THE FUZZY LOGIC CONTROL OF AN

ACTIVE VEHICLE SUSPENSION

by

ANDREW J. BARR
B.S.M.E., Grove City College, 1994

Submitted in Partial Fulfillment of the Requirement

for the Degree of

Master of Science in Engineering

in the

Mechanical Engineering

Program

YOUNGSTOWN STATE UNIVERSITY

January, 1996

THE FUZZY LOGIC CONTROL OF AN

ACTIVE VEHICLE SUSPENSION

ANDREW J. BARR

Signature:

Andrew J. Barr 1-23-96

Student Date

Approvals:

1-23-96

Thesis Advisor Date

1-23-96

Committee Member Date

1-23-96

Committee Member Date

1-23-96

Committee Member Date

1/24/96

Dean of Graduate Studies Date

ABSTRACT

The object of this research is to design, simulate and test a fuzzy logic controller for an active vehicle suspension. First, the necessary background for fuzzy logic and fuzzy logic control are presented. Next, the background information on a 2 Degree of Freedom quarter-car suspension model is examined followed by the design of the fuzzy logic controller. The simulation is completed by using Matlab, a high-performance numeric computation and visualization software package. For comparison purposes, a passive suspension and a Linear-Quadratic-Gaussian (LQG) controlled suspension are also simulated. The three controllers are subjected to five different analytical road input conditions: step, hump-back bridge, ramp, random linear power spectral density and white noise random. For all five inputs, the LQG and fuzzy logic suspension exhibit superior ride characteristics when compared to the passive suspension. The LQG and fuzzy logic perform similarly when subjected to the first three distinct input conditions. However for the random inputs, the fuzzy logic suspension shows modest improvements over the LQG suspension.

ACKNOWLEDGMENTS

I would like to thank my family and friends for there support during this challenging endeavor. I would also like to thank that my committee members, Dr. Ray, Dr. Foulkes, Dr. Tarantine and Dr. Suchora, for taking time out of there busy schedules to critique my thesis. In particular, Dr. Ray for his guidance throughout the project and Dr. Foulkes for his countless hours spent giving me a crash course on LQG control.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Symbol	Quantity	Units
a	bridge elevation, ramp height	m
A	amplitude	m
B	damper Rate	N/m/s
d	bridge length, ramp length	m
f	frequency	Hz
F	variable Actuator Force	N
K_s	main spring rate	N/m
K_u	tire spring rate	N/m
M_s	sprung mass	kg
M_u	unsprung mass	kg
R_c	sensor parameter	m^2
s	power spectral density	m^2/Hz
t	time	s
u	output control force	N
v	velocity	mph
V	velocity	m/s

CHAPTER 1

1. Introduction

Automobile suspensions have for years consisted of a coil or leaf spring in parallel with a viscous damper. Inherent to these passive systems are a natural trade-off between ride comfort and road handling characteristics. Soft springs facilitate good ride characteristics at the expense of increased wheel motion and increased variations in dynamic tire loadings on rough roads. On the flip-side, good road handling characteristics and smaller wheel motion is an attribute of larger spring rates. Therefore, spring rates must be chosen large enough to limit wheel motion and dynamic tire variations yet small enough to provide a comfortable ride. Through many years of experimentation and testing, this arrangement has evolved into a near optimal design. Hence, to further improve suspensions, active elements must be introduced. It is the controlling of the active suspension that will be the focus of this research.

Fuzzy logic controllers have been used in many applications, such as: cruise control, automatic transmissions, Sendai subway operation, cold-rolling mills, self-parking model car, image stabilizer for video camera and a fully automated washing machine [14, 19]. Given the proven diversity of fuzzy logic control, this technique was selected to control the active suspension. In [3], Cherry used a fuzzy logic controller to control a 47

Degree of Freedom (DOF) multibody automotive suspension. By modeling the entire car, Cherry could analyze the vertical, pitch and roll response to selected inputs. For this research only vertical response will be evaluated using a 2 DOF model. Sufficient for developmental testing, a 2 DOF model incorporates the most important suspension characteristics while keeping computations simple. Ro [17] uses a 2 DOF suspension model for the analysis of his fuzzy logic controller. Differing from Ro, a much simpler rule base along with a wider range of input conditions is used for this study. The performance of the fuzzy logic active suspension was compared to a passive suspension and to a benchmark active suspension.

To develop the benchmark suspension controller an optimal control technique, Linear-Quadratic-Gaussian control, was used. Linear-Quadratic-Gaussian (LQG) techniques are one of the most popular forms of control for active suspensions. Much research has been completed in this area with examples of LQG suspension controllers given in [8, 15, 16, 22, 23, 24].

Five different road input conditions: step, humped-back bridge, ramp, random linear power spectral density, and random white noise, were tested on the three different suspensions: passive, LQG and fuzzy logic. The simulations were competed using Matlab's dynamic system simulation software, SIMULINK [18]. The fuzzy logic controller was designed using Matlab's Fuzzy Logic Toolbox [6] while the LQG controller design used Matlab's Control System Toolbox [4].

Chapter 2 introduces the basic concepts of fuzzy logic. Chapter 3 provides the steps necessary to develop a fuzzy logic controller. Chapter 4 gives background information on the suspension model. Chapter 5 presents both the LQG and fuzzy logic active controller designs. Chapter 6 discusses the simulation process and input conditions. Chapter 7 explains the results from the simulations. Lastly, Chapter 8 presents conclusions and recommendations for additional research based on the results.

CHAPTER 2

2. Fuzzy Logic

This chapter provides an overview of fuzzy logic needed to develop a fuzzy logic controller. The key aspects of fuzzy logic are fuzzy sets, membership functions, linguistic variables, fuzzy rules and fuzzy reasoning. These topics will be discussed with formal definitions and specific examples given when possible.

2.1 Fuzzy Sets

To develop the idea of a fuzzy set, a classical set is first examined:

$$C = \{x | x > 6\} \tag{1}$$

where if x is greater than 6 then the number belongs to the set C. If x is equal to or less than 6 then the number does not belong to the set. A fuzzy set, on the other hand, does not contain a distinct or crisp boundary where x does or does not belong to a set.

A formal definition of a fuzzy set is, if X is a collection of objects denoted generically by x then a fuzzy set A in X is a set of ordered pairs:

$$A = \{(x, \mu_A(x)) | x \in X\} \tag{2}$$

where $\mu_A(x)$ is called the grade of membership or membership function of x in A. The membership function maps each element in the domain X to a membership grade between

0 and 1. As a discrete example, let $X = \{50, 60, 70, 80, 90, 100\}$ be a set of numbers describing the temperature of an engineering system in Fahrenheit. The fuzzy set A = "hot" may then be shown as $A = \{(50,0.0),(60,0.1),(70,0.3),(80,0.6),(90,0.9),(100,1.0)\}$.

The first number in each ordered pair is the temperature and the second number is the membership value for the fuzzy set A, "hot". Figure 1 is a graph of this discrete fuzzy set.

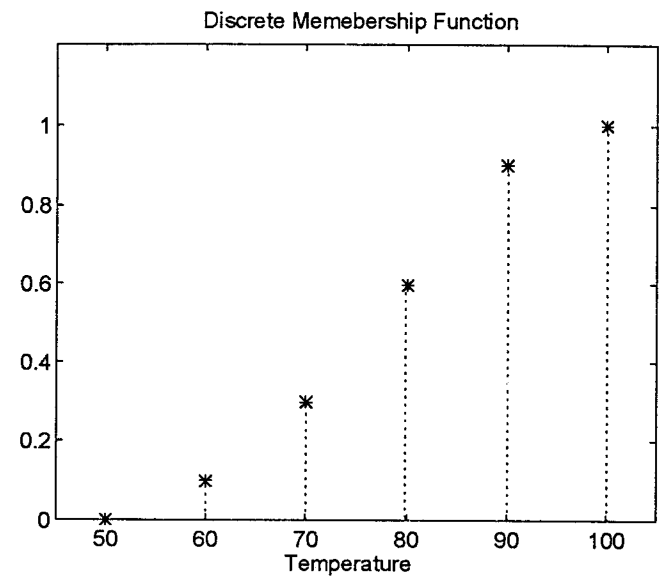

Figure 1. Discrete Membership Function

A continuous case for the same fuzzy set A can be demonstrated with a Gaussian membership function:

$$\mu_A(x) = e^{-\frac{(x-110)^2}{2*22.5^2}} \quad (3)$$

The graph of the continuous case is shown in Figure 2.

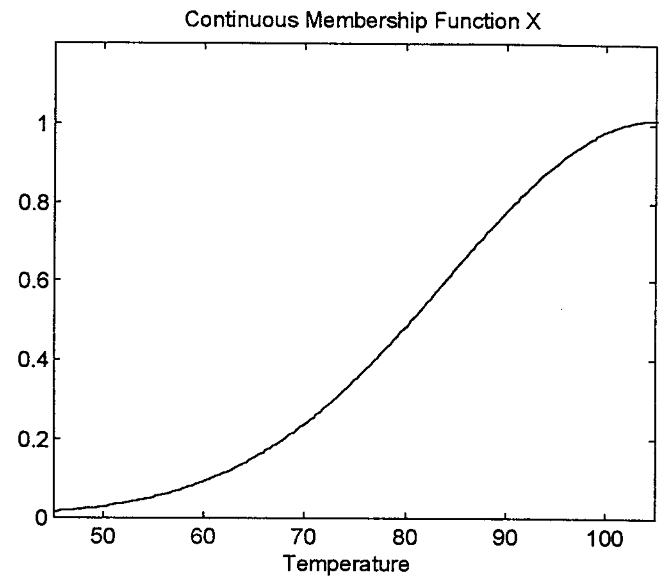

Figure 2. Continuous Membership Function

The classical set, C, given in equation 1 can also thought of as a fuzzy set described by the following membership function:

$$\mu_C = \begin{cases} 0 \text{ if } x \leq 6 \\ 1 \text{ if } x > 6 \end{cases} \tag{4}$$

This shows the crisp, abrupt boundary in the classical set compared to the fuzzy, gradual boundary in a fuzzy set.

An often used method for denoting a fuzzy set A is:

$$A = \sum_{x_i \in X} \mu_A(x_i)/x_i, \text{ if X is discrete.} \tag{5a}$$

$$A = \int_X \mu_A(x)/x \text{ if X is Continuous} \tag{5b}$$

Using this notation the previous examples would be represented as:

$$A = 0.0/50 + 0.1/60 + 0.3/70 + 0.6/80 + 0.9/90 + 1.0/100 \tag{6a}$$

$$A = \int \exp\left(-\frac{(x-110)^2}{2*22.5^2} \right) / x \tag{6b}$$

Since a fuzzy set is characterized by a membership function, it is only natural that operations on fuzzy sets are defined in terms of membership functions [26]. Let A and B be two fuzzy sets, with membership functions μ_A and μ_B, respectively; then the union of A and B is a fuzzy set C, written $C = A \cup B$ or $C = A$ OR B. The subsequent membership function μ_C is defined by:

$$\mu_C(x) = max\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \vee \mu_B(x). \tag{7}$$

Similarly the intersection of A and B is a fuzzy set C, $C = A \cap B$ or $C = A$ AND B with μ_C defined by:

$$\mu_C(x) = min\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \wedge \mu_B(x). \tag{8}$$

The intersection of A and B can also be represented by the algebraic product defined by:

$$\mu_C(x) = \{\mu_A(x)\mu_B(x)\}. \tag{9}$$

The Complement of A, denoted by $\overline{A}$, can be defined as:

$$\mu_{\overline{A}} = 1 - \mu_A(x). \tag{10}$$

Other fuzzy operators have been defined but are not necessary for our application and can be found in any fuzzy logic text or paper [10, 11, 25, 26].

2.2 Membership Functions

The idea of a membership functions was previously introduced in the section 2.1. An important design consideration is the shape of membership functions. Many different membership functions exist, but the most basic ones are the triangle, trapezoidal and

Gaussian membership functions. Given three parameters *{a,b,c}*, the triangle membership function is:

$$triangle(x;a,b,c) = max\left(min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right) \tag{11}$$

where a and c are the feet of the triangle and b is the peak. The trapezoidal membership function is defined by four parameters *{a,b,c,d}*:

$$trapezoidal(x;a,b,c,d) = max\left(min\left(\frac{x-a}{b-a},1,\frac{d-x}{d-b}\right),0\right) \tag{12}$$

where a and d locate the feet of the trapezoid and b and c locate the shoulders. Lastly, the Gaussian membership function defined the by two parameters $\{\sigma,c\}$ is:

$$gaussian(x;\sigma,c) = e^{-\frac{(x-c)^2}{2\sigma^2}} \tag{13}$$

where c locates the center and σ defines the width. Other membership functions can be found in [6, 10].

The shape of the membership functions is determined using expert knowledge from an examination of the system. Generally the width and overlap of the membership function will influence the systems response to noise and external disturbance [12], but there are no strict rules or equations to develop an optimized membership function. Recently, membership functions have been shaped by using neural networks and training data [9, 10].

2.3 Linguistic Variables

Linguistic variables are used to describe a particular characteristic of a system. A linguistic variable is defined by a quintuple (x, T(x), U, G, M) where x is the name of the variable; T(x) is the set of names of linguistic values describing the variable x over the universe of discourse U; G is a syntactic rule in the form of grammar for generating the name, X, of values of x; and M is a semantic rule for associating with each X its meaning. Each particular syntactic rule generated by G is referred to as a term. As an example, if temperature was the linguistic variable x and U = [30, 90], then the term set might be T(temperature) = {cold, comfortable, hot}. M is a rule that assigns a fuzzy set or membership function to a specific term; e.g.,

$$M(\text{cold}) = \{(u, \mu_{\text{cold}}(u)) \mid u \in U\} \quad (14)$$

G(x) is a rule designed by using expert knowledge, which generates the labels of the terms i.e., cold comfortable, and hot [26]. A graph of this linguistic variable temperature is shown in Figure 3. It is noted [12] that in our daily life most of our decisions are based on linguistic information rather than numerical values; thus the use of linguistic variables is an ideal way to characterize human behavior and decision analysis.

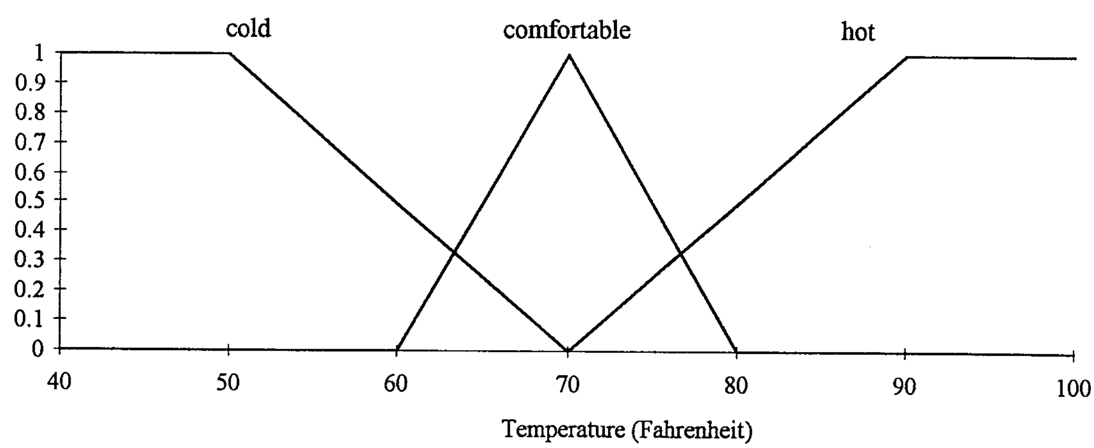

Figure 3. Diagram of Linguistic Variable Temperature

2.4 Fuzzy Rules

Fuzzy rules, also called fuzzy if-then rules, fuzzy conditional statements, or fuzzy implication are explained by using the following example given in Zadeh [25]. Looking at the quantitative relationships between the crisp variables x and y,

x	y
5	10
8	16

These variables described using words becomes:

IF x IS 5 THEN y IS 10

IF x IS 8 THEN y IS 16

The same technique applied to the fuzzy linguistic variables x and y is:

x	y
small	negative big
large	positive small

As before using IF THEN statements,

IF x IS *small* THEN y IS *negative big*

IF x IS *large* THEN y IS *positive small*

Statements of this type are known rather appropriately as conditional statements with formal definition,

if x is A then y is B

or symbolically:

$$A \rightarrow B \tag{15}$$

where A and B are linguistic values defined by fuzzy sets on universes of discourse X and Y, respectively. The antecedent or premise is "*x is A*" while "*y is B*" is the consequence or conclusion.

The relation $A \rightarrow B$ represents a relation between two variables x and y. Using this idea, a fuzzy rule is defined as a binary fuzzy relation R found by using the Cartesian product depicted as,

$$R_{A \rightarrow B} = A \times B. \tag{16}$$

The Cartesian product is defined by the min operator,

$$A \times B = \int_{X \times Y} \frac{\min(\mu_A(x), \mu_B(y))}{(x,y)} \tag{17}$$

or the algebraic operator

$$A \times B = \int_{X \times Y} \frac{\mu_A(x), \mu_B(y)}{(x,y)}. \tag{18}$$

Thus the Cartesian product $X \times Y$ is characterized by the membership function $\mu_R(x,y)$ where each element $(x,y) \in X \times Y$ [14].

2.5 Fuzzy Reasoning

Fuzzy reasoning or approximate reasoning is an inference procedure used to derive conclusions from a set of fuzzy rules. Fuzzy rules describe information about the consequence when the input matches the antecedent, however if the input does not precisely match the antecedent then some method must be used to find an output from the fuzzy rule. Modus ponens is the traditional two-valued logic that allows us to infer the truth of a proposition B, from the truth of A and the statement IF x IS A THEN y IS B.

However, what happens if x is not exactly A but rather A'? Generalized modus ponens (GMP) is an inference procedure used to resolve this conflict. Normally referred to as fuzzy reasoning, the following example [12] illustrates the results,

premise 1:	x is A'
premise 2:	if x is A then y is B,
consequence:	y is B',

where A' is close to A and B' is close to B. This shows that even though x is not exactly A, premise 2 still can be applied and the consequence y is B' is true. The inference procedure of fuzzy reasoning for the preceding example can be defined by the following. If A, A', and B are fuzzy sets of X, X, and Y, respectively and A→B is a fuzzy implication expressed as a fuzzy relation R on $X \times Y$, GMP can be extended to fuzzy logic defining the compositional rule of inference as,

$$y = x \circ R \tag{19}$$

If the sup-star compositional operator is used, the compositional rule of inference can be calculated as:

$$x \circ R = \left\{\left[v, \sup\left(\mu_x(u) * \mu_R(u,v)\right)\right], u \in U, v \in V\right\} \tag{20}$$

where *sup* is the supremum or max over V and star, $*$, is either the minimum or the algebraic product operation. Thus max-min composition would be defined as:

$$\begin{aligned} \mu_{B'}(y) &= \max_x \min\left[\mu_{A'}(x), \mu_R(x,y)\right] \\ &= \vee\left[\mu_{A'}(x) \wedge \mu_R(x,y)\right] \end{aligned} \tag{21}$$

and max-product defined as,

$$\begin{aligned} \mu_{B'}(y) &= \max_x\left[\mu_{A'}(x)\mu_R(x,y)\right] \\ &= \vee\left[\mu_{A'}(x)\mu_R(x,y)\right] \end{aligned} \tag{22}$$

Examples of max-min composition and max-product composition are given in section 3.4.

CHAPTER 3

3. Design of a Fuzzy Logic Controller

A fuzzy logic controller can be designed by using the concepts introduced in Chapter 2. Due to the many disciplines where fuzzy logic control systems have been used, a fuzzy logic controller is also referred to in literature as a fuzzy inference system, fuzzy-rule-based system, fuzzy expert system, fuzzy model, fuzzy associative memory or just fuzzy system.

The key steps involved in designing a fuzzy logic controller are examined. They consist of, defining input and output variables, a knowledge base, fuzzy reasoning inference and a defuzzification procedure. Figure 4 shows the basic elements of a fuzzy logic controller.

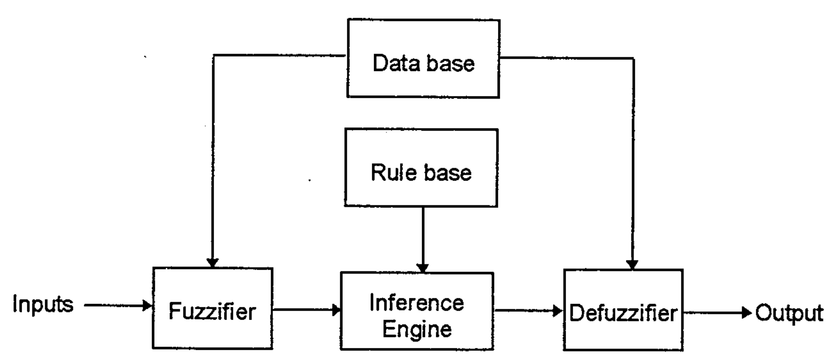

Figure 4. Fuzzy Logic Controller

3.1 Input and Output Variables

For a particular system the necessary input and output variables must be defined. No predefined procedure exists for determining the correct variables and relies mainly on a thorough understanding of the plant being controlled and experimental procedures. If a controller for the plant has been previously developed, this is often a good starting point for selecting the fuzzy input and output variables.

3.2 Knowledge Base

The knowledge base of a fuzzy control system consists of two parts: a data base, which defines linguistic variables and their corresponding membership functions, and a rule base, which uses the data base in the development of the fuzzy if-then rules.

3.2.1 Data Base

The data base consists of a set of membership functions that define each input and output variable. To arrive at a data base for a fuzzy controller, the universe of discourse or interval spanned by each variable is partitioned into a number of fuzzy subsets with a linguistic label applied to each subset. For example if the input to a fuzzy controller is acceleration, then the corresponding linguistic variable may be partitioned into fuzzy subsets named, *Positive Big* (PB), *Positive Medium* (PM), *Positive Small* (PS), *Zero* (ZO), *Negative Small* (NS), *Negative Medium* (NM) and *Negative Big* (NB). The number of fuzzy subsets must be decided by using expert knowledge and or experimentation. Additional fuzzy subsets improve control sensitivity at the expense of additional computational complexity.

3.2.2 Rule Base

A rule base is a group of fuzzy rules that define the controllers output response given a particular set of inputs. The input conditions are the antecedents and the output is the consequence of the fuzzy rules [2]. Depending on the system being controlled, anywhere from 9 rules to control an inverted pendulum, to as many as 105 rules may be necessary to control a truck and trailer system.

Lee gives four ways to derive fuzzy control rules [12]:

1. Use expert experience and control engineering knowledge.
2. Based on operator's control actions.
3. Based on the fuzzy model of a process.
4. Based on learning through experimentation.

The most popular method is to use expert experience and control engineering knowledge. This procedure allows fuzzy control rules to relate state variables in the antecedents to process control variables in the consequence. However, it is necessary to have a general understanding of how a change in certain control input influences the system.

One other increasingly popular method for developing fuzzy control rules is via an adaptive fuzzy rule system. Normally called an Adaptive-Network-Based Fuzzy Inference System (ANFIS), this technique enables rules to be developed directly from training data. This procedure utilizes neural networks in the development of fuzzy control rules and the optimization of the corresponding membership functions.

3.3 Fuzzy Reasoning Inference

Fuzzy reasoning must next be applied to each fuzzy rule. Fuzzy reasoning takes the input values and applies them to each fuzzy rule and determines an output for each rule. This process uses the sup-star compositional operator to evaluate the compositional rule of inference with the two most popular methods being the max-min composition and the max-product composition. Graphical examples of these two methods are shown in the next section on defuzzification. When the sentence connective AND in the fuzzy rules is taken as the algebraic product operator, each input variable contributes to the weight of the output. This differs from the minimum operator where only the minimum variable contributes to the output. Therefore, the product operator tends to produce a smoother control action and recently has become more popular [20].

3.4 Defuzzification Procedure

Once the fuzzy output is computed, it must be defuzzified to a single value in the output universe of discourse z. The most popular method is the centroid of area, defined as:

$$z_{COA} = \frac{\int_Z \mu_{C'}(z) z dz}{\int_Z \mu_{C'}(z) dz}, \tag{23}$$

where $\mu_{C'}(z)$ is the aggregated output membership function. Many different defuzzification strategies can be found in [6, 20] and in general, there is no rigorous way to determine an optimum choice. Reproduced from [10], figures 5 and 6 show examples of a two rule system using max-min composition and max-prod composition. The rule statements would be

R_1: IF X is A_1 AND Y is B_1 THEN Z is C_1

R_2: IF X is A_2 AND Y is B_2 THEN Z is C_2

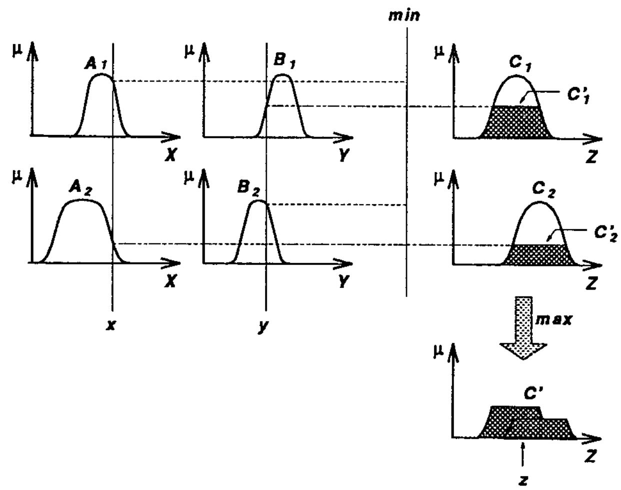

Figure 5. Max-min composition [10]

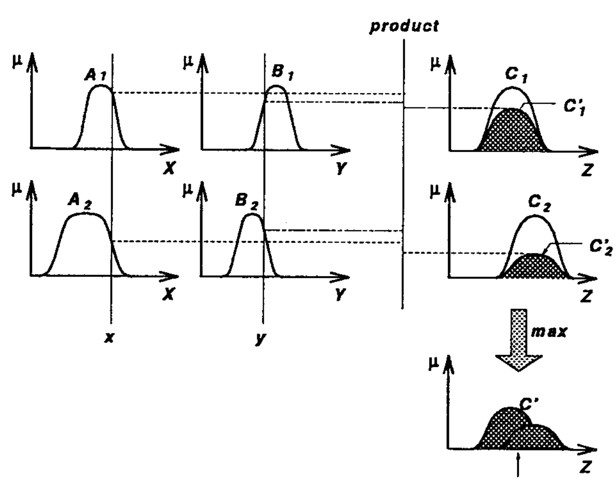

Figure 6. Max-prod composition [10]

For both composition styles the maximum is taken from each rule resulting in the specific area shown at the bottom of each figure. To get a crisp control output, the centroid of the final area is found using the previous defuzzification scheme.

CHAPTER 4

4. Background Information on the Model

For many years passive automotive suspension systems consisting of a coil or leaf spring in parallel with a viscous damper have existed. However, associated with these passive systems are major trade-offs between ride comfort and road handling characteristics. Comfortable rides require soft springs and larger suspension travel while good road handling characteristics require stiffer springs to decrease dynamic load variations between tires and the road. After many years of optimization the passive system has reached its performance limits and the development of active systems is essential for additional improvement. First, the different suspension types are examined followed by a description of the model used for this research. Lastly, a numerical procedure is developed to use for comparing suspensions subjected to random inputs.

4.1 Suspension Types

After many years of experimentation and testing, non-linear springs and dampers have been developed which optimize the characteristics of a passive suspension. Hence, it is necessary to look at active suspension components to further improve ride comfort and road handling. The first type of variable suspension component consists of a time-varying damper usually referred to as a semi-active suspension. Since they consist of a time-varying damper, semi-active suspensions offer significant benefits over passive suspensions while requiring small external power sources.

The next type would be an active suspension with a variable actuator. If the suspension is said to be "fully active" then actuators are used in place of the passive spring and damper units [5]. This type of suspension provides superior characteristics over both passive and semi-active systems; however, large power consumption and the ability to support the full body mass during static conditions leads to problems.

In [15, 24], active suspension models place the actuator in parallel with the passive spring and damper elements. This arrangement supports the vehicle during static loading, retains the high frequency response of the passive system and acts as a backup in case of active component failure. This type of active suspension will be used in this thesis. It should be noted that no distinct definition exists to identify between semi-active and active suspensions. These terms are applied loosely in literature with some suspensions labeled active but consisting of a variable damper.

4.2 Model Description

Many different vehicle models exist from very simple 1 DOF quarter-car models [8] to complex full car models with 47 DOF [3]. To what degree these simple or complex models represent actual vehicle responses can only be determined by actual vehicle tests. In [7] a 7 DOF model was compared to actual vehicle data with results showing that the model and test acceleration data agree up to a maximum frequency of 10 Hz. The previous reference also compared predicted acceleration responses from linear vehicle models of various levels of complexity. It was concluded that the differences between the models were small and that linear lower order models are sufficient for initial development and analysis of active controllers as demonstrated by numerous papers available on active controllers using 2 DOF models [5, 8, 15, 16, 17, 21, 22, 23, 24]. After successful application using simple models, then more elaborate models with nonlinearities and more DOF should be used. The final evaluation of the model's effectiveness would be actual vehicle tests.

The linear 2 DOF model used to test the passive, LQG and the fuzzy logic controller is shown in Figure 7.

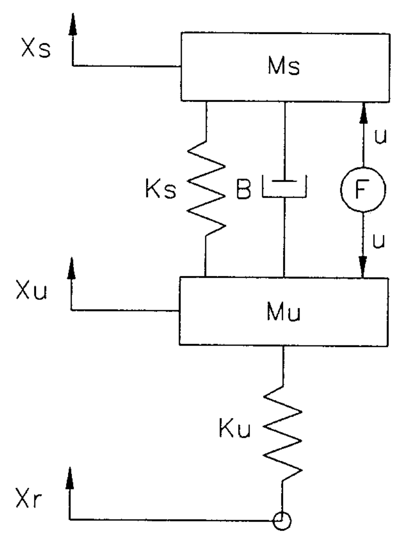

Figure 7. 2 DOF Active Suspension Model

The parameters for the model are from [15] and shown in the following table.

Table 1. Suspension Model Parameters

Description	Symbol	Units	Value
Sprung mass	M_s	kg	250
Unsprung mass	M_u	kg	30
Main spring rate	K_s	N/m	15,000
Tire spring rate	K_u	N/m	150,000
Damper rate	B	N/m/s	1,000
Variable Actuator	F	N	≤1,500

By applying Newton's law of motion to this system the following equations were derived to be solved using Matlab's dynamic system simulation software, SIMULINK.

$$M_u\ddot{x}_u = K_s(x_s - x_u) + K_u(x_r - x_u) + B(\dot{x}_s - \dot{x}_u) - u \tag{24}$$

$$M_s\ddot{x}_s = K_s(x_u - x_s) + B(\dot{x}_u - \dot{x}_s) + u \tag{25}$$

SIMULINK contains many blocks, such as integrators, gains and summers, that can be assembled to model the differential equations. SIMULINK then accesses Matlab's differential equation subroutines to obtain a solution. A 2 DOF model exhibits the unsprung mass mode often called "wheel-hop" and identified by light damping and a natural frequency between 8 and 12 Hz [7]. The sprung mass mode or principal body mode is usually around 1 Hz. Higher DOF models will include sprung mass pitch modes, heave modes and vehicle roll depending of the number of DOF. However, when looking at suspension ride and handling characteristics wheel hop mode and the sprung mass mode are the most important [7].

4.3 Performance Index

To evaluate the performance of a particular suspension, two methods were used. If the suspensions were subjected to a discrete input, then the sprung mass responses were compared for decreases in displacement, velocity and acceleration of the sprung mass. The decreased responses translate into increased ride characteristics of the suspension.

Next if the suspensions were subjected to a random input, an analytical comparison method was used. The analytical method was necessary since the sprung mass response was random making it difficult to draw substantial conclusions. A field study was completed in [7] which used 78 different passengers, 18 different road sections and 2 different vehicles to develop a correlation between subjective ride ratings and hard engineering values to be used for studies. The results showed that lower calculated root-mean-square (RMS) values for vertical acceleration at the driver's seat corresponded to improved subjective ride ratings. To rank the amount of suspension deflection for a particular suspension, the suspension deflection RMS value will be calculated for suspension design comparison as suggested in [8]. Lastly, a road handling related constraint will be introduced to take into account tire deflection or wheel-hop. Wheel-hop is important because substantial variations in tire deflection reduce tire normal forces which leads to loss of contact with the ground causing poor handling when braking or maneuvering the vehicle.

By attempting to minimize these three quantities a suspension may be developed that displays both good body isolation for improved ride characteristics and good road to tire contact producing excellent handling ability.

CHAPTER 5

5. Active Controller Design

This chapter describes the development of the fuzzy logic and LQG controller. The fuzzy logic controller was developed with the aid of Matlab's Fuzzy Logic Toolbox [6]. The key aspects introduced in Chapter 3 are applied to develop the fuzzy logic active suspension controller. The comparison LQG controller was based on Ray's model [15] and an overview of its development is also presented.

5.1 Active Fuzzy Logic Controller

First, appropriate input and output variables were decided upon. Using numerical simulation techniques, the optimum choice for state variables was the velocity and acceleration of the sprung mass [17]. The output of the controller was force, u, representative of the force of the actuator.

The universe of discourse for both the input and output variables was divided into three sections using the following linguistic variables, *positive* (P), *zero* (Z) and *negative* (N). The universe of discourse for the input variables was found by subjecting the passive suspension to several different input conditions and viewing the maximum and minimum values for each particular input variable. The universe of discourse for the output variable was chosen by using engineering judgment and reasonable maximum force for an actual actuator.

Triangular membership functions were initially used because they are very basic and widely used. The controller performed well, but was very quick to react to the slightest change in velocity or acceleration. This was due to the inherent sharpness in the triangular membership function's shape and the coarse partitioning of the universe of discourse. To alleviate this problem trapezoidal membership functions were used. They produced a smoother control action due to the flatness at the top of the trapezoid shape. The membership functions used for the controller are shown in Figures 8-10.

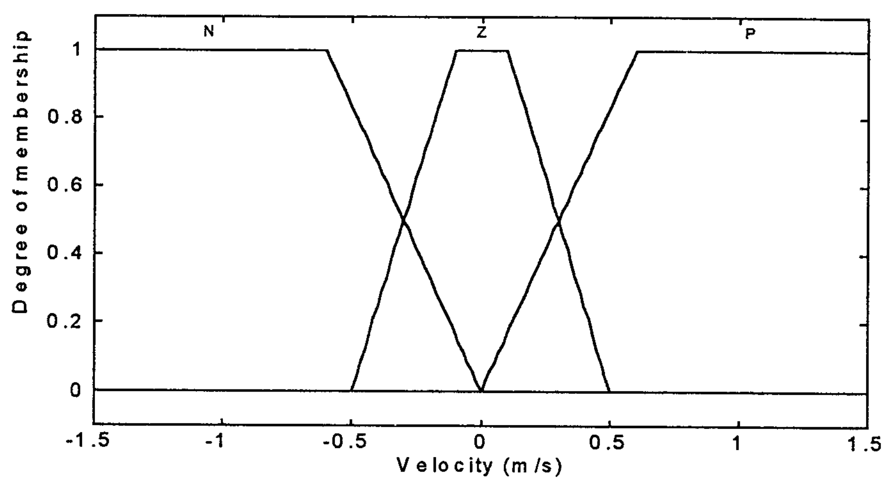

Figure 8. Input Velocity Membership Function

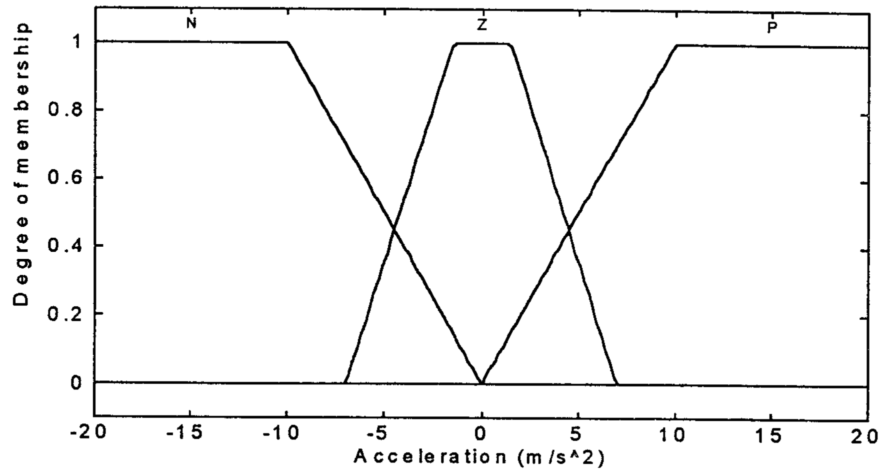

Figure 9. Input Acceleration Membership Function

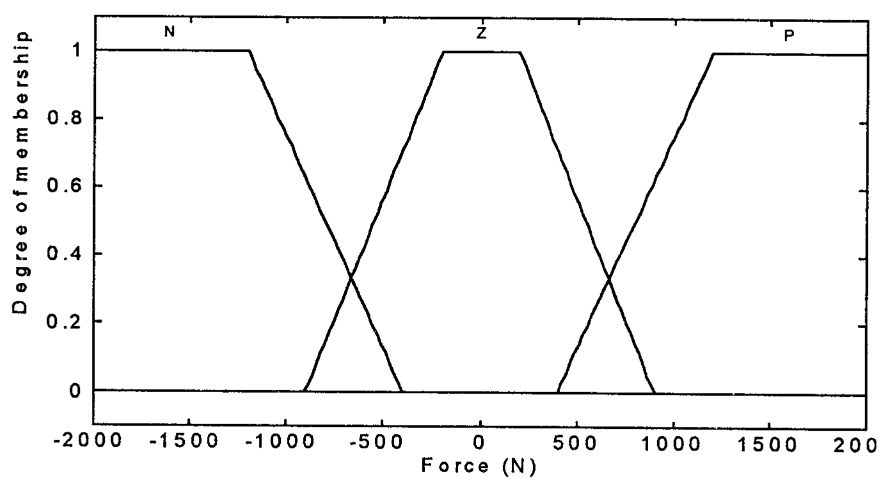

Figure 10. Output Force Membership Function

Using the linguistic variables, a set of fuzzy rules were devised based on a simplified version of the rules in [17]. Where [17] contained 72 rules, the new fuzzy rule base consisted of 9 rules. These are shown in the format of a lookup table in Figure 11.

		Acceleration		
		P	**Z**	**N**
Velocity	**P**	N	N	Z
	Z	N	Z	P
	N	Z	P	P

Figure 11. Rule Base

By using basic engineering sense, these rules were developed. For example, if the velocity is zero and the acceleration is positive then the mass's velocity is going to increase and a negative force should be applied. Similarly, if the velocity is negative and the acceleration is positive implies the velocity is slowing down and the control force would be zero.

The fuzzy reasoning inference procedure used was max-product composition rather than max-min composition. As previously mentioned max-product composition produces a smoother control action. The defuzzification procedure employed was the centroid of area method. This method was sufficient for the controller and is one of the most popular ones available. The overall fuzzy logic controller is shown in Figure 12.

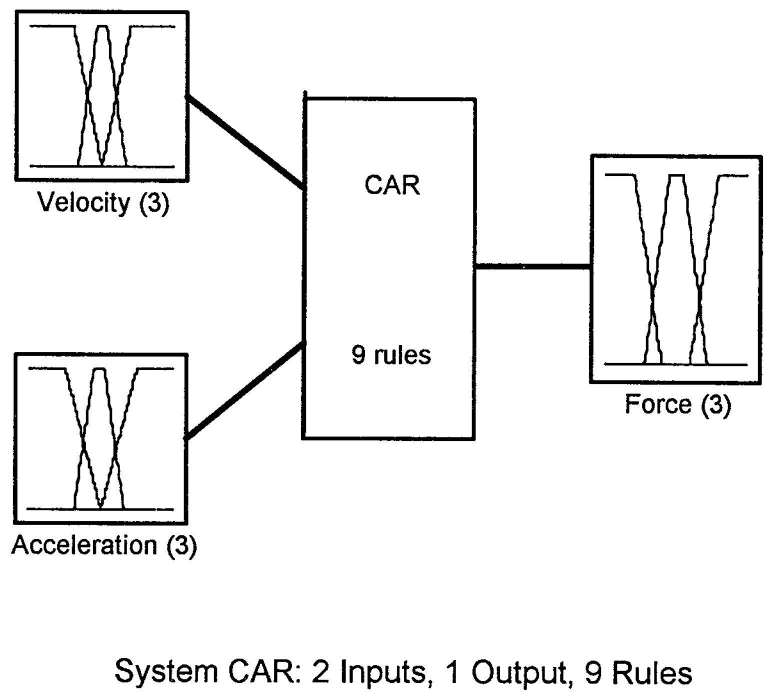

Figure 12. Fuzzy Logic Controller

5.2 The Comparison LQG Controller

Linear optimal control theory has been used by many authors [15, 16, 22, 24] to design controllers for active suspensions. The controller used as a comparison to the fuzzy logic controller was based on [15] and uses suspension deflection as a state input variable. In [24] different types of controllers based on full state feedback, absolute velocity feedback and suspension deflection were analyzed. It was concluded that the controller based on suspension deflection provides the best trade-off between ride quality, suspension packaging and road handling constraints.

By manipulating the equations for the suspension model, the following state space model was developed.

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} + \Gamma\mathbf{w} \tag{26}$$

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} + \mathbf{v} \tag{27}$$

with,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_u}{M_u} & -\frac{B}{M_u} & \frac{K_s}{M_u} & \frac{B}{M_u} \\ 0 & -1 & 0 & 1 \\ 0 & \frac{B}{M_s} & -\frac{K_s}{M_s} & -\frac{B}{M_s} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{M_s}{M_u} \\ 0 \\ -1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{28}$$

$$\mathbf{y} = x_3 \qquad \mathbf{C} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \mathbf{D} = 0, \qquad \mathbf{v} = v_1\,(\mathrm{t}) \tag{29}$$

The state-vector elements are $x_1 = x_u - x_r$, tire deflection, $x_2 = \dot{x}_u$, unsprung mass velocity, $x_3 = x_s - x_u$, suspension deflection and $x_4 = \dot{x}_s$, sprung mass velocity. A slight difference existed in the output variable *u*. In [15] *u* is defined as force divided by the M_s, so before the output could be used it was multiplied by M_s to get force. In the matrix equations, ***w*** and ***v*** are disturbance and measurement noise modeled as Gaussian, white processes, [15]:

$$E\left[w(t)w(\tau)\right] = 2\pi AV\delta(t-\tau) \quad \left(\mathrm{m/s^2}\right) \tag{30}$$

$$E\left[n(t)n(\tau)\right] = R_c\delta(t-\tau) \quad \left(\mathrm{m^2}\right) \tag{31}$$

Following [15], a Linear-Quadratic controller was designed for the nominal system based on a quadratic performance index that weights sprung-mass acceleration, suspension stroke and the control input with higher order sensor and actuator dynamics neglected. The quadratic performance index is given by:

$$J = \lim_{T\to\infty} \frac{1}{T} \int_0^T \left[q_1 \ddot{x}_s^{\,2} + q_2 (x_s - x_u)^2 + q_3 (x_u - x_r)^2 + r u(t)^2 \right] dt \tag{32}$$

In [23], various scalar weighting parameters (q_1, q_2, q_3, r) are used depending on desired performance objectives, such as "soft", small sprung mass acceleration and poor road handling characteristic, "typical" or "harsh", larger sprung mass accelerations and good road handling. The "typical" values were chosen for comparison and are listed along with vehicle, sensor and ground parameters and corresponding estimator and control gains in Table 2.

Table 2. Controller and Estimator Design Parameters and Gain

Vehicle, sensor and ground parameters,[15]	Scalar Controller Weights [23]
$V = 22$ m/s $A = 5\text{E-}6$ m $R_c = 10^{-8}$ m^2	$q_1 = 1.0$ $q_2 = 50{,}000$ $q_3 = 5{,}000$ $r = 0$
Estimator Gains $\mathbf{L} = [106.2\ -12405\ 153.6\ -614.3]^T$	Control Gains $\mathbf{C} = [7.54\ -0.873\ -10.71\ -7.97]$

CHAPTER 6

6. Simulation

This Chapter describes the computer simulations used for each suspension model. A model of the governing differential equations was created using SIMULINK. Each controller was represented by a particular SIMULINK block. The numerical formulation of the five road disturbance inputs are examined in detail.

6.1 Simulation Procedure

The quarter-car suspensions was simulated using Matlab's dynamic system simulation software, SIMULINK [18]. A SIMULINK model was constructed by using the differential equations derived by applying Newton's law to the quarter-car model. One general model was constructed and used with both the fuzzy logic and LQG controllers. The force was then set to zero and the model was used for the passive suspension. The fuzzy logic controller was designed using Matlab's Fuzzy Logic Toolbox [6]. Using a special fuzzy SIMULINK block, the fuzzy logic controller could be added to the SIMULINK model. The input into the fuzzy controller was sprung mass velocity and acceleration with the actuator force as output.

The LQG controller was simulated by using the 'reg' command from Matlab's Controls Toolbox [4]. Inputs for the 'reg' command are the **A**, **B**, **C**, **D** matrices and controller and estimator gains **K** and **L**. The output is a new set of matrices $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$, $\overline{\mathbf{C}}$ and $\overline{\mathbf{D}}$ which incorporate the estimator and controller gains together as:

$$\overline{\mathbf{A}} = \begin{bmatrix} 0 & 1.0 & -106.2 & 0 \\ -5628.8 & -26.1 & 12994.3 & 99.75 \\ 0 & -1.0 & -153.6 & 1.0 \\ 7.5 & 3.1 & 543.6 & -11.9 \end{bmatrix} \quad \overline{\mathbf{B}} = \begin{bmatrix} 106.2 \\ -12405.0 \\ 153.6 \\ 614.3 \end{bmatrix}. \tag{33}$$
$$\overline{\mathbf{C}} = \begin{bmatrix} 7.54 & -0.873 & -10.71 & -7.97 \end{bmatrix} \quad \overline{\mathbf{D}} = 0.0$$

These new matrices were then used as input into a SIMULINK state space block and represent the actual LQG controller. The LQG controller uses suspension deflection as input and force divided by sprung mass as output. The LQG controller output was then multiplied by a gain of M_s to obtain the actuator output force.

For both active suspensions, the maximum output force was limited to 1500 N by adding a limit block to the SIMULINK diagram. The three SIMULINK diagrams are shown in Figures 13-15.

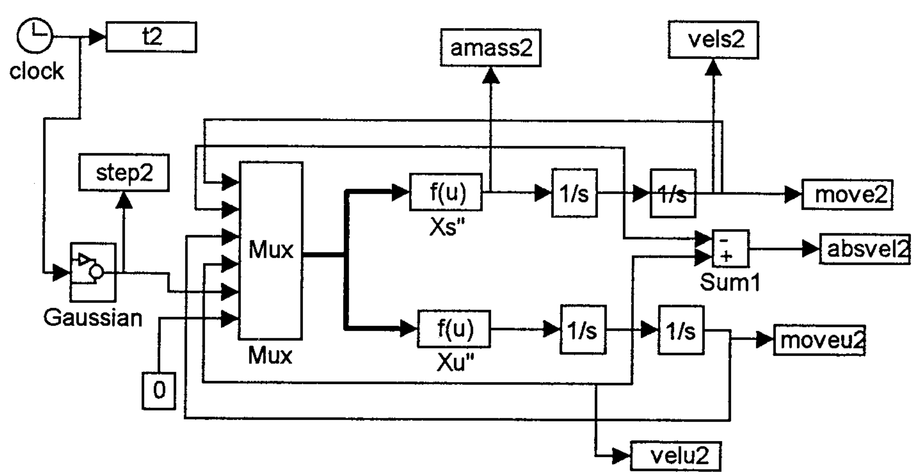

Figure 13. Passive Suspension Model

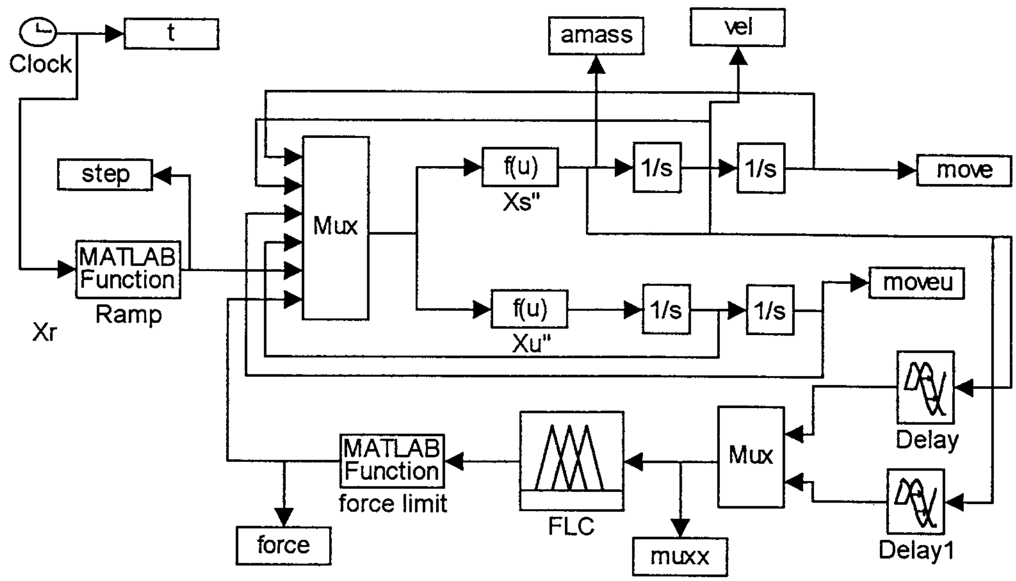

Figure 14. Fuzzy Logic Suspension Model

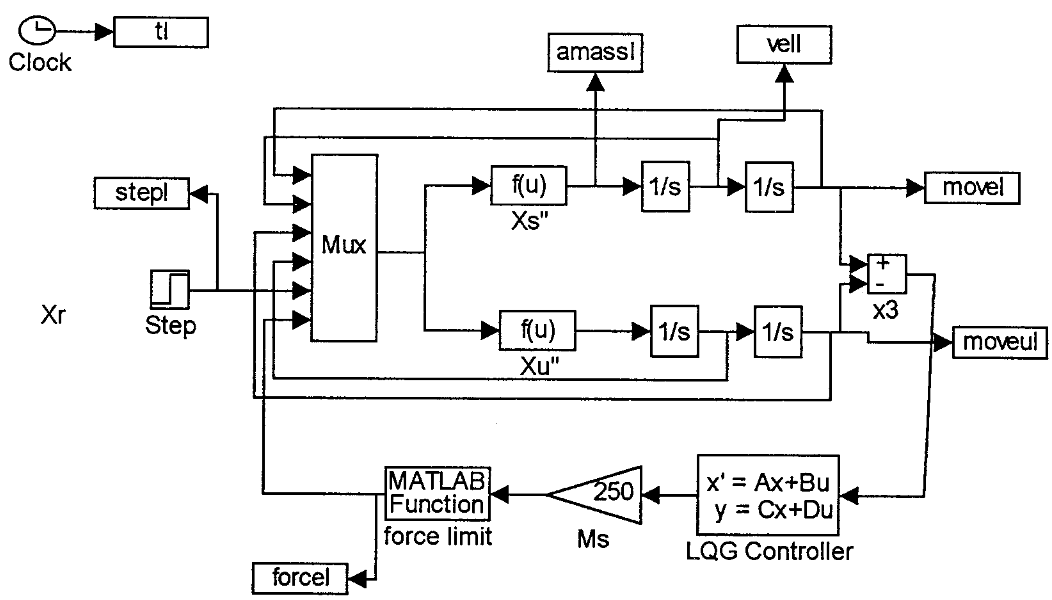

Figure 15. LQG Suspension Model

6.2 Road Disturbance Input

Once the SIMULINK models were developed, input disturbances were necessary to compare the three different suspensions. Five different input disturbance models were developed. The models are described in the proceeding sections followed by their graphical representations at the end of Chapter 6.

6.2.1 Step Input

First the three suspensions were subjected to a step input representing an isolated obstacle such as a curb. The step height was 10 cm applied instantaneously. Figure 17 shows the step input

6.2.2 Humped-Back Bridge

Many different road inputs can be developed to simulate specific road profiles. One such input is a hump-back bridge approximated using a half-sine wave:

$$r(t) = a \sin\left(\frac{\pi v}{d} t\right) \tag{34}$$

where a is the elevation of the bridge, d is the bridge width and v is the velocity of the vehicle. Figure 18 shows the humped-back bridge profile with a=0.25m, d=10.0m and v=30mph.

6.2.3 Ramp Input

A ramp input model was also devised by using the following formula:

$$r(t) = \frac{avt}{d} \tag{35}$$

where a is the height of the ramp, d is the ramp length and v is the velocity of the vehicle. Using a=0.6m, d=6.0m and v=30 mph, produces a 2° ramp as shown in Figure 19.

6.2.4 Random Linear Power Spectral Density Road Input

Most roads contain bumps, dips and small discontinuities. Therefore to evaluate the suspensions performance on a typical road an analytical model for the road input must be developed. It was determined to use the power spectral density (PSD) of a typical road. Small hills and dips represent low frequency, high amplitude and power elements, while surface roughness comprises high frequency, low amplitude and low power [21]. For ride and handling characteristics the most important frequency range is 0.5-50 Hz. Anything below 0.5 Hz is too small to cause any suspension deflection, while frequencies above 50 Hz are outside the bandwidth of tire and suspension dynamics [21]. In order to model the random road input, the quantitative PSD parameterization was used [21]. This model suggests a linear slope spectrum for single track road simulations in the form,

$$s(f) = kf^{-w}v^{w-1} \tag{36}$$

where v is the vehicle's velocity, w is the frequency taper parameter and k is the road surface roughness coefficient. A single track random road profile was generated from the given PSD function by summing many sine waves of a prescribed amplitude and frequency with a uniformly distributed random phase angle. As an example, a particular frequency is used in the following formulas to calculate a corresponding amplitude.

$$\bar{x}^2 = \int_{f_1}^{f_2} s(f)df \tag{37}$$

$$A = \sqrt{2}\sqrt{\bar{x}^2} \tag{38}$$

The integration limits were obtained by dividing the frequency range into 100 frequencies varying from 0.75 Hz to 50.25 Hz. A uniformly distributed random phase angle is then assigned to form one Sine wave. The MATLAB commands [13] used to calculate the amplitude and phase angle for a particular frequency are given in a m-file in Appendix A.

Using u=20mph, w=2.5 and $k=e^{-6}$, the graph of equation (36) for the linear PSD random input is shown in Figure 16.

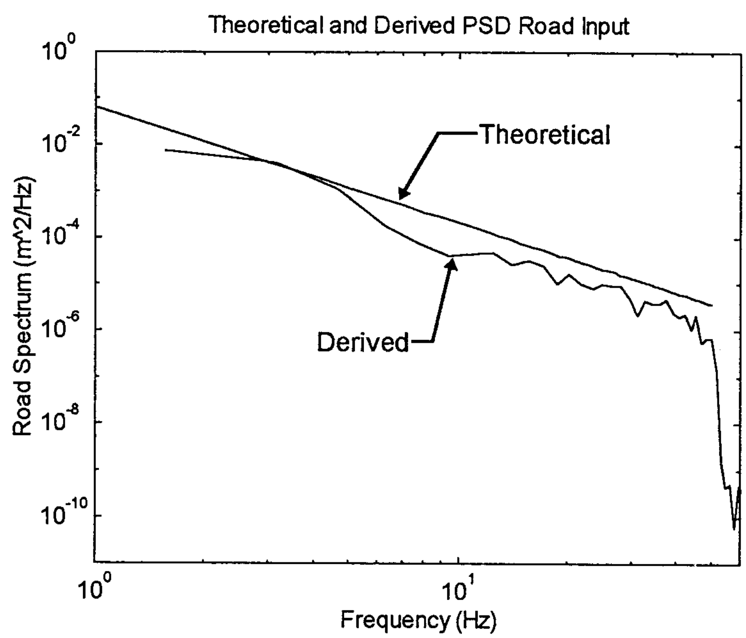

Figure 16. Theoretical and Derived PSD Random Input

Also shown on the graph is the PSD of the summation of sine waves. Although not an exact match, the linear model is a good fit to the derived random road profile. Naturally no random input would exactly fit the linear model and therefore as long as the same random road input is used on each suspension a good comparison of suspension performance can be developed. The graph of the derived input is shown in Figure 20.

6.2.5 White Noise Random Road Input

For a second random input, a standard random number generator with a Gaussian distribution is used to produce a random input. The output from the random generator is then put into a zero-order hold (ZOH) and multiplied by a specified gain. Using a ZOH of 0.1 seconds and a gain of 0.0158, the input shown in Figure 21 was produced. As the figure shows this produces a series of small short step inputs. Although this input was not based on any mathematical model or experimental data, all roads inputs are random thus this random input is acceptable to use for controller comparison. The gain was selected to keep the maximum step change to 3 cm. The following figures show the five inputs.

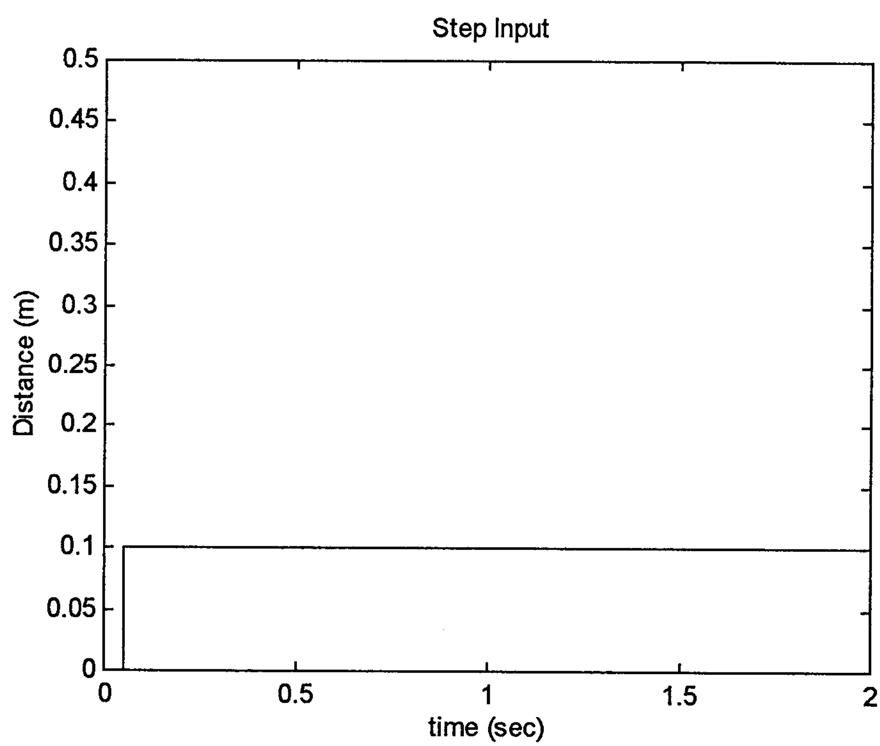

Figure 17. Step Input

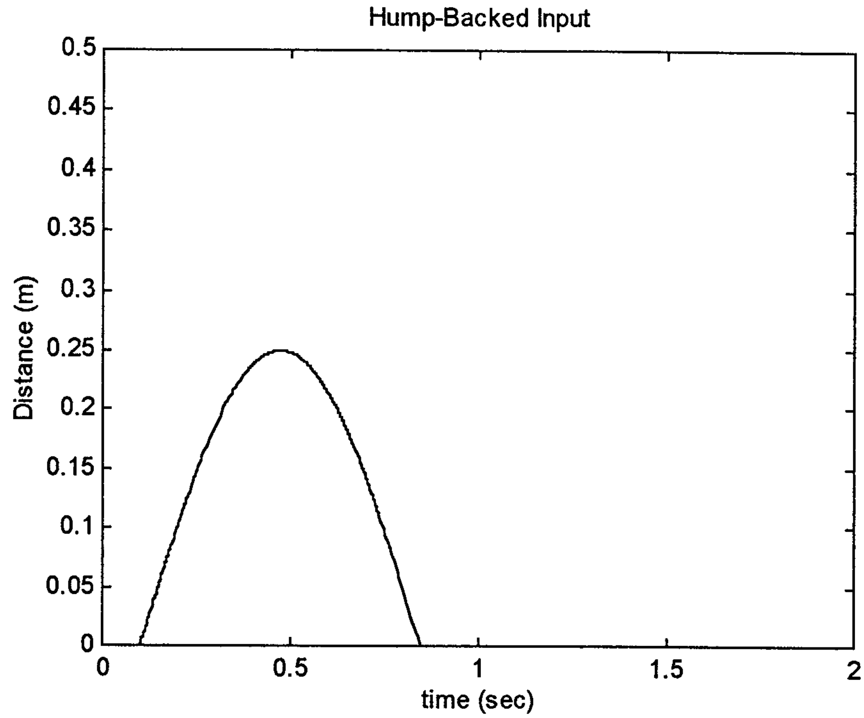

Figure 18. Hump-Backed Input

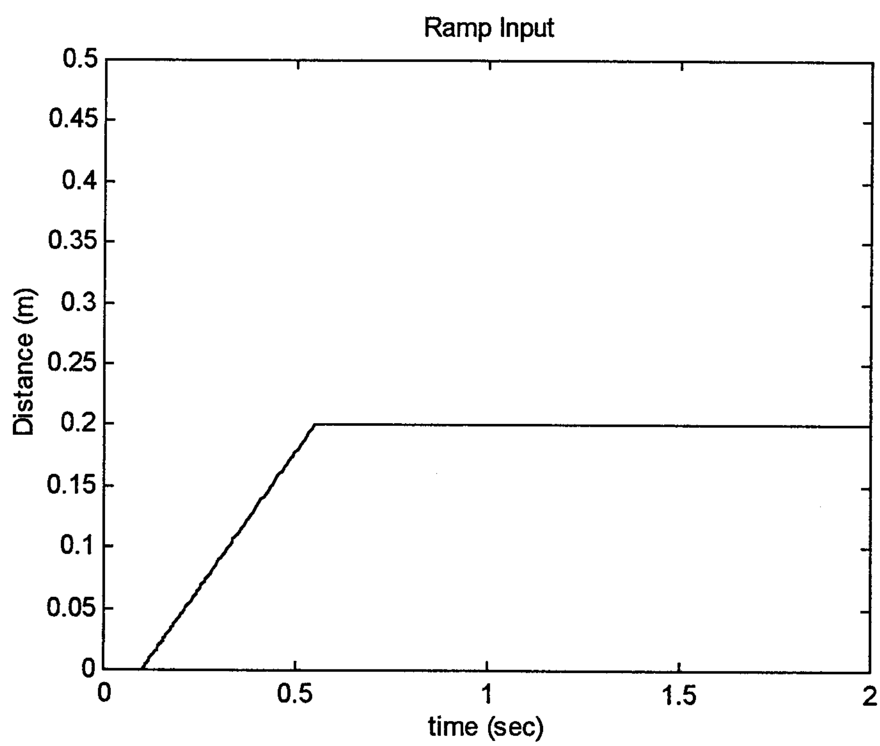

Figure 19. Ramp Input

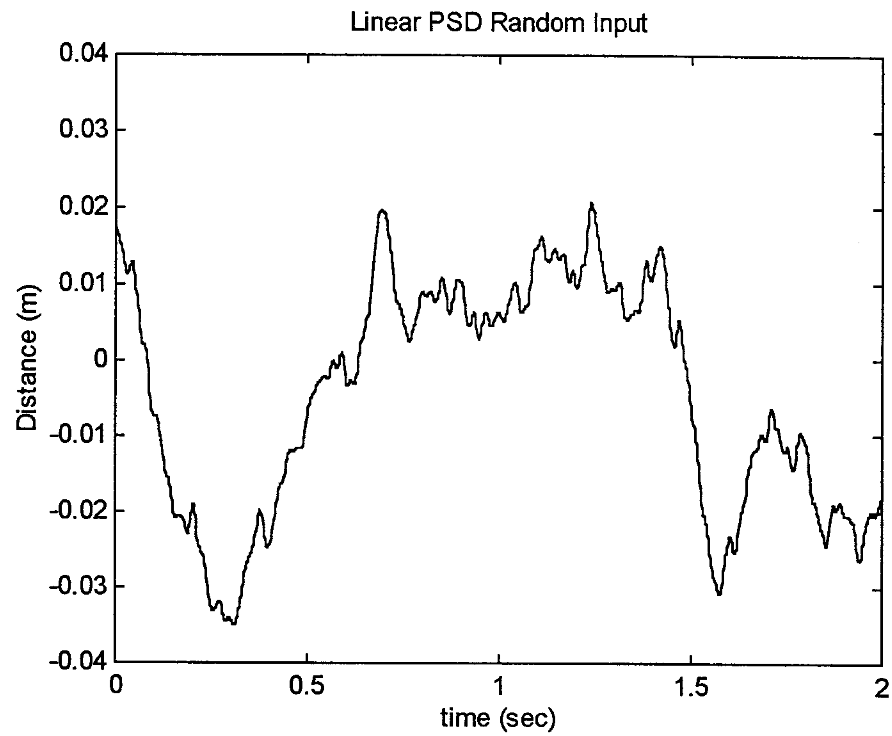

Figure 20. Linear PSD Random Input

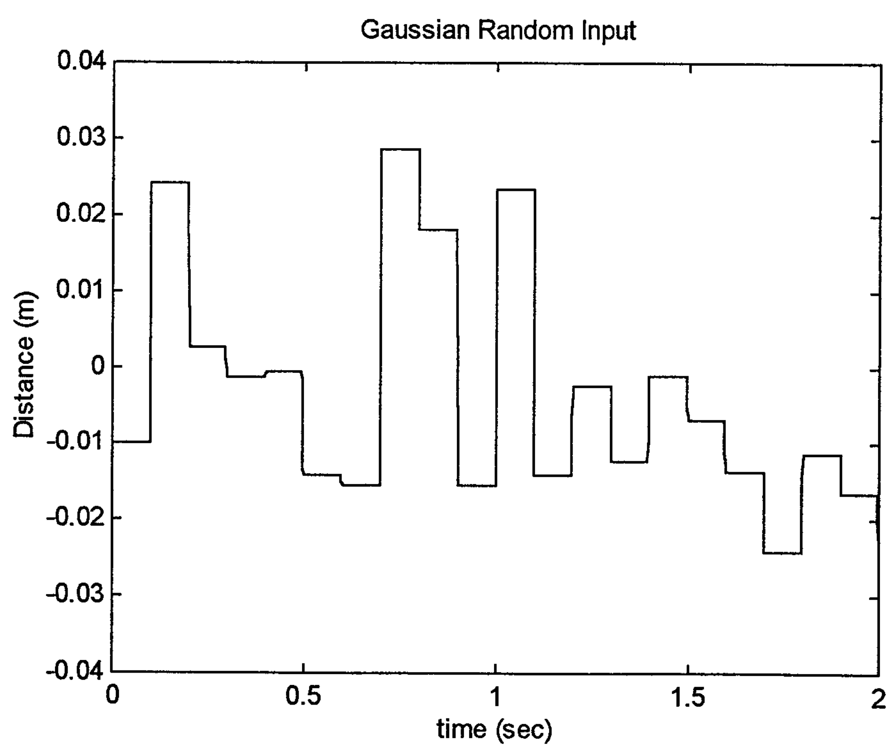

Figure 21. Gaussian Random Input

CHAPTER 7

7. Results

The response to the five inputs were calculated on the three different suspensions: passive, LQG and fuzzy logic . For all simulations Matlab's Runge-Kutta 5 algorithm was used to solve the differential equation using the following parameters: final time=2 seconds, minimum step size=$1e^{-6}$ seconds, maximum step size=0.05 seconds and tolerance=$1e^{-5}$. To compare the different suspension responses, graphs of the displacement, velocity and acceleration responses are shown at the end of Chapter 7.

7.1 Step Input Results

The three controllers were subjected to a step input of height 10 cm at time 0.05 seconds. Figures 22-24 show the displacement, velocity and acceleration of the sprung mass. The fuzzy controller and LQG controller both outperformed the passive case as expected. The fuzzy controller and LQG optimal controller performed similarly when subjected to the step input. The LQG controller had a slightly quicker reaction time, however, examining the velocity response, the fuzzy controller has a smoother transition to zero velocity.

7.2 Hump-Backed Bridge Input

Next, the three suspensions were tested using the hump-backed bridge as input. The simulation was run for 2 seconds with the input starting at 0.1 seconds into the simulation. The bridge parameters were d=10m, a=0.25m and v= 30 mph. Figures 25-27 show the displacement, velocity and acceleration responses. Once again the fuzzy and LQG controller both outperformed the passive case. The fuzzy controller had a lower initial displacement and velocity response than the LQG controller.

7.3 Ramp Input

Next, the suspensions were subjected to a ramp input as described in section 6.2.3. The ramp parameters were set at the following: a=0.2, d=6.0 and v=30 mph. Figures 28-30 show the results. The LQG controller had a quicker reaction time, however this translates into increased velocity and acceleration responses. The fuzzy controller has considerably less velocity and acceleration responses when compared to the LQG controller.

7.4 Random Linear PSD Input

Next, the random linear PSD input was used on the three controllers. The input signal was generated using the procedure outlined in section 6.2.4 with the following parameters: v=20mph, w=2.5 and k=e^{-6}. The displacement, velocity and acceleration responses are shown in the following Figures 31-33. Due to the randomness of the output, it is difficult to draw conclusions using the respective responses; thus, RMS values were calculated.

To measure the performance of the suspensions, the RMS values of sprung mass acceleration, suspension deflection and tire displacement were calculated. Appendix A contains the M-file used to calculate the RMS values. Table 3 shows the results.

Table 3. RMS Values for Random Linear PSD Input

RMS VALUES	Passive	LQG	% decrease	Fuzzy	% decrease
Sprung Mass Acceleration	1.4891	1.2068	18.9	1.0898	26.8
Suspension Deflection	0.0151	0.0117	22.5	0.0132	12.6
Wheel Hop	0.0043	0.0043	0.0	0.0047	-9.3

This shows that the fuzzy suspension outperforms both the passive and LQG suspensions in the category of ride improvement. The LQG suspension reduces suspension deflection more than the fuzzy suspension; however, this measurement is just related to design constraints and does not affect ride or handling characteristics. Neither active suspension improves the wheel hop with the fuzzy system slightly increasing the wheel hop.

7.5 Random Gaussian Input

Finally, the random Gaussian input that was described in section 6.2.5 was used on all three controllers. A ZOH of 0.1s and gain, k=0.0158, were used with the random number generator with a Gaussian distribution. Figures 34-36 show the displacement, velocity and acceleration from using this input. As before the graphs are inconclusive and the RMS values were computed as shown in Table 4.

Table 4. RMS Values for Random Gaussian Input

RMS VALUES	Passive	LQG	% decrease	Fuzzy	% decrease
Sprung Mass Acceleration	2.3975	2.2196	7.4	1.8464	23.0
Suspension Deflection	0.0177	0.0150	15.2	0.0169	4.5
Wheel Hop	0.0101	0.0105	-4.0	0.0103	-2.0

These results are similar to the results from the random linear PSD input. The fuzzy suspension does an excellent job in reducing the sprung mass acceleration and thus increasing the ride comfort. The LQG only has a 7.4% increase in ride comfort compared to the previous 17% increase. The LQG improves the suspension deflection with only a slight improvement from the fuzzy suspension. Both controllers slightly worsened handling characteristics.

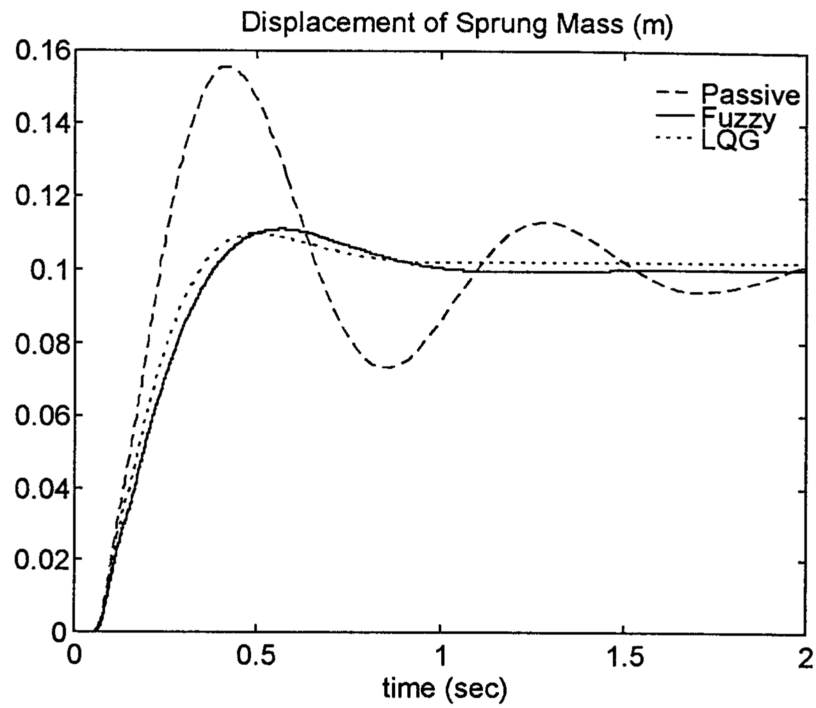

Figure 22. Step Input Displacement Response

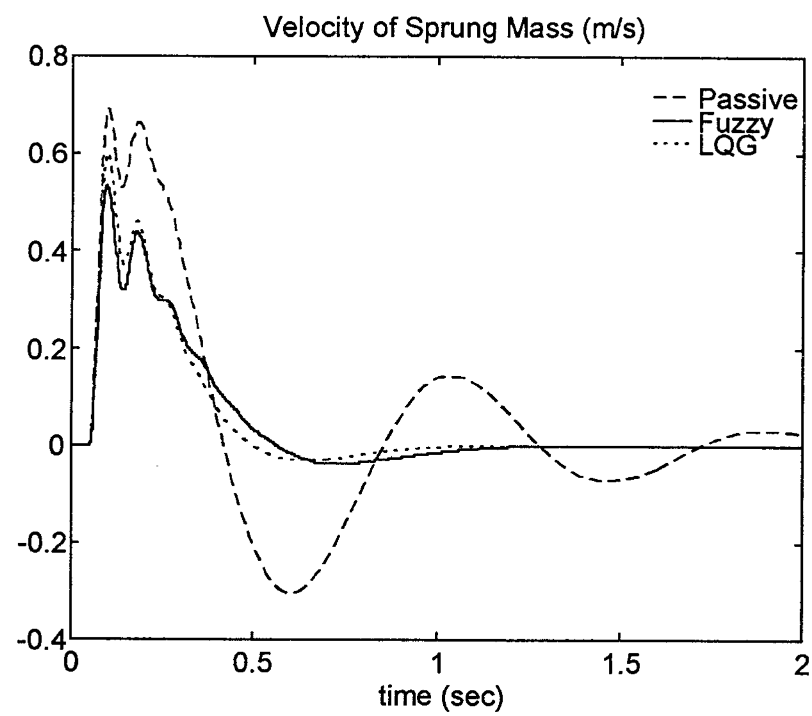

Figure 23. Step Input Velocity Response

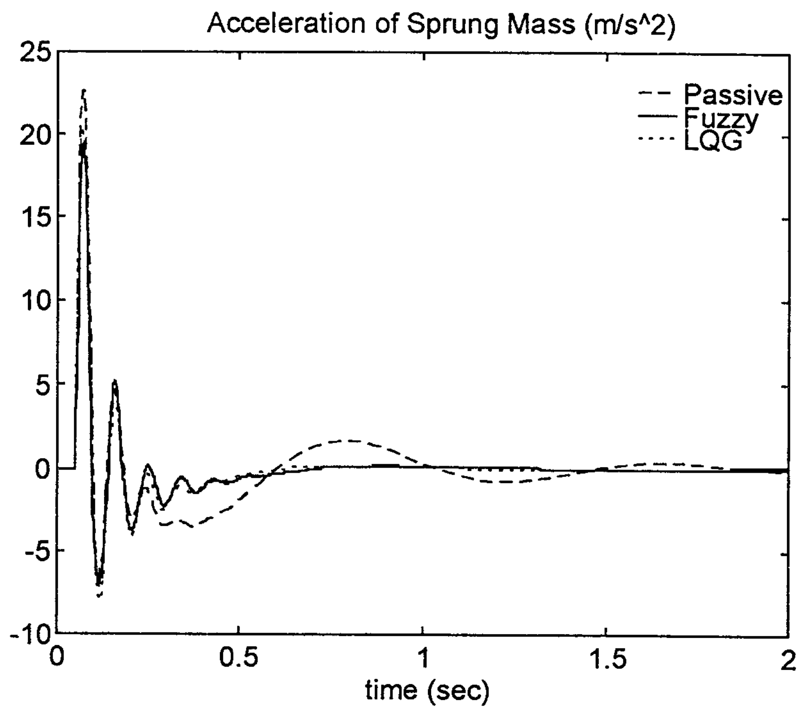

Figure 24. Step Input Acceleration Response

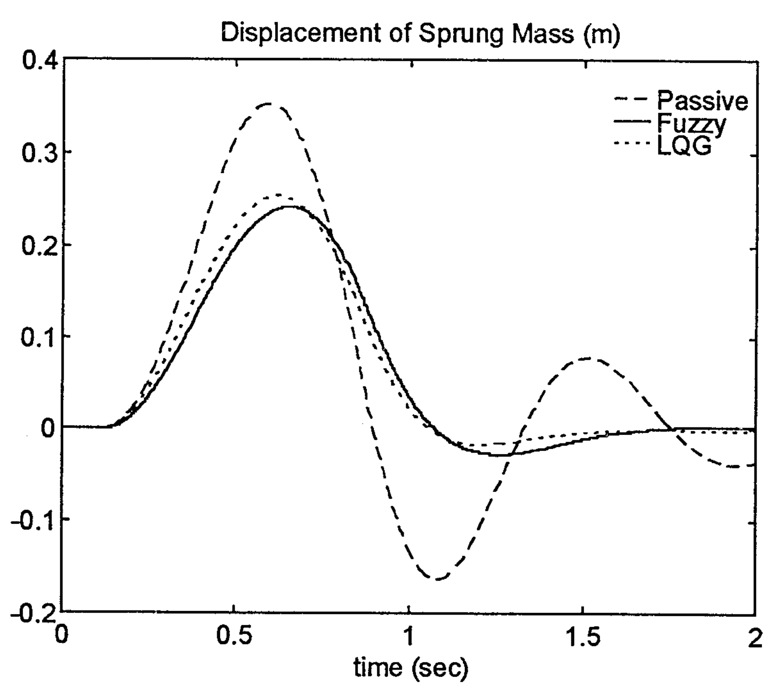

Figure 25. Hump-Backed Bridge Displacement Response

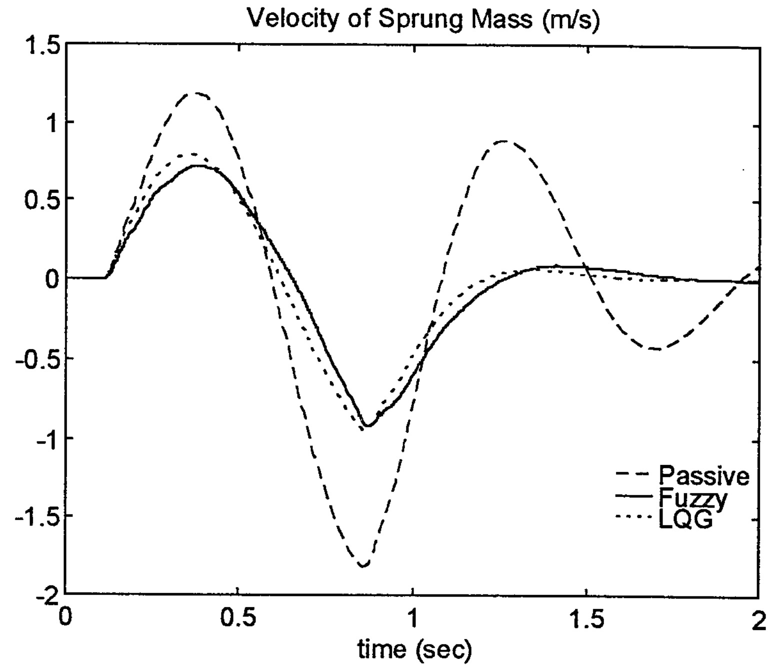

Figure 26. Hump-Backed Bridge Velocity Response

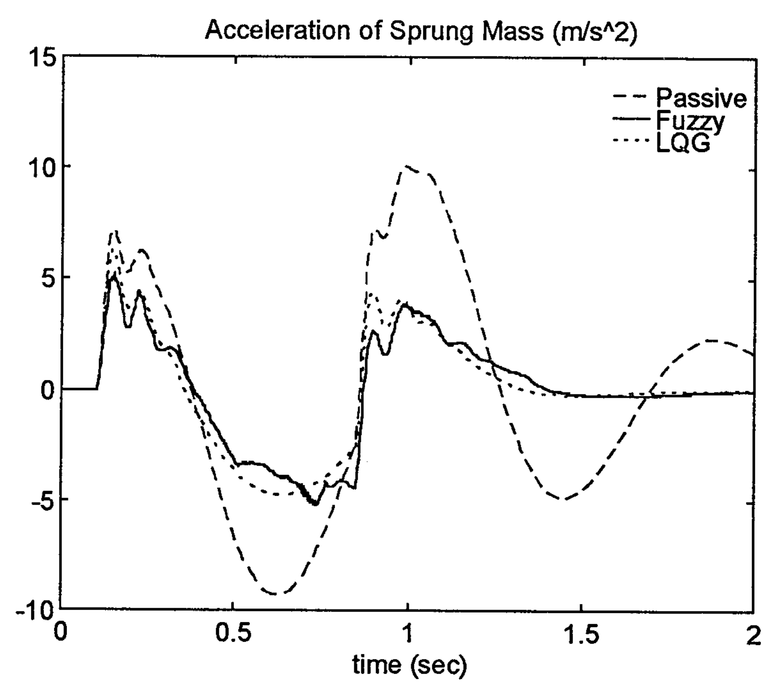

Figure 27. Hump-Backed Bridge Acceleration Response

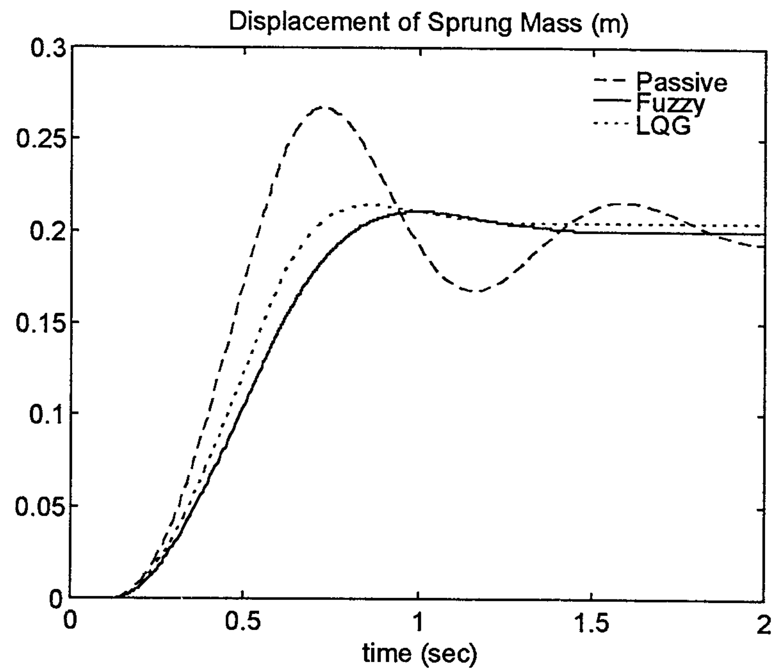

Figure 28. Ramp Input Displacement Response

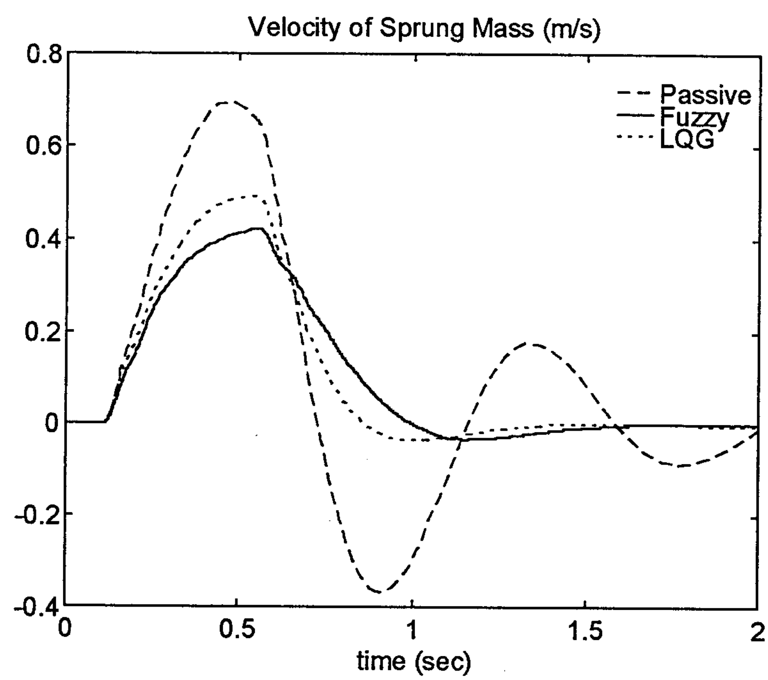

Figure 29. Ramp Input Velocity Response

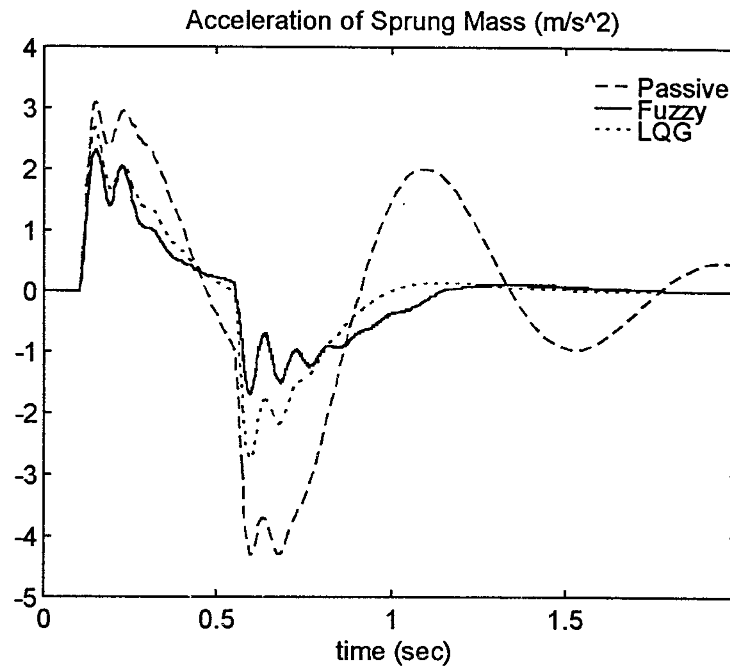

Figure 30. Ramp Input Acceleration Response

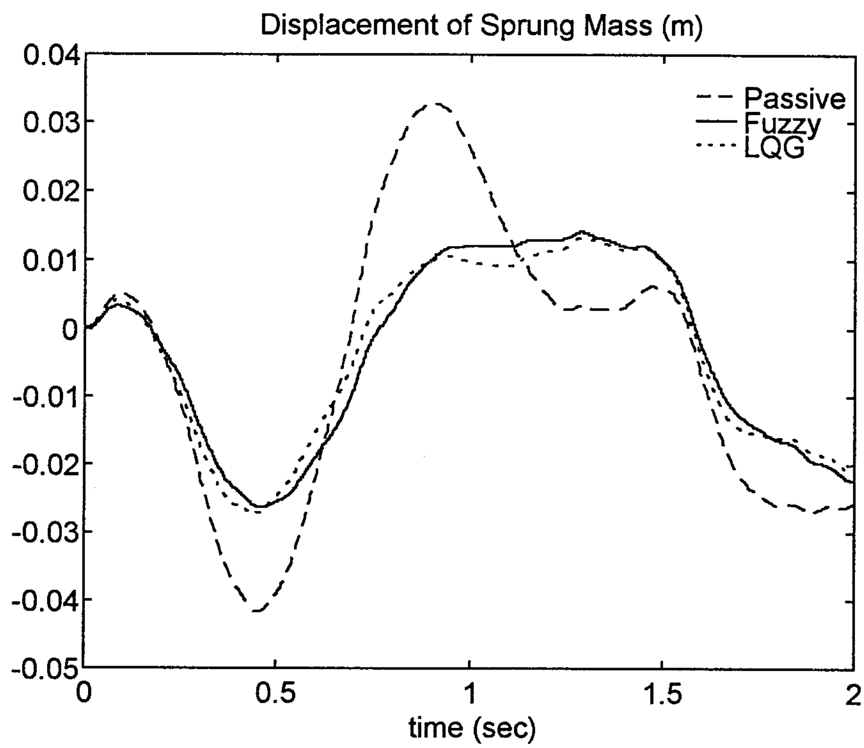

Figure 31. Random Linear PSD Displacement Response

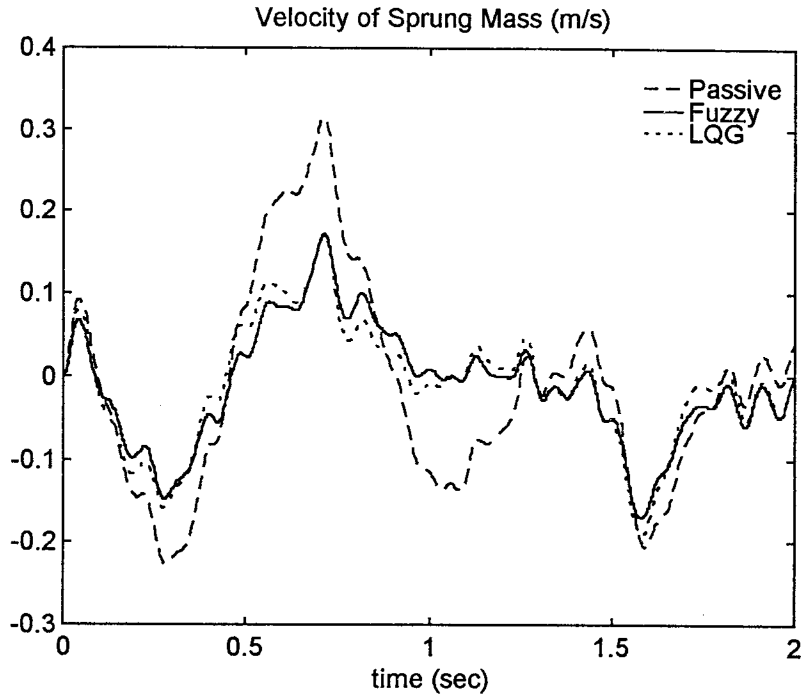

Figure 32. Random Linear PSD Velocity Response

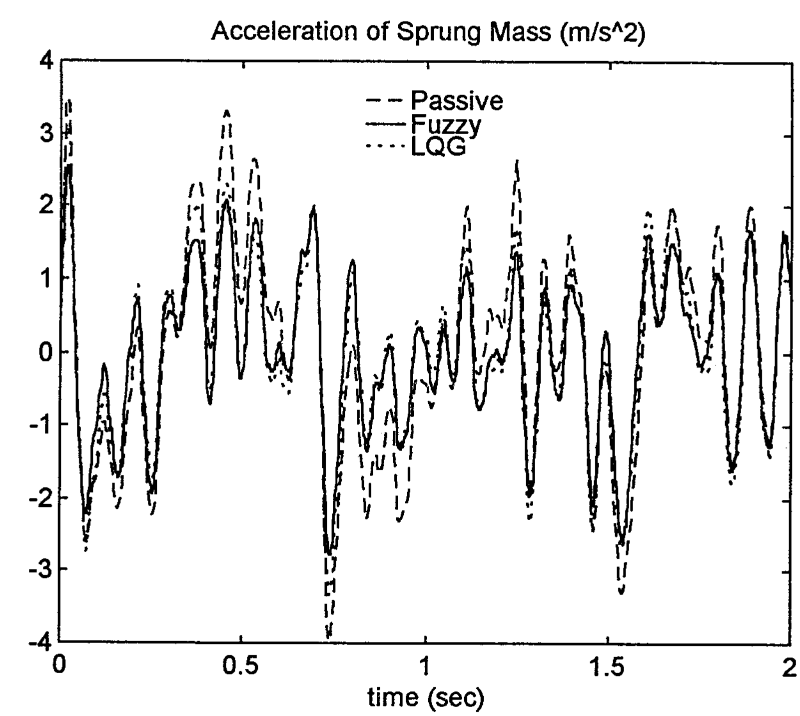

Figure 33. Random Linear PSD Acceleration Response

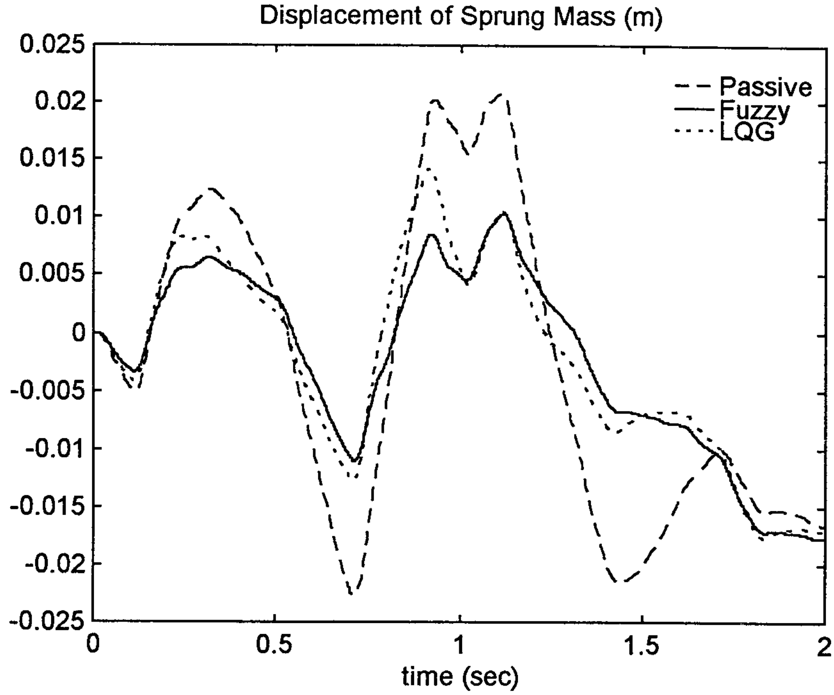

Figure 34. Gaussian Random Displacement Response

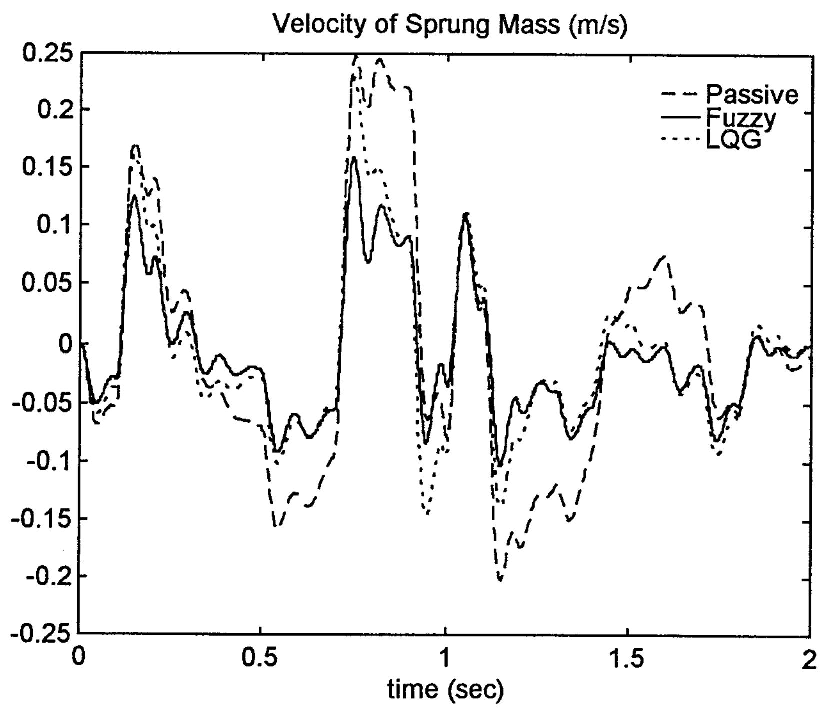

Figure 35. Gaussian Random Velocity Response

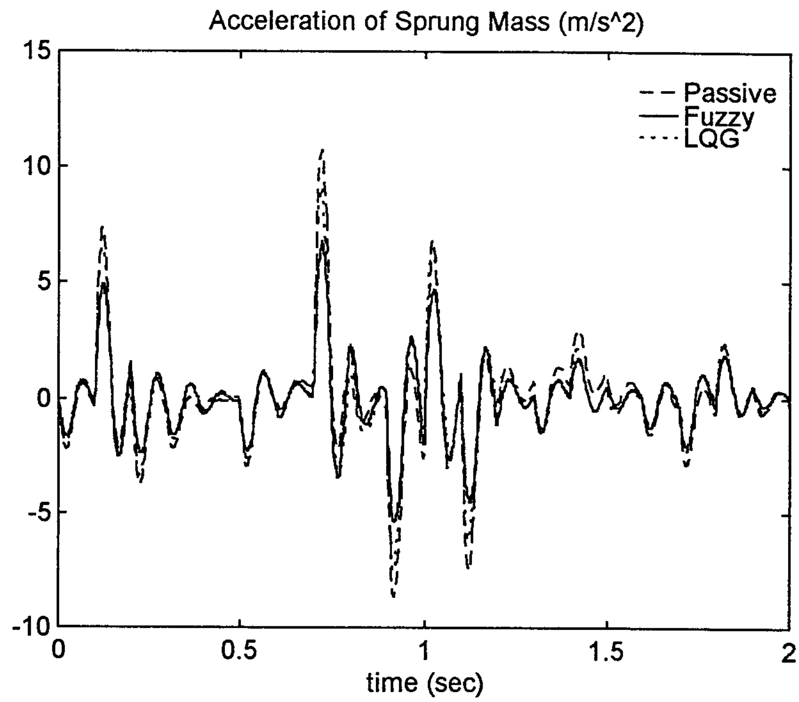

Figure 36. Gaussian Random Acceleration Response

CHAPTER 8

8. Conclusions & Recommendations

8.1 Conclusions

The fuzzy logic controller successfully controlled the active suspension. The fuzzy logic suspension performed as well as and usually better than the LQG suspension. This was proven to be true over a wide variety of input conditions.

The most notable improvements were in the random input cases, which is very important since all actual road inputs are random and do not consist of one step or ramp. The LQG suspension displayed the best decrease in suspension deflection with the fuzzy logic suspension showing a modest improvement. A very simplified rule base proved adequate for the fuzzy logic controller.

Overall, the fuzzy logic controller outperformed the benchmark LQG controller. Most notable was the superiority of the fuzzy logic controller when subjected to the random input. This once again proves the diversity of fuzzy logic in controlling various processes.

8.2 Recommendations for Future Work

This successful application of fuzzy logic to an active suspension lays the groundwork for additional research. Additional areas of future development are listed below:

- Explore partitioning the inputs and output universes of discourse into additional fuzzy subsets.
- Expand the suspension model to include additional DOF and study pitch and roll response.
- Add non-linearities to the model .
- Implement a variable damper as the active element.
- Investigate optimization techniques such as Adaptive-Network-based Fuzzy Inference System, ANFIS.
- Explore the use of fuzzy logic as a gain scheduler for LQG control.
- Compare the stability robustness of LQG control and fuzzy logic control.

Appendix A MATLAB M-Files

The following m-files were used for calculations during the research. Comments statements were added to the files and begin with the % symbol.

ampl.m

```
% Calculates the frequencies and places them in the matrix amp
d=1;
count=1;
for k=0.5:d:50
        f1=k;
        f2=k+d;
% Find the average frequency
        f(count)=(f1+f2)/2;
        xbar2=-1.336e-4/1.5*(f2^(-1.5)-f1^(-1.5));
        amp(count)=sqrt(2)*sqrt(xbar2);
        count=count+1;
end
% Calculates a random equally distibuted phase angle corresponding to each frequency
angle=2*pi*rand(length(f),1);
```

rms.m

```
% Calculates the RMS values for sprung mass acceleration
prmsa=sqrt(cov(amass2));
frmsa=sqrt(cov(amass));
lrmsa=sqrt(cov(amassl));
disp('RMS Acceleration Sprung Mass');
disp('  Passive   Fuzzy   LQG')
disp([prmsa,frmsa,lrmsa]);
rat2=move2-moveu2;
rat=move-moveu;
ratl=movel-moveul;

% Calculates the RMS values for suspension deflection
```

```
prmsrat=sqrt(cov(rat2));
frmsrat=sqrt(cov(rat));
lrmsrat=sqrt(cov(ratl));
disp('RMS Rattlespace');
disp('   Passive   Fuzzy   LQG');
disp([prmsrat,frmsrat,lrmsrat]);

% Calculates the RMS values for wheel hop
hop2=moveu2-step2;
hop=moveu-step;
hopl=moveul-stepl;
prmshop=sqrt(cov(hop2));
frmshop=sqrt(cov(hop));
lrmshop=sqrt(cov(hopl));
disp('RMS Wheel Hop');
disp('   Passive   Fuzzy   LQG');
disp([prmshop,frmshop,lrmshop]);
```

REFERENCES

[1] Abdelnour, G. M., et al, Sept./Oct. 1991, "Design of a Fuzzy Controller Using Input and Output Mapping Factors," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21; No. 5; pp. 952-960.

[2] Burke, D. M., 1992, "A Layered Fuzzy Motion Controller For a Mobile Robot," MS Thesis, Wright State University.

[3] Cherry, A. S. and R. P. Jones, March 1995, "Fuzzy logic control of an automotive suspension system," *IEE Proceedings. Control Theory and Applications*, Vol. 142; No. 2; pp. 149-160.

[4] Control System Toolbox User's Guide. The MathWorks Inc., 1993.

[5] Gordon, T. J., et al, March 1994, "A Learning Automation Methodology for Control System Design in Active Vehicle Suspensions," *IEE Conference Publication, Proceedings of the International Conference on CONTROL '94*, Vol 1; No. 389; pp. 326-331.

[6] Gulley, N. and Jang, J.-S., Fuzzy Logic Toolbox User's Guide. The MathWorks, Inc., 1995.

[7] Healey, A. J., et al, Dec. 1977, "An Analytical and Experimental Study of Automobile Dynamics with Random Roadway Inputs, " *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 99; No. 4; pp. 284-292.

[8] Hrovat D., June 1993, "Applications of Optimal Control to Advanced Automotive Suspension Design," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 115; pp. 328-342.

[9] Jang, J.-S, May/June 1993, "ANFIS: Adaptive-Network-Based Fuzzy Inference System, " *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23; No. 3; pp. 665-684.

[10] Jang, J.-S and C.-T Sun, March 1995, "Neuro-Fuzzy Modeling and Control," *Proceedings of the IEEE*, pp. 378-406.

[11] Kosko, B., Neural Networks and Fuzzy Systems. Englewood Cliffs: Prentice Hall, Inc., 1992.

[12] Lee, C. C., March/April 1990, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller - Part I," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20; No. 2; pp. 404-418.

[13] MATLAB User's Guide. The MathWorks Inc., 1993.

[14] Mendel, J. M., March 1995, "Fuzzy Logic Systems for Engineering: A Tutorial." *Proceedings of the IEEE*, Vol. 83; No. 3; pp. 345-377.

[15] Ray, L. R., Dec. 1992, "Robust Linear-Optimal Control Laws for Active Suspension Systems," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 114; pp. 592-598.

[16] Redfield, R. C. and D. C. Karnopp, March 1989, "Performance Sensitivity of an Actively Damped Vehicle Suspension to Feedback Variation," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 111; pp. 51-60.

[17] Ro, P. I., et al, 1993, "An Active Suspension using Fuzzy Logic Control," 1993 *American Control Conference Proceedings*, Vol. 3; pp. 2252-2253.

[18] SIMULINK User's Guide. The MathWorks Inc., 1994.

[19] Schwartz, D. G., et al, April 1994, "Applications of Fuzzy Sets and Approximate Reasoning," *Proceedings of IEEE*, Vol. 82; No. 4; pp. 482-497.

[20] Sugeno, M., 1985, "An Introductory Survey of Fuzzy Control, " *Information Sciences*, Vol. 36; pp. 59-83.

[21] Truscott, A. J. and A. W. Burton, 1994, "On the Analysis, Modeling and Control of an Advanced Automotive Suspension System," *Proceedings of the International Conference on Control '94, IEE Conference Publication*, Vol. 1; pp. 183-189.

[22] Ulsoy, A. G., et al, March 1994, "Stability Robustness of LQ and LQG Active Suspensions, " *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 116; pp.123-131.

[23] Ulsoy, A. G. and D. Horvat, 1990, "Stability Robustness of LQG Active Suspensions," *1990 American Control Conference Proceedings*, Vol. 2; pp. 1347-1356.

[24] Yue, C., et al, June 1989, "Alternative Control Laws for Automotive Active Suspensions," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 111; pp. 286-291.

[25] Zadeh, L. A., January 1973, "Outline of a new approach to the analysis of complex systems and decision processed," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 3; No. 1; pp. 28-44.

[26] Zimmermann, H. J., Fuzzy Set Theory and its Applications. Boston: Kluwer Academic Publishers, 1991.