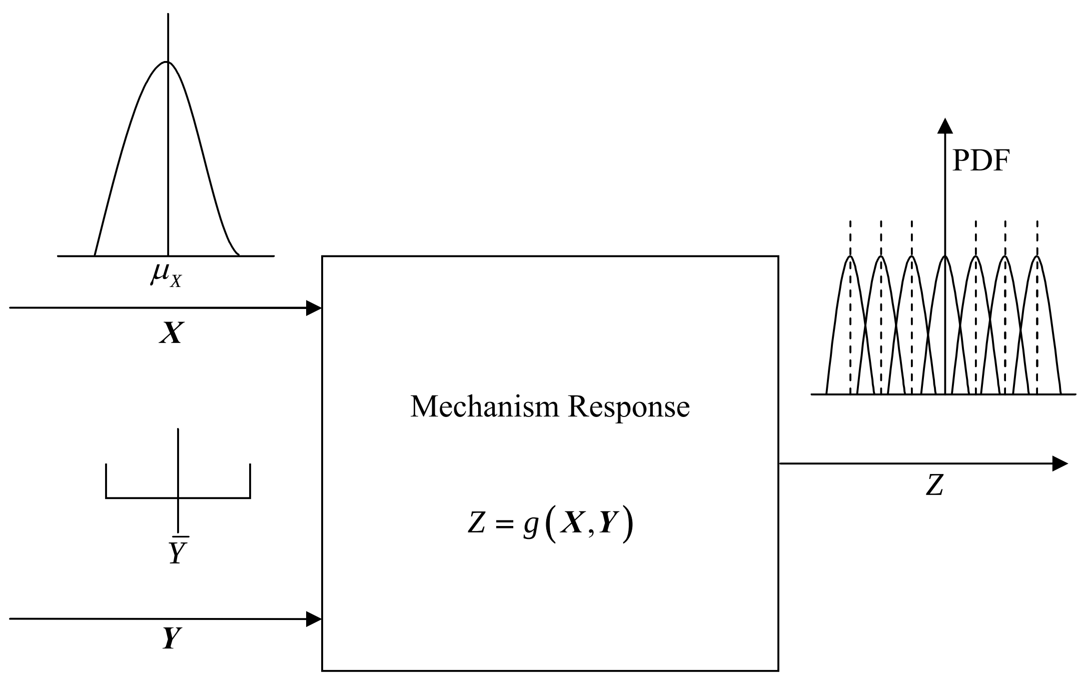

Figure 3.6. Mixture of Random and Interval Variables

The dashed lines represent the intervals of Z due to intervals in $\boldsymbol{Y}$. Because of the randomness in $\boldsymbol{X}$, at each interval of Z randomness is also seen. Consider a response variable Z which is dependant on the random variable $\boldsymbol{X}=(X_1, X_2)$ and interval variable $\boldsymbol{Y}=(Y_1,Y_2)$. Z can be given by the equation 3.9. Consider all the four combinations of the interval bounds for both the interval variables. At each combination, Z has four distributions as shown in Figure 3.7.

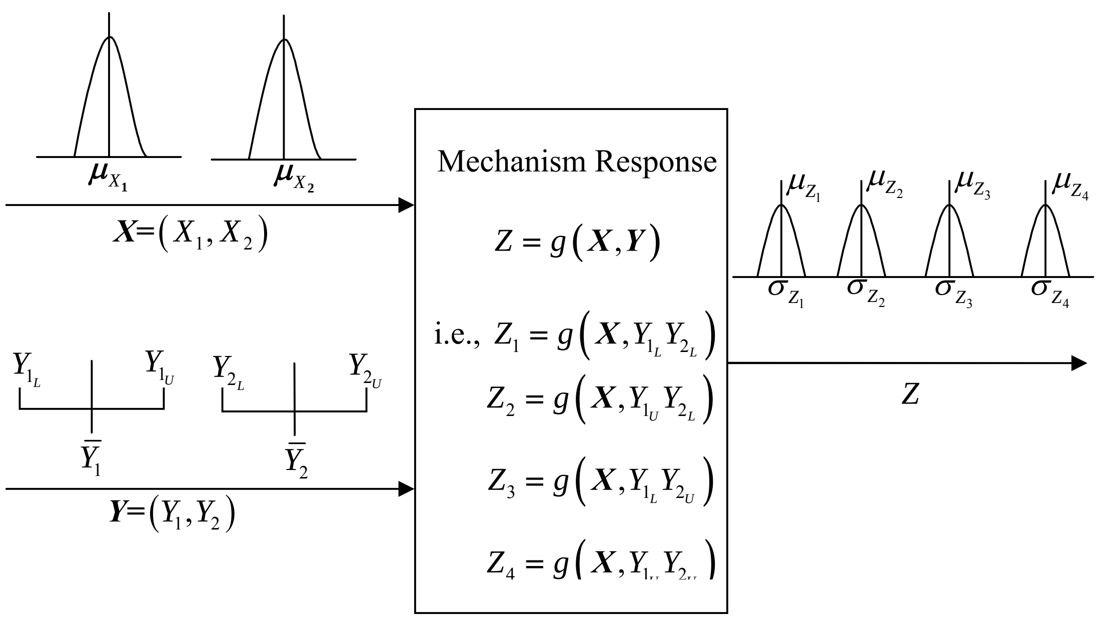

Figure 3.7. Example of Mixture of Random and Interval Variables

An interval of mean values and standard deviations for Z is obtained. The probability distributions indicate the uncertainty obtained due to the effect of random variable on Z. The intervals of distributions indicate the effect of interval variables on Z. From the interval of mean values of Z, the average mean value of Z can be calculated. The average of the mean values of Z is given by,

$$\bar{\mu}_z = \frac{1}{2}\left(\mu_z^{\max} + \mu_z^{\min}\right), \tag{3.10}$$

where $\mu_z^{\max}$ and $\mu_z^{\min}$ are the maximum and minimum means, respectively. $\bar{\mu}_z$ should be equal to the desired or expected value. Now the robustness of this type of model needs to be calculated. The interval of standard deviations of Z is used to assess the robustness. The standard deviations of Z and their bounds is to be considered. Imagine that there is no effect of random variable on Z, then Z will be in the form of intervals. So, the randomness in Z is due to the random variable $\boldsymbol{X}$. To quantify the effect of randomness on Z, the average standard deviation of Z is used. The average of the standard deviation is given by,

$$\bar{\sigma}_z = \frac{1}{2}\left(\sigma_z^{\max} + \sigma_z^{\min}\right), \tag{3.11}$$

where $\sigma_z^{\max}$ and $\sigma_z^{\min}$ are the maximum and the minimum standard deviations, respectively.

Now imagine that there is no effect on Z due to the interval variable. Then Z will be in the form of a probability distribution. So, the interval of randomness is due to the effect of interval variable $\boldsymbol{Y}$. To quantify the effect of interval variable on Z, $\delta\sigma_z$ is used, which is the difference between the maximum standard deviation $\sigma_z^{\max}$ and the minimum standard deviation $\sigma_z^{\min}$. It is similar to the width of an interval, a lower value of $\delta\sigma_z$ is desired to achieve a robust design. The standard deviation difference $\delta\sigma_z$ is computed by,

$$\delta\sigma_z = \sigma_z^{\max} - \sigma_z^{\min} \tag{3.12}$$

To understand more about how randomness is assessed when both random variable and interval variables are considered at the same time, consider four designs which are subjected to similar conditions. Figure 3.8 represents the maximum and minimum probability distributions from the interval of distributions.

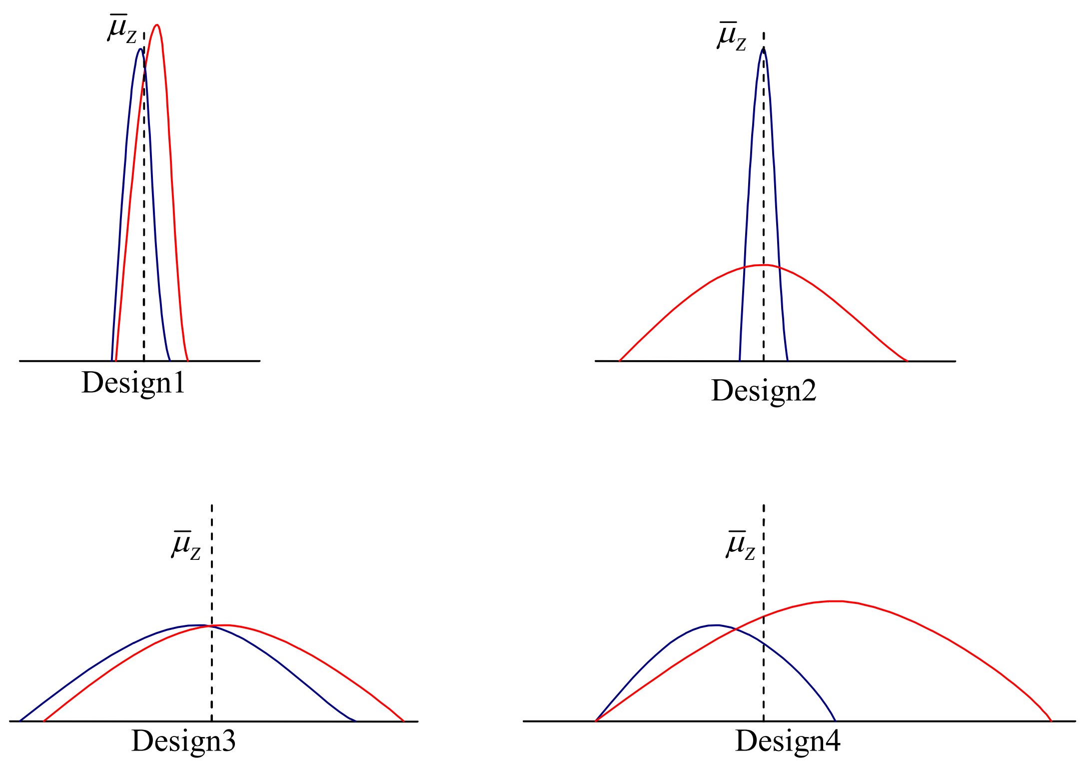

Figure 3.8. Robustness Assessment with a Mixture of Random and Interval Variables

The average of the mean values of Z $(\bar{\mu}_Z)$ is equal for all the four designs. Total uncertainty on the response variable can be divided as the uncertainty due to randomness $(\bar{\sigma}_Z)$ and the uncertainty due to interval variable $(\delta\sigma_Z)$. The design in which $\bar{\sigma}_Z$ and $\delta\sigma_Z$ is less is considered as a more robust design. First, $\bar{\sigma}_Z$ for the four designs is compared and then, $\delta\sigma_Z$ is compared. From the distribution curves, comparing $\bar{\sigma}_Z$ for the four designs, $\bar{\sigma}_{Z_1} < \bar{\sigma}_{Z_2} < \bar{\sigma}_{Z_3} < \bar{\sigma}_{Z_4}$. The effect of randomness on the response variable for Design1 is low compared with other designs. Design1 is a more robust design when only randomness is considered as the factor which affects the uncertainty of the response variable. But, the affect of interval variable on the uncertainty of the response variable is to be considered. From the distribution curves, comparing $\delta\sigma_Z$ for the four designs, $\delta\sigma_{Z_1} < \delta\sigma_{Z_3} < \delta\sigma_{Z_2} < \delta\sigma_{Z_4}$. From the two comparisons, Design1 is the robust design and Design4 is the non robust design of all the four designs. Design2 is robust than Design3 when uncertainty in the response variable is caused only due to randomness. Design3 is robust than Design 2 when uncertainty in the response variable is caused only due to the interval variable. In such cases decision is left to the designer whether to consider Design2 or Design3.

Therefore, the key of mechanism robustness assessment is to calculate the average standard deviation $\bar{\sigma}_z$ and the standard deviation difference $\delta\sigma_z$. It is seen from the equations 3.11 and 3.12 that the maximum and minimum standard deviations must be obtained. Therefore, optimization combined with MCS must be employed for accurate calculations. The method will be very computationally expensive. In this work, a double loop MCS method is used to calculate $\bar{\sigma}_z$ and $\delta\sigma_z$. Figure 3.9 shows the flow chart of a double loop MCS method. This method consists of an outer loop which evaluates the effect of interval variable on the uncertainty of response variable and an inner loop which evaluates the effect of random variable on the uncertainty of response variable. In the outer loop, all the interval variables are divided into a number of small intervals (N_i). The combinations of intervals $\left(Y_{1_1}Y_{2_1}...Y_{n_{Y1}}, Y_{1_2}Y_{2_1}...Y_{n_{Y1}}, ..., Y_{1_{N_i}}Y_{2_{N_i}}...Y_{{n_Y}_{N_i}}\right)$ are evaluated depending on N_i. There will be a total of $n_Y \times N_i$ combination of intervals. For each of the combinations an inner loop is performed. In the inner loop the samples of random variables $\boldsymbol{X}$ are generated according to their distributions. After evaluating Z for each sample, μ_Z and σ_Z are calculated. After completing the simulations the output contains $n_Y \times N_i$ number of μ_Z and σ_Z. If average of all the means is taken, $\bar{\mu}_Z$ is obtained. $\bar{\mu}_Z$ should be equal to the desired or expected value. The maximum $\left(\sigma_Z^{\max}\right)$ and minimum $\left(\sigma_Z^{\min}\right)$ of σ_Z values can also be identified from the obtained σ_Z values. After identifying $\sigma_Z^{\max}$ and $\sigma_Z^{\min}$, $\bar{\sigma}_Z$ and $\delta\sigma_Z$ can be calculated from the equations 3.11 and 3.12. From $\bar{\sigma}_Z$ and $\delta\sigma_Z$, robustness of a system can be assessed. A minimum value for $\bar{\sigma}_Z$ and $\delta\sigma_Z$ is desired for a robust design.

In the next section, discussions are made on how to achieve robustness for the mechanism synthesis under an optimization framework. First, the deterministic mechanism synthesis is reviewed. The existing methodologies, probabilistic method and interval approach for the robust mechanism synthesis are examined. Then the proposed method of robust mechanism synthesis is introduced. In the following section, the proposed method is validated using two examples.

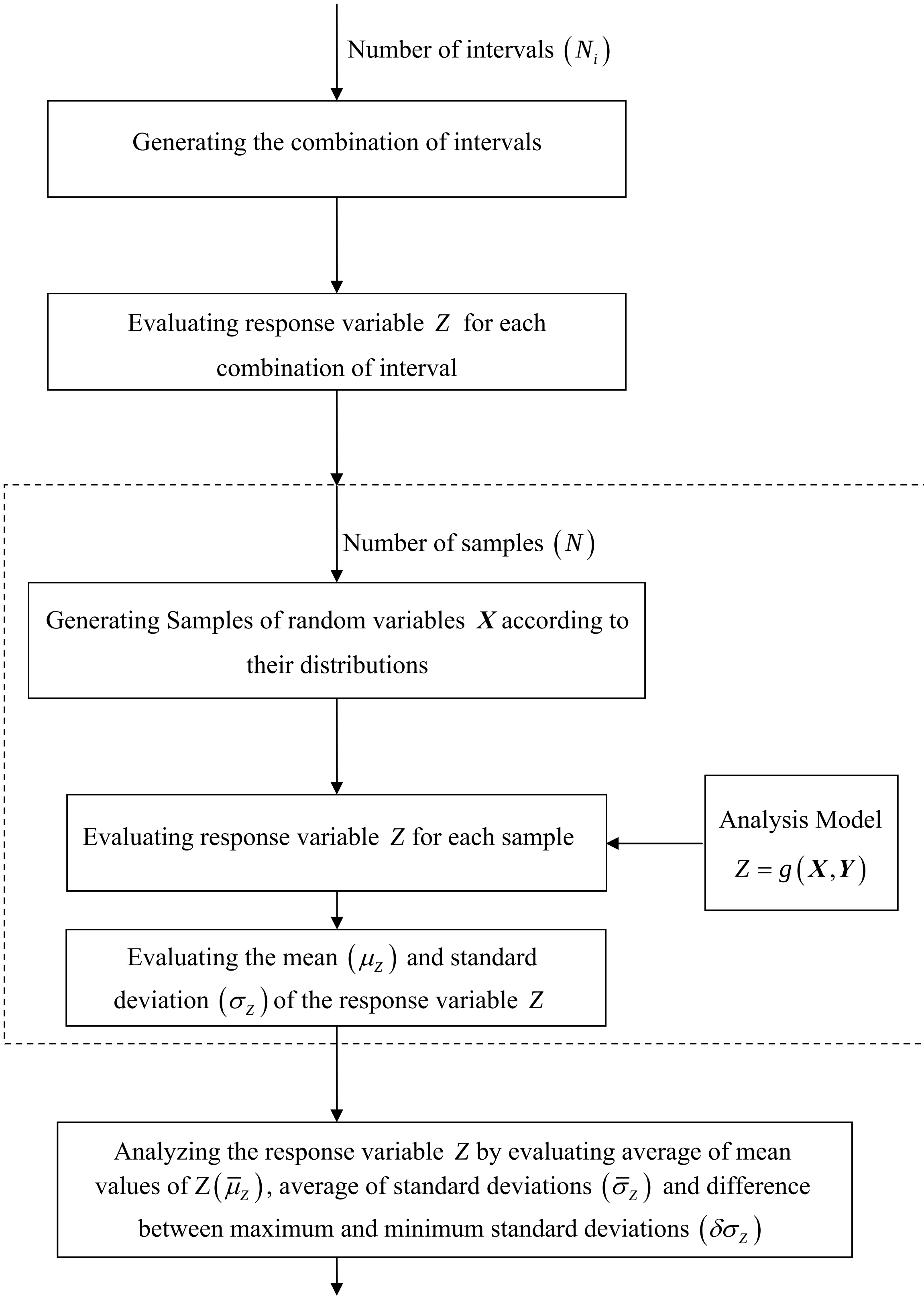

Figure 3.9. Double Loop Monte Carlo Simulation

4. ROBUST MECHANISM SYNTHESIS

Kinematics is defined by Ampère as “the study of the motion of mechanisms and methods of creating them” [42]. In this definition kinematics is divided into two parts. The first part deals with kinematic analysis and the second part deals with the kinematic synthesis. In kinematic analysis, the mechanism performance is determined with an assumption that all principal dimensions of a mechanism are known, the interconnections of the links are defined and the motion of the driver link is prescribed. Kinematic synthesis is the process of systematic design of a mechanism to achieve a specific task. The task that a mechanism should achieve can be one of the following.

i. Motion Generation: A rigid link of the given mechanism has to be guided in a prescribed motion sequence and the guidance may or may not be correlated with the input motion.

ii. Path Generation: In a path generation problem, a point on a coupler link (link which is not connected to the frame) has to be guided along a definite path. The generation of the path may or may not be correlated with the input motion.

iii. Function Generation: The motion parameters (displacement, velocity, acceleration, etc) of the input and output links are to be correlated so as to satisfy a desired function relationship.

Kinematic synthesis can be classified into two groups, Type synthesis and Dimensional synthesis. Type synthesis deals with finding the best suitable mechanism (cam mechanism, linkages, gear trains, etc), number of links the mechanism should have, number of degrees of freedom required and so on, to achieve the required performance. In type synthesis the uncertainty could occur due to lack of knowledge or ignorance. The uncertainty in type synthesis can be reduced by gaining knowledge in the system.

Dimensional synthesis deals with determining the significant dimensions of the mechanism to achieve a specific task. There are two methods in practice for the dimensional synthesis of mechanisms, graphical method and analytical method. In graphical method, the mechanism is constructed geometrically. Tough this method provides a fairly quick and straightforward method of design, it has some limitations of accuracy (due to drawing error) [42]. To achieve accurate results the geometric construction may need to be repeated many times which is a tedious and laborious process. Analytical method mathematically models the mechanism. Approximation techniques are used to solve the model. This method has an advantage of accuracy and repeatability. In this work, focus is on analytical method of mechanism synthesis.

Mathematical techniques such as algebraic method, matrix method and complex numbers are used to mathematically model the linkages for planar mechanism synthesis. After obtaining a mathematical model, optimization techniques are used to achieve an optimal solution to the problem. In the traditional optimization method, the error between the desired performance and the actual performance of a mechanism is to be minimized [43]. The optimization also includes a number of design constraints. In this section, the mechanism synthesis without considering any uncertainties is reviewed and then uncertainties are considered in the design stage. When considering uncertainties in the mechanism synthesis, the existing methods are shown and then the proposed method for robust mechanism synthesis is introduced.

4.1. DETERMINISTIC MECHANISM SYNTHESIS

As mentioned previously mechanism design is a systematic design of a mechanism to achieve a specific task. The task may be motion generation, path generation and function generation. Optimization techniques are used to achieve this task. To perform optimization techniques, first the main objective of the design and the design variables is to be identified. Then, the constraints of the design need to be identified. The objective may be minimization of the difference between the desired path and the actual path of a mechanism. The design parameters may be the dimensions of the links and the constraints may be the existence of crank and transmission angle.

Suppose the objective $f(\boldsymbol{d})$ of a mechanism with design parameters $\boldsymbol{d} = (d_1, d_{2,} ....., d_n)$ is to be minimized. Let the mechanism is subjected to the design constraints $g_i(\boldsymbol{d}) \le 0$ $(i = 1, 2, ..., n_g)$ and $h_j(\boldsymbol{d}) = 0$ $(j = 1, 2, ..., n_j)$. When uncertainties are not considered, the optimal design model of the synthesis problem is given by [5,9]

$$\begin{cases} \min\limits_{\boldsymbol{d}} \; Z = f(\boldsymbol{d}) \\ \\ \text{s.t.} \quad g_i(\boldsymbol{d}) \le 0, \qquad i = 1, 2, ..., n_i \\ \qquad\;\; h_j(\boldsymbol{d}) = 0, \qquad j = 1, 2, ..., n_j \\ \qquad\;\; \boldsymbol{d}_k^l \le \boldsymbol{d}_k \le \boldsymbol{d}_k^u, \quad k = 1, 2, ..., n \end{cases} \tag{4.1}$$

where $\boldsymbol{d}$ is the vector of deterministic design variables. Figure 4.1 shows the flow chart of the optimization model. The design constraints and the objective function are checked for the initial start point. If the constraints are not satisfied and the objective is not minimal, the design variables are changed. The process is iterated until an optimal solution for the design problem is achieved.

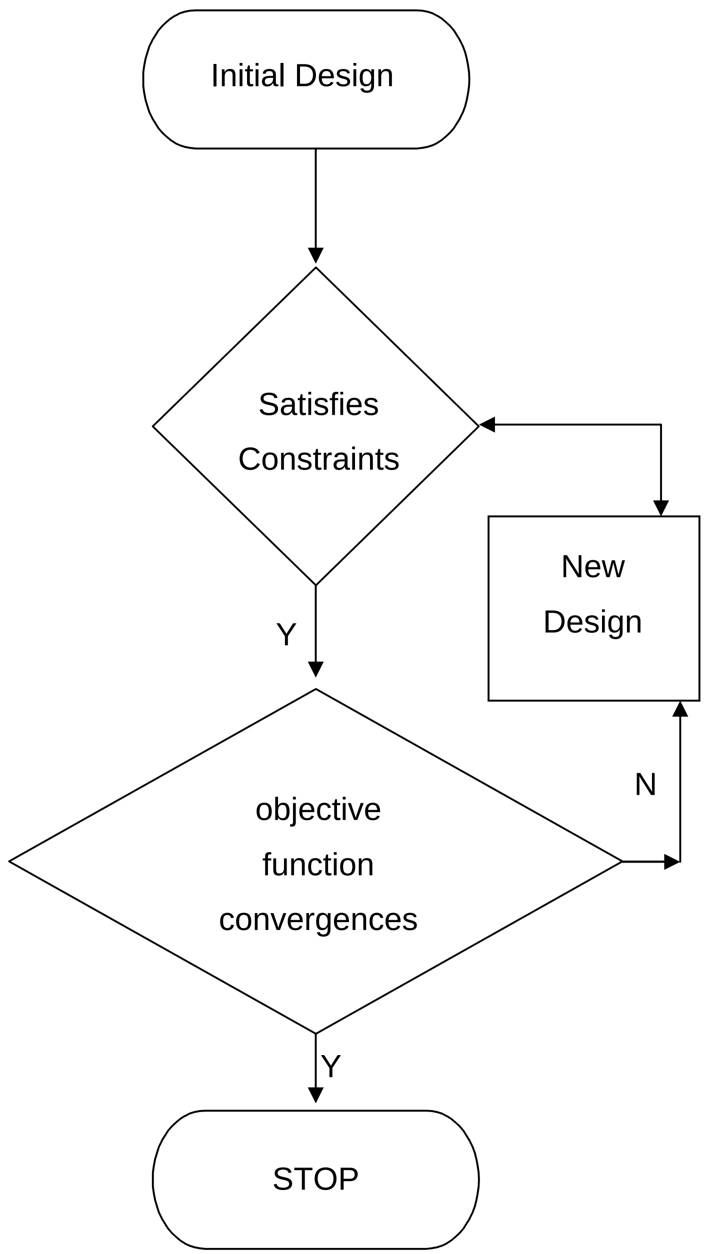

Figure 4.1. Flowchart of Deterministic Optimization

To understand more consider a mathematical example. Suppose a company manufactures a product. \$6000 is allocated for purchasing labor and material. Unit cost of labor and material is \$20 and \$10, respectively. The company will produce d_1d_2 units of products, where d_1 and d_2 are number of units of labor and material respectively.

To formulate this problem mathematically into an optimization model, first the objective which is dependent on the design variables need to be identified and then the constraints of the problem. The number of units of labor and material are to be determined, so $\boldsymbol{d} = (d_1, d_2)$ are the design variables. Our objective is to produce maximum number of units, which is given by d_1d_2. d_1d_2 can be maximized or $-d_1d_2$ can be minimized. The constraint is not to exceed the expenditure in labor and material above \$6000. The constraint can be mathematically modeled as $g(\boldsymbol{d}) = 20d_1 + 10d_2 \leq 6000$. The optimization model is given by

$$\begin{cases} \min\limits_{\boldsymbol{d}} \; -d_1d_2 \\ \text{s.t.} \quad g(\boldsymbol{d}) = 20d_1 + 10d_2 - 6000 \leq 0 \\ \qquad 10 \leq d_1 \leq 500; \\ \qquad 10 \leq d_2 \leq 250 \end{cases} \tag{4.2}$$

In this work, MATLab is used for solving the optimization model. An optimization tool “fmincon” which is available in MATLab is used. The starting point for the optimization is taken as $d_1 = 50$ and $d_2 = 50$. The solution steps obtained from MATLab are given in Table 4.1.

Table 4.1. Solution Steps from MATLab

Iteration	d_1	d_2	$d_1 d_2$
1	1	1	1
2	2	3	6
3	3	500	1,500
4	50	500	25,000
5	150	300	45,000

The optimal solution for this problem is 150 units of labor and 300 units of material. In this optimization methodology, uncertainties in the design parameters are not considered. The nominal values of the design parameters are taken into account. But in engineering applications, some uncertainties are present in the design parameters [18].

4.2 ROBUST MECHANISM SYNTHESIS

As mentioned, due to the uncertainties in the design variables the mechanism performance deviates from the designed value. The uncertainty in the design variables needs to be considered at the design stage to achieve a robust mechanism [18]. The variation in the mechanism performance due to uncertainty can be quantified by the methods which are shown in the previous chapter. First the design variables are treated as random variables. Then, the proposed method of treating uncertainty as both random and interval to assess robustness is introduced.

4.2.1. Robust Mechanism Synthesis with only Random Variables. When the uncertainty in the design variables is treated as random variables, the robustness can be quantified by the measure of standard deviation. For a robust mechanism a minimum standard deviation value is to be achieved. The objective of a robust mechanism synthesis would be not only to minimize the error between the desired performance and actual performance but also the variation of output performance due to the uncertainties in the design variables [9]. Mathematically, our design objective for a robust mechanism can be represented as $f\left(\boldsymbol{X}\right)=w_1\mu_Z+w_2\sigma_Z$. μ_Z represents the mean performance error of mechanism and σ_Z represents the standard deviation of mechanism performance. To calculate σ_z, MCS is used in the optimization loop, as shown in Figure 4.2. w_1 and w_2 are the weighting factors. The constraint function changes to $\mu_{g_i}+k\sigma_{g_i}\le 0\left(i=1,2,..,n_i\right)$. μ_{g_i} is the mean value and σ_{g_i} is the standard deviation of the constraint $g_i\left(\boldsymbol{X}\right)$. μ_{g_i} and σ_{g_i} can be calculated using MCS. k is a constant, If $g_i\left(\boldsymbol{X}\right)$ is assumed to be normally distributed, $\Phi(k)$ is the probability of confidence of the constraint satisfaction, where Φ is the cumulative distribution function of a standard normal variable. Therefore, if $k=3$, the constraint will be satisfied at the probability of $\Phi(3)=0.99865$. The optimization model for the mechanism synthesis can be represented as [5,9,29,41]

$$\left\{\begin{array}{lll} \min\limits_{\boldsymbol{\mu}_X} & w_1\mu_Z+w_2\sigma_Z & \\ \text{s.t.} & \mu_{g_i}+k\sigma_{g_i}\le 0, & i=1,2,....,n_i \\ & h_j\left(\boldsymbol{X}\right)=0, & j=1,2,....,n_j \\ & \boldsymbol{X}_k^l\le\boldsymbol{X}_k\le\boldsymbol{X}_k^u, & k=1,2,....,n \end{array}\right. \tag{4.3}$$

where X is a vector of random variables. Figure 4.2 shows the flowchart of the optimization model when uncertainties are considered as random variables. The constraint function and convergence of objective function for an initial design is checked. If the functions are not satisfied, the design is changed. This process is iterated until an optimal solution is obtained.

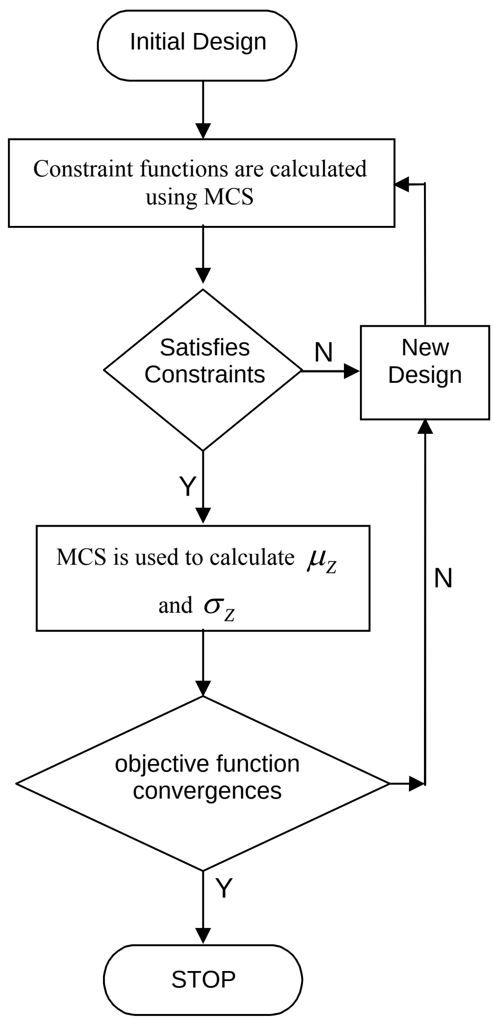

Figure 4.2. Flowchart of Optimization Model when Uncertainties are Considered as Random Variables

To understand more about the difference between the deterministic mechanism synthesis and robust mechanism synthesis with random variables, consider a simple mechanism synthesis problem. Suppose a two position synthesis is to be done on a four bar mechanism. Multiple solutions results for the synthesis problem. In such cases the deterministic mechanism synthesis results in a design which satisfies the design objective and the constraints. Let the design obtained from the deterministic mechanism synthesis be "Design A". Robust mechanism synthesis can be done for the same synthesis problem considering the uncertainties in the length of the links as random variables. Robust mechanism synthesis not only ensures that the design satisfies the design objective and constraints but also results in a design which has a minimum variation of the mechanism performance due to the uncertainties in the design variables. Figure 4.3 illustrates more about the results obtained from both the methods.

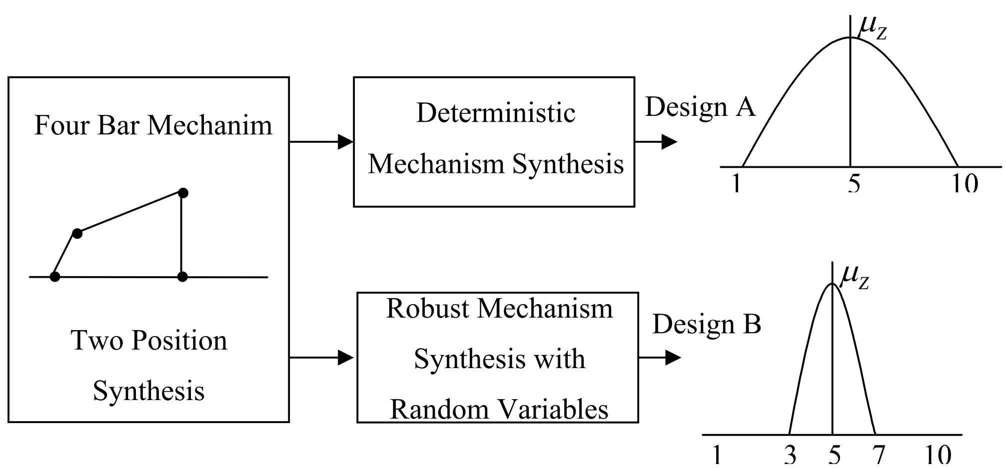

Figure 4.3. Comparison of Deterministic Mechanism Synthesis and Robust Mechanism Synthesis with Random Variables

4.2.2. Robust Mechanism Synthesis with only Interval Variables. When the uncertainties in the design variables are treated as interval variables, interval approach is used for robust mechanism synthesis. The robustness can be quantified by the width of the interval of the mechanism performance (δZ). The objective of the robust mechanism synthesis would be to minimize the error between the desired performance and the actual performance of the mechanism and at the same time, minimizing the affect of uncertainty on the mechanism performance. Mathematically, the design objective for a robust mechanism synthesis can be represented as $f(\boldsymbol{Y}) = w_1\bar{Z} + w_2\delta Z$. Z represents the performance error of mechanism. w_1 and w_2 are the weighting factors. The constraint is modified as $g_i^{\max}(\boldsymbol{Y}) \le 0$ $(i = 1, 2, ..., n_i)$. The optimization model for the robust mechanism synthesis with interval variables can be represented as

$$\begin{cases} \min\limits_{\bar{\boldsymbol{Y}}} \; w_1\bar{Z} + w_2\delta Z \\ \text{s.t.} \quad g_i^{\max}(\boldsymbol{Y}) \le 0, \quad i = 1, 2, ..., n_i \\ \qquad h_j(\boldsymbol{Y}) = 0, \quad j = 1, 2, ..., n_j \\ \qquad \boldsymbol{Y}_k^l \le \boldsymbol{Y}_k \le \boldsymbol{Y}_k^u, \quad k = 1, 2, ..., n \end{cases} \tag{4.4}$$

$\boldsymbol{Y}$ is a vector of interval variables. The flowchart for the optimization model is similar to the one shown in Figure 4.1. To understand more, a mechanism synthesis problem is explained, which is solved using deterministic mechanism synthesis and robust mechanism synthesis using interval variables. Let the uncertainties in the mechanism be interval in nature such as the installation error. Figure 4.4 shows the result obtained from the deterministic mechanism synthesis and robust mechanism synthesis with interval variables. The deterministic mechanism tries to achieve an optimal solution satisfying the design objective and design constraints. The robust mechanism synthesis tries to achieve an optimal solution having minimum δZ and satisfying the design objective and design constraints.

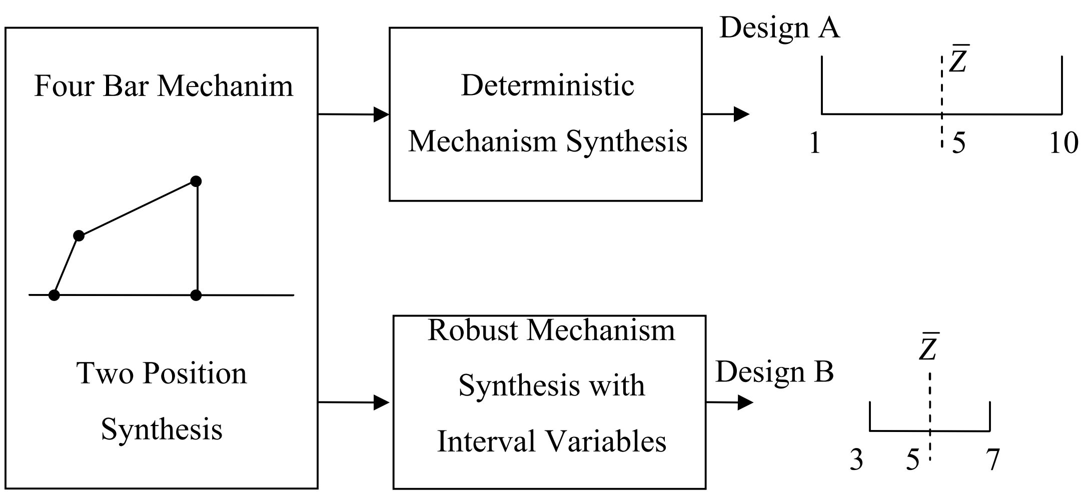

Figure 4.4. Comparison of Deterministic Mechanism Synthesis and Robust Mechanism Synthesis with Interval Variables

4.2.3. Robust Mechanism Synthesis with Random and Interval Variables. In the real world engineering systems, the uncertainties will be in the form of a mixture of random variables and interval variables. In such situations to quantify robustness the proposed method is to use a combined method of probabilistic approach and interval approach. From the previous section, the robustness can be quantified by $\bar{\sigma}_Z$ and $\delta\sigma_Z$. $\bar{\sigma}_Z$ represents the average of the standard deviations and $\delta\sigma_Z$ represents the difference of the standard deviations. Mathematically our design objective for a robust design can be represented as $f\left(\boldsymbol{X},\boldsymbol{Y}\right)=w_1\bar{\mu}_Z+w_2\bar{\sigma}_Z+w_3\delta\sigma_Z$. $\bar{\mu}_Z$ represents the average of mean values of the performance error of the mechanism. Double loop MCS is proposed for evaluating $\bar{\sigma}_Z$ and $\delta\sigma_Z$. w_1, w_2 and w_3 are the weighting factors. The flowchart of the design optimization for the robust mechanism synthesis with random and interval variables is shown in Figure 4.4. The constraint functions need to be changed to maintain robustness of the design feasibility in the worst case of design variables. So, the constraint function is modified as $\mu_{g_i}^{\max}+k\sigma_{g_i}^{\max}\le 0\ \left(i=1,2,....,n_i\right)$. $\mu_{g_i}^{\max}$ and $\sigma_{g_i}^{\max}$ are the maximum of the mean value and the maximum of standard deviation of the constraint function $g_i\left(\boldsymbol{X},\boldsymbol{Y}\right)$, respectively. k is a constant, where $\Phi\left(k\right)$ is the probability of confidence of the constraint satisfaction. The optimization model for the robust mechanism synthesis with random and interval variables can be modeled as

$$\begin{cases} \min\limits_{\boldsymbol{\mu}_x,\,\bar{\boldsymbol{Y}}} \ w_1\bar{\mu}_Z+w_2\bar{\sigma}_Z+w_3\delta\sigma_Z \\ \text{s.t.}\ \ \mu_{g_i}^{\max}+k\sigma_{g_i}^{\max}\le 0, \quad i=1,2,....,n_i \\ \qquad h_j\left(\boldsymbol{\mu}_X,\bar{\boldsymbol{Y}}\right)=0, \qquad j=1,2,....,n_j \end{cases} \tag{4.5}$$

The optimization technique is shown in Figure 4.5. There are two major iterative loops. The first loop is to check the satisfaction of the constraint functions. The second loop is to convergence of the design optimization. Double loop MCS is used in both the loops for evaluating the required terms. Optimization is started with an initial design and checked for the constraint functions. If the constraint functions are not satisfied, the design will be changed. When a design is obtained satisfying the design constraints, design objective is checked for convergence. The process is iterated until a design is obtained satisfying the design constraints and converges at the objective function.

Consider a mechanism synthesis problem such as four bar mechanism. Suppose a two position synthesis is to be done. By using a deterministic mechanism synthesis the optimal result obtained will satisfy the design objective and design constraints. But due to the uncertainties in the design variables the mechanism performance deviated from the expected value. Robust mechanism synthesis can be performed either with random variables or with interval variables. But in reality, the uncertainties are a mixture of random variables and interval variables. For example, the uncertainties in the dimensions due to the manufacturing tolerances can be modeled as a random variable where as the uncertainties in installation errors where there is no information about the probabilistic characteristics can be modeled as an interval variable. In such cases, robust mechanism synthesis with random variables and interval variables results in a more accurate solution. Figure 4.6 shows a comparison of results obtained from the deterministic mechanism synthesis and robust mechanism synthesis. It is evident that the robust mechanism synthesis results in a more robust design compared to the deterministic mechanism synthesis.

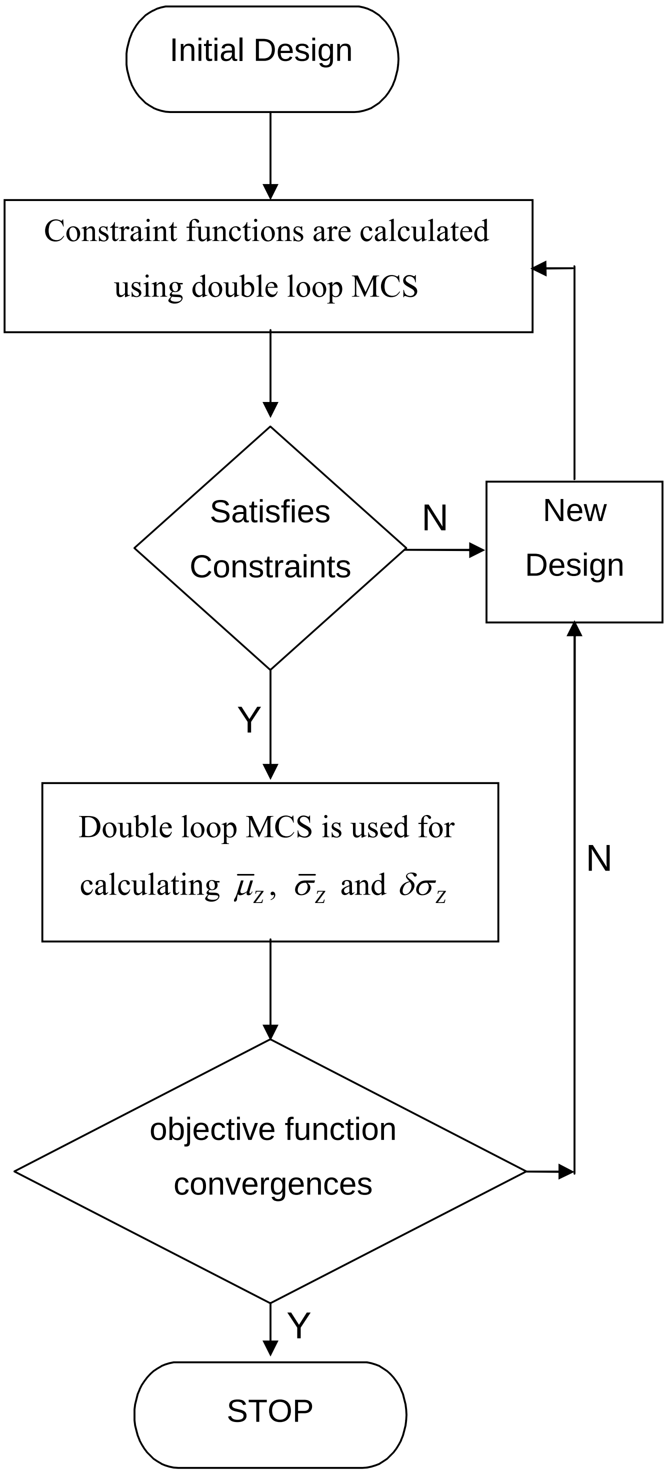

Figure 4.5. Optimization Model for Robust Mechanism Synthesis with Random and Interval Variables

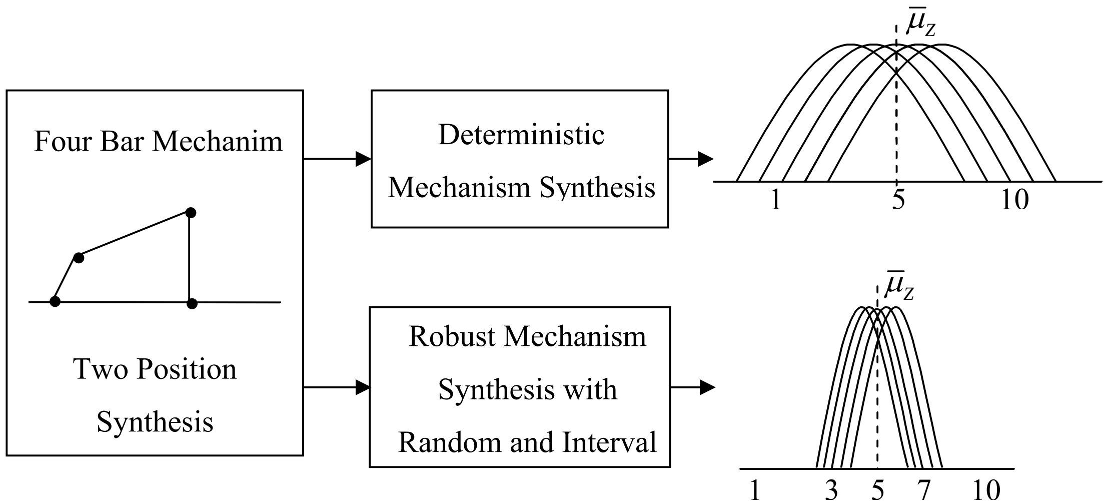

Figure 4.6. Comparison of Deterministic Mechanism Synthesis and Robust Mechanism Synthesis with Random and Interval Variables

In the next section, two examples are presented to validate our proposed method. The first example is a crank slider mechanism and the second example is a four bar mechanism. Crank slider mechanism and four bar mechanism are the typical mechanisms found in most machinery. If the proposed method works for these two mechanisms, it works for almost all the mechanisms.

5. EXAMPLES

In this section, the proposed method is validated and demonstrated with two example problems. The first example is a slider crank mechanism design problem, and the second example is a four bar mechanism design problem.

5.1. EXAMPLE 1 – A SLIDER CRANK MECHANISM DESIGN PROBLEM

A slider crank mechanism as in Figure 5.1 is a fundamental mechanism found in many engineering applications from automotive engines to door-closing mechanisms. The main objective of this example is to design a slider crank mechanism such that for a crank angle (θ) of 10° and 60°, the slider distance (s) should be 3.5″ and 2.5″ respectively. Length of crank (a), length of connecting rod (b) and offset distance (e) are design variables. Links a and b are random variables which are given in Table 5.1.

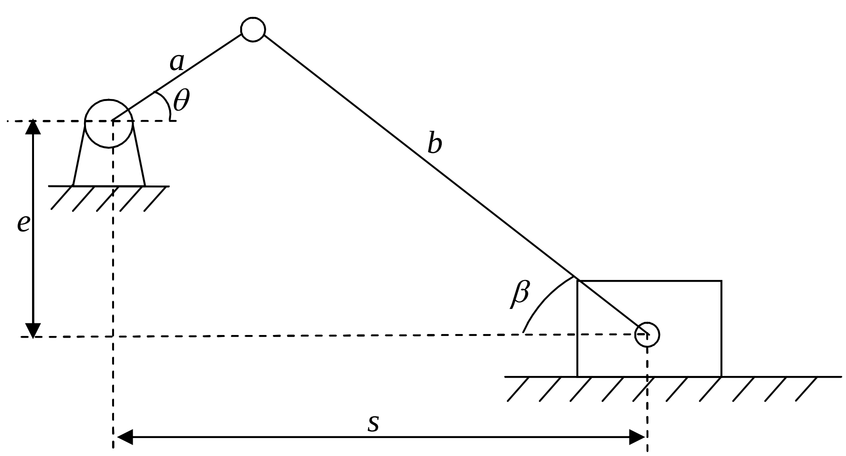

Figure 5.1 Slider Crank Mechanism

Table 5.1. Random Variables

	Variable	Mean (μ)	Standard Deviation (σ)	Distribution
X_1	a	μ_a	1% of μ_a	Normal
X_2	b	μ_b	1% of μ_b	Normal

Because different installation positions of the slider are needed, the offset distance e is specified within a tolerance given in Table 5.2.

Table 5.2. Interval Variable

	Variable	Y_L	Y_U
Y_1	e	e - 5% of $\overline{e}$	e + 5% $\overline{e}$

* $\overline{e}$ is the midpoint of interval e

The distribution of e is not available. Therefore e is treated as an interval variable. The task is to determine the length of the links a and b, and offset distance e

satisfying the objective of the mechanism. First the mechanism synthesis is done deterministically without considering any uncertainties, and then a robust mechanism synthesis is done considering the uncertainties in the design variables. Both the designs are compared.

5.1.1. Deterministic Mechanism Synthesis. Deterministic mechanism synthesis is the common method used for mechanism synthesis. In this method, the nominal values of the design variables are considered without considering any uncertainties.

The slider distance (s) can be calculated by the following equation.

$$f(\boldsymbol{d}) = s = a\cos\theta + \sqrt{b^2 - (e + a\sin\theta)^2} \tag{5.1}$$

The design constraints of this mechanism include the existence of the crank constraint and transmission of energy constraint, which are given by

$$g_1(\boldsymbol{d}) = e - (b - a) \leq 0 \tag{5.2}$$

$$g_2(\boldsymbol{d}) = (e + a) - b\sin 45^\circ \leq 0 \tag{5.3}$$

The deterministic mechanism synthesis can be modeled as

$$\begin{cases} \min\limits_{\boldsymbol{d}} \ \bar{f}(\boldsymbol{d}) = \varepsilon = \sqrt{\varepsilon_{10}^2 + \varepsilon_{60}^2} \\ \text{s.t. } g_1(\boldsymbol{d}) = e - (b - a) \leq 0, \\ \quad g_2(\boldsymbol{d}) = (e + a) - b\sin 45^\circ \leq 0, \\ \quad 0.1 \leq a \leq 20,\ 0.1 \leq b \leq 20 \text{ and } 0.1 \leq e \leq 20 \end{cases} \tag{5.4}$$

$$\varepsilon_{10} = \left(a\cos 10^\circ + \sqrt{b^2 - \left(e + a\sin 10^\circ\right)^2}\right) - 3.5 \tag{5.5}$$

$$\varepsilon_{60} = \left(a\cos 60° + \sqrt{b^2 - \left(e + a\sin 60°\right)^2} \right) - 2.5 \tag{5.6}$$

The objective as given in the equation 5.4 is to minimize the error between the calculated value and desired value. The desired value of slider distance at 10˚ of crank angle is 3.5”. Equation 5.5 shows the calculation of $\varepsilon_{\_10}$, which calculates the error between the calculated value and the desired value at 10˚ of crank angle. Similarly, equation 5.6 shows the error between the calculated value and the desired value at 60˚ of crank angle. The objective is to find a design having the minimum error at both the positions and satisfying the constraint functions. The square root of sum of the squares of the two errors is used as the objective of this design problem. MATLab is used to perform this operation. The optimal solution obtained from MATLab for the deterministic mechanism synthesis is listed in Table 5.3.

Table 5.3. Deterministic Optimal Solution

Number of Iterations	Error (in)	a (in)	b (in)	e (in)	s at 10˚ (in)	s at 60˚ (in)
18	$7.239e^{-12}$	1.133	2.5306	0.65148	3.5	2.5

* s – Slider Distance

ε value obtained from the deterministic optimal solution is $7.239e^{-12}$ which is negligible. The design obtained from the deterministic mechanism synthesis results in the slider distance at 10° and 60° as 3.5″ and 2.5″ respectively and satisfies the design constraints existence of crank and transmission angle. The transmission angle is 45.16°.

5.1.2. Robust Mechanism Synthesis. Robust mechanism synthesis considers the uncertainties in the design variables at the design stage itself. The proposed robust design optimization methodology is applied to the mechanism synthesis problem. Then the robustness of the two designs is compared.

In the proposed design optimization model tolerances in the links and the installation error are considered as uncertainties. In the deterministic design optimization minimizing ε is the objective. Multiple solutions can be obtained for the deterministic optimization model which has the similar value for ε and satisfies the design constraints. Robust mechanism synthesis ensures that the design is satisfactory and also subjected to minimum variations due to the uncertainties. In robust mechanism synthesis the objective is to minimize $\bar{\sigma}_Z$ and $\delta\sigma_Z$. In this design problem, $\bar{\sigma}_Z$ and $\delta\sigma_Z$ needs to be minimized at crank angles of 10° and 60°. The two inequality constraints, existence of crank and transmission of energy are maintained at the worst case of interval variables. Two equality constraints are added to the design constraints. The first equality constraint is maintaining the slider distance as 3.5″ at 10° of crank angle and the other equality constraint is to maintain the slider distance as 2.5″ at 60° of crank angle.

A double loop MCS is used in the robust design optimization model and $\bar{\sigma}_Z$ and $\delta\sigma_Z$ values are calculated. In the double loop MCS, 20 intervals (N_i) for the interval variable and 2000 samples (N) for the random variables are taken. As there is only one interval variable e, 20 σ_Z values can be obtained from which $\bar{\sigma}_Z$ and $\delta\sigma_Z$ are calculated.

The objective is to minimize the effect of random variables and interval variables on the output slider distance at both the positions of the crank.

The robust mechanism synthesis can be modeled as

$$\begin{cases} \min \bar{f}(\boldsymbol{d},\boldsymbol{\mu}_X,\bar{\boldsymbol{Y}}) = w_1 \sum_{i=1}^{2} \bar{\sigma}_{Z_{\theta_i}} \Big/ \bar{\sigma}^*_{Z_{\theta_i}} + w_2 \sum_{i=1}^{2} \delta\sigma_{Z_{\theta_i}} \Big/ \delta\sigma^*_{Z_{\theta_i}}, \theta_1 = 10°, \theta_2 = 60° \\ \text{s.t.} \quad \mu_{g_i}^{\max} + k\sigma_{g_i}^{\max} \le 0,\ i=1,2 \\ \qquad g_1(d,\mu_X,\bar{Y}) = e-(b-a), \\ \qquad g_2(d,\mu_X,\bar{Y}) = (e+a) - b\sin 45°, \\ \qquad h_1\left(d,\mu_X,\bar{Y}\right) = a\cos 10° + \sqrt{b^2 - \left(e + a\sin 10°\right)^2} - 3.5 \\ \qquad h_2\left(d,\mu_X,\bar{Y}\right) = a\cos 60° + \sqrt{b^2 - \left(e + a\sin 60°\right)^2} - 2.5 \\ \qquad 0.1 \le \mu_a \le 20,\ 0.1 \le \mu_b \le 20 \text{ and } 0.1 \le \bar{e} \le 20 \end{cases} \tag{5.7}$$

where $\bar{\sigma}^*_{Z_\theta}$ and $\delta\sigma^*_{Z_\theta}$ are the best achievable optimal solution of $\bar{\sigma}_{Z_\theta}$ and $\delta\sigma_{Z_\theta}$, respectively. Weighting factor method is used to formulate the multiple objective function. w_1 and w_2 are the weighting factors used for the purpose of illustration. w_1 and w_2 are taken as 0.5.

The robust mechanism synthesis solution which is obtained from MATLab is listed in Table 5.4.

Table 5.4. Robust Mechanism Synthesis Solution

Number of Iterations	a (in)	b (in)	e (in)	μ_s at 10° (in)	μ_s at 60° (in)
4	1.3239	2.2209	0.1	3.5	2.5

The design variables which are obtained from the robust design optimization model are different from those obtained from deterministic optimization. The main objective of the mechanism which is to maintain the slider distance as 3.5 in and 2.5 in at crank angles of 10° and 60° respectively is fulfilled in both the designs. As can be seen next, the new design variables produce a more robust design.

5.1.3. Robustness Assessment. Robustness assessment is performed on the two designs. It is shown that the proposed robust mechanism synthesis method results in a robust design. For quantifying the robustness of deterministic mechanism synthesis model mean values of a and b are taken as 1.133″ and 2.5306″, respectively, and midpoint of interval variable e is taken as 0.65148″. For quantifying the robustness of robust mechanism synthesis model, the mean values of a and b are taken as 1.3239″ and 2.2209″, respectively, and midpoint of interval variable e is taken as 0.1″. The number of intervals (N_i) for the interval variable is taken as 20. The Number of samples (N) is taken as 2000. In this design problem, the robustness should be assessed at two positions of the crank angle. The solution obtained from the double loop MCS are shown in Table 5.5.

Table 5.5 Robustness Assessment of Deterministic Mechanism Synthesis and Robust Mechanism Synthesis

	Deterministic Mechanism Synthesis	Robust Mechanism Synthesis
μ_a	1.1330″	1.3239″
μ_b	2.5306″	2.2209″
$\bar{e}$	0.6515″	0.1″
$s_{10°}$	3.5″	3.5″
$s_{60°}$	2.5″	2.5″
$\bar{\sigma}_{Z_{10°}}$	0.02941″	0.02795″
$\delta\sigma_{Z_{10°}}$	2.231e^{-4} in	4.089e^{-5} in
$\bar{\sigma}_{Z_{60°}}$	0.03387″	0.0306″
$\delta\sigma_{Z_{60°}}$	9.6732 e^{-4} in	3.631e^{-4} in

Both the designs fulfilled the primary objective. The slider distance at 10˚ and 60˚ is 3.5″ and 2.5″, respectively. The robustness for both the designs is assessed by looking at the values of $\bar{\sigma}_{Z_{10°}}$, $\delta\sigma_{Z_{10°}}$, $\bar{\sigma}_{Z_{60°}}$ and $\delta\sigma_{Z_{60°}}$. $\bar{\sigma}_{Z_{10°}}$, $\delta\sigma_{Z_{10°}}$, $\bar{\sigma}_{Z_{60°}}$ and $\delta\sigma_{Z_{60°}}$ obtained from the robust mechanism synthesis are less than those from the deterministic mechanism synthesis design. This explains that the variation in the response variable due to the uncertainty in the inputs is less for robust mechanism synthesis design. Using a robust mechanism synthesis technique a robust design for the given synthesis problem is achieved.

To understand more, the output obtained from MATLab is graphically represented. The graphs obtained from both the methods are compared. The family of distribution curves at 10° of crank angle is shown in Figure 5.2 and Figure 5.3. As learned from the previous chapters, a minimum value for $\bar{\sigma}_Z$ and $\delta\sigma_Z$ for a robust design is to be achieved. A narrow distribution curves will have a low $\bar{\sigma}_Z$ value. $\delta\sigma_Z$ will be low if the width of the band is narrow. The design obtained from the robust mechanism synthesis has narrow distribution curves and also a narrow width of the band compared to the design obtained from deterministic mechanism synthesis. Similarly, Figure 5.4 and Figure 5.5 show the family of distribution curves at 60° of crank angle. From the distribution curves, it is evident that the design obtained from robust mechanism synthesis is more robust compared to the design obtained from the deterministic mechanism synthesis.

From the results obtained, it is evident that the robust mechanism synthesis results in a more robust design compared to the deterministic mechanism synthesis. The proposed method of considering a mixture of random variables and interval variables results in a more accurate representation of the uncertainties in the design variables.

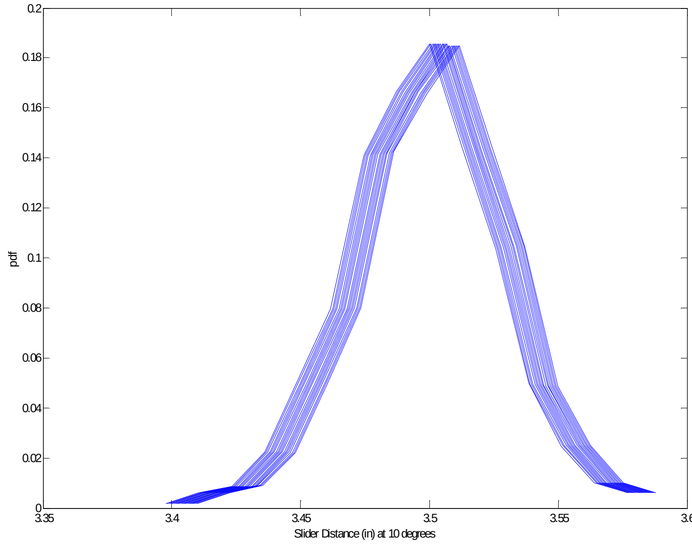

Figure 5.2. Family of Distributions at the Crank Angle of 10˚ for the Design Achieved from Deterministic Mechanism Synthesis

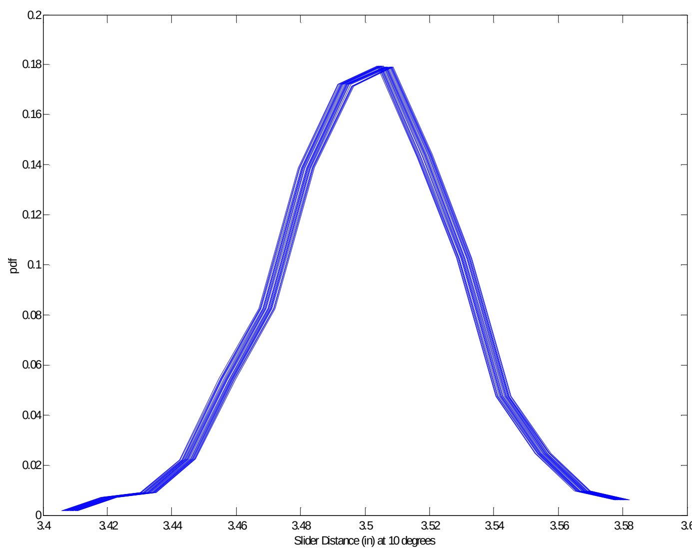

Figure 5.3. Family of Distributions at the Crank Angle of 10˚ for the Design Achieved from Robust Mechanism Synthesis

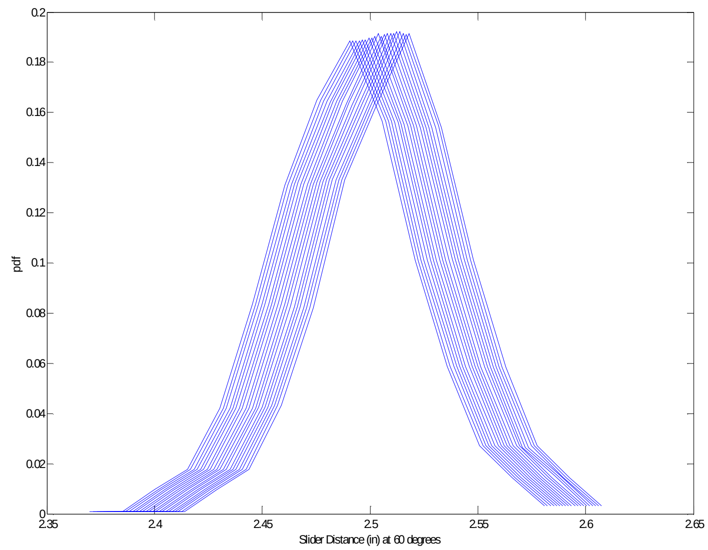

Figure 5.4. Family of Distributions at the Crank Angle of 60˚ for the Design Achieved from Deterministic Mechanism Synthesis

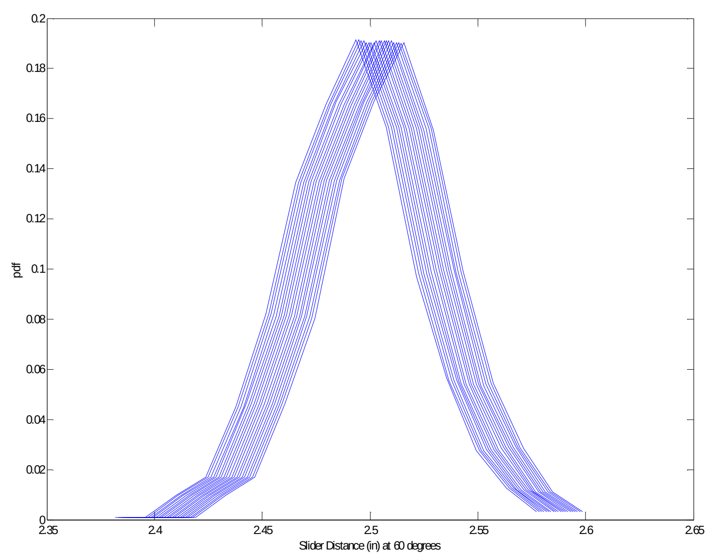

Figure 5.5. Family of Distributions at the Crank Angle of 60˚ for the Design Achieved from Robust Mechanism Synthesis

5.2. EXAMPLE 2 – A FOUR BAR MECHANISM DESIGN PROBLEM

A four bar mechanism as shown in Figure 5.6 is to be designed such that when the angle (θ_2) of the input link is 10° and 45°, the position of P(X, Y) should be (3.8, 3) and (3, 5), respectively.

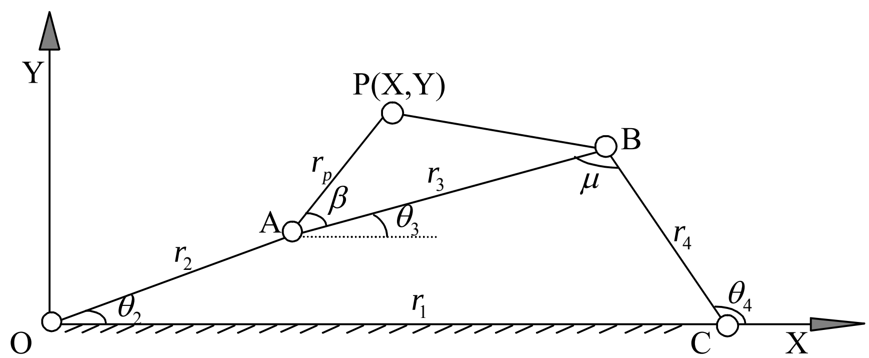

Figure 5.6. Four Bar Mechanism

Length of ground link OC(r_1), length of link OA(r_2), length of link AB(r_3), length of link BC(r_4), length of link AP(r_p) and angle BAP (β) are design variables. The links r_2, r_3, r_4 and r_p are random variables which are given in Table 5.6.

Table 5.6. Random Variables

	Variable	Mean (μ)	Standard Deviation (σ)	Distribution
X_1	r_2	μ_{r_2} mm	0.05 mm	Normal

Table 5.6. Random Variables (Cont.)

X_2	r_3	μ_{r_3} mm	0.10 mm	Normal
X_3	r_4	μ_{r_4} mm	0.05 mm	Normal
X_4	r_p	μ_{r_p} mm	0.05 mm	Normal

As there is no information available on the type of distribution of the variables r_1 and β, they are considered as interval variables which is given in Table 5.7.

Table 5.7. Interval Variables

	Variable	Y_L	Y_U
Y1	r_1	$(r_1 - 0.5)$ mm	$(r_1 + 0.5)$ mm
Y2	β	$(\beta - 1)°$	$(\beta + 1)°$

5.2.1. Deterministic Mechanism Synthesis. Deterministic mechanism synthesis results in obtaining the values of the design variables satisfying the design objective. In deterministic mechanism synthesis uncertainties in the design variables are not considered. The governing equations for finding the position of $P(X,Y)$ are given below:

$$\begin{aligned} P_X &= r_2 \cos\theta_2 + r_p \cos(\beta + \theta_3), \\ P_Y &= r_2 \sin\theta_2 + r_p \sin(\beta + \theta_3) \end{aligned} \tag{5.9}$$

$$\begin{aligned} r_2 \cos\theta_2 + r_3 \cos\theta_3 - r_4 \cos\theta_4 - r_1 &= 0, \\ r_2 \sin\theta_2 + r_3 \sin\theta_3 - r_4 \sin\theta_4 &= 0 \end{aligned} \tag{5.10}$$

The constraints of this mechanism design include the existence of the crank and transmission angle constraint, which are given by,

$$r_2 + r_3 - r_1 - r_4 < 0 \tag{5.11}$$

$$\left|90° - \mu\right|_{\max} - 50° < 0 \tag{5.12}$$

$$\begin{aligned} \mu_1 &= \cos^{-1}\left[\frac{r_3^{\,2} + r_4^{\,2} - \left(r_1 + r_2\right)^2}{2r_3r_4}\right], \\ \mu_2 &= \cos^{-1}\left[\frac{r_3^{\,2} + r_4^{\,2} - \left(r_1 - r_2\right)^2}{2r_3r_4}\right] \end{aligned} \tag{5.13}$$

where μ_1 and μ_2 are maximum and minimum transmission angle, respectively.

The objective is to find the design variables r_1, r_2, r_3, r_4, r_p and β such that, the error between the actual position and desired position of $P\left(X,Y\right)$ at 10° and 45° of crank angle is minimum. The design should also satisfy the design constraints which are given in equation 5.11 and equation 5.12.

The deterministic mechanism synthesis is modeled as

$$\begin{cases} \min\limits_{\boldsymbol{d}} \ \overline{f}(\boldsymbol{d}) = \varepsilon = \sqrt{\left(Px_{10°} - 3.8\right)^2 + \left(Py_{10°} - 3\right)^2 + \left(Px_{45°} - 3\right)^2 + \left(Py_{45°} - 5\right)^2} & (5.14) \\ \text{s.t. } g_1(d) = r_1 + r_2 - r_3 - r_4 \le 0, & \\ \quad g_2(d) = \left|90° - \mu\right|_{\max} - 50 \le 0, & \\ \quad 1 \le r_1 \le 50;\ 1 \le r_2 \le 50;\ 1 \le r_3 \le 50; & \\ \quad 1 \le r_4 \le 50;\ 1 \le r_p \le 50;\ 10 \le \beta \le 70 & \end{cases}$$

The first inequality constraint is Grashof's law for a crank rocker mechanism. The second inequality constraint is a transmission angle constraint of the mechanism. The optimal solution from the deterministic mechanism synthesis is listed in Table 5.8.

Table 5.8. Deterministic Optimal Solution

Variable	Solution	Variable	Solution
r_1	19.5 mm	r_4	9.5831 mm
r_2	3.7887 mm	r_p	2.3431 mm
r_3	19.8555 mm	β	62.3385°
(X_1, Y_1)	(3.8, 3) mm	(X_2, Y_2)	(3, 5) mm

5.2.2. Robust Mechanism Synthesis. In the proposed robust mechanism synthesis model, the tolerances in the links and the installation error are considered as the uncertainties in the design variables. Due to the uncertainties the mechanism performance, in this problem it is the position of $P(X,Y)$, deviates from the designed value. Robust mechanism synthesis ensures that the mechanism performance has a minimum effect due to the uncertainties in the design variables. The objective will be not only to maintain the position of $P(X,Y)$ to the desired value but also to optimize the average standard deviation and difference between the maximum and minimum standard deviations of the output variable. The robust mechanism synthesis can be modeled as

$$\begin{cases} \min \bar{f}(\boldsymbol{d},\boldsymbol{\mu}_X,\bar{\boldsymbol{Y}}) = \begin{array}{l} w_1\varepsilon + w_2\sum_{i=1}^{2}\bar{\sigma}_{X_{\theta_i}}\Big/\bar{\sigma}^*_{X_{\theta i}} + w_3\sum_{i=1}^{2}\bar{\sigma}_{Y_{\theta_i}}\Big/\bar{\sigma}^*_{Y_{\theta i}} + \\ w_4\sum_{i=1}^{2}\delta\sigma_{X_{\theta i}}\Big/\delta\sigma^*_{X_{\theta_i}} + w_5\sum_{i=1}^{2}\delta\sigma_{Y_{\theta_i}}\Big/\delta\sigma^*_{Y_{\theta i}}, \qquad \theta_1 = 10°, \theta_2 = 45° \end{array} \\ \text{s.t.} \quad \mu_{g_i}^{\max} + k\sigma_{g_i}^{\max} \le 0,\ i=1,2 \\ \qquad g_1(d,\mu_X,\bar{Y}) = r_1 + r_2 - r_3 - r_4 < 0 \\ \qquad g_2(d) = \left|90° - \mu\right|_{\max} - 50 < 0 \\ \qquad 1 \le \bar{r}_1 \le 50\,;\ 1 \le \mu_{r_2} \le 50\,;\ 1 \le \mu_{r_3} \le 50\,; \\ \qquad 1 \le \mu_{r_4} \le 50\,;\ 1 \le \mu_{r_p} \le 50\,;\ 10 \le \bar{\beta} \le 70 \end{cases} \tag{5.9}$$

where $\varepsilon = \sqrt{\left(Px_{10°} - 3.8\right)^2 + \left(Py_{10°} - 3\right)^2 + \left(Px_{45°} - 3\right)^2 + \left(Py_{45°} - 5\right)^2}$,

$k = 3$,

$w_1 = w_2 = w_3 = w_4 = w_5 = 0.2$

The constraint is modified to maintain the robustness of the design feasibility at the worst case of the design variables. In the double loop MCS, 5 intervals are taken for each of the interval variables and 1000 samples are taken for the random variables. The robustness is to be achieved at the two positions of the coupler and at each position the robustness is considered at X and Y coordinates.

The optimal solution obtained from the robust mechanism synthesis is listed in Table 5.9.

Table 5.9. Optimal Solution Obtained from Robust Mechanism Synthesis

Variable	Solution	Variable	Solution
r_1	35.6425 mm	r_4	32.9116 mm
r_2	3.7886 mm	r_p	2.3431 mm
r_3	20.6294 mm	β	15.4131°
(X_1, Y_1)	(3.8, 3) mm	(X_2, Y_2)	(3, 5) mm

5.2.3. Robustness Assessment. The uncertainty in the design variables are in the form of a mixture of random variables and interval variables. In such cases, a double loop MCS can be used for assessing the robustness of the system. The mean values of random variables and the midpoint of the interval variables are taken as the nominal values of the design variables obtained from the mechanism synthesis. The robustness is assessed for the designs obtained from the deterministic mechanism synthesis and robust mechanism synthesis and the results are compared. The performance function for the double loop MCS is taken as the performance error which is given by,

$$g(X,Y) = \varepsilon = \sqrt{\left(Px_{10^\circ} - 3.8\right)^2 + \left(Py_{10^\circ} - 3\right)^2 + \left(Px_{45^\circ} - 3\right)^2 + \left(Py_{45^\circ} - 5\right)^2} \tag{5.10}$$

Five intervals for each of the interval variables and 1000 samples for the random variables are taken. The comparison of the robustness of the designs obtained from the deterministic mechanism synthesis and robust mechanism synthesis is listed in Table 5.10.

Table 5.10. Comparison of Designs Obtained from Deterministic Mechanism Synthesis and Robust Mechanism Synthesis

Variable	Deterministic Optimal Solution	Robust Design Optimal Solution
r_1	19.5 mm	35.6425 mm
r_2	3.7887 mm	3.7886 mm
r_3	19.8555 mm	20.6294 mm
r_4	9.5831 mm	32.9116 mm
r_p	2.3431 mm	2.3431 mm
β	62.3385°	15.4131°
(X_1, Y_1)	(3.8, 3) mm	(3.8, 3) mm
(X_2, Y_2)	(3, 5) mm	(3, 5) mm
$(\bar{\sigma}_{X1}, \bar{\sigma}_{Y1})$	(49.92 e^{-3}, 50.2 e^{-3}) mm	(42.84 e^{-3}, 50.2 e^{-3}) mm
$(\delta\sigma_{X1}, \delta\sigma_{Y1})$	(14.697 e^{-4}, 11.6 e^{-5}) mm	(1.75 e^{-4}, 7.73 e^{-5}) mm
$(\bar{\sigma}_{X2}, \bar{\sigma}_{Y2})$	(40.0 e^{-3}, 59.2 e^{-3}) mm	(34.05 e^{-3}, 59.6 e^{-3}) mm
$(\delta\sigma_{X2}, \delta\sigma_{Y2})$	(13.8 e^{-4}, 44.2 e^{-5}) mm	(6.48 e^{-4}, 37.75 e^{-5}) mm

The results clearly show that the design obtained from the robust mechanism synthesis is more robust compared to the deterministic mechanism synthesis. Both the mechanisms (Crank Slider Mechanism and Four Bar Mechanism) resulted in best solutions using a robust mechanism synthesis approach. It is evident that the proposed method of robust mechanism synthesis results a robust mechanism.

6. CONCLUSIONS

Mechanism synthesis is a systematic design of a mechanism to achieve a specific task. Generally deterministic values of the design variables are considered when designing a mechanism. But in reality, uncertainty exists in design variables and other parameters. Uncertainty is classified as aleatory uncertainty and epistemic uncertainty. Aleatory uncertainty is the inherent variation associated with the physical system or the environment. Epistemic uncertainty is due to lack of knowledge or insufficient data. Due to the two types of uncertainty the mechanism performance deviates from the designed values. To minimize the variation in the mechanism performance, either the uncertainties in the design variables and parameters can be reduced, or the effect of uncertainty on the mechanism performance can be controlled by changing the nominal design. The former method is very expensive and some times cannot be achieved.

Robust design uses the later method and ensures that the product perform its intended function regardless of variations. To perform robust design, first robustness of a system needs to be quantified. Currently, there are two approaches, probabilistic approach and interval approach, for quantifying robustness of a system. Probabilistic approach treats uncertain variables as random variables and quantifies robustness by the standard deviation of the performance. Many approximation methods are available for evaluating standard deviation. MCS is used because it is simple and results in accurate estimations. Interval approach treats uncertain variables as intervals. Robustness is quantified by the width of the interval of the performance.

In reality, the uncertainty in the engineering systems may exhibit both random and interval nature. In such situations, to apply robust design methodologies, first the robustness of the system needs to be assessed. The output of the performance will be in the form of family of distributions. Before quantifying the robustness, first we examine why the output behaves as a family of distribution curves. The distribution curves are due to the effect of randomness in the random variables and the intervals are due to the effect of interval nature of the interval variables. $\bar{\mu}_Z$, $\bar{\sigma}_Z$ and $\delta\sigma_Z$ are used to quantify the robustness of the system. $\bar{\mu}_Z$, $\bar{\sigma}_Z$ and $\delta\sigma_Z$ represents the average of the mean values, average of standard deviations and difference between the maximum and minimum standard deviations respectively. In this work, a double loop MCS is used for evaluating $\bar{\mu}_Z$, $\bar{\sigma}_Z$ and $\delta\sigma_Z$. In the outer loop, the interval combinations are generated according to the number of intervals. In the inner loop, the samples of random variables are generated according to their distributions. For each sample the output performance is calculated. So, a set of samples for the output performance is obtained. The mean and standard deviation of a distribution curve can be calculated at each interval combination. From the obtained set of mean and standard deviation values $\bar{\mu}_Z$, $\bar{\sigma}_Z$ and $\delta\sigma_Z$ can be evaluated.

After knowing how to quantify robustness, robust design is performed. First, the existing design optimization techniques are studied and then the proposed method is applied to the robust design. There are three main parts in a design optimization, the design variables, the objective function and the constraint function. For any engineering problem, first the design variables should be identified. Then the objective function and the design constraints, which are dependent on the design variables should be mathematically modeled.

In a mechanism synthesis problem, our objective is to achieve a specific task such as motion generation, path generation and function generation of a mechanism. The design variables would be the dimensions of links, offset distance and so on. The design constraints would be existence of the mechanism and transmission angle of the mechanism. Traditional mechanism synthesis considers the nominal values of the design variables without considering any uncertainty.

In reality there will be various uncertainties in the mechanism such as manufacturing tolerances in the links, clearances in the joints in the links and installation errors. In the robust mechanism synthesis, the uncertainties in the design variables are considered and the objective will be not only to achieve the specific task of a mechanism but also to minimize the variations in the mechanism due to the uncertainties. Formerly, the uncertainties are treated as either random variables or interval variables. But in reality both the random variables and interval variables exist for the same design problem. If the uncertainties are treated as random variables, probability distributions are assumed to the variables where there is no information available. In such cases, probabilistic approach is used in the robust mechanism synthesis. The objective will be minimization of error between the desired performance and the actual performance of the mechanism plus the standard deviation of the output performance. When uncertainties are treated as interval variables, interval approach is used in the robust mechanism synthesis. The objective will be minimization of error between the desired performance and the actual performance of the mechanism plus the width of the interval of the output performance. When a mixture of random and interval variables exist both the probabilistic and the interval approaches may lead to misleading results. In this work, the design variables which have information about the probability distribution are considered as random variables and the design variables with no information except the range are considered as interval variables. Both the probabilistic approach and the interval approach are combined to perform robust mechanism synthesis. The robustness of such a system can be quantified by $\bar{\mu}_Z$, $\bar{\sigma}_Z$ and $\delta\sigma_Z$. The objective will be minimizing $\bar{\mu}_Z$, $\bar{\sigma}_Z$ and $\delta\sigma_Z$. The robustness of the design feasibility is maintained in the worst case of design variables. This methodology results in better representation of the uncertainty and a robust design for a design problem.

Double loop MCS is used for quantifying the robustness of the performance function when the uncertainty in the design variables is a mixture of random variables and interval variables. When the number of simulations increases double loop MCS demands more computational time. As a future work of the proposed methodology, any method which results $\bar{\mu}_Z$, $\bar{\sigma}_Z$ and $\delta\sigma_Z$ with the same accuracy as double loop MCS and takes less computational time can be used. DOE [44] concepts can be used to identify the design variables which prominently affect the output performance. Then the uncertainty to those variables can be considered instead of considering uncertainty for every design variable.

APPENDIX A.

MATLAB PROGRAM FOR EXAMPLE 1

Deterministic Mechanism Synthesis

```
%Deterministic Mechanism Synthesis of Crank Slider Mechanism
%MAIN PROGRAM
clc; warning off; close all; clear all;
format long;
d0=[4,8,1];  % starting point for a, b and e
lb=[0.1,0.1,0.1];  %lower bounds for design variables a, b and e
ub=[20,20,20]; %upper bounds for design variables a, b and e
option = optimset('display','iter'); %set options to show the optimization history
d=fmincon('det_obj_fun',d0,[],[],[],[],lb,ub,'det_constr_fun',option); % call the optimizer

% analysis at the optimal point
X1=d(1);  %a
X2=d(2);  %b
X3=d(3);  %e
S_10=( (X1*cos(10*pi/180)) + ((X2^2) - (X3+X1*sin(10*pi/180))^2)^0.5  );
S_60=( (X1*cos(60*pi/180)) + ((X2^2) - (X3+X1*sin(60*pi/180))^2)^0.5  );
Error1 = ( ( (X1*cos(60*pi/180)) + ((X2^2) - (X3+X1*sin(60*pi/180))^2)^0.5  ) - 2.5 );
Error2 =  ( ( (X1*cos(10*pi/180)) + ((X2^2) -(X3+X1*sin(10*pi/180))^2)^0.5  ) - 3.5 );
Error= Error1^2 + Error2^2;
transmission_angle = acos((X3+X1)/X2)*180/pi;
disp(['s_10 = ', num2str(S_10)]);
disp(['s_60 = ', num2str(S_60)]);
disp(['transmission_angle = ', num2str(transmission_angle)]);
obj = (Error)^.5
c = det_constr_fun(d); %calculate the constraint functions
disp(['the optimal point = ', num2str(d)]);
disp(['the objective funtion = ', num2str(obj)]);
disp(['the constraint functions = ', num2str(c)]);
disp(X1);disp(X2);disp(X3);
%CONSTRAINT FUNCTION
function [c,ceq] = det_constr_fun(d) %constraint function
X1 = d(1);  %a
X2 = d(2);  %b
X3 = d(3);  %e
c(1) = X3 - (X2 - X1);  %Existence of Crank Constraint
```

```
c(2) = -(sin(45*pi/180)*X2) + (X3+X1);  %Transmission Angle Constraint

Error1 = ( ( (X1*cos(60*pi/180)) + ((X2^2) - (X3+X1*sin(60*pi/180))^2)^0.5  ) - 2.5 );
Error2 =  ( ( (X1*cos(10*pi/180)) + ((X2^2) -(X3+X1*sin(10*pi/180))^2)^0.5  ) - 3.5 );

ceq(1) = Error1;
ceq(2) = Error2;

----------------------------------------------------------------------------------------------------------
%OBJECTIVE FUNCTION

function obj = det_obj_fun(d) %objective function

X1=d(1);  %a
X2=d(2);  %b
X3=d(3);  %e

Error1 = ( ( (X1*cos(60*pi/180)) + ((X2^2) - (X3+X1*sin(60*pi/180))^2)^0.5  ) - 2.5 );
Error2 =  ( ( (X1*cos(10*pi/180)) + ((X2^2) -(X3+X1*sin(10*pi/180))^2)^0.5  ) - 3.5 );
Error= Error1^2 + Error2^2;
obj = (Error)^.5; %Objective Square Root of Sum of Squares of the Error
----------------------------------------------------------------------------------------------------------
```

Robust Mechanism Synthesis

```
%Robust Mechanism Synthesis of Crank Slider Mechanism
%MAIN PROGRAM
clc; close all; clear all; warning off;
format long;
d0=[4,8,1];  % starting point of a, b and e
lb=[0.1,0.1,0.1];  %lower bounds for design variables
ub=[20,20,20]; %upper bounds for design variables
option = optimset('display','iter'); %set options to show the optimization history
d=fmincon('std_obj_fun',d0,[],[],[],[],lb,ub,'std_constr_fun',option); % call the optimizer

% analysis at the optimal point
X1=d(1); %length of crank
X2=d(2); %length of connecting rod
X3=d(3); %offset distance
N=2000;  %Number of Samples of Random Variables
Nu=20;    %Number of Intervals
MuX1=X1;  stdX1 = MuX1/100;
MuX2=X2;  stdX2 = MuX2/100;
MuX3=X3;  aX3=MuX3-(MuX3/20); bX3=MuX3+(MuX3/20);
max_std_10_norm = 0.029516808208; max_std_60_norm = 0.03386878670962;
sdiff_10_norm = 2.231075060847541e-004; sdiff_60_norm = 9.673179655116104e-004;

S_10 =(MuX1*cos(10*pi/180)) + ((MuX2^2) - (MuX3+MuX1*sin(10*pi/180))^2)^0.5
S_60 =(MuX1*cos(60*pi/180)) + ((MuX2^2) - (MuX3+MuX1*sin(60*pi/180))^2)^0.5
% Step 1 - Sampling on random variables
randn('state',0)    % Initialize the normal random variable generator
X1_sample =normrnd(MuX1,stdX1,N,1); %sample of X1
X2_sample =normrnd(MuX2,stdX2,N,1); %sample of X2
X3_sample = aX3:(bX3-aX3)/(Nu-1):bX3; %Intervals of X3

% Step 2 - Experimentation
for i=1:Nu
   for j=1:N
         actual_S_10(i,j) = (X1_sample(j,1)*cos(10*pi/180)) + ((X2_sample(j,1)^2)…
…- (X3_sample(1,i)+ X1_sample(j,1)*sin(10*pi/180))^2)^0.5;
```

```
            actual_S_60(i,j) = (X1_sample(j,1)*cos(60*pi/180)) + ((X2_sample(j,1)^2)…
…- (X3_sample(1,i)+ X1_sample(j,1)*sin(60*pi/180))^2)^0.5;
            A_10 (i,j) =10; A_60 (i,j) =60;
    end
end
for i=1:Nu
    std_actual_S_10 (1,i) = std(actual_S_10(i,:));
    std_actual_S_60 (1,i) = std(actual_S_60(i,:));
end
max_std_10 = max(std_actual_S_10)
max_std_60 = max(std_actual_S_60)
sdiff_10 = max_std_10 - min(std_actual_S_10)
sdiff_60 = max_std_60 - min(std_actual_S_60)

obj = (max_std_10/max_std_10_norm) + (max_std_60/max_std_60_norm)…
…+ (sdiff_10/sdiff_10_norm) + (sdiff_60/sdiff_60_norm);
c = std_constr_fun(d); %calculate the constraint functions
ceq = std_constr_fun(d);
disp(['the optimal point = ', num2str(d)]);
disp(['the objective funtion = ', num2str(obj)]);
disp(['the constraint functions = ', num2str(c)]);
disp(['the equality constraint functions = ', num2str(ceq)]);



n_point = 15;
for j=1:Nu
    step_10 = (max(actual_S_10(j,:))-min(actual_S_10(j,:))) / n_point;
    step_60 = (max(actual_S_60(j,:))-min(actual_S_60(j,:))) / n_point;
    for i = 1:n_point
        S_point_10(j,i) = min(actual_S_10(j,:)) + (i-1)*step_10;
        S_point_60(j,i) = min(actual_S_60(j,:)) + (i-1)*step_60;
    end
end
for j=1:Nu
    m_10(j,:) = hist (actual_S_10(j,:),S_point_10(j,:));
    pdf_10(j,:) = m_10(j,:)/N;
```

```
    m_60(j,:) = hist (actual_S_60(j,:),S_point_60(j,:));
    pdf_60(j,:) = m_60(j,:)/N;
end
figure;
for j=1:Nu
    plot(S_point_10(j,:),pdf_10(j,:));
    xlabel('Slider Distance (in) at 10 degrees'); ylabel('pdf');
    hold on;
end
figure;
for j=1:Nu
    plot(S_point_60(j,:),pdf_60(j,:));
    xlabel('Slider Distance (in) at 60 degrees'); ylabel('pdf');
    hold on;
end
----------------------------------------------------------------------------------------------------------
```

```
%CONSTRAINT FUNCTION

function [c,ceq] = std_constr_fun(d) %constraint function
X1 = d(1);  %a
X2 = d(2);  %b
X3 = d(3);  %e
N=2000;     %Number of Samples of Random Variables
Nu=20;       %Number of Intervals
MuX1=X1;  stdX1 = MuX1/100;
MuX2=X2;  stdX2 = MuX2/100;
MuX3=X3;  aX3=MuX3-(MuX3/20); bX3=MuX3+(MuX3/20);

% Step 1 - Sampling on random variables
randn('state',0)    % Initialize the normal random variable generator
X1_sample =normrnd(MuX1,stdX1,N,1); %sample of X1
X2_sample =normrnd(MuX2,stdX2,N,1); %sample of X2
X3_sample = aX3:(bX3-aX3)/(Nu-1):bX3; %Intervals of X3
```

```
% Step 2 - Experimentation
g1_mean = MuX3 - (MuX2 - MuX1);
g2_mean = -(sin(45*pi/180)*MuX2) + (MuX3+MuX1);
for i=1:Nu
   for j=1:N
          actual_g1(i,j) = X3_sample(1,i) - (X2_sample(j,1) - X1_sample(j,1));
          actual_g2(i,j) = -(sin(45*pi/180)*X2_sample(j,1)) + (X3_sample(1,i) + X1_sample(j,1));
   end
end
for i=1:Nu
   std_actual_g1(1,i) = std(actual_g1(i,:));
   std_actual_g2(1,i) = std(actual_g2(i,:));
end
max_std_g1 = max(std_actual_g1);
max_std_g2 = max(std_actual_g2);
max_mean_g1 = max(mean(actual_g1));
max_mean_g2 = max(mean(actual_g2));
k=3;

c(1) = max_mean_g1 + k*max_std_g1; %Existance of Crank Constraint
c(2) = max_mean_g2 + k*max_std_g2;  %Transmission Angle Constraint
ceq(1) = ((X1*cos(10*pi/180)) + ((X2^2) - (X3+X1*sin(10*pi/180))^2)^0.5)-3.5;
ceq(2) = ((X1*cos(60*pi/180)) + ((X2^2) - (X3+X1*sin(60*pi/180))^2)^0.5)-2.5;
------------------------------------------------------------------------------------------------------------

%OBJECTIVE FUNCTION
function obj = std_obj_fun(d)  %objective function

X1=d(1); %length of crank
X2=d(2); %length of connecting rod
X3=d(3); %offset distance
N=2000; Nu=20;
MuX1=X1;  stdX1 = MuX1/100;
MuX2=X2;  stdX2 = MuX2/100;
MuX3=X3;  aX3=MuX3-(MuX3/20); bX3=MuX3+(MuX3/20);
max_std_10_norm = 0.029516808208; max_std_60_norm = 0.03386878670962;
sdiff_10_norm = 2.231075060847541e-004; sdiff_60_norm = 9.673179655116104e-004;
```

```
% Step 1 - Sampling on random variables
randn('state',0)    % Initialize the normal random variable generator
X1_sample =normrnd(MuX1,stdX1,N,1); %sample of X1
X2_sample =normrnd(MuX2,stdX2,N,1); %sample of X2
X3_sample = aX3:(bX3-aX3)/(Nu-1):bX3;

% Step 2 - Experimentation
for i=1:Nu
   for j=1:N
         actual_S_10(i,j) = (X1_sample(j,1)*cos(10*pi/180)) + ((X2_sample(j,1)^2)…
…- (X3_sample(1,i)+ X1_sample(j,1)*sin(10*pi/180))^2)^0.5;
         actual_S_60(i,j) = (X1_sample(j,1)*cos(60*pi/180)) + ((X2_sample(j,1)^2)…
…- (X3_sample(1,i)+ X1_sample(j,1)*sin(60*pi/180))^2)^0.5;
   end
end
for i=1:Nu
   std_actual_S_10 (1,i) = std(actual_S_10(i,:));
   std_actual_S_60 (1,i) = std(actual_S_60(i,:));
end
max_std_10 = max(std_actual_S_10);
max_std_60 = max(std_actual_S_60);
sdiff_10 = max_std_10 - min(std_actual_S_10);
sdiff_60 = max_std_60 - min(std_actual_S_60);

obj = (max_std_10/max_std_10_norm) + (max_std_60/max_std_60_norm)…
… + (sdiff_10/sdiff_10_norm) + (sdiff_60/sdiff_60_norm);
```

Robustness Assessment

```
clc; close all; clear all; format long;

N= input('Enter Number of Samples  ');
Nu=input('Enter Number of intervals for interval variable  =  ');
MuX1=1.133;  stdX1 = 0.0113; %length of crank
MuX2=2.5306;  stdX2 = 0.025; %length of connecting rod
MuX3=0.6515;  aX3=MuX3-0.0163; bX3=MuX3+0.0163; %offset distance

% Step 1 - Sampling on random variables
randn('state',0)    % Initialize the normal random variable generator

X1 = normrnd(MuX1,stdX1,N,1); %sample of X1
X2 = normrnd(MuX2,stdX2,N,1); %sample of X2
X3 = aX3:(bX3-aX3)/(Nu-1):bX3; %Intervals of X3

% Step 2 - Experimentation
S_desired_10 =(MuX1*cos(10*pi/180)) + ((MuX2^2)…
                    …-(MuX3+MuX1*sin(10*pi/180))^2)^0.5
S_desired_60 =(MuX1*cos(60*pi/180)) + ((MuX2^2)…
                    …- (MuX3+MuX1*sin(60*pi/180))^2)^0.5
for i=1:Nu
   for j=1:N
         actual_S_10(i,j) = (X1(j,1)*cos(10*pi/180)) + ((X2(j,1)^2) - (X3(1,i)…
                              …+ X1(j,1)*sin(10*pi/180))^2)^0.5;
         actual_S_60(i,j) = (X1(j,1)*cos(60*pi/180)) + ((X2(j,1)^2) - (X3(1,i)…
                              …+ X1(j,1)*sin(60*pi/180))^2)^0.5;
   end
end
for i=1:Nu
   std_actual_S_10 (1,i) = std(actual_S_10(i,:));
   std_actual_S_60 (1,i) = std(actual_S_60(i,:));
end
std_actual_S_10
std_actual_S_60
```

```
max_std_10 = max(std_actual_S_10)
max_std_60 = max(std_actual_S_60)
sdiff_10 = max_std_10 - min(std_actual_S_10)
sdiff_60 = max_std_60 - min(std_actual_S_60)

% Plotting the pdf curves
n_point = 15;
for j=1:Nu
   step_10 = (max(actual_S_10(j,:))-min(actual_S_10(j,:))) / n_point;
   step_60 = (max(actual_S_60(j,:))-min(actual_S_60(j,:))) / n_point;
   for i = 1:n_point
      S_point_10(j,i) = min(actual_S_10(j,:)) + (i-1)*step_10;
      S_point_60(j,i) = min(actual_S_60(j,:)) + (i-1)*step_60;
   end
end
for j=1:Nu
   m_10(j,:) = hist (actual_S_10(j,:),S_point_10(j,:));
   pdf_10(j,:) = m_10(j,:)/N;
   m_60(j,:) = hist (actual_S_60(j,:),S_point_60(j,:));
   pdf_60(j,:) = m_60(j,:)/N;
end
figure;
for j=1:Nu
   plot(S_point_10(j,:),pdf_10(j,:));
   xlabel('Slider Distance (in) at 10 degrees'); ylabel('pdf');
   hold on;
end
figure;
for j=1:Nu
   plot(S_point_60(j,:),pdf_60(j,:));
   xlabel('Slider Distance (in) at 60 degrees'); ylabel('pdf');
   hold on;
end
```

APPENDIX B.

MATLAB PROGRAM FOR EXAMPLE 2

```
%MAIN PROGRAM
clc; warning off; format long; clear all;

%Deterministic Mechanism synthesis
disp('Deterministic synthesis');
option=1;
d0=[8 1 3 8 3 10];  % starting point of r1,r2,r3,r4,rp,beta
lb=[1 1 1 1 1 10];  %lower bounds for design variables
ub=[20 20 20 20 20 70]; %upper bounds for design variables
method=1;
N=1000;Nu=10;
option = optimset('display','iter'); %set options to show the optimization history
normalization=[];
der_d=fmincon('obj_prog',d0,[],[],[],[],lb,ub,'constr_prog',option,...
   N,Nu,normalization,method); % call the optimizer
%Displaiy results
disp('Design Variables =');
disp(num2str(der_d));
method=2;
[nom_Px,nom_Py,mean_Px,mean_Py,min_std_Px,max_std_Px,mean_std_Px,...
       min_std_Py,max_std_Py,mean_std_Py,diff_std_Px,diff_std_Py]...
   =analysis_obj(der_d,N,Nu,method);
[nom_g1,nom_g2,max_mean_g1,max_mean_g2,max_std_g1,max_std_g2]=...
   analysis_constr(der_d,N,Nu,method);
disp('Nominal positions [x1,y1], [x2,y2]=');
disp(['[',num2str(nom_Px(1)),',',num2str(nom_Py(1)),']',...
' [',num2str(nom_Px(2)),',',num2str(nom_Py(2)),']']);
disp('Maximum std of positions [x1,y1], [x2,y2]=');
disp(['[',num2str(max_std_Px(1)),',',num2str(max_std_Py(1)),']',...
' [',num2str(max_std_Px(2)),',',num2str(max_std_Py(2)),']']);
disp('Minimum std of positions [x1,y1], [x2,y2]=');
disp(['[',num2str(min_std_Px(1)),',',num2str(min_std_Py(1)),']',...
' [',num2str(min_std_Px(2)),',',num2str(min_std_Py(2)),']']);
disp('Mean std of positions [x1,y1], [x2,y2]=');
disp(['[',num2str(mean_std_Px(1)),',',num2str(mean_std_Py(1)),']',...
' [',num2str(mean_std_Px(2)),',',num2str(mean_std_Py(2)),']']);
disp('Diff std of positions [x1,y1], [x2,y2]=');
```

```
disp(['[',num2str(diff_std_Px(1)),',',num2str(diff_std_Py(1)),']',...
' [',num2str(diff_std_Px(2)),',',num2str(diff_std_Py(2)),']']);
disp(['Nominal g1=',num2str(nom_g1)]);
disp(['Max mean g1=',num2str(max_mean_g1)]);
disp(['Max std g1=',num2str(max_std_g1)]);
disp(['Nominal g2=',num2str(nom_g2)]);
disp(['Max mean g2=',num2str(max_mean_g2)]);
disp(['Max std g2=',num2str(max_std_g2)]);
normalization=[nom_Px,nom_Py,...
        mean_std_Px(:,:,1),mean_std_Px(:,:,2),...
        mean_std_Py(:,:,1),mean_std_Py(:,:,2),...
        diff_std_Px(:,:,1),diff_std_Px(:,:,2),...
        diff_std_Py(:,:,1),diff_std_Py(:,:,2)];

%Robust design
robust_d=fmincon('obj_prog',der_d,[],[],[],[],lb,ub,'constr_prog',option,...
    N,Nu,normalization,method); % call the optimizer
%Displaiy results
disp('Design Variables =');
disp(num2str(robust_d));
method=2;
[nom_Px,nom_Py,mean_Px,mean_Py,min_std_Px,max_std_Px,mean_std_Px,...
        min_std_Py,max_std_Py,mean_std_Py,diff_std_Px,diff_std_Py]...
    =analysis_obj(robust_d,N,Nu,method);
[nom_g1,nom_g2,max_mean_g1,max_mean_g2,max_std_g1,max_std_g2]=...
    analysis_constr(robust_d,N,Nu,method);
disp('Nominal positions [x1,y1], [x2,y2]=');
disp(['[',num2str(nom_Px(1)),',',num2str(nom_Py(1)),']',...
' [',num2str(nom_Px(2)),',',num2str(nom_Py(2)),']']);
disp('Maximum std of positions [x1,y1], [x2,y2]=');
disp(['[',num2str(max_std_Px(1)),',',num2str(max_std_Py(1)),']',...
' [',num2str(max_std_Px(2)),',',num2str(max_std_Py(2)),']']);
disp('Minimum std of positions [x1,y1], [x2,y2]=');
disp(['[',num2str(min_std_Px(1)),',',num2str(min_std_Py(1)),']',...
' [',num2str(min_std_Px(2)),',',num2str(min_std_Py(2)),']']);
disp('Mean std of positions [x1,y1], [x2,y2]=');
disp(['[',num2str(mean_std_Px(1)),',',num2str(mean_std_Py(1)),']',...
```

```
' [',num2str(mean_std_Px(2)),',',num2str(mean_std_Py(2)),']']);
disp('Diff std of positions [x1,y1], [x2,y2]=');
disp(['[',num2str(diff_std_Px(1)),',',num2str(diff_std_Py(1)),']',...
' [',num2str(diff_std_Px(2)),',',num2str(diff_std_Py(2)),']']);
disp(['Nominal g1=',num2str(nom_g1)]);
disp(['Max mean g1=',num2str(max_mean_g1)]);
disp(['Max std g1=',num2str(max_std_g1)]);
disp(['Nominal g2=',num2str(nom_g2)]);
disp(['Max mean g2=',num2str(max_mean_g2)]);
disp(['Max std g2=',num2str(max_std_g2)]);
----------------------------------------------------------------------------------------------------------
```

```
%OBJECTIVE FUNCTION
function [obj,mean_Px,mean_Py,s_Px,s_Py]=obj_prog(d,N,Nu,normalization,method)
%objective function
Px1_req=3.8;    Px2_reg=3;
Py1_reg=3;      Py2_reg=5;
[nom_Px,nom_Py,mean_Px,mean_Py,min_std_Px,max_std_Px,mean_std_Px,...
        min_std_Py,max_std_Py,mean_std_Py,diff_std_Px,diff_std_Py]...
        =analysis_obj(d,N,Nu,method);
error=(nom_Px(1)-Px1_req)^2+(nom_Py(1)-Py1_reg)^2+...
   (nom_Px(2)-Px2_reg)^2+(nom_Py(2)-Py2_reg)^2;
obj=error^0.5;
if method==2
   nom_Px1=normalization(1);
   nom_Px2=normalization(2);
   nom_Py1=normalization(3);
   nom_Py2=normalization(4);
   mean_std_Px1=normalization(5);
   mean_std_Px2=normalization(6);
   mean_std_Py1=normalization(7);
```

```
    mean_std_Py2=normalization(8);
    diff_std_Px1=normalization(9);
    diff_std_Px2=normalization(10);
    diff_std_Py1=normalization(11);
    diff_std_Py2=normalization(12);
    obj=mean_std_Px(1)/mean_std_Px1+mean_std_Px(2)/mean_std_Px2...
      +mean_std_Py(1)/mean_std_Py1+mean_std_Py(2)/mean_std_Py2...
      +diff_std_Px(1)/diff_std_Px1+diff_std_Px(2)/diff_std_Px2...
      +diff_std_Py(1)/diff_std_Py1+diff_std_Py(2)/diff_std_Py2;
end
---------------------------------------------------------------------------------------------------------

%CONSTRAINT FUNCTION
function [c,ceq] = constr_prog(d,N,Nu,normalization,method)
%constraint function
[nom_g1,nom_g2,max_mean_g1,max_mean_g2,max_std_g1,max_std_g2]...
    =analysis_constr(d,N,Nu,method);
c(1)=nom_g1;
c(2)=nom_g2-50;
ceq=[];
Px1_req=3.8;    Px2_req=3;
Py1_req=3;      Py2_req=5;
if method==2
    [nom_Px,nom_Py,mean_Px,mean_Py,min_std_Px,max_std_Px,mean_std_Px,...
        min_std_Py,max_std_Py,mean_std_Py,diff_std_Px,diff_std_Py]...
        =analysis_obj(d,N,Nu,method);
    k=3;
    c(1)=max_mean_g1+k*max_std_g1;
    c(2)=max_mean_g2+k*max_std_g2-50;
    ceq(1)=nom_Px(1)-Px1_req;
    ceq(2)=nom_Px(2)-Px2_req;
    ceq(3)=nom_Py(1)-Py1_req;
    ceq(4)=nom_Py(2)-Py2_req;
end
---------------------------------------------------------------------------------------------------------
function F = obj_sub(x,c)
```

```
F= [ c(2)*cos(c(6)*pi/180) + c(3)*cos(x(1)*pi/180) - c(4)*cos(x(2)*pi/180) - c(1);
     c(2)*sin(c(6)*pi/180) + c(3)*sin(x(1)*pi/180) - c(4)*sin(x(2)*pi/180) ];
----------------------------------------------------------------------------------------------------------



%Mechanism analysis for constraints
function [nom_g1,nom_g2,max_mean_g1,max_mean_g2,max_std_g1,max_std_g2]
=analysis_constr(d,N,Nu,method)
r1=d(1); r2=d(2); r3=d(3); r4=d(4);
%option=1: deterministic %option=2: robust
nom_g1=0;nom_g2=0;max_mean_g1=0;max_mean_g2=0;max_std_g1=0;max_std_g2=0;
r1=d(1); r2=d(2); r3=d(3); r4=d(4); rp=d(5); beta=d(6);

the2=[10 45];
the0=[20 80]; % initial values for theta

nom_g1=r2+r1-r3-r4;
mu1=acos((r3^2 + r4^2 - (r1+r2)^2)/(2*r3*r4))*180/pi;
mu2=acos((r3^2 + r4^2 - (r1-r2)^2)/(2*r3*r4))*180/pi;
nom_g2=max(abs(90-mu1),abs(90-mu2));
diff_std_Px=0;
diff_std_Py=0;

if method==2
%Define uncertainties
   m_r1 = r1; l_r1 = m_r1-0.5; u_r1 = m_r1 + 0.5;
   m_r2 = r2; s_r2 = 0.05;
   m_r3 = r3; s_r3 = 0.1;
   m_r4 = r4; s_r4 = 0.05;
   m_rp = rp; s_rp = 0.05;
%Sampling
   randn('state',0)    % Initialize the normal random variable generator
   r1_sample = l_r1:(u_r1-l_r1)/(Nu-1):u_r1;
   r2_sample = normrnd(m_r2,s_r2,N,1);
   r3_sample = normrnd(m_r3,s_r3,N,1);
   r4_sample = normrnd(m_r4,s_r4,N,1);
```

```
   rp_sample = normrnd(m_rp,s_rp,N,1);
   for k=1:Nu
      for i=1:N
         g1(k,i) = r1_sample(k) + r2_sample(i) - r3_sample(i) - r4_sample(i);
         mu1=acos((r3_sample(i)^2+r4_sample(i)^2-(r1_sample(k)+r2_sample(i))^2)...
            /(2*r3_sample(i)*r4_sample(i)))*180/pi;
         mu2=acos((r3_sample(i)^2+r4_sample(i)^2-(r1_sample(k)-r2_sample(i))^2)...
            /(2*r3_sample(i)*r4_sample(i)))*180/pi;
         g2(k,i)=max(abs(90-mu1),abs(90-mu2));
      end
   end
   max_mean_g1=max(mean(g1)); max_mean_g2=max(mean(g2));
   max_std_g1= max(std(g1)); max_std_g2=max(std(g2));
end
---------------------------------------------------------------------------------------------------------
%Mechanism analysis for the objective
function [nom_Px,nom_Py,mean_Px,mean_Py,min_std_Px,max_std_Px,mean_std_Px,...
      min_std_Py,max_std_Py,mean_std_Py,diff_std_Px,diff_std_Py]...
      =analysis_obj(d,N,Nu,method)
%method=1: deterministic %method=2: robust
nom_Px=0; nom_Py=0; mean_Px=0; mean_Py=0; min_std_Px=0; max_std_Px=0; mean_std_Px=0;
min_std_Py=0; max_std_Py=0; mean_std_Py=0;

r1=d(1); r2=d(2); r3=d(3); r4=d(4); rp=d(5); beta=d(6);

the2=[10 45];
the0=[20 80]; % initial values for theta
option=optimset('Display','off');

for i=1:2
   c = [r1 r2 r3 r4 rp the2(i) beta];
   the0 =[30 100];
   x=fsolve(@obj_sub,the0,option,c);
   a3(i) = x(1); a4(i) = x(2);
   Px(i) = r2*cos(the2(i)*pi/180) + rp*cos((beta+a3(i))*pi/180);
   Py(i) = r2*sin(the2(i)*pi/180) + rp*sin((beta+a3(i))*pi/180);
end
```

```
nom_Px=Px; nom_Py=Py;
diff_std_Px=0;
diff_std_Py=0;

if method==2
%Define uncertainties
m_r1 = r1; l_r1 = m_r1-0.5; u_r1 = m_r1 + 0.5;
m_r2 = r2; s_r2 = 0.05;
m_r3 = r3; s_r3 = 0.1;
m_r4 = r4; s_r4 = 0.05;
m_rp = rp; s_rp = 0.05;
m_b = beta; l_b = m_b - 1 ; u_b = m_b + 1;

%Sampling
randn('state',0)    % Initialize the normal random variable generator
r1_sample = l_r1:(u_r1-l_r1)/(Nu-1):u_r1;
r2_sample = normrnd(m_r2,s_r2,N,1);
r3_sample = normrnd(m_r3,s_r3,N,1);
r4_sample = normrnd(m_r4,s_r4,N,1);
rp_sample = normrnd(m_rp,s_rp,N,1);
b_sample = l_b:(u_b-l_b)/(Nu-1):u_b;
for kr1=1:Nu % Intervals of r1
  for kb=1:Nu % Intervals of b
    for i=1:N   %N samples
      for j=1:2  %2 positions
        c = [r1_sample(kr1) r2_sample(i) r3_sample(i) r4_sample(i) rp_sample(i) the2(j)];
        x = fsolve(@obj_sub,the0,option,c);
        Px_mcs(kr1,kb,i,j) = r2_sample(i)*cos(the2(j)*pi/180)...
          + rp_sample(i)*cos((b_sample(kb)+x(1))*pi/180);
        Py_mcs(kr1,kb,i,j) = r2_sample(i)*sin(the2(j)*pi/180)...
          + rp_sample(i)*sin((b_sample(kb)+x(1))*pi/180);
        the0 = x;
      end
    end
  end
end
for kr1=1:Nu
```

```
  for kb=1:Nu
    temp_Px(:,:)=Px_mcs(kr1,kb,:,:);
    temp_Py(:,:)=Py_mcs(kr1,kb,:,:);
    for j=1:2
      m_Px(kr1,kb,j) = mean (temp_Px(:,j));
      m_Py(kr1,kb,j) = mean (temp_Py(:,j));
      s_Px(kr1,kb,j) = std (temp_Px(:,j));
      s_Py(kr1,kb,j) = std (temp_Py(:,j));
    end
  end
end
mean_Px = mean(mean(m_Px)); mean_Py = mean(mean(m_Py));
max_std_Px = max(max(s_Px)); min_std_Px = min(min(s_Px));
std_diff_Px = max_std_Px - min_std_Px;
mean_std_Px = mean(mean(s_Px));
max_std_Py = max(max(s_Py)); min_std_Py = min(min(s_Py));
std_diff_Py = max_std_Py - min_std_Py;
mean_std_Py = mean(mean(s_Py));
diff_std_Px=max_std_Px-min_std_Px;
diff_std_Py=max_std_Py-min_std_Py;
end
```

BIBLIOGRAPHY

[1] G. Taguchi, S. Chowdhury and S. Taguchi, Robust Engineering, McGraw Hill, 2000.

[2] G. Taguchi, Taguchi on Robust Technology Development: Bringing Quality Engineering Upstream, ASME Press, Newyork, 1993.

[3] A. Parkinson, C. Sorensen and N. Pourhassan, "A General Approach for Robust Optimal Design," ASME Journal of Mechanical Design, Vol. 115, 1993.

[4] G.J. Park, T.H. Lee, K.H. Lee and K.H. Hwang, "Robust Design: An Overview," AIAA Journal, Vol. 44, 2006.

[5] S. Sundaresan, K. Ishii and D.R. Houser, "A Robust Optimization Procedure with Variations on Design Variables and Constraints," Advances in Design Automation, ASME DE, Vol. 69, 1993.

[6] J.C. Yu and K.Ishii, "Design for Robustness Based on Manufacturing Variation Patterns," Journal of Mechanical Design, Vol. 120, 1998.

[7] G. Maglaras, E. Ponslet, R. T. Haftka, E. Nikolaidis, P. Sensharma, and H. H. Cudney, "Analytical and Experimental Comparison of Probabilistic and Deterministic Optimization," AIAA Journal, Vol. 34, 1996.

[8] M.S. Phadke, Quality Engineering Using Robust Design, Prentice Hall, 1995.

[9] X. Du and W. Chen, "Towards a Better Understanding of Modeling Feasibility Robustness in Engineering Design," ASME Journal of Mechanical Design, Vol.122, 2000.

[10] M. Kalsi, K. Hacker and K. Lewis, "A Comprehensive Robust Design Approach for Decision Trade-Offs in Complex Systems Design," Journal of Mechanical Design, Vol. 123, 2001.

[11] J. Su and J.E. Renaud, "Automatic Differentiation in Robust Optimization," AIAA Journal, Vol. 35, 1997.

[12] J. Zhu and K.L. Ting, "Performance Distribution Analysis and Robust Design," Journal of Mechanical Design, Vol. 123, 2001.

[13] http://web.umr.edu/~ccli/me360/ch1.pdf Uncertainty, July 2007.

[14] W.L. Oberkampf, J.C. Helton, C.A. Joslyn, S.F. Wojtkiewicz and S. Ferson, “Challenge Problems: Uncertainty in System Response Given Uncertain Parameters,” Reliability Engineering and System Safety, Vol.85, 2004.

[15] B. Benhabib, R.G. Fenton and A.A. Goldenberg, “Computer-Aided Joint Error Analysis of Robots,” IEEE Journal of Robotics and Automation, Vol. RA-3, 1987.

[16] S.G. Dhande and J. Chakraborty, “Analysis and Synthesis of Mechanical Error in Linkages - A Stochastic Approach,” ASME Journal of Engineering for Industry, Vol. 95, 1973.

[17] A.K. Mallik and S.G. Dhande, “Analysis and Synthesis of Mechanical Error in Path-Generating Linkages Using a Stochastic Approach,” Mechanism and Machine Theory, Vol. 22, 1987.

[18] K.H. Lee and G.J. Park, “Robust Optimization Considering Tolerances of Design Variables,” Computers and Structures, Vol. 79, 2001.

[19] J.H. Rhyu and B.M. Kwak, “Optimal Stochastic Design for Four-Bar Mechanisms for Tolerance and Clearance,” ASME Journal of Mechanical Design, Vol. 109, 1988.

[20] R.E. Garrett and A.S. Hall, “Effect of Tolerance and Clearance in Linkage Design,” ASME Journal of Engineering for Industry, Vol. 91, 1969.

[21] C . Stéphane, B. Fouad and W. Philippe, “Tolerance Synthesis of Mechanisms: A Robust Design Approach,” Transactions of the ASME, Vol.127, 2005.

[22] M.I.S. Costa, J.L. Boldrini, M.A. Silva Ramos, S. Frey, J.H.C. de Araujo and M. Da lio, “Robust Design of Linkages: Synthesis by Solving Non Linear Optimization Problems,” Mechanism and Machine Theory, Vol.32, 1997.

[23] W. Weidong and S. S. Rao, “Interval Approach for the Modeling of Tolerances and Clearances in Mechanism Analysis,” ASME Journal of Mechanical Design, Vol.126, July, 2004.

[24] X. Du, S. Agus and H. Beiqing, “Reliability-Based Design With the Mixture of Random and Interval Variables,” Transactions of the ASME, Vol.127, 2005.

[25] S.A. Starks, V. Kreinovich, L. Longpré, M. Ceberio, G. Xiang, R. Araiza, J. Beck, R. Kandathi, A. Nayak and R. Torres, “Towards Combining Probabilistic and Interval Uncertainty in Engineering Calculations,” Proceedings of the NSF Workshop on Reliable Engineering Computing, Savannah, GA, 15-17, 2004.

[26] V. Kreinovich, J. beck, C. Ferregut, A. Sanchez, G.R. Keller, M. Averill and S.A. S.A. Starks, “Monte-Carlo-Type Techniques for Processing Interval Uncertainty, and Their Engineering Applications,” Proceedings of the NSF Workshop on Reliable Engineering Computing, Savannah, GA, 15-17, 2004.

[27] T. Nilsen and T. Aven, “Models and Model Uncertainty in the Context of Risk Analysis,” Reliability Engineering and Systems Safety, Vol.79, 2003.

[28] W.L. Oberkampf and J.C. Helton, “Alternative Representations of Epistemic Uncertainty,” Reliability Engineering and System Safety, Vol.85, 2004.

[29] C.D. McAllister and T.W. Simpson, “Multidisciplinary Robust Design Optimization of an Internal Combustion Engine,” Transactions of the ASME, Vol. 125, 2003.

[30] Y. Wang, “Solving Interval Constraints in Computer-Aided Design,” Proceedings of the NSF Workshop on Reliable Engineering Computing, Savannah, GA, 15-17, 2004.

[31] W. Chen and C. Yuan, “A Probabilistic-Based Design Model for Achieving Flexibility in Design,” ASME Journal of Mechanical Design, Vol. 121, 1999.

[32] H. Beiqing, X. Du, “Analytical Robustness Assessment for Robust Design,” Structural and Multidisciplinary Optimization, Vol.34, 2007.

[33] X. Du and W. Chen, “Efficient Uncertainty Analysis Methods for Multidisciplinary Robust Design,” AIAA Journal, Vol. 40, 2002.

[34] S.S. Rao and L. Berke, “Analysis of Uncertain Structural Systems Using Interval Analysis,” AIAA journal, Vol. 35, 1997.

[35] F. Scott, C.A. Joslyn, J.C. Helton, W.L. Oberkampf and K. Sentz, “Summery from the Epistemic Uncertainty Workshop: Consensus amid Diversity,” Reliability Engineering and Systems Safety, Vol.85, 2004.

[36] C. Ole, M. Kaj and H.B. Nielsen, Introduction to Interval Analysis, IMM DTU, 2002.

[37] G. Subroto and P.Y. Papalambros, “Reliability Optimization with Mixed Continuous-Discrete Random Variables and Parameters,” Transactions of the ASME, Vol.129, February, 2007.

[38] J. Hall and J. Lawry, “Imprecise Probabilities of Engineering System Failure from Random and Fuzzy Set Reliability Analysis,” Proceedings of the Second International Symposium on Imprecise Probabilities and Their Applications, New York, 2001.

[39] K.H. Lee, I.S. Eom, G.J. Park and W.I. Lee, "Robust Design for Unconstrained Optimization Problems Using the Taguchi Method," AIAA Journal, Vol. 34, 1996.

[40] K.N. Otto and E.K. Antonsson, "Extensions to the Taguchi Method of Product Design," ASME Journal of Mechanical Design, Vol. 115, 1993.

[41] W. Chen, M.M. Wiecek and J. Zhang, "Quality Utility – A Compromise Programming Approach to Robust Design," ASME Journal of Mechanical Design, Vol. 121, 1999.

[42] N.S. George, A.G. Erdman, Advanced Mechanism Design: Analysis and Synthesis, Prentice Hall, 1984.

[43] I.Das, "Robustness Optimization for Constrained Nonlinear Programming Problems," Engineering Optimization, Vol. 32, 2000.

[44] D.C. Montgomery, Design and Analysis of Experiments, John Wiley and Sons, 5th Edition, 2004.

VITA

Pavan Kumar Venigella was born on 15th July, 1983 in Vijayawada, a city in the southern part of India. He received his Bachelors in Mechanical Engineering from Jawaharlal Nehru Technological University, Hyderabad, India in July 2004. He underwent to training in “CNC Programming and Machining” at Central Institute of Tool Design, Hyderabad, India. He joined University of Missouri Rolla in August 2005 for the masters program in Mechanical Engineering. He was involved in research with Dr. Xiaoping Du on Robust Design during his masters study. He also worked as a Graduate Teaching Assistant for the course ME 261: Engineering Design from September 2006 to May 2007. He received his Master of Science Degree in Mechanical Engineering in December 2007.