2014

Parameter Sensitivity Analysis For Co-Mediated Sickle Cell De-Polymerization

Yao Selom Messan
North Carolina Agricultural and Technical State University

Parameter Sensitivity Analysis for CO-mediated Sickle Cell De-polymerization

Yao Selom Messan

North Carolina A&T State University

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Department: Mathematics

Major: Applied Mathematics

Major Professor: Dr. Liping Liu

Greensboro, North Carolina

2014

School of Graduate Studies
North Carolina Agricultural and Technical State University
This is to certify that the Master's Thesis of

Yao Selom Messan

has met the thesis requirements of
North Carolina Agricultural and Technical State University

Greensboro, North Carolina
2014

Approved by:

Liping Liu Major Professor	Mufeed Basti Co-Advisor
Nicholas Luke Committee Member	Alexandra Kurepa Committee Member
Guoqing Tang Department Chair	

Sanjiv Sarin
Dean, The Graduate School

Biographical Sketch

Messan, Yao Selom was born on June 08, 1989 in Lome, the capital of Togo. He moved to the United States with the hope to gain a good education; thus, he enrolled in North Carolina Agricultural and Technical State University (NC A&T SU) where he completed his Bachelor's degree in Applied Mathematics in December 2011. He always has a great interest in mathematical biology. Therefore, he enrolled back at NC A&T SU to pursue a Master's degree in Applied Mathematics. There, he conducted a Parameter Sensitivity Analysis for the CO-mediated hemoglobin sickle cell de-polymerization.

Dedication

I would like to dedicate this success to my parents. I am very thankful for their support throughout my entire life. They strove all their life to provide me with a strong foundation. So I would like to devote this work to them.

Acknowledgements

I would like to express my sincere appreciation to my advisor, Professor Liping Liu, and co-advisor, Professor Mufeed Basti, for guidance and providing me with excellent amenities to continue my research and for ensuring the accuracy of my results.

I am grateful to the members of my committee for taking time to guide me through my dissertation. I would like to address my gratitude to Professor Guoqing Tang and Professor Nicholas Luke for their valuable comments, ideas and insightful remarks. I also thank Professor Daniel Kim-Shapiro from University of Wake Forest for allowing us into his laboratory and his valuable explanations on my research. Lastly, I am gratitude to Landon Bellavia, a PhD candidate at University of Wake Forest then, for his assistant throughout the laboratory experiments.

My research has profited from various researchers. Special mention goes to Professor Daniel Kim-Shapiro, whose work served as the starting point for my thesis. I would like to also acknowledge the following researchers and organizations who have provided models and input files for the evaluation of the methods developed here: Dr. Frank Ferrone and his group for the development of the double nucleation model, Dr. Daniel Kim-Shapiro and his group for the experiments conducted to study the carbon monoxide and oxygen effects on sickle hemoglobin polymers and Dr. Mingxiang Chen and her group for the numerical study that lead to the development of the extended model.

I am thankful to my fellow graduate students and colleagues at North Carolina Agricultural and Technical State University for their valuable discussions, suggestions and supports throughout my master's thesis research and the time being in the graduate program. I

am also thankful for the countless authors of all the free software that I have utilized during my research work – their unselfishness actions have meaningfully helped my work.

This thesis would not have appeared but for the help and assistance of Dr. Liping Liu. As my professor, as my mentor, as a colleague and as a friend, she helped me continuously, and has contributed enormously to this thesis. I consider it as a fortune to have her as a mentor.

Table of Contents

List of Figures

List of Tables

Abstract

Sickle cell anemia is an abnormality that causes a deformation in the shape of the red blood cell that hinders the circulation of the red blood cell through the blood vessel. The deformation is caused by the association of monomers to each other to form polymers (polymerization). The oxygenation of the sickle cell may lead to the melting of polymers (de-polymerization). Many studies have been conducted to understand the dynamics of de-polymerization. This study focuses on the impact of various parameters over the output values in the system of de-polymerization. Both mathematical and statistical approaches for the sensitivity analysis of the parameters are developed and conducted on the carbon monoxide (CO) mediated sickle cell hemoglobin (HbS) de-polymerization.

The sensitivity analysis measures how sensitive the model output is with respect to the changes of the model input parameters and which input parameters are key factors that affect the model output. There are many approaches in the parameter sensitivity analysis. This study focuses on two: the traditional sensitivity analysis (TSA) that utilizes the traditional sensitivity functions (TSFs) and the multi-parameter sensitivity analysis (MPSA). The TSA is a local sensitivity analysis that computes the first-order partial derivatives of the system output with respect to the input parameters, i.e. the TSFs. The TSFs are obtained numerically by the Runge-Kutta method on the sensitivity equations. The MPSA is a global sensitivity analysis that enumerates the overall effect of the model input parameters on the output by perturbing the model input parameters within large ranges. The MPSA is implemented by employing a Monte Carlo method over a broad range of parameters values and comparing the cumulative distribution functions of the acceptance and unacceptance groups of the parameters.

All four concentrations as model outputs and four binding/melting rates as input parameters are considered in this study. The sensitivity results from TSA and MPSA are essentially consistent. For the model output de-oxygenated monomers, the most sensitive parameter is the CO-binding rate of monomers, while the most insensitive parameter is the CO-binding rate of polymers being that it does not affect the de-oxygenated monomers at all. For the model output CO-bound monomers, the most sensitive parameter is the melting rate of the de-oxygenated polymers, while both CO-binding rate of polymers and melting rate of polymers are insensitive. For the model output de-oxygenated polymers, the most sensitive parameter is the melting rate of de-oxygenated polymers, while the other three parameters are insensitive. For the model output CO-bound polymers, the most sensitive parameter is the CO-binding rate on polymers, while the most insensitive parameter is the melting rate of CO-bound polymers.

CHAPTER 1

Introduction

1.1 Biology Background

The sickle cell trait originates as a natural mutation of the hemoglobin gene. The mutation is in the position 6 of the beta chain ($\beta6$) of the hemoglobin where glutamic acid is replaced by valine. The mutation results in the aggregation, in the form of a polymer, of the sickle cell hemoglobin (HbS) when it is in the de-oxygenated state (Ingram, 1956). The polymerization process takes place in two stages, which are separated by a time delay (Aroutiounian, 2001): 1) homogeneous polymerization where monomers join to form a polymer; 2) heterogeneous polymerization where monomers join an existing polymer. The polymers formed in the first step can melt yielding monomers if they return to the lungs quickly enough to be re-oxygenated. The polymers that do not return to the lungs in a timely manner will likely go through the second stage. The mechanisms for the homo- and heterogeneous polymerization are referred to as single and double nucleation (Ferrone, 1985).

The formation of HbS polymers causes the deformation of the red blood cell where they adopt a sickle shape, and that gave the name of the disease as sickle cell. Due to their double concave shape and flexibility, the normal red blood cells flow freely through the blood vessels. The sickle red blood cells, on the other hand, do not. They can be trapped within the blood vessels and obstruct the normal circulation of the blood. This obstruction in the blood vessel can result in vascular occlusion causing sudden pain in all parts of the body including fingers, arms, legs, ribs, abdomen, and organs such as the spleen, brain and eyes. Painful crisis vary in duration, intensity, location and time. It may be mild, moderate or severe and swelling may be noticed in the pain area (Vedro, 2002). If the shortage of blood occurs for a long period of time, tissues and organs in the body may suffer severe consequences. Figure 1a shows a normal red blood cell containing normal hemoglobin flowing through a blood vessel. Figure 1b shows the constrictive motion of sickle red blood cells through a blood vessel.

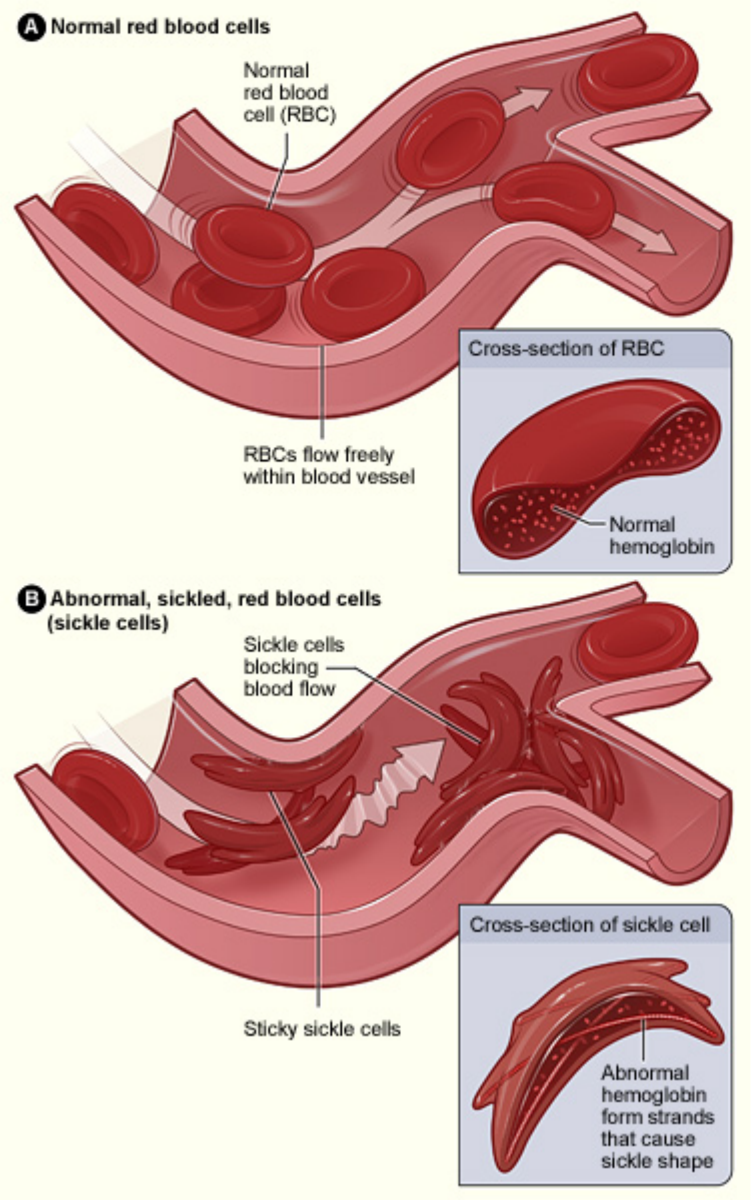
A Normal red blood cells
Normal red blood cell (RBC)
RBCs flow freely within blood vessel
Cross-section of RBC
Normal hemoglobin
B Abnormal, sickled, red blood cells (sickle cells)
Sickle cells blocking blood flow
Sticky sickle cells
Cross-section of sickle cell
Abnormal hemoglobin form strands that cause sickle shape

Figure 1. Hemoglobin passing through red blood vessels (Smith, 2013).

Ferrone (1985) focused on the growth of the sickle hemoglobin (HbS) polymers or fibers. He explained that the growth of the HbS fibers follows the double nucleation mechanism; which, he described as homogeneous and heterogeneous nucleation. The formation of a polymer is initiated by homogeneous nucleation in the solution phase. Local temperature and concentration of the homogenous nucleus can then stimulate the formation of additional polymers on the surface of an existing polymer. This second pathway is called heterogeneous nucleation. The surface of the heterogeneous nucleation is constantly increasing with time (Ferrone, 1985). From the understanding of these two processes, Ferrone developed a mathematical model for the melting of the polymers (de-polymerization) on the assumption that the polymer melting is the reversal of polymerization. The model can be described by two rate equations, one for the formation of polymers and the second for the incorporation of monomers into polymers.

The kinetics of the sickle cell hemoglobin polymer melting has been studied further by using the stopped flow method where the melting was monitored using a light scattering (Aroutiounian, 2001). The results showed that polymers melt more quickly in cells containing oxygen carbon monoxide. Therefore, two sets of experiments were conducted to study polymer melting. The first experiment involved saturating the sickle hemoglobin with oxygen (O_2) and the second experiment consisted of saturating the sickle hemoglobin with carbon monoxide (CO) to conclude which chemical compound would enhance polymer melting more efficiently (Aroutiounian, 2001). The author concluded that polymers melt more efficiently in presence of CO. Additionally, the authors noted that the CO can bind to HbS in the polymeric or in the monomeric form (Aroutiounian, 2001). Furthermore, this study improved Ferrone’s model, which is a single equation. The model was rewritten as a system of two equations by including the fact that polymer only melts from the ends and CO can only bind to the solution phase HbS.

After further investigation of the effects of CO binding to the solution phase of the polymer, a nonlinear model was developed that extended the original two-species model into a four-species model (Chen, et al. 2010). Then, mathematical analyses were conducted to study the effect of CO on the kinetics of polymer melting. Results from the analyses revealed that when sickle hemoglobin solution was saturated with CO, the polymer melting occurred rapidly. Furthermore, it was noted that the CO is bound to the solution phase monomers and polymers (Aroutiounian, 2001; Chen, et al., 2010). Additionally, it was observed that not all the polymers melt in the CO- binding equilibrium stage, indicating that the melting of the CO-bound polymer takes place as an equilibrium process. The four-species model proposed in (Chen, et al. 2010) describes the dynamic interaction in each phase of the melting and CO binding processes.

1.2 Methodology Approach

The behavior of a cell is not only determined by the characteristics of the individual biological component but also by the interaction of such components acting together as a system. Thus, literature review suggests that to complete the mathematical modeling of the CO-mediated HbS polymer melting, the assessment of the impact of parameters is fundamental. The development of a predictive dynamic model requires information concerning the initial conditions and the parameters that describe the system (Zi, 2011). Unfortunately, these parameters are often difficult to determine with biological experiments. Moreover, some parameters may have large variations depending on the instrument or state of measurement. Therefore, our confidence in the model prediction is limited to the uncertainty of the model parameters. Parameter sensitivity analysis is the standard procedure to conclude how the variability in the model output can be attributed to the variations in the model inputs (Salteli, 2008).

Sensitivity analyses methods have been widely applied to study biological systems (Zi, 2011). They can also be a useful guide in model simplification by quantifying the dependence of the model outputs on its input. Through sensitivity analysis, we can gain valuable insights about how robust the model output is with respect to the changes of the model inputs and which inputs are key factors that affect the model output. The value of the insensitive parameters may be fixed and some processes may be simplified or eliminated. This sensitivity analysis provides guidance toward parameters that must be taken into consideration during the experimental design (Raue, 2011). Figure 2 is a modified Phair's scheme that illustrates the importance of the parameter sensitivity analysis in the development of a mathematical model (Phair, 2001).

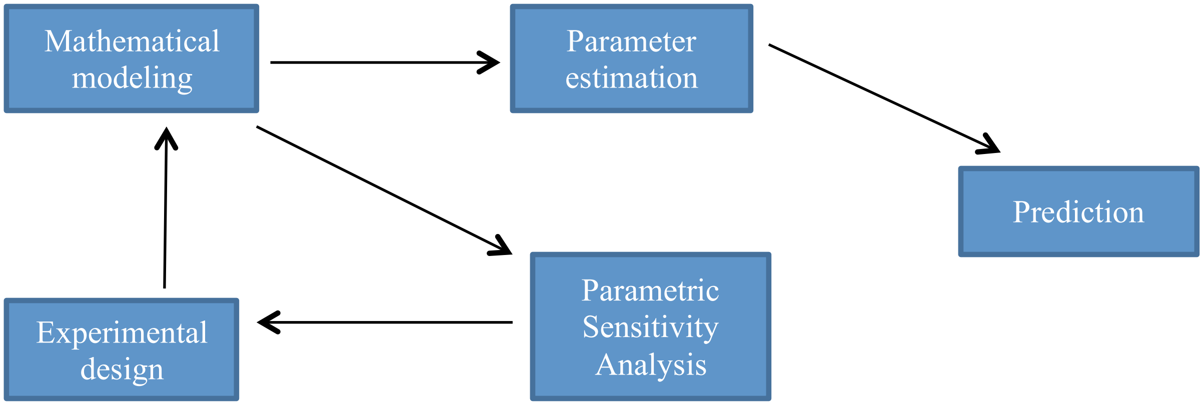

Figure 2. Modified Phair's scheme for mathematical modeling (Phair, 2001)

There have been studies concerning parameter estimation, which have led to approximation of parameters value in the nonlinear dynamic system (Aroutiounian, 2001; Chen, 2010). However there still remains the possibility to improve the measurement of the parameter values. This paper attempts to assess the importance of parameters in the extended model of melting (de-polymerization) of HbS polymers in the presence of CO.

There are many approaches to study the sensitiveness of a model output with respect to the input parameter. These approaches include local and global sensitivity analyses. Local sensitivity analysis is a common method that studies the sensitivity of the output by computing the first-order partial derivatives of the system output with respect to the input parameters. The global sensitivity analysis is used to enumerate the overall effect of the model input parameters on the output by perturbing the model input parameter within large ranges (Zi, 2011). This study particularly focuses on a traditional sensitivity analysis (TSA) and a multi-parameter sensitivity analysis (MPSA) on the extended version of the model with four species (Chen et. al 2010).

The TSA is a local sensitivity analysis that computes the first-order partial derivatives of the system output with respect to the input parameters, i.e. the TSFs. First, the sensitivity functions are derived by differentiating the original equations in the model with respect to each parameter. This leads to a system of total twenty (20) ordinary differential equations (ODEs) in the original variables and the TSFs, i.e. the system of sensitivity equations. The ODE system is then solved numerically by the Runge-Kutta method. The TSA is simple and straightforward. It is useful in finding the sensitivity of a model output given a specific parameter value at a time.

The MPSA is a global sensitivity analysis that enumerates the overall effect of the model input parameters on the model output. A large sampling of the parameters within large ranges is drawn following the uniform distribution. The model output values are then computed and compared with the model output from the system with the nominal parameters. With a criterion, the parameters are categorized as accepted or unaccepted based on the behavior of the corresponding output values versus the criterion. Finally, the sensitivity of the parameters is quantified by employing the Kolmogorov-Smirnov test and computing the distance on the cumulative distribution functions of acceptance and unacceptance parameters. MPSA is very useful in providing the overall impact of the input parameters on the model output.

All four concentrations as model outputs and four binding/melting rates as input parameters are considered in this study. The sensitivity results from TSA and MPSA are essentially consistent. For the model output de-oxygenated monomers, the most sensitive parameter is the CO-binding rate of monomers, while the most insensitive parameter is the CO-binding rate of polymers being that it does not affect the de-oxygenated monomers at all. For the model output CO-bound monomers, the most sensitive parameter is the melting rate of the de-oxygenated polymers, while both CO-binding rate of polymers and melting rate of polymers are insensitive. For the model output de-oxygenated polymers, the most sensitive parameter is the melting rate of de-oxygenated polymers, while the other three parameters are insensitive. For the model output CO-bound polymers, the most sensitive parameter is the CO-binding rate on polymers, while the most insensitive parameter is the melting rate of CO-bound polymers.

The remainder of this thesis consists of the following sections. Chapter 2 presents the derivation of the mathematical model equations, from a description of the double nucleation process to the extended CO-mediated sickle cell polymer melting (de-polymerization). Then, Chapter 3 explains the detailed methodology of the parameter sensitivity analysis including the TSA and MPSA. Chapter 4 reports and discusses the obtained results from our analysis. Chapter 5 summarizes the conclusion of the thesis and provides the recommendation for future work. Lastly, Appendix A and B includes the detailed sensitivity equations and the Matlab codes for TSA and MPSA.

CHAPTER 2

Model Equations

2.1 Double Nucleation Mechanism

Our current version of the model is an extension of the basic model proposed in Ferrone (1985) and Aroutiounian (2001). The model developed for the HbS polymer melting is based on the observation that the HbS fiber melting is the reversal of growth. According to Ferrone 1985), the growth of the fiber occurs through a double nucleation process. Figure 3 depicts the process:

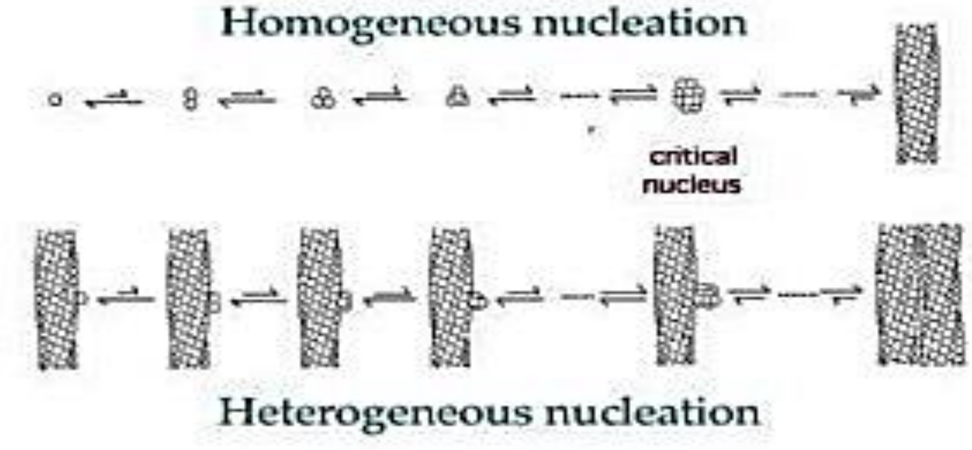

Figure 3. The double nucleation model.

Homogeneous nucleation (top) and heterogeneous nucleation (bottom) nuclei are assumed to be pieces of the infinite polymer (Ferrone, 1985).

As it happens in the solution, the homogeneous nucleation is the simple process where the monomers start attaching to each other to form a polymer. In the second part, with an existing polymer, the monomers add onto it to form a more complex polymer. As Figure 3 depicts, the length of the arrows going back and forth changes in the different stages of the figure, which indicates the change in the relative rates of fibers melting or growth. In the early stages of the homogeneous nucleation phase the rate of melting ($K_{backward}$) is higher than the rate of growth ($K_{forward}$). The equilibrium stage (K_{eq}), where the rates of the growth and melting are the same ($K_{forward} = K_{backward}$), happens when the polymer reaches a certain size. Beyond this size, the rate of the polymer growth becomes higher than the rate of melting. The driving force for this switch in kinetics is the increase in the stability of the polymer as it exceeds the above-mentioned size. This increase in stability is due to the increasing amount of energy released as more and more monomers join the polymers which makes the polymer formation to have more negative ΔG (meaning it becomes thermodynamically more favorable). The rate of the forward process -polymer formation- then becomes more favorable than the reverse -polymer melting. The same cascade of events takes place in the heterogeneous nucleation phase. Again when the size of the polymer exceeds a certain limit the formation –or growth- of the polymer becomes more favorable than the melting of the polymer. The final size of the polymer is limited by its solubility. This means once the size of the polymer –the fiber- in the heterogeneous nucleation exceeds a certain limit it becomes insoluble, or a better explanation is that the polymer stops growing when there are no more monomers to bind to it. When the size of the polymer reaches this limit it would deplete the solution from any monomers –now the concentration of the monomer to begin with is limited by its solubility at the conditions at hand (temperature, phosphate buffer concentration and the fact that the buffer is saturated with carbon monoxide).

The double nucleation model can be described by two rate equations, one for the homogeneous and the other for the heterogeneous. Since in either homogeneous or heterogeneous nucleation there is an addiction of monomer to the nucleus, then both can be represented as (this applies to the CO bound and deoxygenated polymer):

P_i
Monomers
P_j

Figure 4. Polymer formation process.

The diagram above represents the equation $P_i + monomers \rightleftharpoons P_j$, where P_i and P_j are two forms of the polymer where the j form has one monomer more than the i form, or P_i is the homogeneous polymer and P_j is the heterogeneous polymer.

The formation of the polymer is thus expressed by:

$$\frac{dC_{p,j}(t)}{dt} = k_+ C_m(t) C_{p,i}(t) - k_- C_{p,j}(t) \qquad (2.1)$$

and the rate of disappearance of the monomers from the solution phase into polymer is given by:

$$-\frac{dC_m(t)}{dt} = k_+ C_m(t) C_{p,i}(t) - k_- C_{p,j}(t) \qquad (2.2)$$

where $C_{p,i}(t)\ and\ C_{p,j}(t)$ are the time-dependent concentration of polymers *i* and *j*, respectively, and $C_m(t)$ is the monomer concentration, k_+ is the concentration–independent rate constant for addition of monomers to nuclei or polymer *i* and k_- is the concentration- independent rate of the dissociation of a monomer from polymer *j*.

2.2 The Simple Model with two (2) Equations

Briefl (1995) observed that the growth is an elongation and the melting is a shortening of the HbS fiber from the ends. Thus melting of the HbS polymers can only occur at the end of the polymers. On a short time scale, the concentration of polymers $\mathrm{P_i}$ and $\mathrm{P_j}$ remains constant at a certain time. Therefore, Aroutiounian (2001) sets the concentration of polymers P_i and P_j to be the same as $C_p(t)$; he replaced k_+ with $\frac{k_-}{C_s}$ where C_s is the solubility concentration of the solution buffer and k_- with k^d indicating the rate constant of the melting of the de-oxygenated HbS polymer. So equation (2.2) is rewritten as the following in term of the melting rate:

$$\frac{dC_m(t)}{dt} = -k^d \left(\frac{C_m(t)}{C_s} - 1\right) C_p(t) \tag{2.3}$$

Also, the total HbS molar concentration (C_{tot}) is the sum of the molar concentration of the hemoglobin molecules in the polymer phase, C_p, and that in the monomer phase, C_m. $C_{tot}(t) = C_m(t) + C_p(t)$. After differentiating and susbtituting into (2.3) with the fact that $\frac{dC_{tot}(t)}{dt} = 0$, we have:

$$\frac{dC_p(t)}{dt} = k^d \left(\frac{C_m(t)}{C_s} - 1\right) C_p(t) \tag{2.4}$$

2.3 The Extended Model with four (4) Equations

With the CO-mediation incorporated (Chen, 2010), the polymerized and monomerized populations are then divided into two sub-populations: $C_p^{co}(t)$ and $C_m^{co}(t)$ which are CO-bound polymer and monomer HbS, respectively; and $C_p^d(t)$ and $C_m^d(t)$ which are de-oxy polymer and monomer HbS. Since CO binds to the monomer and polymer tightly, the model then assumes that CO-binding results in a decrease from de-oxy HbS $C_m^d(t)$ in the solution phase, which then becomes a gain for the CO-bound solution phase $C_m^{co}(t)$, with a CO-mediated binding rate constant k_m. Likewise with the polymer phase molecules, the assumed CO-binding outcome is a loss from the $C_p^d(t)$ and a gain to $C_p^{co}(t)$ by the CO-mediated binding rate constant k_p. The decrease of the polymer phase molecule leads to a gain in the CO bound monomer with a dissociation rate constant k^{co}. Our model is now the following:

$$\frac{dC_m^d(t)}{dt} = -k^d \left(\frac{C_m^d(t)}{C_s} - 1\right) C_p^d(t) - k_m[\mathrm{co}]C_m^d(t) \tag{2.5}$$

$$\frac{dC_m^{co}(t)}{dt} = k^{co}\left(1 - \frac{C_m^{co}(t)}{C_s^{co}}\right)C_p^{co}(t) + k_m[\mathrm{co}]C_m^d(t) \tag{2.6}$$

$$\frac{dC_p^d(t)}{dt} = k^d\left(\frac{C_m^d(t)}{C_s} - 1\right)C_p^d(t) - k_p[\mathrm{co}]C_p^d(t) \tag{2.7}$$

$$\frac{dC_p^{co}(t)}{dt} = -k^{co}\left(1 - \frac{C_m^{co}(t)}{C_s^{co}}\right)C_p^{co}(t) + k_p[\mathrm{co}]C_p^d(t) \tag{2.8}$$

Note: The CO-binding to the monomers reduces the concentration of $C_m^d(t)$ which results in a loss of $C_m^d(t)$ and a gain of $C_m^{co}(t)$, thus we subtract $k_m[co]C_m^d(t)$ in (2.5) and add it to (2.6). Similarly the CO-binding to the polymers results in a loss of $C_p^d(t)$ and a gain of $C_p^{CO}(t)$, thus the term $k_p[co]C_p^d(t)$ is subtracted from (2.7) and added to (2.8). These models allow CO-binding to polymers and melting occur at the endpoints as well as at the surfaces of polymer fibers (Chen et al., 2010).

The diagram below describes the reaction path in the model equations. With the CO-binding, the de-oxygenated polymers/monomers HbS are decomposed into oxygenated polymers/monomers HbS. With the melting/de-polymerization, the CO-bound/de-oxygenated polymers become CO-bound/de-oxygenated monomers. The de-oxygenated polymers are partially transformed into oxygenated polymers with the CO-binding to polymers. The CO-bound polymers are further melted into CO-bound monomers through the melting or de-polymerization. The rest of the de-oxygenated polymers are also melted into de-oxygenated monomers, which can be further transformed to CO-bound monomers with the CO-binding to the de-oxygenated monomers.

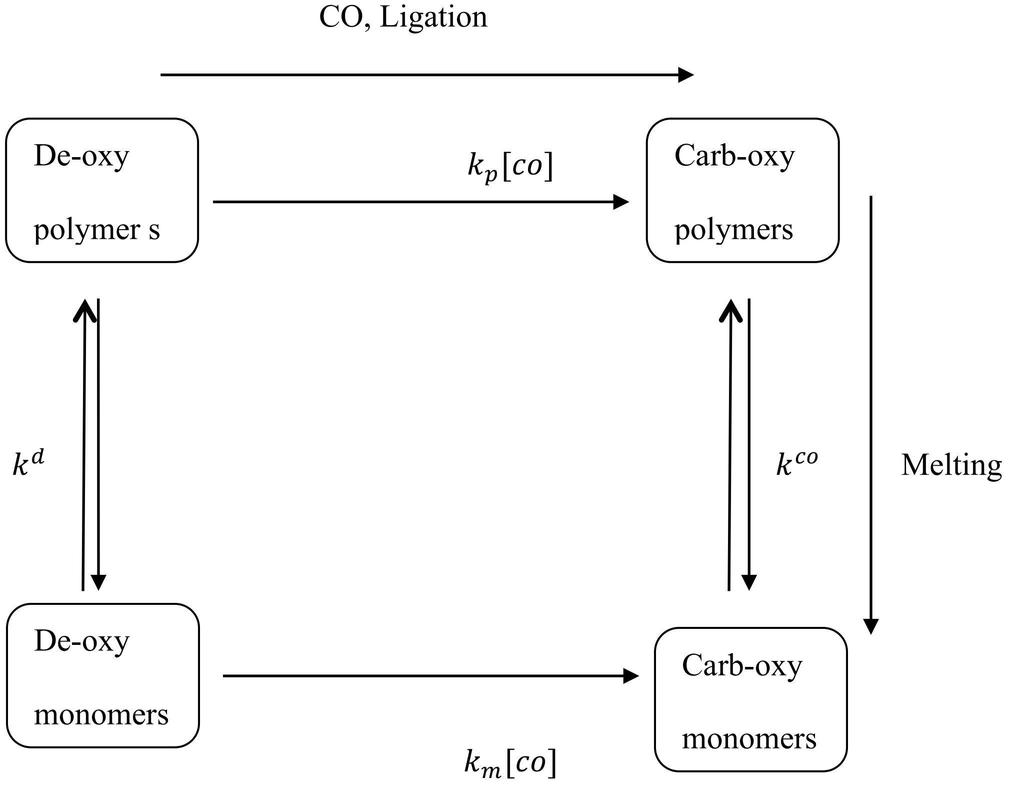

Figure 5. The reaction paths of the CO-mediated de-polymerization of sickle cell HbS

To mathematically analyze the system, we simplify the notations by replacing ($C_m^d(t)$, $C_m^{co}(t)$, $C_p^d(t)$, $C_p^{co}(t)$, k^d, $k_m[co]$, $k^{co}, k_p[co]$, C_s, C_s^{co}) by ($x(t), y(t), z(t), u(t),$ k_1, k_2, k_3, k_4, C_1, C_2) respectively (Daniel-Jones et al., 1999). We then have the following equations:

$$\frac{dx(t)}{dt} = -k_1\left(\frac{x(t)}{C_1} - 1\right)z(t) - k_2x(t) \tag{2.9}$$

$$\frac{dy(t)}{dt} = k_3\left(1 - \frac{y(t)}{C_2}\right)u(t) + k_2x(t) \tag{2.10}$$

$$\frac{dz(t)}{dt} = k_1\left(\frac{x(t)}{C_1} - 1\right)z(t) - k_4z(t) \tag{2.11}$$

$$\frac{du(t)}{dt} = -k_3\left(1 - \frac{y(t)}{C_2}\right)u(t) + k_4z(t) \tag{2.12}$$

CHAPTER 3

Methodology

3.1 Traditional Sensitivity Analysis (TSA)

The traditional sensitivity analysis is frequently used in simulation studies, where one wants to assess the degree of sensitivity of a model output with respect to various parameters on which it depends. The sensitivity functions are used to identify the parameters to which the model output is most/least sensitive. We consider the output model, to quantify the variation in the output variable with respect to changes in the parameter. Therefore, we are led to consider the first order sensitivity functions, also called traditional sensitivity functions (TSFs), which are defined naturally in terms of the partial derivatives (David, 2007).

Many times, the model is described by a system of ordinary differential equations (ODEs). The system can be written in the vector form

$$\frac{dX(t)}{dt} = F(X,P) \tag{3.1}$$

where $X \in \mathbb{R}^n$ denotes the variable vector and $P \in \mathbb{R}^r$ denotes the vector of parameters and the initial condition is $X(t_o) = X_o$. By sensitivity we mean how X changes with respect to P, that is $\frac{\partial X(t)}{\partial P}$. Note that as X is a function of time, the sensitivity is also a function of time. By differentiating both sides of Equation (3.1) with respect to P, we have a system of differential equations for the sensitivities:

$$\frac{\partial}{\partial P}\frac{dX(t)}{dt} = \frac{\partial F}{\partial X}\frac{\partial X}{\partial P} + \frac{\partial F}{\partial P}\,. \tag{3.2}$$

Now if we reverse the order of differentiation and couple the equation with Equation (3.1) then we get a $n + nr$ dimensional system of ordinary differential equations for both the model variables and the sensitivities

$$\frac{dX(t)}{dt} = F(X, P) \tag{3.3}$$

$$\frac{d}{dt}\frac{\partial X}{\partial P} = \frac{\partial F}{\partial X}\frac{\partial X}{\partial P} + \frac{\partial F}{\partial P}\,. \tag{3.4}$$

Here, we assume that $\frac{\partial X(0)}{\partial P} = 0$, because the initial conditions for the model would not be considered to be dependent on the parameters.

We apply the above Equations (3.1) and (3.2) onto our sickle cell model equations (2.5)-(2.8). The new system is obtained with 4 original model equations and 16 equations for the TSFs. The detailed equations for the TSFs are presented in Equations (3.5) – (3.24). The Matlab codes for TSA are included in Appendix B.

$$\frac{dx(t)}{dt} = -k_1\left(\frac{x(t)}{C_1} - 1\right)z(t) - k_2 x(t) \tag{3.5}$$

$$\frac{d}{dt}\frac{\partial x}{\partial k_1} = \left(1 - \frac{x(t)}{C_1}\right)z(t) + k_1\left(1 - \frac{x(t)}{C_1}\right)\frac{\partial z(t)}{\partial k_1} - k_1\,\frac{1}{C_1}\frac{\partial x(t)}{\partial k_1}z(t) - k_2\frac{\partial x(t)}{\partial k_1} \tag{3.6}$$

$$\frac{d}{dt}\frac{\partial x}{\partial k_2} = k_1\left(1 - \frac{x(t)}{C_1}\right)\frac{\partial z(t)}{\partial k_2} - k_1\,\frac{1}{C_1}\frac{\partial x(t)}{\partial k_2}z(t) - k_2\frac{\partial x(t)}{\partial k_2} + x(t) \tag{3.7}$$

$$\frac{d}{dt}\frac{\partial x}{\partial k_3} = k_1\left(1 - \frac{x(t)}{C_1}\right)\frac{\partial z(t)}{\partial k_3} - k_1\,\frac{1}{C_1}\frac{\partial x(t)}{\partial k_3}z(t) - k_2\frac{\partial x(t)}{\partial k_3} \tag{3.8}$$

$$\frac{d}{dt}\frac{\partial x}{\partial k_4} = k_1\left(1 - \frac{x(t)}{C_1}\right)\frac{\partial z(t)}{\partial k_4} - k_1\,\frac{1}{C_1}\frac{\partial x(t)}{\partial k_4}z(t) - k_2\frac{\partial x(t)}{\partial k_4} \tag{3.9}$$

$$\frac{dy(t)}{dt} = k_3 \left(1 - \frac{y(t)}{C_2}\right) u(t) + k_2 x(t) \tag{3.10}$$

$$\frac{d}{dt}\frac{\partial y}{\partial k_1} = k_3 \left(1 - \frac{y(t)}{C_2}\right) \frac{\partial u(t)}{\partial k_1} - k_3 \frac{1}{C_1}\frac{\partial y(t)}{\partial k_1} u(t) + k_2 \frac{\partial x(t)}{\partial k_1} \tag{3.11}$$

$$\frac{d}{dt}\frac{\partial y}{\partial k_2} = k_3 \left(1 - \frac{y(t)}{C_2}\right) \frac{\partial u(t)}{\partial k_2} - k_3 \frac{1}{C_1}\frac{\partial y(t)}{\partial k_2} u(t) + k_2 \frac{\partial x(t)}{\partial k_2} + x(t) \tag{3.12}$$

$$\frac{d}{dt}\frac{\partial y}{\partial k_3} = \left(1 - \frac{y(t)}{C_2}\right) u(t) + k_3 \left(1 - \frac{y(t)}{C_2}\right) \frac{\partial u(t)}{\partial k_3} - k_3 \frac{1}{C_1}\frac{\partial y(t)}{\partial k_3} u(t) + k_2 \frac{\partial x(t)}{\partial k_3} \tag{3.13}$$

$$\frac{d}{dt}\frac{\partial y}{\partial k_4} = k_3 \left(1 - \frac{y(t)}{C_2}\right) \frac{\partial u(t)}{\partial k_4} - k_3 \frac{1}{C_1}\frac{\partial y(t)}{\partial k_4} u(t) + k_2 \frac{\partial x(t)}{\partial k_4} \tag{3.14}$$

$$\frac{dz(t)}{dt} = k_1 \left(\frac{x(t)}{C_1} - 1\right) z(t) - k_4 z(t) \tag{3.15}$$

$$\frac{d}{dt}\frac{\partial z}{\partial k_1} = \left(\frac{x(t)}{C_1} - 1\right) z(t) + k_1 \left(\frac{x(t)}{C_1} - 1\right) \frac{\partial z(t)}{\partial k_1} - k_1 \frac{1}{C_1}\frac{\partial x(t)}{\partial k_1} z(t) - k_4 \frac{\partial z(t)}{\partial k_1} \tag{3.16}$$

$$\frac{d}{dt}\frac{\partial z}{\partial k_2} = k_1 \left(\frac{x(t)}{C_1} - 1\right) \frac{\partial z(t)}{\partial k_2} + k_1 \frac{1}{C_1}\frac{\partial x(t)}{\partial k_2} z(t) - k_4 \frac{\partial z(t)}{\partial k_2} \tag{3.17}$$

$$\frac{d}{dt}\frac{\partial z}{\partial k_3} = k_1 \left(\frac{x(t)}{C_1} - 1\right) \frac{\partial z(t)}{\partial k_3} + k_1 \frac{1}{C_1}\frac{\partial x(t)}{\partial k_3} z(t) - k_4 \frac{\partial z(t)}{\partial k_3} \tag{3.18}$$

$$\frac{d}{dt}\frac{\partial z}{\partial k_4} = k_1 \left(\frac{x(t)}{C_1} - 1\right) \frac{\partial z(t)}{\partial k_4} + k_1 \frac{1}{C_1}\frac{\partial x(t)}{\partial k_4} z(t) - k_4 \frac{\partial z(t)}{\partial k_4} + z(t) \tag{3.19}$$

$$\frac{du(t)}{dt} = -k_3 \left(1 - \frac{y(t)}{C_2}\right) u(t) + k_4 z(t) \tag{3.20}$$

$$\frac{d}{dt}\frac{\partial u}{\partial k_1} = k_3 \left(\frac{y(t)}{C_1} - 1\right) \frac{\partial u(t)}{\partial k_1} + k_3 \frac{1}{C_1}\frac{\partial y(t)}{\partial k_1} u(t) + k_4 \frac{\partial z(t)}{\partial k_1} \tag{3.21}$$

$$\frac{d}{dt}\frac{\partial u}{\partial k_2} = k_3\left(\frac{y(t)}{C_1} - 1\right)\frac{\partial u(t)}{\partial k_2} + k_3\,\frac{1}{C_1}\frac{\partial y(t)}{\partial k_2}u(t) + k_4\frac{\partial z(t)}{\partial k_2} \tag{3.22}$$

$$\frac{d}{dt}\frac{\partial u}{\partial k_3} = \left(\frac{y(t)}{C_2} - 1\right)u(t) + k_3\left(\frac{y(t)}{C_1} - 1\right)\frac{\partial u(t)}{\partial k_3} + k_3\,\frac{1}{C_1}\frac{\partial y(t)}{\partial k_3}u(t) + k_4\frac{\partial z(t)}{\partial k_3} \tag{3.23}$$

$$\frac{d}{dt}\frac{\partial u}{\partial k_4} = k_3\left(\frac{y(t)}{C_1} - 1\right)\frac{\partial u(t)}{\partial k_4} + k_3\,\frac{1}{C_1}\frac{\partial y(t)}{\partial k_4}u(t) + k_4\frac{\partial z(t)}{\partial k_4} + z(t) \tag{3.24}$$

3.2 Multi-Parameter Sensitivity Analysis (MPSA)

The MPSA method was suggested by Hornberger (1981) and Chang (1992) and further developed by Choi et al. (1998) in the field of hydrology. MPSA is a tool that can be used to define the relative importance of the factors related to the model (Cho, 2003). The idea of MPSA is to insert uncertainty of the parameters into the model by randomly selecting parameter values from probability distributions rather than using fixed values. This is achieved by using a Monte-Carlo method, in which the model is run repeatedly using sets of parameters drawn randomly from the distributions. The selected sets of parameters must however, influence the dynamics. Since the natural distributions of parameter values for actual biological systems are unknown, we use uniform probability distribution (Alves, 2000). The ranges of the parameter distributions are usually determined from the available literature. The dynamics must be deferentially influenced by the variation of each parameter (Cho, 2003). Next, a criterion is introduced into the algorithm to classify the output of each model simulation as either acceptable or unacceptable. The final step of MPSA is a statistical evaluation of the occurrences of the acceptable and unacceptable cases, summarized for each parameter. The larger the difference between the cumulative distributions of the two cases, the more significant the given parameter is (Zi, 2005). The detailed procedure of the MPSA is described as follows:

- Step1: Select the parameters to be tested.
- Step 2: Set the range of each parameter large enough to cover all possible variations. For the sickle cell model, we set the range between one half of the nominal value and two times the nominal value since roughly speaking the rates may be doubled or halved when the temperature changes within ten degrees.
- Step 3: Generate a series of independent random numbers with uniform distribution within the range.
- Step 4: Simulate the model for each parameter and calculate the corresponding objective function (See below for definition of objective function).
- Step 5: Determine whether the chosen parameter values are acceptable or unacceptable by comparing the objective function value to a given criterion. If the objective function is greater than the criterion, the set of parameter values is classified as unacceptable; otherwise, it is acceptable. Here, the criterion is chosen as the average of the objective functions for the entire range of parameters.
- Step 6: Evaluate the parameter sensitivity by comparing the two cumulative distribution functions (CDFs) of the parameter values associated with the acceptable and the unacceptable results. The Kolmogorov-Smirnov (K-S) test is employed and the K-S distance between the two CDFs is calculated. The larger the distance is, the more sensitive the parameter is. If the K-S distance is small, which means the two CDFs are similar to each other, then the parameter is classified as insensitive.

The objective value is defined as the sum of squared differences between the output values from the sampling parameters and the output values from the nominal parameters:

$$f_{obj}(k) = \sum_{i=1}^{q} \left(f_{nominal}(i) - f_{sampling}(i,k)\right)^2 \qquad (3.25)$$

where $f_{obj}(k)$ is the objective function that describes how much the system output with the sampling parameters changes from the data with the nominal parameters; $f_{nominal}(i)$ denotes an output value from the nominal parameters at the i^{th} time; $f_{sampling}(i,k)$ denotes the output value from the sampling parameter k at the i^{th} time; and q is the number of time point. The Matlab codes for MPSA are included in Appendix B.

CHAPTER 4

Results

We utilize the TSA and the MPSA to examine the sensitivities of the variables that are presented in our model with respect to the changes in the parameters values. The output variables are ($C_m^d(t)$, $C_m^{co}(t)$, $C_p^d(t)$, $C_p^{co}(t)$), which are represented in our results by the variables $(x(t), y(t), z(t), u(t))$ respectively. The input parameters are (k^d, $k_m[co]$, $k^{co}, k_p[co]$), which are represented in our results by the parameters (k_1, k_2, k_3, k_4) respectively.

The nominal values and the ranges of the parameters are presented in Table 1.The initial conditions of the system are presented in Table 2. For the simulations in this study, we use the Runge-Kutta method for the time-marching integration of the ODEs, with time step dt=0.01 and the time interval [0 400].

Figure 6 describes the behavior on the variables of the model over time. The graph shows that the de-oxy polymers and de-oxy monomers population dies off after a period of time. The population of CO-bound polymers and CO-bound monomers increases and stabilizes after a period of time.

Table 1

Summary of the nominal value and range of variation for binding/melting rates

Parameters	Nominal values	Range of variation
$\mathrm{k}^{\mathrm{d}}(k_1)$	0.028	0.014 – 0.056
$\mathrm{k}_{\mathrm{m}}(k_2)$	0.07	0.035 – 0.14
$\mathrm{k}^{\mathrm{co}}(k_3)$	0.1	0.05 – 0.2
$\mathrm{k}_{\mathrm{p}}(k_4)$	0.01	0.005 – 0.02

Table 2

The initial conditions for the numerical simulations

Variables	Values
$\mathrm{C}_{\mathrm{m}}^{\mathrm{d}}(0), x(0)$	0.0036
$\mathrm{C}_{\mathrm{m}}^{\mathrm{co}}(0), y(0)$	0
$\mathrm{C}_{\mathrm{p}}^{\mathrm{co}}(0), z(0)$	0
$\mathrm{C}_{\mathrm{p}}^{\mathrm{d}}(0), u(0)$	0.175

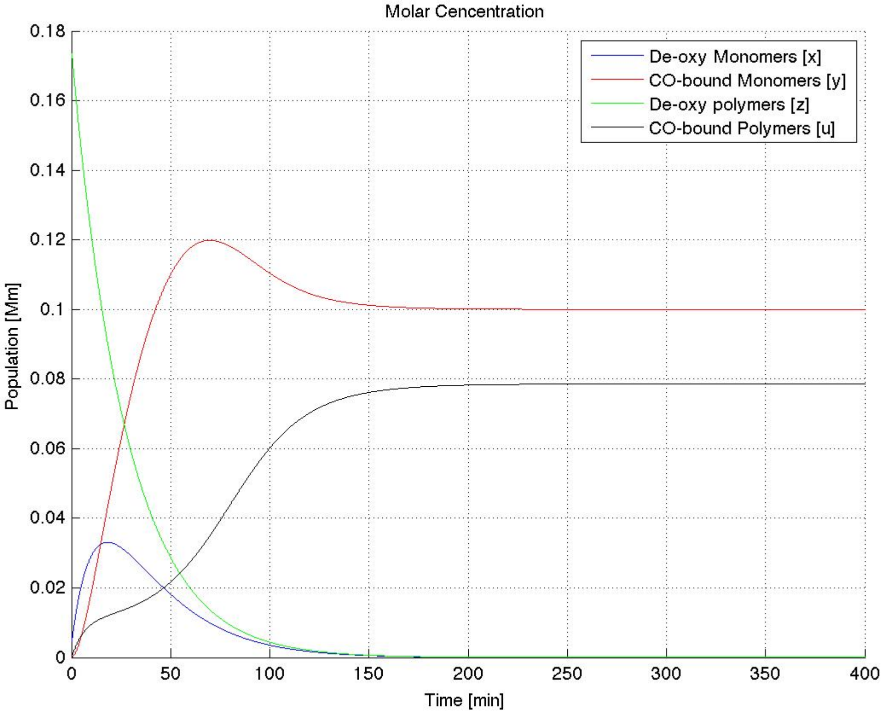

Figure 6. Solution behavior of the model

4.1 Results from TSA

4.1.1 TSFs with parameters at nominal values. Figures 7—10 illustrate the behavior of the variables with respect to the parameters over time t. To better estimate the sensitivity, we rank the relative sensitivities with respect to the parameter. For this analysis, we use the L^2 norm to compare these TSFs. Note this norm is defined as (David, 2007)

$$\|f\|_2^2 = \int_a^b f^2(t)dt \qquad (4.1)$$

Since we are approximating the solution to our sensitivity $\frac{\partial X}{\partial P}$, we also approximate this norm. The L^2 norms of the TSFs are shown in Table 3.

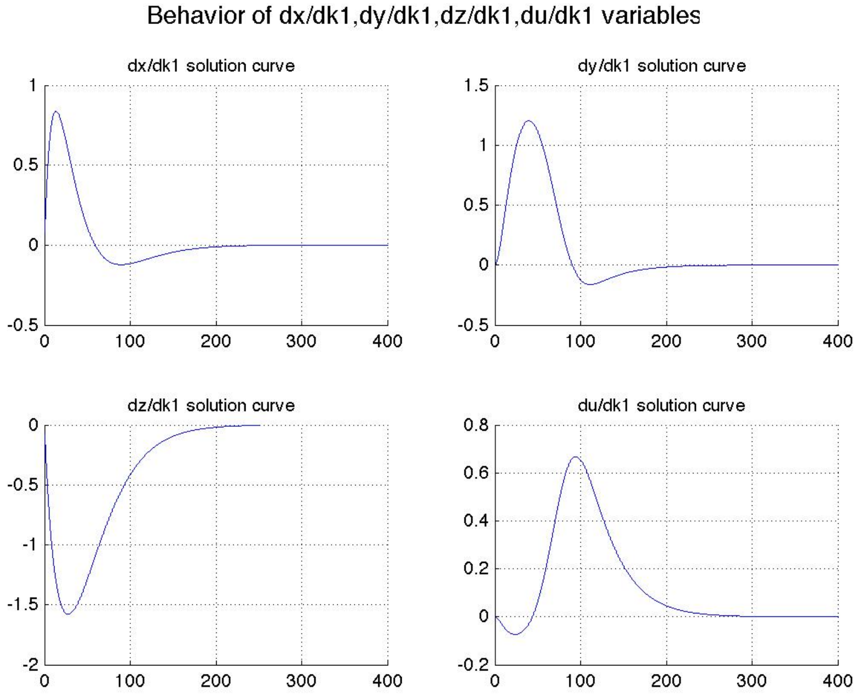

Figure 7. Sensitivity of the variables with respect to k_1

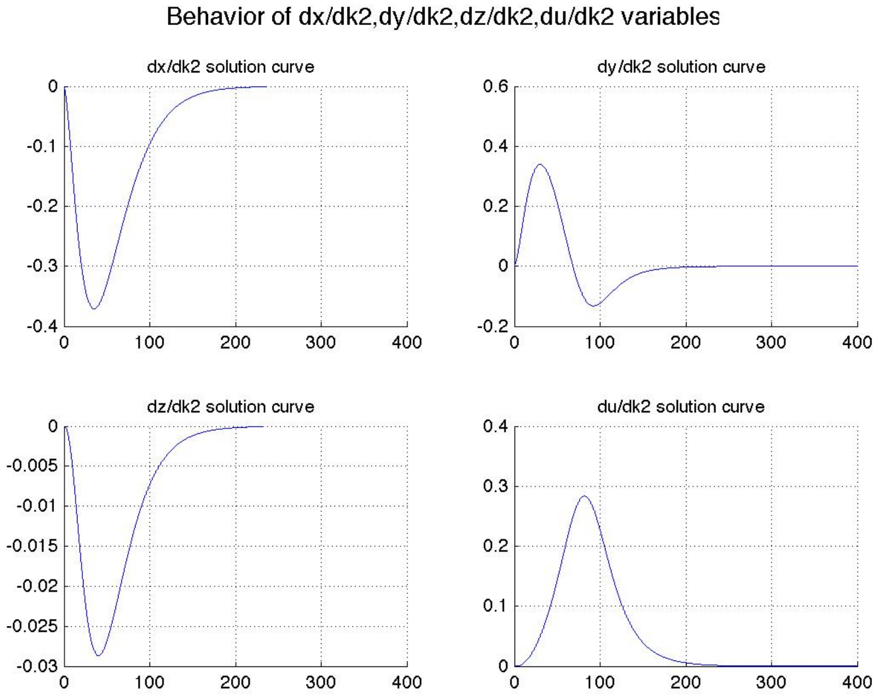

Figure 8. Sensitivity of the variables with respect to k_2

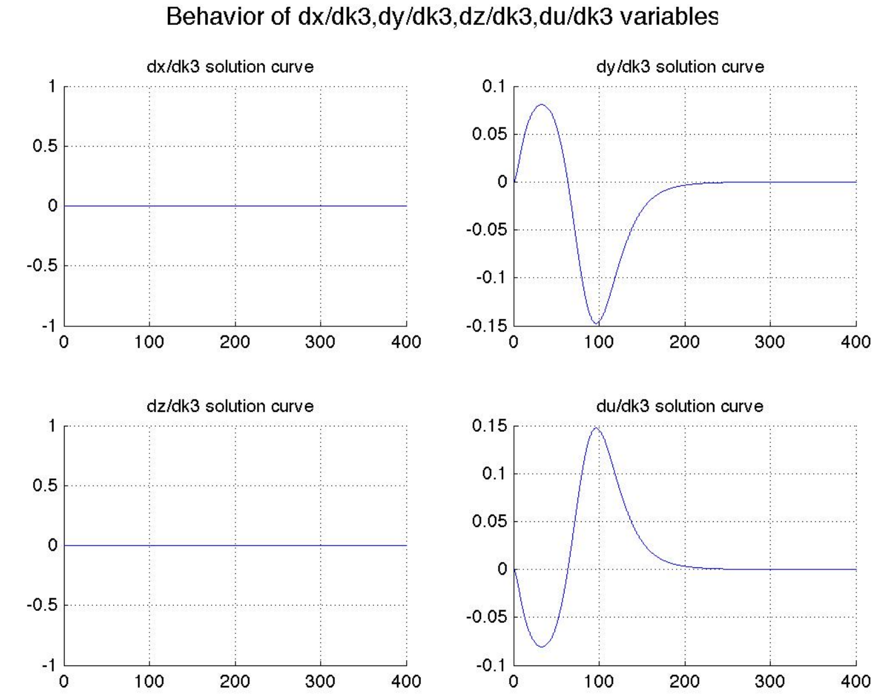

Figure 9. Sensitivity of the variables with respect to k_3

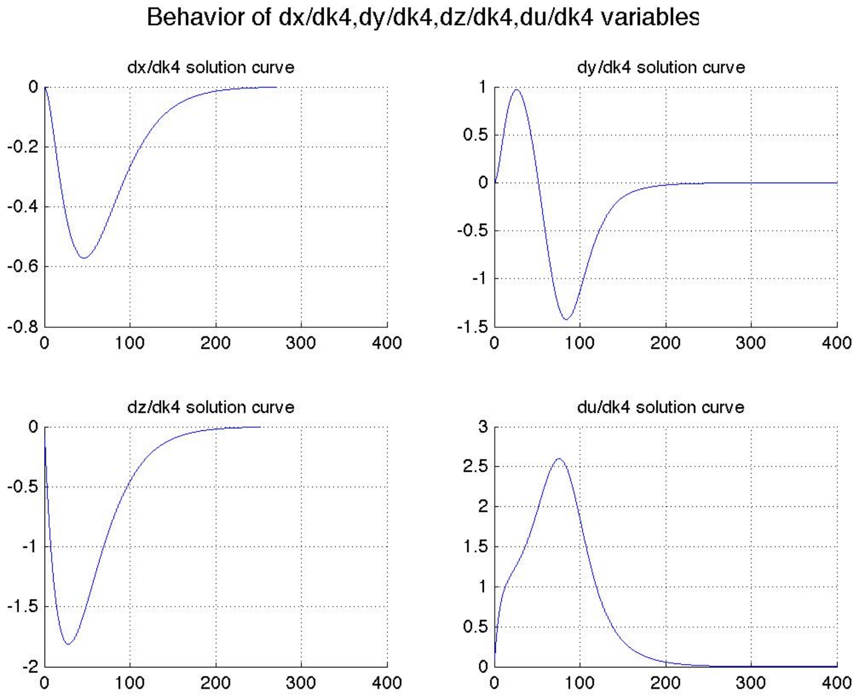

Figure 10. Sensitivity of the variables with respect to k_4

Table 3

Summary of the L^2norms of the TSFs

Variables/Parameters	k_1	k_2	k_3	k_4
x	41.5601	25.9384	0	44.3415
y	77.1920	20.7210	10.3865	100.7236
z	113.7541	1.9437	0	128.8967
u	47.7038	20.0600	10.3865	203.1208

Based on the L^2norm results, we notice the dominance of the sensitivities of the variables with respect to k_4 among all the others parameters. The sensitivity of $\frac{\partial u}{\partial k_4}$ has reached a magnitude of 203.1208 as Table 3 has shown. This indicates that there is a major effect on the variable u. This sort of magnitude for a sensitivity function has not been seen in any other sensitivity functions.

The sensitivity of $\frac{\partial z}{\partial k_4}$ has the highest magnitude, 128.8967, among the parameters that can possibly affect z. k_4 dominates the effect caused by the parameters. The variable z is secondly most affected by k_1. $\frac{\partial z}{\partial k_1}$ has a magnitude of 113.7541. It should be noted that the magnitude of the sensitivity $\frac{\partial z}{\partial k_3}$ remains 0 over the entire time length. This indicates that there is no effect of parameter k_3 over the variable z. The sensitivity $\frac{\partial z}{\partial k_2}$ has a magnitude of 1.9437, which shows that there is an insignificant effect of k_2 on the variable z.

The sensitivity $\frac{\partial y}{\partial k_4}$ has a magnitude of 100.7236. Thus, the variable k_4 has the dominant effect in the variable y. The variable y is also affected by k_1. We notice a magnitude of 77.1920 for $\frac{\partial y}{\partial k_1}$. The parameters k_2 and k_3 appear to have little effect on the variable y.

As of for the output variable x, k_4 shows dominance with a magnitude of 44.3415 for the sensitivity $\frac{\partial x}{\partial k_4}$. But this magnitude is not far from $\frac{\partial x}{\partial k_1}$ whose magnitude is 41.5601. Here, we also notice the magnitude of the sensitivity $\frac{\partial x}{\partial k_3}$ remains 0 over the entire time length indicating that k_3 has no effect over the variable x.

It should be noted that all the sensitivities converge asymptotically to 0 over a period of time.

4.1.2 TSFs when Varying all Parameter Values. To further analyze the behavior of the sensitivity, we vary the value of the parameters in a defined range. The range of the each parameter is listed in Table 1. Figures 11 – 15 display the resultant graphs of the sensitivity functions when all k's are varied. Figure 11 displays the behavior of the variables x, y, z and u.

The behavior of the graphs is more or less following a similar pattern due to the increase or the decrease of the k's values.

In Figures 12 – 15, there are some inconsistencies in the patterns in some subplots. Thus, there is not a clear understanding of the ranking of the sensitivities. We cannot determine the exact ranking of the parameter sensitivities. Therefore, we semi-normalize the TSFs. The normalization is a process that is used to eliminate redundancy, reduce the potential for anomalies during data processing and maintain the consistency and integrity of the data.

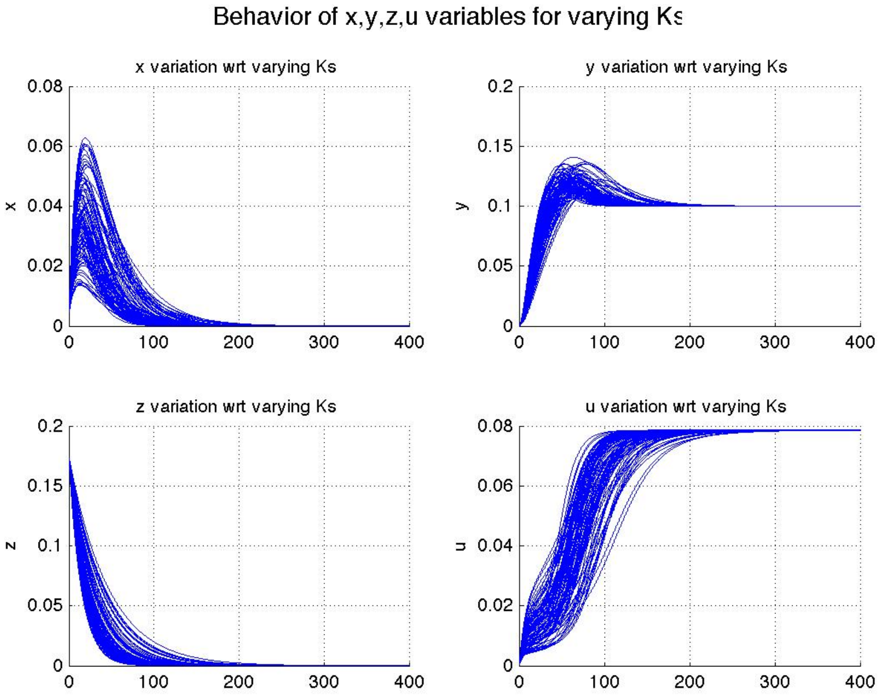

Figure 11. Behavior of the variables as all k's vary

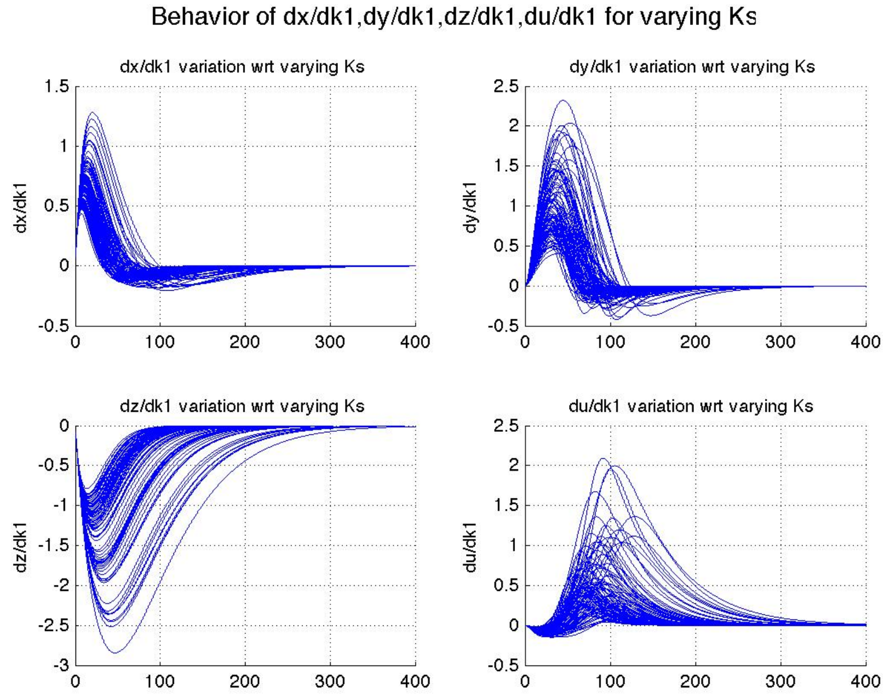

Figure 12. Sensitivity functions of the variables with respect to k_1 as all k's vary

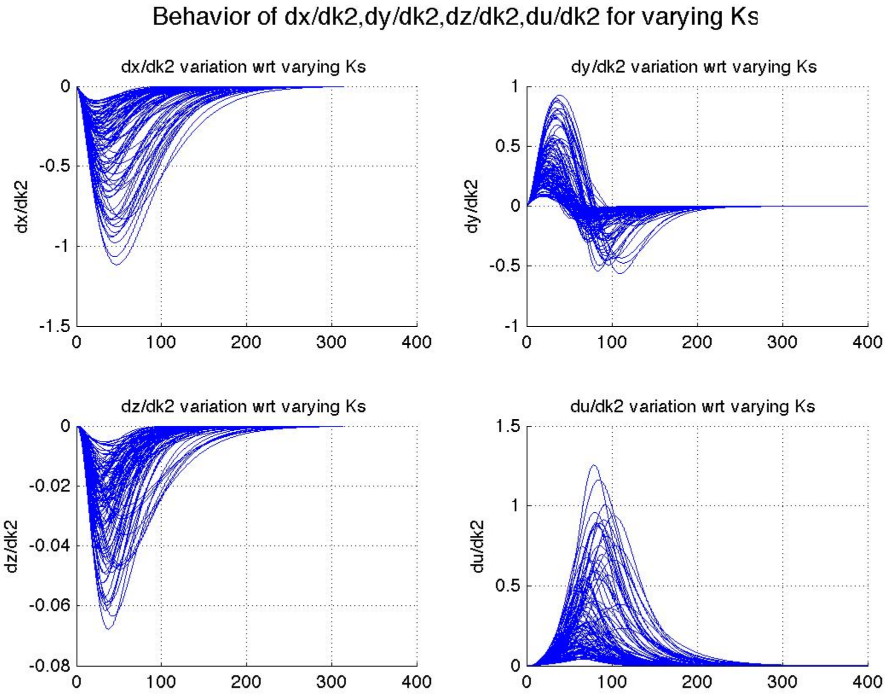

Figure 13. Sensitivity functions of the variables with respect to k_2 as all k's vary

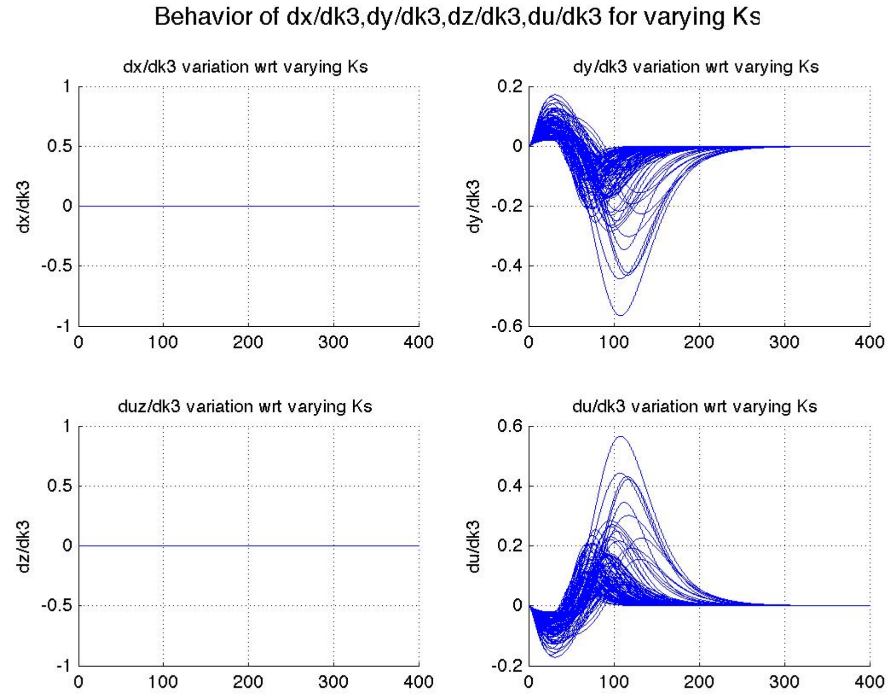

Figure 14. Sensitivity functions of the variables with respect to k_3 as all k's vary

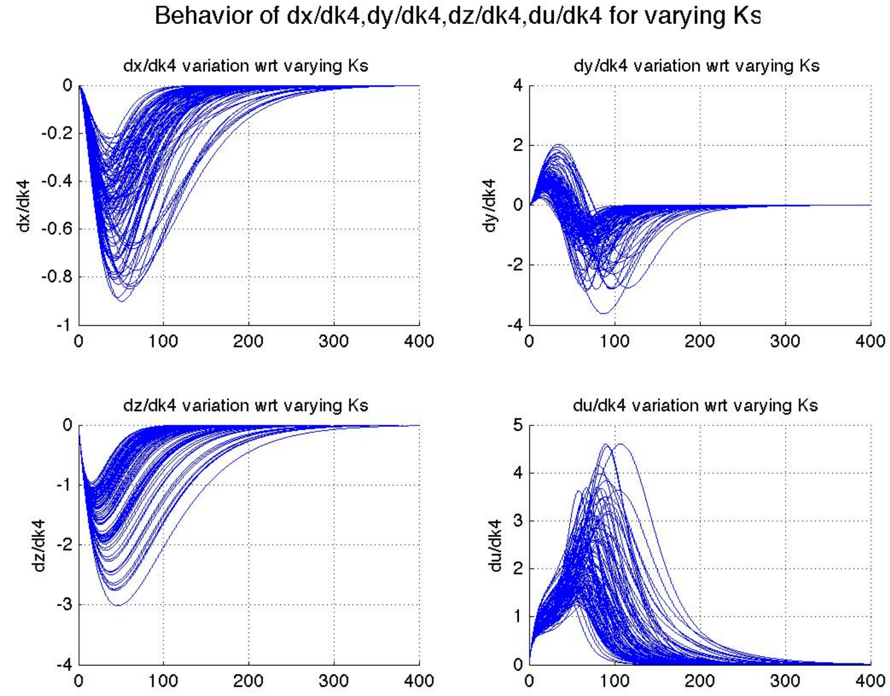

Figure 15. Sensitivity functions of the variables with respect to k_4 as all k's vary

The L^2 norm of the TSF with the sampling parameters referenced to the TSF with the nominal parameters is given as

$$\left\|\frac{\partial X}{\partial P}|_{P=P_{sample}} - \frac{\partial X}{\partial P}|_{P=P_o}\right\|_2^2 = \sum_i \left|\frac{\partial X(t_i)}{\partial P}|_{P=P_{sample}} - \frac{\partial X(t_i)}{\partial P}|_{P=P_o}\right|. \tag{4.2}$$

where $\mathrm{X} = \{x,\ y, z,\ u\}',\ \ P = \{k_1, k_2, k_3, k_4\}'$.

The semi-normalizer consists of multiplying the norm by the distance between the parameters. It is defined in Equation (4.3).

$$|P_{sample} - P_o| \left\|\frac{\partial X}{\partial P}|_{P=P_{sample}} - \frac{\partial X}{\partial P}|_{P=P_o}\right\|_2 \tag{4.3}$$

Figure 16 displays the behavior of the variables *x, y, z* and *u*, where the behavior of the variables at different sampling points is referenced to the behavior of the variables at the nominal value. Here, we also note the inconsistency in the graph except for the variable *z.* The ranking of the semi-normalization norm for each TSF is shown in Table 4. In Table 4, the values are determined by computing the difference between the lowest and highest magnitudes of each TSF. Thus, the larger the difference between the magnitudes of the TSFs of a variable is, the more sensitive the parameter is.

Table 4

Ranking results of the sensitivity functions as all k's vary

Variables/Parameters	k_1	k_2	k_3	k_4
x	0.6195	0.9640	0	0.0772
y	1.0130	0.8312	0.4162	0.7778
z	1.5828	0.0025	0	0.7172
u	1.0570	0.9943	0.4162	1.9409

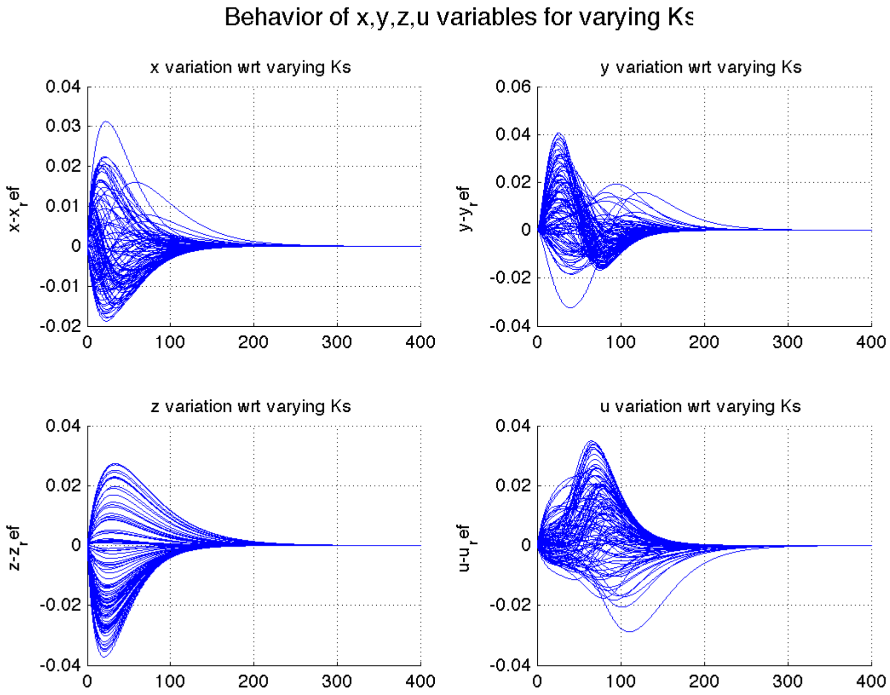

Figure 16. Behavior of the variable as all k's vary

4.2 Results from MPSA

We use the random values following the uniform distribution for the sampling of parameters k_1, k_2, k_3, k_4. The sample size is 10,000 points. In another word, we take random values within the ranges of k_1, k_2, k_3, k_4 10,000 times and simulate the differential equations (DEs) system 10,000 times. Six (6) different criterions including, the 25th, 37.5th, 50th, 62.5th, 75th and the average of the objective function values are chosen in the computation of the probability mass functions (PMFs) and the cumulative distribution functions (CDFs). Nonetheless, the rank of the sensitivity of each parameter remains the same in each chosen criterion. The results from larger size samplings are essentially identical to the results reported here.

4.2.1 MPSA results for output variable x. The PMFs of acceptance and non-acceptance for the output variable x are presented in the Figures 17 – 20. The cumulative distribution functions of acceptance and non-acceptance for the output variable x are displayed in Figures 21-24. The distributions are tested, and the Kolmogorov-Smirnov distance is computed to rank the sensitivity of each parameter, with the results shown in the Table 5. The p-value attributed to each ranking is showed in Table 6. From these results including the figures and the tables, with the output variable x the ranking for all four parameters sensitivities is k_2, k_1, k_4, k_3.

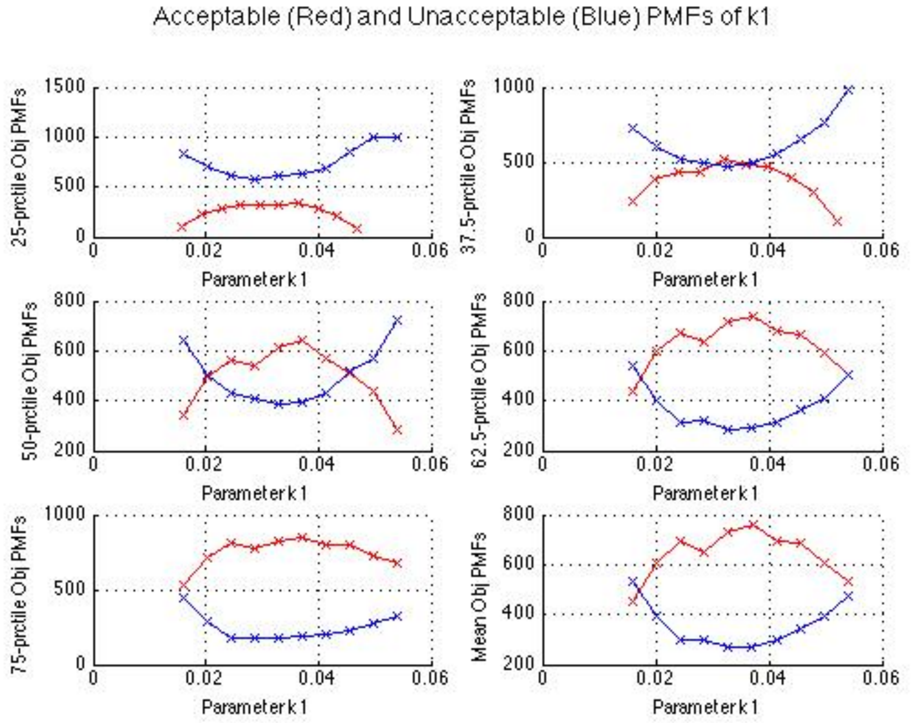

Figure 17. Probability Mass functions of k_1 considering output x

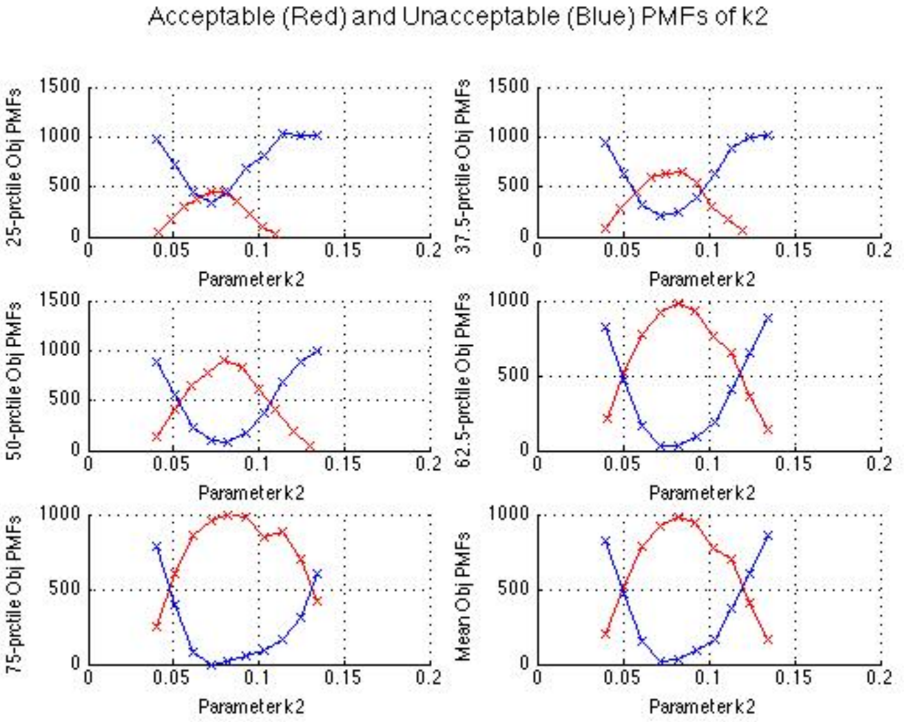

Figure 18. Probability Mass functions of k_2 considering output x

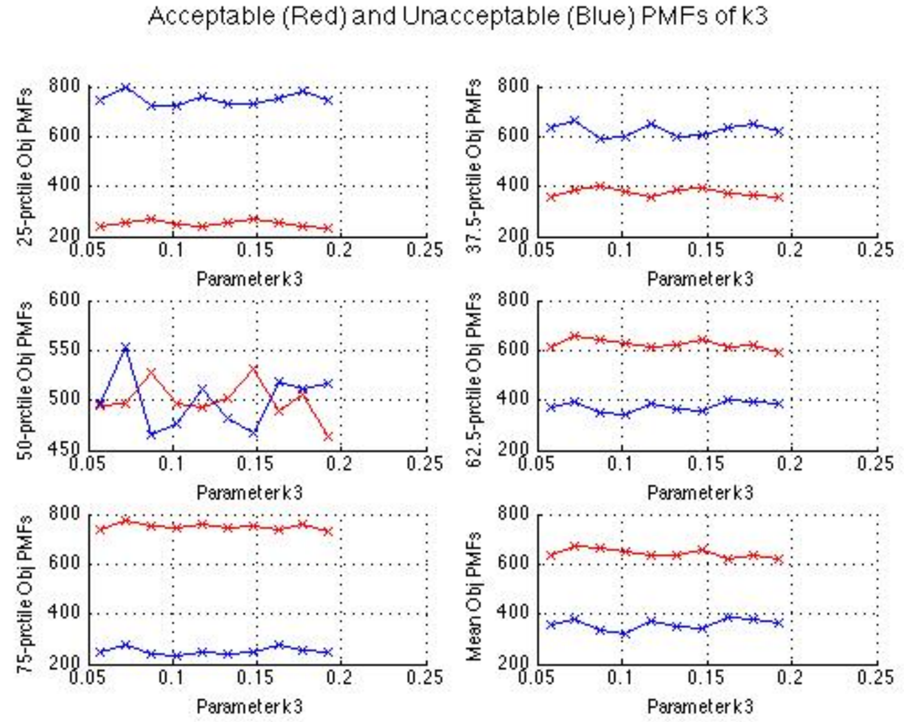

Figure 19. Probability Mass functions of k_3 considering output x

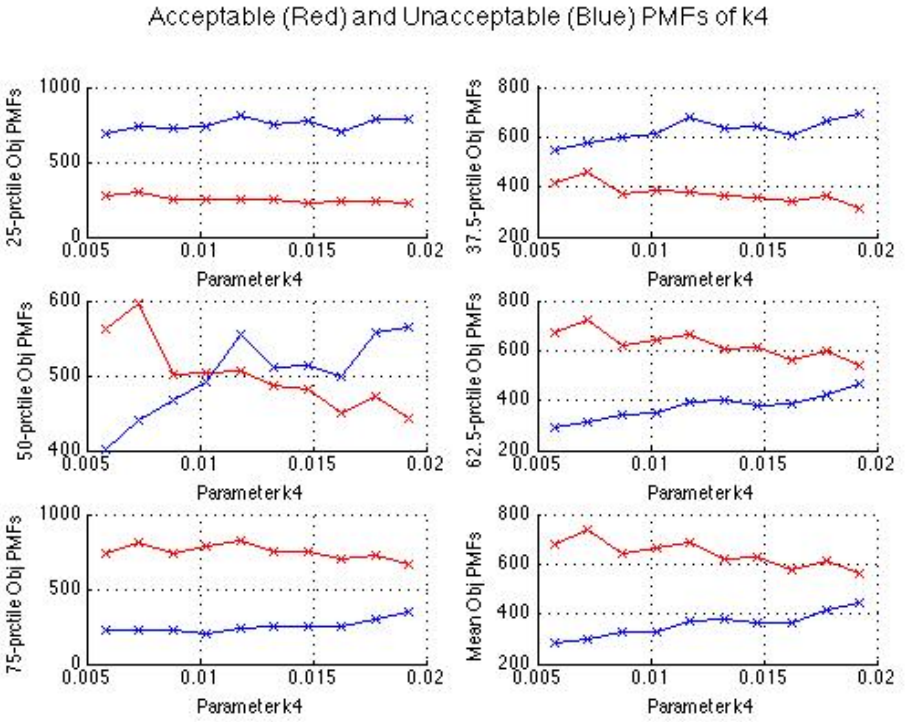

Figure 20. Probability Mass functions of k_4 considering output x

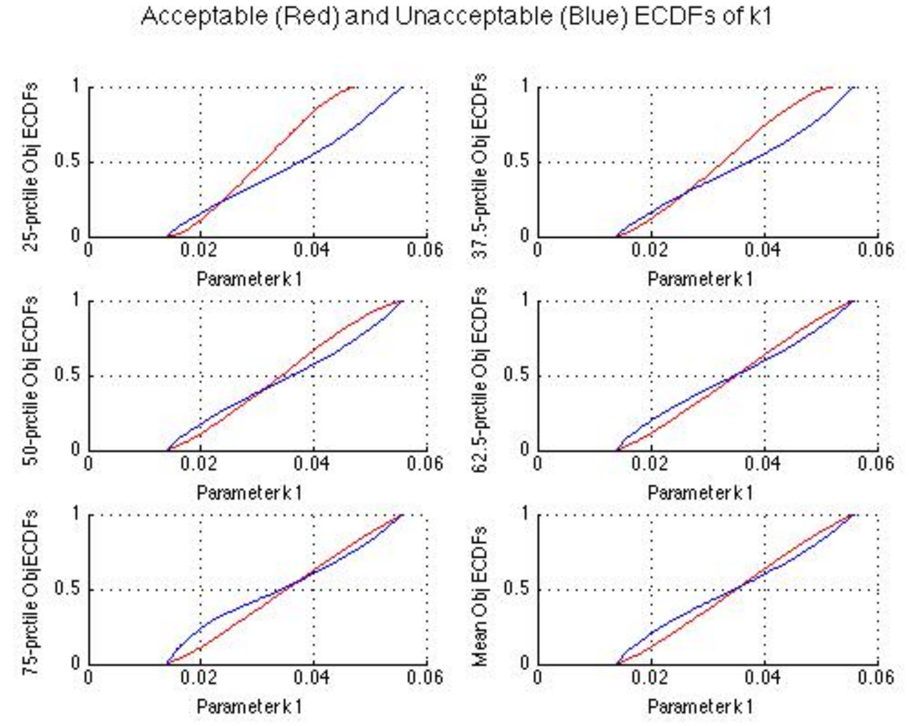

Figure 21. Cumulative distribution functions of k_1 considering output x

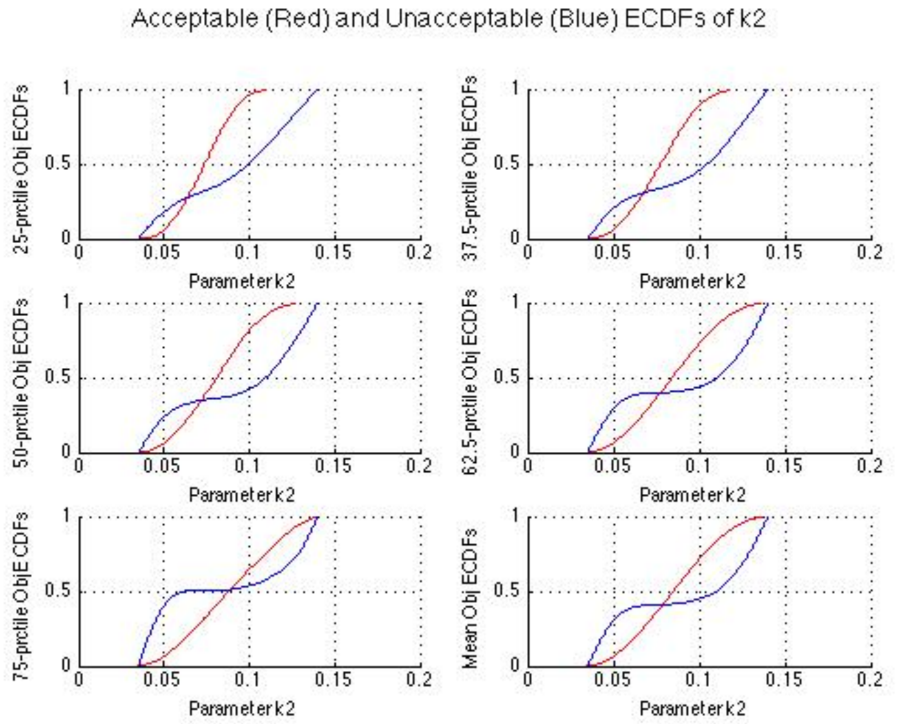

Figure 22. Cumulative distribution functions of k_2 considering output x

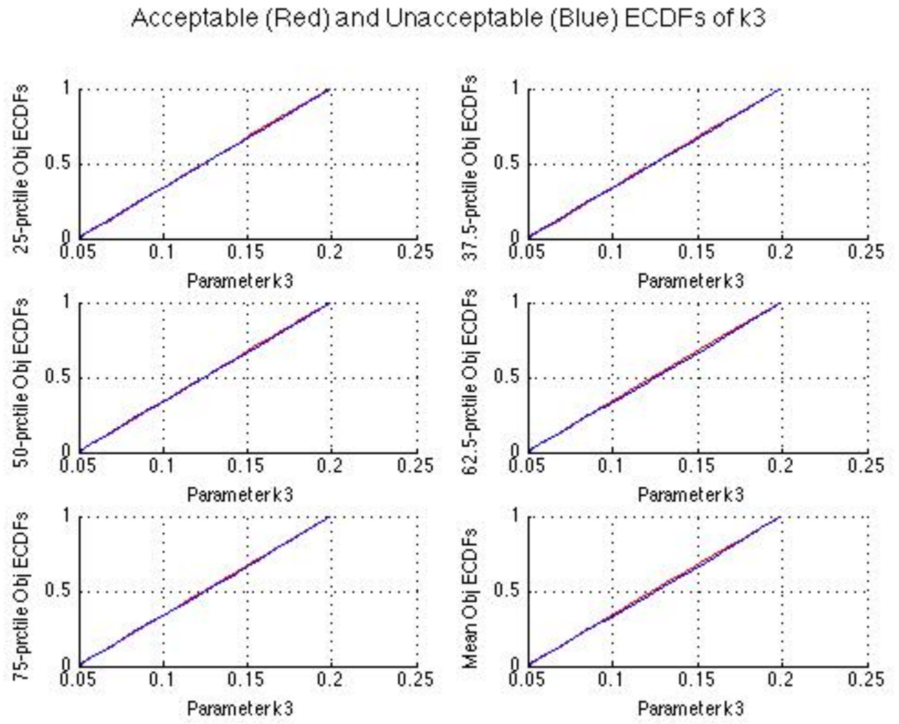

Figure 23. Cumulative distribution functions of k_3 considering output x

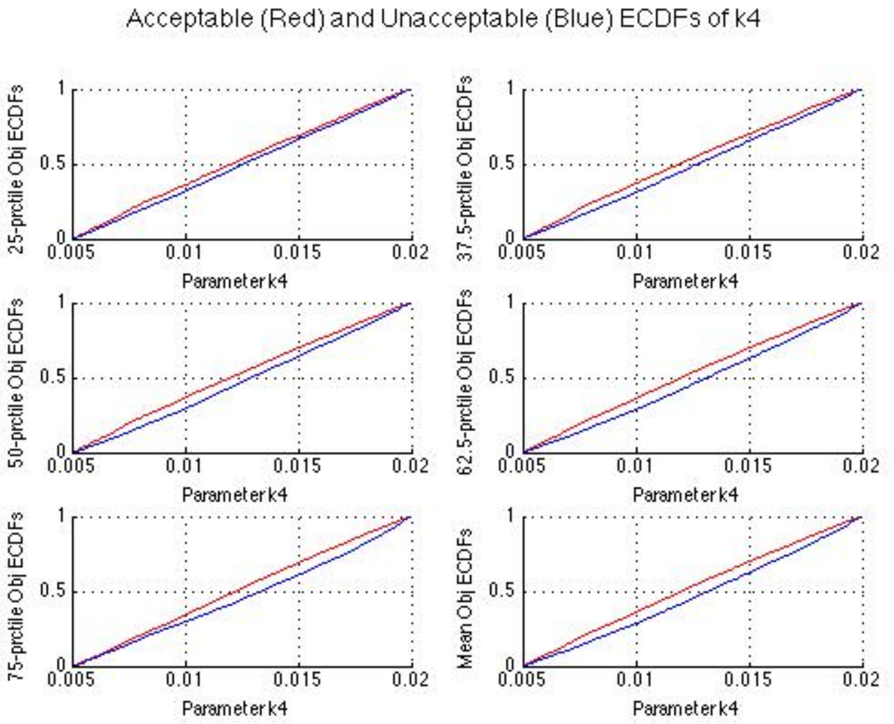

Figure 24. Cumulative distribution functions of k_4 considering output x

Table 5

Rank at 5% significance level of the MPSA results for x

Rank at 5% significance level of the MPSA results for $x, C_m^d(t)$				
Criterions/Parameters	k_1	k_2	k_3	k_4
25th percentile of objectives	0.3056	0.4615	0.0160	0.0468
37.5th percentile of objectives	0.2157	0.4419	0.0173	0.0633
50th percentile of objectives	0.1210	0.4176	0.0188	0.0762
62.5th percentile of objectives	0.0871	0.3555	0.0238	0.0842
75th percentile of objectives	0.1312	0.3580	0.0151	0.0837
Mean of the objectives	0.0942	0.3389	0.0235	0.0870

Table 6

P-values at 5% significance level of the MPSA results for x

P-values at 5% significance level of the MPSA results for $x, C_m^d(t)$				
Criterions/Parameters	k_1	k_2	k_3	k_4
25th percentile of objectives	0.0	0.0	0.7199	0.0005
37.5th percentile of objectives	0.0	0.0	0.4827	0.0
50th percentile of objectives	0.0	0.0	0.3370	0.0
62.5th percentile of objectives	0.0	0.0	0.1374	0.0
75th percentile of objectives	0.0	0.0	0.7854	0.0
Mean of the objectives	0.0	0.0	0.1564	0.0

4.2.2 MPSA results for output variable y. The PMFs of acceptance and non-acceptance for the output variable y are presented in the Figures 25 – 28. The cumulative distribution functions of acceptance and non-acceptance for the output variable y are displayed in Figures 29—32. The distributions are tested, and the Kolmogorov-Smirnov distance is computed to rank the sensitivity of each parameter, with the results shown in the Table 7. The p-value attributed to each ranking is showed in Table 8. From these results including the figures and the tables, with the output variable y the ranking for all four parameters sensitivities is k_1, k_2, k_3, k_4.

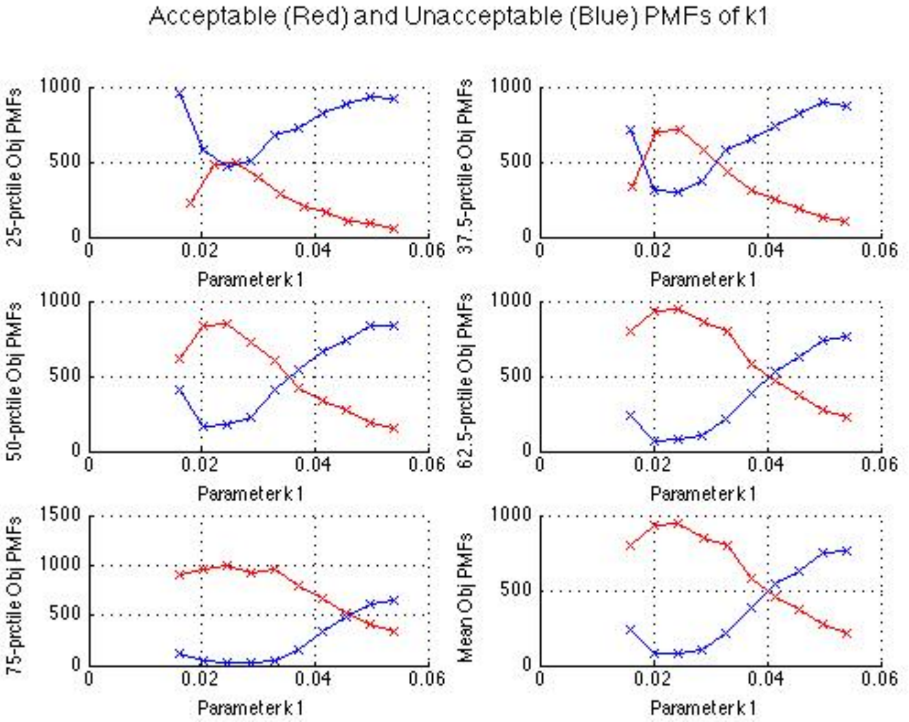

Figure 25. Probability Mass functions of k_1 considering output y

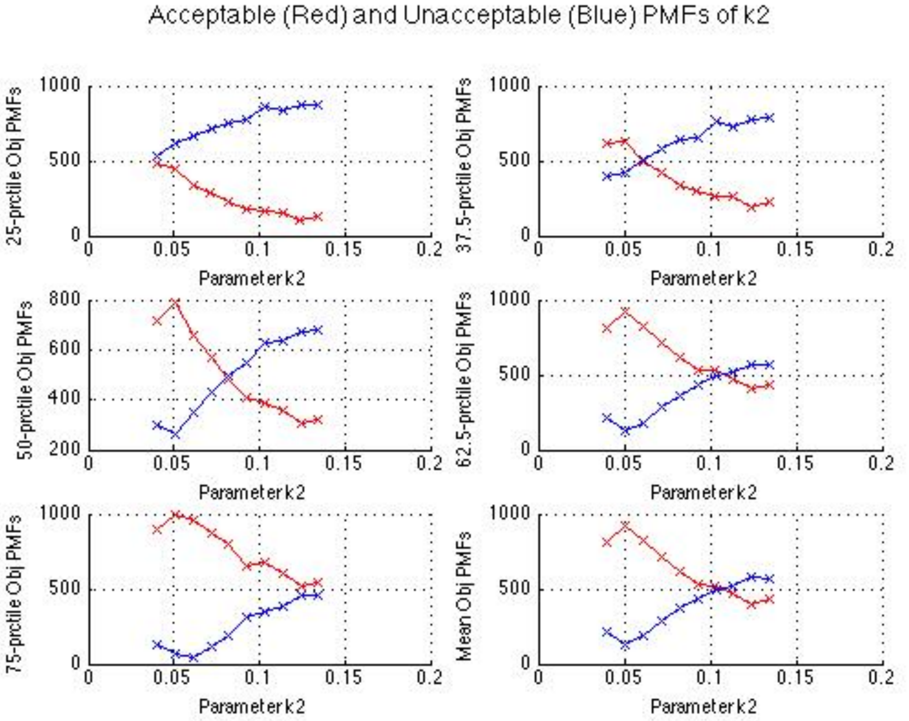

Figure 26. Probability Mass functions of k_2 considering output y

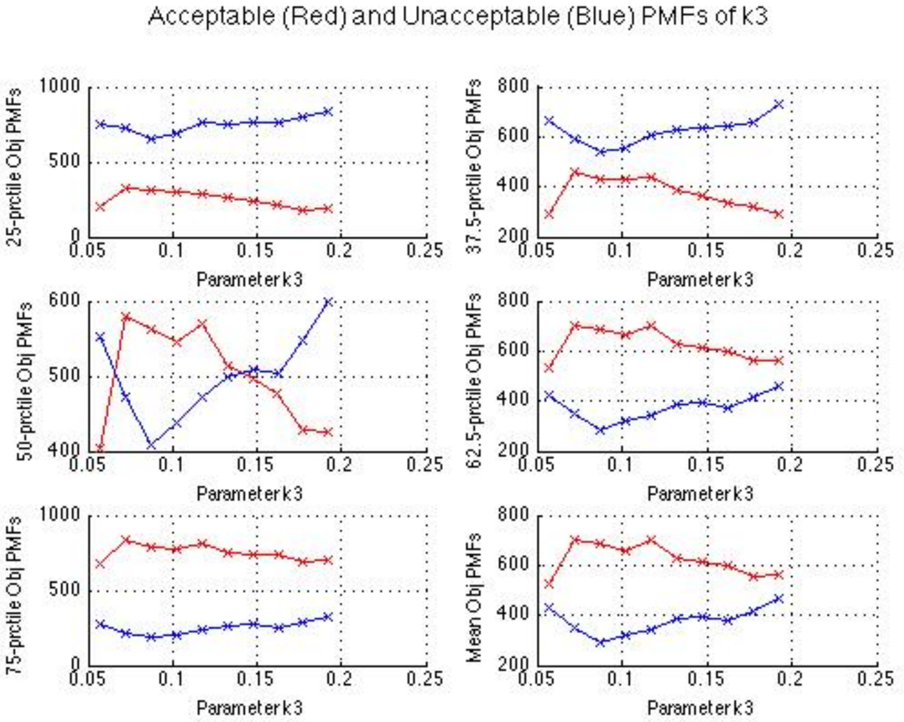

Figure 27. Probability Mass functions of k_3 considering output y

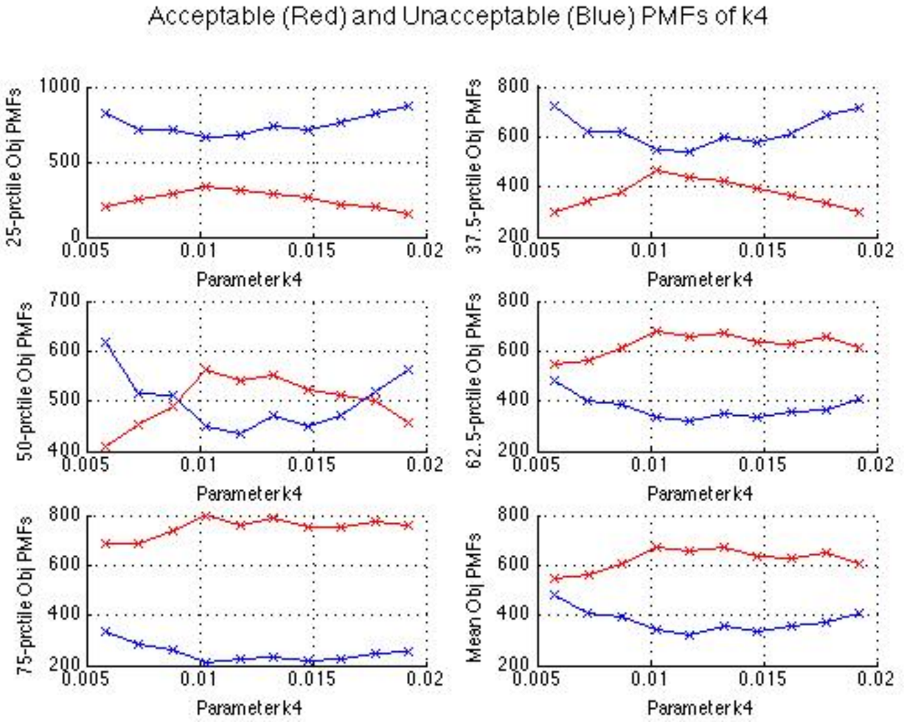

Figure 28. Probability Mass functions of k_4 considering output y

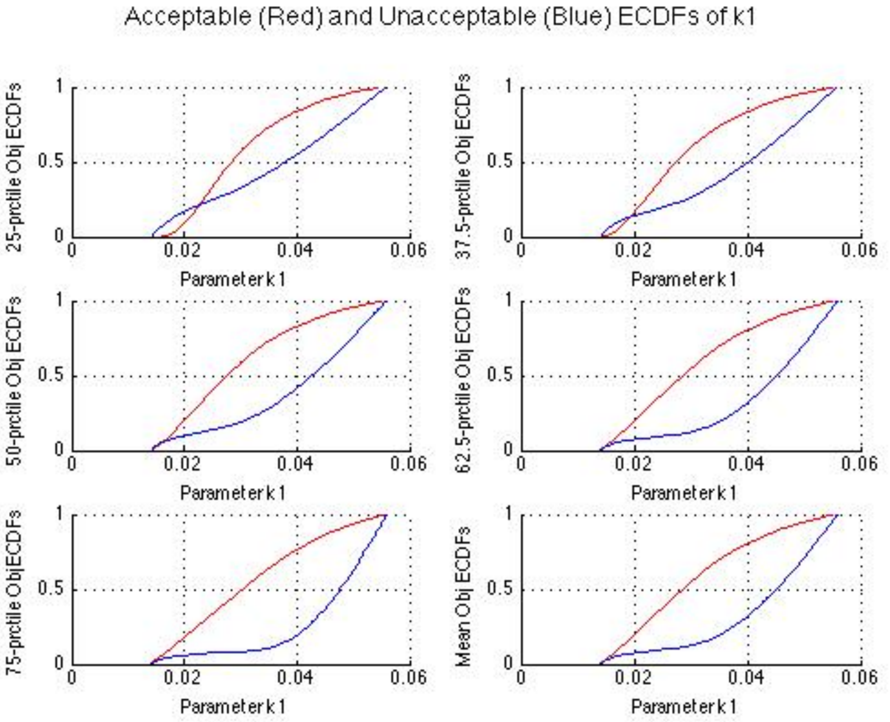

Figure 29. Cumulative distribution functions of k_1 considering output y

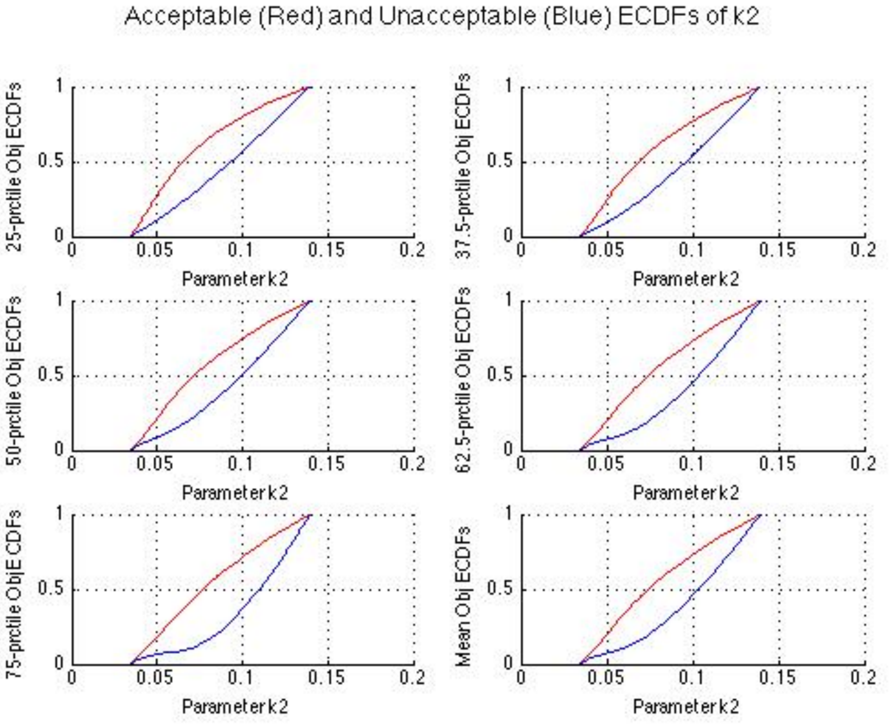

Figure 30. Cumulative distribution functions of k_2 considering output y

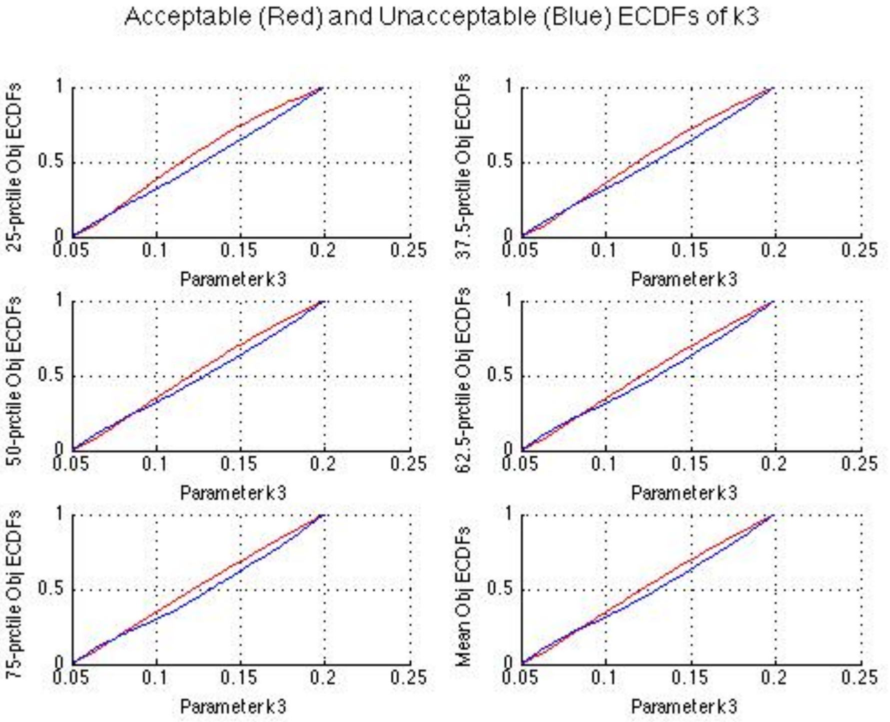

Figure 31. Cumulative distribution functions of k_3 considering output y

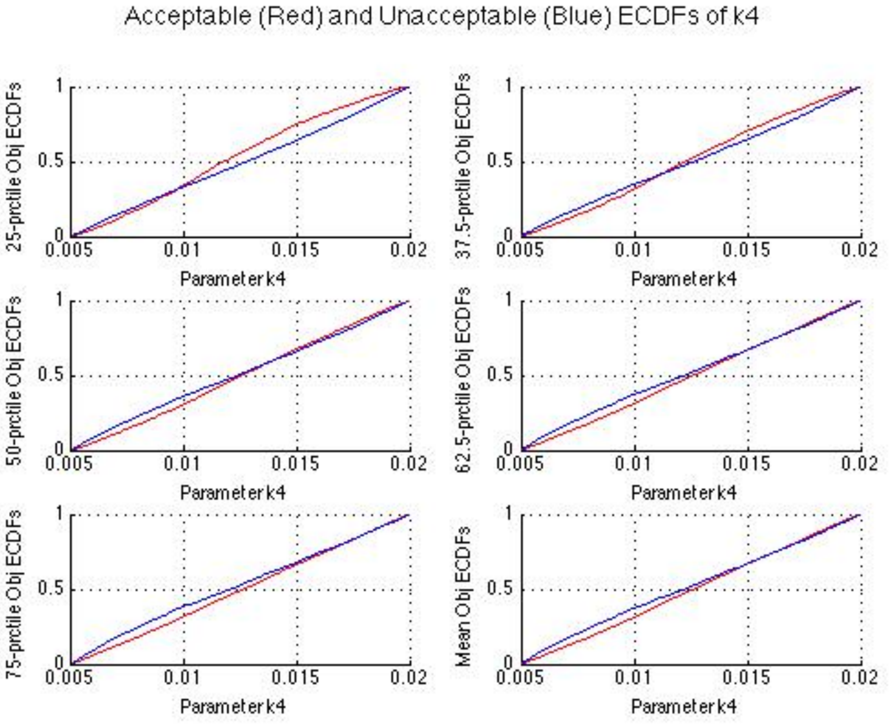

Figure 32. Cumulative distribution functions of k_4 considering output y

Table 7

Rank at 5% significance level of the MPSA results for y

Rank at 5% significance level of the MPSA results for $y, C_m^{co}(t)$				
Criterions/Parameters	k_1	k_2	k_3	k_4
25th percentile of objectives	0.3060	0.2864	0.1005	0.1111
37.5th percentile of objectives	0.3723	0.2804	0.0831	0.0604
50th percentile of objectives	0.4514	0.2844	0.0770	0.0608
62.5th percentile of objectives	0.5103	0.3160	0.0721	0.0660
75th percentile of objectives	0.5757	0.3877	0.0767	0.0744
Mean of the objectives	0.5097	0.3150	0.0730	0.0653

Table 8

P-values at 5% significance level of the MPSA results for y

P-values at 5% significance level of the MPSA results for $y, C_m^{co}(t)$ (1*e^{-07})				
Criterions/Parameters	k_1	k_2	k_3	k_4
25th percentile of objectives	0.0	0.0	0.0	0.0
37.5th percentile of objectives	0.0	0.0	0.0	0.6971
50th percentile of objectives	0.0	0.0	0.0	0.1716
62.5th percentile of objectives	0.0	0.0	0.0005	0.0249
75th percentile of objectives	0.0	0.0	0.0047	0.0172
Mean of the objectives	0.0	0.0	0.0002	0.0354

4.2.3 MPSA Results for Output Variable z. The PMFs of acceptance and non-acceptance for the output variable z are presented in the Figures 33 – 36. The cumulative distribution functions of acceptance and non-acceptance for the output variable z are displayed in the Figures 37—40. The distributions are tested, and the Kolmogorov-Smirnov distance is computed to rank the sensitivity of each parameter, with the results shown in the Table 9. The p-value attributed to each ranking is showed in Table 10. From these results including the figures and the tables, with the output variable z the ranking for all four parameters sensitivities is k_1, k_4, k_2, k_3.

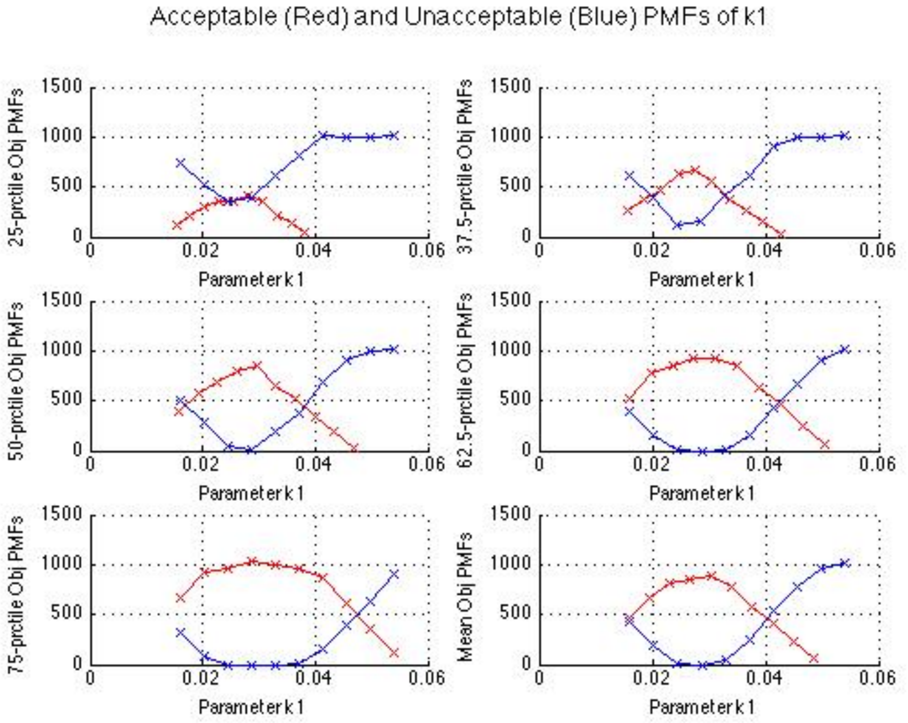

Figure 33. Probability Mass functions of k_1 considering output z

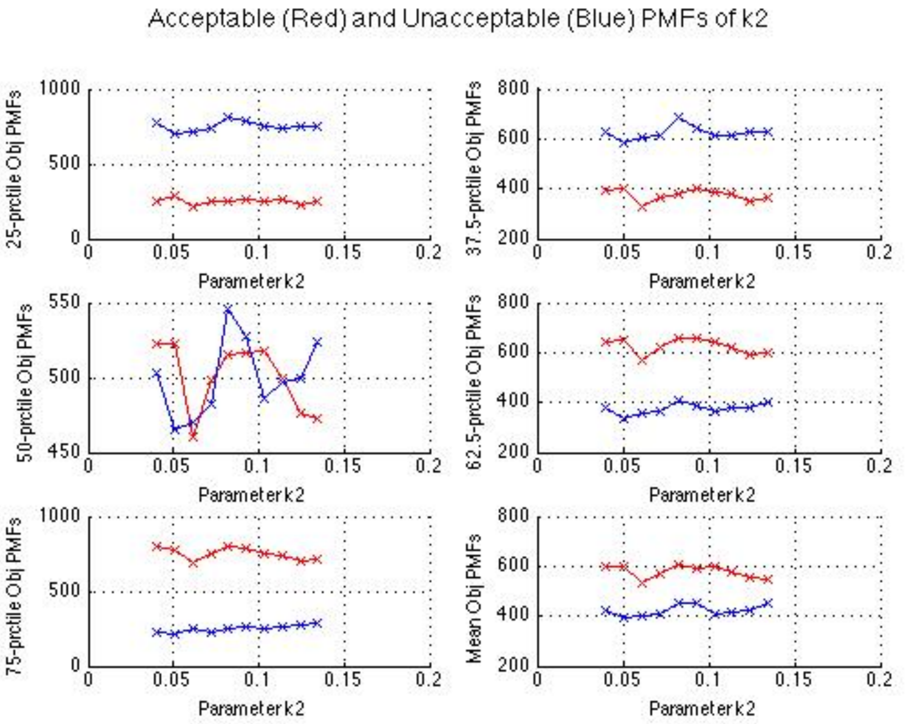

Figure 34. Probability Mass functions of k_2 considering output z

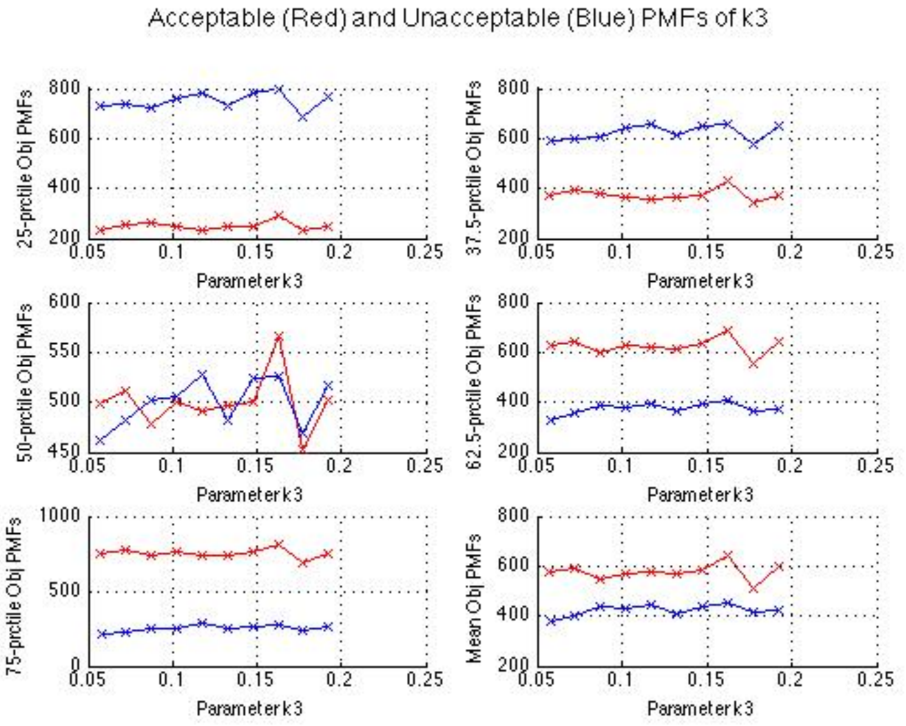

Figure 35. Probability Mass functions of k_3 considering output z

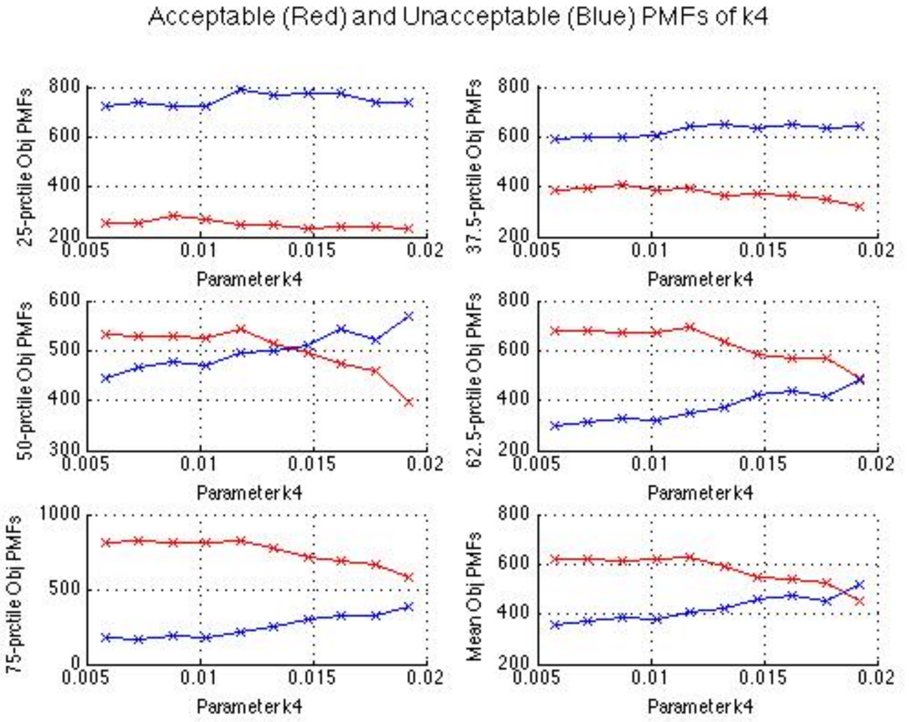

Figure 36. Probability Mass functions of k_4 considering output z

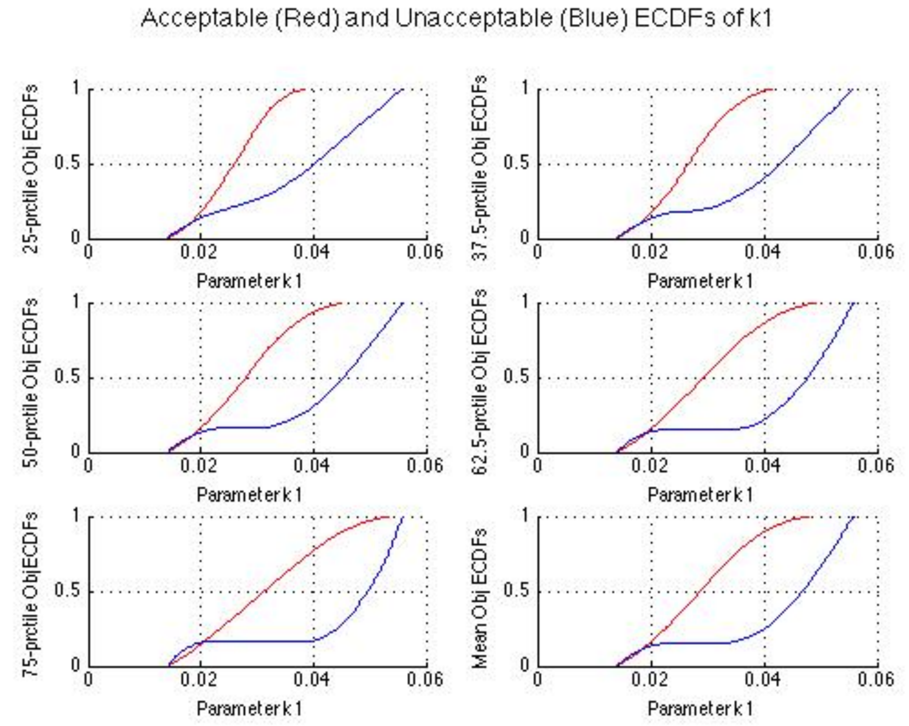

Figure 37. Cumulative distribution functions of k_1 considering output z

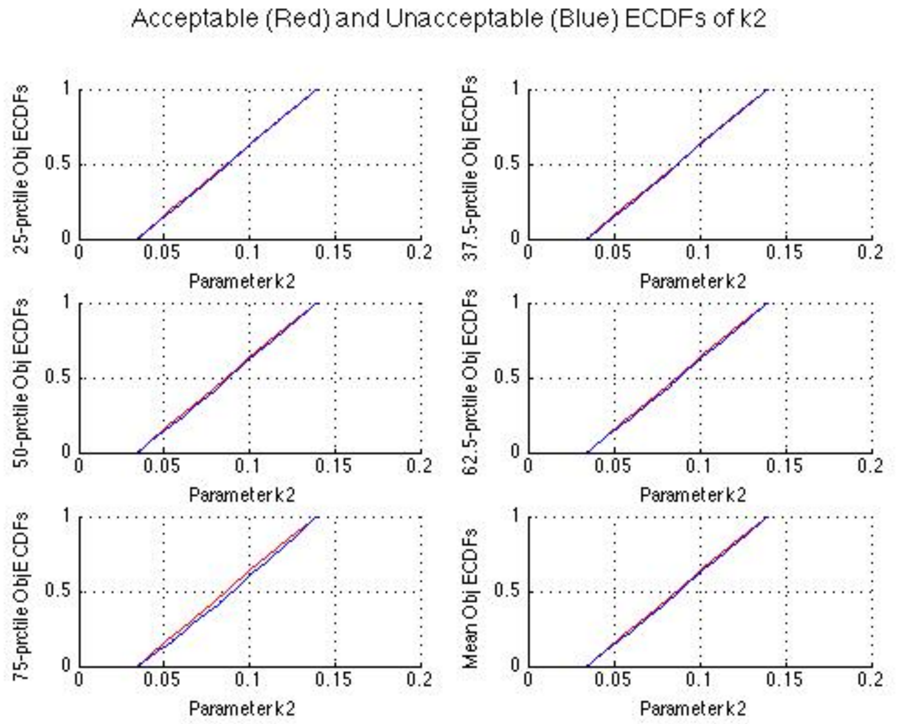

Figure 38. Cumulative distribution functions of k_2 considering output z

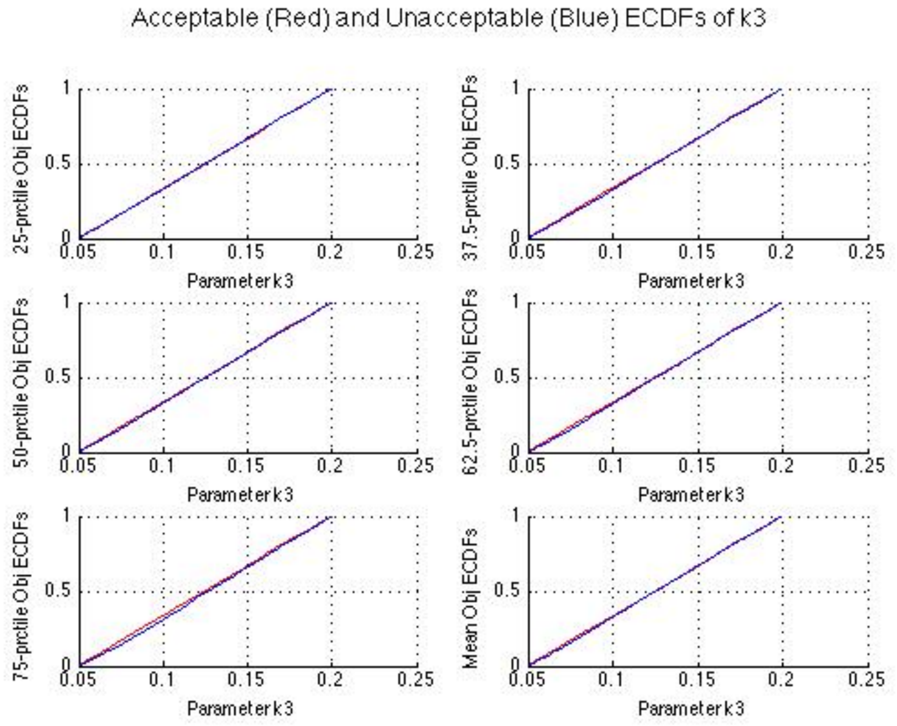

Figure 39. Cumulative distribution functions of k_3 considering output z

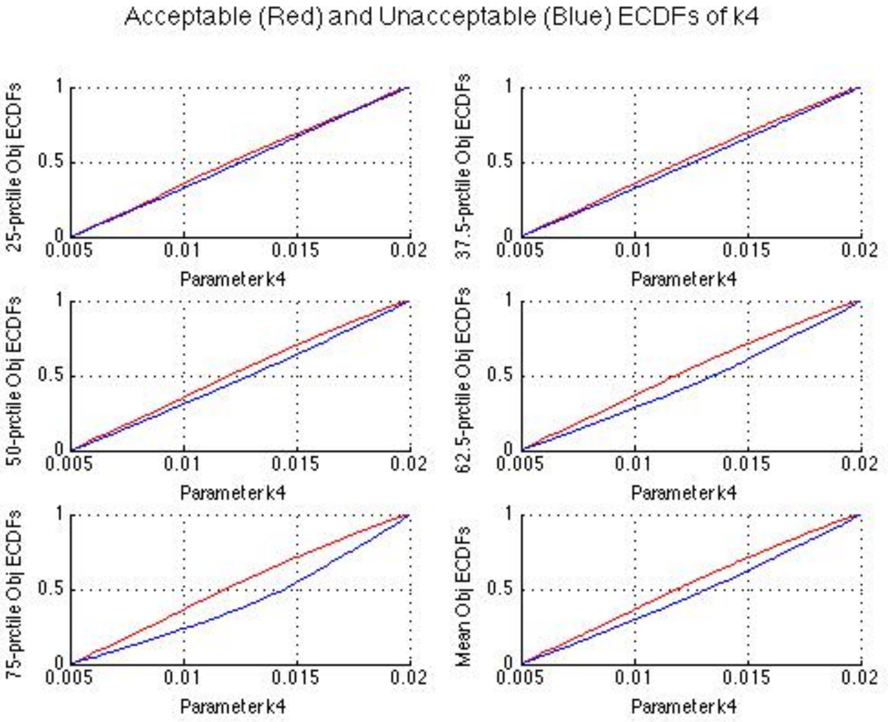

Figure 40. Cumulative distribution functions of k_4 considering output *z*

Table 9

Rank at 5% significance level of the MPSA results for z

Rank at 5% significance level of the MPSA results for $z, C_p^d(t)$				
Criterions/Parameters	k_1	k_2	k_3	k_4
25th percentile of objectives	0.5903	0.0220	0.0107	0.0392
37.5th percentile of objectives	0.6152	0.0210	0.0192	0.0438
50th percentile of objectives	0.6368	0.0238	0.0166	0.0708
62.5th percentile of objectives	0.6466	0.0228	0.0238	0.1186
75th percentile of objectives	0.6316	0.0473	0.0339	0.1831
Mean of the objectives	0.6505	0.0211	0.0205	0.0958

Table 10

P-values at 5% significance level of the MPSA results for z

P-values at 5% significance level of the MPSA results for $z, C_p^d(t)$				
Criterions/Parameters	k_1	k_2	k_3	k_4
25th percentile of objectives	0.0	0.3210	0.9828	0.0061
37.5th percentile of objectives	0.0	0.2493	0.3503	0.0002
50th percentile of objectives	0.0	0.1161	0.4930	0.0
62.5th percentile of objectives	0.0	0.1736	0.1391	0.0
75th percentile of objectives	0.0	0.0004	0.0265	0.0
Mean of the objectives	0.0	0.2243	0.2547	0.0

4.2.4 MPSA Results for Output Variable u. The PMFs of acceptance and non-acceptance for the output variable u are presented in the Figures 41 – 44. The cumulative distribution functions of acceptance and non-acceptance for the output variable u are displayed in the Figures 45 – 48. The distributions are tested, and the Kolmogorov-Smirnov distance is computed to rank the sensitivity of each parameter, with the results shown in the Table 11. The p-value attributed to each ranking is showed in Table 12. From these results including the figures and the tables, with the output variable u the ranking for all four parameters sensitivities is k_4, k_2, k_1, k_3.

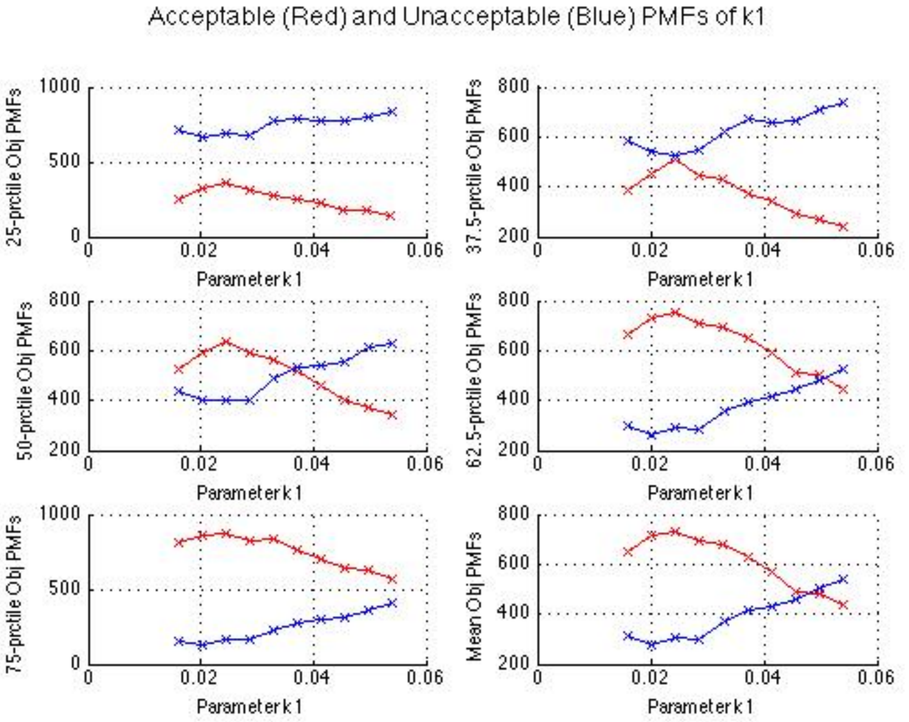

Figure 41. Probability Mass functions of k_1 considering output u

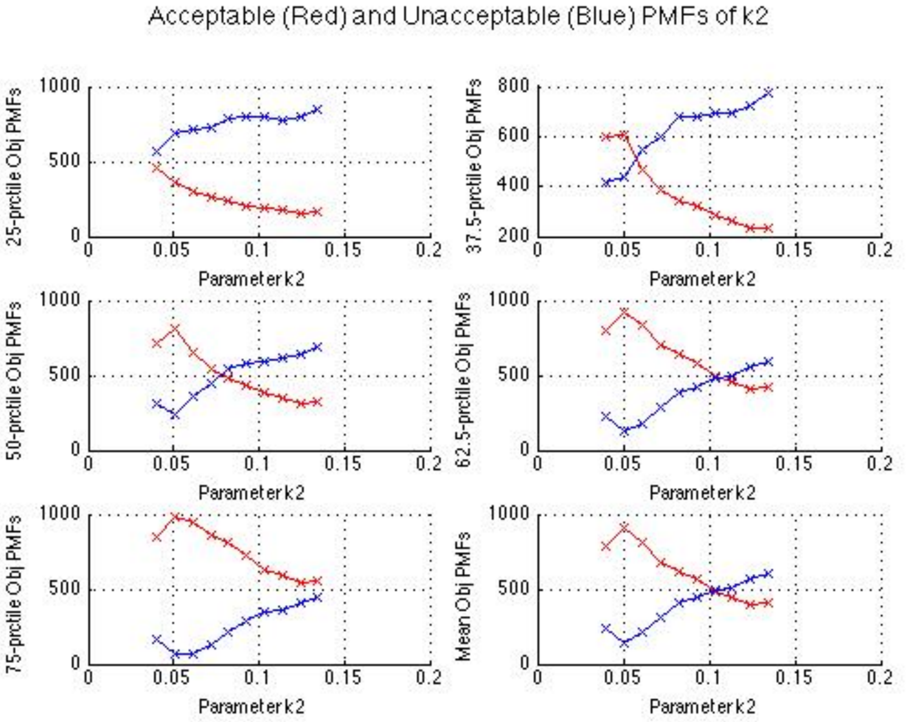

Figure 42. Probability Mass functions of k_2 considering output u

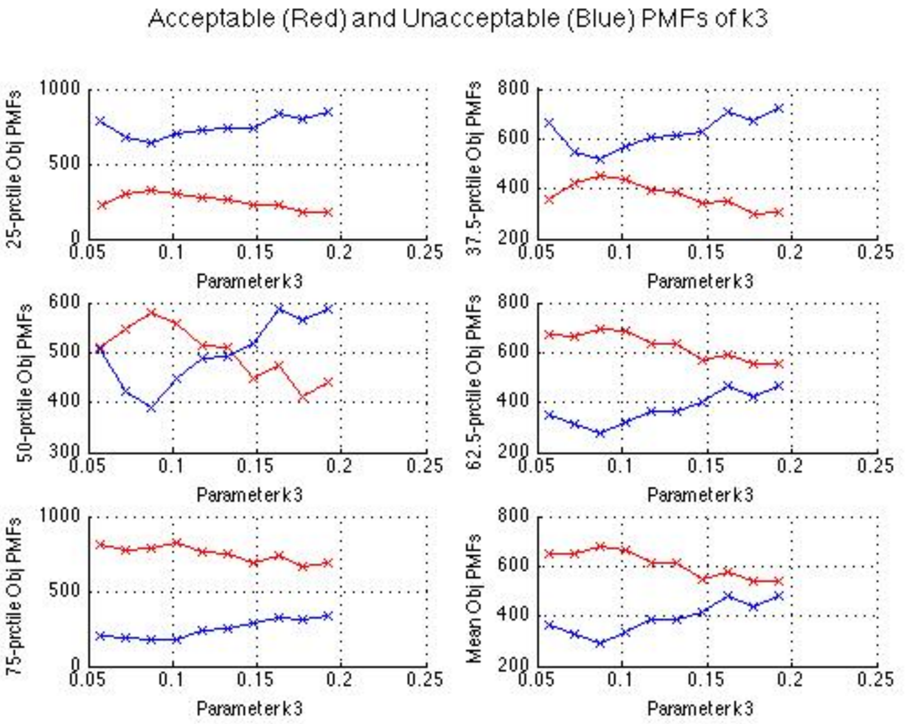

Figure 43. Probability Mass functions of k_3 considering output u

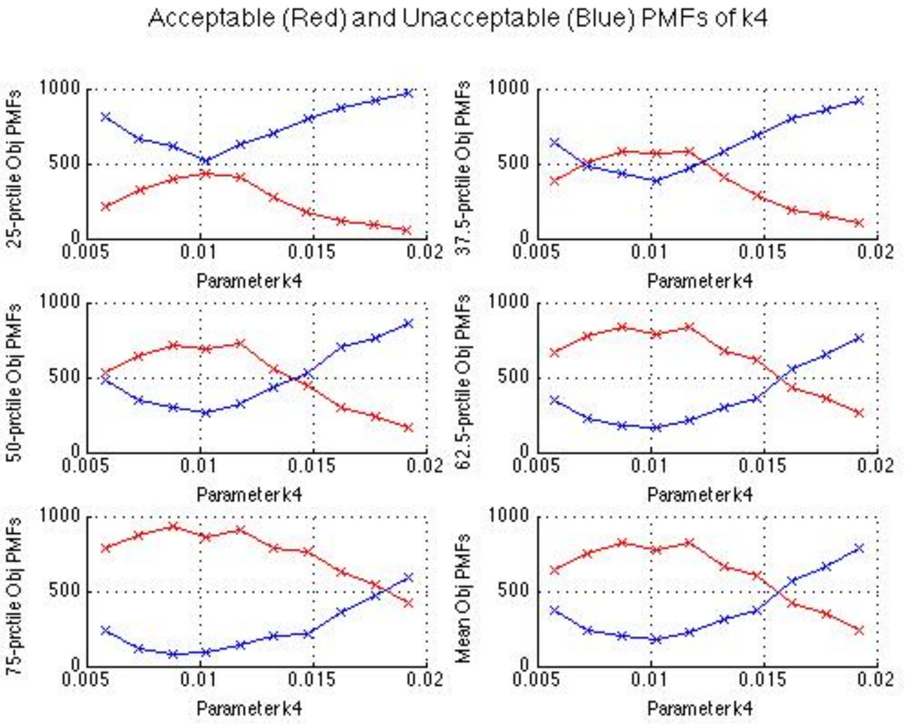

Figure 44. Probability Mass functions of k_4 considering output u

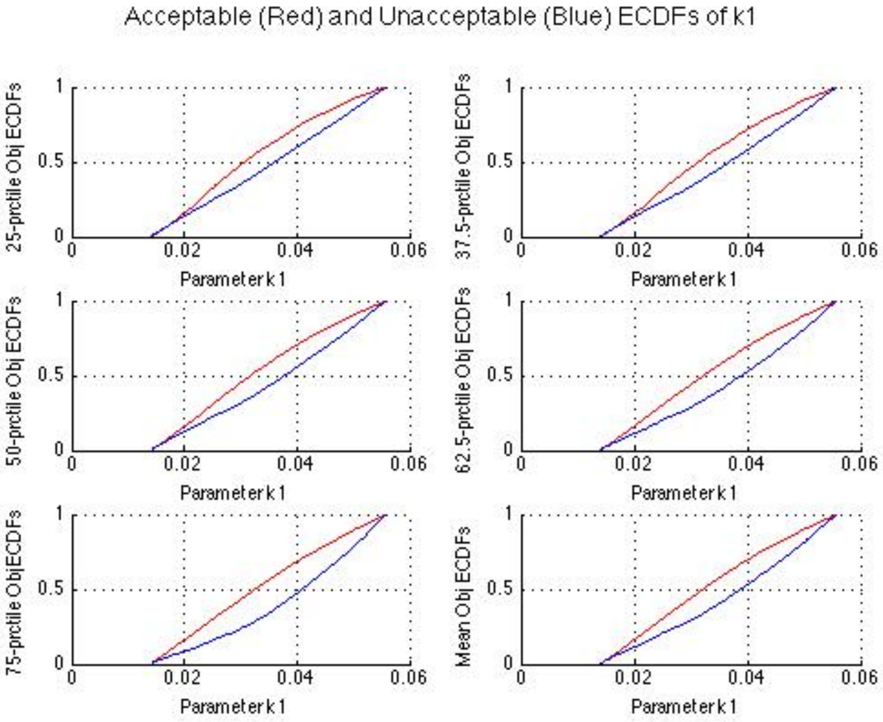

Figure 45. Cumulative distribution functions of k_1 considering output u

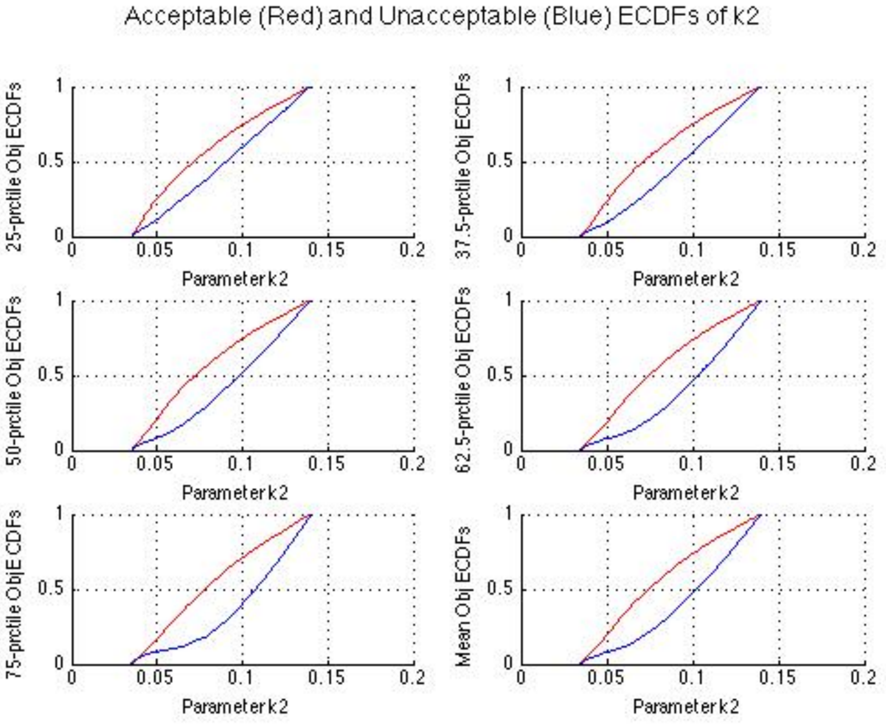

Figure 46. Cumulative distribution functions of k_2 considering output u

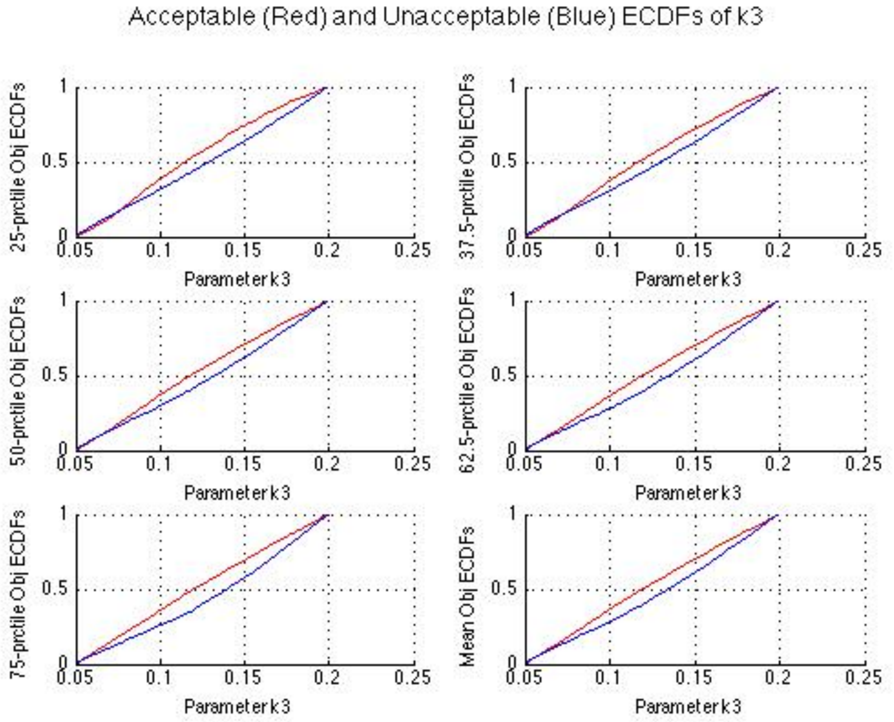

Figure 47. Cumulative distribution functions of k_3 considering output u

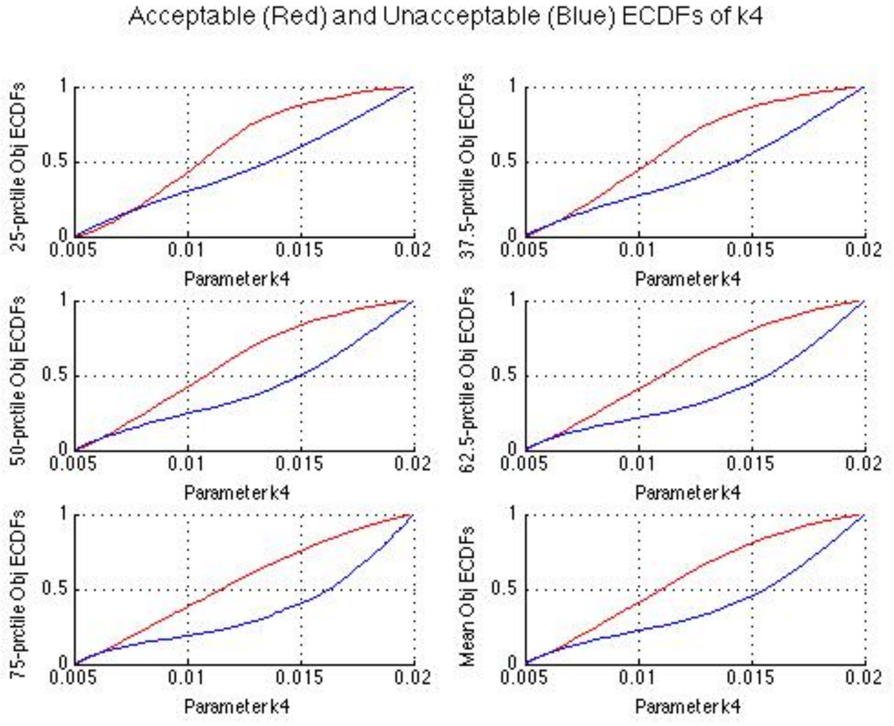

Figure 48. Cumulative distribution functions of k_4 considering output u

Table 11

Rank at 5% significance level of the MPSA results for u

Rank at 5% significance level of the MPSA results for $u, C_p^{co}(t)$				
Criterions/Parameters	k_1	k_2	k_3	k_4
25th percentile of objectives	0.1629	0.1807	0.1105	0.2933
37.5th percentile of objectives	0.1671	0.2227	0.1006	0.3268
50th percentile of objectives	0.1740	0.2542	0.1030	0.3550
62.5th percentile of objectives	0.2046	0.2862	0.1164	0.3695
75th percentile of objectives	0.2556	0.3303	0.1235	0.3796
Mean of the objectives	0.1996	0.2837	0.1135	0.3681

Table 12

P-values at 5% significance level of the MPSA results for u

P-values at 5% significance level of the MPSA results for $u, C_p^{co}(t)$ $(1*e^{-19})$				
Criterions/Parameters	k_1	k_2	k_3	k_4
25th percentile of objectives	0.0	0.0	0.1952	0.0
37.5th percentile of objectives	0.0	0.0	0.0378	0.0
50th percentile of objectives	0.0	0.0	0.0001	0.0
62.5th percentile of objectives	0.0	0.0	0.0	0.0
75th percentile of objectives	0.0	0.0	0.0	0.0
Mean of the objectives	0.0	0.0	0.0	0.0

4.3 Comparison of the Results from TSA and MPSA

Table 13

Ranking results of TSA and MPSA

Type of Sensitivity analysis	Ranking order
TSA on variable X	$k_2 - k_1 - k_4 - k_3$
MPSA on variable X	$k_2 - k_1 - k_4 - k_3$
TSA on variable y	$k_1 - k_2 - k_4 - k_3$
MPSA on variable y	$k_1 - k_2 - k_3 - k_4$
TSA on variable z	$k_1 - k_4 - k_2 - k_3$
MPSA on variable z	$k_1 - k_4 - k_2 - k_3$
TSA on variable u	$k_4 - k_1 - k_2 - k_3$
MPSA on variable u	$k_4 - k_2 - k_1 - k_3$

In comparison of the TSA and MPSA methods, Table 13 displays the ordering of the ranks. Here, we notice that both results are consistent except for the output variable y whose 3nd and 4rd sensitivity ranks are reversed in the TSA and MPSA methods, and the variable u whose 2nd and 3rd sensitivity ranks are also reversed in the TSA and MPSA methods. Nevertheless, the variable x is most sensitive to changes in the parameter k_2, the variable y and z are most sensitive to changes in the parameter k_1, and the variable u is most sensitive to changes in the parameter k_4. k_3 does not affect any output variable significantly.

CHAPTER 5

Discussion and Future Topics

From the previous studies (Ferrone et al., 1985; Chen et al., 2010), the melting of HbS polymers into monomers is beneficial, if not mandatory in order to eliminate the sickling of the red blood cell. Furthermore, carbon monoxide (CO) has been shown to improve the process of HbS polymers melting. A recent study (Sangart Inc., 2012) has developed a drug called MP4CO that delivers CO at a therapeutic level to the sickle cells to facilitate the melting process. In the extended model (Chen et al., 2010), we see that many parameters play important roles in the breaking down of the de-oxy HbS polymers which leads to the formation of the CO-bound HbS monomers. Thus, it is crucial to determine the most important parameters that affect de-oxy HbS polymers and the CO-bound HbS monomers. Therefore we analyze the sensitivity of all parameters to identify the most important parameters in the de-polymerization process.

Our first set of numerical experiments addresses the sensitivity of the variables with respect to the parameters using the TSA method. The sensitivity graphs are plotted as a function of time in Figures 2 – 5. The L^2 norms of the sensitivity functions are displayed in Table 3, which shows the strong effect of the CO-binding rate to de-oxygenated polymers ($\mathrm{k_p}[co]$) on all the variables, i.e. the most sensitive parameter. The melting rate of CO-bound polymers ($\mathrm{k^{co}}$) shows a weak effect on the variables, thus it appears to be the least sensitive or insensitive. Therefore, the concentration of de-oxygenated HbS polymers and the concentration of CO-bound HbS monomers are affected the most by the rate $\mathrm{k_p}[co]$ and the least by $\mathrm{k^{co}}$.

In the next analysis, using the previous TSFs , we study the sensitivities while all parameters are varied. As all the parameters are varied simultaneously, we notice quite a change in the sensitivity rankings. Here, we notice the importance of melting rate of de-oxygenated polymers ($\mathrm{k^d}$) on the concentration of de-oxygenated HbS polymers and CO-bound HbS monomers. It should be noted that the $\mathrm{k^d}$ is the most important parameter in the breaking down of the de-oxygenated HbS polymers and the formation of the CO-bound HbS monomers. Followed by $\mathrm{k^d}$, $\mathrm{k_p}[co]$ is the second sensitive parameter to the concentration of de-oxygenated HbS polymers and $\mathrm{k_m}[co]$ is the second sensitive parameter to the concentration of CO-bound HbS monomers. As for the concentration of de-oxy HbS monomers, $\mathrm{k_m}[co]$ followed by $\mathrm{k^d}$ are the two most sensitive parameters. For the concentration of CO-bound HbS polymers $\mathrm{k_p}[co]$ is the parameter that causes the most disturbance. It should be noted that $\mathrm{k^{co}}$ has demonstrated the weakest effect in the sensitivity of all output variables.

Lastly we perform the MPSA on the CO-mediated sickle cell de-polymerization with the same set of initial conditions and parameter ranges. The sensitivity rankings of all output values with respect to all input parameters are obtained by the MPSA analysis directly. These results are essentially identical to the results from TSA with semi-normalization of the TSFs.

In comparing the methods of sensitivity analysis employed in this study, both TSA and MPSA have pros and cons. The TSA method is simple to derive mathematically and simple to implement in the program codes to obtain the TSFs numerically. But it is limited for its large computational cost. In a case where there are many parameters (r) and output variables (n), there will be a lot of equations (n+n*r) for the TSFs. The TSA also focuses on the local effect of the parameter on the output variables. As for the MPSA method, multiple parameters can be considered. This method studies the overall effect of the parameters on the outputs variables. The disadvantage of this method is the large size of sampling for the parameters because of the Monte-Carlo simulation.

In chemical reactions, temperature plays an important role. The fluctuation of temperature influences the rate of the reactions. The constant rates presents in the model are influenced by the temperature variation (Louderback et al., 1999). Thus, it is important that the results from the sensitivity analysis be taken into consideration for further medical experiments. It is hoped that the results here may provide some insights on designing experiments that may lead to the determination of a suitable temperature range that provides a rapid de-polymerization of the sickle cell.

For the future topics, first of all, both the TSA and MPSA analyses can be conducted for the initial conditions $C_m^d(0), C_m^{CO}(0), C_p^d(0),$ and $C_p^{CO}(0)$, for each output variable $C_m^d(t), C_m^{CO}(t), C_p^d(t),$ and $C_p^{CO}(t)$. The sensitivity equations for the TSFs have been derived and given in Appendix A. This is the first topic for the continuation of the current study.

The TSF used in this study are in fact simply the derivatives of the components of the output variable with respect to the parameters. As long as one is only interested in the sensitivity of the output solely with respect to individual parameters, it is sufficient to use the TSFs. However, if we want to compare the sensitivity of outputs with respect to different parameters, then the TSFs can be misleading. Instead of the TSFs, one should use the relative sensitivity functions. More details can be found in (Salteli, 2008; Zi, 2011). Finding the relative sensitivity functions for the parameters in the CO-mediated sickle cell de-polymerization is the second topic of the future study.

Thirdly, the MPSA can be further improved. For example, the K-S measuring is a rough approximation for the distance between two CDFs. A more delicate technique may be developed to measure the difference between the acceptance and non-acceptance distributions. Another example, if there exists appreciable correlation between parameters, the current version MPSA may not be efficient. Taking the correlation of parameters into consideration, we may project the distributions onto new axes to obtain a more accurate result.

References

Alves, R., and Savageau, M.A. (2000). Systemic properties of ensembles of metabolic networks: application of graphical and statistical methods to simple unbranched pathways, Bioinoformatics, 16 (2000), 534-547

Aroutiounian, S. K., Louderback, J. G., Ballas, S. K., Kim-Shapiro, D. B. (2001). Evidence for carbon monoxide binding to sickle cell polymers during melting, Biophysical Chemistry, 91(2) (2001), 167-181

Briehl, R.W. (1995). Nucleation, fiber growth and melting, domain formation and structure in sickle hemoglobin gels, Journal of Molecular Biology, 245 (1995) 710-2723

Chang, F.-J., and Delleur, J.W. (1992). Systematic parameter estimation of watershed acidification model, Hydrology Processes, 6 (1992), 29-44

Chen, M., Clemence, D. P., Gibson, G. (2010). Analysis of numerical schemes of a mathematical model for sickle cell depolymerization, Applied Mathematics and Computation, 216 (2010), 1489-1500

Cho, Kwang-Hyun, Shin, Sung-Young, Kolch, Walter and Wilkenhauer, Olaf (2003). Experimental design in systems biology, based on parameter sensitivity analysis using a Monte Carlo method: a case study for the TNF α-mediated NF-κ B signal transduction pathway, Simulation, 79 (2003), 726-739

Choi, J., Hulseapple, S.M., Conklin, M.H., and Harvery, J.W. (1998). Modeling CO_2 degassing and pH in a stream-aquifer system, Journal of Hydrology, 209 (1998), 297-310

Daniel-Jones, Nikki, Chen, Mingxiang, Clemence, Dominic P., Gibson, Gregory (2008). A mathematical model for the sickle cell depolymerization dynamical properties and numerical experiments, International Journal of Qualitative Theory of Differential Equations and Applications, 2 (2008) 183-200.

David, John Andrew, Optimization control, Estimation, and Shape Design: Analysis and Applications, PhD Dissertation, North Carolina State University, NC, USA, 2007

Ferrone, F. A., Hofrichter, J., and Eaton, W. A. (1985). Kinetics of sickle hemoglobin polymerization II, A double nucleation mechanism, Journal of Molecular Biology, 183 (1985), 611-631

Hornberger, G. and Spear, R. (1981). An approach to the preliminary analysis of environmental system, Journal of Environmental Management, 12 (1981), 7-8

Ingram, V.M. (1956). A specific chemical difference between the globins of normal human and sickle cell anemia hemoglobin, Nature, 178 (1956) 792-794

Khalil, K.K. (2002). Nonlinear Systems, Englewood Cliffs, NJ: Prentice Hall, 2002

Louderback, J.G., Aroutiounian, S.Kh., Kerr, W.C., Ballas, S.K., Kim-Shapiro, D.B. (1999). Temperature and domain size dependence of sickle cell hemoglobin polymer melting in high concentration phosphate buffer, Biophysical Chemistry, 80 (1999), 21-30

Newman, M. M., and Barkema, G. T. (1999). Monte Carlo methods in statistical physics, New York, Oxford University Press, 1999

Phair, R. D., and Mistli, T. (2001). Kenetic modeling approaches to in vivi imaging, Nature Reviews: Molecular Cell Biology, 2 (2001), 898-907

Raue, A., Kreutz, C., Maiwald, T., Klingmuller, U., Timmer, J.(2011). Addressing parameter identifiability by model based experimentation, IET Systems Biology, 5(2) (2011), 120-130

Salteli, A., Compolongo, F., Cariboni, J., et al. (2008). Global sensitivity analysis: the primer, Wiley-Interscience, 2008

Sangart Inc., 2012. Retrieved from http://www.sangart.com/

Smith, Angela, 2013. Retrieved from http://hemeplace.wikispaces.com/Hemoglobinopathies

Vedro, Debra A. and Morrison, Rebecca A. (2002). What is Sickle Cell, Children's medical Center of Dallas, Dallas Texas, Texas Department of Health 2002

Zi, Z. (2011). Sensitivity analysis approaches applied to systems biology models, IET Systems Biology, 5(6) (2011), 336-346

Zi, Z., Cho, K.H., Sung, M.H. et al (2005). In silico identification of the key components and steps in IFN-γ induced JAK-STAT signaling pathway, FEBS Letters, 597(5) (2005), 1101-1108

Zieve, David, Chen, Yi-Bin (2011). "Sickle cell anemia", February 28, 2011, Retrieved from http://www.ncbi.nlm.nih.gov/pubmedhealth/PMH0001554/

Appendix A

Sensitivity equations for the initial conditions:

$$\frac{dx(t)}{dt} = -k_1\left(\frac{x(t)}{C_1} - 1\right)z(t) - k_2x(t) \tag{A21}$$

$$\frac{d}{dt}\frac{\partial x}{\partial x_o} = k_1\left(1 - \frac{x(t)}{C_1}\right)\frac{\partial z(t)}{\partial x_o} - k_1\,\frac{1}{C_1}\frac{\partial x(t)}{\partial x_o}z(t) - k_2\frac{\partial x(t)}{\partial x_o} \tag{A22}$$

$$\frac{d}{dt}\frac{\partial x}{\partial y_o} = k_1\left(1 - \frac{x(t)}{C_1}\right)\frac{\partial z(t)}{\partial y_o} - k_1\,\frac{1}{C_1}\frac{\partial x(t)}{\partial y_o}z(t) - k_2\frac{\partial x(t)}{\partial y_o} \tag{A23}$$

$$\frac{d}{dt}\frac{\partial x}{\partial z_o} = k_1\left(1 - \frac{x(t)}{C_1}\right)\frac{\partial z(t)}{\partial z_o} - k_1\,\frac{1}{C_1}\frac{\partial x(t)}{\partial z_o}z(t) - k_2\frac{\partial x(t)}{\partial z_o} \tag{A24}$$

$$\frac{d}{dt}\frac{\partial x}{\partial u_o} = k_1\left(1 - \frac{x(t)}{C_1}\right)\frac{\partial z(t)}{\partial u_o} - k_1\,\frac{1}{C_1}\frac{\partial x(t)}{\partial u_o}z(t) - k_2\frac{\partial x(t)}{\partial u_o} \tag{A25}$$

$$\frac{dy(t)}{dt} = k_3\left(1 - \frac{y(t)}{C_2}\right)u(t) + k_2x(t) \tag{A26}$$

$$\frac{d}{dt}\frac{\partial y}{\partial x_o} = k_3\left(1 - \frac{y(t)}{C_2}\right)\frac{\partial u(t)}{\partial x_o} - k_3\,\frac{1}{C_1}\frac{\partial y(t)}{\partial x_o}u(t) + k_2\frac{\partial x(t)}{\partial x_o} \tag{A27}$$

$$\frac{d}{dt}\frac{\partial y}{\partial y_o} = k_3\left(1 - \frac{y(t)}{C_2}\right)\frac{\partial u(t)}{\partial y_o} - k_3\,\frac{1}{C_1}\frac{\partial y(t)}{\partial y_o}u(t) + k_2\frac{\partial x(t)}{\partial y_o} \tag{A28}$$

$$\frac{d}{dt}\frac{\partial y}{\partial z_o} = k_3\left(1 - \frac{y(t)}{C_2}\right)\frac{\partial u(t)}{\partial z_o} - k_3\,\frac{1}{C_1}\frac{\partial y(t)}{\partial z_o}u(t) + k_2\frac{\partial x(t)}{\partial z_o} \tag{A29}$$

$$\frac{d}{dt}\frac{\partial y}{\partial u_o} = k_3\left(1 - \frac{y(t)}{C_2}\right)\frac{\partial u(t)}{\partial u_o} - k_3\,\frac{1}{C_1}\frac{\partial y(t)}{\partial u_o}u(t) + k_2\frac{\partial x(t)}{\partial u_o} \tag{A30}$$

$$\frac{dz(t)}{dt} = k_1\left(\frac{x(t)}{C_1} - 1\right)z(t) - k_4z(t) \tag{A31}$$

$$\frac{d}{dt}\frac{\partial z}{\partial x_o} = k_1\left(\frac{x(t)}{C_1} - 1\right)\frac{\partial z(t)}{\partial x_o} + k_1\,\frac{1}{C_1}\frac{\partial x(t)}{\partial x_o}z(t) - k_4\frac{\partial z(t)}{\partial x_o} \tag{A32}$$

$$\frac{d}{dt}\frac{\partial z}{\partial y_o} = k_1\left(\frac{x(t)}{C_1} - 1\right)\frac{\partial z(t)}{\partial y_o} + k_1\ \frac{1}{C_1}\frac{\partial x(t)}{\partial y_o}z(t) - k_4\frac{\partial z(t)}{\partial y_o} \tag*{A33}$$

$$\frac{d}{dt}\frac{\partial z}{\partial z_o} = k_1\left(\frac{x(t)}{C_1} - 1\right)\frac{\partial z(t)}{\partial z_o} + k_1\ \frac{1}{C_1}\frac{\partial x(t)}{\partial z_o}z(t) - k_4\frac{\partial z(t)}{\partial z_o} \tag*{A34}$$

$$\frac{d}{dt}\frac{\partial z}{\partial u_o} = k_1\left(\frac{x(t)}{C_1} - 1\right)\frac{\partial z(t)}{\partial u_o} + k_1\ \frac{1}{C_1}\frac{\partial x(t)}{\partial u_o}z(t) - k_4\frac{\partial z(t)}{\partial u_o} \tag*{A35}$$

$$\frac{du(t)}{dt} = -k_3\left(1 - \frac{y(t)}{C_2}\right)u(t) + k_4 z(t) \tag*{A36}$$

$$\frac{d}{dt}\frac{\partial u}{\partial x_o} = k_3\left(\frac{y(t)}{C_1} - 1\right)\frac{\partial u(t)}{\partial x_o} + k_3\ \frac{1}{C_1}\frac{\partial y(t)}{\partial x_o}u(t) + k_4\frac{\partial z(t)}{\partial x_o} \tag*{A37}$$

$$\frac{d}{dt}\frac{\partial u}{\partial y_o} = k_3\left(\frac{y(t)}{C_1} - 1\right)\frac{\partial u(t)}{\partial y_o} + k_3\ \frac{1}{C_1}\frac{\partial y(t)}{\partial y_o}u(t) + k_4\frac{\partial z(t)}{\partial y_o} \tag*{A38}$$

$$\frac{d}{dt}\frac{\partial u}{\partial z_o} = k_3\left(\frac{y(t)}{C_1} - 1\right)\frac{\partial u(t)}{\partial z_o} + k_3\ \frac{1}{C_1}\frac{\partial y(t)}{\partial z_o}u(t) + k_4\frac{\partial z(t)}{\partial z_o} \tag*{A39}$$

$$\frac{d}{dt}\frac{\partial u}{\partial u_o} = k_3\left(\frac{y(t)}{C_1} - 1\right)\frac{\partial u(t)}{\partial u_o} + k_3\ \frac{1}{C_1}\frac{\partial y(t)}{\partial u_o}u(t) + k_4\frac{\partial z(t)}{\partial u_o} \tag*{A40}$$

Appendix B

Matlab codes for TSF

Main code for the TSA

```
% This Matlab code analyzes sensitivity of the system to four parameters
% k1, k2, k3 and k4
% Initialization of the parameter of the model
global k1 k2 k3 k4 C_1 C_2
k1 = .028;
k2 = .07;
k3 = .1;
k4 = .01;
C_1 = .4;
C_2 = .1;
k0 = [k1,k2,k3,k4];
eps = min([k1,k2,k3,k4])/10.0;
t0 = 0;
tf = 400;
dt = 0.01;
X0 = [0.0036,0.0,0.0,0.0,0.0, 0.0,0.0,0.0,0.0,0.0,...
   0.1750,0.0,0.0,0.0,0.0, 0.0,0.0,0.0,0.0,0.0];
% Calling My runge-kutta 4 method to compute the original (nominal)
% behaviour
[t,Y0] = MYRK4COSens(X0,t0,tf,dt);
product = Y0(:,6);
figure(1)
hold on; grid on;
plot(t,Y0(:,1),'b')
plot(t,Y0(:,6),'r')
plot(t,Y0(:,11),'g')
plot(t,Y0(:,16),'k')
legend('De-oxy Monomers [x]','CO-bound Monomers [y]',...
   'De-oxy polymers [z]','CO-bound Polymers [u]' );
title('Molar Cencentration') % put a title on the plot
xlabel('Time [min]')%  label the x-axis
ylabel('Population [Mm]') %  label the y-axis
h1 = figure(1);
saveas(h1,'Fig1_solnCurves4model','jpg')
figure(2)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
plot(t,Y0(:,2))
title('dx/dk1 solution curve');
subplot(2,2,2)
```

```
hold on; grid on;
plot(t,Y0(:,7))
title('dy/dk1 solution curve');
subplot(2,2,3)
hold on; grid on;
plot(t,Y0(:,12))
title('dz/dk1 solution curve');
subplot(2,2,4)
hold on; grid on;
plot(t,Y0(:,17))
title('du/dk1 solution curve');
suptitle('Behavior of dx/dk1,dy/dk1,dz/dk1,du/dk1 variables');
h2 = figure(2);
saveas(h2,'Fig2_solnCurves4variationsOfK1','jpg')
figure(3)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
plot(t,Y0(:,3))
title('dx/dk2 solution curve');
subplot(2,2,2)
hold on; grid on;
plot(t,Y0(:,8))
title('dy/dk2 solution curve');
subplot(2,2,3)
hold on; grid on;
plot(t,Y0(:,13))
title('dz/dk2 solution curve');
subplot(2,2,4)
hold on; grid on;
plot(t,Y0(:,18))
title('du/dk2 solution curve');
suptitle('Behavior of dx/dk2,dy/dk2,dz/dk2,du/dk2 variables');
h3 = figure(3);
saveas(h3,'Fig3_solnCurves4variationsOfK2','jpg')
figure(4)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
plot(t,Y0(:,4))
title('dx/dk3 solution curve');
subplot(2,2,2)
hold on; grid on;
plot(t,Y0(:,9))
title('dy/dk3 solution curve');
subplot(2,2,3)
```

```
hold on; grid on;
plot(t,Y0(:,14))
title('dz/dk3 solution curve');
subplot(2,2,4)
hold on; grid on;
plot(t,Y0(:,19))
title('du/dk3 solution curve');
suptitle('Behavior of dx/dk3,dy/dk3,dz/dk3,du/dk3 variables');
h4 = figure(4);
saveas(h4,'Fig4_solnCurves4variationsOfK3','jpg')
figure(5)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
plot(t,Y0(:,5))
title('dx/dk4 solution curve');
subplot(2,2,2)
hold on; grid on;
plot(t,Y0(:,10))
title('dy/dk4 solution curve');
subplot(2,2,3)
hold on; grid on;
plot(t,Y0(:,15))
title('dz/dk4 solution curve');
subplot(2,2,4)
hold on; grid on;
plot(t,Y0(:,20))
title('du/dk4 solution curve');
suptitle('Behavior of dx/dk4,dy/dk4,dz/dk4,du/dk4 variables');
h5 = figure(5);
saveas(h5,'Fig5_solnCurves4variationsOfK4','jpg')
saveas(h1,'Fig1_solnCurves4model','png')
saveas(h2,'Fig2_solnCurves4variationsOfK1','png')
saveas(h3,'Fig3_solnCurves4variationsOfK2','png')
saveas(h4,'Fig4_solnCurves4variationsOfK3','png')
saveas(h5,'Fig5_solnCurves4variationsOfK4','png')
% Define the number of points for the simulation
dim = 100;
% Defines the initial starting point: 1/2 of the nominal value
k1_init = 0.5*k1;
k2_init = 0.5*k2;
k3_init = 0.5*k3;
k4_init = 0.5*k4;
% Defines the end point: 2 times of the nominal value
k1_fin = 2.0*k1;
k2_fin = 2.0*k2;
```

```
k3_fin = 2.0*k3;
k4_fin = 2.0*k4;
% Creates a uniform random number distribution between the set of 2 points
k1v = random('unif',k1_init,k1_fin,1,dim);
k2v = random('unif',k2_init,k2_fin,1,dim);
k3v = random('unif',k3_init,k3_fin,1,dim);
k4v = random('unif',k4_init,k4_fin,1,dim);
k1 = k1v(1);
k2 = k2v(1);
k3 = k3v(1);
k4 = k4v(1);
[t,Y] = MYRK4COSens(X0,t0,tf,dt);
figure(6)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
plot(t,Y(:,1)-Y0(:,1))
ylabel('x-x_ref')
%title('x variation wrt to Ks');
subplot(2,2,2)
hold on; grid on;
plot(t,Y(:,6)-Y0(:,6))
ylabel('y-y_ref')
%title('y variation wrt Ks');
subplot(2,2,3)
hold on; grid on;
plot(t,Y(:,11)-Y0(:,11))
ylabel('z-z_ref')
%title('z variation wrt Ks');
subplot(2,2,4)
hold on; grid on;
plot(t,Y(:,16)-Y0(:,16))
ylabel('u-u_ref')
k1x_min = abs(k1-k0(1))*sum(abs(Y(:,2)-Y0(:,2)).^2)*dt;
k1y_min = abs(k1-k0(1))*sum(abs(Y(:,7)-Y0(:,7)).^2)*dt;
k1z_min = abs(k1-k0(1))*sum(abs(Y(:,12)-Y0(:,12)).^2)*dt;
k1u_min = abs(k1-k0(1))*sum(abs(Y(:,17)-Y0(:,17)).^2)*dt;
k2x_min = abs(k2-k0(2))*sum(abs(Y(:,3)-Y0(:,3)).^2)*dt;
k2y_min = abs(k2-k0(2))*sum(abs(Y(:,8)-Y0(:,8)).^2)*dt;
k2z_min = abs(k2-k0(2))*sum(abs(Y(:,13)-Y0(:,13).^2))*dt;
k2u_min = abs(k2-k0(2))*sum(abs(Y(:,18)-Y0(:,18)).^2)*dt;
k3x_min = abs(k3-k0(3))*sum(abs(Y(:,4)-Y0(:,4)).^2)*dt;
k3y_min = abs(k3-k0(3))*sum(abs(Y(:,9)-Y0(:,9)).^2)*dt;
k3z_min = abs(k3-k0(3))*sum(abs(Y(:,14)-Y0(:,14)).^2)*dt;
k3u_min = abs(k3-k0(3))*sum(abs(Y(:,19)-Y0(:,19)).^2)*dt;
k4x_min = abs(k4-k0(4))*sum(abs(Y(:,5)-Y0(:,5)).^2)*dt;
```

```
k4y_min = abs(k4-k0(4))*sum(abs(Y(:,10)-Y0(:,10)).^2)*dt;
k4z_min = abs(k4-k0(4))*sum(abs(Y(:,15)-Y0(:,15)).^2)*dt;
k4u_min = abs(k4-k0(4))*sum(abs(Y(:,20)-Y0(:,20)).^2)*dt;
k1x_max = abs(k1-k0(1))*sum(abs(Y(:,2)-Y0(:,2)).^2)*dt;
k1y_max = abs(k1-k0(1))*sum(abs(Y(:,7)-Y0(:,7)).^2)*dt;
k1z_max = abs(k1-k0(1))*sum(abs(Y(:,12)-Y0(:,12)).^2)*dt;
k1u_max = abs(k1-k0(1))*sum(abs(Y(:,17)-Y0(:,17)).^2)*dt;
k2x_max = abs(k2-k0(2))*sum(abs(Y(:,3)-Y0(:,3)).^2)*dt;
k2y_max = abs(k2-k0(2))*sum(abs(Y(:,8)-Y0(:,8)).^2)*dt;
k2z_max = abs(k2-k0(2))*sum(abs(Y(:,13)-Y0(:,13)).^2)*dt;
k2u_max = abs(k2-k0(2))*sum(abs(Y(:,18)-Y0(:,18)).^2)*dt;
k3x_max = abs(k3-k0(3))*sum(abs(Y(:,4)-Y0(:,4)).^2)*dt;
k3y_max = abs(k3-k0(3))*sum(abs(Y(:,9)-Y0(:,9)).^2)*dt;
k3z_max = abs(k3-k0(3))*sum(abs(Y(:,14)-Y0(:,14)).^2)*dt;
k3u_max = abs(k3-k0(3))*sum(abs(Y(:,19)-Y0(:,19)).^2)*dt;
k4x_max = abs(k4-k0(4))*sum(abs(Y(:,5)-Y0(:,5)).^2)*dt;
k4y_max = abs(k4-k0(4))*sum(abs(Y(:,10)-Y0(:,10)).^2)*dt;
k4z_max = abs(k4-k0(4))*sum(abs(Y(:,15)-Y0(:,15)).^2)*dt;
k4u_max = abs(k4-k0(4))*sum(abs(Y(:,20)-Y0(:,20)).^2)*dt;
x_diff_min = sum(abs(Y(:,1)-Y0(:,1)).^2)*dt;
y_diff_min = sum(abs(Y(:,6)-Y0(:,6)).^2)*dt;
z_diff_min = sum(abs(Y(:,11)-Y0(:,11)).^2)*dt;
u_diff_min = sum(abs(Y(:,16)-Y0(:,16)).^2)*dt;
x_diff_max = sum(abs(Y(:,1)-Y0(:,1)).^2)*dt;
y_diff_max = sum(abs(Y(:,6)-Y0(:,6)).^2)*dt;
z_diff_max = sum(abs(Y(:,11)-Y0(:,11)).^2)*dt;
u_diff_max = sum(abs(Y(:,16)-Y0(:,16)).^2)*dt;
% For loop1: sensitivity of x,y,z,u wrt k1,k2,k3,k4 when all four
% parameters vary simultaneously
for i_k = 2:1:dim
    k1 = k1v(i_k);
    k2 = k2v(i_k);
    k3 = k3v(i_k);
    k4 = k4v(i_k);
    [t,Y] = MYRK4COSens(X0,t0,tf,dt);
    x_diff = sum(abs(Y(:,1)-Y0(:,1)).^2)*dt;
    y_diff = sum(abs(Y(:,6)-Y0(:,6)).^2)*dt;
    z_diff = sum(abs(Y(:,11)-Y0(:,11).^2))*dt;
    u_diff = sum(abs(Y(:,16)-Y0(:,16)).^2)*dt;
    if (x_diff<x_diff_min)
        x_diff_min = x_diff;
    elseif (x_diff>x_diff_max)
        x_diff_max = x_diff;
    end
    if (y_diff<y_diff_min)
        y_diff_min = y_diff;
```

```
elseif (y_diff>y_diff_max)
    y_diff_max = y_diff;
end
if (z_diff<z_diff_min)
    z_diff_min = z_diff;
elseif (x_diff>x_diff_max)
    z_diff_max = z_diff;
end
if (u_diff<u_diff_min)
    u_diff_min = u_diff;
elseif (u_diff>u_diff_max)
    u_diff_max = u_diff;
end
figure(6)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
plot(t,Y(:,1)-Y0(:,1))
ylabel('x-x_ref')
%title('x variation wrt to Ks');
subplot(2,2,2)
hold on; grid on;
plot(t,Y(:,6)-Y0(:,6))
ylabel('y-y_ref')
%title('y variation wrt Ks');
subplot(2,2,3)
hold on; grid on;
plot(t,Y(:,11)-Y0(:,11))
ylabel('z-z_ref')
%title('z variation wrt Ks');
subplot(2,2,4)
hold on; grid on;
plot(t,Y(:,16)-Y0(:,16))
ylabel('u-u_ref')
%title('u variation wrt k1');
%suptitle('Behavior of x,y,z,u variables wrt Ks');
k1x_tmp = abs(k1-k0(1))*sum(abs(Y(:,2)-Y0(:,2)).^2)*dt;
k1y_tmp = abs(k1-k0(1))*sum(abs(Y(:,7)-Y0(:,7)).^2)*dt;
k1z_tmp = abs(k1-k0(1))*sum(abs(Y(:,12)-Y0(:,12)).^2)*dt;
k1u_tmp = abs(k1-k0(1))*sum(abs(Y(:,17)-Y0(:,17)).^2)*dt;
k2x_tmp = abs(k2-k0(2))*sum(abs(Y(:,3)-Y0(:,3)).^2)*dt;
k2y_tmp = abs(k2-k0(2))*sum(abs(Y(:,8)-Y0(:,8)).^2)*dt;
k2z_tmp = abs(k2-k0(2))*sum(abs(Y(:,13)-Y0(:,13)).^2)*dt;
k2u_tmp = abs(k2-k0(2))*sum(abs(Y(:,18)-Y0(:,18)).^2)*dt;
k3x_tmp = abs(k3-k0(3))*sum(abs(Y(:,4)-Y0(:,4)).^2)*dt;
k3y_tmp = abs(k3-k0(3))*sum(abs(Y(:,9)-Y0(:,9)).^2)*dt;
```

```
    k3z_tmp = abs(k3-k0(3))*sum(abs(Y(:,14)-Y0(:,14)).^2)*dt;
    k3u_tmp = abs(k3-k0(3))*sum(abs(Y(:,19)-Y0(:,19)).^2)*dt;
    k4x_tmp = abs(k4-k0(4))*sum(abs(Y(:,5)-Y0(:,5)).^2)*dt;
    k4y_tmp = abs(k4-k0(4))*sum(abs(Y(:,10)-Y0(:,10)).^2)*dt;
    k4z_tmp = abs(k4-k0(4))*sum(abs(Y(:,15)-Y0(:,15)).^2)*dt;
    k4u_tmp = abs(k4-k0(4))*sum(abs(Y(:,20)-Y0(:,20)).^2)*dt;
    if (k1x_tmp < k1x_min)
        k1x_min = k1x_tmp;
    elseif (k1x_tmp > k1x_max)
        k1x_max = k1x_tmp;
    end
if (k1y_tmp < k1y_min)
        k1y_min = k1y_tmp;
    elseif (k1y_tmp > k1y_max)
        k1y_max = k1y_tmp;
    end
if (k1z_tmp < k1z_min)
        k1z_min = k1z_tmp;
    elseif (k1z_tmp > k1z_max)
        k1z_max = k1z_tmp;
    end
if (k1u_tmp < k1u_min)
        k1u_min = k1u_tmp;
    elseif (k1u_tmp > k1u_max)
        k1u_max = k1u_tmp;
    end
if (k2x_tmp < k2x_min)
        k2x_min = k2x_tmp;
    elseif (k2x_tmp > k2x_max)
        k2x_max = k2x_tmp;
    end
if (k2y_tmp < k2y_min)
        k2y_min = k2y_tmp;
    elseif (k2y_tmp > k2y_max)
        k2y_max = k2y_tmp;
    end
if (k2z_tmp < k2z_min)
        k2z_min = k2z_tmp;
    elseif(k2z_tmp > k2z_max)
        k2z_max = k2z_tmp;
    end
if (k2u_tmp < k2u_min)
        k2u_min = k2u_tmp;
    elseif (k2u_tmp > k2u_max)
        k2u_max = k2u_tmp;
    end
```

```
if (k3x_tmp < k3x_min)
        k3x_min = k3x_tmp;
    elseif (k3x_tmp > k3x_max)
        k3x_max = k3x_tmp;
    end
if (k3y_tmp < k3y_min)
        k3y_min = k3y_tmp;
    elseif (k3y_tmp > k3y_max)
        k3y_max = k3y_tmp;
    end
if (k3z_tmp < k3z_min)
        k3z_min = k3z_tmp;
    elseif(k3z_tmp > k3z_max)
        k3z_max = k3z_tmp;
    end
if (k3u_tmp < k3u_min)
        k3u_min = k3u_tmp;
    elseif (k3u_tmp > k3u_max)
        k3u_max = k3u_tmp;
    end
if (k4x_tmp < k4x_min)
        k4x_min = k4x_tmp;
    elseif (k4x_tmp > k4x_max)
        k4x_max = k4x_tmp;
    end
    if (k4y_tmp < k4y_min)
        k4y_min = k4y_tmp;
    elseif (k4y_tmp > k4y_max)
        k4y_max = k4y_tmp;
    end
if (k4z_tmp < k4z_min)
        k4z_min = k4z_tmp;
    elseif(k4z_tmp > k4z_max)
        k4z_max = k4z_tmp;
    end
    if (k4u_tmp < k4u_min)
        k4u_min = k4u_tmp;
    elseif (k4u_tmp > k4u_max)
        k4u_max = k4u_tmp;
    end
end
% Add titles to figures in for loop1 above
figure(6)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
```

```
ylabel('x-x_ref')
title('x variation wrt varying Ks');
subplot(2,2,2)
hold on; grid on;
ylabel('y-y_ref')
title('y variation wrt varying Ks');
subplot(2,2,3)
hold on; grid on;
ylabel('z-z_ref')
title('z variation wrt varying Ks');
subplot(2,2,4)
hold on; grid on;
ylabel('u-u_ref')
title('u variation wrt varying Ks');
suptitle('Behavior of x,y,z,u variables for varying Ks');
h1 = figure(6);
saveas(h1,'Fig6_DiffWRefSolnCurve_varyKs','jpg')
saveas(h1,'Fig6_DiffWRefSolnCurve_varyKs','png')
RefSolnDiff_range = [x_diff_min,y_diff_min,z_diff_min,u_diff_min;
   x_diff_max,y_diff_max,z_diff_max,u_diff_max;
   x_diff_max-x_diff_min,y_diff_max-y_diff_min,z_diff_max-z_diff_min,...
   u_diff_max-u_diff_min]
fileID = fopen('SolnDiffRange.txt','w');
fprintf(fileID,'%10s %12s %12s %12s\r\n','x_diff','y_diff',...
   'z_diff','u_diff');
fprintf(fileID,'%12.8f %12.8f %12.8f %12.8f\r\n',RefSolnDiff_range');
fclose(fileID);
k1_range = [k1x_min,k1y_min,k1z_min,k1u_min;
   k1x_max,k1y_max,k1z_max,k1u_max;
   k1x_max-k1x_min,k1y_max-k1y_min,k1z_max-k1z_min,k1u_max-k1u_min]
fileID = fopen('K1Range.txt','w');
fprintf(fileID,'%10s %12s %12s %12s\r\n','k1x','k1y','k1z','k1u');
fprintf(fileID,'%12.8f %12.8f %12.8f %12.8f\r\n',k1_range');
fclose(fileID);
k2_range = [k2x_min,k2y_min,k2z_min,k2u_min;
   k2x_max,k2y_max,k2z_max,k2u_max;
   k2x_max-k2x_min,k2y_max-k2y_min,k2z_max-k2z_min,k2u_max-k2u_min]
fileID = fopen('K2Range.txt','w');
fprintf(fileID,'%10s %12s %12s %12s\r\n','k2x','k2y','k2z','k2u');
fprintf(fileID,'%12.8f %12.8f %12.8f %12.8f\r\n',k2_range');
fclose(fileID);
k3_range = [k3x_min,k3y_min,k3z_min,k3u_min;
   k3x_max,k3y_max,k3z_max,k3u_max;
   k3x_max-k3x_min,k3y_max-k3y_min,k3z_max-k3z_min,k3u_max-k3u_min]
fileID = fopen('K3Range.txt','w');
fprintf(fileID,'%10s %12s %12s %12s\r\n','k3x','k3y','k3z','k3u');
fprintf(fileID,'%12.8f %12.8f %12.8f %12.8f\r\n',k3_range');
fclose(fileID);
k4_range = [k4x_min,k4y_min,k4z_min,k4u_min;
   k4x_max,k4y_max,k4z_max,k4u_max;
   k4x_max-k4x_min,k4y_max-k4y_min,k4z_max-k4z_min,k4u_max-k4u_min]
fileID = fopen('K4Range.txt','w');
fprintf(fileID,'%10s %12s %12s %12s\r\n','k4x','k4y','k4z','k4u');
fprintf(fileID,'%12.8f %12.8f %12.8f %12.8f\r\n',k4_range');
fclose(fileID);
% For loop1: sensitivity of x,y,z,u wrt k1,k2,k3,k4 when all four
% parametersvary simultaneously
for i_k = 1:1:dim
        k1 = k1v(i_k);
        k2 = k2v(i_k);
        k3 = k3v(i_k);
        k4 = k4v(i_k);
        [t,Y] = MYRK4COSensNew(X0,t0,tf,dt);
        figure(6)
        hold on; grid on;
        subplot(2,2,1)
        hold on; grid on;
        plot(t,Y(:,1))
        ylabel('x')
        %title('x variation wrt to k1');
        subplot(2,2,2)
        hold on; grid on;
        plot(t,Y(:,6))
        ylabel('y')
        %title('y variation wrt k1');
        subplot(2,2,3)
        hold on; grid on;
        plot(t,Y(:,11))
        ylabel('z')
        %title('z variation wrt k1');
        subplot(2,2,4)
        hold on; grid on;
        plot(t,Y(:,16))
        ylabel('u')
        %title('u variation wrt k1');
        %suptitle('Behavior of x,y,z,u variables wrt k1');
        figure(7)
        hold on; grid on;
        subplot(2,2,1)
        hold on; grid on;
        plot(t,Y(:,2))
        ylabel('dx/dk1')
```

```
%title('x variation wrt to k1');
subplot(2,2,2)
hold on; grid on;
plot(t,Y(:,7))
ylabel('dy/dk1')
%title('y variation wrt k1');
subplot(2,2,3)
hold on; grid on;
plot(t,Y(:,12))
ylabel('dz/dk1')
%title('z variation wrt k1');
subplot(2,2,4)
hold on; grid on;
plot(t,Y(:,17))
ylabel('du/dk1')
%title('u variation wrt k1');
%suptitle('Behavior of dx,dy,dz,du variables wrt k1');
figure(8)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
plot(t,Y(:,3))
ylabel('dx/dk2')
%title('x variation wrt k2');
subplot(2,2,2)
hold on; grid on;
plot(t,Y(:,8))
ylabel('dy/dk2')
%title('y variation wrt k2');
subplot(2,2,3)
hold on; grid on;
plot(t,Y(:,13))
ylabel('dz/dk2')
%title('z variation wrt k2');
subplot(2,2,4)
hold on; grid on;
plot(t,Y(:,18))
ylabel('du/dk2')
%title('u variation wrt k2');
%suptitle('Behavior of x,y,z,u variables wrt k2')
figure(9)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
plot(t,Y(:,4))
ylabel('dx/dk3')
```

```
    %title('x variation wrt k3');
    subplot(2,2,2)
    hold on; grid on;
    plot(t,Y(:,9))
    ylabel('dy/dk3')
    %title('y variation wrt k3');
    subplot(2,2,3)
    hold on; grid on;
    plot(t,Y(:,14))
    ylabel('dz/dk3')
    %title('z variation wrt k3');
    subplot(2,2,4)
    hold on; grid on;
    plot(t,Y(:,19))
    ylabel('du/dk3')
    %title('u variation wrt k3');
    %suptitle('Vehavior of x,y,z,u variables wrt k3');
    figure(10)
    hold on; grid on;
    subplot(2,2,1)
    hold on; grid on;
    plot(t,Y(:,5))
    ylabel('dx/dk4')
    %title('x variation wrt k4');
    subplot(2,2,2)
    hold on; grid on;
    plot(t,Y(:,10))
    ylabel('dy/dk4')
    %title('y variation wrt k4');
    subplot(2,2,3)
    hold on; grid on;
    plot(t,Y(:,15))
    ylabel('dz/dk4')
    %title('z variation wrt k4');
    subplot(2,2,4)
    hold on; grid on;
    plot(t,Y(:,20))
    ylabel('du/dk4')
    %title('u variation wrt k4');
    %suptitle('Behavior of x,y,z,u variables wrt k4');
end
% Add titles to figures in for loop1 above
figure(6)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
```

```
plot(t,Y(:,1))
ylabel('x')
title('x variation wrt varying Ks');
subplot(2,2,2)
hold on; grid on;
plot(t,Y(:,6))
ylabel('y')
title('y variation wrt varying Ks');
subplot(2,2,3)
hold on; grid on;
plot(t,Y(:,11))
ylabel('z')
title('z variation wrt varying Ks');
subplot(2,2,4)
hold on; grid on;
plot(t,Y(:,16))
ylabel('u')
title('u variation wrt varying Ks');
suptitle('Behavior of x,y,z,u variables for varying Ks');
figure(7)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
plot(t,Y(:,2))
ylabel('dx/dk1')
title('dx/dk1 variation wrt varying Ks');
subplot(2,2,2)
hold on; grid on;
plot(t,Y(:,7))
ylabel('dy/dk1')
title('dy/dk1 variation wrt varying Ks');
subplot(2,2,3)
hold on; grid on;
plot(t,Y(:,12))
ylabel('dz/dk1')
title('dz/dk1 variation wrt varying Ks');
subplot(2,2,4)
hold on; grid on;
plot(t,Y(:,17))
ylabel('du/dk1')
title('du/dk1 variation wrt varying Ks');
suptitle('Behavior of dx/dk1,dy/dk1,dz/dk1,du/dk1 for varying Ks');
figure(8)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
```

```
plot(t,Y(:,3))
ylabel('dx/dk2')
title('dx/dk2 variation wrt varying Ks');
subplot(2,2,2)
hold on; grid on;
plot(t,Y(:,8))
ylabel('dy/dk2')
title('dy/dk2 variation wrt varying Ks');
subplot(2,2,3)
hold on; grid on;
plot(t,Y(:,13))
ylabel('dz/dk2')
title('dz/dk2 variation wrt varying Ks');
subplot(2,2,4)
hold on; grid on;
plot(t,Y(:,18))
ylabel('du/dk2')
title('du/dk2 variation wrt varying Ks');
suptitle('Behavior of dx/dk2,dy/dk2,dz/dk2,du/dk2 for varying Ks');
figure(9)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
plot(t,Y(:,4))
ylabel('dx/dk3')
title('dx/dk3 variation wrt varying Ks');
subplot(2,2,2)
hold on; grid on;
plot(t,Y(:,9))
ylabel('dy/dk3')
title('dy/dk3 variation wrt varying Ks');
subplot(2,2,3)
hold on; grid on;
plot(t,Y(:,14))
ylabel('dz/dk3')
title('duz/dk3 variation wrt varying Ks');
subplot(2,2,4)
hold on; grid on;
plot(t,Y(:,19))
ylabel('du/dk3')
title('du/dk3 variation wrt varying Ks');
suptitle('Behavior of dx/dk3,dy/dk3,dz/dk3,du/dk3 for varying Ks');
figure(10)
hold on; grid on;
subplot(2,2,1)
hold on; grid on;
```

```
plot(t,Y(:,5))
ylabel('dx/dk4')
title('dx/dk4 variation wrt varying Ks');
subplot(2,2,2)
hold on; grid on;
plot(t,Y(:,10))
ylabel('dy/dk4')
title('dy/dk4 variation wrt varying Ks');
subplot(2,2,3)
hold on; grid on;
plot(t,Y(:,15))
ylabel('dz/dk4')
title('dz/dk4 variation wrt varying Ks');
subplot(2,2,4)
hold on; grid on;
plot(t,Y(:,20))
ylabel('du/dk4')
title('du/dk4 variation wrt varying Ks');
suptitle('Behavior of dx/dk4,dy/dk4,dz/dk4,du/dk4 for varying Ks');
h1 = figure(6);
h2 = figure(7);
h3 = figure(8);
h4 = figure(9);
h5 = figure(10);
saveas(h1,'Fig6_solnCurves_varyKs','jpg')
saveas(h2,'Fig7_variations2k1_varyKs','jpg')
saveas(h3,'Fig8_variations2k2_varyKs','jpg')
saveas(h4,'Fig9_variations2k3_varyKs','jpg')
saveas(h5,'Fig10_variations2k4_varyKs','jpg')
saveas(h1,'Fig6_solnCurves_varyKs','png')
saveas(h2,'Fig7_variations2k1_varyKs','png')
saveas(h3,'Fig8_variations2k2_varyKs','png')
saveas(h4,'Fig9_variations2k3_varyKs','png')
saveas(h5,'Fig10_variations2k4_varyKs','png')
```

Subroutines for the TSA

```
function dxdt = COSensNew(X)
% This Matlab code defines an extended set of 20 PDEs defining the dynamic
% of Sickle Cell model and the variations of the dependant variables with
% respect to four parameters k1,k2,k3 and k4.
global k1 k2 k3 k4 C_1 C_2
dxdt(1)= -k1*(X(1)/C_1-1)*X(11)-k2*X(1);
dxdt(2)= (1-X(1)/C_1)*(X(11)+k1*X(12))-(k1*(1/C_1)*X(11)+k2)*X(2);
dxdt(3)= -k1*((1/C_1)*X(3))*X(11)+k1*(1-X(1)/C_1)*X(13)-k2*X(3)-X(1);
dxdt(4)= -k1*((1/C_1)*X(4))*X(11)+k1*(1-X(1)/C_1)*X(14)-k2*X(4);
dxdt(5)= -k1*((1/C_1)*X(5))*X(11)+k1*(1-X(1)/C_1)*X(15)-k2*X(5);
dxdt(6)= k3*(1-(X(6)/C_2))*X(16)+k2*X(1);
dxdt(7)= -k3*((1/C_2)*X(7))*X(16)+k3*X(17)*(1-X(6)/C_2)+k2*X(2);
dxdt(8)= -k3*((1/C_2)*X(8))*X(16)+k3*X(18)*(1-X(6)/C_2)+k2*X(3)+X(1);
dxdt(9)= (1-X(6)/C_2)*(X(16)+k3*X(19))-k3*((1/C_2)*X(9))*X(16)+k2*X(4);
dxdt(10)= -k3*((1/C_2)*X(10))*X(16)+k3*(1-X(6)/C_2)*X(20)+k2*X(5);
dxdt(11)= k1*(X(1)/C_1-1)*X(11)-k4*X(11);
dxdt(12)= ((X(1)/C_1)-1)*(X(11)+k1*X(12))+k1*((1/C_1)*X(2))*X(11)-k4*X(12);
dxdt(13)= k1*((1/C_1)*X(3))*X(11)+k1*((X(1)/C_1)-1)*X(13)-k4*X(13);
dxdt(14)= k1*((1/C_1)*X(4))*X(11)+k1*((X(1)/C_1)-1)*X(14)-k4*X(14);
dxdt(15)= k1*((1/C_1)*X(5))*X(11)+k1*((X(1)/C_1)-1)*X(15)-k4*X(15)-X(11);
dxdt(16)= -k3*(1-(X(6)/C_2))*X(16)+k4*X(11);
dxdt(17)= k3*((1/C_2)*X(7))*X(16)+k3*((X(6)/C_2)-1)*X(17)+k4*X(12);
dxdt(18)= k3*((1/C_2)*X(8))*X(16)+k3*((X(6)/C_2)-1)*X(18)+k4*X(13);
dxdt(19)= ((X(6)/C_2)-1)*(X(16)+k3*X(19))+k3*((1/C_2)*X(9))*X(16)+k4*X(14);
dxdt(20)= k3*((1/C_2)*X(10))*X(16)+k3*((X(6)/C_2)-1)*X(20)+k4*X(15)+X(11);
dxdt = [dxdt(1),dxdt(2),dxdt(3),dxdt(4),dxdt(5),dxdt(6),dxdt(7),dxdt(8),...
   dxdt(9),dxdt(10),dxdt(11),dxdt(12),dxdt(13),dxdt(14),dxdt(15),...
   dxdt(16),dxdt(17),dxdt(18),dxdt(19),dxdt(20)];
end

function [t,Y] = MYRK4COSensNew(X,t0,tf,dt)
% This Matlab code runs the 4th-order Runge-Kutta method for the extended
% system of 20 ODEs defined in COSens.m for the Sickle Cell model
% Calling the xx (derivative) function
% dxdt = COSens(X);
% Storing results
n_max = floor((tf-t0)/dt);
% Storing the initial condition
Y = X;
t = t0;
for n = 1:1:n_max
   K1 = COSensNew(X);
   K2 = COSensNew(X+dt/2*K1);
   K3 = COSensNew(X+dt/2*K2);
   K4 = COSensNew(X+dt*K3);
   Yt = X+(dt/6)*(K1+2*K2+2*K3+K4);
   X = Yt;
   Y = [Y;Yt];
   t = [t;n*dt];
end
end
```

Matlab codes for MPSA

The main program for the MPSA

```
%Initialization of the parameter of the model
```

```
k1=.028;
k2=.07;
k3=.1;
k4=.01;
K = [k1 k2 k3 k4];
C_1=.4;
C_2=.1;
t0=0;
tf=250;
dt=0.01;
%Calling My runge-kutta 4 method to compute the original (nominal)
%behaviour
[t,Y] =pMYRK4COSensMPSA1([0.0036,0.0,.1750,0],K,t0,tf,dt,C_1,C_2);
%To compute “x” column must = 1, “y” column must = 2, “z” column
%must = 3 and “u” column must equal 4.
column = 1;  % The considered output is x, the 1st variable
product=squeeze(Y(:,column,:))';
figure(5)
hold on; grid on;
plot(t,squeeze(Y(:,1,:)),'b')
plot(t,squeeze(Y(:,2,:)),'r')
plot(t,squeeze(Y(:,3,:)),'g')
plot(t,squeeze(Y(:,4,:)),'k')
legend('De-oxy Monomers[x(1)]','CO-bound Monomers [x(2)]',...
'De-oxy polymers [x(3)]','CO-bound Polymers [x(4)]');
title('Molar Cencentration') % put a title on the plot
xlabel('Time [min]')%  label the x-axis
ylabel('Population [Mm]') %  label the y-axis
%is the number of the point we would like to use for the
%simulation
%% n = 1000000;  original setting
n = 10000;
K_init = 0.5 * K;
K_fin = 2 * K;
K_mat = zeros(n,length(K));
%Creates a uniform random number distribution between the set of 2 points
for i = 1:length(K)
   K_mat(:,i) = random('unif',K_init(i),K_fin(i),n,1);
end
%Stores the objective function of the simulation
Allfobj=zeros(n,1);
%Simulation
%% BlockSize = 10000; original setting
BlockSize = 1000;
Blocks = n / BlockSize;
initX = repmat([0.0036,0.0,0.1750,0.0],BlockSize,1);
```

```
for b = 1:Blocks
   b
   tic
   blockK = K_mat((b-1)*BlockSize+1:b * BlockSize,:);
   Allfobj((b-1)*BlockSize+1:b * BlockSize) = ...
      pMYRK4COSensMPSA(initX,blockK,t0,tf,dt,C_1,C_2,column,product);
   toc
end
%Store the sum of the total objective function
sum_fobj = sum(Allfobj);
% amin = min(Allfobj)
% amax = max(Allfobj)
% asize = size(Allfobj)
save data_5;
%Calculates the criterion value
%% Compute the criterion
criterion = prctile(Allfobj,[25,37.5,50,62.5,75]); %sum_fobj/n
criterion = [criterion,sum_fobj/n];
%% Plot out the CDF for four parameters
h = [];
p = [];
rank = [];
for i=1:1:4
   h2 = [];
   p2 = [];
   covk2 = [];
   rank2 = [];
   for j=1:6
      criterion1 = criterion(j);
      mar_1 = K_mat(Allfobj <= criterion1,i);
      [f1,x1] =ecdf(mar_1);
      [n1,y1] = hist(mar_1);
      mar_2 = K_mat(Allfobj > criterion1,i);
      [f2,x2] =ecdf(mar_2);
      [n2,y2] = hist(mar_2);
      [h1,p1,rank1]= kstest2(mar_1,mar_2,0.05);
      h2 = [h2;h1];
      p2 = [p2;p1];
      rank2 = [rank2;rank1];
      figure(i);
      hold on; grid on;
      subplot(3,2,j)
      hold on; grid on;
      plot(x1,f1,'r-',x2,f2,'b-');
      %legend('Acceptable','Unacceptable');
      figure(i+4)
```

```
        hold on; grid on;
        subplot(3,2,j)
        hold on; grid on;
        plot(y1,n1,'rx-',y2,n2,'bx-');
        %legend('Acceptable (Red)','Unacceptable(Blue)');
    end
    h = [h,h2];
    p = [p,p2];
    rank = [rank,rank2];
end
figure(1);
subplot(3,2,1)
hold on; grid on;
xlabel('Parameter k1');
ylabel('25-prctile Obj ECDFs');
subplot(3,2,2)
hold on; grid on;
xlabel('Parameter k1');
ylabel('37.5-prctile Obj ECDFs');
subplot(3,2,3)
hold on; grid on;
xlabel('Parameter k1');
ylabel('50-prctile Obj ECDFs');
subplot(3,2,4)
hold on; grid on;
xlabel('Parameter k1');
ylabel('62.5-prctile Obj ECDFs');
subplot(3,2,5)
hold on; grid on;
xlabel('Parameter k1');
ylabel('75-prctile ObjECDFs');
subplot(3,2,6)
hold on; grid on;
xlabel('Parameter k1');
ylabel('Mean Obj ECDFs');
suptitle('Acceptable (Red) and Unacceptable (Blue) ECDFs of k1');
figure(2);
subplot(3,2,1)
hold on; grid on;
xlabel('Parameter k2');
ylabel('25-prctile Obj ECDFs');
subplot(3,2,2)
hold on; grid on;
xlabel('Parameter k2');
ylabel('37.5-prctile Obj ECDFs');
subplot(3,2,3)
```

```
hold on; grid on;
xlabel('Parameter k2');
ylabel('50-prctile Obj ECDFs');
subplot(3,2,4)
hold on; grid on;
xlabel('Parameter k2');
ylabel('62.5-prctile Obj ECDFs');
subplot(3,2,5)
hold on; grid on;
xlabel('Parameter k2');
ylabel('75-prctile ObjE CDFs');
subplot(3,2,6)
hold on; grid on;
xlabel('Parameter k2');
ylabel('Mean Obj ECDFs');
suptitle('Acceptable (Red) and Unacceptable (Blue) ECDFs of k2');
figure(3);
subplot(3,2,1)
hold on; grid on;
xlabel('Parameter k3');
ylabel('25-prctile Obj ECDFs');
subplot(3,2,2)
hold on; grid on;
xlabel('Parameter k3');
ylabel('37.5-prctile Obj ECDFs');
subplot(3,2,3)
hold on; grid on;
xlabel('Parameter k3');
ylabel('50-prctile Obj ECDFs');
subplot(3,2,4)
hold on; grid on;
xlabel('Parameter k3');
ylabel('62.5-prctile Obj ECDFs');
subplot(3,2,5)
hold on; grid on;
xlabel('Parameter k3');
ylabel('75-prctile Obj ECDFs');
subplot(3,2,6)
hold on; grid on;
xlabel('Parameter k3');
ylabel('Mean Obj ECDFs');
suptitle('Acceptable (Red) and Unacceptable (Blue) ECDFs of k3');
figure(4);
subplot(3,2,1)
hold on; grid on;
xlabel('Parameter k4');
```

```
ylabel('25-prctile Obj ECDFs');
subplot(3,2,2)
hold on; grid on;
xlabel('Parameter k4');
ylabel('37.5-prctile Obj ECDFs');
subplot(3,2,3)
hold on; grid on;
xlabel('Parameter k4');
ylabel('50-prctile Obj ECDFs');
subplot(3,2,4)
hold on; grid on;
xlabel('Parameter k4');
ylabel('62.5-prctile Obj ECDFs');
subplot(3,2,5)
hold on; grid on;
xlabel('Parameter k4');
ylabel('75-prctile Obj ECDFs');
subplot(3,2,6)
hold on; grid on;
xlabel('Parameter k4');
ylabel('Mean Obj ECDFs');
suptitle('Acceptable (Red) and Unacceptable (Blue) ECDFs of k4');
figure(5);
subplot(3,2,1)
hold on; grid on;
xlabel('Parameter k1');
ylabel('25-prctile Obj PMFs');
subplot(3,2,2)
hold on; grid on;
xlabel('Parameter k1');
ylabel('37.5-prctile Obj PMFs');
subplot(3,2,3)
hold on; grid on;
xlabel('Parameter k1');
ylabel('50-prctile Obj PMFs');
subplot(3,2,4)
hold on; grid on;
xlabel('Parameter k1');
ylabel('62.5-prctile Obj PMFs');
subplot(3,2,5)
hold on; grid on;
xlabel('Parameter k1');
ylabel('75-prctile Obj PMFs');
subplot(3,2,6)
hold on; grid on;
xlabel('Parameter k1');
```

```
ylabel('Mean Obj PMFs');
suptitle('Acceptable (Red) and Unacceptable (Blue) PMFs of k1');
figure(6);
subplot(3,2,1)
hold on; grid on;
xlabel('Parameter k2');
ylabel('25-prctile Obj PMFs');
subplot(3,2,2)
hold on; grid on;
xlabel('Parameter k2');
ylabel('37.5-prctile Obj PMFs');
subplot(3,2,3)
hold on; grid on;
xlabel('Parameter k2');
ylabel('50-prctile Obj PMFs');
subplot(3,2,4)
hold on; grid on;
xlabel('Parameter k2');
ylabel('62.5-prctile Obj PMFs');
subplot(3,2,5)
hold on; grid on;
xlabel('Parameter k2');
ylabel('75-prctile Obj PMFs');
subplot(3,2,6)
hold on; grid on;
xlabel('Parameter k2');
ylabel('Mean Obj PMFs');
suptitle('Acceptable (Red) and Unacceptable (Blue) PMFs of k2');
figure(7);
subplot(3,2,1)
hold on; grid on;
xlabel('Parameter k3');
ylabel('25-prctile Obj PMFs');
subplot(3,2,2)
hold on; grid on;
xlabel('Parameter k3');
ylabel('37.5-prctile Obj PMFs');
subplot(3,2,3)
hold on; grid on;
xlabel('Parameter k3');
ylabel('50-prctile Obj PMFs');
subplot(3,2,4)
hold on; grid on;
xlabel('Parameter k3');
ylabel('62.5-prctile Obj PMFs');
subplot(3,2,5)
```

```
hold on; grid on;
xlabel('Parameter k3');
ylabel('75-prctile Obj PMFs');
subplot(3,2,6)
hold on; grid on;
xlabel('Parameter k3');
ylabel('Mean Obj PMFs');
suptitle('Acceptable (Red) and Unacceptable (Blue) PMFs of k3');
figure(8);
subplot(3,2,1)
hold on; grid on;
xlabel('Parameter k4');
ylabel('25-prctile Obj PMFs');
subplot(3,2,2)
hold on; grid on;
xlabel('Parameter k4');
ylabel('37.5-prctile Obj PMFs');
subplot(3,2,3)
hold on; grid on;
xlabel('Parameter k4');
ylabel('50-prctile Obj PMFs');
subplot(3,2,4)
hold on; grid on;
xlabel('Parameter k4');
ylabel('62.5-prctile Obj PMFs');
subplot(3,2,5)
hold on; grid on;
xlabel('Parameter k4');
ylabel('75-prctile Obj PMFs');
subplot(3,2,6)
hold on; grid on;
xlabel('Parameter k4');
ylabel('Mean Obj PMFs');
suptitle('Acceptable (Red) and Unacceptable (Blue) PMFs of k4');
h1 = figure(1);
h2 = figure(2);
h3 = figure(3);
h4 = figure(4);
h5 = figure(5);
h6 = figure(6);
h7 = figure(7);
h8 = figure(8);
saveas(h1,'AccepRejectECDF4K1','jpg');
saveas(h2,'AccepRejectECDF4K2','jpg');
saveas(h3,'AccepRejectECDF4K3','jpg');
saveas(h4,'AccepRejectECDF4K4','jpg');
saveas(h5,'AccepRejectPMF4K1','jpg');
saveas(h6,'AccepRejectPMF4K2','jpg');
saveas(h7,'AccepRejectPMF4K3','jpg');
saveas(h8,'AccepRejectPMF4K4','jpg');
display('The sensitivity ranks at 5% significance level are:');
rank
fileID = fopen('SensivityRank.txt','w');
fprintf(fileID,'%10s %12s %12s %12s\r\n','k1','k2',...
   'k3','k4');
fprintf(fileID,'%12.8f %12.8f %12.8f %12.8f\r\n',rank');
fclose(fileID);
display('The K-S test outcomes at 5% significance level are:');
h
fileID = fopen('KSTestResult.txt','w');
fprintf(fileID,'%10s %12s %12s %12s\r\n','Null or Alt','Null or Alt',...
   'Null or Alt','Null or Alt');
fprintf(fileID,'%12.8f %12.8f %12.8f %12.8f\r\n',h');
fclose(fileID);
display('The K-S test asymptotic p-values at 5% significance level are:');
p
fileID = fopen('pValue.txt','w');
fprintf(fileID,'%10s %12s %12s %12s\r\n','p-vlaue','p-vlaue',...
   'p-vlaue','p-vlaue');
fprintf(fileID,'%12.8f %12.8f %12.8f %12.8f\r\n',p');
fclose(fileID);
```

<u>The subroutines for the MPSA</u>

```
function dxdt = pCOSensMPSA(X,K,C_1,C_2)
   n = size(K,1);
   dxdt = zeros(n,4);
   T1 =  K(:,2).*X(:,1);
   T2 =  K(:,4).*X(:,3);
   T3 =  K(:,1).*(X(:,1)/C_1-1).*X(:,3);
   T4 = -K(:,3).*(X(:,2)/C_2-1).*X(:,4);
   dxdt(:,1)= -T3 - T1;
   dxdt(:,2)=  T4 + T1;
   dxdt(:,3)=  T3 - T2;
   dxdt(:,4)= -T4 + T2;
end

function difference = pMYRK4COSensMPSA(X,K,t0,tf,dt,C_1,C_2,column,product)
n_max = floor((tf-t0)/dt);
difference = (X(:,column)-product(1)).^2;%/1000;
t = zeros(n_max+1,1); t(1) = t0;
for n = 1:1:n_max
   K1 = pCOSensMPSA(X,K,C_1,C_2);
```

```
    K2 = pCOSensMPSA(X+dt/2*K1,K,C_1,C_2);
    K3 = pCOSensMPSA(X+dt/2*K2,K,C_1,C_2);
    K4 = pCOSensMPSA(X+dt*K3,K,C_1,C_2);
    X = X + (K1+2*K2+2*K3+K4)*(dt/6);
    difference = difference + ((X(:,column)-product(n+1)).^2);%/1000;
    t(n+1) = n * dt;
end
end
```