Neurofuzzy Controller Based Full Vehicle Nonlinear Active Suspension Systems

A Thesis Submitted for the Degree of

Doctor of Philosophy

by

Ammar Abdulshaheed Aldair

School of Engineering and Informatics

University of Sussex

Brighton

UK

March 2012

Declaration

I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other university for a degree.

Signed……..

Ammar Abdulshaheed Aldair

To

My Mother,

My Wife,

My Children,

and

Soul of My Father.

THE UNIVERSITY OF SUSSEX

NEUROFUZZY CONTROLLER BASED FULL VEHICLE NONLINEAR ACTIVE SUSPENSION SYSTEMS

Submitted for the degree of Doctor of Philosophy

Ammar Abdulshaheed Aldair

ABSTRACT

To design a robust controller for active suspension systems is very important for guaranteeing the riding comfort for passengers and road handling quality for a vehicle.

In this thesis, the mathematical model of full vehicle nonlinear active suspension systems with hydraulic actuators is derived to take into account all the motions of the vehicle and the nonlinearity behaviours of the active suspension system and hydraulic actuators.

Four robust control types are designed and the comparisons among the robustness of those controllers against different disturbance types are investigated to select the best controller among them. The MATLAB SIMULINK toolboxes are used to simulate the proposed controllers with the controlled model and to display the responses of the controlled model under different types of disturbance. The results show that the neurofuzzy controller is more effective and robust than the other controller types.

The implementation of the neurofuzzy controller using FPGA boards has been investigated in this work. The Xilinx ISE program is employed to synthesis the VHDL codes that describe the operation of the neurofuzzy controller and to generate the configuration file used to program the FPGA. The ModelSim program is used to simulate the operation of the VHDL codes and to obtain the expected output data of the FPGA boards. To confirm that FPGA the board used as the neurofuzzy controller system operated as expected, a MATLAB script file is used to compare the set of data obtained from the ModelSim program and the set of data obtained from the MATLAB SIMULINK model. The results show that the FPGA board is effective to be used as a neurofuzzy controller for full vehicle nonlinear active suspension systems.

The active suspension system has a great performance for vibration isolation. However the main drawback of the active suspension is that it is high energy consumptive. Therefore, to use this suspension system in the proposed model, this drawback should be solved. Electromagnetic actuators are used to convert the vibration energy that arises from the rough road to useful electrical energy to reduce the energy consumption by the active suspension systems. The results show that the electromagnetic devices act as a power generator, i.e. the vibration energy excited by the rough road surface has been converted to a useful electrical energy supply for the actuators. Furthermore, when the nonlinear damper models are replaced by the electromagnetic actuators, riding comfort and the road handling quality are improved. As a result, two targets have been achieved by using hydraulic actuators with electromagnetic suspension systems: increasing fuel economy and improving the vehicle performance.

Acknowledgement

I would like to express my sincere gratitude to my supervisor Dr. W.J. Wang for his firm support, encouragement and assistant for me throughout the duration of my study in the University of Sussex. His invaluable guidance provided many ideas, which have lead to the completion of this thesis.

I am extremely grateful to Dr. Ahmet Aydin, who helped me to understand the structure of the FBGA board and guide me to program the FPGA IC.

My many thanks go to all control group members. I also would like to thank the technical support staff in the School of Engineering and Design and IT services for their help.

My special thanks to the Iraqi Government for their financial support in order for me to study aboard. Many thanks go to my home university- University of Basrah and my colleagues of the Electrical Engineering Department, especially Dr. Areef, for their support and encouragement.

I would like to express endless gratitude to my parents and my wife" Israa" who gave me every thing they could to enable me to reach the highest possible education level. I only hope that they know how their love, support and patience encouraged me to fulfil their dream.

Finally, all of my thanks go to my children "Yasir, Haider and Sora", my brothers, my sisters and my uncles for their assistance and encouragement.

Table of Contents

Table of Figures

List of Tables

List of Symbols

Symbol	Description	Unit
A_p	Cross section area of piston	m^2
$\{a_{ik}, b_{ik}, c_{ik}\}$	Premise parameters	-
b	Width of track	m
b_m	Centers of the membership functions of the output linguistic	-
$b_1(k)$	Bias of the k^{th} node of the hidden layer	-
$b_2(l)$	Bias of the l^{th} node of the output layer	-
Cm	Damping force constant	N.s/m
C_1, C_2, C_3 and C_4	Front-left, Front-right, Rear-right and rear-left suspension damping factors, respectively	N.s/m
c_1, c_2, c_3 and c_4	Front-left, Front-right, rear-right and rear-left tire damping factors, respectively	N.s/m
F	Actual nonlinear force of hydraulic actuator	N
F_a	Vertical force electromagnetic actuator	N
F_C	Nonlinear force of damping model	N
F_f	Frictional force inside the actuator	N
F_h	Applied nonlinear force of hydraulic actuator	N
F_K	Nonlinear force of spring model	N
F_m	Damping force of the electromagnetic actuator	N
J_f	Objective function	m^2
GE, GED, GEI and GU	Scaling gain of the neural controller system	-

Symbol	Description	Unit
I	Electric current flow through the motor's coils	A
J	Jacobian matrix	-
J_x	Moment of inertia, x-direction	$kg.m^2$
J_y	Moment of inertia, y-direction	$kg.m^2$
K_d	Derivative constant	-
K_i	Integral constant	-
K_p	Proportional constant	-
K_1, K_2, K_3 *and* K_4	Front-left, Front-right, Rear-right and rear-left suspension stuffiness constant, respectively	N/m
k_1, k_2, k_3 *and* k_4	Front-left, Front-right, rear-right and rear-left tire stuffiness factors, respectively	N/m
l_1	Distance between the center of gravity of sprung mass and front axle	m
l_2	Distance between the center of gravity of sprung mass and rear axle	m
N_{ind}	Number of individuals in the population	-
M	Sprung mass	kg
m_1, m_2, m_3 *and* m_4	Front-left, Front-right tire, Rear-right and rear-left tire mass, respectively	kg
P_e	Power of the motor	W
P_H	Lead of the ball-screw	m
P_L	Load pressure	Pa
Pos	Position of an individual in this population	-
P_s	Pump supply pressure	Pa

Symbol	Description	Unit
$\{p_{0k}, p_{1k}, p_{2k}, \dots, p_{nk}\}$	Consequent parameters	-
r	Effective radius for force conversion	m
SP	Selective Pressure	-
T_e	Torque of the DC motor	N.m
Tx	Cornering torque	N.m
Ty	Braking torque	N.m
u_m	Voltage signal input to the servovalve	V
$v(j,l)$	Weight between j^{th} node and l^{th} node of output layer	-
$w(k,i)$	Weight between i^{th} input and k^{th} node of hidden layer	-
w_1, w_2, w_3 and w_4	Vertical displacements at Front-left, Front-right, Rear-right and rear-left unsprung mass respectively	m
x_v	spool displacement of the servovalve	m
z_c	Vertical displacements at centre of gravity	m
z_1, z_2, z_3 and z_4	Vertical displacements at suspention 1, suspention 2, suspention 3 and suspention 4, respectively	m
Γ1	Activation function of the output layer	-
Γ_2	Activation function of the output layer	-
$\boldsymbol{\Phi}$	Flux linkage	V.s
ϕ	Cost function	m^2
α	Pitch angle	rad
α, β and γ	Actuator parameters	-

Symbol	Description	Unit
γ	Learning rate	-
ζ	Empirical parameter	-
η	Roll angle	rad
λ	Integral order	-
μ	Derivative order	-
τ	Time constant	s
φ	Thread lift angle	degree
ϱ	Adaptive gain	-

List of Abbreviations

Abbreviation	Description
FOPID	Frictional Order Proportional, Integral and Derivative
EA	Evolutionary Algorithm
GSS	Golden Section Search
FMRLC	Fuzzy Model Reference Learning Controller
NC	Neural Controller
NN	Neural Network
NF	Neurofuzzy
FPGA	Field Programmable Gate Array
VHDL	Very High speed integrated circuit Hardware Description Language

1. Introduction

1.1 Vehicle suspension systems

By the early 19th century, most British horse carriages were equipped with springs; wooden springs in the case of light one-horse vehicles, and steel springs in larger vehicles. These were made of low-carbon steel and usually took the form of multiple layer leaf springs (Adams 1971).

Automobiles were initially developed as self-propelled versions of horse drawn vehicles. However, horse drawn vehicles had been designed for relatively slow speeds and their suspension was not well suited to the higher speeds permitted by the internal combustion engine (IC). In 1901, Mors of Germany first fitted an automobile with shock absorbers. In 1920, Leyland used torsion bars in a suspension system. In 1922, independent front suspension was pioneered on the Lancia Lambda car and became more common in mass market cars from 1932 (Jain and Asthana 2006).

The drawbacks of excess vehicle body vibrations are that: reduced vehicle-frame life, reduced limitation of vehicle speed, negative biological effects on passengers and detrimental consequences to cargo. Modern active suspensions are tried to provide good handling characteristics and to improve riding comfort under harmful vibrations caused by road irregularities.

Very important objectives set for designing a suspension controller system deal with the riding comfort, suspension relative motion, road gripping and body inclination. The riding comfort is directly associated with the level of vertical vibration sensed by the passengers when travelling on a rough road. The suspension relative motion describes the displacement magnitude between the sprung and unsprung masses. The road gripping is associated with the friction forces between the tyre and road surface. The body inclination occurs during sudden manoeuvres such as braking and cornering. The difficulties faced in the active suspension design often are: the performance requirements are usually conflicting and the model used in the control design contains unresolved uncertainties.

As shown in Figure 1.1, there are three types of suspension system: passive suspension systems, semi-active suspension systems and active suspension systems.

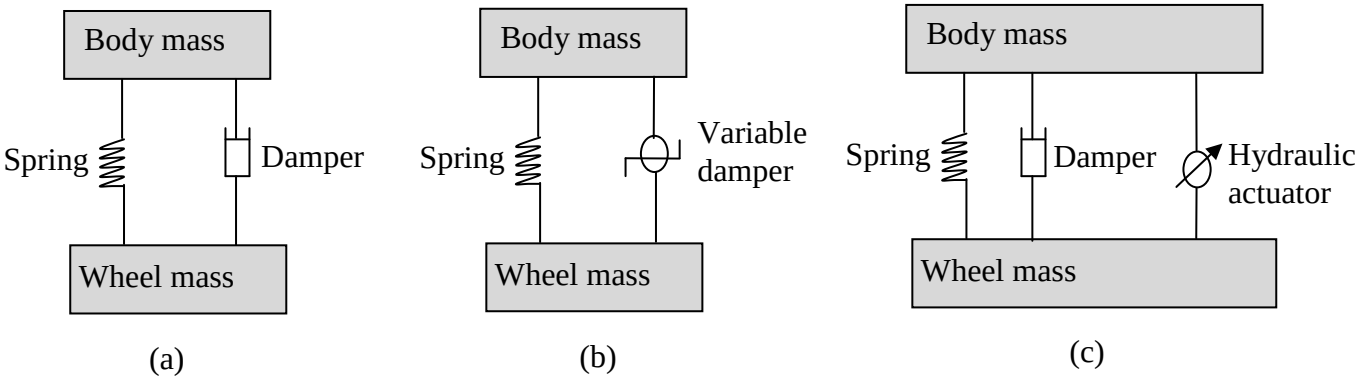


Figure 1.1 (a) Passive Suspension, (b) Semi-Active Suspension, (c) Active Suspension

The passive suspension systems consist of spring and damper or shock absorber model as illustrated in Figure 1.1a, which are most commonly used and may be found in most vehicles. The model systems of springs and dampers are assumed to have linear characteristic, whereas most of shock absorbers exhibit nonlinear relationship between the force and velocity. In passive systems, the suspension damping and stiffness are fixed coefficients, hence, have no mechanism for feedback control.

The semi-active suspension systems have the capability of modifying the damping coefficient of the suspension systems dependent on outside demands. It can be achieved by using a mechanical device called variable damper, which is used in parallel with a conventional spring as shown in Figure 1.1b. It requires some form of measuring devices (sensors) and controller board for tuning the damping coefficient. Many researchers proposed very attractive and efficient semi-active suspension systems featured by electrorheological (ER) fluids or magnetorheological (MR) fluids.

The active suspension systems supplies external forces to increase the passengers comfort, safety and road handling. It consists of actuators, which can be hydraulic, pneumatic or electro mechanic, and passive suspension system (spring plus damper) which is secured in parallel with the actuator device as shown in Figure 1.1c. The active suspension requires sensing devices, control system and an external power source. The sensing devices, which can be accelerometers, force transducers or potentiometers, located at different points of the vehicle to measure the motions of the body. The control system is employed to supply suitable control commands to the actuators. The external power source, which can be hydraulic pump or electrical pump, draws power from the vehicle's engine to generate appropriate external forces for the suspension systems.

In general, the road handling and safety rules request harsh suspensions, while the passengers comfort feeling require a soft damping. Driver action and vehicle speed effect on safety aspects. However, the perception of rider's comfort depends on the environment such as road excitation. Therefore, the active suspension system should be used to solve the conflict between riding comfort and road handling requirements. Existing active suspension systems have several drawbacks, such as high energy consumption and complexity. Therefore, by using the active suspension systems in modern vehicles, these problems should be solved.

The active suspension elements demonstrate a nonlinear behaviour especially in the case of large deflections. The suspension forces generated by the hydraulic actuators introduce nonlinearities to the system. The dynamic characteristics of suspension components, i.e. dampers and springs, also have nonlinear properties, which change during the vehicle life cycles. The nonlinearity behaviours of the active suspension system make the control design for the suspension system very difficult.

Several models and controllers have been developed in attempts to enhance and improve the riding and handling qualities in modern vehicles. In most of the studies related to the active suspension control system, a quarter or half vehicle model is usually adopted to design the control system. The quarter vehicle models only deal with vertical motions of the vehicle body and do not take into account the pitch and roll motions. While half vehicle models only deal with the vehicle body vertical motions and the pitch motions, but they do not include the roll motions of the vehicle body. On the other hand, there are several researchers who used a full vehicle model to design the control system to take into account all the vehicle motions such as vertical motions, pitch motions and roll motions.

Due to the complex mathematical relationships in nonlinear active suspension system, most of those researchers approximate the active suspension systems as linear models when designing the controllers. Since the nonlinearity and uncertainties inherently and significantly exist in suspension systems, the nonlinear effect has to be taken into account when designing the control units for the active suspension systems.

1.2 Literature review of control for suspension systems

A literature search in control systems by the author for vehicle suspension models shows many researchers have proposed control systems for linear or nonlinear quarter, half or full vehicle models. For simplification, linear model of spring and damper are usually used to

design the control system. A few of researchers used the nonlinear model of vehicle suspension systems, which include electro-hydraulic actuators and nonlinear characteristics of dampers and springs, to propose a nonlinear controller for suspension systems.

To create an appropriate structure for discussing the literature review of control systems for suspension models, the following topics are to be addressed:

1) Control systems for linear quarter vehicle model.
2) Control systems for linear half vehicle model.
3) Control systems for linear full vehicle model.
4) Control systems for nonlinear quarter vehicle model.
5) Control systems for nonlinear half vehicle model.
6) Control systems for nonlinear full vehicle model.

1.2.1 Control systems for linear quarter vehicle model

Many researchers have proposed the control systems for the linear quarter vehicle model. The model consists of a sprung mass, i.e. vehicle body mass, connect to single unsprung mass, i.e. wheel mass, via suspension system. In this case, the characteristic of the spring and damper are considered as linear. The model has two degrees of freedom: one degree for sprung mass and one for unsprung mass.

Peters et al. (1993) proposed a control method based on fuzzy logic system theory to improve the passengers comfort. It considerably reduces the number of rules in the input matrix and adapts the inference rules without the need of learning phase. The proposed control system delivered an improved riding comfort to drivers while guaranteeing high level of safety. The performance of the system shows the advantage over a conventional suspension system.

Sung et al. (2007) presented real-time control characteristics of electrorheological suspension system via a fuzzy sliding mode control algorithm. Equation of motion of the proposed system is derived and the discrete time sliding mode controller with system uncertainties is

formulated. A stable sliding surface is then designed and followed by the formulation of the discrete-time sliding mode controller which consists of a discontinuous part and an equivalent part. The fuzzy control algorithm is also adopted to enhance system robustness to the mass variation and reaching time to the sliding surface. The robustness of the proposed controller is tested under various road conditions.

Sharkawy (2005) described Fuzzy Logic Controller (FLC) and Adaptive Fuzzy Controller (AFC) schemes for the automobile Active Suspension System (ASS). The design objective is to provide smooth vertical motion so as to achieve the road holding and riding comfort over a wide range of road profiles. To achieve robustness and adaptability controller, the development of an adaptive fuzzy controller have been investigated. In comparison with the LQR, the FLC has similar performance as it can be noticed from simulations. Indeed, the output surface of the FLC can be shaped so as to get this performance; which is an inherent property of fuzzy systems. However, this kind of shaping requires a trial and error procedure. To overcome this difficulty, an AFC was developed based on the Lyapunov direct method. The adaptive law ensures fast convergence of the parameter vector, which enforces robustness against terrain irregularities. Computer simulations demonstrated that the proposed AFC achieves the best performance.

Jianmin and Yang (2007) applied a fuzzy control method based on adaptive technology for active suspension system. In order to improve the passengers comfort and road handling, an analytic fuzzy control algorithm is advanced. There is no membership function choice of fuzzy subset for input and output of the controller. The factor of the analytic fuzzy controller is adjusted online according to adaptive method. By using this advance algorithm, the disadvantages of fuzzy logic can be avoided. In addition, the advantages of fuzzy logic can be emphasised.

Kuo and Li (1999) presented a genetic algorithm based fuzzy proportional-plus- integral/ proportional-plus-derivative (PI/PD) controller for active suspension system to improve the riding quality. The fuzzy PI controller is employed to reduce the sprung mass acceleration due to the step-like road surface. In the situation of common terrain, the fuzzy PD controller will be employed to improve the riding comfort for the vehicle. By using the merit of genetic algorithm, the optimal decision-making rules for both types of controllers are constructed. The performance of the designed controller is compared with conventional optimal controller and passive system. The simulation results demonstrated that the fusion of genetic algorithm and fuzzy controller can provide passengers much more ride comfort.

Kumar (2007) designed a proportional derivative controller for an active suspension system to improve riding comfort. The optimal parameters of proportional and derivative gains are selected by using genetic algorithm. In the PD controller the proportional term provides inputs that correct for "current" errors. Derivative term provides "anticipation" of upcoming changes. Various road profiles applied as excitation road inputs to examine the performance of the proposed controller. The results show that the performance of the active system has improved as both the peak overshoot of sprung mass acceleration, suspension travel and tire deflections have reduced compared to passive system.

Foda (2001) presented a neurofuzzy controller for semi-active suspension system to improve both suspension travel and riding quality in terms of both vehicle acceleration and dynamic tyre loading. The neural network is used to learn rules and infer conclusions. Simulation of a quarter vehicle model with the designed fuzzy controller under various road conditions is used to confirm the validity of the proposed controller.

Biglarbegian et al. (2008) proposed a neurofuzzy strategy to implement a semi-active suspension. The proposed method consists of two parts: a neurofuzzy controller, to establish an efficient controller strategy to improve riding comfort and road handling (RCH), and an

inverse mapping based on the neural networks to estimate the semi-active suspension current. The inverse mapping is used to modify the rules of fuzzy system to supply the current needed to control the semi-active suspension system. To verify the performance of the fuzzy controller, two sets of results are reported. Firstly, an experimental analysis was performed to demonstrate the effectiveness of the fuzzy controller in comparison with the benchmark skyhook and Rakheja–Sankar controllers. Secondly, a random input was considered to examine the robustness of the neurofuzzy controller in comparison with the other adopted controllers.

In the above investigations, the quarter linear vehicle model was used to design a control system. By designing the control system with the quarter vehicle model, just the vertical displacement can be improved. Therefore, other degrees of freedom which should be taken into account to improve all the control objectives have been ignored.

1.2.2 Control systems for linear half vehicle model

In the case of the quarter vehicle model, just the heave motion can be investigated. To take into account the pitch motion, some researchers used a half vehicle model to design the control system. The half model consists of a single sprung mass connected to two unsprung masses, one at each corner, via suspension systems. Therefore, it has four degrees of freedom: two degrees of freedom for the heave motion and the pitch motion for sprung mass; and two degrees of freedom for the two heave motions of the unsprung masses. In this case, the nonlinear model of the actuator not include in the mathematical equations of the half vehicle model. The dynamic characteristics of suspension model, dampers and springs, are assumed linear.

Krtolica and Hrova (1990) investigated all optimal suspension control problem for a half vehicle model. Using an efficient equivalent representation, a complete analytical solution of the related fourth-order LQ problem was obtained. The problem structure and associated

analytical results are used to deduce generic properties of the optimal solution, which in turn form a basis for global, centralized, and decentralized optimal suspension performance investigation.

Lin and Huang (2004) developed a fresh nonlinear backstepping scheme for the control of half vehicle active suspension systems to improve the inherent tradeoff between riding quality and suspension travel. The novelty of this active suspension design is in the use of two particular nonlinear filters at both the front and rear wheels. The effective bandwidths of these two nonlinear filters depend on the magnitudes of the front and rear suspension travels, individually. When suspension travel is small, the proposed controllers soften the suspension for enhancing passenger comfort. The results demonstrated that the closed-loop system performance is good enough to minimize the displacements and accelerations of both the front and rear body.

In Reference by Karkoub and Zribi (2001), an optimal control scheme used to generate a control signal to magnetorheological dampers (MR) which are used as variable dampers in semi-active suspension systems. A half vehicle model including passenger dynamics is used in this investigation. Two MR dampers attached to the front and back tyres are used to reduce the vibrations of the passenger seats and the vertical motion of the vehicle body.

Yue et al. (2008) presented a fuzzy logic control for the active suspension system. The model used in this investigation is described by a linear system with five degree of freedom. It consists of single sprung mass connected to two unsprung masses, the seat mass and the passenger mass. Road roughness height is modeled as a filtered white noise stochastic process. By using the MATLAB/SIMULINK software, the simulation model is obtained. In particular, a mechanical dynamic animation of five degree of freedom half-body vehicle suspension system is obtained, with the aid of the software ADAMS and the media player.

Moon and Kwon (1998) designed a fuzzy controller for active suspension systems to improve the riding quality and maintain vehicle manoeuvrability. The genetic algorithm is applied to select the optimal values of membership function of the fuzzy controller rules. To measure the performance of the designed genetic based fuzzy active suspension system, three road disturbance models are designed to simulate the actual road conditions. The performance of the designed system is evaluated with respect to these disturbance models.

Wu et al. (2005)) developed an artificial neural based fuzzy scheme to design a self learning optimal controller for active suspension systems. A robust optimal fuzzy controller is designed based on the proposed fuzzy model to improve riding quality and to support appropriate movement in the suspension system. T-S fuzzy model from a six layer self-constructing neural fuzzy inference network was obtained.

The drawback of investigations above is that the third rolling movement had not been taken into account; therefore, the rolling angle can not be improved when sharp cornering occurs.

1.2.3 Control systems for linear full vehicle model

To take in to account all the vehicle body motions such as heave motion, pitch motion and roll motion some researchers presented a full vehicle model to design the control systems. In this case, the model has seven degrees of freedom because the sprung mass of vehicle body has three degrees of freedom (heave motion, pitch motion and roll motion) and each unsprung mass has one degree (the heave motion).

Braghin et al. (2007) designed a feedback controller for active and semi-active suspension systems to improve riding comfort with respect to traditional suspension systems. The proposed controller is allowed to control separately the vehicle body motions of heave, pitch and roll taking into account the dynamics of the active and semi-active actuators. Simulation results showed that significant improvements of ride comfort can be achieved with respect to passive absorbers.

Ikenaga et al. (2000) investigated a control approach of combining a filtered feedback control scheme and an input decoupling transformation for active suspension systems. Motions of the sprung mass (car body) above and below the wheel frequency modes ($\omega_0 \cong 57$ rad/s) are mitigated by using active filtering of spring and damping coefficients through inner control loops (ride controller) plus skyhook damping of heave, pitch and roll velocities through outer control loops (attitude controller). The inner ride control loops and the outer attitude loops are blended with the input decoupling transformation.

In the reference by Esmailzadeh and Fahimi (1997), an optimal active suspension is designed for a linear seven degrees of freedom model. The optimization is achieved via minimization of a quadratic cost function and subsequently this optimal system is used as a model for the vehicle which in turn is being controlled by the Model Reference Adaptive Control method. The actuators are controlled with optimal full state vector feedback. After determining feedback coefficients, the responses of active and passive systems are compared.

Chamseddine et al. (2006) designed a sliding mode Controller for active suspension systems to diagnose and accommodate faults. Two levels are investigated: the level of vehicle prototypes and the level of industrial vehicles. Difficulties of vehicle instrumentation and possible solutions are discussed. The performance of the proposed controller is compared with the corresponding passive suspension system. Simulation result stated that the controlled active suspension system improved the riding comfort for the frequencies below the wheel frequency.

Choi et al. (2002) presented a feedback control of full vehicle suspension system with a feature of magnetorheological (MR) dampers. A H_∞ controller which has inherent robustness against system uncertainties is formulated by treating the sprung mass of the vehicle as uncertain parameter. This is accomplished by adopting the loop shaping design procedure. For the demonstration of a practical feasibility, control performance characteristics for vibration

suppression of the proposed MR suspension system are evaluated under various road conditions through the hardware-in-the-loop simulation.

Onat et al. (2007) described a design of gain-scheduling nonlinear controller for active suspension system to maximize the riding comfort and to minimize the suspension deflection. The proposed method is based on a Linear Parameter Varying (LPV) model of the full vehicle system. The variations in the suspension deflection and mass are chosen as the scheduling parameters. Different road profiles, having high and low bumps, hollows and combinations of the two are used to test the performance of the proposed controller.

Yagiz and Sakman (2005) designed a robust controller for active suspension system without losing the suspension working space to obtain the desired improvement in the riding comfort. Force actuators are placed parallel to the suspensions and a non-chattering sliding mode control is applied. The performance of the proposed approach without suspension gab in the case of travelling over a step road profile is compared with the performance of the conventional approach to test the robustness of the proposed controller. Simulation results stated that this type of controller with the proposed approach improved the riding comfort significantly with robust behaviour.

In the reference by Liu et al. (2005), a full vehicle model used to design a fuzzy sliding mode switch hyperplane for semi-active suspension systems to improve the riding comfort and road handling.

In the reference by Zheng et al. (2007), a fuzzy-sliding mode controller is presented to control the dynamics of semi-active suspension systems of a vehicle using magnetorheological (MR) fluid dampers. The performance of the semi-active suspension systems is evaluated in both the time and frequency domains. The numerical simulations are carried out to check the effectiveness of proposed fuzzy-sliding mode controller. The obtained results demonstrate

that the proposed controller can effectively suppress the vibration of vehicles and improve the riding comfort.

Cheng and Li (2007) presented a fuzzy logic control for active suspension system to improve the riding comfort and to reduce the suspension deflection. The evolutionary programming method is used to determine optimum design parameters of the fuzzy logic control by minimizing a fitness function that consists of all the desired performance of a full-car model. The optimal active suspension control scheme is also introduced to compare the results of this controller with the fuzzy controller. To test the performance of the proposed controller two kinds of road profiles, including a bumped road and a white noise random road, are demonstrated.

Park and Kim (1998) designed a decentralized variable structure controller for active suspensions of a full vehicle model. The performance of the proposed controller is compared with the Linear-quadratic regulator. The results showed that the proposed controller improved the performance of the sprung mass in the heave, pitching and rolling directions when the vehicle travelling on a normal road or through an unequal bump.

Although the investigations above used a full vehicle model to design the control system, the nonlinearity of the suspension parts had not been taken into account. As mentioned above, since the nonlinearity inherently exists in vehicle suspension systems, it must be included when designing an accurate controller.

1.2.4 Control systems for nonlinear quarter vehicle model

As mentioned before, due to the complex mathematical relationships in nonlinear active suspension system, most researchers approximate the active suspension systems as linear systems when designing the controllers. The suspension forces generated by the electro-hydraulic actuators are inherently nonlinear. Also, the dynamic characteristics of suspension components, i.e. dampers and springs, have nonlinear properties. Since the nonlinearity

inherently and significantly exists in suspension systems, the nonlinear effect must be taken into account when designing the control systems.

Tahboub (2005) presented a control method applied to two degrees of freedom systems with nonlinear suspension dynamics. In this reference, the nonlinear structure of the hydraulic actuator did not take into account when the mathematical equations of the model system are derived. An extended observer is used to estimate the nonlinear effects as well as the road profile. Based on the estimation, a simple adaptive feedback-linearization scheme is implemented to cancel the effects of the nonlinear suspension.

In the reference by Renn and Wu (2007), two control schemes are designed for the quarter vehicle nonlinear active suspension system to improve riding comfort and to achieve control performance. The first one is a conventional PID controller obtained from Ziegler-Nichols method and the second one is the neural network controller. This work emphasized on the engineering aspects including the design of a servo-hydraulic test rig, the implementation technology of the controllers and the subsequent experimental study.

Alleyne and Hedrick (1995) developed nonlinear sliding control for active suspension systems. To reduce the error in the model, a standard parameter adaptation scheme, based on Lyapunov analysis was introduced. A modified adaptation scheme, which enables the identification of parameters whose values change with the regions of the state space, was then presented. The adaptation algorithms are coupled with the control algorithm and the resulting system performance was analyzed experimentally. The performance is determined by the capability of tracking a specified force by the actuator output. The performance of the active system, with and without the adaptation, is analyzed.

Chen and Huang (2006) presented an adaptive sliding controller for active suspension systems of a quarter vehicle model with hydraulic actuator. The function approximation technique is used to represent the uncertainties with finite combinations of some basic functions, and the

Lyapunov method is employed to find updated laws for the coefficients of the approximate series. The results of the proposed controller are compared with passive system. The comparison showed that the riding comfort of the vehicle system was improved.

Tusset et al. (2008) presented control strategies for nonlinear semi-active suspension system. Two different approaches are used for modelling and control of the mechanical and electrical devices in the suspension system. First, in order to design the liner feedback dumping force controller, the control problem is formulated and resolved. Then the values of the control dumping force functions are transformed into electrical control signals by the application of a fuzzy logic control method.

Fialho and Balas (2000) designed the nonlinear controller based on the linear parameter-varying control techniques for the active vehicle suspension system. The parameter-dependent weighting functions are used to design active suspensions that stiffen when the suspension limits were reached. The controllers used only suspension deflection as a feedback signal.

Ikenaga et al. (1999) used a filtered feedback control scheme and novel compressible fluid suspension system to design an active suspension control for a quarter vehicle suspension systems. Analysis of feedback for the mechanical subsystem shows that motions of the sprung mass above and below the wheel frequency can be mitigated using skyhook damping plus active filtering of spring and damping coefficients. The frequency-dependent filtering is accomplished through an outer control loop, which generates the target strut force, plus an inner force control loop.

Campos et al. (2000) designed a backstepping based fuzzy logic scheme for active suspension system. The servo-valve dynamics are also included. A fuzzy logic system is used to estimate the dynamics of nonlinear hydraulic strut. The backstepping fuzzy logic scheme was shown to give superior performance over passive suspension and active suspension control using conventional proportional-integral-derivative (PID) control. The FL system is adapted in such

a way as to estimate on-line the unknown hydraulic dynamics and provide the backstepping loop with the desired servo-valve positioning so that the scheme becomes adaptive, guaranteeing bounded tracking errors and parameter estimates.

D'Amato and Viassolo (2000) proposed a fuzzy logic controller for active suspension systems to minimize vertical vehicle body acceleration and to avoid hitting suspension limits. A controller consisting of two control loops is proposed to achieve this goal. The inner loop controls a nonlinear hydraulic actuator to achieve tracking of a desired actuation force. The outer loop implements a fuzzy logic controller which interpolates linear locally optimal controllers to provide the desired actuation force. Final controller parameters are computed via genetic algorithm based optimization.

In the reference by Huang and Chen (2006), a novel model-free adaptive sliding controller is proposed to suppress the position oscillation of the sprung mass in response to road surface variation. In addition, a fuzzy scheme with online learning ability is introduced to compensate the functional approximation error for improving the control performance and reducing the implementation difficulty. The important advantages of this approach are to achieve the sliding mode controller design without requirement of the system dynamic model and to release the trial-and-error work of selecting approximation function.

In the reference by Spentzas and Kanarachos (2002), a neural network is used to design a robust controller for nonlinear quarter vehicle suspension systems with hydraulic actuators to minimize the vertical acceleration imposed on the body and passengers. The neural network is obtained by a Taylor approximation of an unknown non-linear control function. Because of the existence of numerous local minima of the neural network, an evolutionary algorithm is used to solve the resulting neural network problem.

In the reference by Huang and Lin (2007), a novel neural network based sliding mode control is proposed by combining the advantages of the adaptive, radial basis function neural network

and sliding mode control strategies to design an accurate controller. It has online learning ability for handling the system time-varying and non-linear uncertainty behaviours by adjusting the neural network weightings and/or radial basis function parameters.

1.2.5 Control systems for nonlinear half vehicle model

Like the linear half vehicle model, the nonlinear half vehicle model has four degrees of freedom: heave motion and pitch motion for sprung mass; and each unsprung mass has one degree. The roll motion does not take into account in this case.

Yoshimura et al. (1999) proposed linear and fuzzy logic controllers for the nonlinear active suspension system. The former is obtained by vertical acceleration of the vehicle body as the principal source of control, and the latter is obtained by using fuzzy-logic control as the complementary control. In the derivation of fuzzy control rules, linear combinations of the vertical and rotary velocities and displacements of the vehicle body are denoted as the input variables. Emphasis is laid on the minimization of the vertical and rotary acceleration of the vehicle body from the view-point of passengers riding comfort.

Feng et al. (2003) designed a new control strategy for active suspension system by using combined control scheme. A genetic algorithm based self tuning PID controller and a fuzzy logic controller are used in two loops. The PID controller is used to improve the riding comfort, while the fuzzy controller is used to minimize the pitch acceleration. A genetic algorithm is used to tune the parameter of PID controller; the scaling factors; the gain values and the membership functions of fuzzy logic controller on-line. The stability and adaptability are also showed even when the system is subject to severe road conditions, such as a pothole, an obstacle or a step input. Comparisons are made between the conventional passive suspensions and the active vehicle suspension systems by using linear fuzzy logic control; the combined PID and fuzzy control without parameters self-tuning; and the new proposed control system with GA-based self learning ability. The results show that the new proposed

controller improves vehicle riding comfort performance significantly and offers better system robustness.

Shariati et al. (2004) designed a robust H_∞ controller for active suspension system with hydraulic actuator in a cascade feedback structure to improve the riding comfort and road holding. The proposed controller has two loops. In the inner loop a proportional controller is used, while a robust H_∞ controller is used in the outer loop. Two types of robust H_∞ controller are designed. Firstly, unstructured uncertainty is not considered in the design procedure and secondly, the controller is designed with consider action of uncertainty. Each of these controllers is designed in a decentralized fashion and the vehicle oscillation in the human sensitivity frequency range is reduced to a minimum. For both cases the random input as road roughness used to examine the efficiency of the proposed controller.

Wn et al. (2004) designed an optimal fuzzy controller for an active suspension system based on Takagi-Sugeno (T-S) fuzzy control system. First, the half-car suspension system dynamics was converted to a T-S fuzzy model. Then each local controller design was derived by applying linear optimal control theory and the global effect can be achieved by fuzzily blending with local optimal controller if a sufficient condition can be met. The results showed that the performance of the proposed controller is much better than the conventional passive suspension system.

1.2.6 Control systems for nonlinear full vehicle model

The works in this subsection are focused on the design of a control system for nonlinear full vehicle model to improve the performances of the vehicle at sharp manoeuvre occurring such as travelling on a rough road, cornering or braking.

Bui et al. (2002) presented a robust H_∞ and nonlinear adaptive control methods for active suspension systems. The controller showed good performance, i.e. small gains from the road disturbances to the heave, pitch and rolling accelerations of the vehicle body. H_∞ controller

is designed so as to guarantee the robustness of a closed-loop system in the presence of uncertainties and disturbances. The system parameter variations are taken into account by multiplicative uncertainty model and the system robustness is guaranteed by the small gain theorem. The nonlinearity of a hydraulic actuator is handled by nonlinear adaptive control based on the back-stepping method.

Du and Zhang (2009)) presented a Takagi-Sugeno model based fuzzy control design approach for a full vehicle nonlinear active suspension system with hydraulic actuators. The T–S fuzzy model is first applied to represent the nonlinear uncertain electro-hydraulic suspension. Then, a fuzzy state feed-back controller is designed for the obtained T–S fuzzy model with optimized H_∞ performance for riding comfort by using the Parallel-Distributed Compensation (PDC) scheme. Simulations result shown that the designed controller can achieve good suspension performance despite the existence of nonlinear actuator dynamics, sprung mass variation, and control input constraints.

Yagiz and Sakman (2006) designed and checked the performance of the fuzzy logic control for a semi-active suspension system of a nonlinear full vehicle model. The linear combination of the vertical velocities of the suspension ends and the accelerations of the suspension connection points to the body are used as input variables. The results obtained with the vehicle system controlled by fuzzy logic were compared with those from the conventional proportional-integral-derivative (PID) controlled system and passive suspension system. The results shown the riding comfort and road holding have been improved.

Huang et al. (2009) presented a fuzzy feedback linearization control of nonlinear multi-input/multi-output systems for tracking and almost disturbance decoupling performances based on the fuzzy logic control. The main contribution of this study is to construct a controller, under appropriate conditions, such that the resulting closed-loop system is valid for any initial conditions.

Guclu and Gulez (2008) designed a robust neural network for full vehicle nonlinear active suspension system with a passenger seat model. In the NN structure, the Fast Back-Propagation Algorithm (FBA) is employed. The user inputs a set of 16 variables while the output from the NN consists of f1–f16 non-linear functions. Furthermore, the PMSM controller is also determined using the same NN structure. Various tests of the NN structure demonstrated that the model is capable of giving highly sensitive outputs for vibration condition, even using a more restricted input data set. The non-linearity occurs due to dry friction on the dampers. The vehicle body and the passenger seat using PMSM are fully controlled at the same time. The time responses of the non-linear vehicle model due to road disturbance and the frequency responses are obtained. Finally, uncontrolled and controlled cases are compared. The result shown that seat vibrations of a non-linear full vehicle model are controlled by a NN-based system with almost zero error between desired and achieved outputs.

From the literature review, it can be seen that the intelligent control design problems for the full vehicle nonlinear active suspension system with hydraulic actuators is not covered. Also, the frictional forces due to rubbing of the piston with the actuator walls, which is opposite to the hydraulic forces provided by the actuators, are not covered either. Most researchers, who dealt with the full vehicle model, neglected the effect of the fractional forces. This frictional force is found to be of a significant magnitude (>200 N) and cannot be neglected (Rajamani and Hedrick 1995). Therefore, the effect of the frictional forces should be taken into account when deriving the mathematical equations of full vehicle nonlinear active suspension systems with hydraulic actuators.

In the work by the author, an Adaptive Neurofuzzy Inference System (ANFIS) based on intelligent control for the full vehicle nonlinear active suspension system will be designed. An optimal Fraction Order $PI^{\lambda}D^{\mu}$ (FOPID) controller will be used to control the master controlled

system. An Evolutionary Algorithm (EA) is proposed to modify the control parameters of the FOPID. The data which have been obtained from the FOPID controller will be used as a training data to modify the parameters of the neurofuzzy controller. The SIMULINK tool boxes in MATLAB will be used to simulate the controlled system with the proposed controller. A Very High speed integrated circuit Hardware Description Language (VHDL) will be used to implement of neurofuzzy (NF) controller. Xilinx ISE Project Navigator Version 10.1 is used as environment to synthesis the VHDL codes and to generate the configuration file which is used to program the FPGA boards. For the simulation results, ModelSim XE III 6.4b simulation program will be used to confirm that the FPGA boards work like neurofuzzy controllers. It provides a novel technique to design NF controller for the full vehicle nonlinear active suspension system with hydraulic actuators and to implement the proposed system using the FPGA board.

The active suspension systems are very good to bring significant performance of vibration isolation, but they are of high energy consumption and complex. Active suspension is essential technology for high performance machine system. Therefore, to use the active suspension in the research, the energy consumption by the suspension system must be minimised.

When a vehicle is driving on a bumpy road, plus driver's acceleration and deceleration operations, there will be shocks between the sprung mass and the unsprung mass. This part of the mechanical power is normally converted into heat power by a damper and is dissipated in a natural way. If the wasted energy can be reclaimed in a proper way, the overall energy consumption demand for the vehicle can be reduced. In this thesis, the vibration excited by road unevenness will be treated as a source of mechanical energy. It is being converted into electrical energy to compensate the energy consumption by the active suspension. To achieve this task, an electromagnetic active suspension system will be introduced. The power

generated from this device will contribute to run the electrical pumps of the hydraulic actuators.

1.3 Existing problems and the research objectives

As mentioned before, the main objective of designing the controller for a vehicle suspension system is to reduce the discomfort felt by passengers which arises from road roughness and to improve the road handling associated with the pitching and rolling movements. This necessitates very fast and accurate controller to meet the key control objectives as much as possible. Therefore, a novel neurofuzzy controller for the full vehicle nonlinear active suspension system with hydraulic actuators will be designed in this project. The advantage of this controller is that it can handle the nonlinearities faster than other control types. It is believed that, this is the first time to use the neurofuzzy control method to design the controller for full vehicle nonlinear active suspension systems with nonlinear hydraulic actuators.

The objectives of this work are described as follow:

i. Deriving the mathematical equations for the full vehicle nonlinear active suspensions systems with nonlinear hydraulic actuators.
ii. Designing robust controllers for the full vehicle nonlinear active suspensions systems with nonlinear hydraulic actuators including Optimal Frictional $PI^{\lambda}D^{\mu}$ Controller, Fuzzy Model Reference Learning Controller, Neural Network Controller and Neurofuzzy Controller.
iii. Comparison between the performances of the robust controllers based on the measurement of the cost functions will be carried out.
iv. Selection of the good performance robust controller.
v. Implementation of the selected controller for suspension systems using FPGA platform.

vi. Introduction of an electromagnetic active suspension system to convert the mechanical energy into electrical energy to compensate for the energy consumption by the active suspension.

1.4 Organisation of the thesis

In Chapter 2, a full vehicle nonlinear active suspension model including hydraulic actuators, nonlinear dampers and springs will be proposed with structural and analytical details. The nonlinear frictional forces due to rubbing of piston seals with the cylinders wall inside the cylinder are taken into account to find the actual supplying forces generated by the hydraulic actuator.

In Chapter 3, an optimal Fractional Order $PI^{\lambda}D^{\mu}$ (FOPID) controller is designed for a full vehicle nonlinear active suspension system. The optimal values of the FOPID controller parameters for minimizing the cost function are selected using an Evolutionary Algorithm (EA), which offers optimal solution to a multi-dimensional rough objective function. The fitness parameters of the FOPID controller are selected from ranges of reliable values depending on survival-to-the-fitness principle borrowed from the biology science. The results of the response of full vehicle nonlinear suspension system using the FOPID controller will be compared with the corresponding responses of a full vehicle nonlinear suspension model without controller (passive suspension system). The controlled suspension system has been investigated under typical vehicle manoeuvres: cruising on rough road surface, sharp braking and cornering.

In Chapter 4, a Fuzzy Model Reference Learning Controller (FMRLC) system is designed to be capable of improving the performance of the closed loop control system by generating the command inputs to the controlled process and utilizing feedback information from the controlled process. Four fuzzy controllers will be trained and applied to individual actuators

in the vehicle suspension system. The proposed fuzzy controllers adjust the hydraulic actuator forces to minimize the vertical displacement in each suspension point when travelling on rough roads and to reduce the tendency of vehicle to rollover during sharp manoeuvres such as breaking and cornering. Six types of the disturbances will be investigated to verify the robustness of the proposed controller. To show the effectiveness and robustness of the proposed controller, comparison is made between the cost function of the system with proposed controller and the cost function of the passive system for each type of the disturbances.

In Chapter 5, the neural network has been used to design a control system for the nonlinear active suspension system with hydraulic actuator. The optimal Fractional Order $PI^{\lambda}D^{\mu}$ controller, which has been designed in Chapter 3, will be used to design the neural control. The parameters of neural network have been trained using Levenberg-Maquardt Algorithm (LMA) to have the same behaviour of optimal Fractional Order $PI^{\lambda}D^{\mu}$ controller. The performance of the neural network will be improved by tuning the scaling gains included in the structure of the neural network. The robustness of the proposed controller will be tested by applying six types of disturbances. The results will illustrate the efficiency of the proposed controller and robustness against the disturbances.

In Chapter 6, the data obtained from the optimal Fractional Order $PI^{\lambda}D^{\mu}$ controller will be used as training data to design the NF controller. The learning ability of the neural network will be used to tune the parameters of the Fuzzy Inference Systems (FIS). The performance of the NF controller will be improved by tuning the scaling gains. The results show that the intelligent NF controller improves the dynamic response measured by mininmizing the cost functions. The performance of the proposed controller will be compared with the passive system. Comparisons will be made among the control systems designed in Chapters 3, 4, 5

and 6 depend on the measurement of the cost functions for selecting the best controller among them.

In Chapter 7, the implementation of the neurofuzzy controller using FPGA will be investigated. The Very High speed integrated circuit Hardware Description Language (VHDL) will be used to describe the operation of the Neurofuzzy controller that is being designed in Chapter 6. The Xilinx Integrated Software Environment (Xilinx ISE 10.1) will be used as an environment to synthesis the VHDL codes. After successfully compiling the design, the generated programmes files (configuration files) will be downloaded via USB port to program the FPGA. For displaying the simulation results, ModelSim XE III 6.4b simulation program will be used. In order to compare the expected responses of the FPGA design with the SIMULINK design, MATLAB M-file will be used to plot the data collected by the ModelSim program and the other data collected form the SIMULINK design to make sure the FPGA board will work like the neurofuzzy controllers.

In Chapter 8, the vibration excited by the road unevenness will be treated as excitation of mechanical energy. It will be converted into electrical energy to compensate the energy consumed by the active suspension system. To achieve this task, an electromagnetic active suspension system will be introduced. The power generated from this device will contribute as supply power to run the pumps of the hydraulic actuators.

Chapter 9 describes the conclusion of this thesis and the recommendations for the future work.

2. Development of a Model for the Full Vehicle Nonlinear Active Suspension Systems with Hydraulic Actuators

2.1 Basic considerations in the mathematical modeling

The design and analysis of control systems are based on the mathematical models of physical complex systems. A mathematical model of a dynamic system is described by a set of equations which can be obtained by using physical laws governing the particular systems. For description of the given system, several types of mathematical models can be proposed from the differential equation, the state space equation, the transfer function and the impulse response depend on its circumstances.

For obtaining a model, a compromise between the simplicity of the model and the accuracy of results of the analysis should be made. Such that, the neglected certain inherent physical properties of the system should not affect the accuracy of the results of the experimental study of the physical system.

For the active suspension problems, nonlinear models have been proposed by many researches such as Sharp and Goodall (1969) obtained the differential equations for vehicle motions using Lagrange's method and Joo et al. (2000) developed a nonlinear model of a

vehicle suspension system from standard kinematics and kinetics. On the other hand, some other researchers took into account the nonlinearity effect of hydraulic actuators to derive the mathematical model for the active suspension system. Szaszi et al. (2002) used Newton's law to derive the model of mathematical equations for a half vehicle nonlinear active suspension system with electro-hydraulic actuators and the nonlinear characteristics of dampers and springs. While the frictional forces inside the hydraulic actuator are neglected. Since the effects of the frictional forces are important for finding the actual forces supplied by the actuators and cannot be neglected (Rajamani and Hedrick 1995), therefore, in this project, the effect of the frictional forces will be taken into account to develop a model with high accuracy. The state space equation will be used to describe the final mathematical form of the full vehicle nonlinear active suspension systems with hydraulic actuators.

2.2 Nonlinear force's characteristics in full vehicle active suspension model

In this section, the mathematical model for full vehicle nonlinear active suspension systems with hydraulic actuators will be derived. The forces between the body of the vehicle and axle are generated by using nonlinear spring forces, nonlinear damping forces and nonlinear hydraulic forces. Firstly, the mathematical model of the nonlinear hydraulic actuator will be derived. Then, the derived model of nonlinear hydraulic actuator will be used to derive the mathematical model of full vehicle nonlinear active suspension system.

2.2.1 Nonlinear hydraulic actuator model

To understand the performance of suspension system and develop a robust controller for the system, it is essential to develop an accurate dynamic model for the hydraulic servo system. Therefore, a description of the dynamics for the fluid subsystem, the servovalve, the cylinder

and the load is required. The electro-hydraulic system here is a piston which is controlled by the voltage signal input to the servovalve. The cylinder is attached to sprung mass of the vehicle and connected in parallel with a passive system combination, including spring and damper. The hydraulic actuators are used to generate appropriate forces between the vehicle's body and the axle to enhance and improve the riding comfort and handling qualities in some advanced modern vehicles.

Figure 2.1 illustrates a physical model of the hydraulic actuator with nonlinear spring model and nonlinear damper model for quarter vehicle.

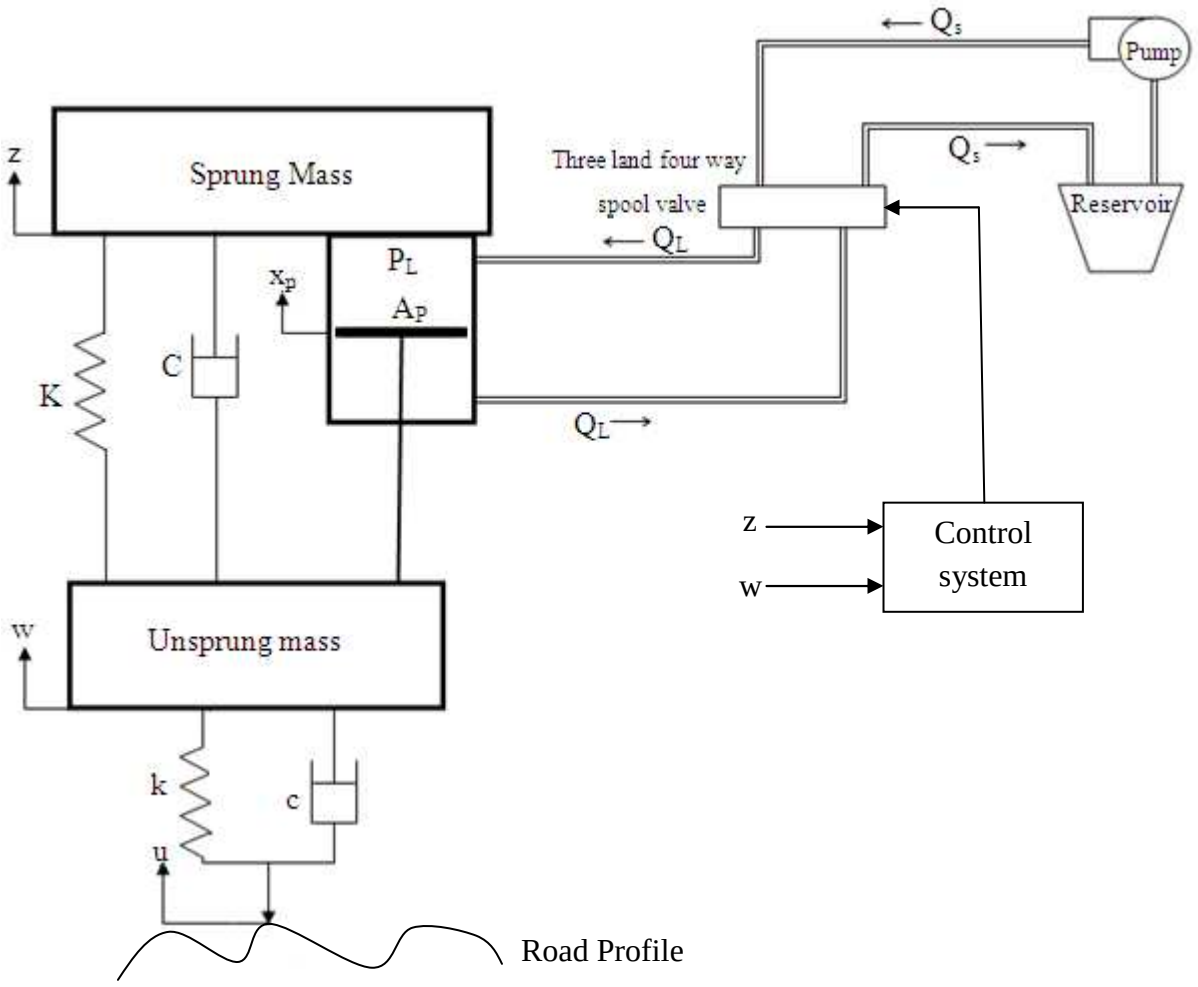


Figure 2.1 Physical model of nonlinear suspension system with hydraulic actuator

Before deriving the nonlinear mathematical equations of hydraulic actuator, the flow through orifices concept should be addressed.

2.2.1.1 Flow through orifices

Flow characteristics of orifices play a major role in the design of many hydraulic control devices. An orifice can be defined as sudden restriction of short length in the flow passage and may have a fixed or variable area. Depending on the law of continuity, the flow velocity through an orifice must increase above that in the upstream region. Figure 2.2 depicts the flow through an orifice. As clear the path between cross-section 1 and 2, the velocity of the fluid are increased from upstream velocity to jet velocity. The points along the jet where the jet area becomes a minimum is called the vena contracta. The vena contracta area (A_2) is equal to the contraction coefficient C_C times the orifice area (A_0).

$$A_2 = C_c\, A_0 \tag{2.1}$$

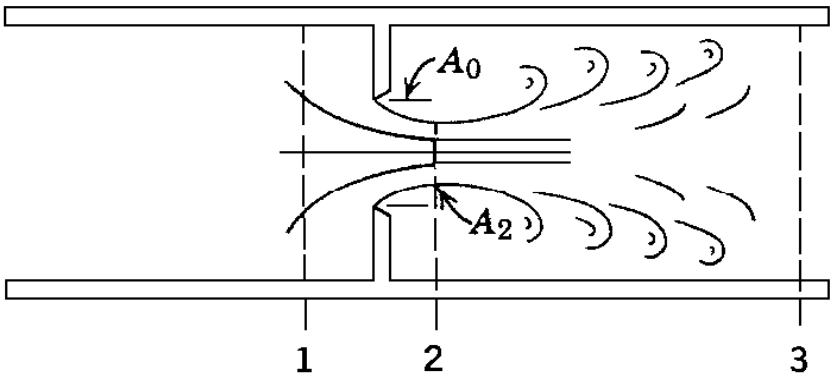


Figure 2.2 Flow through orificesMerritt (1969)

By applying the continuity equation the relationship between upstream velocity V_{e1} and jet velocity V_{e2}can be written as follows

$$A_1 V_{e1} = A_2 V_{e2} \tag{2.2}$$

By applying the Bernoulli's equation between cross-section 1 and 2, the pressure difference that is required for accelerating the fluid particles from upstream pressure to jet pressure can be given by

$$V_{e2}^2 - V_{e1}^2 = \frac{2}{\rho}(P_1 - P_2) \tag{2.3}$$

where ρ is the fluid mass density; P_1 is the fluid pressure at cross-section 1 and P_2 is the fluid pressure at cross-section 2.

By substituting Eq. (2.3) in (2.2) the jet velocity can be written as follows

$$V_{e2} = \left[1 - \left(\frac{A_2}{A_1}\right)^2\right]^{-1/2} \sqrt{\frac{2}{\rho}(P_1 - P_2)} \tag{2.4}$$

Due to the viscous friction, the jet velocity is slightly less than that given in Eq. (2.4). Therefore, the velocity coefficient C_v can be introduced into account.

The flow rate at cross-section 2 can be obtained from the following equation

$$Q = A_2 V_{e2} \tag{2.5}$$

Substitution of Equations (2.1) and (2.4) in Eq. (2.5) the flow rate can be written as

$$Q = \frac{C_v C_c A_0}{\sqrt{1 - \left(\frac{A_2}{A_1}\right)^2}} \sqrt{\frac{2}{\rho}(P_1 - P_2)} \tag{2.6}$$

The discharge coefficient C_d can be written as $C_d = \frac{C_v C_c}{\sqrt{1-\left(\frac{A_2}{A_1}\right)^2}}$, so that Eq. (2.6) can be rewritten as

$$Q = C_d A_0 \sqrt{\frac{2}{\rho}(P_1 - P_2)} \tag{2.7}$$

2.2.1.2 General flow equations

Consider the three land four-way spool valve as shown in Figure 2.3a. The four orifices are analogous to four arms of a Wheatstone bridge (Merritt 1969) as shown in Figure 2.3b. By using continuity equations, the flow rate through the load (Q_L) can be given by the following equations

$$Q_L = Q_1 - Q_4 \tag{2.8}$$

or

$$Q_L = Q_3 - Q_2 \tag{2.9}$$

Also, the pressure drops across the load is given in the following equation

$$P_L = P_1 - P_2 \tag{2.10}$$

When the difference between P_2 and P_l exists, the hydraulic cylinder extends or compresses. The flow rates of the fluid through the valve land orifices are described by the orifice equation (2.6). Therefore

$$Q_1 = C_d A_1 \sqrt{\frac{2}{\rho}(P_s - P_1)} \tag{2.11}$$

$$Q_2 = C_d A_2 \sqrt{\frac{2}{\rho}(P_s - P_2)} \tag{2.12}$$

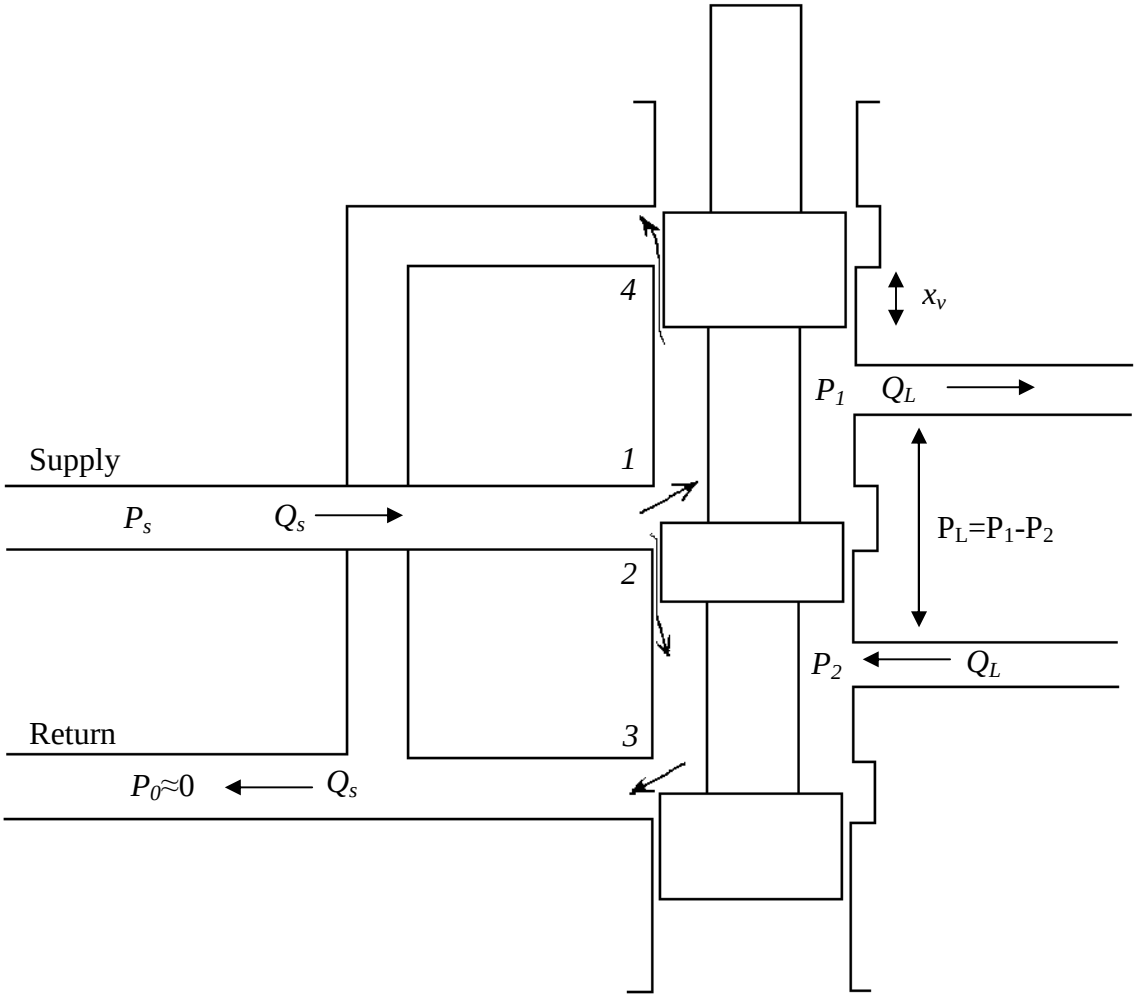
$$Q_3 = C_d A_3 \sqrt{\frac{2}{\rho}(P_2 - P_0)} \tag{2.13}$$

$$Q_4 = C_d A_4 \sqrt{\frac{2}{\rho}(P_1 - P_0)} \tag{2.14}$$

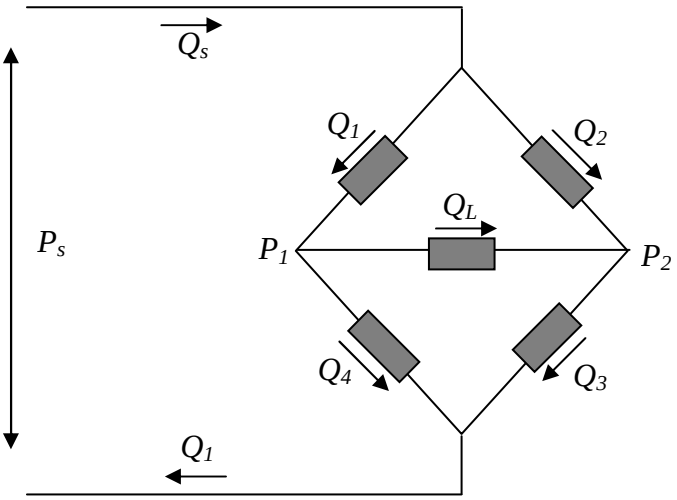
where P_0 is the pressure of the return line and it is very small. It has been neglected.

P_s is the pressure of the supply line; P_1 is the pressure of chamber one and P_2 is the pressure of chamber two.

$A_1, A_2, A_3$and A_4 are the cross-sectional area at 1,2,3 and 4 respectively. These areas are a function of the spool valve displacement x_v .



(a)



(b)

Figure 2.3 Three land four way spool valve

In general, the valving orifices are matched and symmetrical. Matched orifices require:

$$A_1(x_v) = A_3(x_v) \tag{2.15}$$

$$A_2(x_v) = A_4(x_v) \tag{2.16}$$

and symmetrical orifices require

$$A_1(x_v) = A_2(-x_v) \tag{2.17}$$

$$A_3(x_v) = A_4(-x_v) \tag{2.18}$$

Additionally, the flow rates in diagonally opposite arms of the Wheatstone bridge are equal, which means

$$Q_1 = Q_3 \tag{2.19}$$

$$Q_2 = Q_4 \tag{2.20}$$

From equations (2.11), (2.13), (2.15) and (2.19) the pressure of the supply line is governed by following equation

$$P_s = P_1 + P_2 \tag{2.21}$$

By solving Eq. (2.10) and Eq. (2.21) the following equations can be obtained

$$P_1 = \frac{P_s + P_L}{2} \tag{2.22}$$

$$P_2 = \frac{P_s - P_L}{2} \tag{2.23}$$

By assuming that the critical centre valve has an ideal geometry, which implies the edges of the orifice to be perfectly square with no rounding, thus the leakage flows at 2 and 4 (Q_2 and Q_4) when x_v is positive are zero while when x_v is negative, the leakage flows at 1 and 3 (Q_1 and Q_3) are zero. Therefore, by substituting Eq. (2.22) and Eq. (2.11) into Eq. (2.8) the equation of the load flow rate can be obtained

$$Q_L = C_d A_1 \sqrt{\frac{2}{\rho}\left(\frac{P_s - P_L}{2}\right)} \quad \text{for } x_v > 0 \tag{2.24}$$

By substituting Eq. (2.22) and Eq. (2.14) and ignoring P_0 since it is very small, the equation of Q_L can be written as

$$Q_L = -C_d A_1 \sqrt{\frac{2}{\rho}\left(\frac{P_s + P_L}{2}\right)} \quad \text{for } x_v < 0 \tag{2.25}$$

The Eq. (2.17) is applicable. Eq. (2.24) and (2.25) can be combined into a single equation

$$Q_L = C_d |A_1| \frac{x_v}{|x_v|} \sqrt{\frac{1}{\rho}(P_s - sgn(x_v) P_L)} \tag{2.26}$$

The sigmoid function $sgn(.)$ is defined as

$$sgn(x_v) = \begin{cases} 1 & for \ x_v > 0 \\ -1 & for \ x_v < 0 \end{cases}$$

The term $\left|\frac{A_1}{x_v}\right|$ is called area gradient ω, therefore, the relationship between spool valve displacement x_v, and the load flow Q_L, is given

$$Q_L = C_d \omega x_v \sqrt{\frac{1}{\rho}(P_s - sgn(x_v) P_L)} \tag{2.27}$$

This is the general equation for the pressure-flow curves of an ideal critical center valve with matched and symmetrical orifices.

2.2.2 Physical model of nonlinear valve-piston combination

As mentioned before, the electro-hydraulic actuator model consists of a servo valve, a hydraulic cylinder and piston. The valve-piston combination is shown in Figure 2.4.

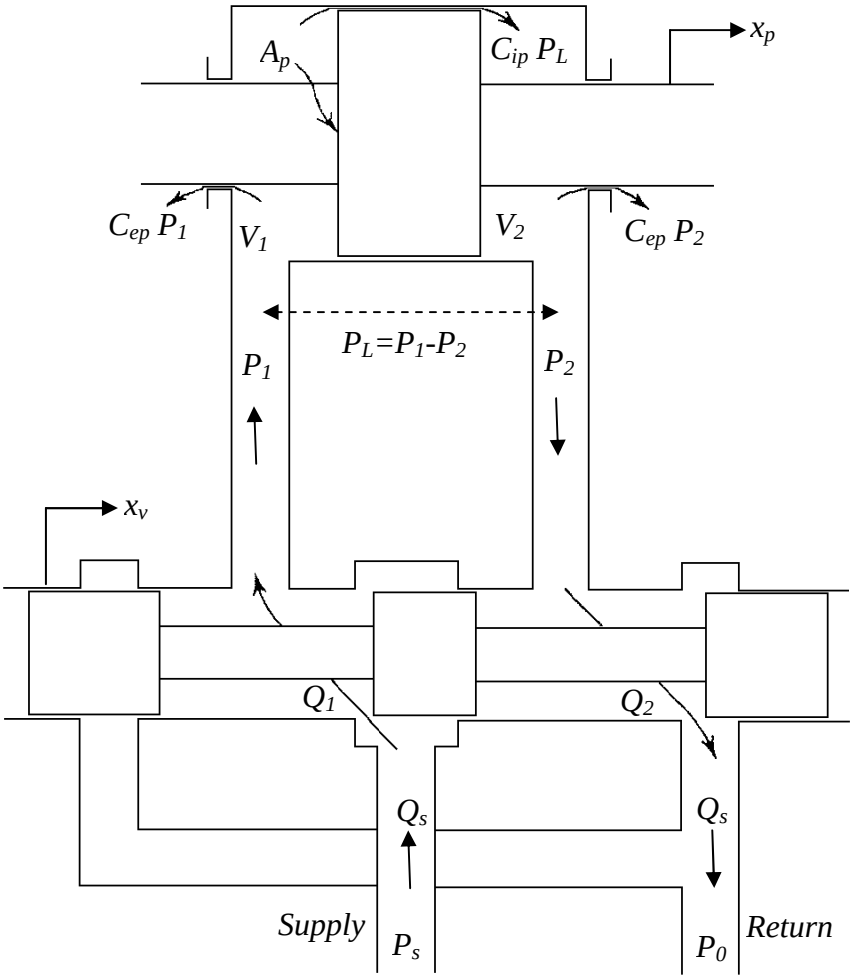


Figure 2.4 Valve-piston combinations

The spool valve orifices are assumed matched and symmetrical, according to that the continuity equation can be given as

$$\sum Q_{in} - \sum Q_{out} = \frac{dV}{dt} + \frac{V}{\beta_e}\frac{dP}{dt} \tag{2.28}$$

By applying Eq. (2.28) to each of the piston chambers yields

$$Q_1 - C_{ip}P_L - C_{ep}P_1 = \frac{dV_1}{dt} + \frac{V_1}{\beta_e}\frac{dP_1}{dt} \tag{2.29}$$

$$C_{ip}P_L - C_{ep}P_2 - Q_2 = \frac{dV_2}{dt} + \frac{V_2}{\beta_e}\frac{dP_2}{dt} \tag{2.30}$$

where V_1 volume of supply chamber; V_2 volume of return chamber; P_1 pressure of supply chamber; P_2 pressure of return chamber; C_{ip} internal or cross port leakage coefficient of piston; C_{ep} external leakage coefficient of piston and β_e effective bulk modulus of system.

The volumes of the piston chambers can be written as

$$V_1 = V_{01} + A_p x_p \tag{2.31}$$

$$V_2 = V_{02} - A_p x_p \tag{2.32}$$

where A_p cross-sectional area of the piston; x_p displacement of the piston inside the cylinder; V_{01} initial volume of supply chamber and V_{02} initial volume of return chamber.

The sum of the two volumes is constant and independent of the position of the piston inside the cylinder. Therefore, the total volume (V_t) of the fluid under compression in both chambers can be given by

$$V_t = V_1 + V_2 = V_{01} + V_{02} \tag{2.33}$$

The flow rate through the load (i.e. piston) can be governed by the following equation

$$Q_L = \frac{Q_1 + Q_2}{2} \tag{2.34}$$

Substituting Eq. (2.31) in (2.29) yields

$$Q_1 - C_{ip}P_L - C_{ep}P_1 = A_p\dot{x}_p + \frac{V_1}{\beta_e}\frac{dP_1}{dt} \tag{2.35}$$

Also, Substituting Eq. (2.32) in (2.30) yields

$$C_{ip}P_L - C_{ep}P_2 - Q_2 = \frac{V_2}{\beta_e}\frac{dP_2}{dt} - A_p\dot{x}_p \tag{2.36}$$

By substituting Eq. (2.35) and (2.36) in (2.34) yields

$$2Q_L = 2A_p\dot{x}_p + 2C_{ip}P_L + \frac{V_1}{\beta_e}\frac{dP_1}{dt} - \frac{V_2}{\beta_e}\frac{dP_2}{dt} + C_{ep}P_1 - C_{ep}P_2 \tag{2.37}$$

Substituting Eq. (2.10) in (2.37) yields

$$Q_L = A_p\dot{x}_p + C_{tp}P_L + \frac{V_t}{4\beta_e}\dot{P}_L \tag{2.38}$$

where $C_{tp} = C_{ip} + \frac{C_{ep}}{2}$ total leakage coefficient of piston.

If the following assumptions are made: $\sigma = \frac{4\beta_e}{V_t}$ and $\beta = \sigma C_{tp}$, Eq. (2.38) can be rewritten as

$$\dot{P}_L = -\beta P_L - \sigma A_p \dot{x}_p + \sigma Q_L \tag{2.39}$$

The dynamic equation for spool displacement of the servo valve (x_v) is controlled by the voltage signal input to the servovalve (u_m). The corresponding dynamic relation is approximated by a linear filter with time constant (τ)

$$\dot{x}_v = \frac{1}{\tau}(u_m - x_v) \tag{2.40}$$

By supplying the control input (u_m), the spool-valve is moved by x_v units which cause high pressure difference across the piston. This pressure difference multiplied by the piston cross-sectional area provides the hydraulic force F_h on the suspension system. Therefore, the spool-valve movement has to be properly controlled to produce and track the desired force.

The provided hydraulic force by the actuator is calculated by

$$F_h = A_p P_L \tag{2.41}$$

The relationship between control input signal (u_m) and the output force generated from actuator possess a nonlinear dynamics (Merritt 1969).

Based on the above, the nonlinear hydraulic actuator model is given by the following equations

$F_h = A_p P_L$;

$\dot{P}_L = -\beta P_L - \sigma A_p \dot{x}_p + \sigma Q_L$;

$Q_L = C_d \omega x_v \sqrt{\frac{1}{\rho}(P_s - sgn(x_v) P_L)}$;

$\dot{x}_v = \frac{1}{\tau}(u_m - x_v)$.

2.3 Nonlinear force characteristics of full vehicle nonlinear active suspension systems with hydraulic actuators

To investigate the problem of balancing riding comfort and road handling, the mathematical model of four-wheel nonlinear active suspension systems with hydraulic actuators should be introduced. As shown in Figure 2.5, the tyres simulated as springs with a constant stiffness k_i (i=1, 2, 3 and 4) and dampers with a constant damping c_i are connected in parallel. The road profile inputs u_i excite the tyres and the surrounding mechanical components that are attached to these tyres. The total mass of the components under the suspension system is called unsprung mass, labelled as m_i, while the mass of the components above the suspension is called sprung mass, *M*. The suspension system consists of a nonlinear spring with stiffness constant K_i and a nonlinear damper with a damping constant C_i, which are connected in parallel with the hydraulic actuator. The nonlinear forces F_i are applied between the sprung mass and unsprung masses which are generated from hydraulic actuators to minimize: the vertical motion (z_c) sensed by passengers when travelling on a rough road; and vehicle body motions that are made during sharp manoeuvres such as roll angle (α) and pitch angle (η). The nonlinear frictional forces F_{fi} due to rubbing of the piston seals with the cylinder wall inside the actuators are also taken into account to derive the accurate model.

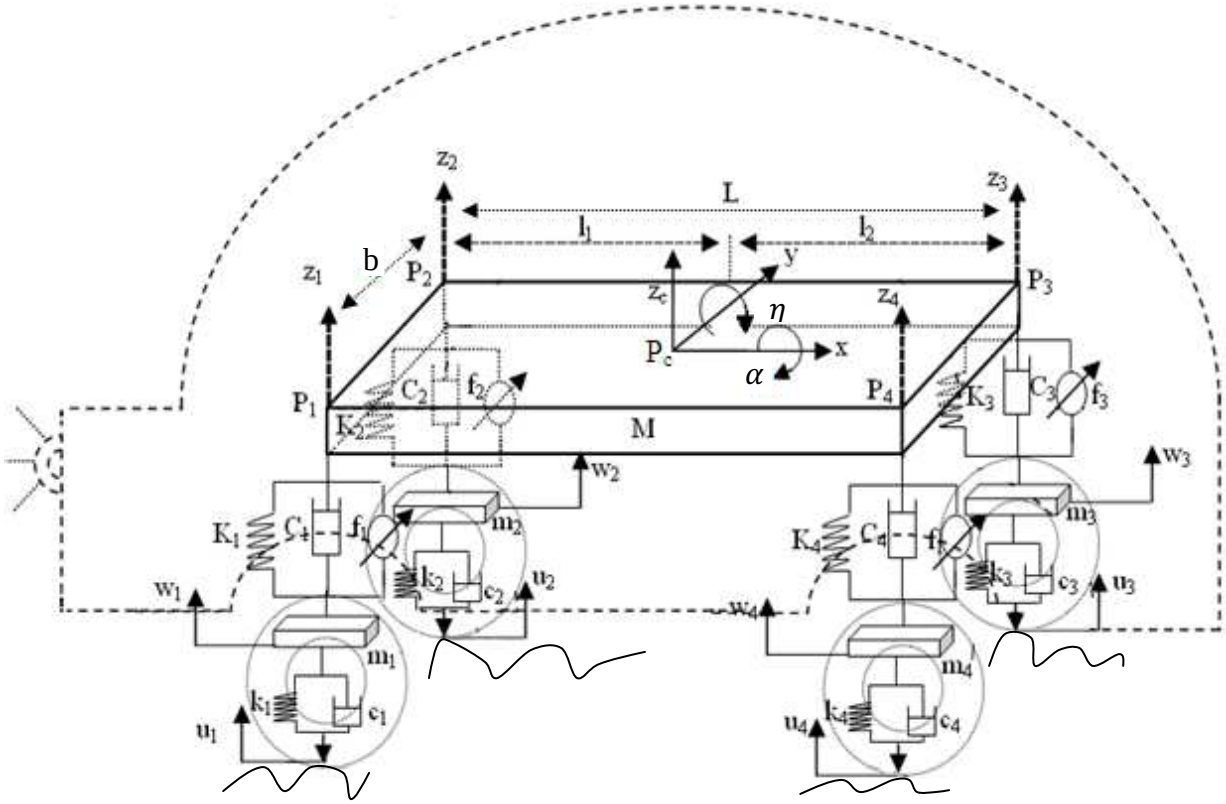


Figure 2.5 Full vehicle model

The plane equation will be used to derive the full model of nonlinear active suspension system which can be written as

$$\begin{vmatrix} x-x_1 & y-y_1 & z-z_1 \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_4-x_1 & y_4-y_1 & z_4-z_1 \end{vmatrix} = 0 \tag{2.42}$$

Any point at the sprung mass plane satisfies the above plane equation.

The coordinates of the points P_1, P_2, P_3, P_4 and P_c can be given as $(0,0,z_1)$, $(0,b,z_2)$, (L,b,z_3), $(L,0,z_4)$ and $\left(l_1,\frac{b}{2},z_c\right)$ respectively.

The points P_1, P_2 and P_4 are located on the sprung mass plane, so that these points satisfy the plane equation, as mention above. By substituting the coordinates of these points in Eq. (2.42) yields

$$z = \frac{(z_4-z_1)}{L}x + \frac{(z_2-z_1)}{b}y + z_1 \tag{2.43}$$

The vertical displacement at the centre of gravity (z_c) and at point P_3 (z_3) can be obtained by substituting the coordinates of P_3 and P_c in Eq. (2.43)

$$z_3 = -z_1 + z_2 + z_4 \tag{2.44}$$

$$z_c = az_1 + 0.5z_2 + \frac{l_1}{L}z_4 \tag{2.45}$$

where $a = 0.5 - \frac{l_1}{L}$.

The roll and pitch angles can be obtained from the following equations

$$\alpha = \frac{\partial z}{\partial y} = \frac{z_2 - z_1}{b} \tag{2.46}$$

$$\eta = \frac{\partial z}{\partial x} = \frac{z_4 - z_1}{L} \tag{2.47}$$

From equations (2.44), (2.45), (2.46) and (2.47) the vertical displacements z_1, z_2, z_3 and z_4 can be obtained with respect to z_c, α and η as follows

$$z_1 = z_c - 0.5b\alpha - l_1\eta \tag{2.48}$$

$$z_2 = z_c + 0.5b\alpha - l_1\eta \tag{2.49}$$

$$z_3 = z_c + 0.5b\alpha + l_2\eta \tag{2.50}$$

$$z_4 = z_c - 0.5b\alpha + l_2\eta \tag{2.51}$$

The differential equations of the full vehicle nonlinear active suspension system can be obtained from the Newton's third law of motion as follows

1. The motions of the sprung mass are governed by the following equations

- ***Vertical motion***

$$M\ddot{z}_c = -\sum_{i=1}^{4} F_{Ki} - \sum_{i=1}^{4} F_{Ci} + \sum_{i=1}^{4} F_i \tag{2.52}$$

where F_{Ki} and F_{Ci} are nonlinear forces of springs and dampers, respectively, which can be written as (Ando and Suzuki 1996)

$$F_{Ki} = K_i(z_i - w_i) + \zeta K_i(z_i - w_i)^3 \tag{2.53}$$

$$F_{Ci} = C_i(\dot{z}_i - \dot{w}_i) + \zeta C_i(\dot{z}_i - \dot{w}_i)^2 sgn(\dot{z}_i - \dot{w}_i) \quad (2.56)$$

where w_i vertical displacements of unsprung masses; ζ empirical operator.

The applied nonlinear forces F_i that are generated from the hydraulic actuators can be given by

$$F_i = F_{hi} - F_{fi} \quad (2.57)$$

where F_{hi} is the nonlinear hydraulic real forces, F_{fi} nonlinear frictional forces.

- ***Rolling motion***

$$J_x\ddot{\alpha} = (F_{K1} - F_{K2} - F_{K3} + F_{K4})\frac{b}{2} + (F_{C1} - F_{C2} - F_{C3} + F_{C4})\frac{b}{2} + (F_3 - F_1 + F_2 - F_4)\frac{b}{2} + T_x \quad (2.78)$$

where J_x is roll moments of inertia about x-axis; b is distance between the front wheels (or rear wheels) and T_x is cornering torque.

- ***Pitch motion***

$$J_y\ddot{\eta} = (F_{k3} + F_{k4})l_2 - (F_{K1} + F_{K2})l_1 + (F_{C3} + F_{C4})l_2 - (F_{C1} + F_{C2})l_1 + (F_1 + F_2)l_1 - (F_3 + F_4)l_2 + T_y \quad (2.79)$$

where J_y is the pitch moments of inertia about y-axis; l_1 is the distance between the centre of front wheel axle and centre of gravity of the vehicle; l_2 is the distance between the centre of gravity of the vehicle and the centre of rear wheel axle and T_x is the braking torque.

2. The motion of the unsprung masses are governed by the following equation

$$m_i\ddot{w}_i = -k_i(w_i - u_i) - c_i(\dot{w}_i - \dot{u}_i) + F_{Ki} + F_{Ci} - F_i \quad (2.80)$$

where u_i is the road profile input.

The forces generated by the hydraulic actuators can be written as

$$F_{hi} = A_p P_{Li} \quad (2.81)$$

Using the equation for the flow through an orifice, the relationship between spool valve displacement x_{vi} and the load flow Q_{Li} is given by

$$Q_{Li} = C_d \omega x_{vi} \sqrt{\frac{1}{\rho}(P_{si} - sgn(x_{vi})P_{Li})} \tag{2.82}$$

The continuity equation gives the relationship between the pressure across the actuator's piston P_{Li} and total load flow through the actuator Q_{Li} as

$$\dot{P}_{Li} = -\beta P_{Li} - \sigma A_p \dot{x}_{pi} + \sigma Q_{Li} \tag{2.83}$$

where x_{pi} is the difference between the vertical displacement of i^{th} corner of sprung mass z_i and the vertical displacement of corresponding i^{th} unsprung mass w_i, i.e. $x_{pi} = z_i - w_i$.

By substituting Eq. (2.82) in (2.83), the relationship between the load pressure and spool valve displacement can be obtained as follows

$$\dot{P}_{Li} = -\beta P_{Li} - \sigma A_p \dot{x}_{pi} + \sigma C_d \omega x_{vi} \sqrt{\frac{1}{\rho}(P_{si} - sgn(x_{vi})P_{Li})} \tag{2.84}$$

The relationship between the spool valve displacement and output of the controller is described as follows

$$\dot{x}_{vi} = \frac{1}{\tau}(u_{mi} - x_{vi}) \tag{2.85}$$

The actuator friction represents the friction associated with mechanical surfaces rubbing together, bearing friction, viscous friction, and so on.

Frictional forces are modeled with an approximation of Signum function (Rajamani and Hedrick 1995) as illustrated in Figure (2.6). From Figure (2.6) the mathematical model of friction forces can be written as

$$F_{fi} = \begin{cases} \mu \, sgn(\dot{x}_{pi}) & for \; |\dot{x}_{pi}| \geq 0.01 \\ \mu \, sin\left(\frac{\pi \dot{x}_{pi}}{0.02}\right) & for \; |\dot{x}_{pi}| < 0.01 \end{cases} \tag{2.86}$$

where μ is an empirical operator.

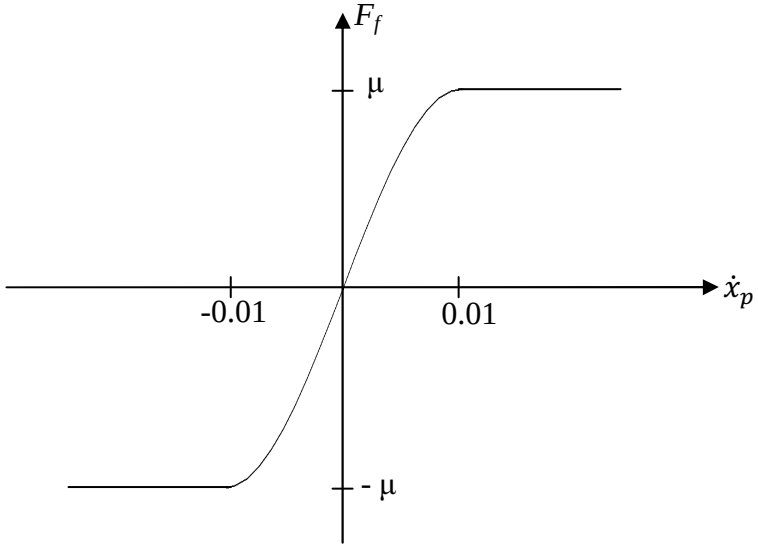


Figure 2.6 Friction model

2.4 Mathematical equations of full vehicle nonlinear active suspension systems with hydraulic actuators in state space form

The heave motion (vertical motion) of sprung mass (Eq. (2.52)) can be rewritten as follows

$$M\ddot{z}_c = -(C_1 + C_2 + C_3 + C_4)\dot{z}_c + 0.5b(C_1 - C_2 - C_3 + C_4)\dot{\alpha} + \big((C_1 + C_2)l_1 - C3+C4l2\eta - \zeta C1z1 - w12sgnz1 - w1 - \zeta C2z2 - w22sgnz2 - w2 - \zeta C3z3 - w32sgnz3 - w3 - \zeta C_4(\dot{z}_4 - \dot{w}_4)^2 sgn(\dot{z}_4 - \dot{w}_4) + C_1\dot{w}_1 + C_2\dot{w}_2 + C_3\dot{w}_3 + C_4\dot{w}_4 - (K_1 + K_2 + K_3 + K_4)z_c + 0.5b(K_1 - K_2 - K_3 + K_4)\alpha + \big((K_1 + K_2)l_1 - (K_3 + K_4)l_2\big)\eta - \zeta K_1(z_1 - w_1)^3 - \zeta K_2(z_2 - w_2)^3 - \zeta K_3(z_3 - w_3)^3 - \zeta K_4(z_4 - w_4)^3 + K_1w_1 + K_2w_2 + K_3w_3 + K_4w_4 + A_P(P_{L1} + P_{L2} + P_{L3} + P_{L4}) - F_{f1} - F_{f2} - F_{f3} - F_{f4} \quad (2.87)$$

Eq. (2.78) of rolling motion can be rewritten as follows

$$\frac{2J_x}{b}\ddot{\alpha} = (C_1 - C_2 - C_3 + C_4)\dot{z}_c - 0.5b(C_1 + C_2 + C_3 + C_4)\dot{\alpha} + \big((C_2 - C_1)l_1 + (C_4 - C_3)l_2\big)\dot{\eta} + \zeta C_1(\dot{z}_1 - \dot{w}_1)^2 sgn(\dot{z}_1 - \dot{w}_1) - \zeta C_2(\dot{z}_2 - \dot{w}_2)^2 sgn(\dot{z}_2 - \dot{w}_2) - \zeta C_3(\dot{z}_3 -$$

$\dot{w}_3)^2 sgn(\dot{z}_3 - \dot{w}_3) + \zeta C_4(\dot{z}_4 - \dot{w}_4)^2 sgn(\dot{z}_4 - \dot{w}_4) - C_1\dot{w}_1 + C_2\dot{w}_2 + C_3\dot{w}_3 - C_4\dot{w}_4 +$

$(K_1 - K_2 - K_3 + K_4)z_c - 0.5b(K_1 + K_2 + K_3 + K_4)\alpha + \big((K_2 - K_1)l_1 + (K_4 - K_3)l_2\big)\eta +$

$\zeta K_1(z_1 - w_1)^3 - \zeta K_2(z_2 - w_2)^3 - \zeta K_3(z_3 - w_3)^3 + \zeta K_4(z_4 - w_4)^3 - K_1w_1 + K_2w_2 +$

$K_3w_3 - K_4w_4 + A_p(P_{L3} - P_{L1} + P_{L2} - P_{L4}) + F_{f1} - F_{f2} - F_{f3} + F_{f4} + \frac{2}{b}T_x$ (2.88)

The pitching motion of sprung mass (Eq. (2.79)) can be rewritten as

$J_y\ddot{\eta} = \big((C_3 + C_4)l_2 - (C_1 + C_2)l_1\big)\dot{z}_c + 0.5b\big((C_1 - C_2)l_1 + (C_3 - C_4)l_2\big)\dot{\alpha} + \big((C_1 +$

$C2l12+C3+C4l22\eta+l1C1w1+l1C2w2-l2C3w3-l2C4w4-\zeta l1C1z1-w12sgnz1-w1-\zeta l$

$1C2z2-w22sgnz2-w2+\zeta l2C3z3-w32sgnz3-w3+$

$\zeta l_2C_4(\dot{z}_4 - \dot{w}_4)^2 sgn(\dot{z}_4 - \dot{w}_4) + \big((K_3 + K_4)l_2 - (K_1 + K_2)l_1\big)z_c + 0.5b\big((K_1 - K_2)l_1 +$

$K3-K4l2\alpha+K1+K2l12+K3+K4l22\eta+l1K1w1+l1K2w2-l2K3w3-l2K4w4-\zeta l1K1z1$

$-w13-\zeta l1K2z2-w23+\zeta l2K3z3-w33+\zeta l2K4z4-w43+Apl1PL1+Apl1PL2-Apl2PL3$

$-Apl2PL4-l1Ff1-l1Ff2+l2Ff3+l2Ff4+Ty$ (2.89)

The heave motions equation of unsprung masses can be rewritten as

$m_1\ddot{w}_1 = C_1\dot{z}_c - 0.5bC_1\dot{\alpha} - C_1l_1\dot{\eta} + \zeta C_1(\dot{z}_1 - \dot{w}_1)^2 sgn(\dot{z}_1 - \dot{w}_1) - (C_1 + c_1)\dot{w}_1 + c_1\dot{u}_1 +$

$K_1z_c - 0.5bK_1\alpha - K_1l_1\eta + \zeta K_1(z_1 - w_1)^3 - (K_1+k_1)w_1 + k_1u_1 - A_pP_{L1} + F_{f1}$ (2.90)

$m_2\ddot{w}_2 = C_2\dot{z}_c + 0.5bC_2\dot{\alpha} - C_2l_1\dot{\eta} + \zeta C_2(\dot{z}_2 - \dot{w}_2)^2 sgn(\dot{z}_2 - \dot{w}_2) - (C_2 + c_2)\dot{w}_2 + c_2\dot{u}_2 +$

$K_2z_c + 0.5bK_2\alpha - K_2l_1\eta + \zeta K_2(z_2 - w_2)^3 - (K_2+k_2)w_2 + k_2u_2 - A_pP_{L2} + F_{f2}$ (2.91)

$m_3\ddot{w}_3 = C_3\dot{z}_c + 0.5bC_3\dot{\alpha} + C_3l_2\dot{\eta} + \zeta C_3(\dot{z}_3 - \dot{w}_3)^2 sgn(\dot{z}_3 - \dot{w}_3) - (C_3 + c_3)\dot{w}_3 + c_3\dot{u}_3 +$

$K_3z_c + 0.5bK_3\alpha + K_3l_2\eta + \zeta K_3(z_3 - w_3)^3 - (K_3 + k_3)w_3 + k_3u_3 - A_pP_{L3} + F_{f3}$ (2.92)

$m_4\ddot{w}_4 = C_4\dot{z}_c - 0.5bC_4\dot{\alpha} + C_4l_2\dot{\eta} + \zeta C_4(\dot{z}_4 - \dot{w}_4)^2 sgn(\dot{z}_4 - \dot{w}_4) - (C_4 + c_4)\dot{w}_4 + c_4\dot{u}_4 +$

$K_4z_c - 0.5bK_4\alpha + K_4l_2\eta + \zeta K_4(z_4 - w_4)^3 - (K_4 + k_4)w_4 + k_4u_4 - A_pP_{L4} + F_{f4}$ (2.93)

From Eq. (2.84), the hydraulic actuator model for each suspension can be written as

$$\dot{P}_{L1} = -\beta P_{L1} - \sigma A_p \dot{z}_c + 0.5b\sigma A_p \dot{\alpha} + l_1 \sigma A_p \dot{\eta} + \sigma A_p \dot{w}_1 + \sigma C_d \omega x_{v1} \sqrt{\frac{1}{\rho}(P_{s1} - sgn(x_{v1}) P_{L1})} \quad (2.94)$$

$$\dot{P}_{L2} = -\beta P_{L2} - \sigma A_p \dot{z}_c - 0.5b\sigma A_p \dot{\alpha} + l_1 \sigma A_p \dot{\eta} + \sigma A_p \dot{w}_2 + \sigma C_d \omega x_{v2} \sqrt{\frac{1}{\rho}(P_{s2} - sgn(x_{v2}) P_{L2})} \quad (2.95)$$

$$\dot{P}_{L3} = -\beta P_{L3} - \sigma A_p \dot{z}_c - 0.5b\sigma A_p \dot{\alpha} - l_2 \sigma A_p \dot{\eta} + \sigma A_p \dot{w}_3 + \sigma C_d \omega x_{v3} \sqrt{\frac{1}{\rho}(P_{s3} - sgn(x_{v3}) P_{L3})} \quad (2.96)$$

$$\dot{P}_{L4} = -\beta P_{L4} - \sigma A_p \dot{z}_c + 0.5b\sigma A_p \dot{\alpha} - l_2 \sigma A_p \dot{\eta} + \sigma A_p \dot{w}_4 + \sigma C_d \omega x_{v4} \sqrt{\frac{1}{\rho}(P_{s4} - sgn(x_{v4}) P_{L4})} \quad (2.97)$$

$$\dot{x}_{v1} = \frac{1}{\tau}(u_{m1} - x_{v1}) \quad (2.98)$$

$$\dot{x}_{v2} = \frac{1}{\tau}(u_{m2} - x_{v2}) \quad (2.99)$$

$$\dot{x}_{v3} = \frac{1}{\tau}(u_{m3} - x_{v3}) \quad (2.100)$$

$$\dot{x}_{v4} = \frac{1}{\tau}(u_{m4} - x_{v4}) \quad (2.101)$$

For simplification of the state space form for full vehicle nonlinear active suspension system with hydraulic actuator the following assumptions can be assumed

$s_1 = (z_1 - w_1)^3$; $s_2 = (z_2 - w_2)^3$;

$s_3 = (z_3 - w_3)^3$; $s_4 = (z_4 - w_4)^3$;

$s_5 = (\dot{z}_1 - \dot{w}_1)^2 sgn(\dot{z}_1 - \dot{w}_1)$; $s_6 = (\dot{z}_2 - \dot{w}_2)^2 sgn(\dot{z}_2 - \dot{w}_2)$;

$s_7 = (\dot{z}_3 - \dot{w}_3)^2 sgn(\dot{z}_3 - \dot{w}_3)$; $s_8 = (\dot{z}_4 - \dot{w}_4)^2 sgn(\dot{z}_4 - \dot{w}_4)$;

$s_9 = x_{v1}\sqrt{(P_{s1} - sgn(x_{v1}) P_{L1})}$; $s_{10} = x_{v2}\sqrt{(P_{s2} - sgn(x_{v2}) P_{L2})}$;

$s_{11} = x_{v3}\sqrt{(P_{s3} - sgn(x_{v3}) P_{L3})}$; $s_{12} = x_{v4}\sqrt{(P_{s4} - sgn(x_{v4}) P_{L4})}$;

$$s_{13} = \begin{cases} sgn(\dot{z}_1 - \dot{w}_1) & for\ |\dot{z}_1 - \dot{w}_1| \geq 0.01 \\ sin\left(\frac{\pi(\dot{z}_1 - \dot{w}_1)}{0.02}\right) & for\ |\dot{z}_1 - \dot{w}_1| < 0.01 \end{cases} \quad s_{14} = \begin{cases} sgn(\dot{z}_2 - \dot{w}_2) & for\ |\dot{z}_2 - \dot{w}_2| \geq 0.01 \\ sin\left(\frac{\pi(\dot{z}_2 - \dot{w}_2)}{0.02}\right) & for\ |\dot{z}_2 - \dot{w}_2| < 0.01 \end{cases}$$

$$s_{15} = \begin{cases} sgn(\dot{z}_3 - \dot{w}_3) & for\ |\dot{z}_3 - \dot{w}_3| \geq 0.01 \\ sin\left(\frac{\pi(\dot{z}_3 - \dot{w}_3)}{0.02}\right) & for\ |\dot{z}_3 - \dot{w}_3| < 0.01 \end{cases} \quad s_{16} = \begin{cases} sgn(\dot{z}_4 - \dot{w}_4) & for\ |\dot{z}_4 - \dot{w}_4| \geq 0.01 \\ sin\left(\frac{\pi(\dot{z}_4 - \dot{w}_4)}{0.02}\right) & for\ |\dot{z}_4 - \dot{w}_4| < 0.01 \end{cases}$$

Substituting these assumptions into equations (2.87)-(2.97) leads to

$$\ddot{z}_c = -\frac{(C_1+C_2+C_3+C_4)}{M}\dot{z}_c + \frac{0.5b(C_1-C_2-C_3+C_4)}{M}\dot{\alpha} + \frac{\left((C_1+C_2)l_1-(C_3+C_4)l_2\right)}{M}\dot{\eta} + \frac{C_1}{M}\dot{w}_1 + \frac{C_2}{M}\dot{w}_2 +$$

$$\frac{C_3}{M}\dot{w}_3 + \frac{C_4}{M}\dot{w}_4 - \frac{(K_1+K_2+K_3+K_4)}{M}z_c + \frac{0.5b(K_1-K_2-K_3+K_4)}{M}\alpha + \frac{\left((K_1+K_2)l_1-(K_3+K_4)l_2\right)}{M}\eta + \frac{K_1}{M}w_1 +$$

$$\frac{K_2}{M}w_2 + \frac{K_3}{M}w_3 + \frac{K_4}{M}w_4 + \frac{A_P}{M}(P_{L1}+P_{L2}+P_{L3}+P_{L4}) - \frac{\zeta K_1}{M}s_1 - \frac{\zeta K_2}{M}s_2 - \frac{\zeta K_3}{M}s_3 - \frac{\zeta K_4}{M}s_4 -$$

$$\frac{\zeta C_1}{M}s_5 - \frac{\zeta C_2}{M}s_6 - \frac{\zeta C_3}{M}s_7 - \frac{\zeta C_4}{M}s_8 - \frac{\mu}{M}s_{13} - \frac{\mu}{M}s_{14} - \frac{\mu}{M}s_{15} - \frac{\mu}{M}s_{16} \qquad (2.102)$$

$$\ddot{\alpha} = \frac{b(C_1-C_2-C_3+C_4)}{2J_x}\dot{z}_c - \frac{b^2(C_1+C_2+C_3+C_4)}{4J_x}\dot{\alpha} + \frac{b\left((C_2-C_1)l_1+(C_4-C_3)l_2\right)}{2J_x}\dot{\eta} - \frac{bC_1}{2J_x}\dot{w}_1 + \frac{bC_2}{2J_x}\dot{w}_2 +$$

$$\frac{bC_3}{2J_x}\dot{w}_3 - \frac{bC_4}{2J_x}\dot{w}_4 + \frac{b(K_1-K_2-K_3+K_4)}{2J_x}z_c - \frac{b^2(K_1+K_2+K_3+K_4)}{4J_x}\alpha + \frac{b\left((K_2-K_1)l_1+(K_4-K_3)l_2\right)}{2J_x}\eta - \frac{bK_1}{2J_x}w_1 +$$

$$\frac{bK_2}{2J_x}w_2 + \frac{bK_3}{2J_x}w_3 - \frac{bK_4}{2J_x}w_4 + \frac{bA_p}{2J_x}(P_{L3}-P_{L1}+P_{L2}-P_{L4}) + \frac{\zeta bK_1}{2J_x}s_1 - \frac{\zeta bK_2}{2J_x}s_2 - \frac{\zeta bK_3}{2J_x}s_3 +$$

$$\frac{\zeta bK_4}{2J_x}s_4 + \frac{\zeta bC_1}{2J_x}s_5 - \frac{\zeta bC_2}{2J_x}s_6 - \frac{\zeta bC_3}{2J_x}s_7 + \frac{\zeta bC_4}{2J_x}s_8 + \frac{b\mu}{2J_x}s_{13} - \frac{b\mu}{2J_x}s_{14} - \frac{b\mu}{2J_x}s_{15} + \frac{b\mu}{2J_x}s_{16} + \frac{1}{J_x}T_x$$

(2.103)

$$\ddot{\eta} = \frac{\left((C_3+C_4)l_2-(C_1+C_2)l_1\right)}{J_y}\dot{z}_c + \frac{0.5b\left((C_1-C_2)l_1+(C_3-C_4)l_2\right)}{J_y}\dot{\alpha} + \frac{\left((C_1+C_2)l_1^2+(C_3+C_4)l_2^2\right)}{J_y}\dot{\eta} + \frac{l_1C_1}{J_y}\dot{w}_1 +$$

$$\frac{l_1C_2}{J_y}\dot{w}_2 - \frac{l_2C_3}{J_y}\dot{w}_3 - \frac{l_2C_4}{J_y}\dot{w}_4 + \frac{\left((K_3+K_4)l_2-(K_1+K_2)l_1\right)}{J_y}z_c + \frac{0.5b\left((K_1-K_2)l_1+(K_3-K_4)l_2\right)}{J_y}\alpha +$$

$$\frac{\left((K_1+K_2)l_1^2+(K_3+K_4)l_2^2\right)}{J_y}\eta + \frac{l_1K_1}{J_y}w_1 + \frac{l_1K_2}{J_y}w_2 - \frac{l_2K_3}{J_y}w_3 - \frac{l_2K_4}{J_y}w_4 + \frac{A_pl_1}{J_y}P_{L1} + \frac{A_pl_1}{J_y}P_{L2} -$$

$$\frac{A_pl_2}{J_y}P_{L3} - \frac{A_pl_2}{J_y}P_{L4} - \frac{\zeta l_1K_1}{J_y}s_1 - \frac{\zeta l_1K_2}{J_y}s_2 + \frac{\zeta l_2K_3}{J_y}s_3 + \frac{\zeta l_2K_4}{J_y}s_4 - \frac{\zeta l_1C_1}{J_y}s_5 - \frac{\zeta l_1C_2}{J_y}s_6 + \frac{\zeta l_2C_3}{J_y}s_7 +$$

$$\frac{\zeta l_2C_4}{J_y}s_8 - \frac{l_1\mu}{J_y}s_{13} - \frac{l_1\mu}{J_y}s_{14} + \frac{l_2\mu}{J_y}s_{15} + \frac{l_2\mu}{J_y}s_{16} + \frac{1}{J_y}T_y \qquad (2.104)$$

$$\ddot{w}_1 = \frac{C_1}{m_1}\dot{z}_c - \frac{0.5bC_1}{m_1}\dot{\alpha} - \frac{C_1 l_1}{m_1}\dot{\eta} - \frac{(C_1+c_1)}{m_1}\dot{w}_1 + \frac{K_1}{m_1}z_c - \frac{0.5bK_1}{m_1}\alpha - \frac{K_1 l_1}{m_1}\eta - \frac{(K_1+k_1)}{m_1}w_1 -$$

$$\frac{A_p}{m_1}P_{L1} + \frac{\zeta K_1}{m_1}s_1 + \frac{\zeta C_1}{m_1}s_5 + \frac{\mu}{m_1}s_{13} + \frac{c_1}{m_1}\dot{u}_1 + \frac{k_1}{m_1}u_1 \quad (2.105)$$

$$\ddot{w}_2 = \frac{C_2}{m_2}\dot{z}_c + \frac{0.5bC_2}{m_2}\dot{\alpha} - \frac{C_2 l_1}{m_2}\dot{\eta} - \frac{(C_2+c_2)}{m_2}\dot{w}_2 + \frac{K_2}{m_2}z_c + \frac{0.5bK_2}{m_2}\alpha - \frac{K_2 l_1}{m_2}\eta - \frac{(K_2+k_2)}{m_2}w_2 -$$

$$\frac{A_p}{m_2}P_{L2} + \frac{\zeta K_2}{m_2}s_2 + \frac{\zeta C_2}{m_2}s_6 + \frac{\mu}{m_2}s_{14} + \frac{c_2}{m_2}\dot{u}_2 + \frac{k_2}{m_2}u_2 \quad (2.106)$$

$$\ddot{w}_3 = \frac{C_3}{m_3}\dot{z}_c + \frac{0.5bC_3}{m_3}\dot{\alpha} - \frac{C_3 l_2}{m_3}\dot{\eta} - \frac{(C_3+c_3)}{m_3}\dot{w}_3 + \frac{K_3}{m_3}z_c + \frac{0.5bK_3}{m_3}\alpha - \frac{K_3 l_2}{m_3}\eta - \frac{(K_3+k_3)}{m_3}w_3 -$$

$$\frac{A_p}{m_3}P_{L3} + \frac{\zeta K_3}{m_3}s_3 + \frac{\zeta C_3}{m_3}s_7 + \frac{\mu}{m_3}s_{15} + \frac{c_3}{m_3}\dot{u}_3 + \frac{k_3}{m_3}u_3 \quad (2.107)$$

$$\ddot{w}_4 = \frac{C_4}{m_4}\dot{z}_c - \frac{0.5bC_4}{m_4}\dot{\alpha} + \frac{C_4 l_2}{m_4}\dot{\eta} - \frac{(C_4+c_4)}{m_4}\dot{w}_4 + \frac{K_4}{m_4}z_c - \frac{0.5bK_4}{m_4}\alpha + \frac{K_4 l_2}{m_4}\eta - \frac{(K_4+k_4)}{m_4}w_4 -$$

$$\frac{A_p}{m_4}P_{L4} + \frac{\zeta K_4}{m_4}s_4 + \frac{\zeta C_4}{m_4}s_8 + \frac{\mu}{m_4}s_{16} + \frac{c_4}{m_4}\dot{u}_4 + \frac{k_4}{m_4}u_4 \quad (2.108)$$

$$\dot{P}_{L1} = -\sigma A_p \dot{z}_c + 0.5b\sigma A_p \dot{\alpha} + l_1 \sigma A_p \dot{\eta} + \sigma A_p \dot{w}_1 - \beta P_{L1} + \gamma s_9 \quad (2.109)$$

$$\dot{P}_{L2} = -\sigma A_p \dot{z}_c - 0.5b\sigma A_p \dot{\alpha} + l_1 \sigma A_p \dot{\eta} + \sigma A_p \dot{w}_2 - \beta P_{L2} + \gamma s_{10} \quad (2.110)$$

$$\dot{P}_{L3} = -\sigma A_p \dot{z}_c - 0.5b\sigma A_p \dot{\alpha} - l_2 \sigma A_p \dot{\eta} + \sigma A_p \dot{w}_3 - \beta P_{L3} + \gamma s_{11} \quad (2.111)$$

$$\dot{P}_{L4} = -\sigma A_p \dot{z}_c + 0.5b\sigma A_p \dot{\alpha} - l_2 \sigma A_p \dot{\eta} + \sigma A_p \dot{w}_4 - \beta P_{L4} + \gamma s_{12} \quad (2.112)$$

where $\gamma = \alpha C_d \omega \sqrt{\frac{1}{\rho}}$

The state of the system can be given by

$$X_1 = [z_c \quad \alpha \quad \eta \quad w_1 \quad w_2 \quad w_3 \quad w_4 \quad P_{L1} \quad P_{L2} \quad P_{L3} \quad P_{L4} \quad x_{v1} \quad x_{v2} \quad x_{v3} \quad x_{v4}]$$

$$X_2 = [\dot{z}_c \quad \dot{\alpha} \quad \dot{\eta} \quad \dot{w}_1 \quad \dot{w}_2 \quad \dot{w}_3 \quad \dot{w}_4]$$

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

$$S = [s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5 \quad s_6 \quad s_7 \quad s_8 \quad s_9 \quad s_{10} \quad s_{11} \quad s_{12} \quad s_{13} \quad s_{14} \quad s_{15} \quad s_{16}]^T$$

$$U_1 = [T_x \quad T_y \quad u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_{m1} \quad u_{m2} \quad u_{m3} \quad u_{m4}]^T$$

$$U_2 = [\dot{u}_1 \quad \dot{u}_2 \quad \dot{u}_3 \quad \dot{u}_4]^T$$

$$U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$$

$$y = [z_1 \quad z_2 \quad z_3 \quad z_4 \quad w_1 \quad w_2 \quad w_3 \quad w_4 \quad z_c \quad \alpha \quad \eta]^T$$

The input state space equation can be written as

$$\dot{X} = B_1 X + B_2 S + B_3 U$$

The output equation can be written as

$$Y = C_0 X + DU$$

where B_1 is called the state matrix, B_2 the nonlinear matrix, B_3 the input matrix, C_0 the output matrix and D the direct transmission matrix.

The matrix B_1 takes big size when trying to write it on full page, therefore, the following assumptions are introduced

$$K_{1z_c} = \frac{(K_1+K_2+K_3+K_4)}{M};\ K_{1\alpha} = \frac{0.5b(K_2+K_3-K_1-K_4)}{M};\ K_{1\eta} = \frac{\left(l_2(K_3+K_4)-l_1(K_1+K_2)\right)}{M}$$

$$C_{1z_c} = \frac{(C_1+C_2+C_3+C_4)}{M};\ C_{1\alpha} = \frac{0.5b(C_2+C_3-C_1-C_4)}{M};\ C_{1\eta} = \frac{\left(l_2(C_3+C_4)-l_1(C_1+C_2)\right)}{M}$$

$$K_{2z_c} = \frac{b(K_2-K_1+K_3-K_4)}{2J_x};\ K_{2\alpha} = \frac{b^2(K_1+K_2+K_3+K_4)}{4J_x};\ K_{2\eta} = \frac{b\left(l_2(K_3-K_4)-l_1(K_2-K_1)\right)}{2J_x}$$

$$C_{2z_c} = \frac{b(C_2-C_1+C_3-C_4)}{2J_x};\ C_{2\alpha} = \frac{b^2(C_1+C_2+C_3+C_4)}{4J_x};\ C_{2\eta} = \frac{b\left(l_2(C_3-C_4)-l_1(C_2-C_1)\right)}{2J_x}$$

$$K_{3z_c} = \frac{\left(l_1(K_1+K_2)-l_2(K_3+K_4)\right)}{J_y}; K_{3\alpha} = \frac{0.5b\left(l_1(K_2-K_1)+l_2(K_4-K_3)\right)}{J_y}; K_{3\eta} = \frac{-\left(l_1^2(K_1+K_2)+l_2^2(K_3+K_4)\right)}{J_y}$$

$$C_{3z_c} = \frac{\left(l_1(C_1+C_2)-l_2(C_3+C_4)\right)}{J_y}; C_{3\alpha} = \frac{0.5b\left(l_1(C_2-C_1)+l_2(C_4-C_3)\right)}{J_y}; C_{3\eta} = \frac{-\left(l_1^2(C_1+C_2)+l_2^2(C_3+C_4)\right)}{J_y}$$

$$B_1 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & $-\beta$ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & $-\sigma A_P$ & $0.5b\sigma A_p$ & $l_1\sigma A_P$ & σA_p & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & $-\beta$ & 0 & 0 & 0 & 0 & 0 & 0 & $-\sigma A_P$ & $-0.5b\sigma A_P$ & $l_1\sigma A_P$ & 0 & σA_P & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & $-\beta$ & 0 & 0 & 0 & 0 & 0 & $-\sigma A_P$ & $-0.5b\sigma A_P$ & $-l_2\sigma A_P$ & 0 & 0 & σA_P & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & $-\beta$ & 0 & 0 & 0 & 0 & $-\sigma A_P$ & $0.5b\sigma A_p$ & $-l_2\sigma A_P$ & 0 & 0 & 0 & σA_P \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & $-\frac{1}{\tau}$ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & $-\frac{1}{\tau}$ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & $-\frac{1}{\tau}$ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & $-\frac{1}{\tau}$ & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
$-K_{1z_c}$ & $-K_{1\alpha}$ & $-K_{1\eta}$ & $\frac{K_1}{M}$ & $\frac{K_2}{M}$ & $\frac{K_3}{M}$ & $\frac{K_4}{M}$ & $\frac{A_P}{M}$ & $\frac{A_P}{M}$ & $\frac{A_P}{M}$ & $\frac{A_P}{M}$ & 0 & 0 & 0 & 0 & $-C_{1z_c}$ & $-C_{1\alpha}$ & $-C_{1\eta}$ & $\frac{C_1}{M}$ & $\frac{C_2}{M}$ & $\frac{C_3}{M}$ & $\frac{C_4}{M}$ \\
$-K_{2z_c}$ & $-K_{2\alpha}$ & $-K_{2\eta}$ & $\frac{-K_1b}{2J_x}$ & $\frac{K_2b}{2J_x}$ & $\frac{K_3b}{2J_x}$ & $\frac{-K_4b}{2J_x}$ & $\frac{-bA_p}{2J_x}$ & $\frac{bA_p}{2J_x}$ & $\frac{bA_p}{2J_x}$ & $\frac{-bA_p}{2J_x}$ & 0 & 0 & 0 & 0 & $-C_{2z_c}$ & $-C_{2\alpha}$ & $-C_{2\eta}$ & $\frac{-C_1b}{2J_x}$ & $\frac{C_2b}{2J_x}$ & $\frac{C_3b}{2J_x}$ & $\frac{-C_4b}{2J_x}$ \\
$-K_{3z_c}$ & $-K_{3\alpha}$ & $-K_{3\eta}$ & $\frac{K_1l_1}{J_y}$ & $\frac{K_2l_1}{J_y}$ & $\frac{-K_3l_2}{J_y}$ & $\frac{-K_4l_2}{J_y}$ & $\frac{l_1A_p}{J_y}$ & $\frac{l_1A_p}{J_y}$ & $\frac{-l_1A_p}{J_y}$ & $\frac{-l_1A_p}{J_y}$ & 0 & 0 & 0 & 0 & $-C_{3z_c}$ & $-C_{3\alpha}$ & $-C_{3\eta}$ & $\frac{C_1l_1}{J_y}$ & $\frac{C_2l_1}{J_y}$ & $\frac{-C_3l_2}{J_y}$ & $\frac{-C_4l_2}{J_y}$ \\
$\frac{K_1}{m_1}$ & $\frac{-bK_1}{2m_1}$ & $\frac{-l_1K_1}{m_1}$ & $\frac{-(K_1+k_1)}{m_1}$ & 0 & 0 & 0 & $\frac{-A_p}{m_1}$ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & $\frac{C_1}{m_1}$ & $\frac{-bC_1}{2m_1}$ & $\frac{-l_1C_1}{m_1}$ & $\frac{-(C_1+c_1)}{m_1}$ & 0 & 0 & 0 \\
$\frac{K_2}{m_2}$ & $\frac{bK_2}{2m_2}$ & $\frac{-l_1K_2}{m_2}$ & 0 & $\frac{-(K_2+k_2)}{m_2}$ & 0 & 0 & 0 & $\frac{-A_p}{m_2}$ & 0 & 0 & 0 & 0 & 0 & 0 & $\frac{C_2}{m_2}$ & $\frac{bC_2}{2m_2}$ & $\frac{-l_1C_2}{m_2}$ & 0 & $\frac{-(C_2+c_2)}{m_2}$ & 0 & 0 \\
$\frac{K_3}{m_3}$ & $\frac{bK_3}{2m_3}$ & $\frac{-l_2K_3}{m_3}$ & 0 & 0 & $\frac{-(K_3+k_3)}{m_3}$ & 0 & 0 & 0 & $\frac{-A_p}{m_3}$ & 0 & 0 & 0 & 0 & 0 & $\frac{C_3}{m3}$ & $\frac{bC_3}{2m_3}$ & $\frac{-l_2C_3}{m_3}$ & 0 & 0 & $\frac{-(C_3+c_3)}{m_3}$ & 0 \\
$\frac{K_4}{m_4}$ & $\frac{bK_4}{2m_4}$ & $\frac{+l_2K_4}{m_4}$ & 0 & 0 & 0 & $\frac{-(K_4+k_4)}{m_4}$ & 0 & 0 & 0 & $\frac{-A_p}{m_4}$ & 0 & 0 & 0 & 0 & $\frac{C4}{m4}$ & $\frac{bC_4}{2m_4}$ & $\frac{l_2C_4}{m_4}$ & 0 & 0 & 0 & $\frac{-(C_3+c_3)}{m_3}$
\end{bmatrix}$$

$$
[B_2]=\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&\gamma&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&\gamma&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&\gamma&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&\gamma&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
\frac{-\zeta K_1}{M}&\frac{-\zeta K_2}{M}&\frac{-\zeta K_3}{M}&\frac{-\zeta K_4}{M}&\frac{-\zeta C_1}{M}&\frac{-\zeta C_2}{M}&\frac{-\zeta C_3}{M}&\frac{-\zeta C_4}{M}&0&0&0&0&\frac{-\mu}{M}&\frac{-\mu}{M}&\frac{-\mu}{M}&\frac{-\mu}{M}\\
\frac{\zeta K_1 b}{2J_x}&\frac{-\zeta K_2 b}{2J_x}&\frac{-\zeta K_3 b}{2J_x}&\frac{\zeta K_4 b}{2J_x}&\frac{\zeta C_1 b}{2J_x}&\frac{-\zeta C_2 b}{2J_x}&\frac{-\zeta C_3 b}{2J_x}&\frac{\zeta C_4 b}{2J_x}&0&0&0&0&\frac{b\mu}{2J_x}&\frac{-b\mu}{2J_x}&\frac{-b\mu}{2J_x}&\frac{b\mu}{2J_x}\\
\frac{-\zeta K_1 l_1}{J_y}&\frac{-\zeta K_2 l_1}{J_y}&\frac{\zeta K_3 l_2}{J_y}&\frac{\zeta K_4 l_2}{J_y}&\frac{-\zeta C_1 l_1}{J_y}&\frac{-\zeta C_2 l_1}{J_y}&\frac{\zeta C_3 l_2}{J_y}&\frac{\zeta C_4 l_2}{J_y}&0&0&0&0&\frac{-l_1\mu}{J_y}&\frac{-l_1\mu}{J_y}&\frac{l_2\mu}{J_y}&\frac{l_2\mu}{J_y}\\
\frac{\zeta K_1}{m_1}&0&0&0&\frac{\zeta C_1}{m_1}&0&0&0&0&0&0&0&\frac{\mu}{m_1}&0&0&0\\
0&\frac{\zeta K_2}{m_2}&0&0&0&\frac{\zeta C_2}{m_2}&0&0&0&0&0&0&0&\frac{\mu}{m_2}&0&0\\
0&0&\frac{\zeta K_3}{m_3}&0&0&0&\frac{\zeta C_3}{m_3}&0&0&0&0&0&0&0&\frac{\mu}{m_3}&0\\
0&0&0&\frac{\xi K_4}{m_4}&0&0&0&\frac{\zeta C_4}{m_4}&0&0&0&0&0&0&0&\frac{\mu}{m_4}
\end{bmatrix}
$$

$$C_0 = \begin{bmatrix} 1 & -0.5b\alpha & -l_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0.5b\alpha & -l_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0.5b\alpha & l_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -0.5b\alpha & l_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

D= [0].

2.5 Implementation of the process model using MATLAB/SIMULIK package

MATLAB is a technical computing environment for high-performance numeric computation and visualization. SIMULINK is a part of MATLAB that can be used to simulate any linear or nonlinear dynamic systems. The SIMULINK model can be created by adding a new class of windows called the block diagram windows. In these windows, models are created and edited primarily by mouse-driven commands. To simulate any process model, a new Simulink environment should be opened by clicking on SIMULINK icon in the Matlab toolbar. The process model is constructed by drag-and-dropping the appropriate blocks from main SIMULINK Library Browser to environment windows. Figure 2.7 shows the MATLAB SIMULINK model of the full vehicle nonlinear active suspension systems with the road profile inputs: u_i exciting, bending torque input T_x and braking torque input T_y.

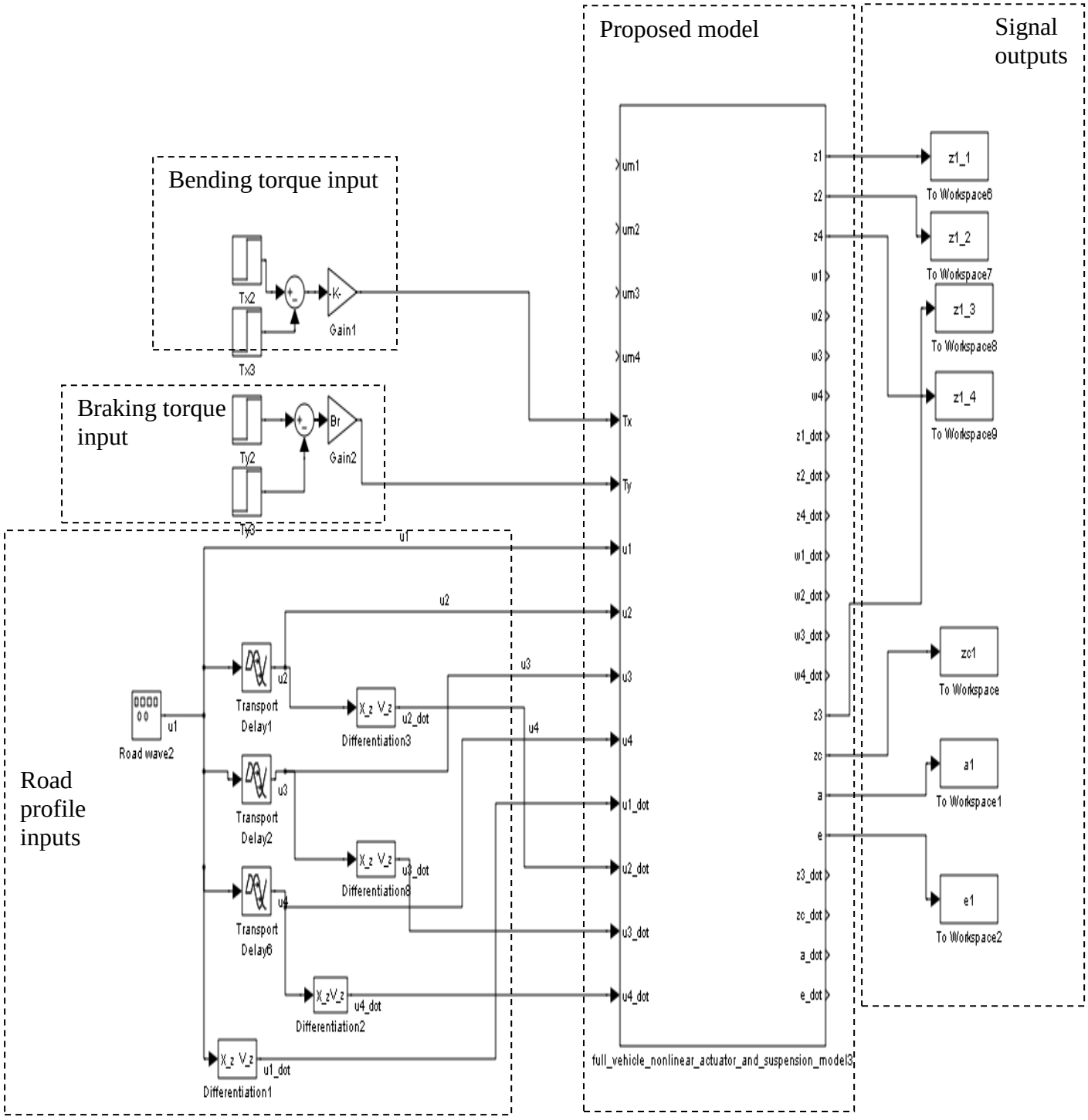


Figure 2.7 MATLAB SIMULINK model of the proposed system under test

As shown in Figure 2.7, the inputs of the proposed model are $U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$ where

$U_1 = [T_x \quad T_y \quad u_1 \quad u_2 \quad u_3 \quad u_4]^T$

$U_2 = [\dot{u}_1 \quad \dot{u}_2 \quad \dot{u}_3 \quad \dot{u}_4]^T$

The source block  is the signal generator that represents the random input with 0.01m amplitude and 0.1Hz frequency. To simulate the time delay of road exciting inputs for each tyre, the time delay blocks  are added. In this simulation, it is assumed that the time delay for the road exiting inputs u_1, u_2, u_3 and u_4 are 0, 0.3, 0.8 and 0.5 seconds respectively. To simulate the bending torque input, two unit step inputs have been used. The amplitude of the first unit step input is 5500 N/m and it is applied at 0 s. While, the amplitude of the second unit step input is -5500N/m and it is applied at 10 s. By adding these two signals, the amplitude of the equivalent signal is 5500 N/m and its effect is applied from 0 s to 10 s. The same technique is followed to simulate the braking torque, but the effect of the second unit step signal is applied at 5s. Therefore, the amplitude of equivalent signal, which is applied to represent the braking torque, is 5500 N/m from 0 s to 5 s. The output signals z_1, z_2, z_3, z_4, z_c, α and η are sent to the workspace of MATLAB program to save their data and these data will be used to plot the output responses of the proposed system. The construction of proposed model is very huge; therefore, the whole Simulink structure of the proposed model is explained individually in Appendix 1.

2.6 Simulation and results

To confirm that the design of the control system for full vehicle model with hydraulic actuators is necessary for meeting the control objectives or not, the time responses of the proposed model should be investigated without controller.

The numerical values of the hydraulic actuators and full vehicle model that are used in the simulation are taken from previous data referenced by Alleyne and Hedrick (1995) and Park and Kim (1998) respectively. The numerical values of the proposed model are given in Table (2.1) and (2.2).

Table 2.1 Numerical values of hydraulic actuators

Notation	Description	Values	Units
α, β and γ	Actuator parameters	$4.515*10^{13}$,1 and $1.545*10^{9}$	-
A_p	Cross section area of piston	$3.35*10^{-4}$	m^2
P_s	Supply pressure	10342500	Pa
τ	Time constant	1/30	s

The MATLAB and SIMULINK programme has been used to simulate the full vehicle nonlinear active suspension model with hydraulic actuators. The open-loop responses of full vehicle suspension systems, when the random inputs are applied as road excitation, are investigated. The plots of the open-loop responses will show if the control objectives can be met without using a control system for proposed model or not. The control objectives in this case are: minimizing the vibration sensed by the passengers when travelling on the rough roads and avoidance of the rollover of the vehicle when critical manoeuvres occur such as braking and cornering.

Figures 2.8-2.14 illustrate the time outputs response of full vehicle suspension systems with hydraulic actuators. The road profiles are assumed to be random input. Those Figures depict that the vertical displacement at each corner are unacceptable which means that the passive suspension systems are incapable of absorbing the vibrations excited by the road unevenness.

Table 2.2 Numerical values of full vehicle model

Notations	Description	Values	Units
K_1, K_2	Front-left and Front-right suspension stuffiness, respectively.	19960	N/m
K_3, K_4	Rear-right and rear-left suspension stuffiness, respectively.	17500	N/m
k_1-k_4	Front-left, Front-right, rear-right and rear-left tire stuffiness respectively.	175500	N/m
C_1, C_2	Front-left and Front-right suspension damping, respectively.	1290	N.s/m
C_3, C_4	Rear-right and rear-left suspension stuffiness, respectively.	1620	N.s/m
c_1-c_4	Front-left, Front-right, rear-right and rear-left tire damping, respectively.	14.6	N.s/m
M	Sprung mass.	1460	kg
m_1, m_2	Front-left, Front-right tire mass, respectively.	40	kg
m_3, m_4	Rear-right and rear-left tire mass, respectively.	35.5	kg
J_x	Moment of inertia x-direction.	460	$kg.m^2$
J_y	Moment of inertia y-direction.	2460	$kg.m^2$
l_1	Distance between the center of gravity of vehicle body and front axle.	1.011	m
l_2	Distance between the center of gravity of vehicle body and rear axle.	1.803	m
b	Width of track	1.51	m
ζ	Empirical parameter	0.1	-

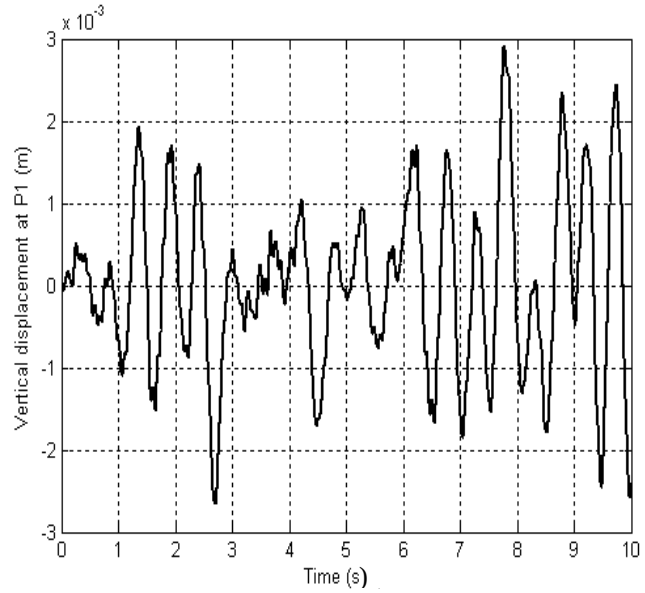


Figure 2.8 Time response of vertical displacment at P_1

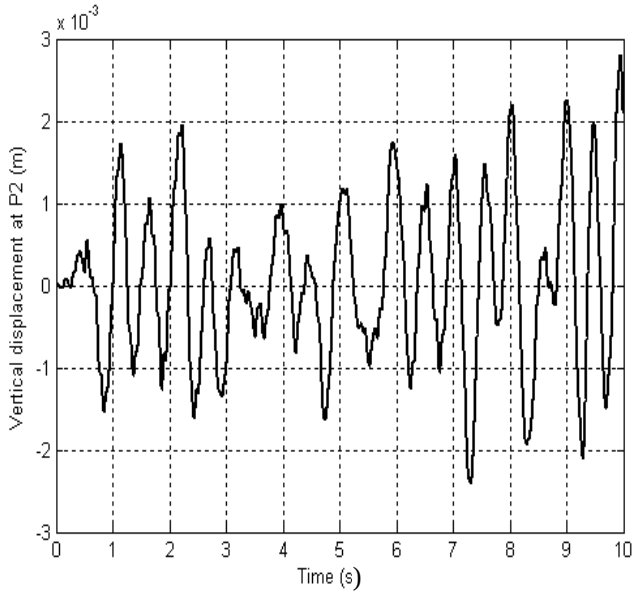


Figure 2.9 Time response of vertical displacement at P_2

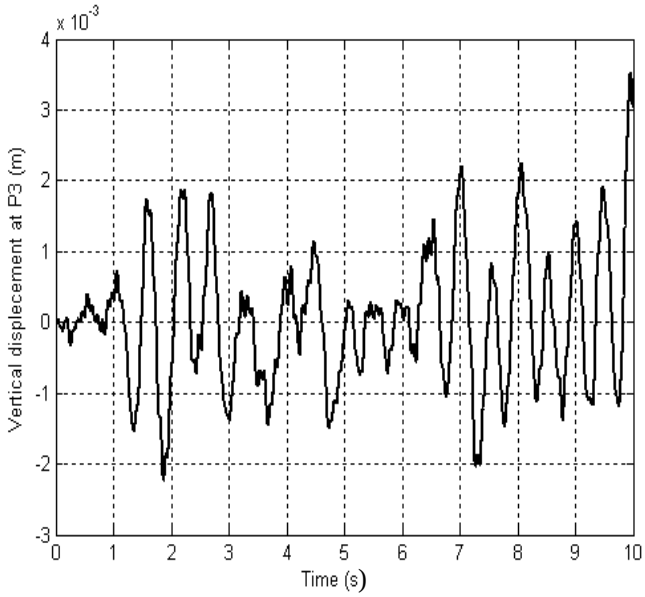


Figure 2.10 Time response of vertical displacement at P_3

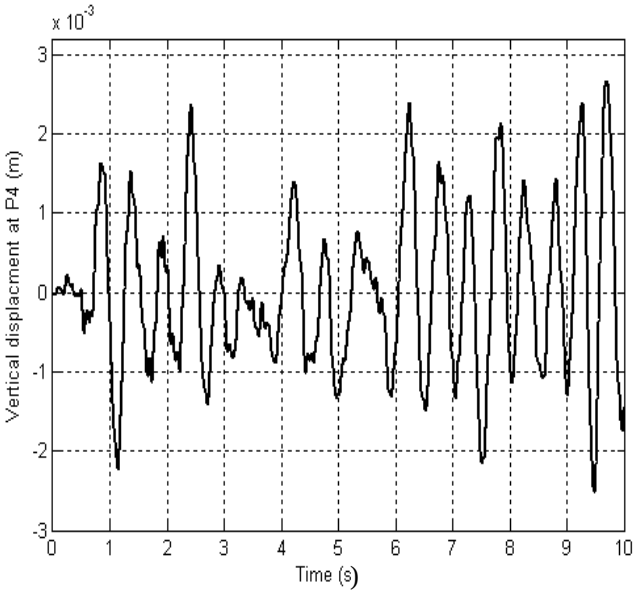


Figure 2.11 Time response of vertical displacement at P_4

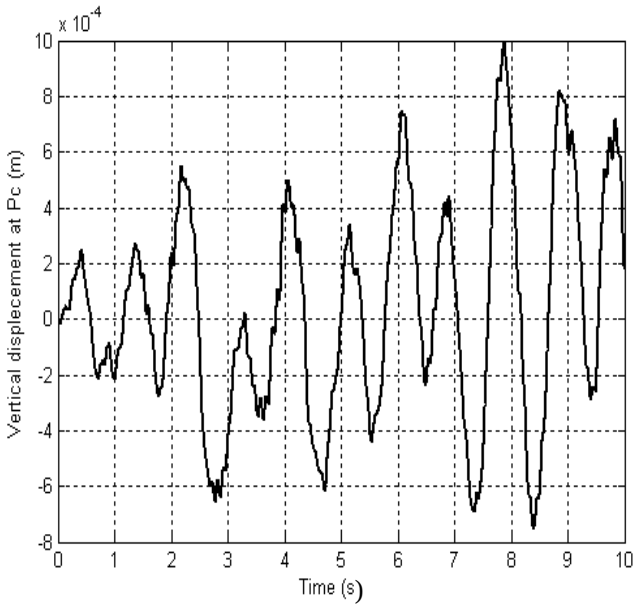


Figure 2.12 Time response of a vertical displacement at P_c

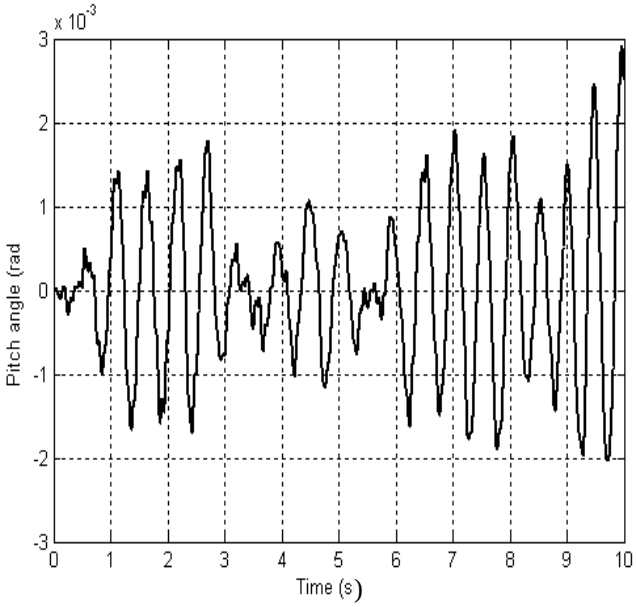


Figure 2.13 Time response of pitch angle

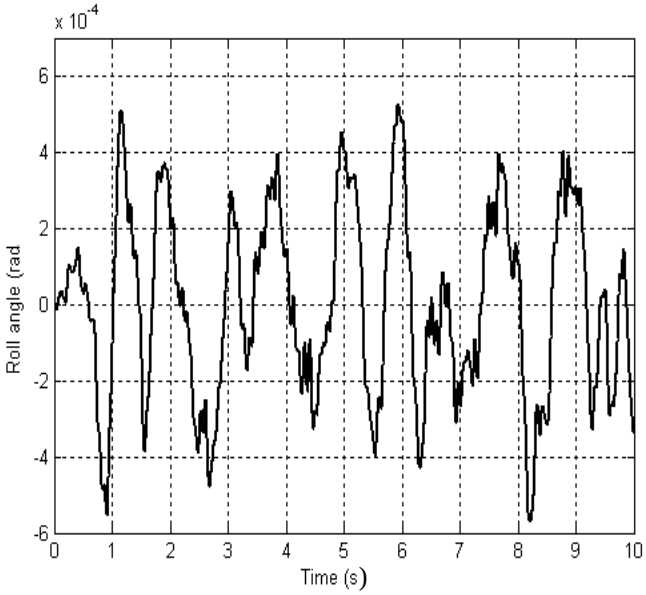


Figure 2.14 Time response of roll angle

To simulate the cornering of the vehicle, a bending torque has been applied as shown in Figure 2.15. The road profiles have been assumed to be random input. Figures 2.16-2.22 demonstrate the time output responses of the full vehicle suspension systems with hydraulic actuators. Those figures show that when the bending torque is applied, both the riding comfort and road handling are decreased.

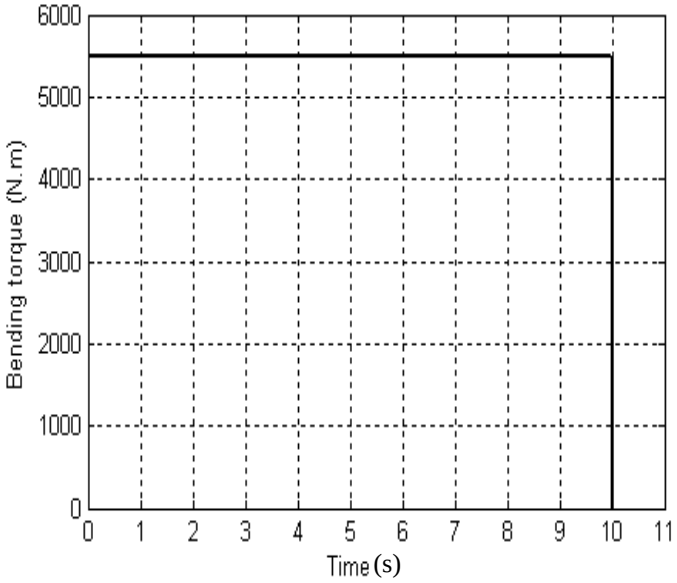


Figure 2.15 Bending torque

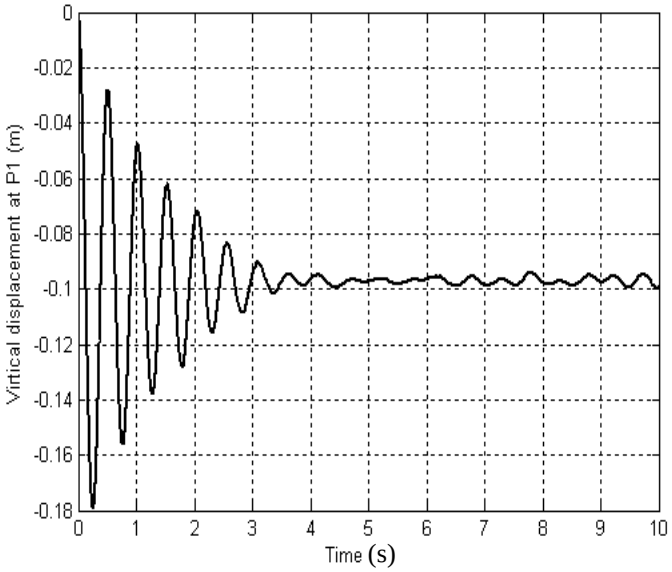


Figure 2.16 Time response of a vertical displacement at P_1

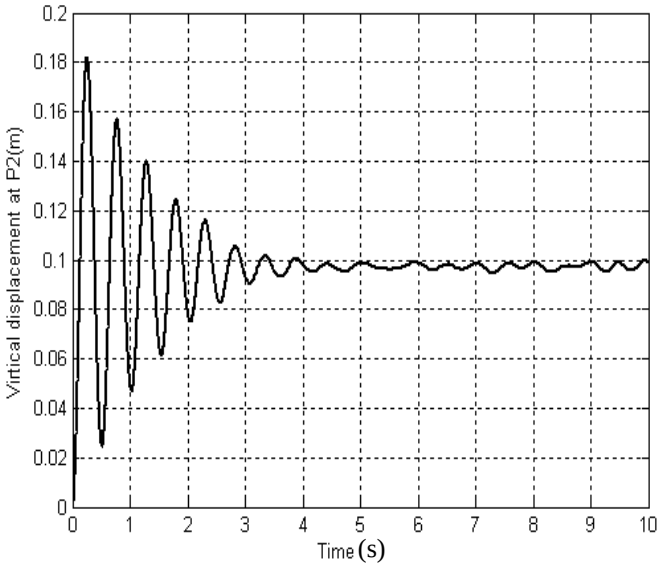


Figure 2.17 Time response of a vertical displacement at P_2

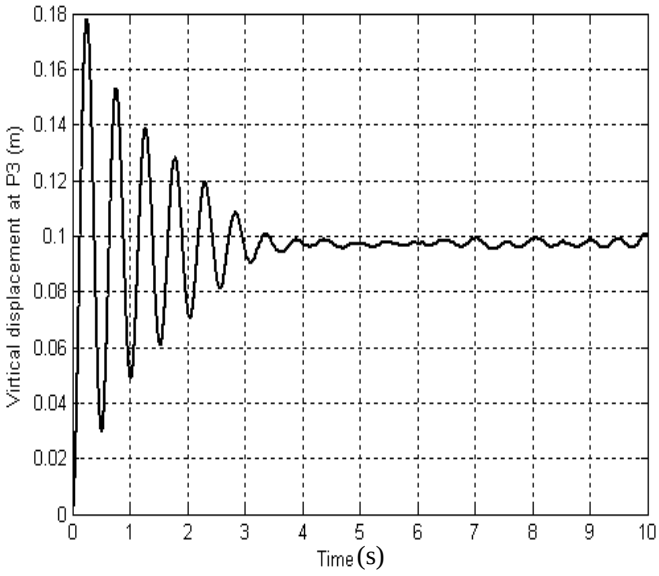


Figure 2.18 Time response of a vertical displacement at P_3

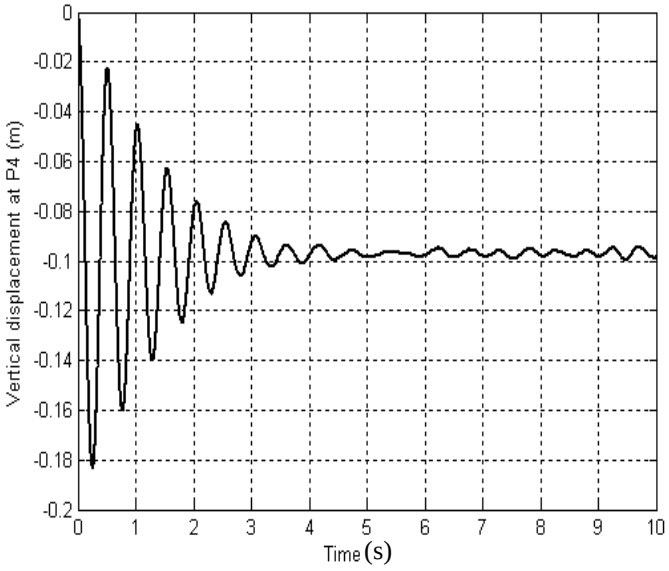


Figure 2.19 Time response of a vertical displacement at P_4

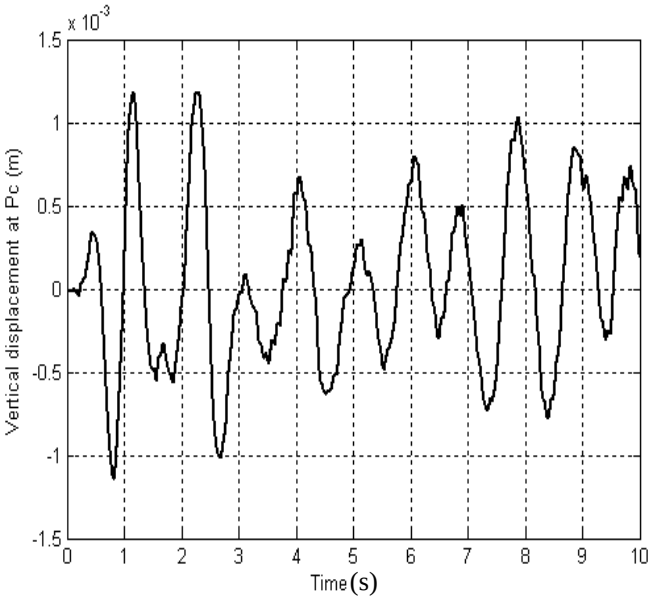


Figure 2.20 Time response of a vertical displacement at P_c

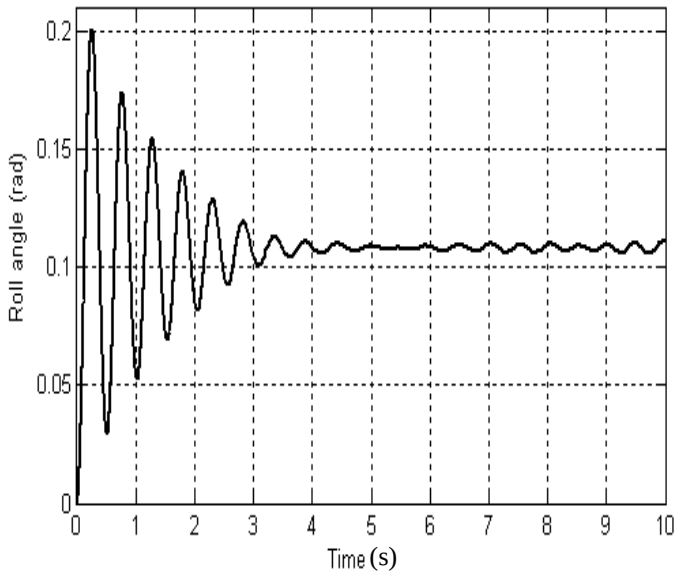


Figure 2.21 Time response of roll angle

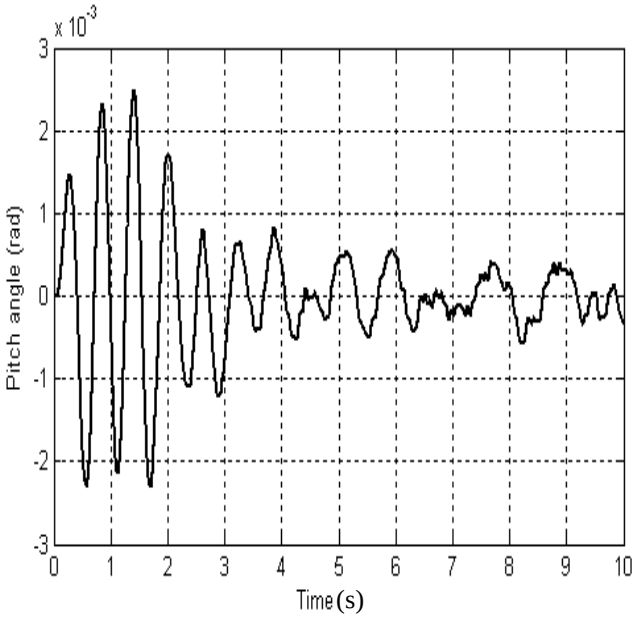


Figure 2.22 Time response of pitch angle

To simulate the braking of the vehicle, a braking torque has been applied as shown in Figure 2.23. The road profiles have been assumed to be random input. Figures 2.24-2.30 illustrate the time output responses of the full vehicle suspension systems with hydraulic actuators. After the braking torque is applied, the discomfort felt by the passangers is increased and the contact forces between the tyres and the road surface are decreased.

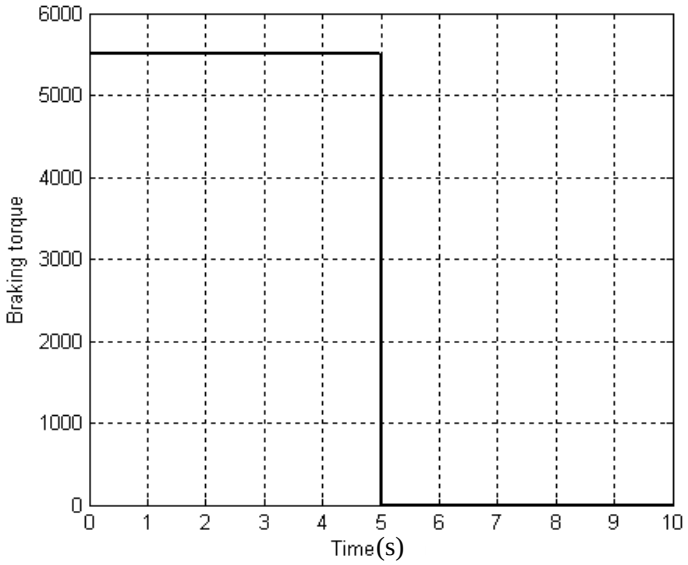


Figure 2.23 Braking torque

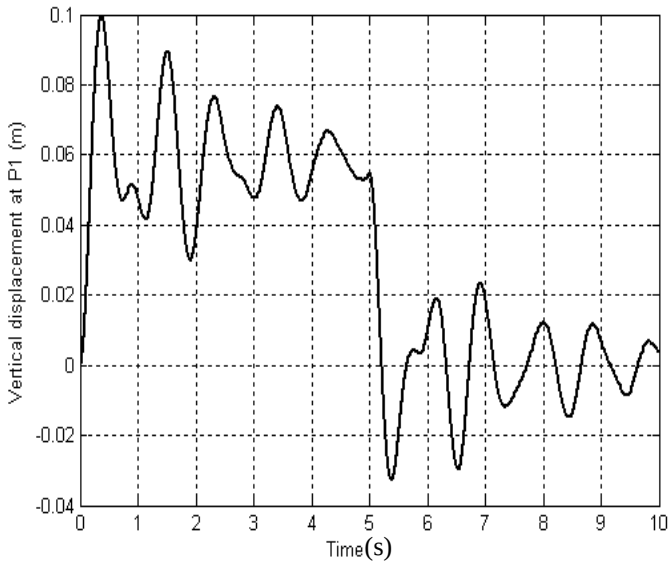


Figure 2.24 Time response of a vertical displacement at P_1

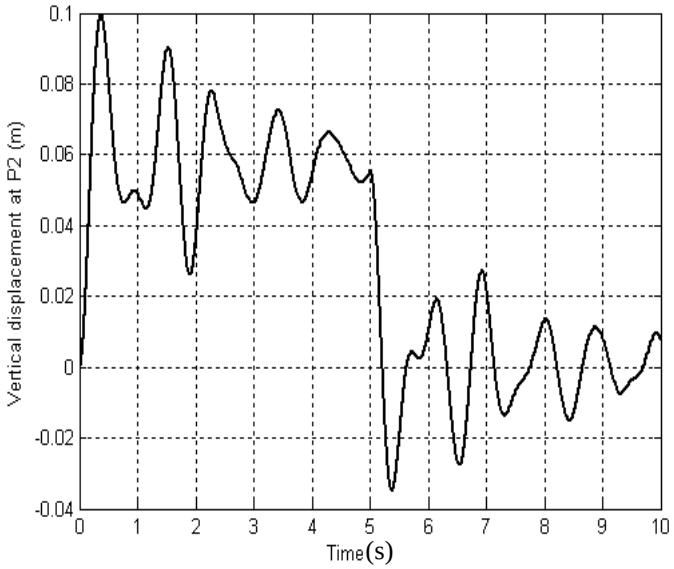


Figure 2.25 Time response of a vertical displacement at P_2

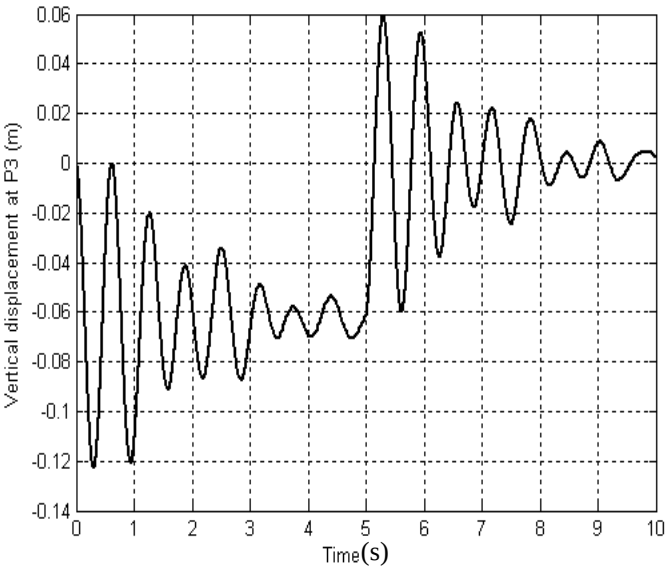


Figure 2.26 Time response or a vertical displacement at P_3

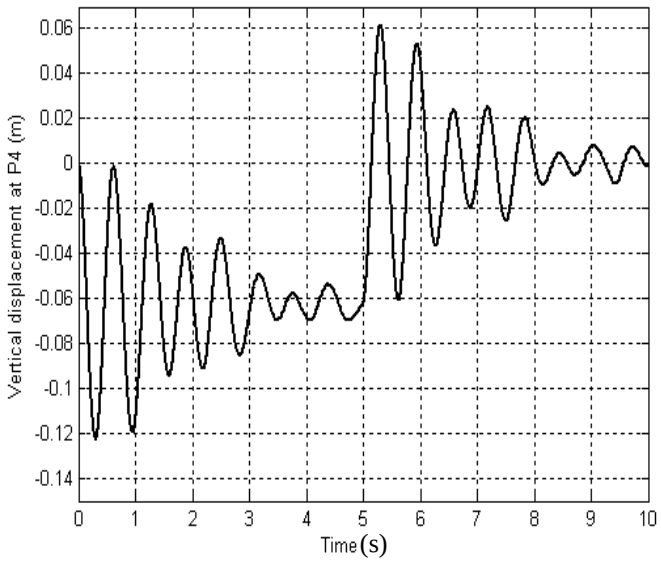


Figure 2.27 Time response of a vertical displacement at P_4

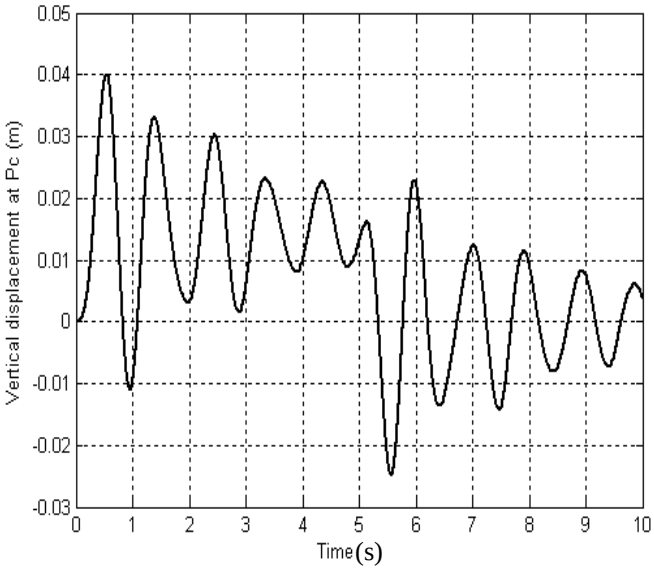


Figure 2.28 Time response of a vertical displacement at P_c

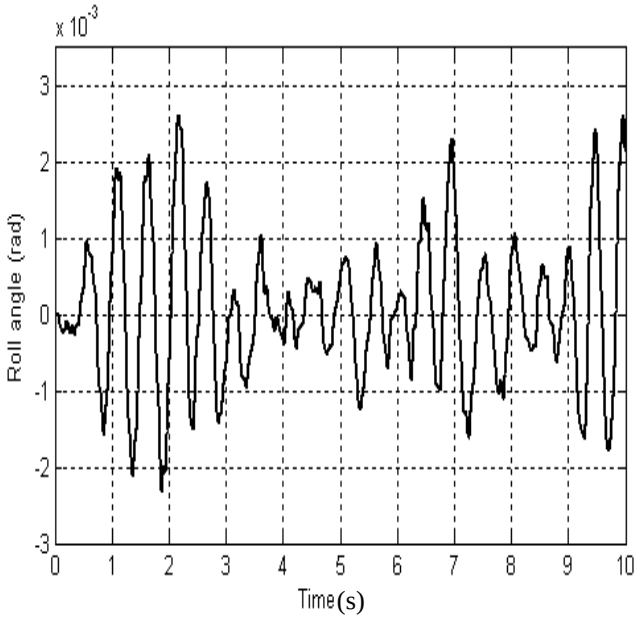


Figure 2.29 Time response of roll angle

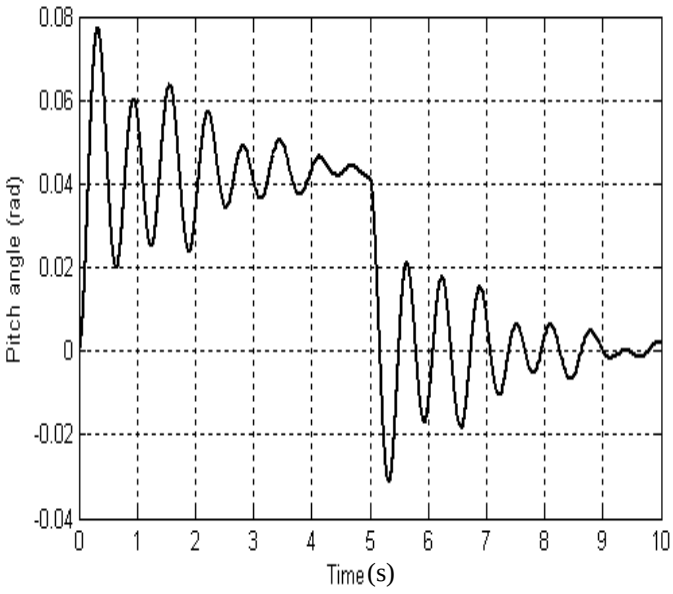


Figure 2.30 Time response of pitch angle

2.7 Summary

Regarding the historical review, the researchers designed control systems for quarter, half or full model of suspension systems. Some researchers suggested control systems for linear suspension models, while the other suggested control systems for nonlinear suspension models. Without doubt, the nonlinearity and uncertainties inherently exist in the suspension system; therefore, the nonlinear effect has to be taken into account when designing the robust control system. In this work, full vehicle nonlinear active suspension model has been used to take into account the three movements: vertical movement, pitching movement and rolling movement. From the literature review, the report of the intelligent control design problems for full vehicle nonlinear active suspension systems with hydraulic actuators has not been found. Also, the frictional forces inside the hydraulic actuators were not covered, either. Therefore, the effects of the nonlinear hydraulic actuators with frictional forces were taken into account. The mathematical model equations for full vehicle nonlinear active suspension systems with nonlinear hydraulic actuators have been derived in this chapter.

The results in the previous section show that the vibrations sensed by the passengers are sensible. When the inputs to the vehicle system are just the road random profile inputs, both the riding comfort and road handling are decreased. The amplitude of the output transient responses should be small values. Furthermore, when the bending torque is applied to simulate the cornering (or when the braking torque is applied to simulate braking) with random inputs as road profile, the output transient responses oscillate around a specific value. The desired output responses should be over damping responses with very small steady state value. Therefore, the passive suspension systems without control element are not suitable to eliminate the vibrations which arise from travelling on the rough road or sharp manoeuvres

such as pitching and rolling movements. In following chapters, an intelligent control system will be designed to achieve the design requirements.

3. Evolutionary Algorithm Based Fractional Order $PI^{\lambda}D^{\mu}$ Controller

3.1 PID's function and improvement

The PID term refers to the first letter of the names of the individual terms that make up the standard three term controller. These are: *P* for the proportional term; *I* for the integral term and *D* for the derivative term in the controller. The PID controllers are widely being used in the industries for process control applications. Even for complex industrial control systems, the PID control module is used to build the main controller. The merit of using PID controllers lies in its simplicity of design and good performance including low percentage overshoot and small settling time for normal industrial processes (Astrom and Hagglund 1995). Several formulas have been proposed to select suitable parameters of PID controllers. Zielger-Nichols tuning formula was proposed in 1942 to tune the PID control parameters. This tuning formula can be applied when the plant model is given by a first-order plus dead time. Many variants of the traditional Ziegler-Nichols PID tuning methods have been proposed such as the Chien-Hrones-Reswick formula, Cohen-Coon formula, refine Ziegler-

Nichols tuning formula, Wang-Juang-Chan formula and Zhuang-Atherton optimum PID controller (Xue et al. 2007).

The performance of the PID controllers can be further improved by appropriate setting of fractional-I and fractional-D actions. In Fractional Order PID (FOPID) controller, besides tuning the proportional constant; derivative constant; and integral constant (K_p, K_d and K_i), the derivative order μ parameter and the integral order λ parameter must be tuned too. Therefore, the two extra degrees of freedom, which come from adding the fractional order integrator and differentiator, will add more flexibility to the design the FOPID controller. Several methods have been introduced to tune the five parameters of FOPID controller. An alternative design method is introduced based on Particle Swarm Optimization (PSO) to design the FOPID by Ghartemani et al. (2007); Cao and Cao (2006) and Maiti et al. (2008). In Reference Vinagre et al. (2000), a frequency domain approach based on the expected crossover frequency and phase margin was performed to design the FOPID controller. A frequency-band broken–line approximation method was proposed to tune the parameters of fractional-order $PI^{\lambda}D$ controller (Ma and Hori 2004). Dorcak et al. (2001) proposed state-space design method based on feedback poles placement to design the FOPID controller. The optimization design process based on Genetic Algorithm was designed by Cao et al. (2005) and Meng and Xue (2009) to tune the parameters of FOPID controller. Biswas et al. (2008) proposed Differential Evolution to design of fractional-order PID controllers involving fractional-order integrator and differentiator. The Evolutionary Algorithms was proposed as a design method to tune the parameters of the FOPID controller (Kiani and Pariz 2007).

In this Chapter, the parameters of FOPID controller will be tuned using an Evolutionary Algorithm (EA) to control the full vehicle nonlinear active suspension systems with hydraulic actuators. The Evolutionary Algorithm (EA) is a modern technique for searching complex spaces for an optimum. This algorithm operates on a population of potential solutions by

applying the principle of survival of the fittest to produce perfect approximations to a solution. Therefore, the evolutionary algorithm has been used for modifying the optimal values of K_p, K_i, K_d, λ and μ. The proposed FOPID controller has been designed for full vehicle nonlinear active suspension model including hydraulic actuators in order to improve the riding comfort and road handling during various manoeuvres: travelling, braking and cornering. Six types of disturbances have been applied to examine the robustness and effectiveness of the proposed controller.

3.2 Fractional order $PI^{\lambda}D^{\mu}$ controller

Fractional order differential equation is used to describe the fractional order $PI^{\lambda}D^{\mu}$ controller. As mentioned before, in PID controller case, three parameters K_p, K_d and K_i should be tuned to design the controller. One of the possibilities to improve PID controllers is to use fractional-order $PI^{\lambda}D^{\mu}$ controllers with real order of derivative and integral.

Recently, some of authors proposed the FOPID controller for different applications. Ghartemani et al. (2007) applied Fractional Order $PI^{\lambda}D^{\mu}$ (FOPID) controller for an Automatic Voltage Regulator (AVR). Ma and Hori (2004) proposed a frequency band fractional order $PI^{\lambda}D$ controller for speed control of the two inertia system, which is a basic control problem in motion control. A fractional order $PI^{\lambda}D^{\mu}$ controller is performed for a position servomechanism control system considering actuator saturation and the shaft tensional flexibility.

For actually implementation, the authors introduced a modified approximation method to realise the designed fractional order $PI^{\lambda}D^{\mu}$ controller (Xue et al. (2006)). The differential equation of fractional order controller was described as (Xue et al. (2007))

$$u(t) = K_p e(t) + K_i D_t^{-\lambda} e(t) + K_d D_t^{\mu} e(t) \tag{3.1}$$

where *e(t)* is the error between a measured process output variable and a desired set point, and *u(t)* is the control signal. Eq. (3.1) illustrates that the five control parameters (K_p, K_i, K_d, λ and μ) should be tuned to design an accurate controller. The integral order and derivative order add more flexibility to design an FOPID controller.

The Laplace transform of the fractional order differentiation equation is defined as (Xue et al. 2007)

$$L[D_t^{\alpha} f(t)] = s^{\alpha} L[f(t)] - \sum_{k=1}^{n-1} s^k [D_t^{\alpha-k-1} f(t)]_{t=0} \tag{3.2}$$

where $(n-1) \leq \alpha < n$.

If the derivatives of the function *f(t)* are all equal to *0* at *t=0*, the second part of the right hand side of Eq. (3.2) is omitted. Therefore, Eq. (3.2) can be rewritten as

$$L[D_t^{\alpha} f(t)] = s^{\alpha} F(s) \tag{3.3}$$

where *F(s)* is the Laplace transform of *f(t)*.

Therefore, the continuous transfer function of FOPID controller can be given by

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu} \tag{3.4}$$

The orders λ and μ can be any real numbers. From Eq. (3.4) the classical PID controller can be obtained by setting λ=μ=1. When λ=1 (or 0) and μ=0 (or 1) a normal PI (or PD) controller can be obtained. Figure 3.1 shows that the fractional order $PI^{\lambda}D^{\mu}$ controller generalises and expands the classical PID controller from a point to plane. This expansion gives the designers more flexibility to design an accurate controller. The fractional-order $PI^{\lambda}D^{\mu}$ controller is explained with the horizontal axis as the order of the integrator and the vertical axis as the order of the differentiator. It can be seen that the ordinary PI ,PD and PID controllers are special cases of the fractional-order $PI^{\lambda}D^{\mu}$ controller since the values of λ and μ can be selected freely, which adds two more degree of freedom to the controller design.

The big challenge here is that how can the designers select the optimal values of the control parameters to design an accurate controller. As mentioned in Section 3.1, the researchers proposed different algorithms to modify the parameters of FOPID controller. In this chapter, Evolutionary Algorithm has been proposed to design the FOPID controller.

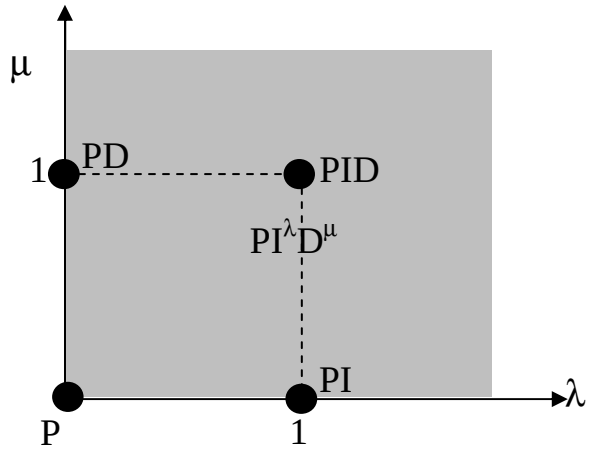


Figure 3.1 Generalized FOPID controller

3.3 Evolutionary algorithm

The Evolutionary Algorithm (EA) is an optimization algorithm that is used to search for optimal solutions to a problem (Back 1996). This algorithm operates on a population of potential solutions applying the principle of survival of the fittest to produce best approximations to a solution. Evolutionary algorithms provide a universal optimization technique that mimics the type of genetic adaptation that occurs in natural evolution (Goldberg 1989 and Back 1996). Unlike specialised methods designed for particular types of optimization tasks, they require no particular knowledge about the problem structure other than the objective function it self. At each iteration step, a new set of approximations is assumed by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators, such as mutation, crossover and selection borrowed from natural genetics in order to generate the new generations (Sumathi et al. 2008). It shares the same features with the genetic algorithm (GA) in terms of selection

method, crossover strategy, mutation scheme and reinsertion policy. The main difference between evolutionary algorithm and genetic algorithm lies in the representation of the individual solutions (population). In evolutionary algorithm, the individuals represent as real numbers, whereas genetic algorithm encodes the values of individuals to generate the new population (Hoffmann 2001).

3.3.1 Structure of evolutionary algorithm

By using the Evolutionary Algorithm, the natural processes such as selection, crossover, mutation and reinsertion are modeled. Figure 3.2 depicts the structure of a generic evolutionary algorithm. At the beginning of the computation a number of individuals (each individual is D-dimension variables vector) representing the candidate solutions which are randomly initialised. Those candidate solutions represent the current population *P(t)* (first generation). The objective function is then evaluated for these individuals to obtain the fitness for each individual. If the optimization criteria are not met, the creation of new generation will be started. The individuals that have high fitness are more likely to be selected as a parent to generate the offspring for a new generation. The best parents are recombined to produce offspring. This process is called crossover. The crossover operator is applied to the mating pool with the hope that it creates a better offspring. After a crossover, offspring will be subjected to mutation which is randomly reforms an offspring to generate new variants. The fitness of the offspring is then computed. The offsprings that have high fitness are inserted into the population to create a new generation *P(t+1).* The cycle is performed until the maximum number of generations elapse or a desired level of fitness is reached.

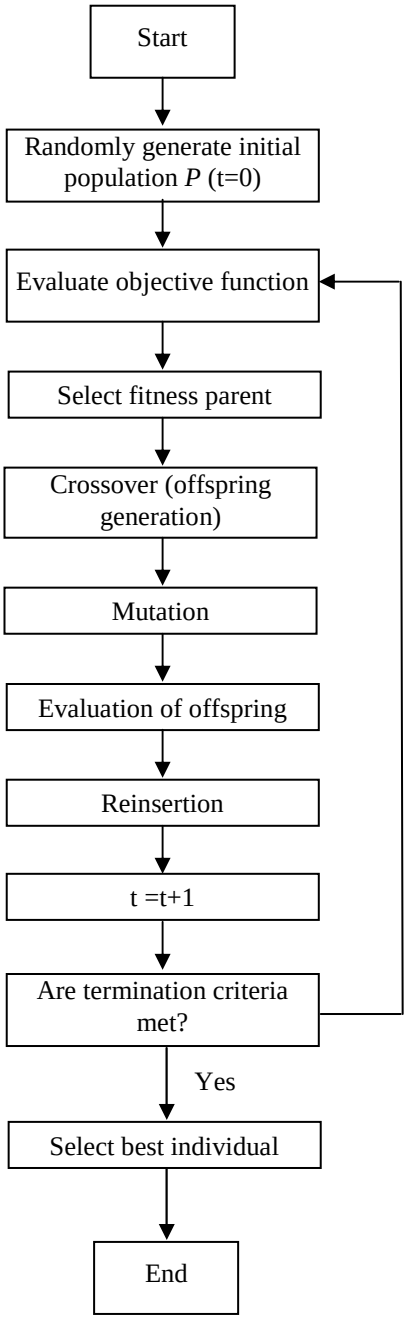


Figure 3.2 Structure of an evolutionary

3.3.2 Components of evolutionary algorithm

The evolutionary algorithm (EA) has a number of components; the most important components of EA can be listed as

- ***Evaluation or Fitness Function**:* It is a measurement function that quantifies the fitness for each individual, so that particular individual may be ranked against all the other candidate solutions. The individuals that have high fitness are allowed to breed and mix their values to produce the next generations which may be even better than the pervious generations. There are two formulas that can be used to compute the fitness value for an individuals

  1. *Linear ranking fitness formula*: in this formula the fitness value can be computed by this equation

$$Fitness(Pos) = 2 - SP + 2 * (SP - 1) * \frac{(Pos - 1)}{(N_{ind} - 1)} \tag{3.5}$$

  where N_{ind} is the number of individuals in the population; *Pos* is the position of an individual in this population (least fit individual has *Pos*=1, the fittest individual *Pos*= N_{ind}); and SP is the selective pressure. The selective pressure can be defined as: the probability of the best individual is selected to compare with the average probability of selection of all individuals.

  2. Nonlinear Ranking fitness formula: the fitness value can be given by

$$Fitness(Pos) = \frac{N_{ind} * X^{Pos-1}}{\sum_{i=1}^{N_{ind}} X^{i-1}} \tag{3.6}$$

  where X is computed as the root of the following polynomial

$$0 = (SP - N_{ind}) * X^{N_{ind}-1} + SP * X^{N_{ind}-2} + \cdots + SP * X + SP$$

- ***Selection**:* It is used to determine the best individuals according to their fitness values. The selected individuals are represented the parents that will recombine to generate the offsprings. There are many methods that can be used to select the best parents such as Roulette-Wheel Selection, Stochastic Universal Sampling, Local Selection, Truncation Selection, and Tournament Selection.

In this project, the Roulette-wheel selection has been used to determine the best parents.

Roulette-Wheel Selection: This is a stochastic algorithm and involves the following technique: the individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness value. A random number is generated and the individual whose segment matches that random number is selected. The process is repeated until the desired number of individuals is obtained (called mating population). Figure 3.3 shows the distribution of six individuals in roulette wheel according to their fitness value

Pos	*Fitness*
1	0.15
2	0.95
3	0.36
4	0.45
5	0.84
6	0.22

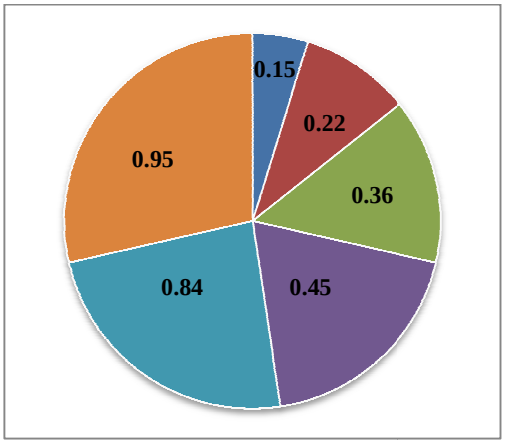


Figure 3.3 Roulette Wheel Selection

Table 3.1 demonstrate the fitness values and selection probability for 11 individuals (linear ranking and selective pressure *SP=2*). For selecting the mating population, the appropriate number of uniformly distributed random numbers (uniform distributed between 0.0 and 1.0) is independently generated.

For a sample of 6 random numbers: 0.81, 0.32, 0.96, 0.01, 0.65, and 0.42.

Position of individual	1	2	3	4	5	6	7	8	9	10	11
Fitness value	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.0
Selection probability	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02	0.0

Table 3.1 Selection probability and fitness value

Figure 3.4 shows the selection process of the individuals for the example in the table together with the above sample trials. After selection the mating, population consists of the individuals: 1, 2, 3, 5, 6, and 9.

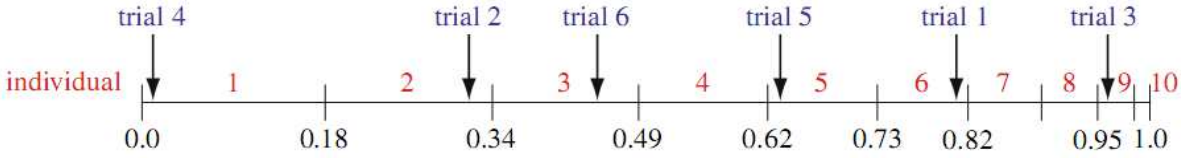


Figure 3.4 The selected individuals using roulette wheel selection

- ***Crossover:*** In this step the best parents are recombined to produce the new offsprings. The types of recombination can be listed as

1) Intermediate Crossover
2) Line Crossover
3) Extended Line Crossover.

The intermediate crossover has been selected in this work.

Intermediate Crossover: Offsprings are produced according to the following formula:

$$Var_j^O = Var_j^{P2} * a_j + Var_j^{P1} * \left(1 - a_j\right) \quad j \in (1, 2, \dots, N_{var}) \tag{3.7}$$

where a_j uniform at random, Var_j^O is the j^{th} offspring (new individual) variable, Var_j^{P1} is the j^{th} parent1 (individual1) variable, Var_j^{P2} is the j^{th} parent2 (individual2) variable and N_{var} is the number of variables in each individual.

Consider the following two individuals with 3 variables each:

individual1 12 25 5

individual2 123 4 34

The *a* chosen for this example are:

sample1 0.5 1.1 −0.1

sample2 0.1 0.8 0.5

The new individuals are calculated as:

offspring1 67.5 1.9 2.1

offspring2 23.1 8.2 19.5

- ***Mutation:*** It means that reformation the offsprings by adding random values to the variables of an offspring with low probability. The probability of mutating a variable is inversely proportional to the number of variables (dimensions).
- ***Reinsertion:*** Once the offspring have been produced by selection, recombination and mutation of individuals from the old population, the fitness of the offspring may be determined. If the numbers of the produced offsprings are less than the size of the original population, the produced offsprings should be reinserted into the old population to maintain the size of the original population. Similarly, if not all produced offsprings are to be used at each generation or if the numbers of the produced offsprings are bigger than the size of the old population, then the reinsertion scheme must be used to determine which individuals are to exist in the new population. Different methods are used to reinsert the offspring into the old population to obtain the new generation. These methods can be listed as

  1. *Pure reinsertion method*: Produce as many offspring as parents and replace all parents by the offspring.

2. *Uniform reinsertion method*: Produce less offspring than parents and replace parents uniformly at random.
3. *Elitist reinsertion method*: Produce less offspring than parents and replace the worst parents.
4. *Fitness-based reinsertion method*: Produce more offspring than needed for reinsertion and reinsert only the best offspring.

The Elitist reinsertion method has been used in this work to generate the new population.

3.4 Implementation of the FOPID controller with proposed system using MATLAB/SIMULINK package

Figure 3.5 shows the layout of the Simulink model of FOPID controller with controlled system.

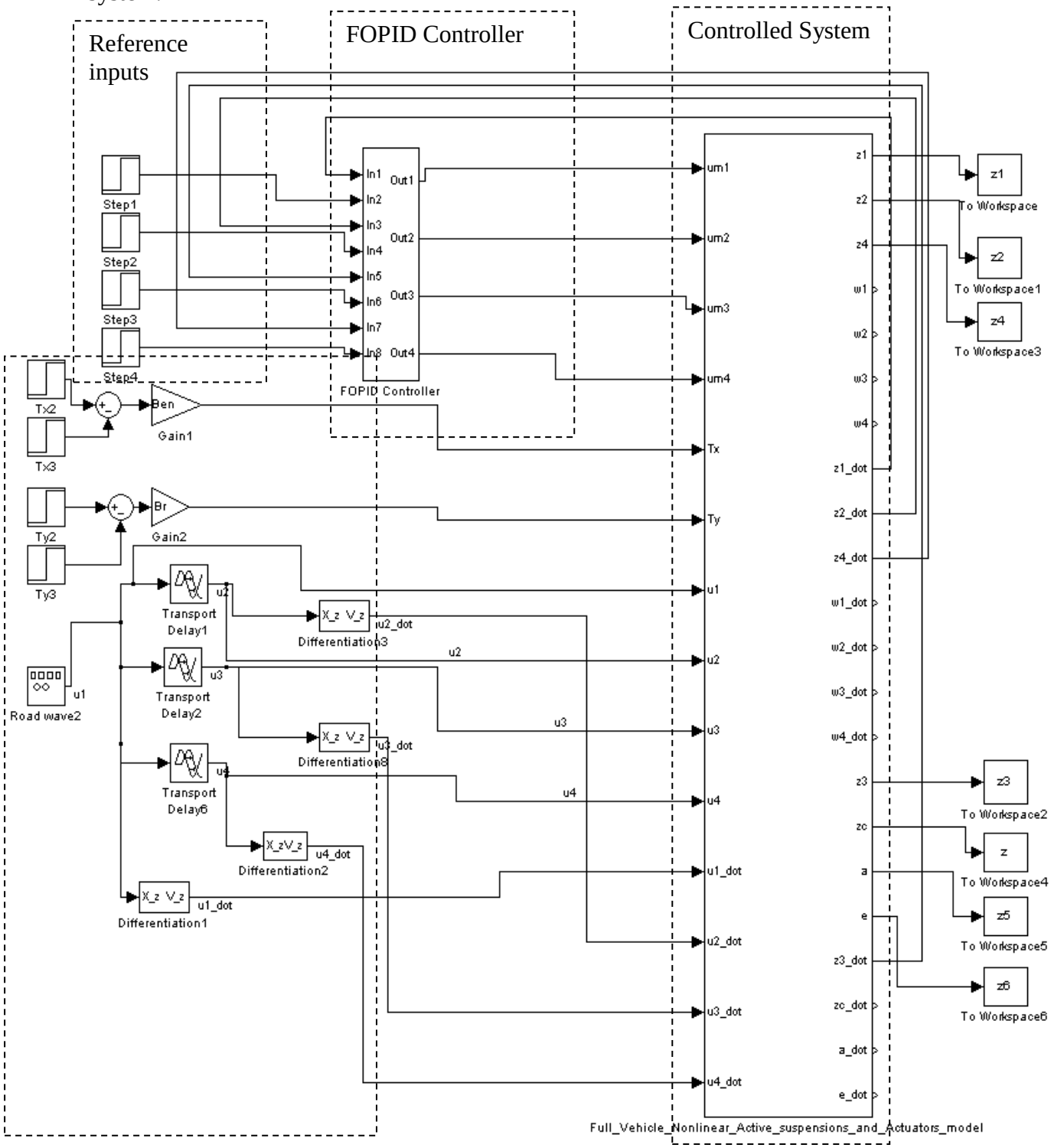


Figure 3.5 Simulink model of FOPID Controller with the Proposed System

The amplitude of reference inputs are set to be 0 m, because the desire outputs of the controlled system should be forced to zero. The FOPID controller model has eight inputs four outputs. The outputs of this model are applied as valves control signals of the hydraulic actuators (u_{m1}, u_{m2}, u_{m3} and u_{m4}). The construction of FOPID controller model is depicted in Figure 3.6.

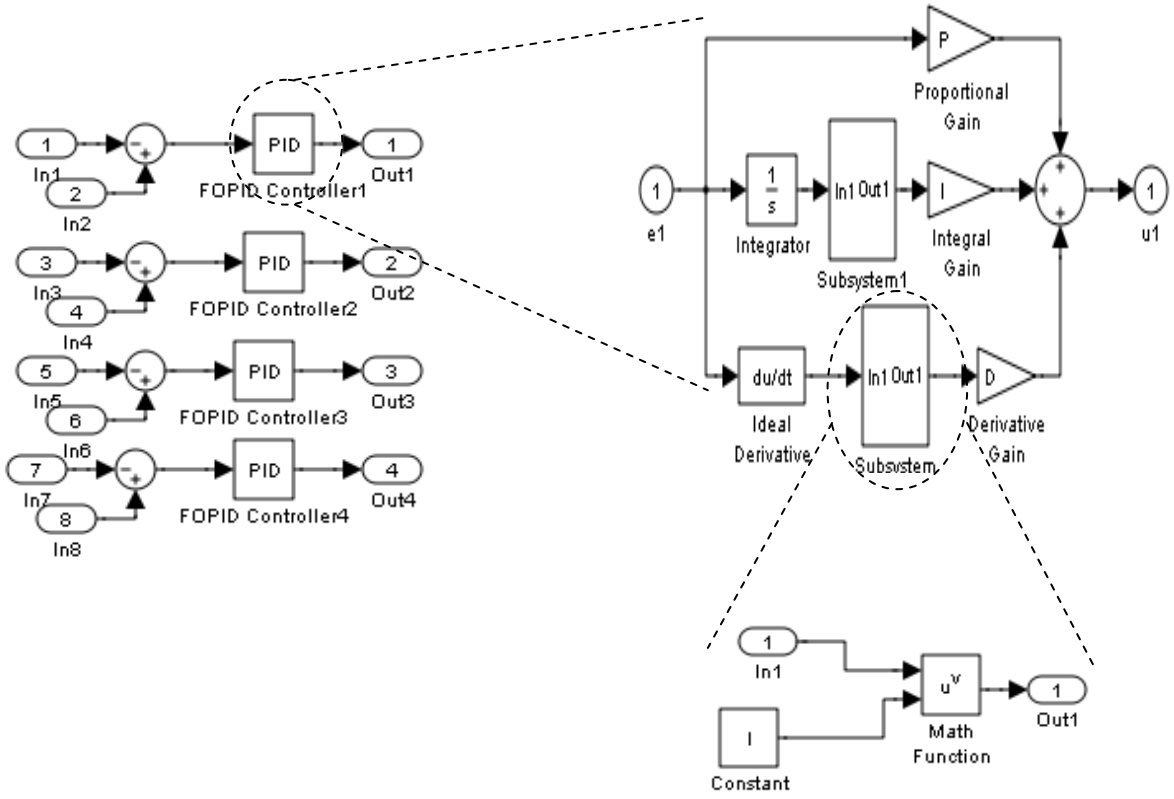
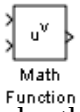


Figure 3.6 Construction of FOPID Controller Model

As shown from Figure 3.6, the FOPID controller consists of four FOPID sub-controllers each one is used to generate a control signal for one hydraulic actuator. The math function block illustrated in Figure 3.6 

has two inputs ports. The first port corresponds to the base of the function, which may be the derivative of the error or integral of the error and the other port corresponds to the power of the function (constant).

3.5 Simulation and results

The Evolutionary Algorithm has been used to modify the parameters of the FOPID controller for full vehicle nonlinear active suspension system with hydraulic actuators. The nonlinear model of the active suspension system with nonlinear hydraulic actuators that is described in Chapter 2 has been use as a controlled system. As shown in Figure 2.5, the full vehicle model has four suspensions system, therefore, four FOPID sub-controllers should be designed at which one controller for each suspension system. The OptiY 4.0 program is used to obtain the optimal parameters of the FOPID controller.

From Eq. 3.4, five parameters K_P, K_d, K_i, λ and μ should be designed for each suspension. Each individual vector has five parameters (five-dimension variables vector). For reducing the time of optimization, the ranges of FOPID parameters are selected as $K_P \in [0\ 20000], K_d \in [0\ 4000], K_i \in [0\ 1500], \lambda \in [0\ 1]\ and\ \mu \in [0\ 1]$. To evaluate the objective function, the following function will be used:

$$J_f = 0.5 \sum_{\varepsilon=1}^{4} z_{\varepsilon}^2 \tag{3.8}$$

The stop criterion of the computation of evolutionary loop is the one that defines the maximum number of generations being produced.

The initial values and optimal values of the FOPID controller parameters are shown in Table 3.2. Figures 3.7- 3.11 demonstrate the changing of the control parameters (values of K_p, K_d, K_i, λ and μ) during the optimization steps. After 225 optimization steps, the parameters of the FOPID controller reach to steady state values (them optimal values) while the objective function J_f value reach to small value as illustrated in Figure 3.12.

During optimization phase, it is assumed that the inputs to the full vehicle model are just the road uneven excitation and control forces. The white noise input is applied as a rode profile.

Parameter	Initial value	Optimal value
K_p	100	12678.26
K_d	20	3253.92
K_i	1	768.1
λ	0.3	0.45
μ	0.7	0.886

Table 3.2 Initial and optimal values of FOPID controller

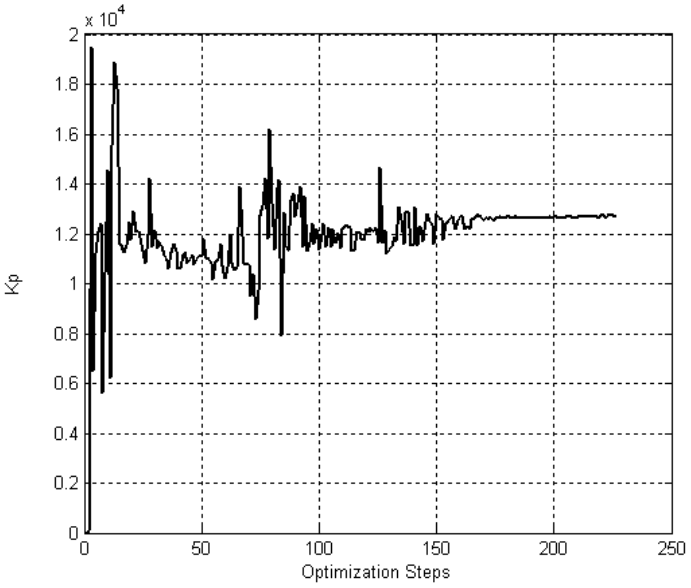


Figure 3.7 Changing value of K_p during optimization steps

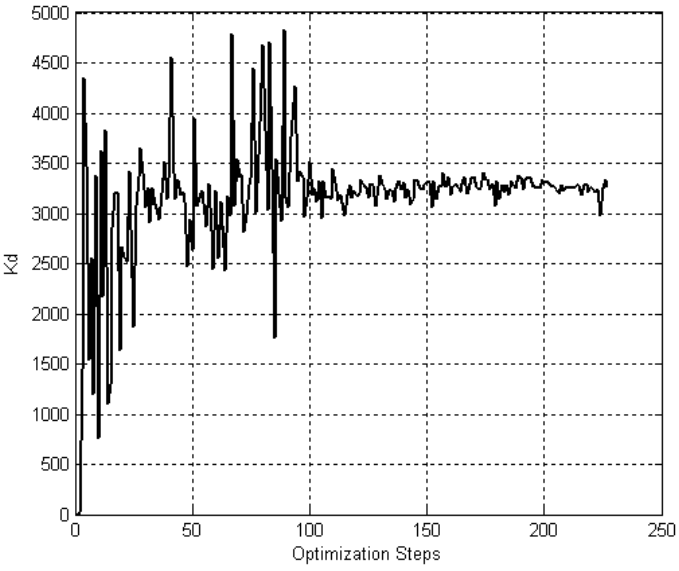


Figure 3.8 Changing value of K_d during optimization steps

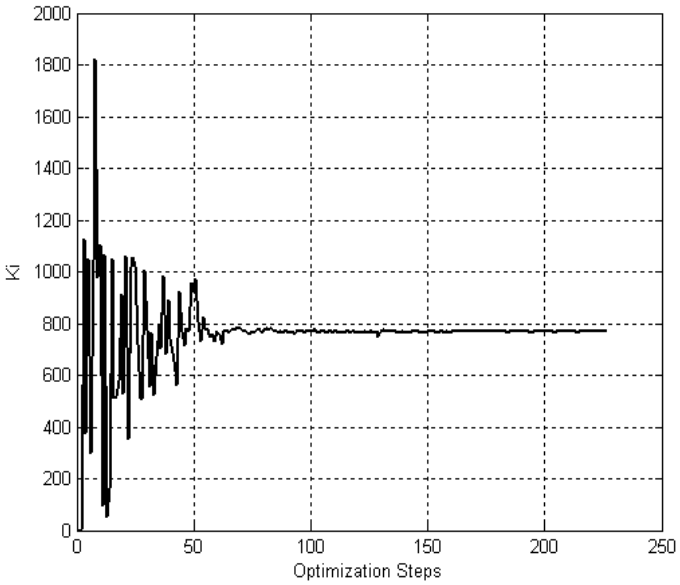


Figure 3.9 Changing value of K_i during optimization steps

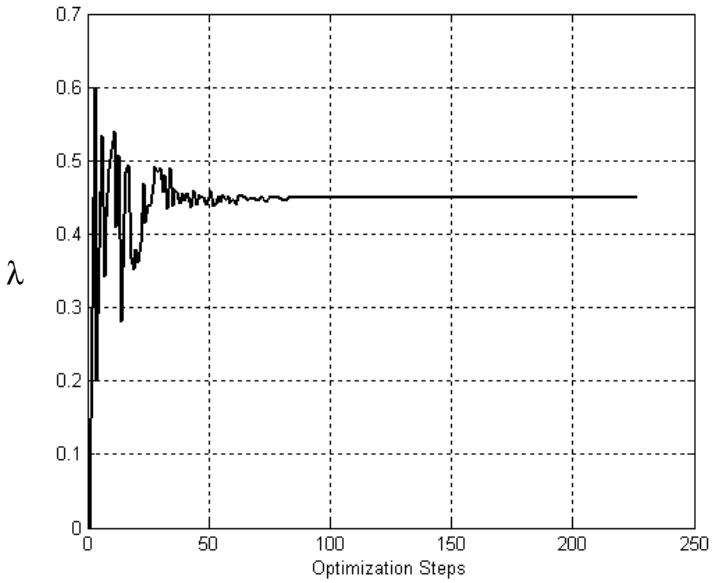


Figure 3.10 Changing value of λ during optimization steps

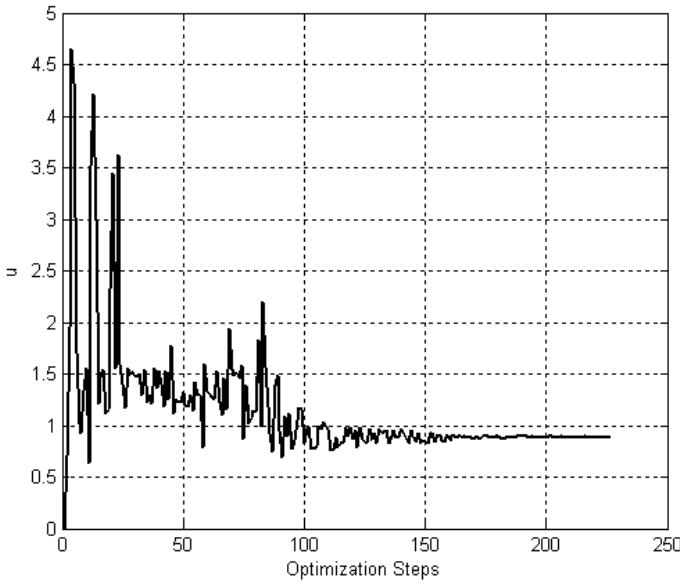


Figure 3.11 Changing value of μ during optimization steps

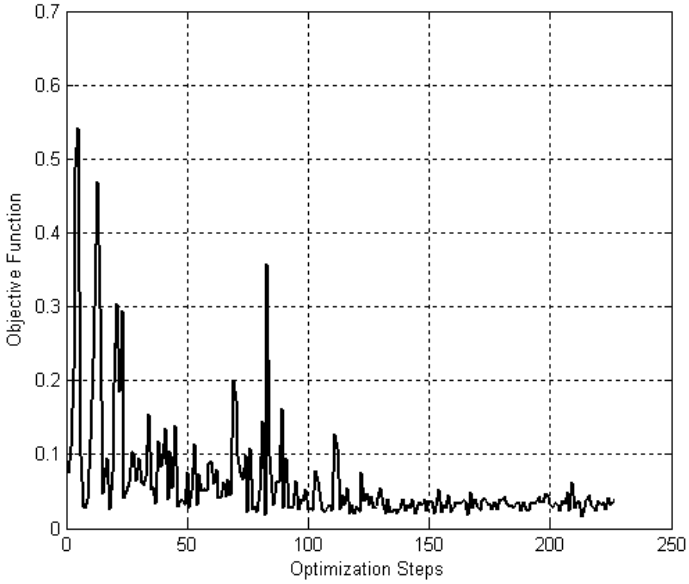


Figure 3.12 Objective function

The suspension deflection ($z_i - w_i$) and body acceleration ($\ddot{z}_c$) are used to evaluate the road handling and riding comfort of the riders, respectively. By supplying the control signal, just the vertical displacements of sprung mass (z_i) and body acceleration ($\ddot{z}_c$) will be reduced to minimum while the vertical displacements of unsprung masses (w_i) will remain as it is. When z_i decreases, the road handling performance will be improved, while the decreasing of body acceleration ($\ddot{z}_c$) means that the riding comfort is also improved.

Figures 3.13-3.19 illustrate the comparison between the time outputs response of controlled system with FOPID controller and the passive system, i.e. the system without controller. The road profiles are assumed to be random inputs. Those Figures show that the vertical displacements at each corner (at P_1, P_2, P_3, and P_4); the vertical displacement at center of gravity (at P_c); the rolling movement and the pitching movement have been improved when the FOPID controller are used. In another word, the riding comfort and road handling are become better.

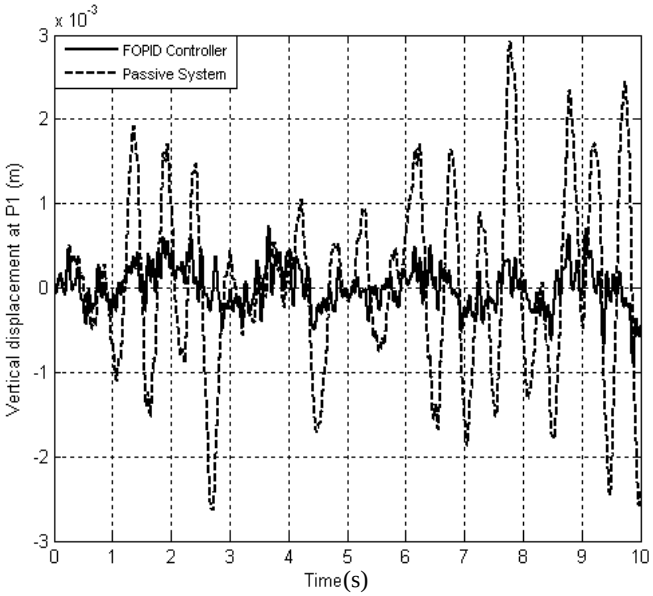


Figure 3.13 Time response of a vertical displacement at P_1

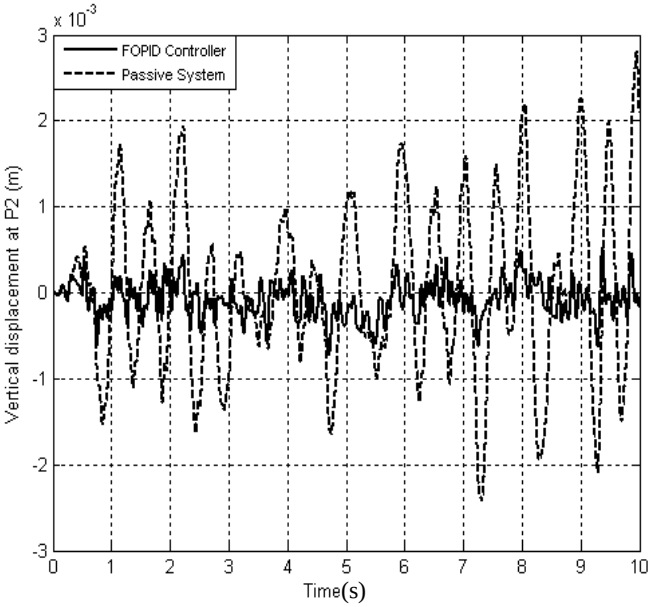


Figure 3.14 Time response of a vertical displacement at P_2

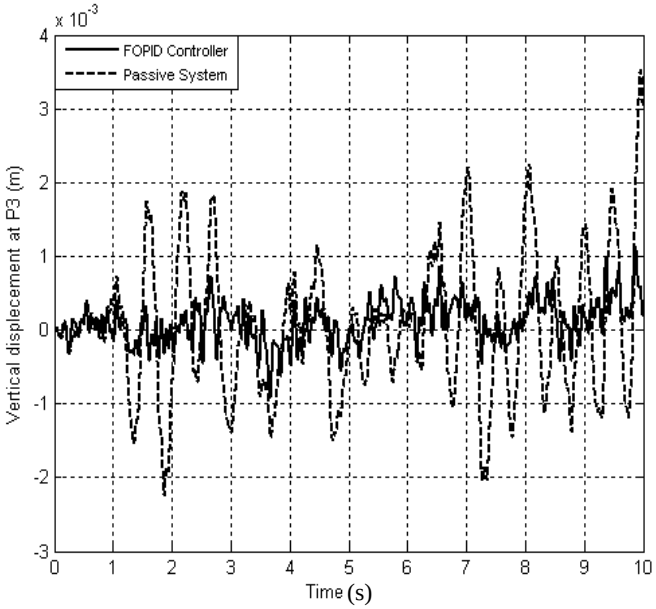


Figure 3.15 Time response of a vertical displacement at P_3

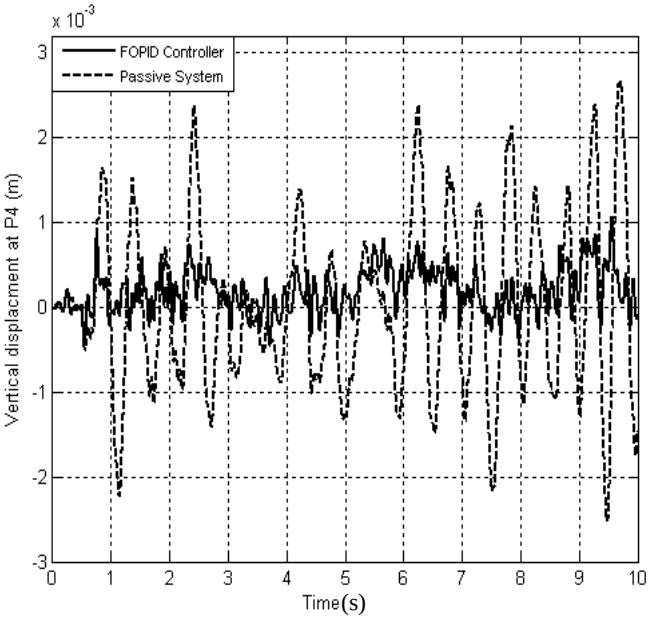


Figure 3.16 Time response of a vertical displacement at P_4

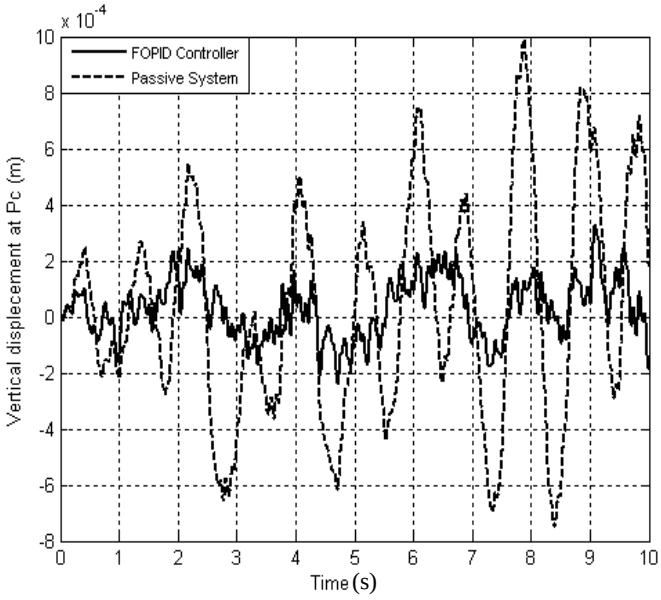


Figure 3.17 Time response of a vertical displacement at P_c

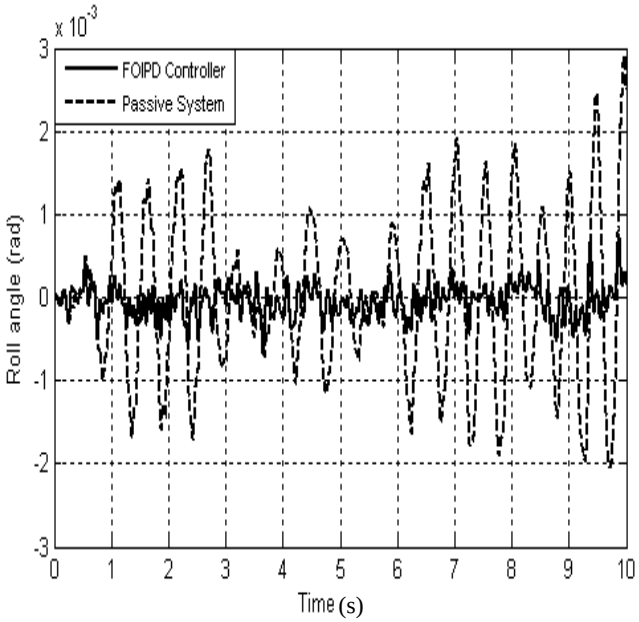


Figure 3.18 Time response of roll angle

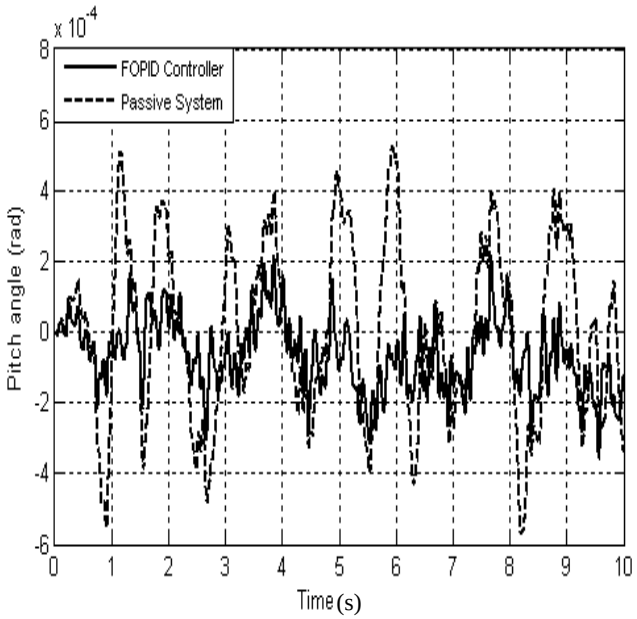


Figure 3.19 Time response of pitch angle

A bending torque has been applied to simulate the cornering of the vehicle (Figure 3.20). The road profiles are assumed to be random input. Figures 3.21-3.27 show the comparison between the time outputs response of controlled system with FOPID controller and passive system.

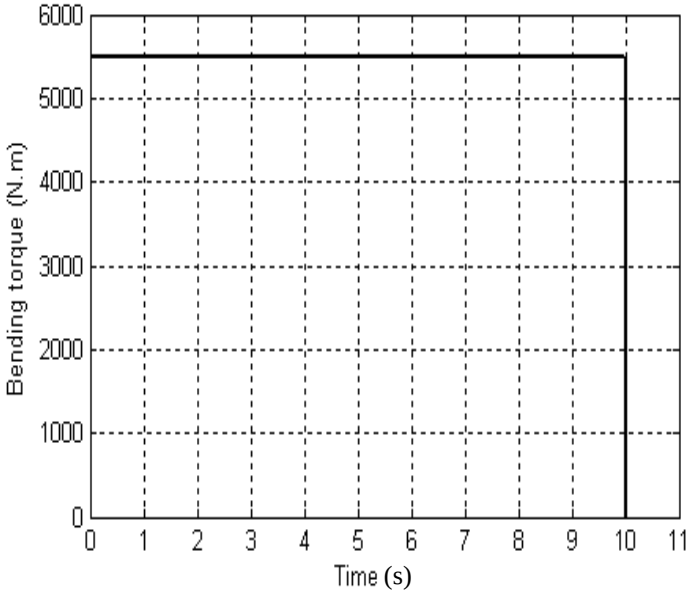


Figure 3.20 Bending torque

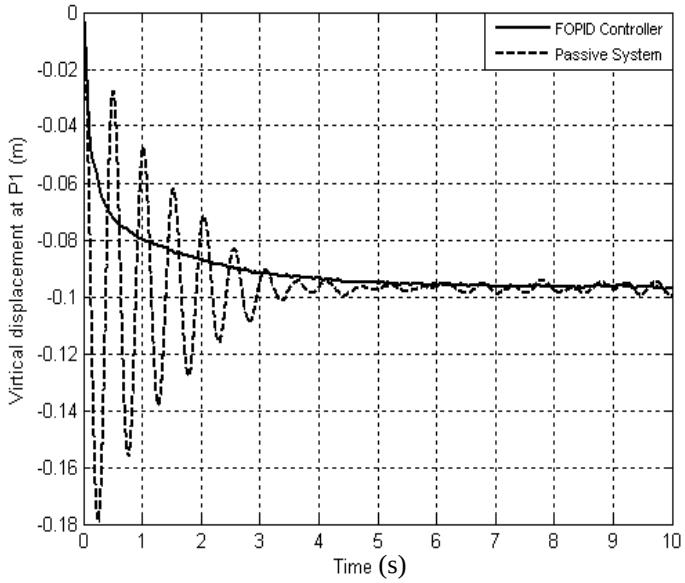


Figure 3.21 Time response of a vertical displacement at P_1

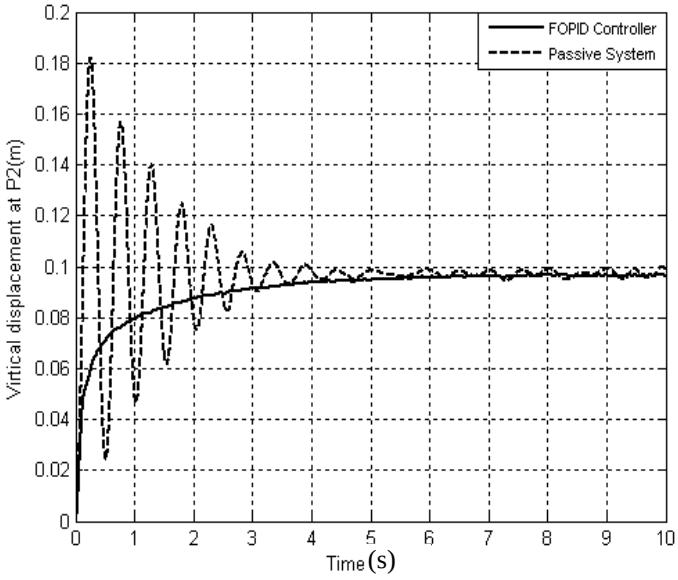


Figure 3.22 Time response of a vertical displacement at P_2

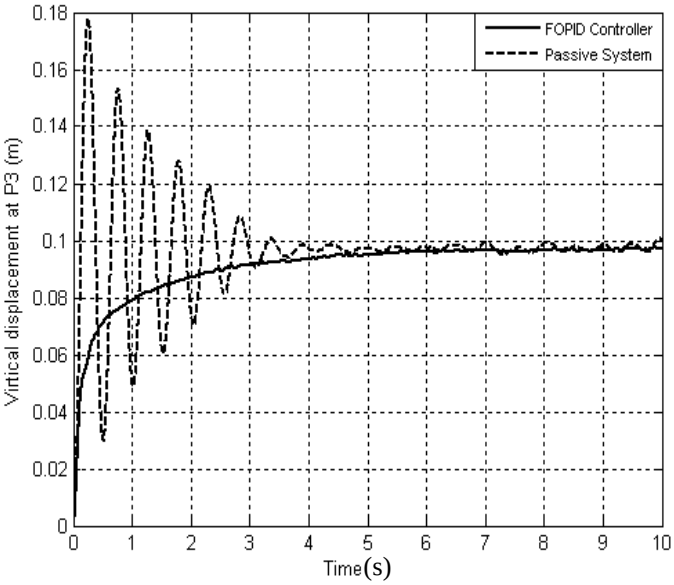


Figure 3.23 Time response of a vertical displacement at P_3

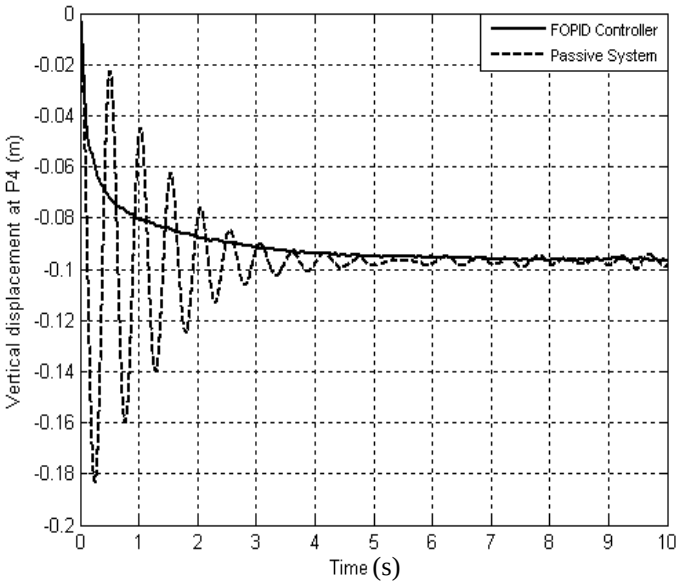


Figure 3.24 Time response of a vertical displacement at P_4

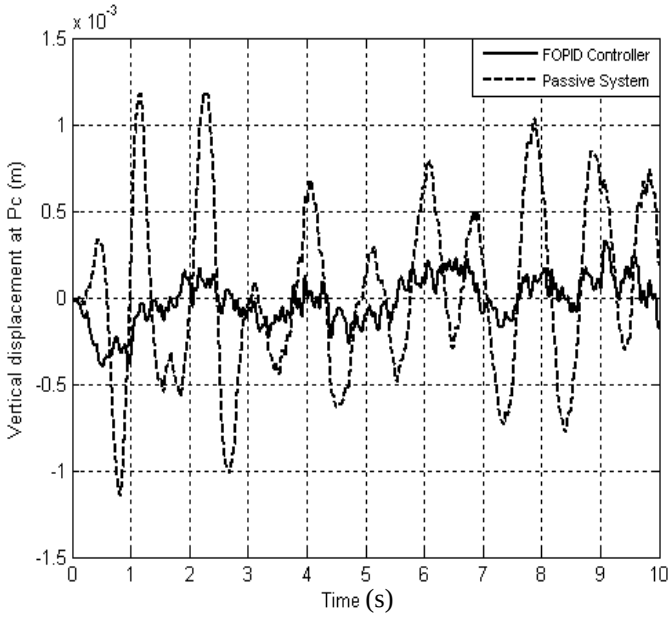


Figure 3.25 Time response of a vertical displacement at P_c

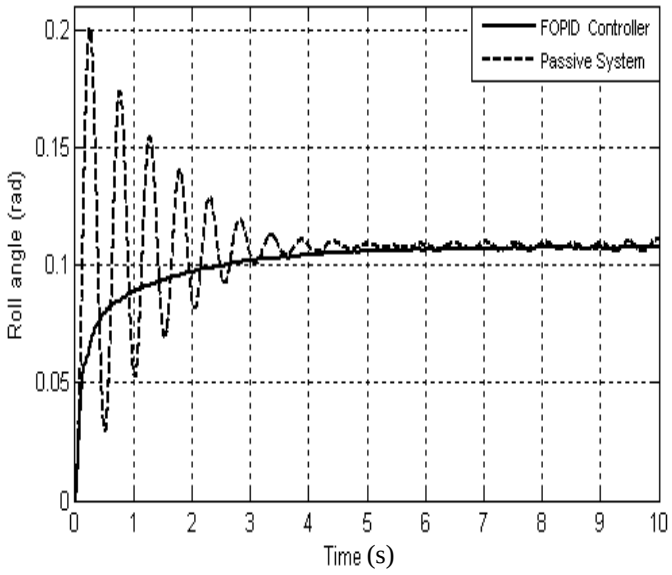


Figure 3.26 Time response of roll angle

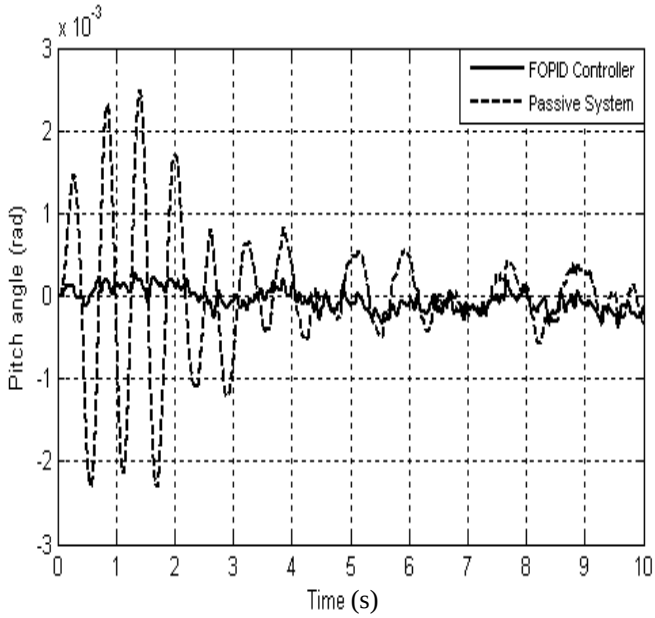


Figure 3.27 Time response of pitch angle

When the bending torque is applied, the outputs of the controlled system have been improved. When the FOPID controllers are used, the oscillations of the outputs of the controlled system have been omitted, but the final steady state values of the outputs remain the same.

To simulate the braking of the vehicle, a braking torque has been applied as shown in Figure 3.28. The road profiles have been assumed to be random input. Figures 3.29-3.35 show the comparison between the time outputs response of controlled system with FOPID controller and passive system.

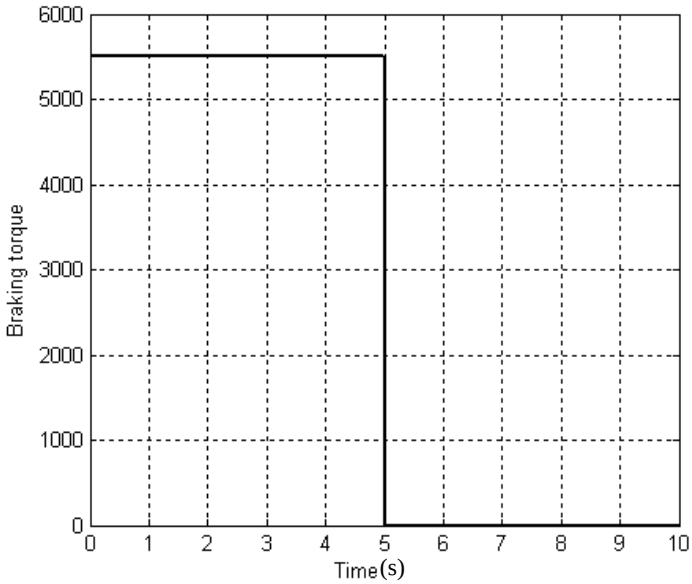


Figure 3.28 Braking torque

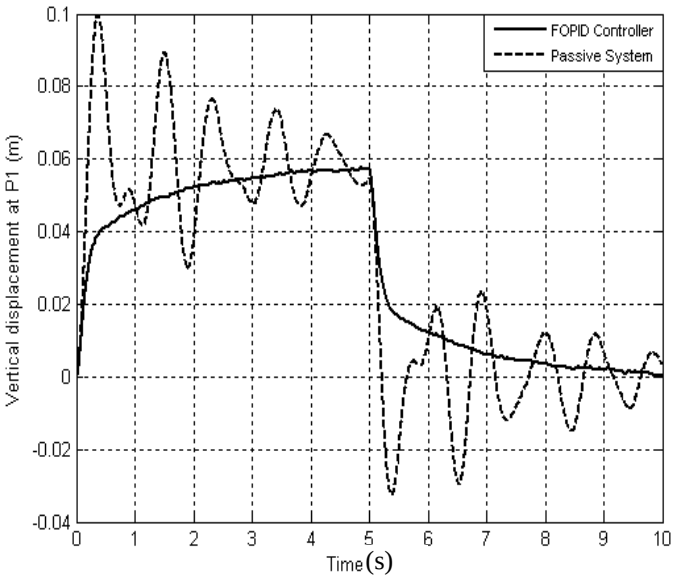


Figure 3.29 Time response of a vertical displacement at P_1

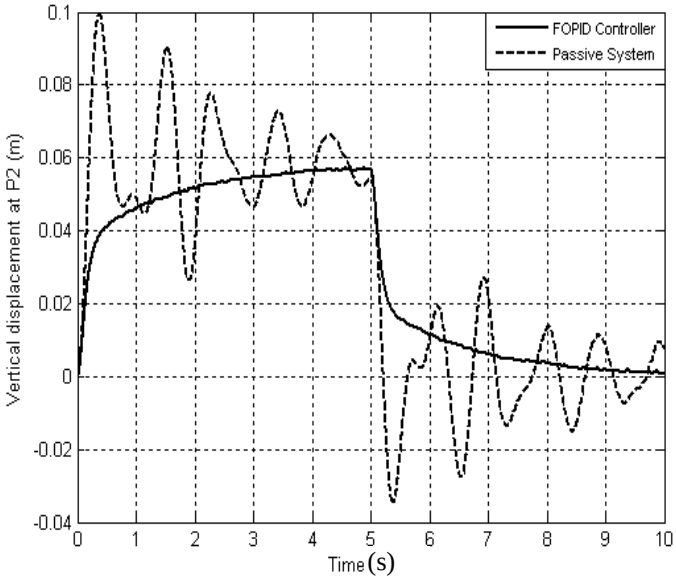


Figure 3.30 Time response of a vertical displacement at P_2

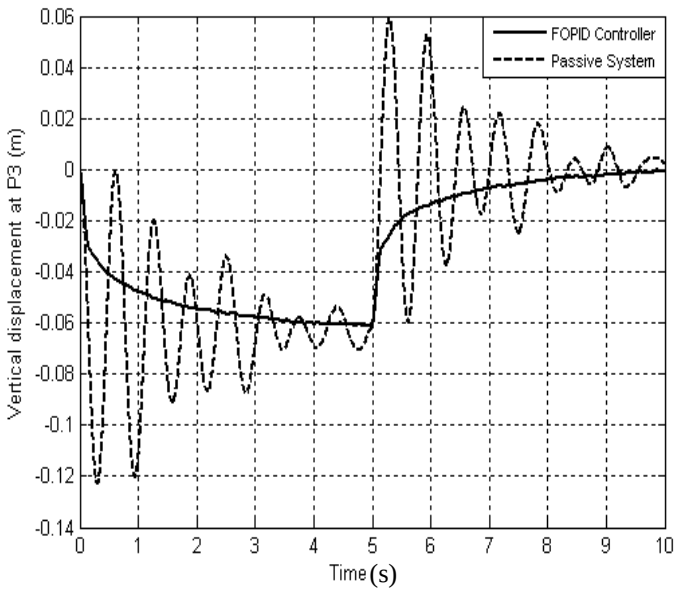


Figure 3.31 Time response of a vertical displacement at P_3

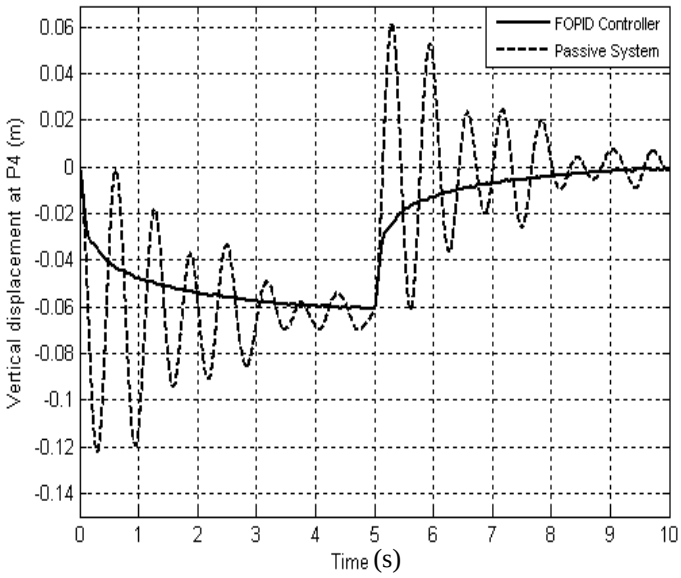


Figure 3.32 Time response of a vertical displacement at P_4

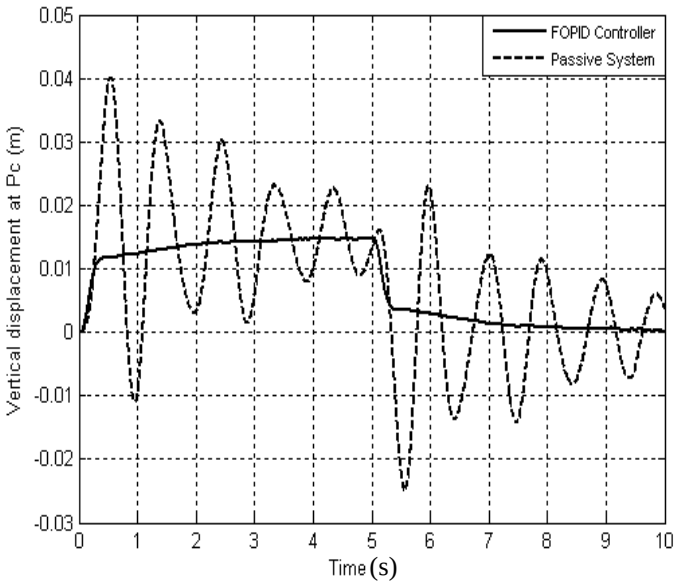


Figure 3.33 Time response of a vertical displacement at P_c

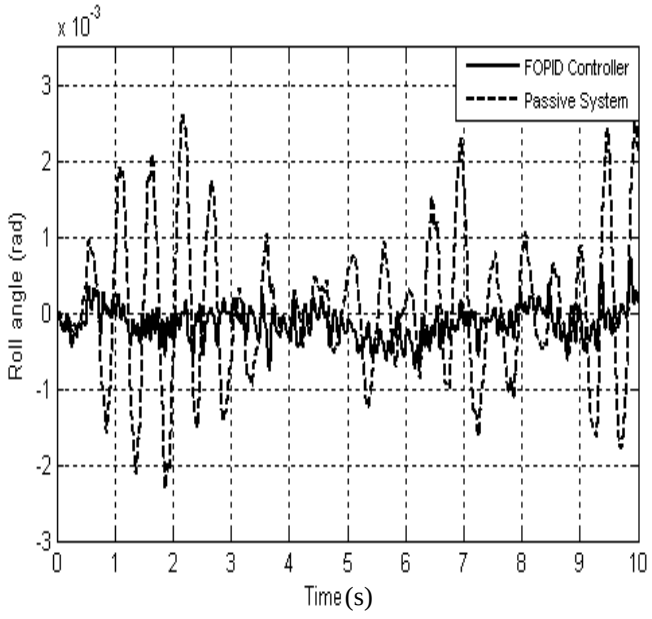


Figure 3.34 Time response of roll angle

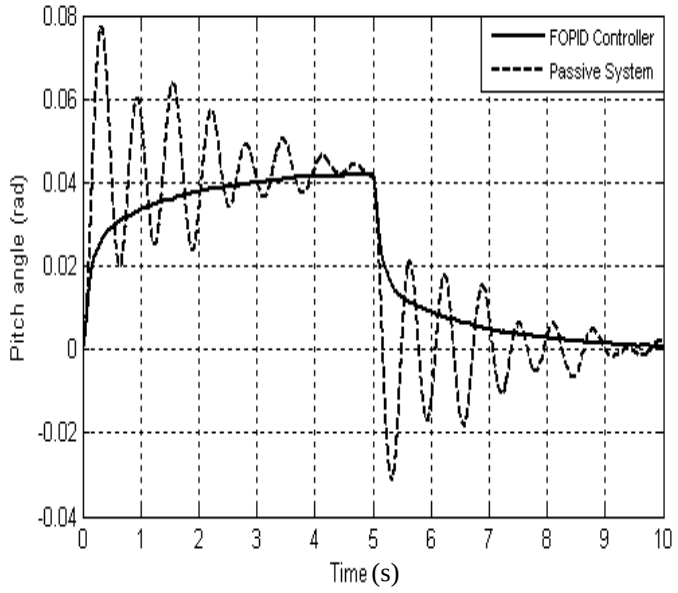


Figure 3.35 Time response of pitch angle

It can be seen that when the braking torque is applied, the outputs of the controlled system have been improved. When the FOPID controllers are used, the oscillations of the outputs of

the controlled system have been omitted, but the final steady states values of the outputs remain.

3.6 Robustness test of the FOPID controller

An efficient controller should be insenstive to disturbances signal which is applied on the plant. Therefore, the robustness of the FOPID controller should be examined to establish the effectiveness of the proposed controller. After obtaining the optimal values of the FOPID, the robustness of the proposed FOPID controllers with optimal values should be tested. Six types of disturbances are performed in turn to test the robustness of the FOPID controller.

- *Square wave with varying amplitude is applied as the input of road profile*

The square input signal is applied as road input with fixed frequency (1 Hz). Its amplitude is changed from 0.01m to 0.08m. At each value, the cost function (as described in Eq. 3.13) is calculated from

$$\phi = 0.5 \sum_{h=1}^{4} z_h^2 \tag{3.13}$$

Figure 3.36 shows the response of the cost function against the different amplitude values of square signal input.

- *Sine wave input signal with varying amplitude is applied as the input road profile*

Different amplitudes of sine wave input from 0.01m to 0.08m have been applied as the input of road profile with fixed frequency (1 Hz). The response of the cost function for the full vehicle without control and a result of controlled system with FOPID controller are shown in Figure 3.37.

Square wave with varying frequency is applied as the input of road profile

The square wave with fixed amplitude (0.03 m) has been applied as road profile. The frequency of the input signal is changed from 0.1Hz to 20Hz. The cost function has been calculated for each value of square input frequency. Figure 3.38 illustrate the cost function against different values of the square wave frequencies.

- *Square wave with varying frequency is applied as the input of road profile*

A sine wave input with different frequencies (from 0.1Hz to 20Hz) has been applied with fixed amplitude (0.03) to test the robustness of the proposed controller. Figure 3.39 shows the cost function against different values of the sine wave input frequencies.

- *Different values of bending inertia torque (T_x) are applied.*

The values of bending torque (from 1000 Nm to 9000Nm) with random signal as road profile has been applied. The cost function response is plotted as function of *Tx* in Figure 3.40.

- *Different values of breaking inertia torque (T_y) are applied*

The values of breaking torque (from 1000 Nm to 9000Nm) with random signal as road profile are applied. The cost function response is plotted as function of *Ty* as in Figure 3.41.

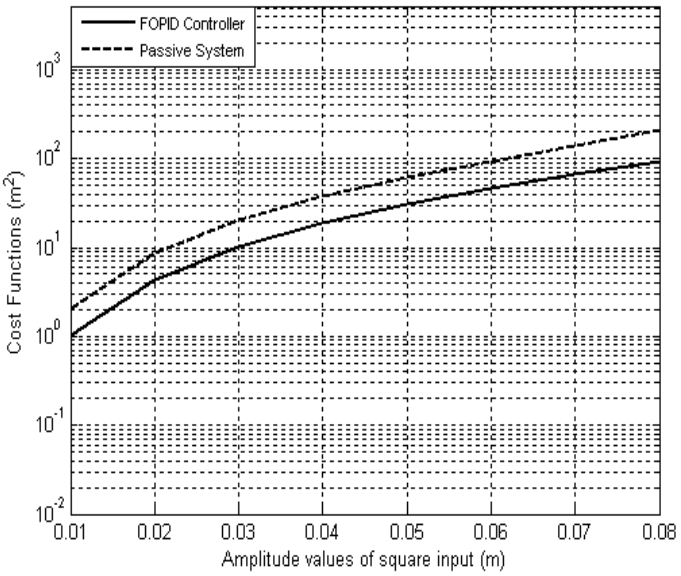


Figure 3.36 Time response of the cost functions against the different amplitude of square wave.

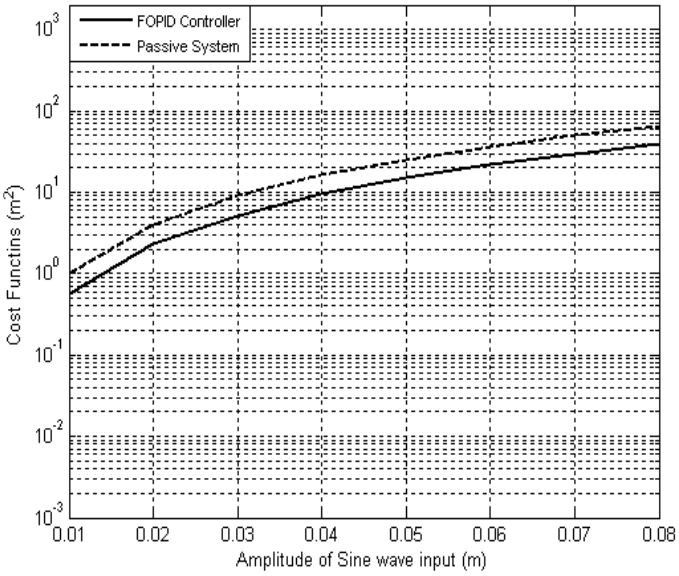


Figure 3.37 Time response of the cost functions against the different amplitude of sine wave.

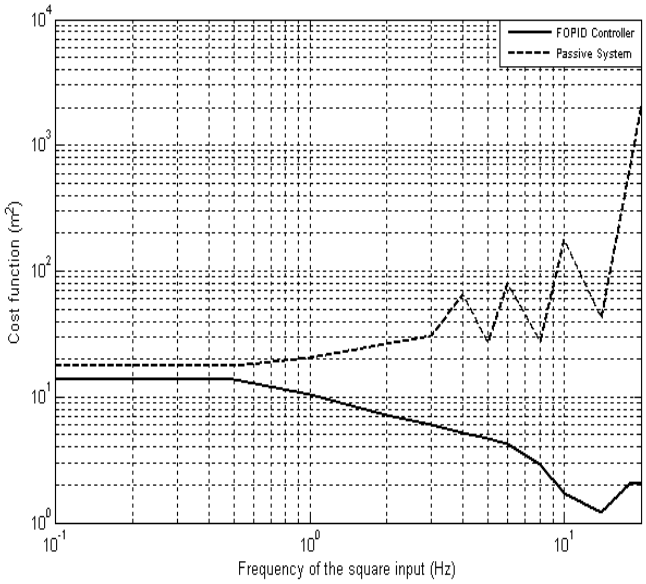


Figure 3.38 Time response of the cost function against different values of square wave frequencies

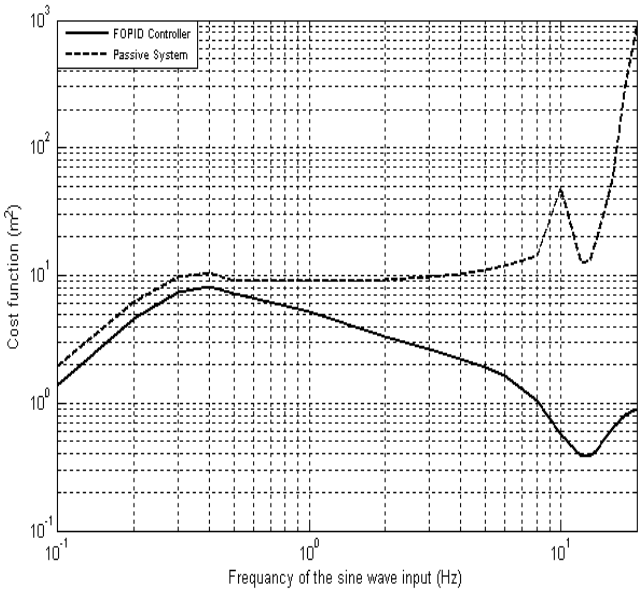


Figure 3.39 Time response of the cost function against different values of sine wave frequencies

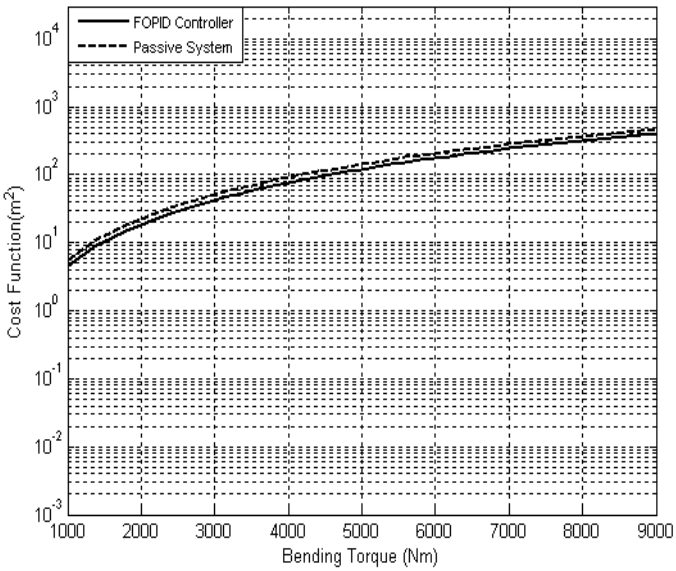


Figure 3.40 Time response of the cost functions against bending torque (T_x)

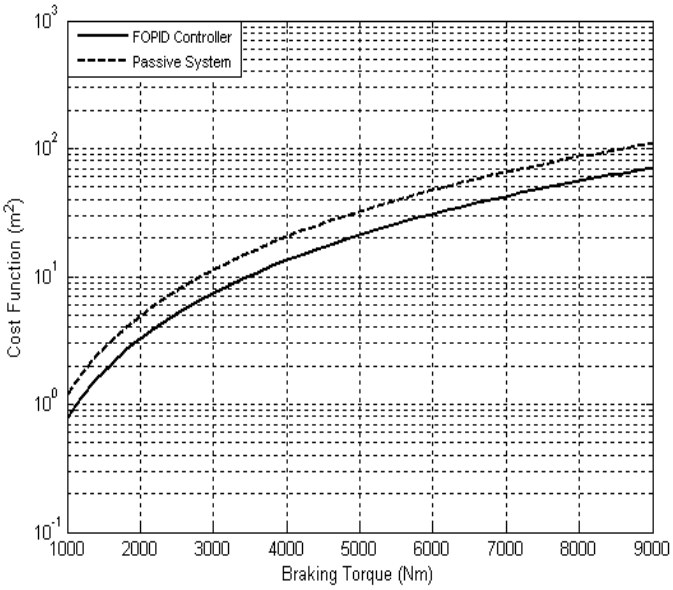


Figure 3.41Time response of the cost functions against breaking torque (T_y)

From the above results, it is clear that when the FOPID controllers are used, the cost functions of the controlled system are slightly improved. Only the response of the cost function against different values of the bending torque is still unchanged which means that when the different values of the bending torque are applied, the FOPID controller will be not a robust controller.

3.7 Summary

Fractional Order $PI^{\lambda}D^{\mu}$ controller has been developed to improve the performance of vehicle motion by minimizing the cost function. The evolutionary algorithm has been used to select the fitness parameters of the FOPID controller. The FOPID controller has been designed by using OptiY 4.0 software to control the nonlinear active suspension systems with hydraulic actuators under typical disturbances. Simulation results stated that the output transient responses of the controlled system with proposed controller are better than the outputs transient responses of the passive system. Nevertheless, the amplitude of the output transient responses of the controlled system with FOPID controller is not small (can be sensed by the

passengers) so that the control objectives can not be met totally. On the other hand, when the bending torque with random inputs as the road profile is applied (or the braking torque applied), the oscillation of the output responses of the controlled system are reduced to nearly zero, but the steady state outputs are still unchanged. Furthermore, when the six types of typical disturbances are applied, simulation results illustrated that the performances of the cost function for the system with the FOPID controller have been slightly improved while for the fifth disturbances type (different values of bending torque applied) the performance of the cost function of the system with proposed controller is close to the performance of the cost function of the passive system which means when different values of the bending torque are applied, the proposed controller will lose its robustness. Due to that the FOPID controller is not adaptive controller, i.e. it does not have the ability to modify its parameters when sudden disturbances occur, the adaptive controller should be suggested to design a robust controller for the full vehicle nonlinear active suspension system with nonlinear hydraulic actuators.

4. Fuzzy Model Reference Learning Control

4.1 Self-learning of fuzzy logic controller

The complexity and the nonlinearity inherently exist in most practical systems which present a big challenge for designing robust controllers for these systems (Buckley and Ying 1989). Knowledge based control more and more tries to integrate the knowledge of human operators or process engineers into the controller's design. Fuzzy control shows good performance for controlling nonlinear and uncertain systems that could not be controlled satisfactorily using conventional PID controller. The main advantages of the fuzzy logic controller (FLC) are: it can be applied to plants that are difficult to model mathematically and the controller can be designed to apply heuristic rules that reflect the experience of human experts. The drawbacks of designing the fuzzy controller depend on the experience of human experts are: it is difficult to select the parameters of the fuzzy system (the parameters of the input and output membership functions of fuzzy rules-based) and the fuzzy controller that is constructed for the nominal application may later perform inadequately if significant and unpredictable plant parameter variations occur. In this case, a fuzzy learning system is effective to overcome these difficulties. A fuzzy learning system possesses the capability to improve its performances over time. Therefore, the designed learning controller has the ability to improve the performance of the closed-loop system by generating command inputs to the plant and utilizing feedback information from the plant. Over the last two decades, some researchers

proposed adaptive algorithms to design the fuzzy system. In Reference Damousis et al. (2002), fuzzy logic system was used to simulate the induced electromagnetic filed problem. In order to create the rule base for the fuzzy logic system, a special incremental learning scheme is performed during the training phase. The genetic algorithm was used to train the fuzzy system. In Reference Chiang et al. (1997), a reinforcement learning technique was applied to learn fuzzy logic control rules. The proposed self-learning fuzzy logic control used the genetic algorithm through reinforcement learning architecture. Qiao et al. (1992) proposed a method to design fuzzy control rules where the rules are represented by an analytic expression with a regulating factor. Simplified fuzzy control algorithm was used in this reference to regulate the fuzzy control rules online. Kovacic et al. (1995) described a model reference based self-organizing fuzzy control scheme which is suitable for high-order systems of well-known structure. Kovacic et al. (1997) described a self-learning fuzzy logic controller (SLFLC) whose learning algorithm utilizes a second-order reference model and a sensitivity model. Fuzzy if-then rules with non-fuzzy singletons in the consequent part are adjusted by a gradient descent method in fuzzy systems (Ishibuchi et al. 1994). Gurocak and Lazaro (1994) proposed a method that treats the rule base of the fuzzy system as a multivariate function and performs parameter optimization to approach the desired rule base output with the given set of rules.

Some other researchers used the Fuzzy Model Reference Learning Controller (FMRLC) to determine the optimal values of fuzzy controller parameters, such as Chen and Teng (1995); Kamnik et al. (1998); Ismail (1998); Mayhan and Washington (1998) and Layne and Passino (1993). In Reference by Layne and Passino (1992), the learning algorithm of the FMRLC is developed by synthesizing several basic ideas from fuzzy set and control theory, self-organizing control, and conventional adaptive control. Learning mechanism which observes the plant outputs and adjusts the output membership functions of the rules in a direct fuzzy

controller is applied so that the overall system behaves like a reference model (a model of how the designer would like the controlled system to behave). The reference model is employed to provide closed loop performance feedback for synthesising and modifying the parameters of the fuzzy controller.

In this chapter, the FMRLC has been used for full vehicle nonlinear active suspension systems with hydraulic actuators. The control forces between the body and wheel axles are generated by hydraulic actuators to achieve the objectives of both good riding performance and suitable road stability. The proposed fuzzy controller adjusts the hydraulic actuator force to minimize the vertical displacement at each suspension when travelling on rough roads and to reduce the tendency of vehicle to rollover during sharp manoeuvres such as breaking and cornering. Four fuzzy controllers will be trained and applied to individual actuators in the vehicle suspension system. To show the effectiveness and robustness of the proposed controller, comparison is made with the passive system for different types of the disturbances.

4.2 The fuzzy logic systems

A Fuzzy logic system, which was proposed by Lotfy Zadeh in 1965, emerged as a tool to deal with uncertain, imprecise, or qualitative decision-making problems. It is a static nonlinear mapping between its inputs and outputs. The following subsections show the general concept of the fuzzy logic system.

4.2.1 General definitions

To deal with fuzzy logic system, the following definitions should be introduced (Passino 1998).

- ***Definition 4.1 "Linguistic Variables"***

It is a constant symbolic description that is used to specify the rules for rule based of fuzzy system inputs and outputs. If the fuzzy system has two inputs (u_1 and u_2) and one output(y), the following linguistic variables (for example) "$\tilde{u}_1 =$ position error, $\tilde{u}_2 =$ velocity error and $\tilde{y} =$ control signal" can be used to describe u_1, u_2 and y, respectively.

- ***Definition 4.2 "Linguistic Values"***

Linguistic values are the symbolic domain of the linguistic variables. They are used to describe characteristics of the variables. In other words, the linguistic variable ($\tilde{u}_i$ or $\tilde{y}_i$) takes on the elements from the set of linguistic values denoted by $\widetilde{H}_i = \{\widetilde{H}_i^j : j=1, 2, \ldots, N_i\}$, where N_i is the total number of the linguistic values for each of linguistic variable. For example, if $\tilde{u}_1$ denotes the linguistic variable "position error", then it may assign $\widetilde{H}_1^1$ = "negative", $\widetilde{H}_1^2$ = " zero" and $\widetilde{H}_1^3$= "positive" so that $\tilde{u}_1$ has a value from $\widetilde{H}_1 = \{\widetilde{H}_1^1, \widetilde{H}_1^2, \widetilde{H}_1^3\}$.

- ***Definition 4.3 "Linguistic Rules"***

The linguistic rules of fuzzy system can be described by using ***If-Then*** form

If *premise* ***Then*** *consequent*

The inputs of the fuzzy set are associated with the premise (sometime called antecedents) and the outputs are associated with the consequent (sometime called action). It is considered that the MIMO form of rule can be divided into a number of MISO rules using simple rules from logic. Therefore, the general form of the linguistic q^{th} rule can be described by

R_q: **If** $\tilde{u}_1$ is $\tilde{A}_1^{j,q}$ **and** $\tilde{u}_2$ is $\tilde{A}_2^{k,q}$ **and**… **and** $\tilde{u}_n$ is $\tilde{A}_n^{l,q}$ **Then** $\tilde{y}_s$ is $\tilde{B}_s^{r,q}$

If the fuzzy system has just one output, then the total number of the possible rules which can be written for any problem depends on: the number of fuzzy system inputs and number of linguistic values for each linguistic variable. The total number of the rules in the fuzzy inference system can be calculated as: $N_r = \prod_{i=1}^{n} N_{mi}$, where N_{mi} is number of linguistic

value of i^{th} linguistic variable and n is the number of the fuzzy system inputs. Notice that each premise can be composed of the conjunction of several terms (e.g. $\tilde{u}_1$ is $\tilde{A}_1^{j,q}$ **and** $\tilde{u}_2$ is $\tilde{A}_2^{k,q}$). The "**and**" operator is called conjunction operator. The conjunction operator can be any other logic operators such as: "**or**", "**not**", etc.

- ***Definition 4.4 "Membership Function"***

It is the function that maps the real numbers (crisps) of fuzzy system inputs to any numerical values between 0 and 1. The symbol $\mu_{A_i^{j,q}}(u_i)$ is used to describe the membership function associated with linguistic value $\tilde{A}_i^{j,q}$. There are many shapes can be chosen as membership function such as Sigmoid function, Gaussian function, Trapezoidal function, Triangular function, Bell-shape function, etc.

- ***Definition 4.5 "Fuzzy Sets"***

Given a linguistic variable $\tilde{u}_i$ with a linguistic value $\tilde{A}_i^{j,q}$ defined on the universe of discourse U_i and membership function $\mu_{A_i^{j,q}}(u_i)$ that maps U_i to [0,1], a "fuzzy set" denoted with $A_i^{j,q}$ is defined as

$$A_i^{j,q} = \{(u_i, \mu_{A_i^{j,q}}(u_i)) : u_i \in U_i\}$$

4.2.2 Fuzzy logic system structure

The fuzzy system consists of three main blocks: Fuzzification, Inference Mechanism and Defuzzification as shown in Figure 4.1. One of the famous fuzzy systems is the Mamdani fuzzy system which is constructed to work with crisp inputs and it takes one or more real value inputs and transforms them into fuzzy sets in the fuzzification part. These sets are then programmed to the inference system where the actual computation is performed.

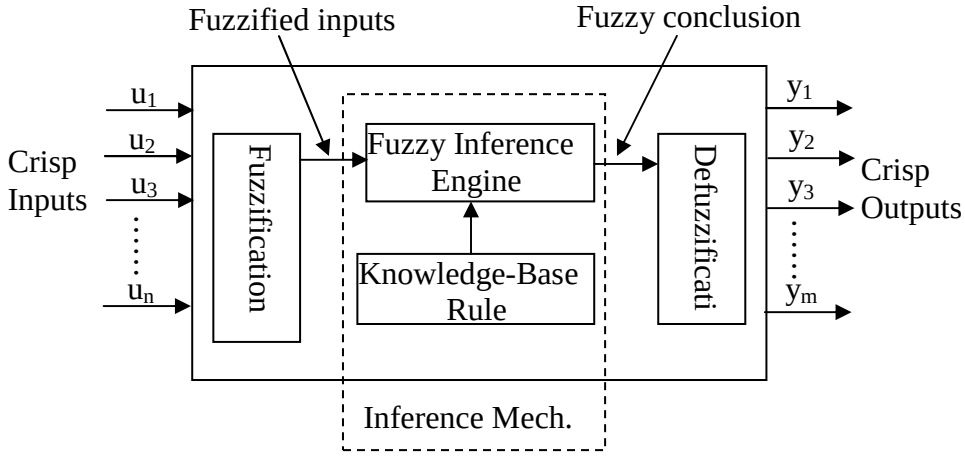


Figure 4.1 MIMO fuzzy logic system

The inputs and outputs of the fuzzy system are crisps (real numbers) not fuzzy sets. The fuzzification block transfer the crisp inputs (u_1, u_2…u_n) values to fuzzified inputs while the inference engine block uses the fuzzy rules from the rule-based block to generate fuzzy conclusions and the defuzzification block converts these fuzzy conclusions into the crisp outputs (y_1, y_2…y_m). These crisp outputs values represent the outputs of the fuzzy system. Each part of fuzzy logic system will be explained in detail below.

- ***Fuzzification***

Fuzzification is a mapping from the observed numerical input space to the fuzzy sets that are defined in the corresponding universes of discourse. The fuzzifier maps numerical values (universe of discourse) of any input denoted by (u_1, u_2, u_3, …, u_n) into fuzzy sets represented by membership functions. Therefore, the fuzzifier's duty is to transform (encode) the crisp valued inputs into fuzzy sets. The encoded information is then used in the fuzzy inference process.

- ***Inference mechanism***

Inference mechanism has two main tasks. In the first task, the premises of all rules are compared to the controller inputs u_i to determine which rules can be applied to the current

situation (this step is called *matching*). In the second task, the conclusions (related to control action) are determined using the active current rules (this step is called *inference*).

The main item to focus on is how to quantify the logical operation "**and**" in the premise part of the rules. There are several ways to defined the value of the premise part of q^{th} fuzzy rule ($\mu_{premise(q)}$) when the "**and**" operator is used.

1. *Minimum*: Define $\mu_{premise(q)} = min\ \{\mu_{A_1^{j,q}}(u_1), \mu_{A_2^{j,q}}(u_2), \dots, \mu_{A_n^{j,q}}(u_n)\}$.
2. *Product*: Defined $\mu_{premise(q)}=\mu_{A_1^{j,q}}(u_1) * \mu_{A_2^{j,q}}(u_2) * \dots * \mu_{A_n^{j,q}}(u_n)$.

The value of the conclusion part of q^{th} fuzzy rule ($\mu_{(q)}(y_s)$) can be described by using one of the following formulas (implication formulas)

1. *Minimum*: Define $\mu_{(q)}(y_s) = \min\{\mu_{premise(q)}, \mu_{\tilde{B}_s^{r,q}}(y_s)\}$.
2. *Product*: Define $\mu_{(q)}(y_s) = \mu_{premise(q)} * \mu_{\tilde{B}_s^{r,q}}(y_s)$.

where $\mu_{\tilde{B}_s^{r,q}}(y_s)$ is the output membership function that is corresponding to the current rule.

- ***Defuzzification***

It maps the fuzzy conclusion values defined over a universe of discourse to crisp outputs (converting decisions into actions). In other words, defuzzification operates on the implied fuzzy sets produced by the inference mechanism and combines their effects to provide the fuzzy output. It is employed because in many practical applications a crisp output is required. There are many approaches to perform the defuzzification such as

1. *The Center Of Gravity (COG) method*

   The center of area generates the center of gravity of the possibility distribution of the implication fuzzy output.

   Let b_q denotes the centre of the membership function (at which the membership function reaches the maximum value) of the consequent of q^{th} rule. The output of the fuzzy system can be calculated by

$$y_s^{crisp} = \frac{\sum_q b_q \int \mu_{(q)}(y_s)}{\sum_q \int \mu_{(q)}(y_s)} \tag{4.1}$$

where $\int \mu_{(q)}(y_s)$ is the area under the membership function $\mu_{(q)}(y_s)$.

2. The Center Average (CA) method

By using this method the output of the fuzzy system can be given by

$$y_s^{crisp} = \frac{\sum_q b_q \mu_{premise(q)}}{\sum_q \mu_{premise(q)}} \tag{4.2}$$

The fuzzy logic controller is usually based on the operator's knowledge. Therefore, it is difficult to determine the appropriate parameters of the fuzzy controller (constants of membership functions) which are used to generate the control action especially when there are changes in the parameters of the controlled process. In this case, the fuzzy controller will be improper controller unless its parameters are modified to overcome the changes in the plant parameters. In order to perform that, Fuzzy Model Reference Learning Control (FMRLC) has been proposed.

4.3 Fuzzy model reference learning control (FMRLC)

This type of controller combines two control technologies: the Fuzzy Logic Control (FLC) and the Model Reference Adaptive Control (MRAC). The FMRLC goes one step beyond direct fuzzy control by possessing the ability to improve the performance of the closed loop system. Conventional adaptive methods are just proposed to tune the controller parameters. In other words, most conventional adaptive control methods would need to re-tune a condition that it has already tuned for. In the Fuzzy Model Reference Learning Control, the word "learning" is used instead of "adaptive" because the FMRLC does not only tune the parameters of membership functions, but it remembers the values that has been tuned previously. Figure 4.2 illustrates the structure of fuzzy model reference learning control which

has three main parts: the Fuzzy Controller, Reference Model and Learning Mechanism. Each part of the FMRLC is described in detail in the following subsection.

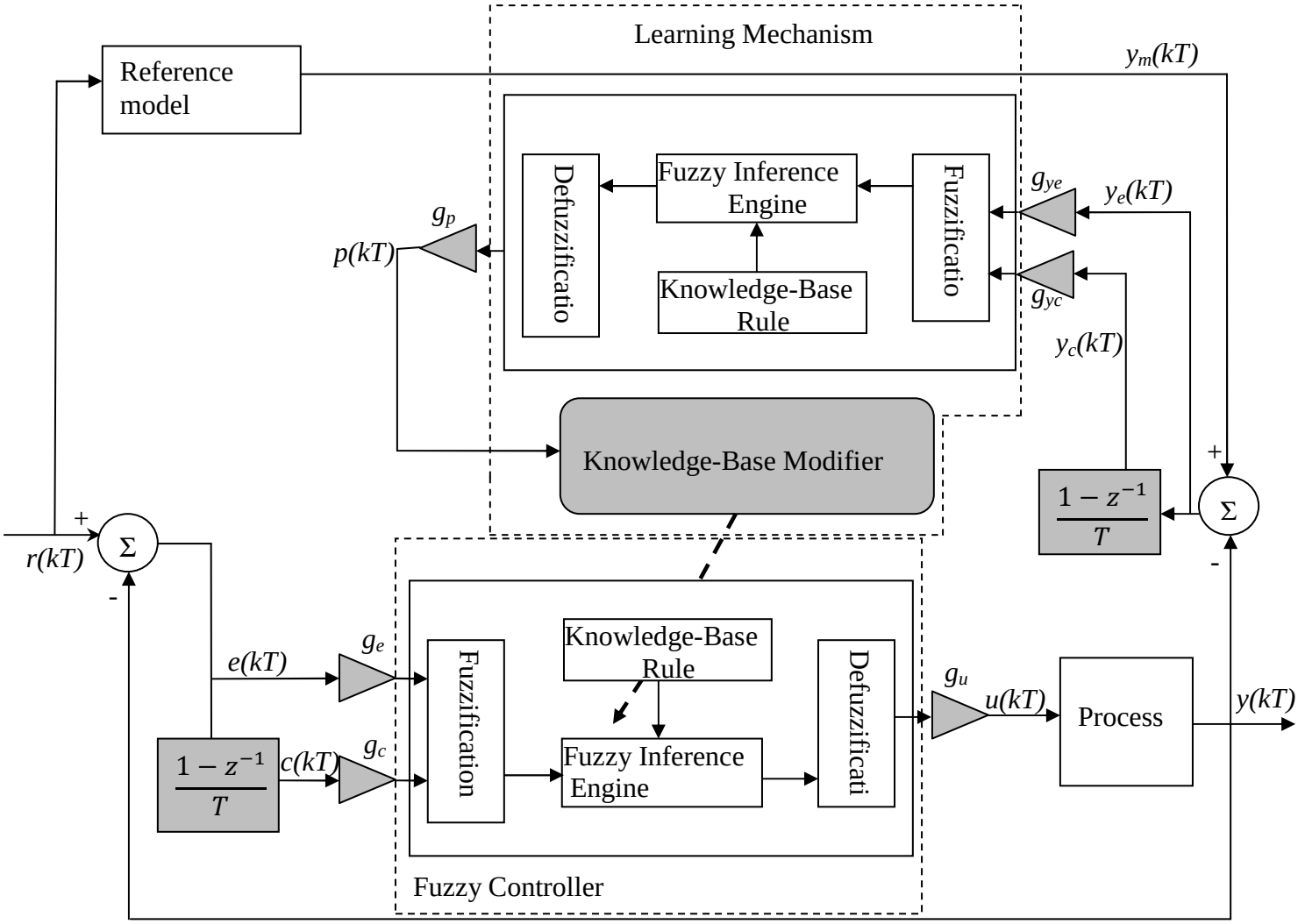


Figure 4.2 Architecture of FMRLC

4.3.1 The fuzzy controller

The fuzzy controller consists of three main parts: fuzzification, inference mechanism and defuzzification which are explained previously. As shown in Figure 4.2, the fuzzy controller has two inputs and one output. Discrete time equivalents will be used to help clarify how the signals flow through the controller. The inputs to the fuzzy controller are generated via some functions of the plant output *y(kT)* and reference input *r(kT)*. Therefore, the inputs to the fuzzy

controller are represented by the error *e(kT)* and the change in error *c(kT)* which are shown in Eq. 4.3 and 4.4 respectively.

$$e(kT) = r(kT) - y(kT) \tag{4.3}$$

$$c(kT) = \frac{e(kT) - e(kT-1)}{T} \tag{4.4}$$

The output of the fuzzy controller, i.e. the control signal input into the plant, is *u(kT)*. The signals error, change in error and controller output are scaled using the scaling gains *ge, gc* and *gu* respectively.

The rule base for the fuzzy controller has rule of the form

$$R_q: IF\ \tilde{e}\ is\ \tilde{E}_q^j\ and\ \tilde{c}\ is\ \tilde{C}_q^l\ Then\ \tilde{u}\ is\ \tilde{U}_q^r$$

where $\tilde{e}$ and $\tilde{c}$ denote the linguistic variables associated with controller inputs *e(kT)* and *c(kT)* respectively, $\tilde{u}$ denote the linguistic variables associated with controller output *u(kT)*, R_q denote the q^{th} rule of the fuzzy controller and $\tilde{E}_q^j$, $\tilde{C}_q^l$ and $\tilde{U}_q^r$ denote the j^{th}, l^{th} and r^{th} linguistic values of the q^{th} rule associated with $\tilde{e}$, $\tilde{c}$ and $\tilde{u}$ respectively. For example, if the total numbers of linguistic values for each input are five then the total number of the possible rules for this controller will be $5 \times 5 = 25$. The input membership functions are defined for the premises part to take into account all possible situations in which the rules should be applied. The parameters of the input membership function are not tuned by the learning mechanism. Whereas, the learning mechanism is just modifying the parameters of the output membership functions that are associated with the consequent part of the rule-base.

In this work, the triangular shape membership function has been used for both input and output linguistic values. The fuzzy controller's rule base can be initialized by either setting all centres of the output membership functions to zero or providing an initial guess as to how the controller should act. The CA defuzzification method has been applied to compute the output of the fuzzy controller.

4.3.2 The reference model

The reference model should be chosen to quantify the desired performance of the close feedback loop. In general the reference model could be any type of dynamical system, for example linear or non-linear, time-invariant or time-variant, discrete or continuous time etc. Given that the reference model characterizes design criteria such as stability, rise time, overshot, settling time, etc. The input to the reference model is the reference input *r(kT).* The desired performance of the controlled process is met if the error between reference model output ($y_m(kT)$) and plant output $y(kT)$ remains very small for all time such that $y_e(kT) = y_m(kT) - y(kT) \approx 0$. If the performance is achieved, the learning mechanism will not make significant modifications to the fuzzy controller. Otherwise, if $y_e(kT)$ is big, the learning mechanism must adjust the parameters of the fuzzy controller.

4.3.3 The learning mechanism

The main benefit of learning mechanism is to tune the rule of the direct fuzzy controller to achieve small value of $y_e(kT)$, i.e. to make the closed-loop system behaves like the reference model. The learning mechanism consists of two parts: a fuzzy inverse model and a knowledge-base modifier. The fuzzy inverse model produces the suitable outputs *p(kT)* to force the error $y_e(kT)$ to be zero. Knowledge-base modifier performs the function of modifying the fuzzy controller's rule-base to applied the required changes in the plant inputs control signal *u(kT).* Each of these parts is explained in detail as follows.

- *The fuzzy inverse model*

The fuzzy inverse model is a direct fuzzy logic system that is explained in Section 4.2. The inputs of the inverse model are the error $y_e(kT)$ and possible function of $y_e(kT)$ such as the rate of the error $y_c(kT) = \frac{1}{T}(y_e(kT) - y_e(kT - T))$ or any other closed loop system data and the output of this model is the process inputs *p(kT).* The main benefit of this model is to

generate a process signal *p(kT)* to press the plant output to follow the model reference output. As shown in Figure 4.2, the fuzzy inverse model contains normalizing scaling factors, namely g_{ye}, g_{yc} and g_p for each universe of discourse. Selection of the normalizing gains can impact the overall performance of the system. The rules form of the fuzzy inverse model can be written as

$$R_v: If\ \tilde{y}_e is\ \tilde{Y}_{e,v}^{j}\ and\ \tilde{y}_c\ is\ \tilde{Y}_{c,v}^{l}\ Then\ \tilde{p}\ is\ \tilde{P}_v^r$$

where $\tilde{y}_e$ and $\tilde{y}_c$ denote the linguistic variables associated with fuzzy inverse model inputs $y_e(kT)$ and $y_e(kT)$, respectively; $\tilde{p}$ denotes the linguistic variables associated with fuzzy inverse model output *p(kT);* R_v denoted the v^{th} rule of the fuzzy inverse model; $\tilde{Y}_{e,v}^{j}$, $\tilde{Y}_{c,v}^{l}$ and $\tilde{P}_v^r$ denoted the j^{th}, l^{th} and r^{th} linguistic values of the q^{th} rule associated with $\tilde{y}_e$, $\tilde{y}_c$ and $\tilde{p}$, respectively. Any membership function shapes can be assumed for the inputs and output universe of discourse. In this work a symmetric triangular membership functions have been assumed.

It is important to note that:

(a) The development of the fuzzy inverse model does not depend on the existence and specification of the mathematical model of the plant or its inverse, i.e. the plant inverse does not need to be existing.

(b) The fuzzy inverse model should not be confused with the mathematical model of the inverse of the plant that is sometimes used in the fixed control, i.e. the non-adaptive control.

- *The Knowledge-Base Modifier*

The knowledge-base modifier performs the function of modifying of the fuzzy controller so that the error $y_e(kT)$ will be very small. The knowledge-base modifier changes the rule-base of the fuzzy controller so that the previously applied control action will be modified by the amount *p(kT)*. Consider the previously computed control action is *u(kT-T)* which contribute to

the present good or bad system performance. Note that *e(kT-T)* would have been the processed error and the change in error *c(kT-T)* respectively at time *kT-T*. Likewise, *u(kT-T)* would have been the controller output at that time. In order to produced a small error ($y_e(kT)$), the fuzzy controller's knowledge-base should be tuned to produce a desired output *u(kT-T)+ ϱp(kT).* Then, the next time fuzzy controller has similar values for the error and change in error, control signal will reduce the error between the reference model and plant output.

By assuming that the symmetric output membership functions are used for the fuzzy controller, and b_m denote the centre of the output membership functions associated with $\tilde{B}_q^r$. Knowledge-base modification is performed by shifting centers of the membership functions (b_m) of the output linguistic value $\tilde{B}_q^r$ that are associated with the fuzzy controller rules that contributed to the previous control action *u(kT −T).*

To modify the parameters of the fuzzy controller, the following steps should perform:

1. Finding all the rules in the fuzzy controller whose premise certainty is greater than zero ($\mu_i(e(kT - T),c(kT - T)) > 0$) and those rules set is called "active set" which can be characterized by indices of the input membership functions of each rule that is on. Since all possible combinations of rules have been used, there will be one output membership function for each possible rule that is on.

2. The following equation should be used to modify the centres of fuzzy controller outputs membership function

$$b_m(kT) = b_m(kT - T) + \varrho p(kT) \tag{4.5}$$

where $b_m(kT)$ represents the centre of the m^{th} output membership function at the time *kT* and ϱ is defined as adaptive gain or learning rate. Rules that are not in the active set do not have their modified output membership functions. For the case, where the fuzzy controller has triangular membership functions in the premise part as shown in Figure 4.3, there will only be

at most four rules in the active set at any individual time. Then, only at the most of four output membership function centres are required to update via Eq. (4.5).

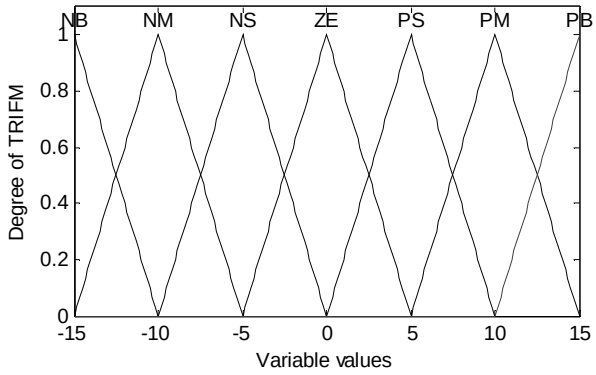


Figure 4.3 Degree of input membership functions

This type of learning is very valuable because the entire rule base does not need to be modified for every time step. Only the rules that have been applied to the current situations will be changed and stored. Therefore, when the plant returns to a familiar operating point, it does not need to re-adapt or re-tune again. The controller will already know how to handle the process. This kind of controller is worth for systems in which robustness issues are extremely relevant.

4.4 Using MATLAB program to implement the controlled system with FMRLC

The SIMULINK package in MATLAB program has many nonlinear toolboxes which can be used to simulate the linear and nonlinear systems. To modify the centre of the output membership functions of the fuzzy controller, the road profile inputs are assumed as a periodic square input with 0.01m amplitude and 0.3334 Hz frequency as shown in Figure 4.4. The Figure 4.5 illustrates the simulation blocks that have been used to implement the FMRLC with the controlled system at the training phase. The S-functions provide a powerful

mechanism for extending the capabilities of the Simulink environment. An S-function is a computer language description of a SIMULINK block written in MATLAB. This S-functions block (see Appendix 2.1) has been used to simulate the FMRLC. This program is used to tune the centre of the output membership functions of the fuzzy controller. Figure 4.5 shows that the inputs of i^{th} *FMRLC_S_Function* are (i) the error (e_i) between the derivative of the vertical displacement at i^{th} corner of the vehicle body and reference input and (ii) the error rate ($\dot{e}_i$). The output of i^{th} *FMRLC_S_Function* is the control signal u_{mi} for i^{th} hydraulic actuator.

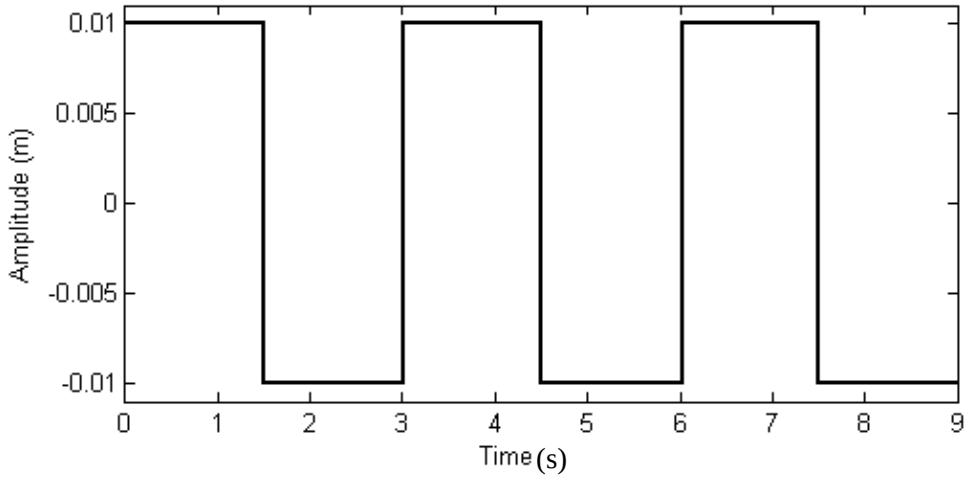


Figure 4.4 Road profile input of the first three periods

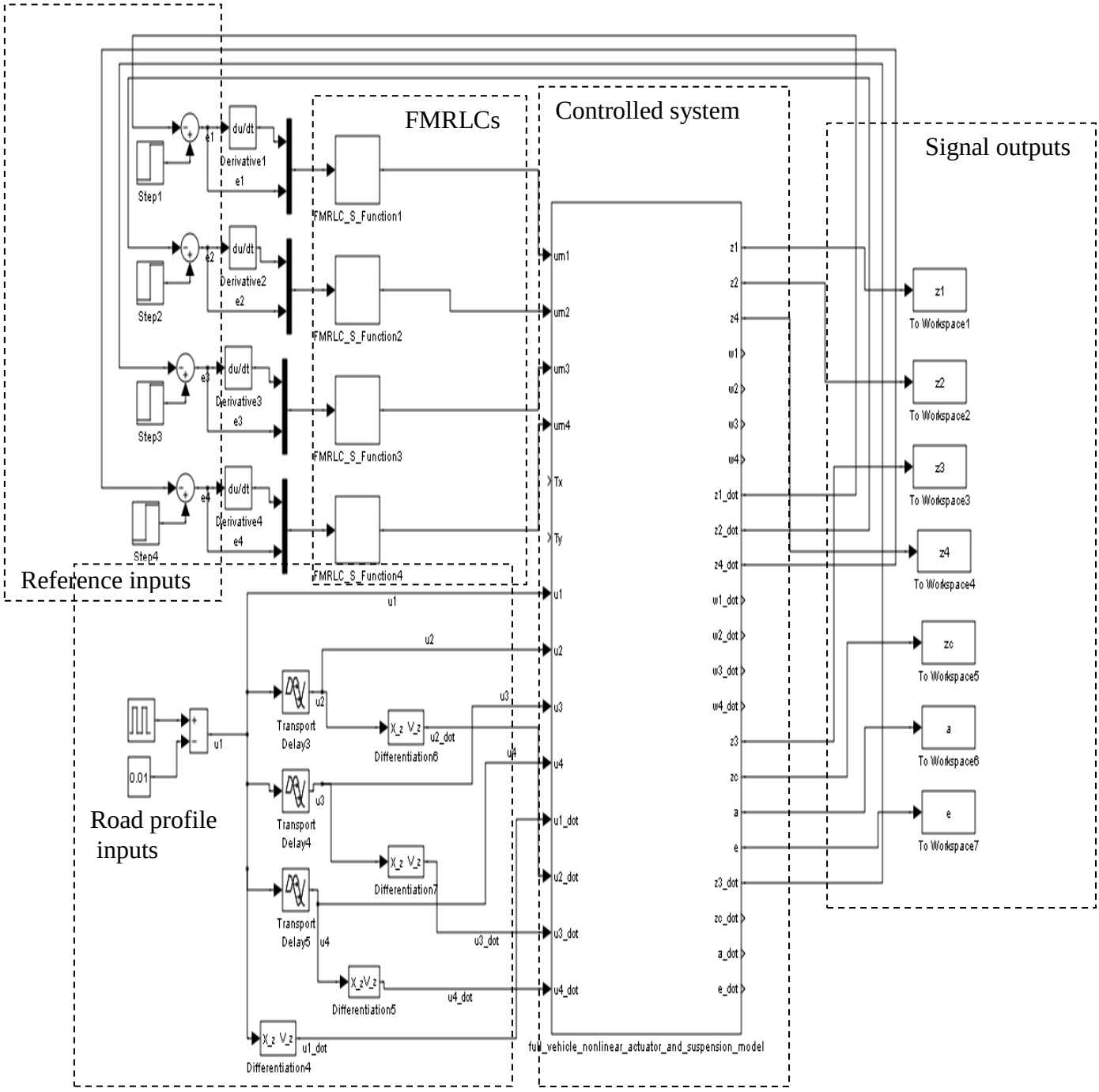


Figure 4.5 MATLAB Simulink model of FMRLC with the proposed system at training phase

4.5 Simulation and results

In this section, the FRMLC will be designed for the full vehicle nonlinear active suspension systems with hydraulic actuators, which was explained in Chapter 2. Four FRMLCs are designed, one for each suspension system. The S-function has been used to write the codes to train the FRMLC. The MATLAB/SIMULINK programme package has been used to simulate the training phase of the FRMLC with the controlled system. The reference model has been selected to be a unity block because the reference input is similar to the desired output. Figure 4.6 shows the FRMLC with controlled system at training phase.

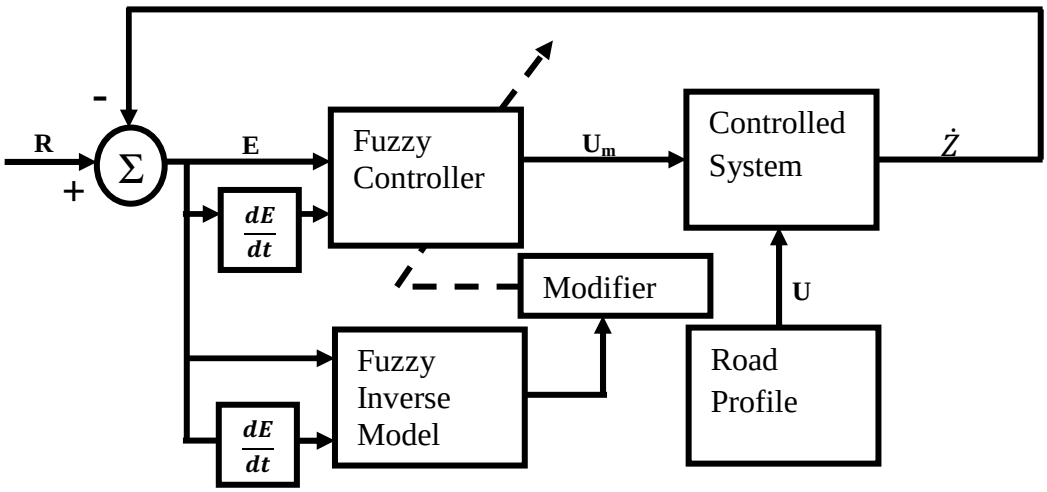


Figure 4.6 FRMLC with controlled system at training phase

In Figure 4.6, the reference input vector R is defined as $R = [r_1\ r_2\ r_3\ r_4]$, where r_i is the desired signals of $\dot{z}_i$ ($i \in \{1, 2, 3, 4\}$). The input vectors to the FMRLC is E $(R - \dot{Z})$ and $\dot{E}$. The output vector, i.e. control input vector, of the FRMLC U_m is defined as $U_m = [u_{m1}\ u_{m2}\ u_{m3}\ u_{m4}]$. By supplying the control input (u_{mi}), the spool-valve of the i^{th} hydraulic actuator is moved by x_{vi} units. Therefore, the hydraulic actuator will generate a suitable force to improve the performance of the controlled system. The output vector of the controlled system $\dot{Z}$ is the derivative of vertical displacement at each corner of the vehicle body, and is defined as $\dot{Z} = [\dot{z}_1\ \dot{z}_2\ \dot{z}_3\ \dot{z}_4]$.

The direct fuzzy controller has two linguistic variables associated with seven linguistic values. The input membership functions are uniformly distributed on the normalized input universe of discourse. Figure 4.7 shows the triangular functions that have been selected as the input membership function for the inputs of fuzzy controllers and fuzzy inverse models. Another seven triangular membership functions are used for output and the locations of each membership function on the output universe of discourse are initialised randomly. The input membership function has seven grades: negative big (NB), negative medium (NM), negative small (NS), zero (ZE), and positive small (PS), positive medium (PM) and positive big (PB). The fuzzy rule table shown in Table 1 is developed bases on the if-then rules for the fuzzy controller. The total numbers of the rules for the fuzzy controller are $7 \times 7 = 49$. These rules have been taken from the behaviour of the FOPID controller which was designed in Chapter 3, i.e. the behaviour of the control output with respect to control inputs of the FOPID controller. Then the inference mechanism and the standard CA defuzzification method, which is discussed previously, are used to obtain the crisp outputs according to the learning mechanism. The crisp outputs are scaled to produce the control signals for the controlled system.

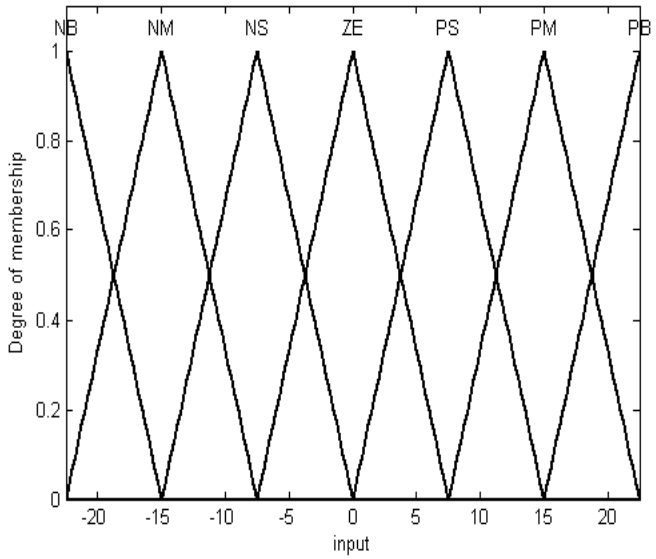


Figure 4.7 Input membership functions

u_m		$\dot{e}$						
		NB	NM	NS	ZE	PS	PM	PB
e	NB	NM	NS	NS	NS	ZE	PS	PM
	NM	NM	NM	NM	NS	PS	PM	PM
	NS	NB	NM	NM	NS	PM	PB	PB
	ZE	NB	NB	NM	ZE	PM	PB	PB
	PS	NB	NB	NM	PS	PM	PM	PB
	PM	NM	NM	NS	PS	PM	PM	PM
	PB	NM	NS	ZE	PS	PS	PS	PM

Table 4.1 The rule of fuzzy controller

The inputs of the fuzzy inverse model are represented by E and $\dot{E}$ vector. If the error between the reference input and the controlled system output is zero, the output of the fuzzy inverse model will be zero which means the learning mechanism is turned off. Otherwise, the fuzzy inverse model will modify the knowledge base of the fuzzy controller. Seven fuzzy sets are defined for inputs of fuzzy inverse model with triangular type membership functions uniformly distributed in the appropriate normalized universe of discourses. The linguistic values for each input of fuzzy inverse model can be defined as: NB, NM, NS, ZE, PS, PM, PB. The triangular shape has been used to describe the linguistic values of the fuzzy inverse model output. For simplicity, the rule base of the fuzzy inverse model is chosen similar to the rule base of the fuzzy controller. In the training phase, the scaling gains are assumed as follows: $g_{ye}=2$, $g_{yc}=2$ and $g_p=5$.

The knowledge base modifier changes the knowledge base of the direct fuzzy controller according to *p(kT)*. The CA defuzzification method is used to obtain the crisp output of the

fuzzy inverse model *p(kT)*. The changes are implemented by modifying the centres of the output membership functions in the direct fuzzy controller.

Figures 4.8 and 4.9 show the fuzzy rule surface viewer before and after the training phase, respectively.

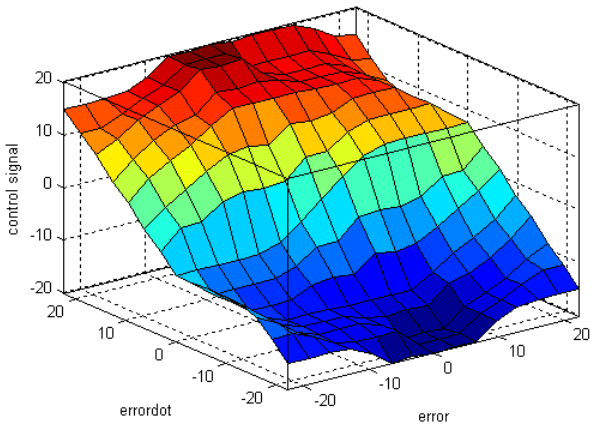


Figure 4.8 Fuzzy rule surface viewer before training

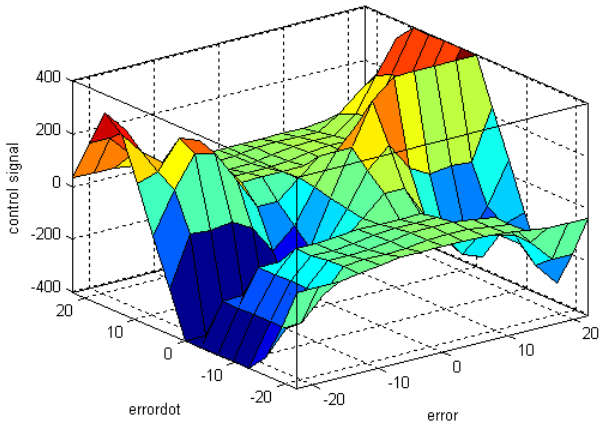


Figure 4.9 Fuzzy rule surface viewer after training

The adaptive gain ϱ has been selected to be equal to 0.015. The initial and the optimal values of centres of output membership functions of the fuzzy controller are shown in Table 2:

Initial values of b_m	-22.5	-15	-7.5	0	7.5	15	22.5
Optimal values of b_m	0	-35.115	-403.744	-0.197	403.547	34.918	0

Table 4. 2 The Initial and optimal values of b_m vector

The square input is used as the road profile with a range [-0.01 0.01] meters to perform the training task. After 47 iterations, the vertical displacement at the each corner of the vehicle body reaches to a small value.

Figures (4.10-4.17) illustrate the vertical displacements at each corner during training phase.

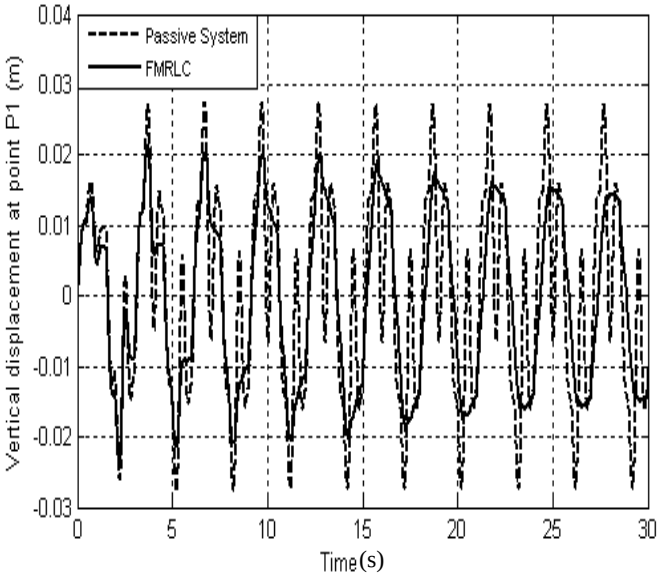


Figure 4.10 Vertical displacement at point P_1 for first 10 training periods

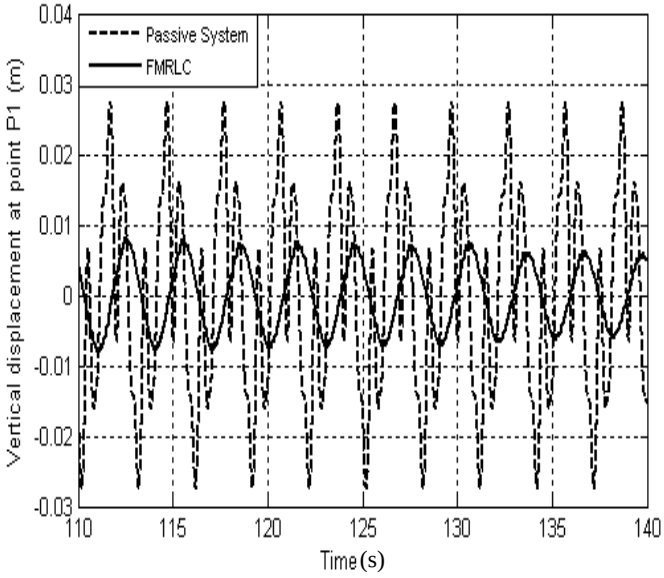


Figure 4.11 Vertical displacement at point P_1 for last 10 training periods

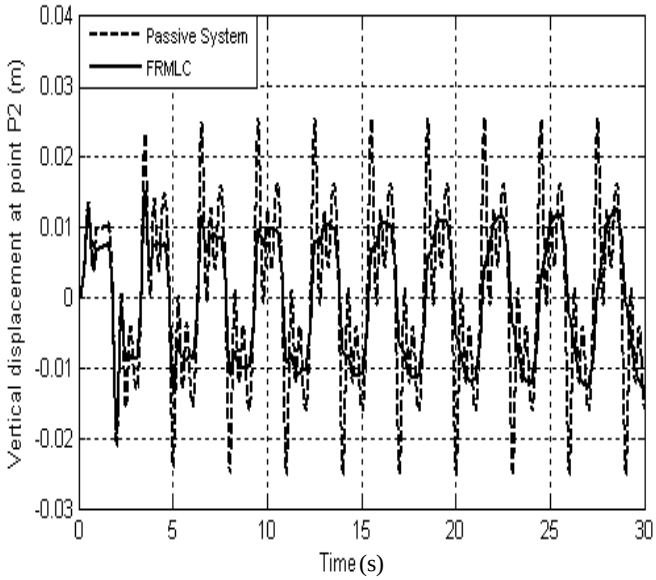


Figure 4.12 Vertical displacement at point P_2 for first 10 training periods

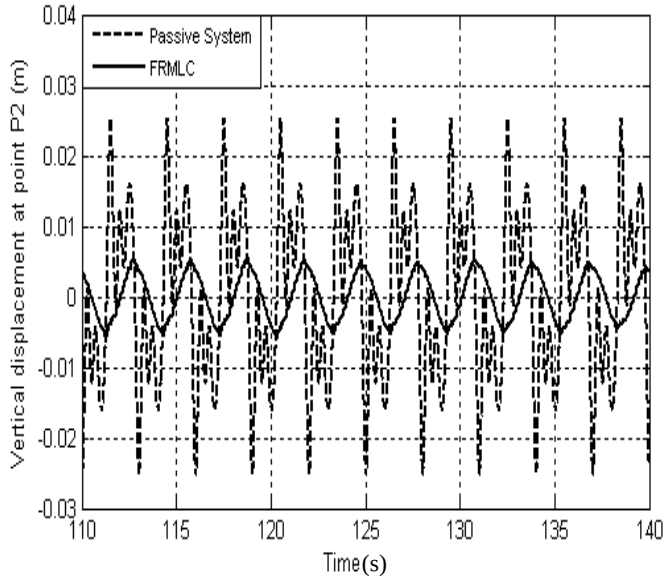


Figure 4.13 Vertical displacement at point P_2 for last 10 training periods

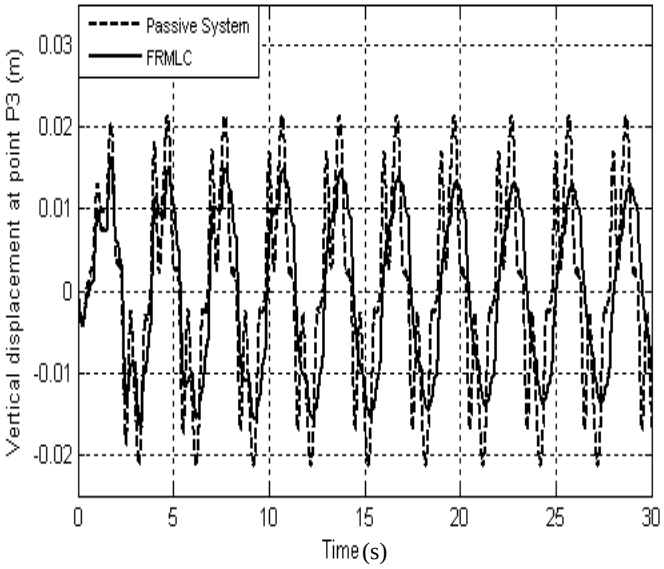


Figure 4.14 Vertical displacement at point P_3 for first 10 training periods

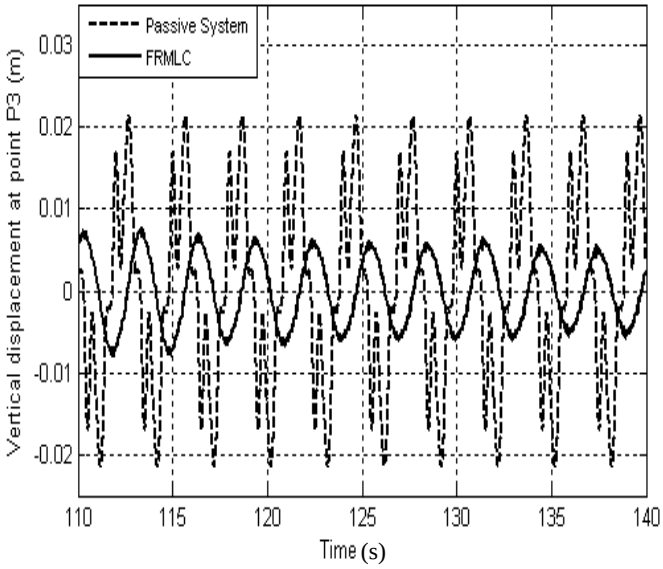


Figure 4.15 Vertical displacement at point P_3 for last 10 training periods

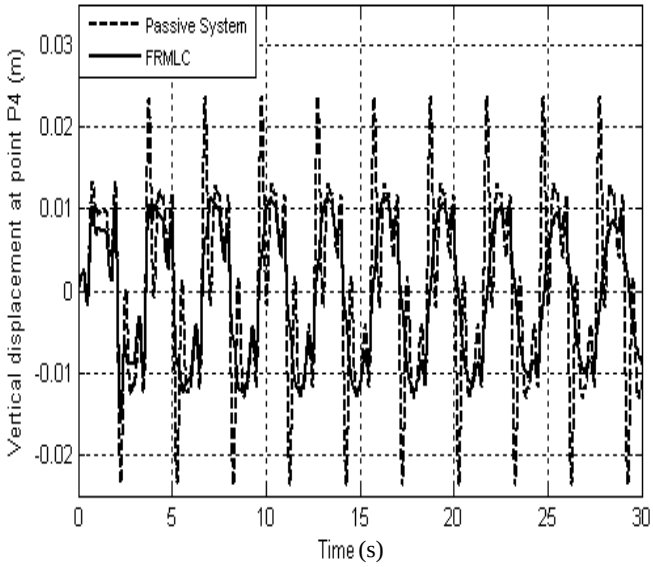


Figure 4.16 Vertical displacement at point P_4 for first 10 training periods

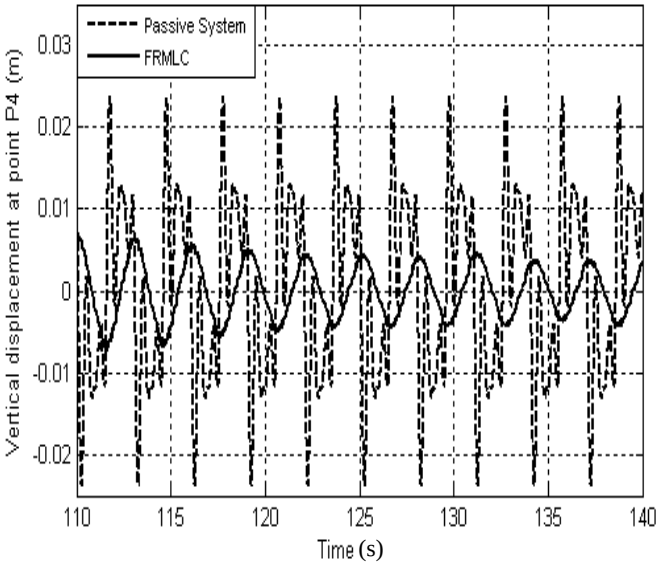


Figure 4.17 Vertical displacement at point P_4 for last 10 training periods

After the optimal values of the output membership function centres of the fuzzy controller have been obtained, the scaling gains (g_e, g_{ec} and g_u) must be tuned to improve the performance of the proposed controller. To select the optimal values of scaling gains, three-dimensional Golden Section Search (3-D GSS) method will be used. For more detail about

the 3-D GSS method, see Appendix 2.2. The optimal vlaes of the scaling gains are $g_e=55$, $g_{ec}=2$ and $g_u=10$.

The effectiveness of the proposed FMRLC with the optimal values should be examined. Figures 4.18-4.24 show a comparison between the outputs of controlled process with FMRLC and corresponding outputs of the passive system. In this case, the random input has been applied as a road profile.

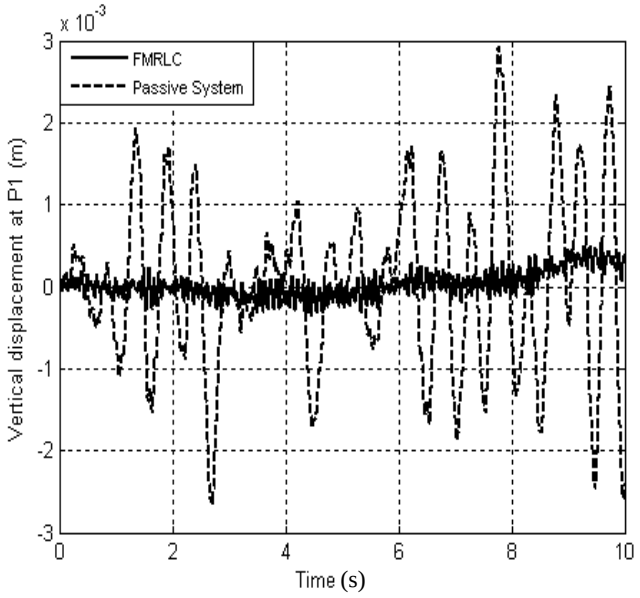


Figure 4.18 Time response of a vertical displacement at P_1

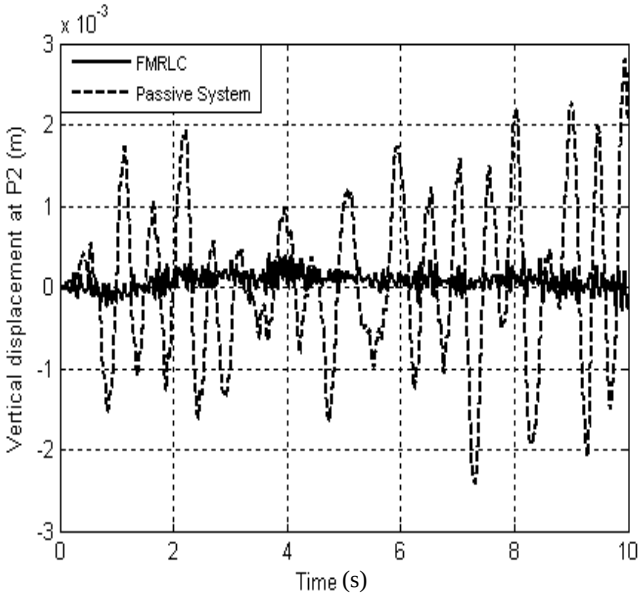


Figure 4.19Time response of a vertical displacement at P_2

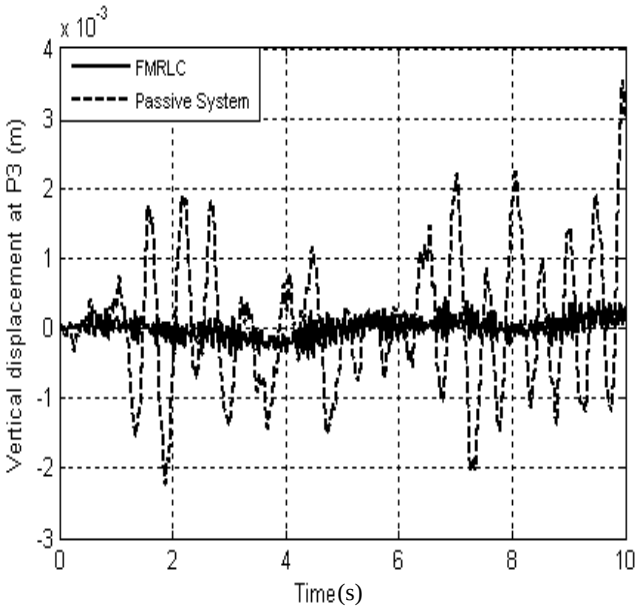


Figure 4.20 Time response of a vertical displacement at P_3

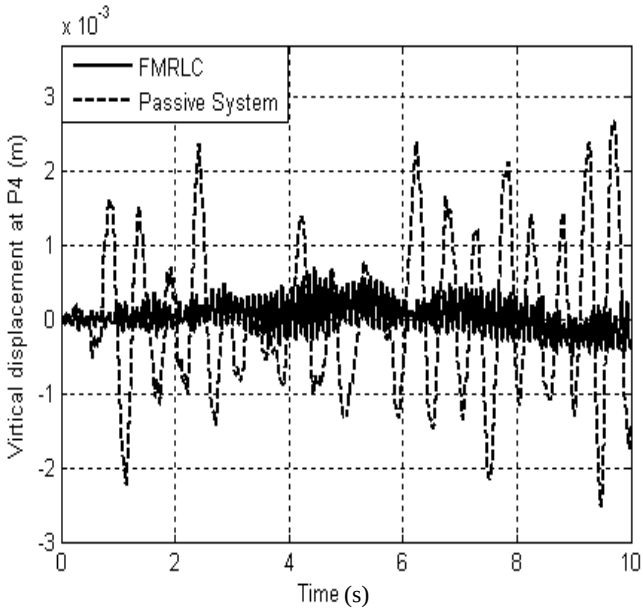


Figure 4.21 Time response of a vertical displacement at P_4

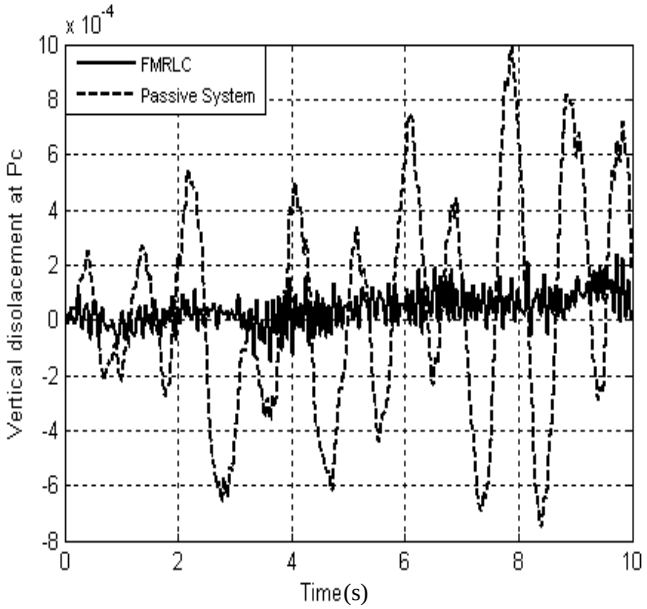


Figure 4.22 Time response of a vertical displacement at P_c

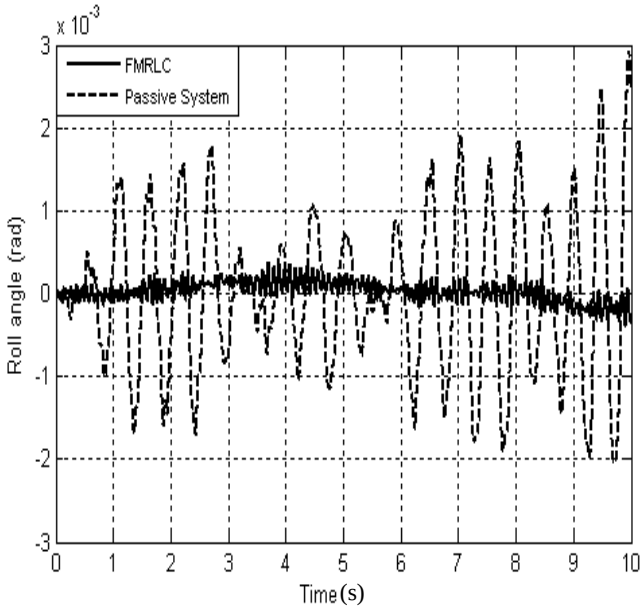


Figure 4.23 Time response of a roll angle

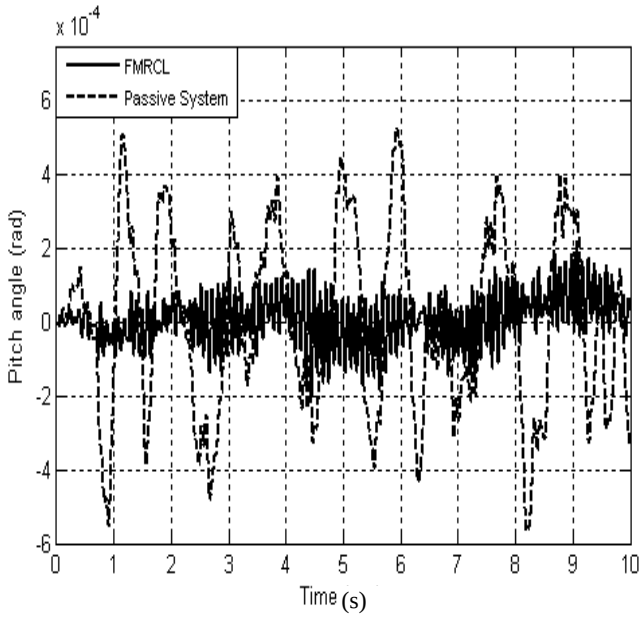


Figure 4.24 Time response of pitch angle

The Figures above illustrate that the outputs of the controlled system have been improved by using the fuzzy controller which means the vibrations felt by the passengers have been minimized and the contact forces between the road surfaces and the tyres have been maximized.

To test the effectiveness of the proposed controller during the cornering of the vehicle, a bending torque (shown in Figure 4.25) has been applied. Additionally, the random input has been chosen as the road profile. The Figures 4.26-4.32 show the comparison between the outputs of the controlled system with the FMRL controller, the outputs of the controlled system with FOPID controller and the outputs of the passive system.

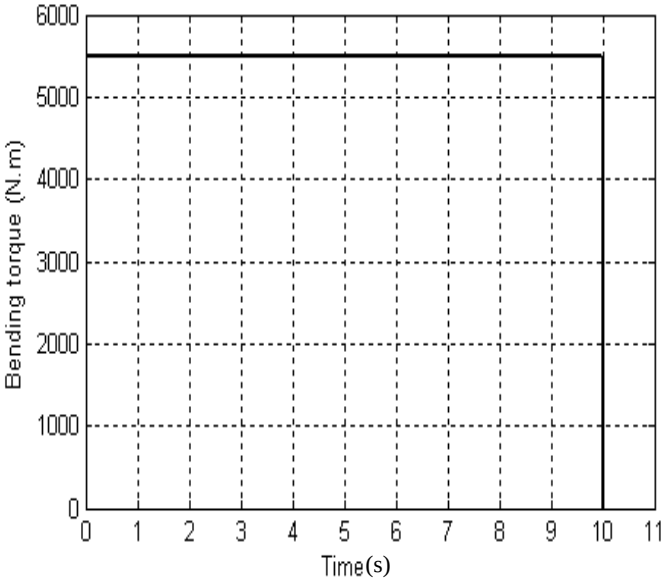


Figure 4.25 Bending torque

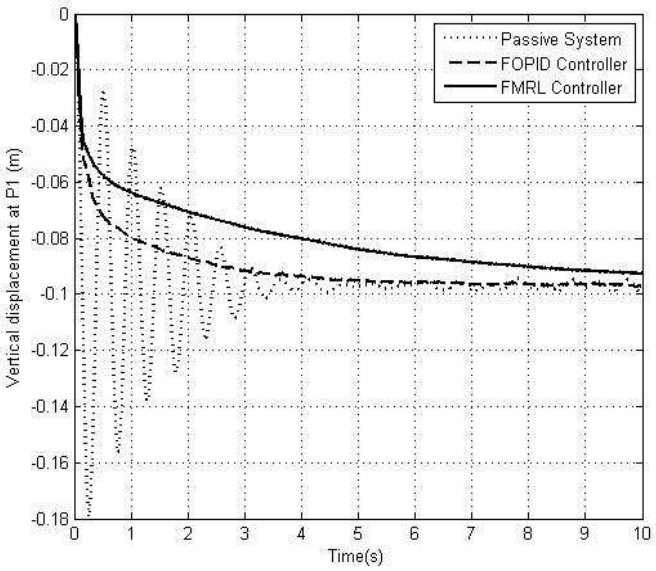


Figure 4.26 Time response of a vertical displacement at P_1

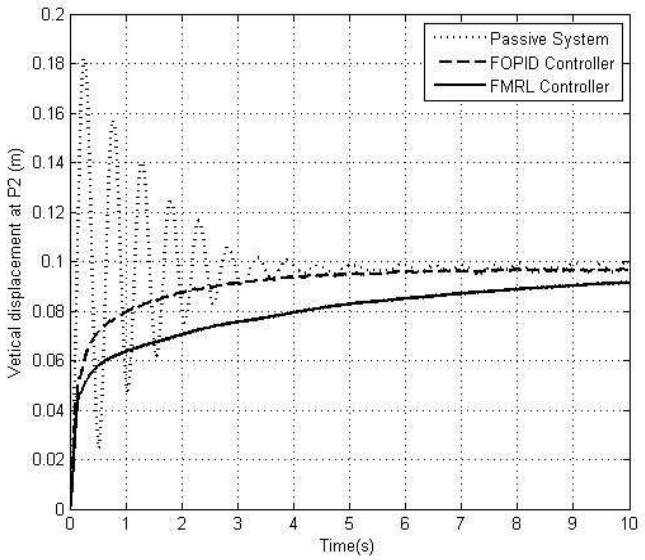


Figure 4.27 Time response of a vertical displacement at P_2

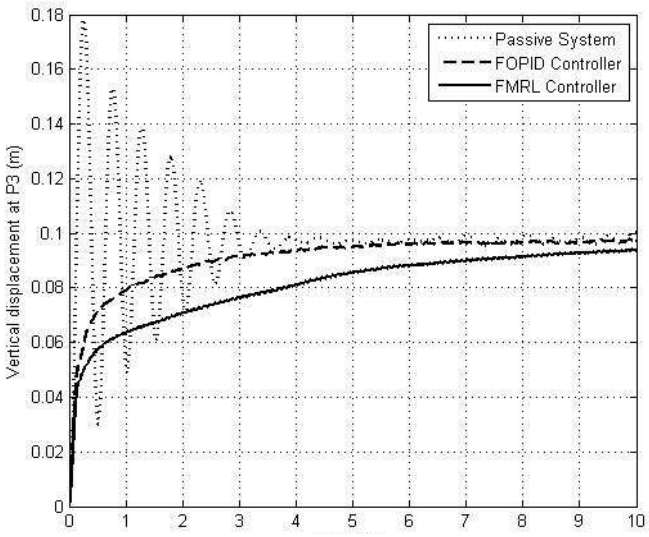


Figure 4.28Time response of a vertical displacement at P_3

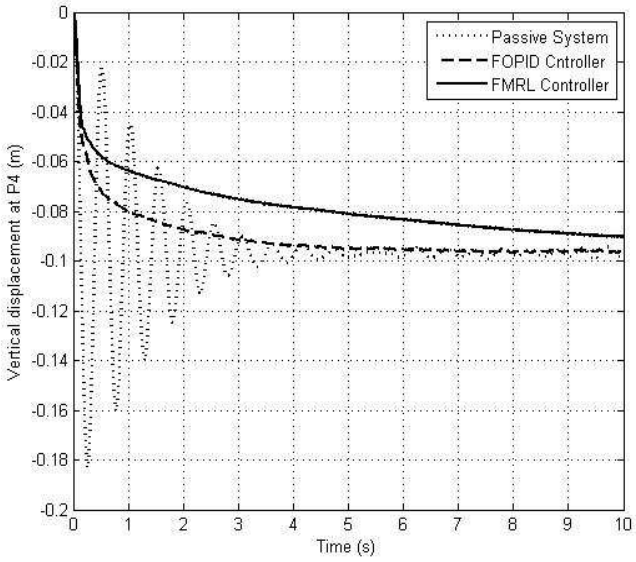


Figure 4.29 Time response of a vertical displacement at P_4

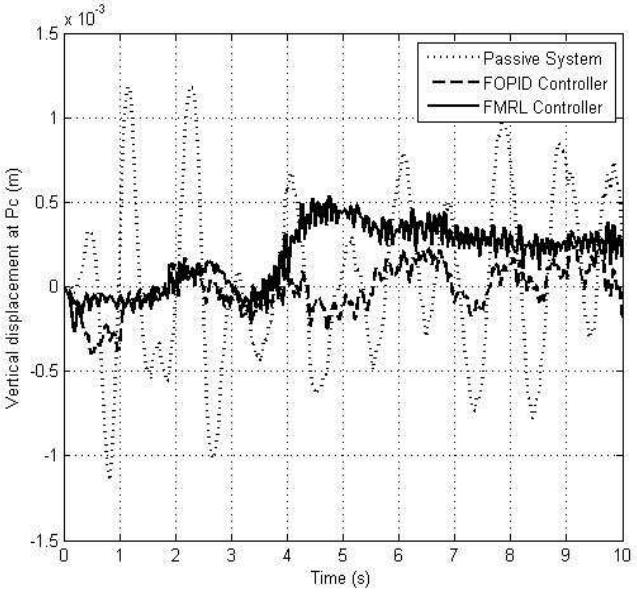


Figure 4.30 Time response of a vertical displacement at P_c

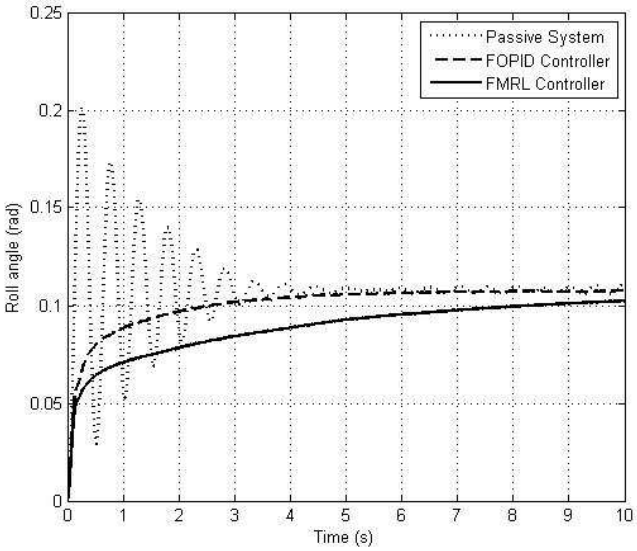


Figure 4.31 Time response of a roll angle

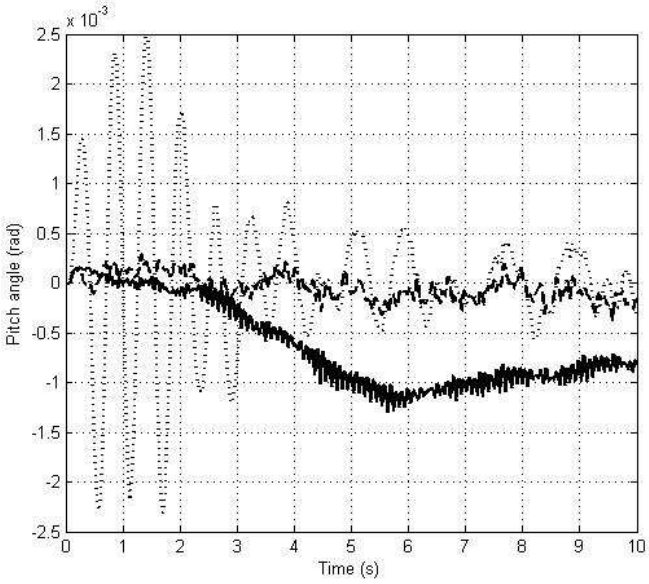


Figure 4.32 Time response of a pitch angle

When the bending torque is applied, the responses of the roll angle and the vertical displacement at the each corner of the vehicle system are improved as shown in the figures above. On the other hand, the responses of the vertical displacement at P_c and the pitch angle of the controller have not modified, but they are still acceptable because their amplitudes are still small.

To test the effectiveness of the proposed controller during the braking of the vehicle, a braking torque (shown in Figure 4.33) has been applied. Also, the random input has been chosen as the road profile. The Figures 4.34-4.40 show the comparison between the outputs of the controlled system with FMRL controller, the outputs of the controlled system with FOPID controller and the corresponding outputs of the passive system when the braking torque is applied.

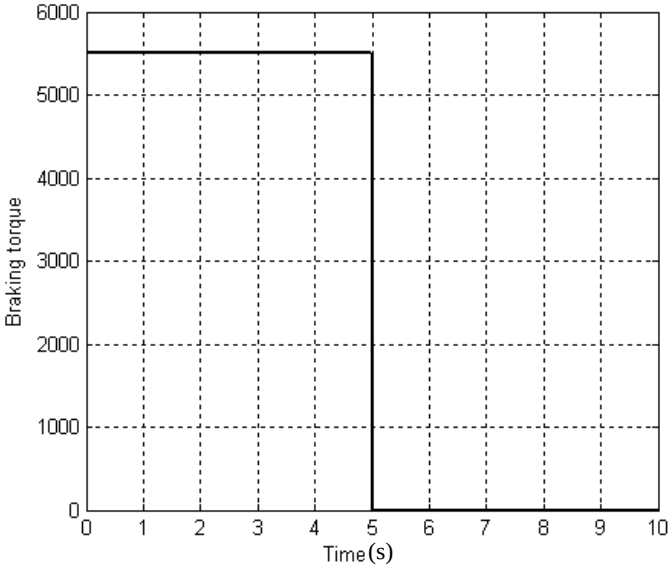


Figure 4.33 Braking torque

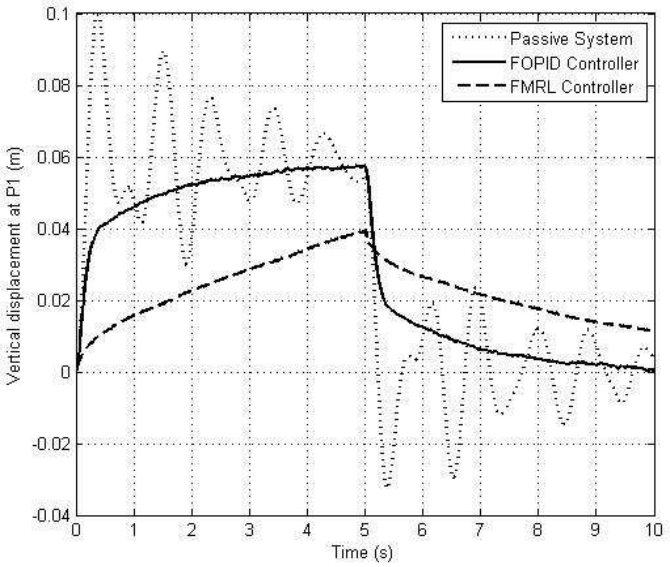


Figure 4.34 Time response of a vertical displacement at P_1

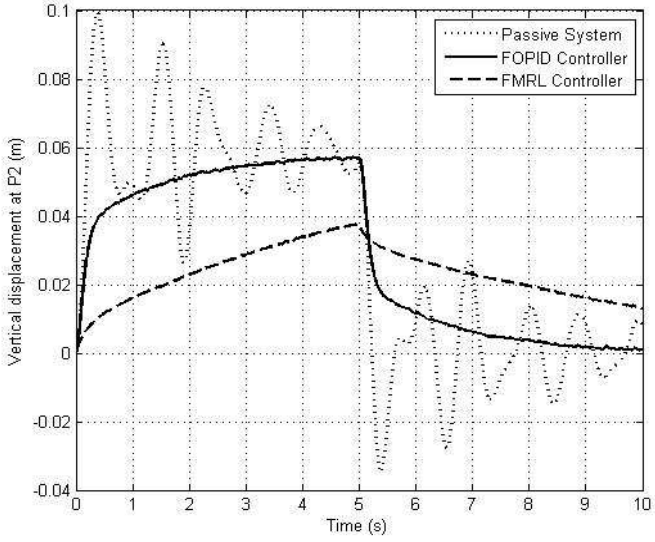


Figure 4.35 Time response of a vertical displacement at P_2

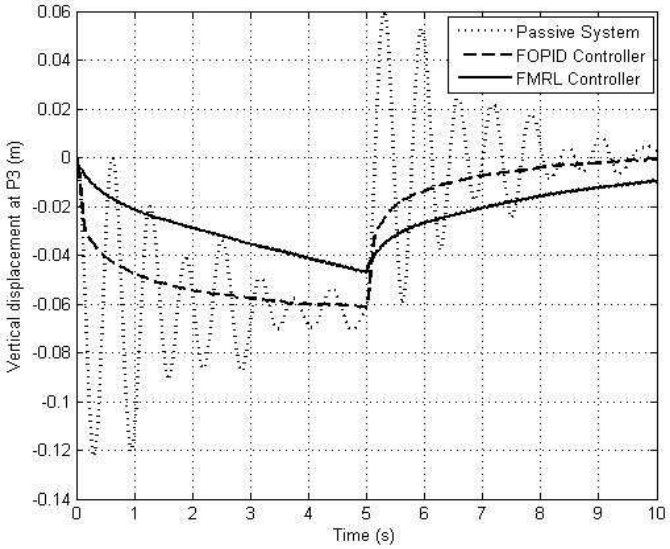


Figure 4.36 Time response of a vertical displacement at P_3

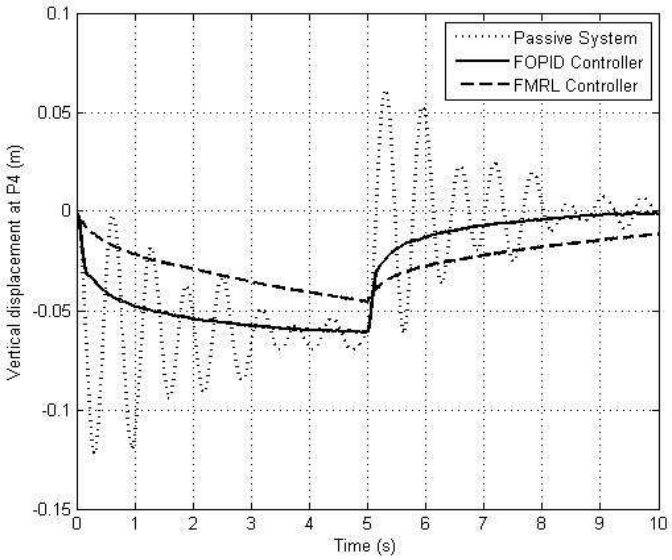


Figure 4.37 Time response of a vertical displacement at P_4

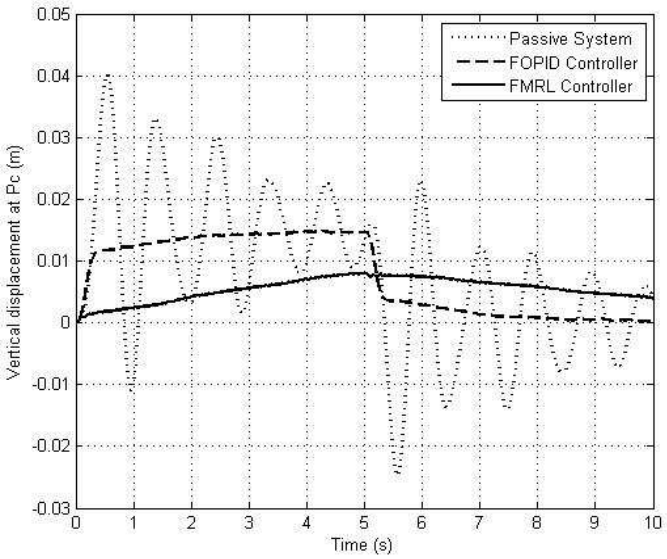


Figure 4.38 Time response of a vertical displacement at P_c

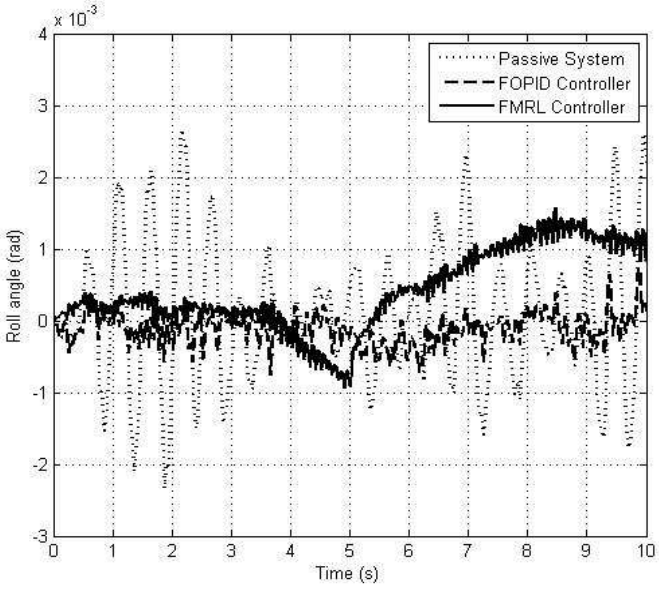


Figure 4.39 Time response of a roll angle

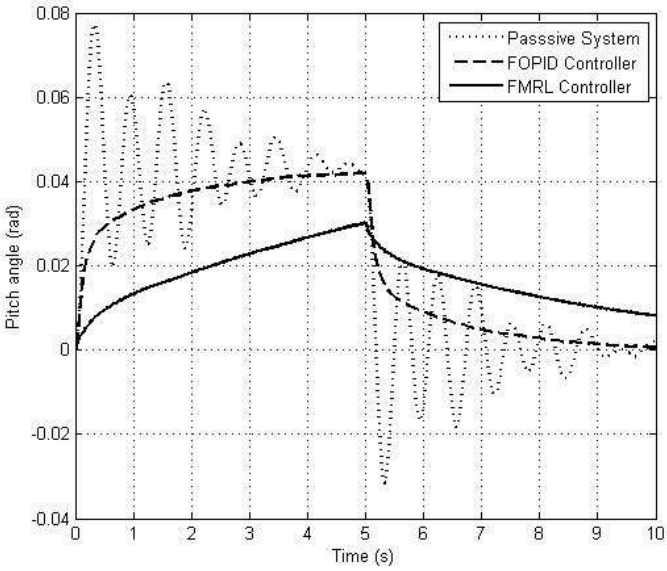


Figure 4.40 Time response of a pitch angle

By applying the braking torque on the controlled system, the riding comfort and road handling have been improved as shown in the figures above. Just the response of the roll angle has not improved, but it is still acceptable as the amplitude is still small.

4.6 Robustness test of the FMRLC

The robustness of any proposed controller is an important issue and should be studied to establish the effectiveness of the controller. Therefore, in this section the robustness of the FMRLC will be studied by applied six types of disturbances in turn to test the effectiveness of the FMRL controller. The same types of disturbances that used in Chapter 3 will be used here to compare between the performances of controlled system with proposed controller and the corresponding passive system.

- *Square wave with varying amplitude is applied as the input of road profile.*
- *Sine wave input signal with varying amplitude is applied as the input of road profile.*
- *Square wave with varying frequency is applied as the input of road profile.*
- *Sine wave input signal with varying frequency is applied as the input of road profile.*
- *Different values of bending inertia torque (T_x) are applied.*
- *Different values of breaking inertia torque (T_y) are applied.*

Figures 4.41-4.46 illustrate the comparison between the performance of plant with the FMRLC and the passive system under different disturbances types.

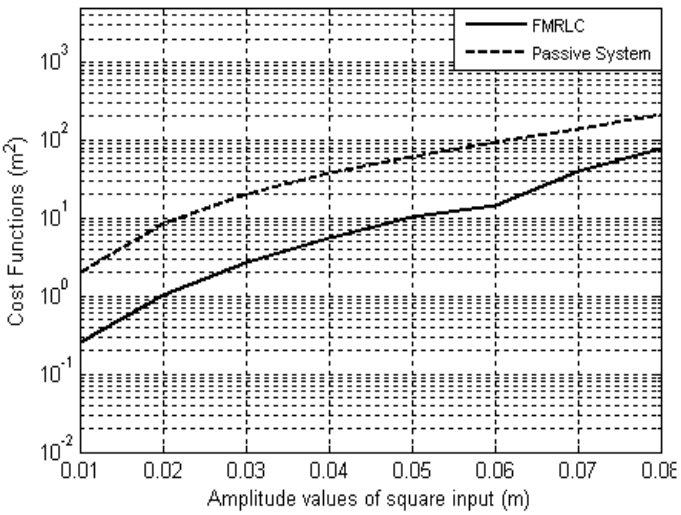


Figure 4.41 Time response of the cost functions against the different amplitude of square wave

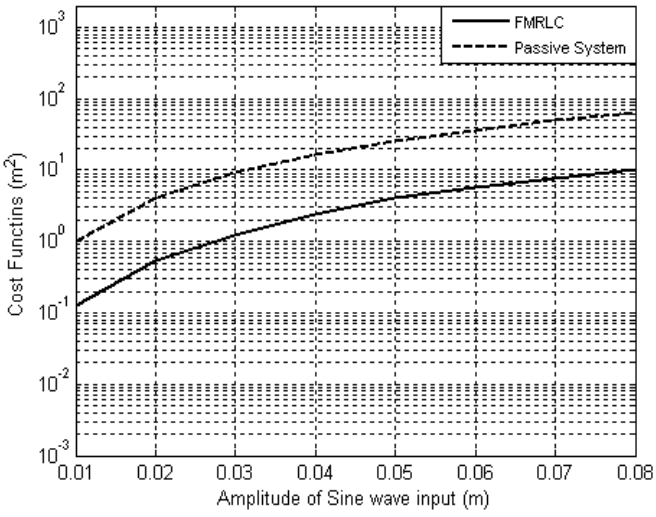


Figure 4.42 Time response of the cost functions against the different amplitude of sine wave

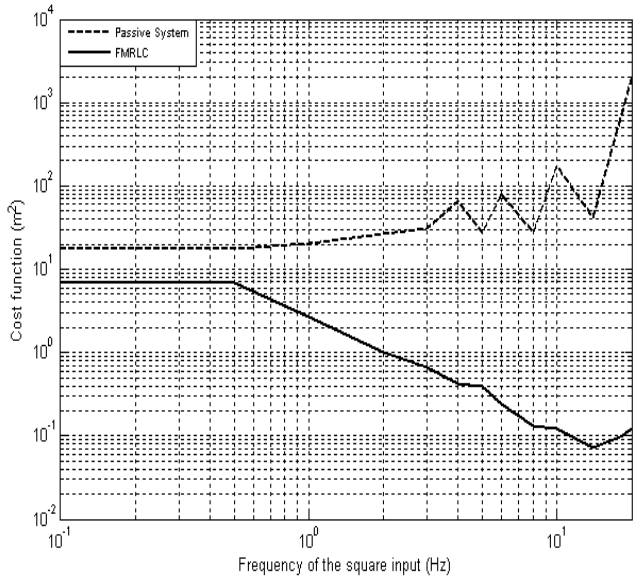


Figure 4.43 Time response of the cost function against different values of square wave frequencies

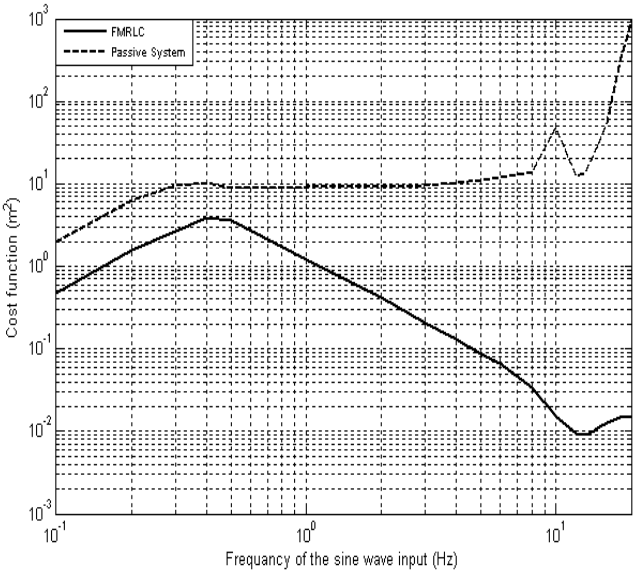


Figure 4.44 Time response of the cost function against different values of sine wave frequencies

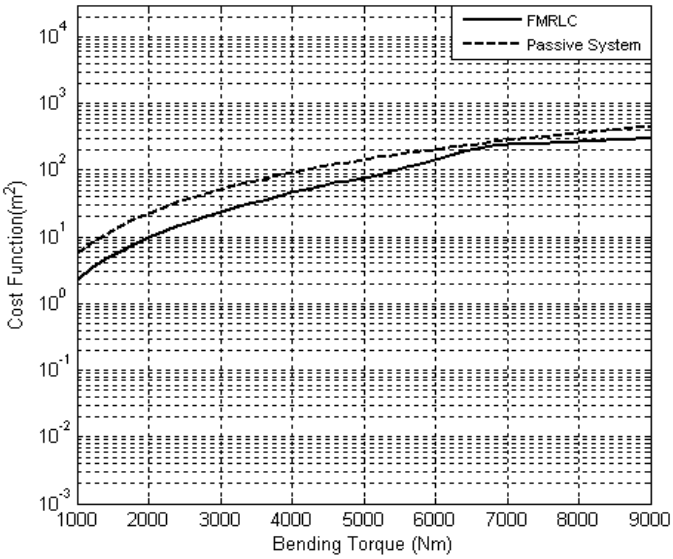


Figure 4.45 Time response of the cost functions against bending torque (T_x)

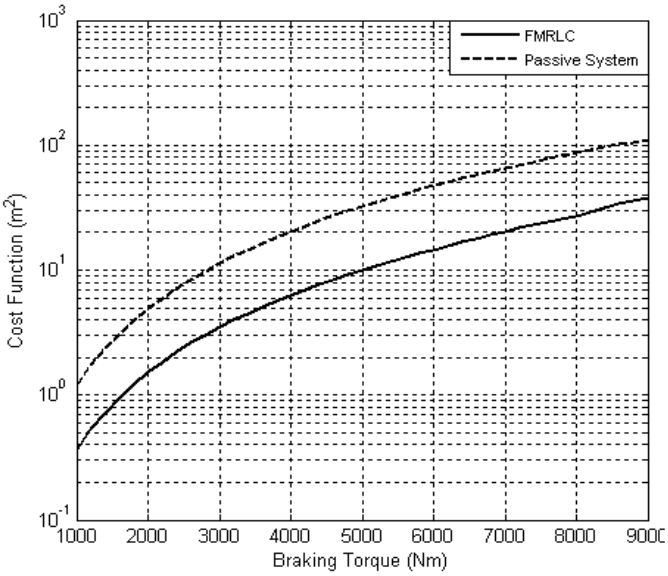


Figure 4.46 Time response of the cost functions against breaking torque (T_y)

The figures above demonstrate that when different types of disturbances are applied, the responses of the cost function of the controlled system have been improved. The response of the cost function against the different values of the bending torque has been slightly improved. It means that when the different values of the bending torque are applied, the fuzzy controller will not be a robust controller.

4.7 Summary

Many researchers suggested the Fuzzy controller for different applications for many reasons such as that it can be applied to plants that are difficult to model mathematically; and that the controller can be designed to apply heuristic rules that reflect the experience of human experts. Though, the drawback of using the fuzzy system as a controller is that it is hard to justify the choice of fuzzy controller parameters. The self- learning of a fuzzy control system has been designed in this thesis. The Fuzzy Model Reference Learning (FMRL) controller has the capability of tuning some of its parameters depending on the errors between the plant outputs and the desired outputs to generate a suitable control signals to achieve the objectives control.

The proposed fuzzy controllers adjust the hydraulic actuators forces to minimize the vertical displacements at each suspension when travelling on rough roads and to reduce the tendency of vehicle to rollover during sharp manoeuvres such as breaking and cornering. Four fuzzy controllers have been designed, one for each suspension system. The S-function has been used to write the codes to simulate the FRMLC. The MATLAB/SIMULINK programme package has been used to simulate the training phase of the FRMLC with controlled system. The results illustrated that after 47 iterations, the centres of the output membership functions of the fuzzy controllers are adjusted. Then, the new parameters of the output membership functions, obtained from the training phase, are set as parameters of the output membership functions of the fuzzy controller during the working phase.

The results in the previous section indicate that the proposed controller can improve the output transient responses of the full vehicle nonlinear active suspension systems when just the random input has been applied as road profile. The result illustrated that the amplitude of the output transient responses of the controlled system with the proposed controller are

smaller then the amplitude of the output transient responses of the passive system. When the bending torque is applied, the oscillation of the output transient responses are eliminated and the steady state value has become smaller than the steady state value of transient response of the passive system which means the proposed controller improves the output transient responses of the controlled system. But the response of the vertical displacement at the center of gravity and pitching movement are not improved. On the other hand, when the braking torque has been applied, only the response of the rolling movement has not improved.

When different disturbances types are applied, the performance of the cost function of the controlled system has been compared with the corresponding performance of the cost function of the passive system. The results show that the fuzzy controller improves the performance of the cost function of the controlled system to a small value. From the results in Chapter 3, the performances of the cost functions by using the FMRLC are better than the performances of the cost functions by using the FOPID controller for the same disturbances applied.

5. Neural Control

5.1 Structure of neural networks

Artificial Neural Networks (NNs) are capable of handling the complex and the nonlinear problems, processing the information rapidly and reducing the engineering effort required in controller model development. They consist of a pool of simple processing units that communicate by sending signals to each other over a large number of weighted connections. The important properties of neural networks are: capability of learning information by example, ability to generalize to new input and robustness to noisy data. From these properties, the neural networks are able to solve complex problems that are currently intractable.

In general, there are two types of neural networks: *(i)* feed forward neural networks which propagates the signals in forward direction only; *(ii)* the recurrent neural networks which propagates the signals in both directions (forward and backward). Examples of feed forward networks are Perceptron and Adaline. While, the important examples of recurrent networks are presented by Kohonen (1977) and Hopfield (1982).

Derivation of a mathematical model for nonlinear system is an important step to design a robust controller. But this task is very difficult, some time impossible, and time-consuming. The neural networks are consequently often used to approximate function; therefore, trained neural networks might give good alternatives for physical modelling techniques for nonlinear systems.

The standard neural network consists of three layers: the input layer, hidden layer and output layer. Each layer consists of many basic units called Neuron. The structure of a single artificial neural network or a Neuron is shown in figure 5.1.

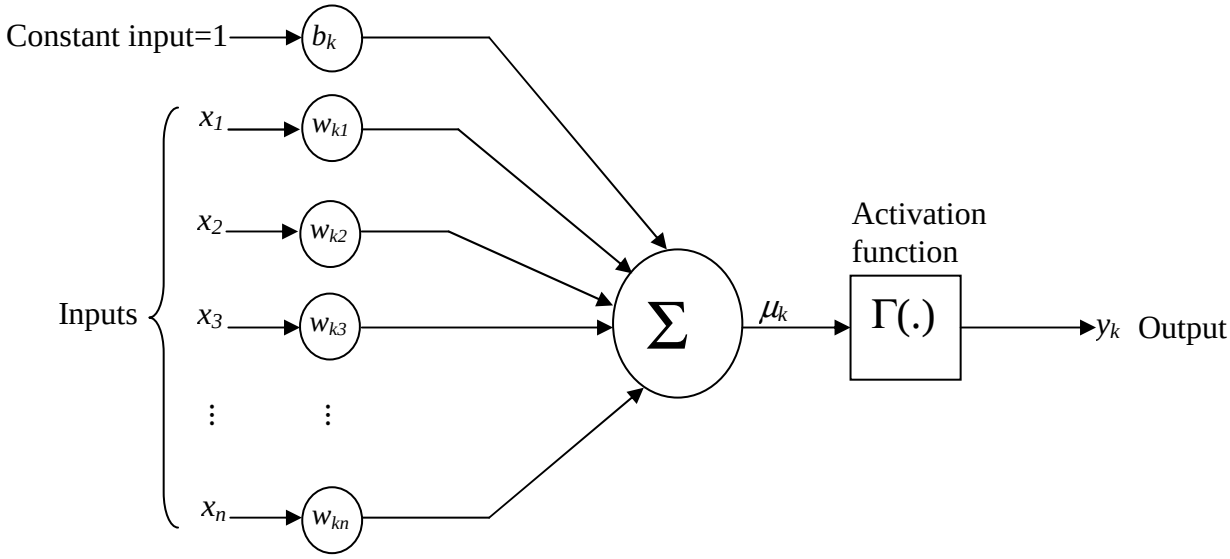


Figure 5.1 Structure of the neuron

The output of the neuron can be written as:

$$y_k = \Gamma(\sum_{i=1}^{n} x_i w_{ki} + b_k) \tag{5.1}$$

where $\Gamma(.)$ is an activation function; w_{ki} is the k^{th} weight of the i^{th} input and b_k is the k^{th} bias. The activation function can be: Sigmoid function, Tanh (Hard nonlinearity) function, Signum function or Step function.

Neural network architecture is quite simple to create and involves two or more neurons combined to form one or more layers. Figure 5.2 depicts the structure of multilayer neural network or some time called multilayer perceptron network. In this Figure, the neural network model has three layers: input layer, hidden layer and output layer. The input layer represents the input variables related to the problem. The output layer represents the desired output response of the system. The nodes in the hidden layer and output layer are the processing

elements that allow the network to develop a behavioural representation of the problem space being addressed. Processing of the input information occurs at each of the hidden and output nodes within the network and is computationally relatively simple. Each node in the particular layer of the network is connected to all of the nodes in the previous layer. There is a weight value associated with each of the connection between nodes. The weighted inputs to a particular node are summed and the resultant value is passed through a nonlinear activation function to determine the output value of the node. Thus, each node has multiple inputs and single output.

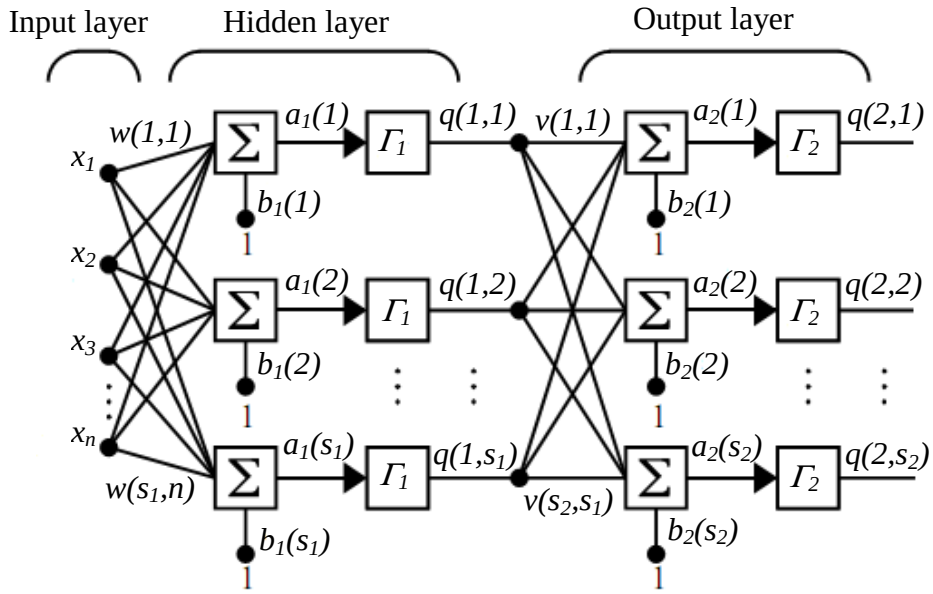


Figure 5.2 Multilayer neural networks

The output of the k^{th} node in the hidden layer can be given as

$$q(1,k) = \Gamma_1\left(\sum_{i=1}^{n} w(k,i)x_i + b_1(k)\right) = \Gamma_1\big(a_1(k)\big); \qquad k = 1,2,\dots,s_1 \qquad (5.2)$$

where $q(1,k)$ is the output of the k^{th} node in the hidden layer; Γ_1 the activation function of the hidden layer; $w(k,i)$ the weight between i^{th} input and k^{th} node, x_i the i^{th} input and $b_1(k)$ the bias of the k^{th} node.

The output of the l^{th} node in the output layer can be given as

$$q(2,l) = \Gamma_2\left(\sum_{k=1}^{s_1} v(l,k)q(1,k) + b_2(l)\right) = \Gamma_2\big(a_2(l)\big); \qquad l = 1,2,\dots,s_2 \tag{5.3}$$

where $q(2,l)$ is the output of the l^{th} node in the output layer; Γ_2 the activation function of the output layer, $v(l,k)$ the weight between k^{th} node and l^{th} node and $b_2(l)$ the bias of the l^{th} node.

There are many training algorithms which can be used to determine the optimal values for the weights and biases between each of the nodes to minimise a Mean Squared Error (MSE) function

$$MSE = \frac{1}{2P}\sum_{h=1}^{P}\sum_{j=1}^{s_2}(q_h(2,j) - T_h(j))^2 \tag{5.4}$$

where P is the number of data for each epoch, $q_h(2,j)$ is the output of j^{th} node in the output layer for h^{th} epoch and $T_h(j)$ is the j^{th} desire output for h^{th} epoch.

5.2 Training phase of neural networks

The training phase of the neural networks requires a set of examples of proper network behaviour, i.e. the network inputs and target output. The trainable parameters of neural network (weights and biases) are adjusted during training phase to minimize the mean squared error (MSE) between the neural network outputs and the target outputs. At the first iteration, the trainable parameters of the neural network are randomly initialized. Then the neural network processes each input vector. After that, the output of the neural network is compared with the desired output or target output. Finally, the MSE is computed to indicate the performance of the neural network. Initially, the error would be large due to the random assignment of the values to trainable parameters. The values of the trainable parameters are updated to determine the new values of these parameters to use in the next iteration. The following update equation is used to find the new values of the trainable parameter vectors:

$$\psi^{p+1} = \psi^{p} + \Delta\psi$$

$$\Delta\psi = -\gamma g_p \tag{5.5}$$

where ψ^p is the vector of the current trainable parameters, $\Delta\psi$ the increment value of the trainable parameters vector, γ the learning rate and g_p the gradient of the error with respect to trainable parameter vectors. The negative sign indicates that the new trainable parameters vector ψ^{p+1} is moving in a direction opposite to that of the gradient.

There are two different ways in which this gradient descent algorithm can be implemented: the incremental mode and batch mode. In the incremental mode, the gradient is computed and the trainable parameters are updated after each input is applied to the neural network. In the batch mode, the trainable parameters are updated after all of the inputs are applied to the neural network.

Figure 5.3 indicates a slice through the error changes with one weight. If the starting point on the error surface is at 1, the slope is negative as shown in the figure and the value of the weight needs to be increased to reach the optimum weight value $W_{opt,}$ at which the MSE is minimal. If the current point is at 2, the slope is positive and the weight has to be decreased.

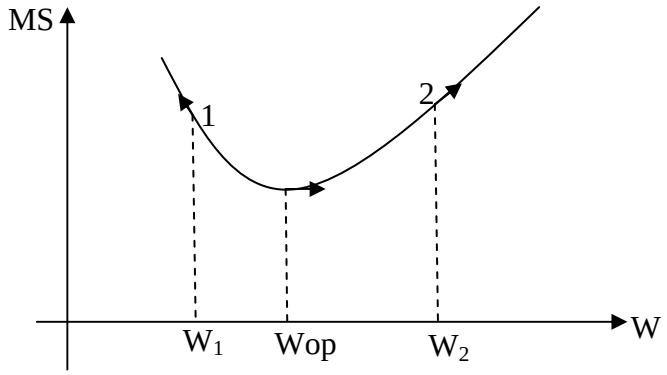


Figure 5.3 Slice through the error surface at tow values of a weight

5.3 Training algorithms of neural networks

The main target from using the training algorithm is to determine the optimal values of the trainable parameters at which the MSE is minimal. There are many training algorithms were

suggested to identify the neural networks such as: the Backpropagation algorithm, Variable learning rate algorithm, Conjugate gradient algorithm, Levenberg marquardt algorithm, etc. To determine the new values of the trainable parameters, the following steps should be followed:

1. The trainable parameters are randomly initialised.
2. The input and output data, which are obtained from the physical model, are normalized.
3. The number of the neurons in each layer and the activation function for neurons are selected, i.e. determination the neural network structure.
4. Applying the input vector to the neural networks to compute the outputs of the neural network depend on the current training parameters.
5. Computing the error between the neural network outputs and the desired outputs.
6. Employed the error to calculate the new values of the trainable parameters from the update equation which is different from one algorithm to other.
7. Repeating steps 4-6 until the MSE becomes lower than or equal to the required minimum value.

After adequate iterations are carried out, the outputs of the neural network should match the desired outputs of the model system. At this point, the neural network said to be trained and the optimal values of the trainable parameters are obtained. The new structure of the neural network can be used in the generalization phase.

In the next subsection, the update equation for some training algorithm will be derived.

5.3.1 Backpropagation algorithm

The Back-Propagation Algorithm (BPA) has a significant improvement in neural network researches (Narendra and Parthasarathy 1990; Rumelhart et al. 1986; Samarasinghe 2007;

Sharkaway 2005; Yue et al. 2008). The simplest implementation of backpropagation learning updates the trainable parameters in the direction in which the performance function (mean squared error) decreases. Consider a multilayer feedforward neural network in Figure 5.2, to train the neural network, the update equation for weight vector in hidden layer (***W***); bias vector in hidden layer (***b₁***); weight vector in output layer (***V***); and the bias vector in output layer (***b₂***) should be derived.

I. The update equation for any weight in output layer

The mean squared error for h^{th} input data can be written as:

$$E_h = \frac{1}{2}\sum_{l=1}^{s_2}(q_h(2,l) - T_h(l))^2 \tag{5.6}$$

For simplification, the subscript *h* will be omitted, so the Eq. (5.6) can be rewritten as:

$$E = \frac{1}{2}\sum_{l=1}^{s_2}(q(2,l) - T(l))^2 = \frac{1}{2}\sum_{l=1}^{s_2} e_l^2 \tag{5.7}$$

To derive the update equation for weight in output layer, the derivative of the mean squared error to the corresponding weight ($\frac{\partial E}{\partial v(l,k)}$) should be obtained as

$$\frac{\partial E}{\partial v(l,k)} = e_l \frac{\partial q(2,l)}{\partial v(l,k)} \tag{5.8}$$

where $e_l = (q(2,l) - T(l))$, $l = 1,2,\dots,s_2$ and $k = 1,2,\dots,s_1$.

By using chain rule the derivative of the l^{th} output of neural network ($q(2,l)$) to corresponding weight ($v(l,k)$) can be written as:

$$\frac{\partial q(2,l)}{\partial v(l,k)} = \frac{\partial q(2,l)}{\partial a_2(l)} . \frac{\partial a_2(l)}{\partial v(l,k)} \tag{5.9}$$

By Substituting Eq. (5.9) in (5.8)

$$\frac{\partial E}{\partial v(l,k)} = e_l \frac{\partial q(2,l)}{\partial a_2(l)} . \frac{\partial a_2(l)}{\partial v(l,k)} \tag{5.10}$$

The term $\frac{\partial q(2,l)}{\partial a_2(l)}$ in the Eq. (5.10) is equal to the derivative of activation function in output layer with respect to time. Therefore, this term can be written as $\frac{\partial q(2,l)}{\partial a_2(l)} = \dot{\Gamma}_2(a_2(l))$. From Eq. (5.3), the second term $\frac{\partial a_2(l)}{\partial v(l,k)}$ is equal to $q(1,k)$.

The Eq. (5.10) is written as:

$$\frac{\partial E}{\partial v(l,k)} = e_l \dot{\Gamma}_2(a_2(l)) q(1,k) \tag{5.11}$$

From Eq. (5.5) the update equation for any weight in output layer can be given as:

$$v^{p+1}(l,k) = v^p(l,k) + \gamma e_l \dot{\Gamma}_2(a_2(l)) q(1,k) \tag{5.12}$$

II. The update equation for any bias in output layer

The derivative of mean squared error with respect to any bias in output layer is given by:

$$\frac{\partial E}{\partial b_2(l)} = e_l \frac{\partial q(2,l)}{\partial b_2(l)} = e_l \frac{\partial q(2,l)}{\partial a_2(l)} . \frac{\partial a_2(l)}{\partial b_2(l)} \tag{5.13}$$

From Eq. (5.3), the term $\frac{\partial a_2(l)}{\partial b_2(l)}$ is equal to 1.

The Eq. (5.13) can be written as

$$\frac{\partial E}{\partial b_2(l)} = e_l \dot{\Gamma}_2(a_2(l)) \tag{5.14}$$

The update equation of any bias in output layer is given as

$$b_2^{p+1}(l) = b_2^p(l) + \gamma e_l \dot{\Gamma}_2(a_2(l)) \tag{5.15}$$

III. The update equation for any weight in hidden layer

The derivative of mean squared error with respect to any weight in hidden layer is given by:

$$\frac{\partial E}{\partial w(k,i)} = \sum_{l=1}^{s_2} e_l \frac{\partial q(2,l)}{\partial w(k,i)} \tag{5.16}$$

where $k = 1,2, \dots, s_1$ and $i = 1, 2, \dots, n$.

By using the chain rule the term $\frac{\partial q(2,j)}{\partial w(k,i)}$ is written as

$$\frac{\partial q(2,l)}{\partial w(k,i)} = \frac{\partial q(2,l)}{\partial a_2(l)} \frac{\partial a_2(l)}{\partial q(1,k)} \frac{\partial q(1,k)}{\partial a_1(k)} \frac{\partial a_1(k)}{\partial w(k,i)} \tag{5.17}$$

From Eq. (5.2) and (5.3), Eq. (5.17) can be written as

$$\frac{\partial q(2,l)}{\partial w(k,i)} = \dot{\Gamma}_2\big(a_2(l)\big).v(l,k).\dot{\Gamma}_1\big(a_1(k)\big).x_i \tag{5.18}$$

By substituting Eq. (5.16) in Eq. (5.16)

$$\frac{\partial E}{\partial w(k,i)} = \sum_{l=1}^{s_2} e_l\, \dot{\Gamma}_2\big(a_2(l)\big).v(l,k).\dot{\Gamma}_1\big(a_1(k)\big).x_i \tag{5.19}$$

The update equation for any weight in hidden layer is given by

$$w^{p+1}(k,i) = w^p(k,i) + \gamma \sum_{l=1}^{s_2} e_l\, \dot{\Gamma}_2\big(a_2(l)\big).v(l,k).\dot{\Gamma}_1\big(a_1(k)\big).x_i \tag{5.20}$$

IV. The update equation for any bias in hidden layer

The derivative of men squared error with respect to any bias in hidden layer can be given by:

$$\frac{\partial E}{\partial b_1(k)} = \sum_{l=1}^{s_2} e_l \frac{\partial q(2,l)}{\partial b_1(k)} \tag{5.21}$$

By using the chain rule the term $\frac{\partial q(2,l)}{\partial b_1(k)}$ is written as

$$\frac{\partial q(2,l)}{\partial b_1(k)} = \frac{\partial q(2,l)}{\partial a_2(l)} \frac{\partial a_2(l)}{\partial q(1,k)} \frac{\partial q(1,k)}{\partial a_1(k)} \frac{\partial a_1(k)}{\partial b_1(k)} \tag{5.22}$$

From Eq. (5.2) and (5.3), the Eq. (5.20) is given as

$$\frac{\partial q(2,l)}{\partial b_1(k)} = \dot{\Gamma}_2\big(a_2(l)\big).v(l,k).\dot{\Gamma}_1\big(a_1(k)\big) \tag{5.23}$$

The update equation of for any bias in hidden layer is given as:

$$b_1^{p+1}(k) = b_1^p(k) + \gamma \sum_{l=1}^{s_2} e_l\, \dot{\Gamma}_2\big(a_2(l)\big).v(l,k).\dot{\Gamma}_1\big(a_1(k)\big) \tag{5.24}$$

From Equations (5.12), (5.15), (5.20) and (5.24), at each iteration, the new values of the trainable parameters can be obtained.

5.3.2 Fast training algorithms

The Backpropagation Algorithm is known as an algorithm with a very poor converging rate for practical problems (Wilamowski et al. 2001). Many researches carried out to accelerate the convergence of the algorithm. These researches focused on two different categories. The first category uses heuristic techniques, which developed from an analysis of the performance of the backpropagation algorithm (Rigler et al. 1991; Tollenaere 1990; Vogl et al. 1988). There are three different heuristic techniques: Momentum Technique, Variable Learning Rate Technique and Resilient Technique. On the other hand, the second category of fast algorithms uses standard numerical optimization techniques. There are three main techniques of numerical optimization: Conjugate Gradient Technique by Peng and Magoulas 2007; Quasi-Newton Technique by Ishikamn et al. 1996; and Levenberg Marquardt Technique by Lera and Pinzolas 2002.

The conjugate gradient technique produces generally faster convergence than steepest descent directions by searching along conjugate direction. The Quasi-Newton technique is faster than the conjugate gradient technique, but it is complex and expensive to compute the Hessian matrix for feedforward NNs. These two algorithms lead to a little acceptable result when the nonlinearity is heavy.

The Levenberg-Marquardt Technique (LMT) is an iterative technique that locates the minimum of a multivariate function expressed as the sum of squares of nonlinear real-valued functions. It has become a standard technique for nonlinear least-squares problems widely adopted in a broad spectrum of disciplines (Wilamowski et al. 2001). The LMT can be thought of a combination of Backpropagation Algorithm and the Quasi-Newton Technique (Suratgar et al. 2005). When the current solution is far from the correct one, the algorithm behaves like a Backpropagation Algorithm: slow but guaranteed to converge. When the current solution is close to the correct solution, it becomes a Quasi-Newton method.

With the LMA, the increment of the trainable parameters vector can be calculated as follows:

$$\boldsymbol{\Delta\psi} = [\boldsymbol{J^T J} + \boldsymbol{\mu I}]^{-1}\boldsymbol{J^T e} \tag{5.25}$$

where $\boldsymbol{\psi}$ is the trainable parameters vector; $\boldsymbol{I}$ is identity matrix; $\boldsymbol{J}$ is the Jacobian matrix; $\boldsymbol{\mu}$ is the learning rate which automatically adjust during learning phase; and $\boldsymbol{e}$ is the cumulative error vector, it can be written as follows:

$$\boldsymbol{e} = \left[e_{11}\ e_{21}\ \dots\ e_{s_2 1}\ e_{12}\ e_{22}\ \dots\ e_{s_2 2}\ \dots\ e_{1P}\ e_{2P}\ \dots\ e_{s_2 P}\right]^T$$

where P is the number of data for each epoch, s_2 number of neural network outputs.

$$e_{lh} = q_h(2, l) - T_h(l)$$

where $h = 1,2, \dots, P$ and $l = 1, 2, \dots, s_2$.

If the performance measure (MSE) in epoch *p+1* is greater than the performance measure in epoch p, μ is divided by a constant number ζ $(0 < \zeta < 1)$, whenever the performance measure decreased, μ is multiplied by ζ. Eq. (5.25) shows that if μ is equal to zero the LMT becomes Quasi-Newton Technique, where in this technique the increment of the trainable parameters vector $\boldsymbol{\Delta\psi} = [\boldsymbol{J^T J}]^{-1}\boldsymbol{J^T e}$, and if μ is high the LMT becomes Backpropagation Algorithm. The Jacobian matrix (it has s_2 $P\times$ β dimension) can be computed as follows:

$$[\boldsymbol{J}] = \begin{bmatrix} \frac{\partial e_{11}}{\partial \psi_1} & \frac{\partial e_{11}}{\partial \psi_2} & \cdots & \frac{\partial e_{11}}{\partial \psi_\beta} \\ \frac{\partial e_{21}}{\partial \psi_1} & \frac{\partial e_{21}}{\partial \psi_2} & \cdots & \frac{\partial e_{21}}{\partial \psi_\beta} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial e_{m1}}{\partial \psi_1} & \frac{\partial e_{m1}}{\partial \psi_2} & \cdots & \frac{\partial e_{m1}}{\partial \psi_\beta} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial e_{1P}}{\partial \psi_1} & \frac{\partial e_{1P}}{\partial \psi_2} & \cdots & \frac{\partial e_{1P}}{\partial \psi_\beta} \\ \frac{\partial e_{2P}}{\partial \psi_1} & \frac{\partial e_{2P}}{\partial \psi_2} & \cdots & \frac{\partial e_{2P}}{\partial \psi_\beta} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial e_{s_2 P}}{\partial \psi_1} & \frac{\partial e_{s_2 P}}{\partial \psi_2} & \cdots & \frac{\partial e_{s_2 P}}{\partial \psi_\beta} \end{bmatrix}_{(\beta X s_2 P)}$$

where β is the total number of trainable parameters in neural network which must be optimized; e_{lp} is the error of l^{th} output at p^{th} epoch.

Therefore, trainable parameters vector can be updated as follows:

$$\boldsymbol{\psi^{p+1} = \psi^{p} + \Delta\psi = \psi^{p} + [J^{T}J + \mu I]^{-1}J^{T}e} \tag{5.26}$$

Since the Jacobian matrix contains first derivatives of the network errors with respect to the weights and biases (as shown above), therefore, the terms in the Jacobian matrix can be computed from Equations 5.11, 5.14, 5.19 and 5.23.

5.4 Design of the proposed neural controller

An intelligent controller can be used to design a control system for full vehicle nonlinear active suspension systems such as a Neural Controller (NC). The neural network is an intelligent device wildly used to design robust controllers for nonlinear processes in the engineering problems. In many publications, neural networks are employed to design control system, such as the model reference adaptive control, model predictive control, nonlinear internal model control, adaptive inverse control system and neural adaptive feedback linearization (Canete et al. 2008; Hussain 1999; Norgaard et al. 2001). The control architectures in those papers depend on designing a neural network identifier which is used as a path to propagate the error between the output of the process and output of the reference model to train and select the optimal values of the trainable parameters of the neural network control. Therefore, in those methods two neural networks were trained to track several control objectives: neural network identifier and neural network controller.

One of the main advantages of using a neural network as a controller is that the neural networks are universal function approximations which learn on the basis of examples and might be immediately applied in an adaptive control system because of their capacity to adapt in the real time.

Figure 5.4 depicts the controlled vehicle system with a neural controller as a key component. A neural controller has been designed to generate suitable control signals, which will be applied as a control input signals to govern the hydraulic actuators to generate suitable damping forces for improving the vehicle performance. To find the optimal values of the trainable parameters of the neural controller for driving the plant to meet the control objectives, an FOPID sub-controller should be designed (the details for the full design of FOPID controller for full vehicle nonlinear suspension are described in Chapter 3). The inputs and outputs data which are obtained from the FOPID controller will be used as training data set to train the parameters of neural controller (weights and biases) using the Levenberg Marquardt Training Algorithm. Figure 5.5 explains the training phase of the neural controller. The bold line means multiple input or output signals. Four neural control systems have been designed, one for each suspension unit. The inputs and outputs data of the FOPID controllers, which was designed in Chapter 3, are used as training data to design the neural controller.

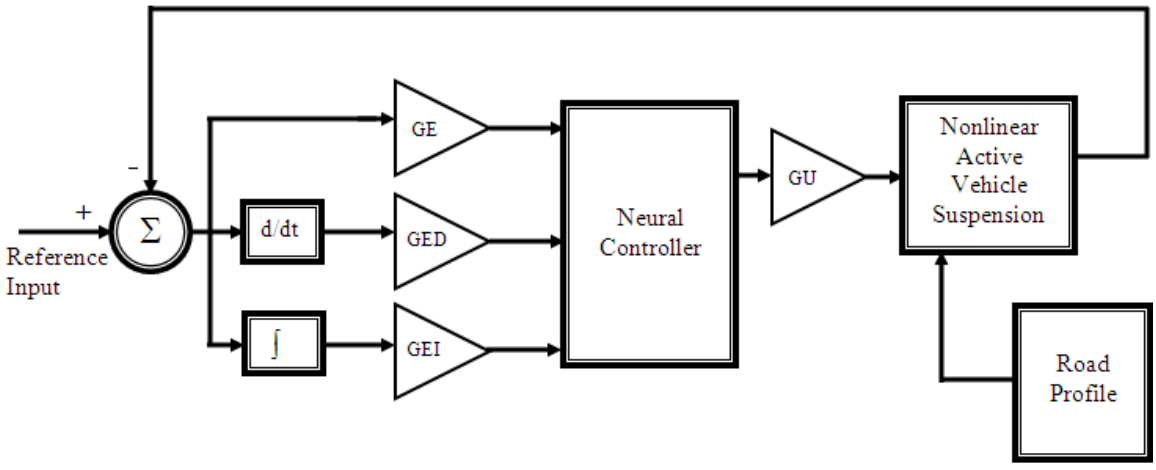


Figure 5.4 Neural controller for a full vehicle model

The Levenberg-Marquardt training algorithm has been used to obtain the optimal values of the trainable parameters of neural controllers. After the optimal values of trainable parameters

of the neural controller are obtained the neural controller design should be improved by adjusting the scaling gains, i.e. GE, GED, GEI and GU, which are shown in Figure 5.4.

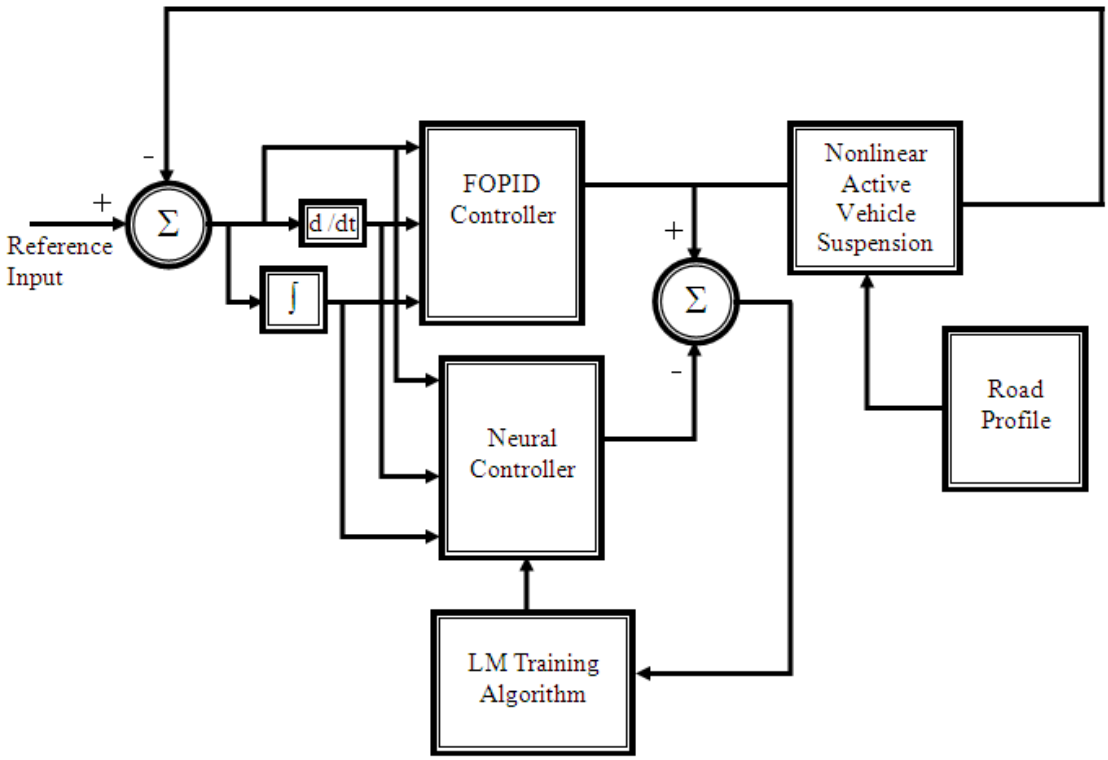


Figure 5.5 Training phase of neural controller

To select the optimal values of the scaling gains, four dimensional Golden Section Search (4-D GSS) method are introduced to reduce the trial time (For more detail about 4-D GSS method, see Reference Chang (2009)).

5.5 Using Matlab program to design the neural controller for the controlled system

Four Matlab script files have been written to train four neural controllers as shown in Appendix 3. Each neural controller has three inputs: error, derivative of the error and integral of the error, where the error is the deference between the reference input and output of the controlled system, and one output (control signal output). In the training phase, the trainable

parameters are randomly initialized. The structure of each neural network has been selected as follows: *(i)* the numbers of the neurones in the hidden layer are selected as 20; *(ii)* the activation functions of the hidden layer are selected as hyperbolic tangent sigmoid transfer function (*tansig*); *(iii)* the activation function of the output layer is selected as a linear transfer function (*purelin*); (*iv*) the training algorithm has been selected as Levenberg Marquardt algorithm (*trainlm*) and *(v)* the number of the iteration has been selected as 3000 for front-right and rear-right neural controller and 10000 for front-left and rear-left neural controller. The inputs and output data, which are obtained from the FOPID controllers, have been applied as training data set to train the parameters of the neural controllers.

After the last iteration, the outputs of the neural controllers track the outputs of the FOPID controller. The mean squared errors between the desired output and the output of the neural controller for each neural network are shown in Figures 5.6-5.9.

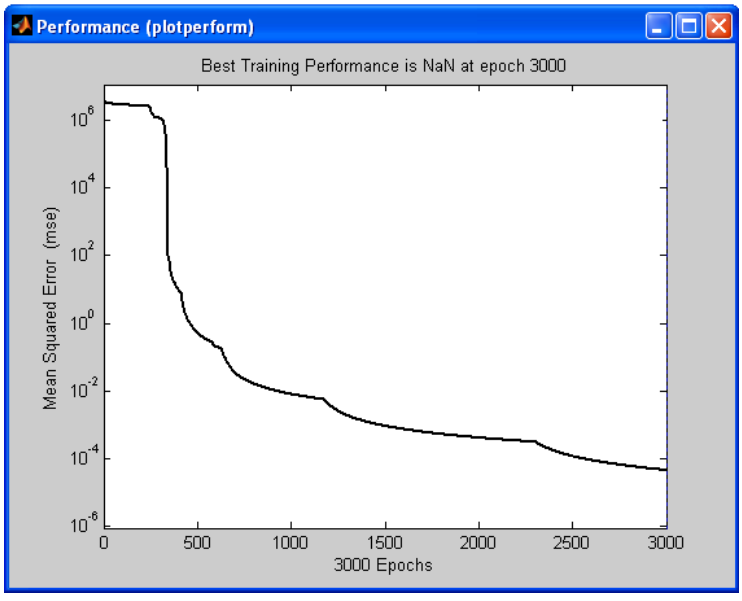


Figure 5.6 Mean squared error for front-right neural controller

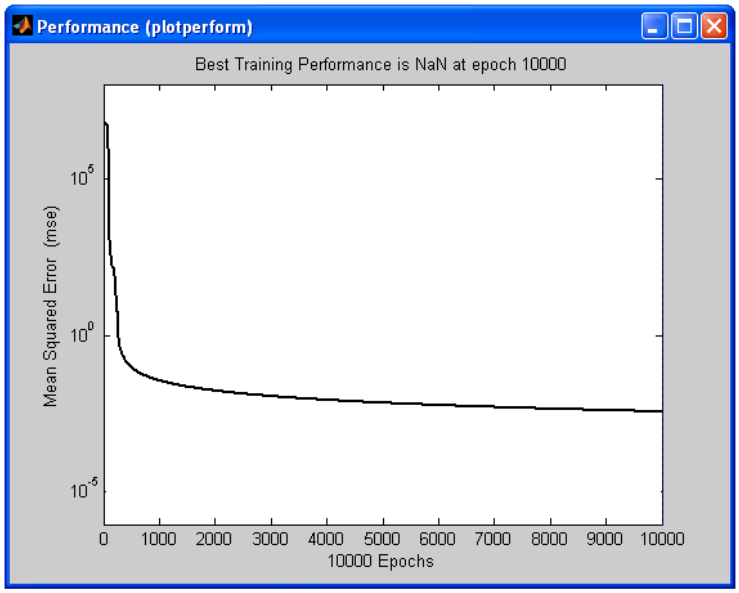


Figure 5.7 Mean squared error for front-left neural controller

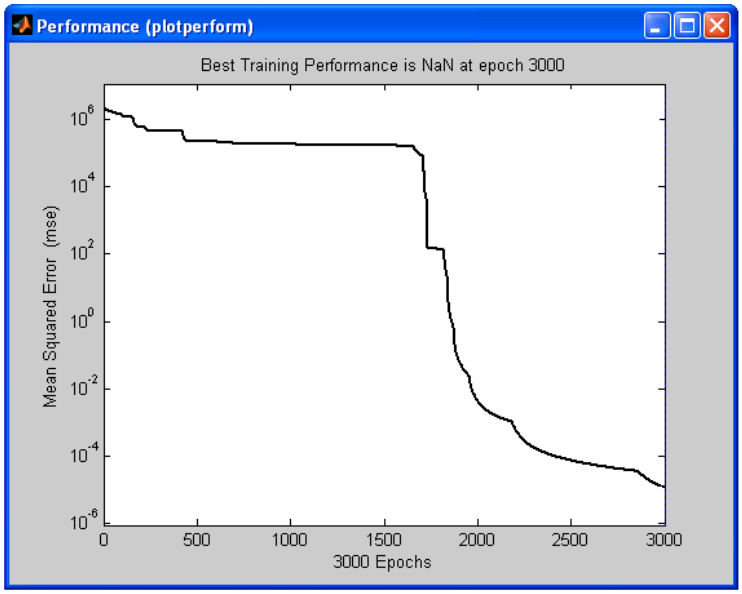


Figure 5.8 Mean squared error for rear-right neural controller

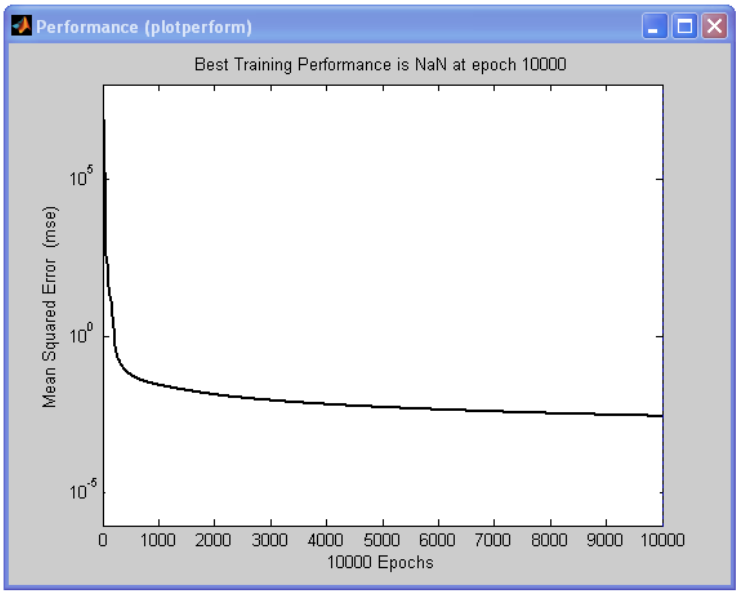


Figure 5.9 Mean squared error for rear-left neural controller

After obtaining the optimal values of the trainable parameters, the scaling gains should be adjusted to improve the performance of the neural controller. The 4-D GSS method has been used to adjust the scaling gains (GE, GED, GEI and GU). As a result, the suitable values of scaling gains are equal to 4, 3, 10 and 0.5, respectively.

After the full designing of the four neural controllers are completed, the SIMULINK model shown in Figure 5.10 is used to examin the robustness of the proposed controller.

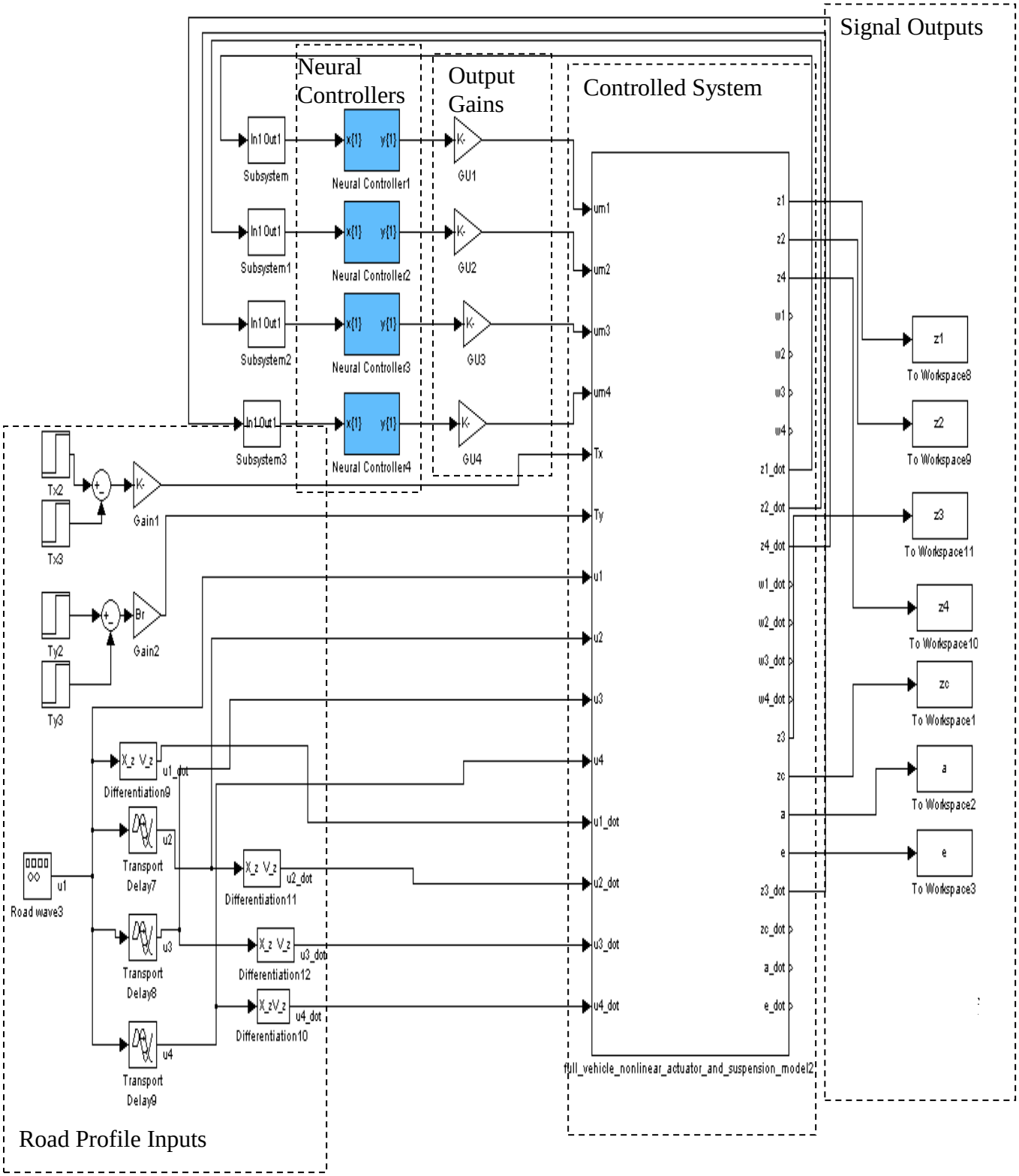


Figure 5.10 MATLAB SIMULINK model of neural controllers with controlled system

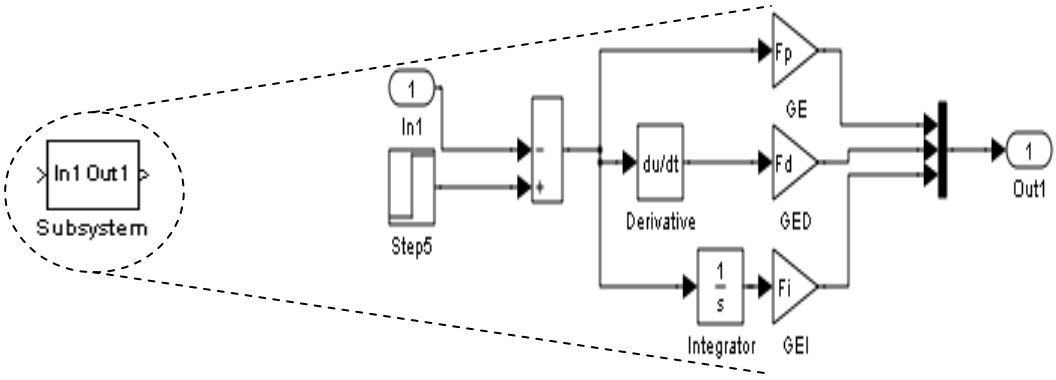


Figure 5.11 Construction of input subsystem

As shown in Figure 5.11, the neural controller has three scaled inputs and one scaled control signal output. Suitable control signals of the neural controllers have been applied as inputs to the suspension systems to control the position of the spool valves of hydraulic actuators. The hydraulic actuators generate appropriate forces to improve the performance of the vehicle system.

5.6 Simulation and results

The proposed neural controllers have been tested to examine the robustness of the controller. Random input is assumed as road profile. The outputs of the neural controllers are applied as control input signals to the hydraulic actuators. To test the effectiveness of the proposed controller, the outputs of the full vehicle system with proposed neural controllers are compared with the corresponding outputs of the full vehicle system without controller. Figures 5.12-5.18 show the result of the comparison between the outputs of controlled system with neural controllers and the outputs of passive system.

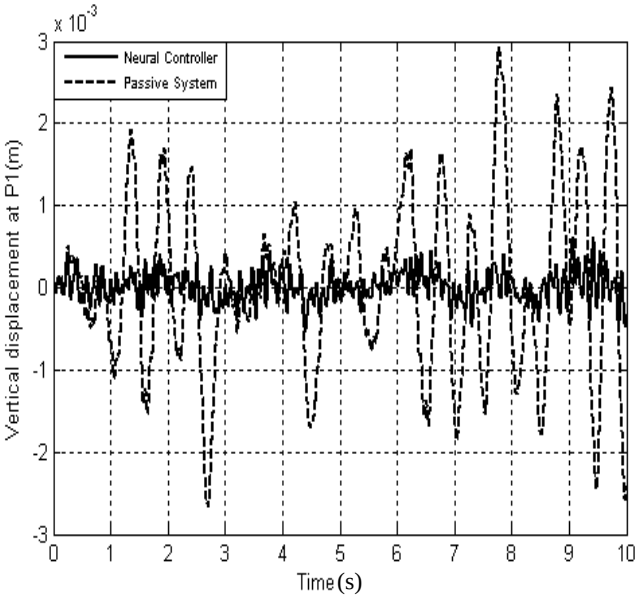


Figure 5.12 Time response of a vertical displacement at P_1

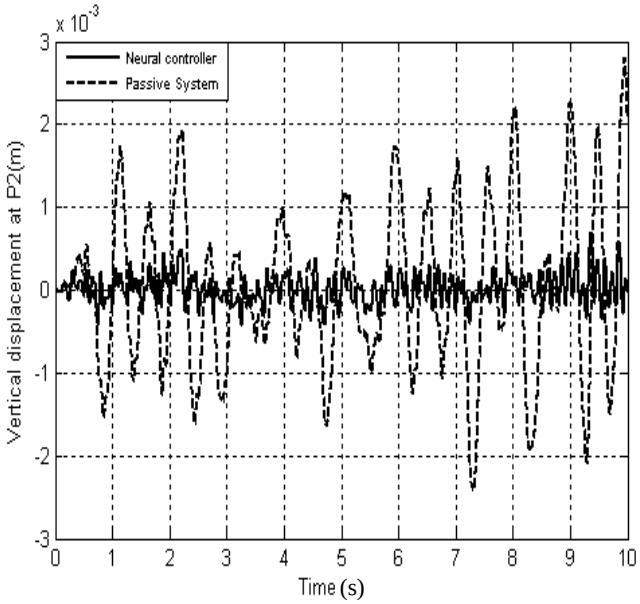


Figure 5.13 Time response of a vertical displacement at P_2

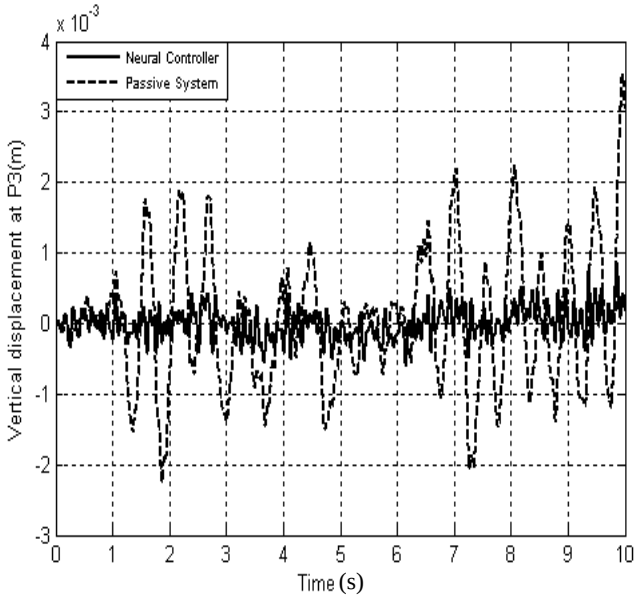


Figure 5.14 Time response of a vertical displacement at P_3

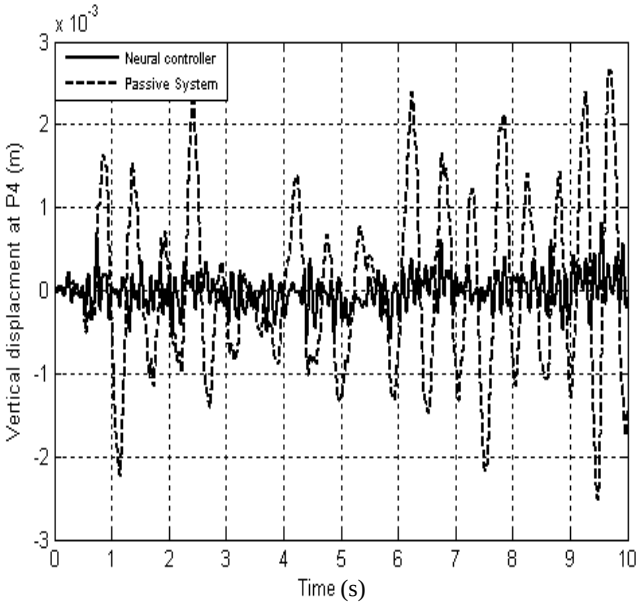


Figure 5.15 Time response of a vertical displacement at P_4

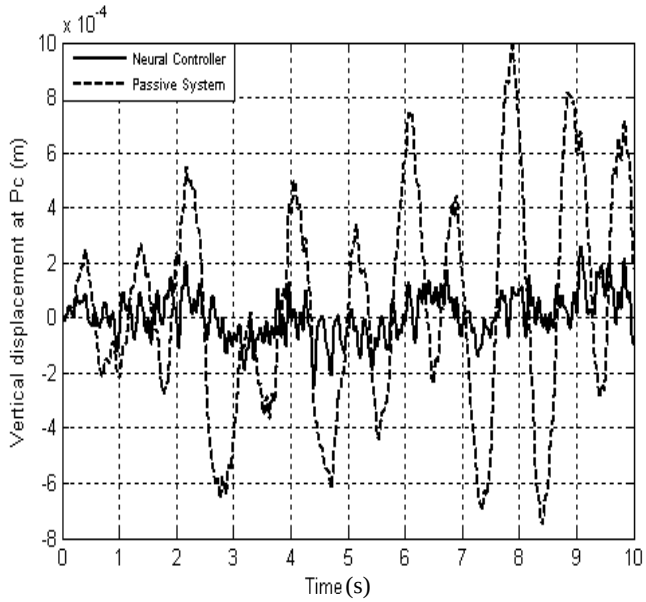


Figure 5.16 Time response of a vertical displacement at P_c

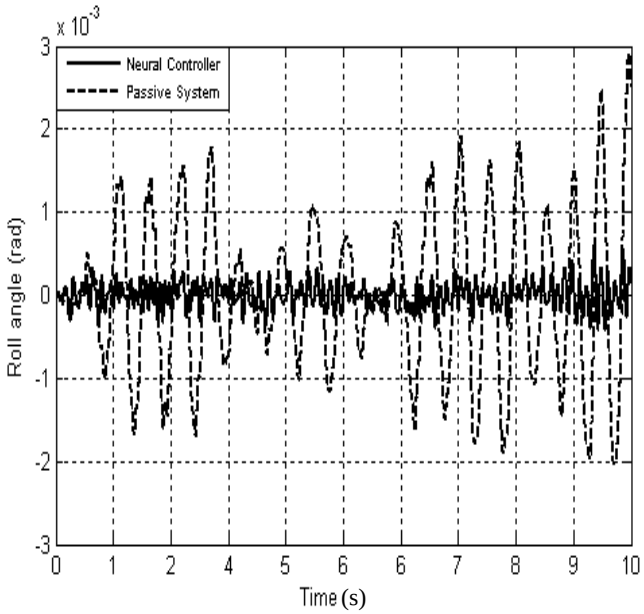


Figure 5.17 Time response of a roll angle

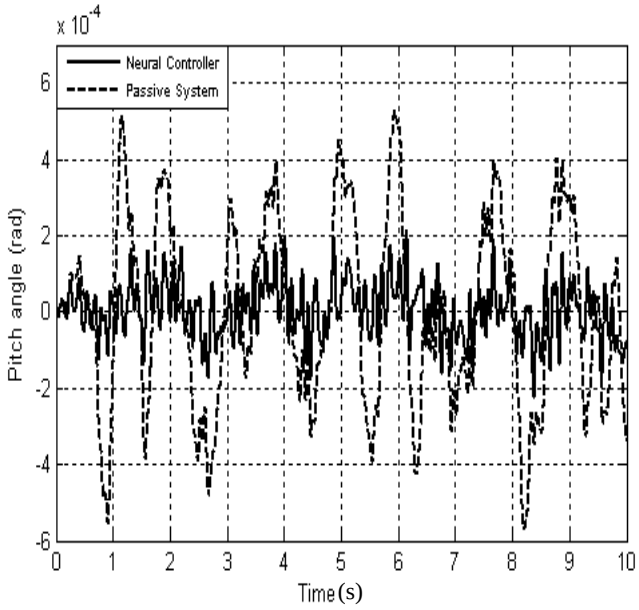


Figure 5.18 Time response of a pitch angle

Figures above show that the vertical displacement at each suspension point; the vertical displacement at the centre of gravity, the rolling and pitching movements of the controlled system have been minimized. It means the vibrations sensed by the passengers are minimized and the contact forces between the tyers and the road surface are maximized.

The bending torque has been applied (as shown in Figure 5.19) to test the robustness and effectiveness of the proposed neural controller. The random input has been selected as the road profile. Comparisons have been made between the outputs of the controlled system with the neural controller, the outputs of the controlled system with the FMRL controller and outputs of the passive system. The results are shown in Figures 5.20-5.26.

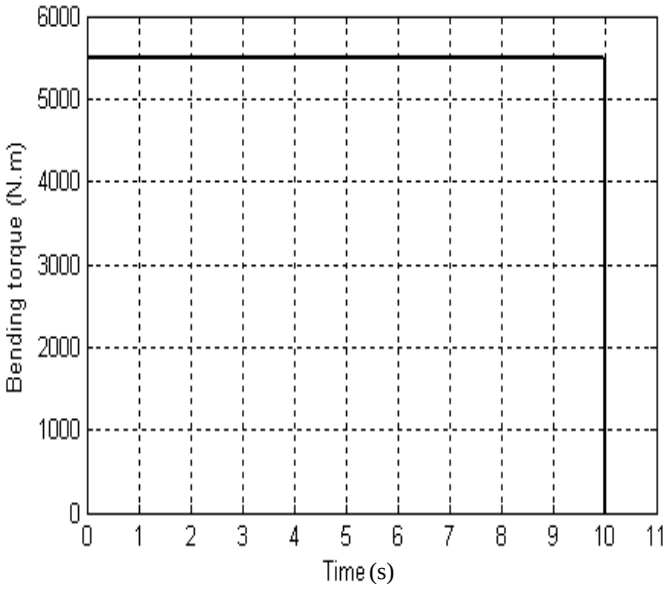


Figure 5.19 Bending torque

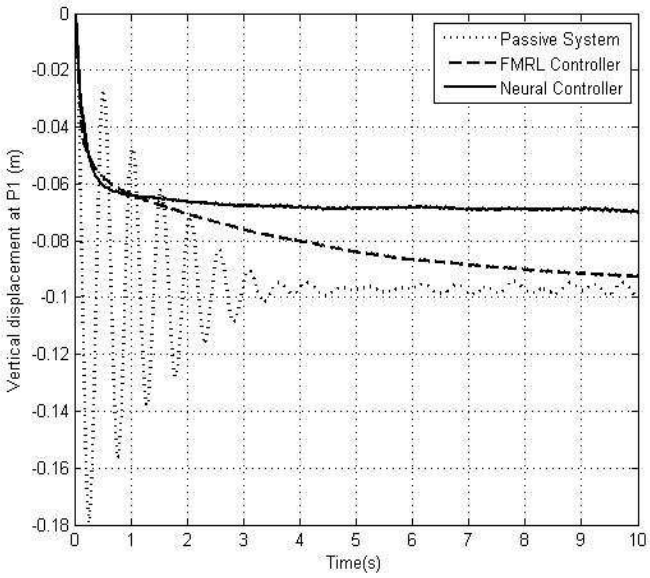


Figure 5.20 Time response of a vertical displacement at P_1

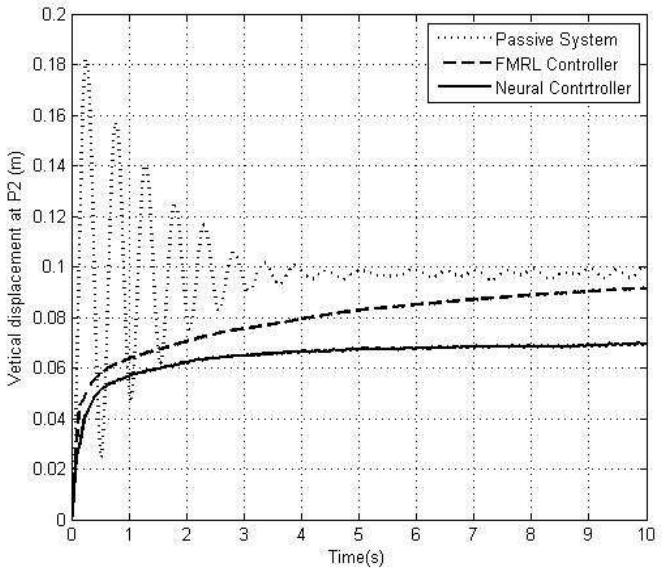


Figure 5.21 Time response of a vertical displacement at P_2

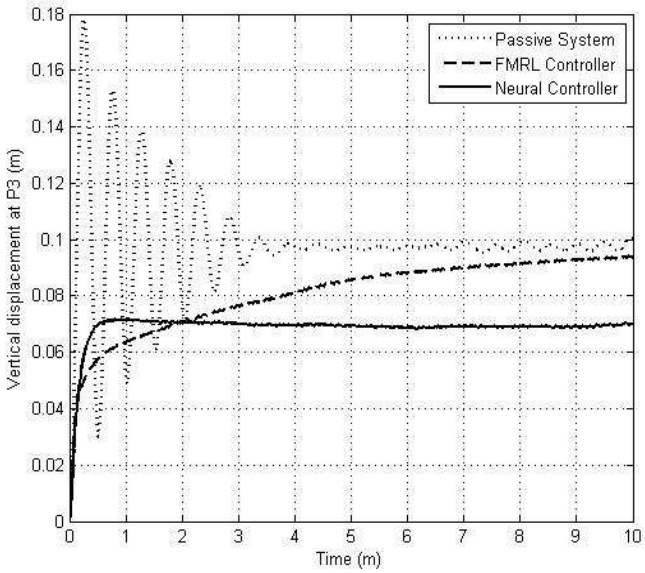


Figure 5.22 Time response of a vertical displacement at P_3

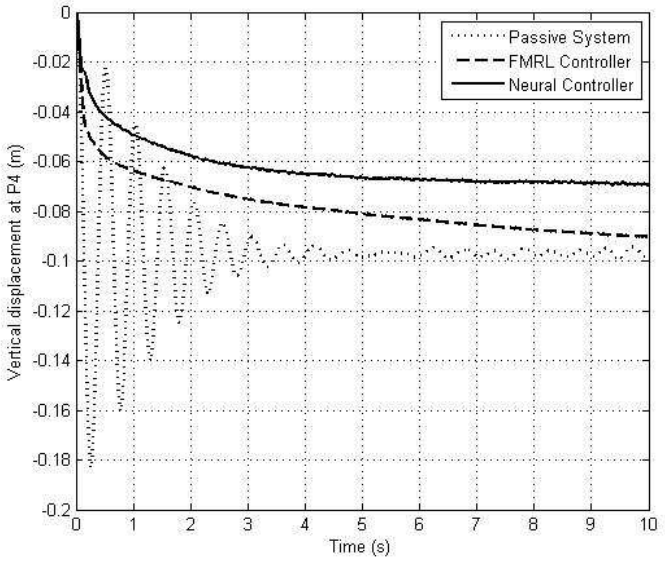


Figure 5.23 Time response of a vertical displacement at P_4

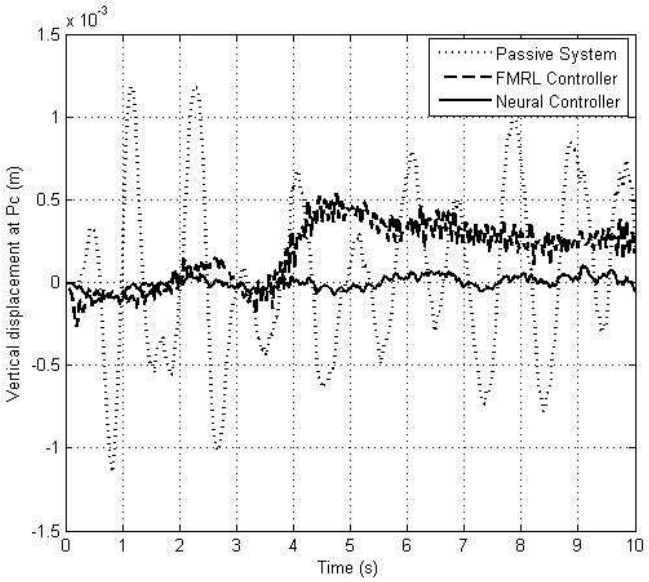


Figure 5.24 Time response of a vertical displacement at P_c

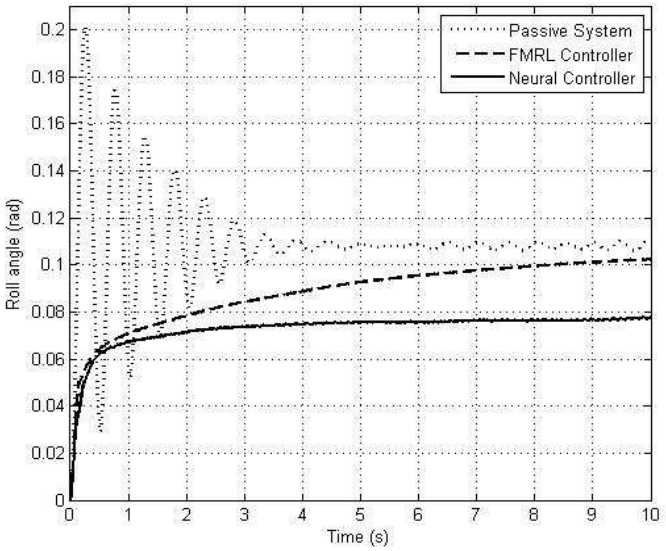


Figure 5.25 Time response of roll angle

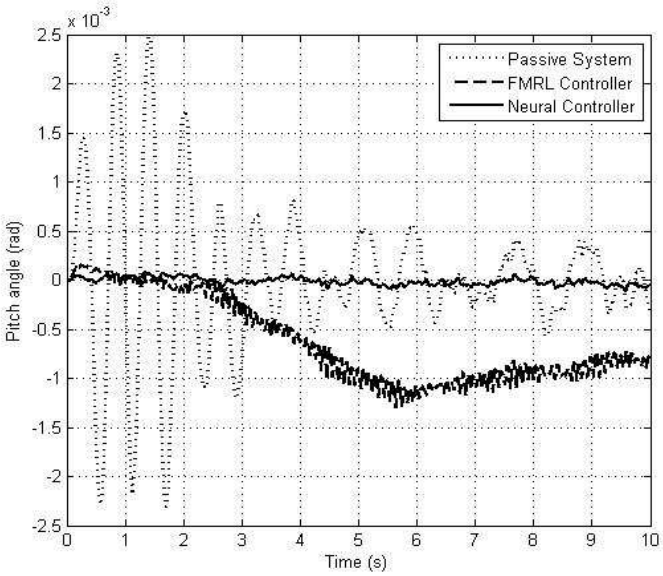


Figure 5.26 Time response of pitch angle

The aim of using the controller for the full vehicle nonlinear active suspension systems with hydraulic actuator is to force the outputs of the controlled system to reach small magnitudes. Figures above show that when neural controllers are used, the oscillations of the controlled system outputs are eliminated and the steady state responses are minimized. It means the neural controllers are effective controllers.

The braking torque has been applied (as shown in Figure 5.27) to test the robustness and effectiveness of the proposed neural controller. The random input has been selected as road profile. Comparisons have been made between the outputs of the process system with neural controller the corresponding outputs of the controlled system with the FMRL controller and passive system. The results are shown in Figures 5.28-5.34.

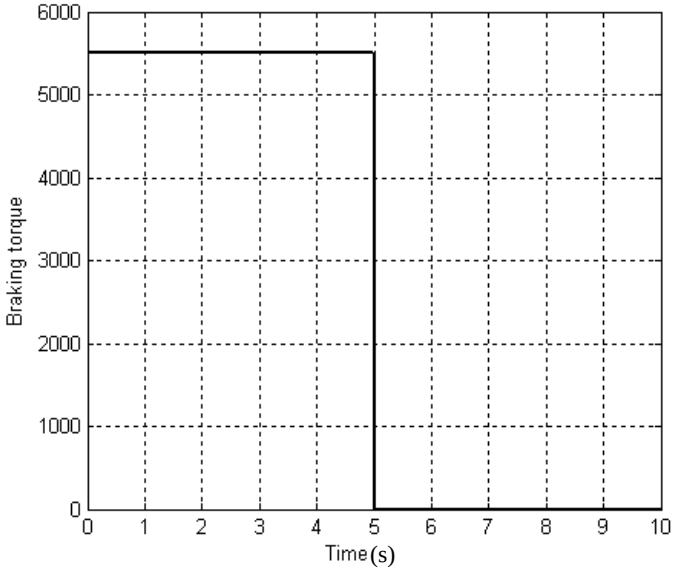


Figure 5.27 Braking torque

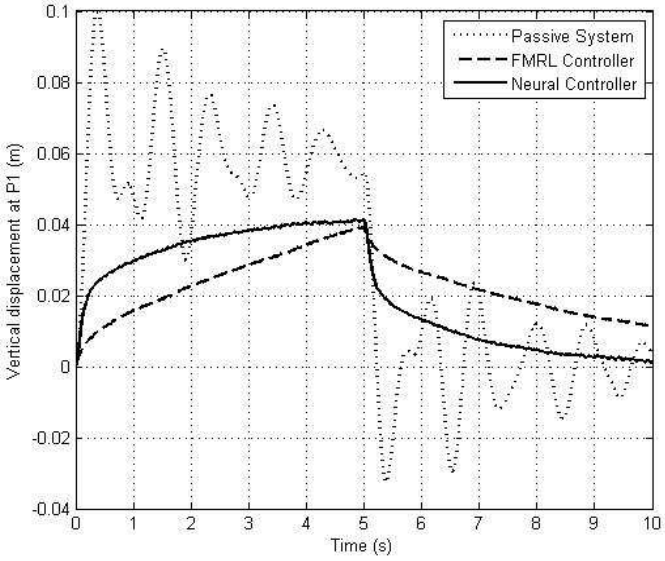


Figure 5.28 Time response of a vertical displacement at P_1

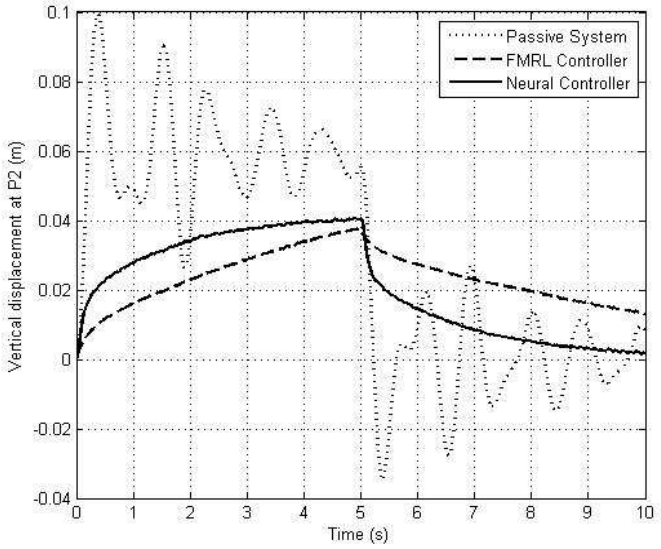


Figure 5.29 Time response of a vertical displacement at P_2

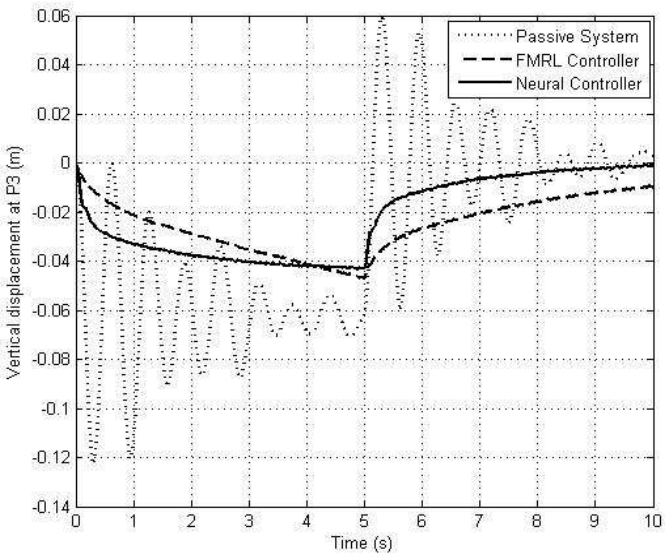


Figure 5.30 Time response of a vertical displacement at P_3

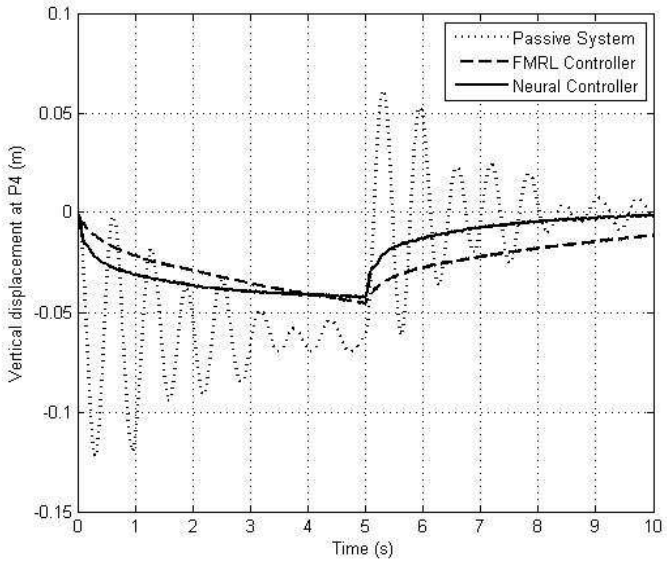


Figure 5.31 Time response of a vertical displacement at P_4

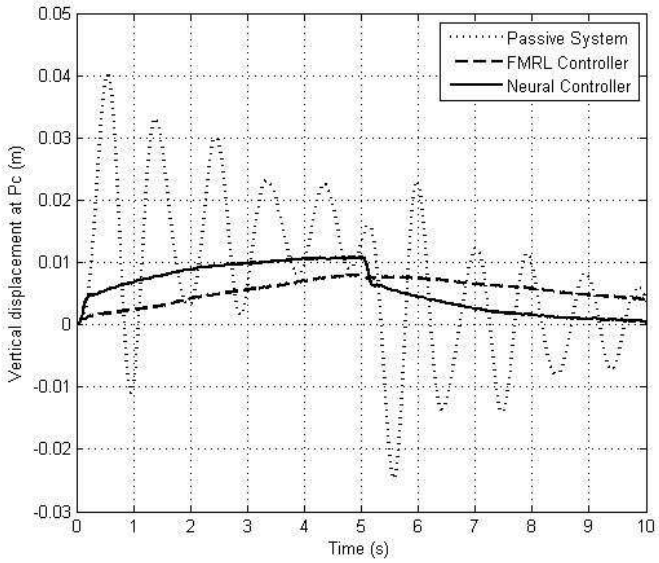


Figure 5.32 Time response of a vertical displacement at P_c

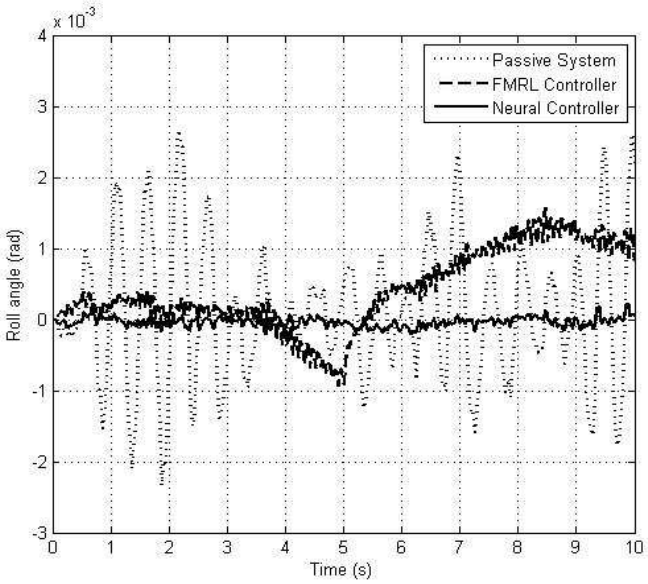


Figure 5.33 Time response of pitch angle

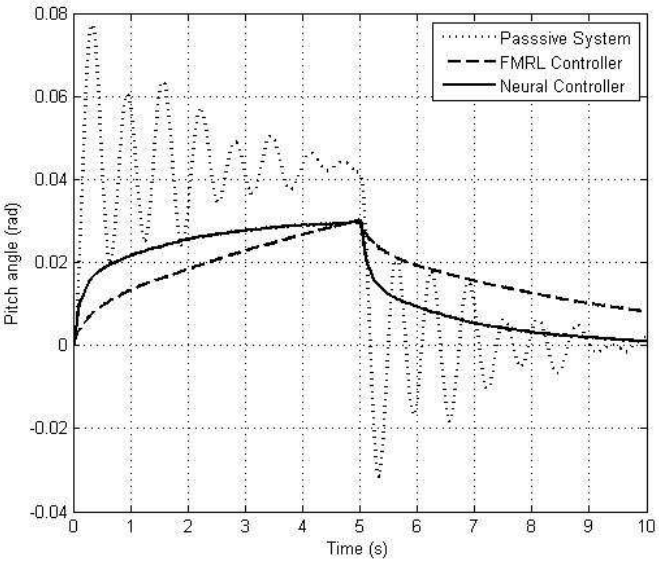


Figure 5.34 Time response of roll angle

The figures above show that when the neural controllers are used, the oscillations of the controlled system outputs are reduced to zero and the steady state responses are minimized.

5.7 Robustness of the proposed neural controller

Six types of disturbances have been applied individually to test the robustness of the proposed neural controller. The same disturbances types that are used in Chapter 3 are used here to compare between the performances of controlled systems with proposed controller and passive system.

- Square wave with varying amplitude is applied as the input of road profile.
- Sine wave input signal with varying amplitude is applied as the input of road profile.
- Square wave with varying frequency is applied as the input of road profile.
- Sine wave input signal with varying frequency is applied as the input of road profile.
- Different values of bending inertia torque (T_x) are applied.
- Different values of breaking inertia torque (T_y) are applied.

Figures 5.35-5.40 show the comparison between the performance of plant with FMRLC and passive system under different disturbance types.

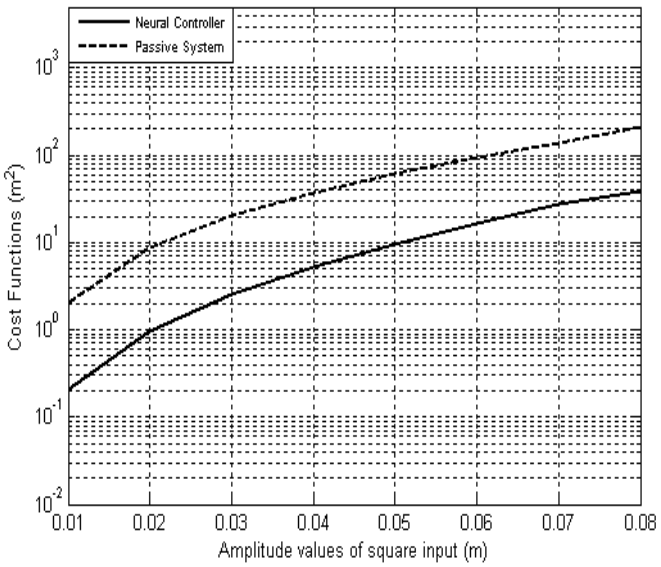


Figure 5.35 Time response of the cost functions against the different amplitude of square wave

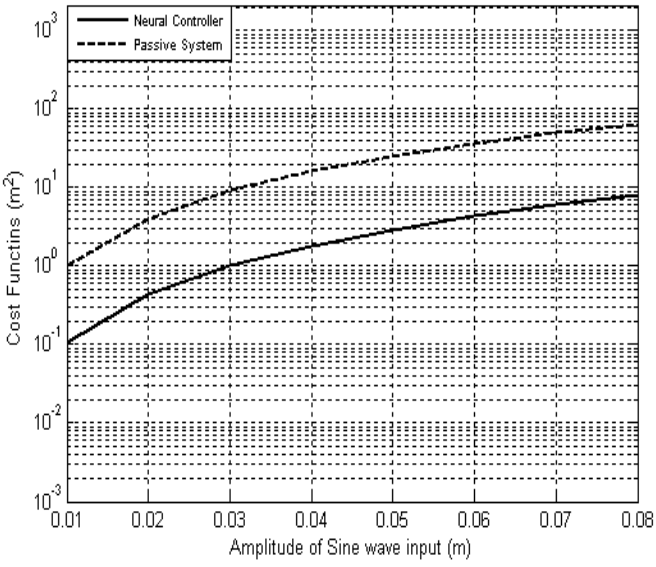


Figure 5.36 Time response of the cost functions against the different amplitude of sine wave

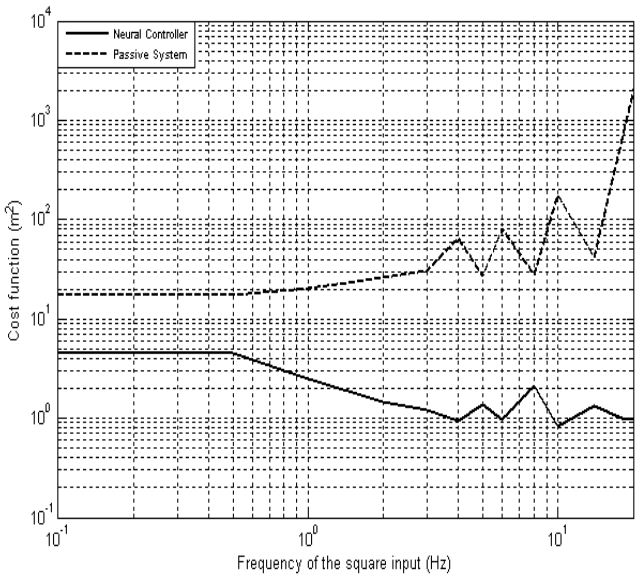


Figure 5.37 Time response of the cost functions against the different frequency values of square wave

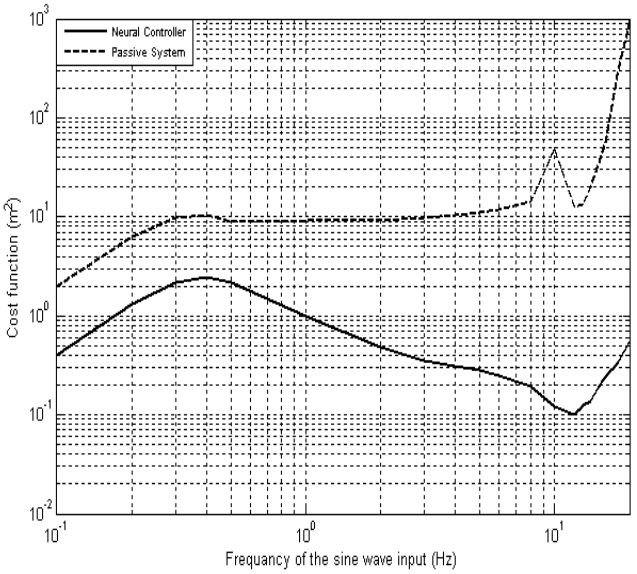


Figure 5.37 Time response of the cost functions against the different frequency values of sine wave

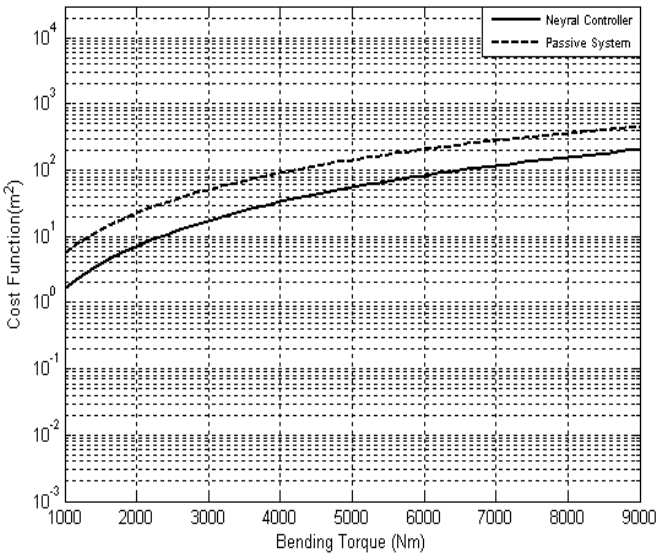


Figure 5.39 Time response of the cost functions against bending torque (T_x)

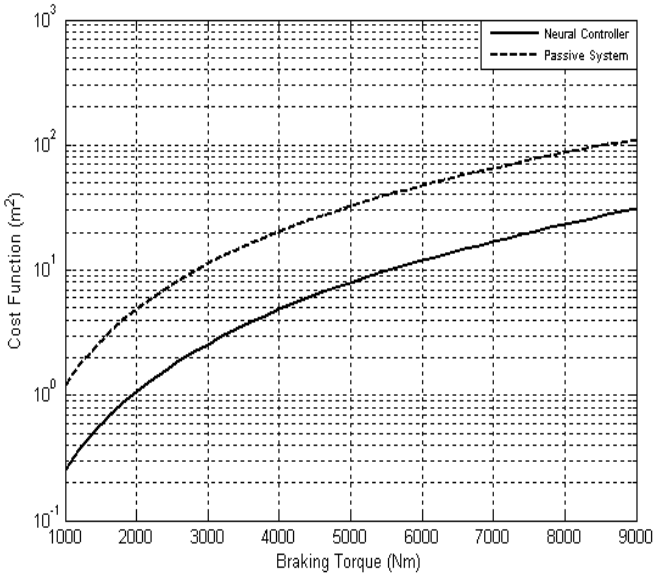


Figure 5.40 Time response of the cost functions against breaking torque (T_y)

Figures above show that the cost functions of the controlled system are improved when the neural controllers are used. If a comparison is made between the cost function of the controlled system in this Chapter and the cost functions in Chapter 3 and Chapter 4, it can be

seen that the performances of the controlled system with neural network are better than the performances of the controlled system with the FOPID controller or FMRL controller. But the response of the cost function against bending torque should be more improved. Therefore, the Neurofuzzy controller will be designed for the full vehicle nonlinear active suspension systems with hydraulic actuators in the next chapter.

5.8 Summary

One of the main advantages of using a neural network as a controller is that the neural network is universal function approximations which learn on the basis of examples and may be immediately applied in an adaptive control system because of their capability to adapt real time.

Four neural controllers were designed for the full vehicle nonlinear active suspension systems with hydraulic actuators. The inputs and outputs data which were obtained from the design of an FOPID controller have been used to design the proposed neural controller. The trainable parameters of the neural controller have been obtained by using the Levenberg-Marquardt training algorithm. After obtaining the optimal values of the trainable parameters, the performance of the neural controller has been improved by adjusting the input and output scaling gains. The scaling gains of the neural controller have been adjusted by using 4-D Golden Section Search method. The performance of the proposed controller has been tested and the results show that the output responses of the vehicle system with proposed controller are better than the corresponding responses of the vehicle system with the FOPID controller or FMRL controller. When the bending torque is applied (or the braking torque applied) with random inputs as the road profile, the oscillation of the output responses of the controlled system are eliminated and the steady state of the output responses are minimized. The robustness of the proposed controller has been tested by applying six types of the typical

disturbances. The results show that the cost functions have been improved. But it is only slightly improved when different values of the bending inertia torque are applied. To sum up, when neural controller is used, the improvement of the cost function is better than the improvement of the cost function when the FOPID controller or FMRL controller is used.

6. Adaptive Neuro Fuzzy Inference System (ANFIS)

6.1 Neurofuzzy networks

The conventional PID controllers have fixed gains; therefore, they are very sensitive to the external disturbances, parameters variations and system nonlinearity. For those reasons, the researchers have suggested the adaptive controllers for nonlinear systems which do not need the exact mathematical model of the systems. The artificial neural networks and fuzzy logic systems are good examples for the adaptive systems. Artificial neural networks are good at recognizing patterns. However, they are not good at explaining how they reach their decisions. Fuzzy logic systems are good at explaining their decisions but they cannot automatically acquire the rules used to make those decisions. Furthermore, fuzzy system controllers are very fast control. But it is difficult to determine the optimal structures of those controllers such as the shape of the membership functions and the exact rule number. These problems have been a central driving force behind the creation of intelligent hybrid systems where two or more techniques are combined in a manner that overcomes the limitations of the individual techniques. The neurofuzzy system is a good example for such hybrid system. Therefore, by using the neurofuzzy system, the disadvantages of the fuzzy logic systems and the artificial neural networks will be omitted. The neuro-adaptive training techniques provide a method for the fuzzy modelling procedure to train information about a data set. This technique gives the fuzzy logic capability to compute the parameters of the membership

functions that allow the associated fuzzy inference system to track the given input and output data.

There are two principal types of neurofuzzy networks, which are commonly used to adjust the parameters of fuzzy system: Mamdani system and Takagi and Sugeno system (Negoita et al. 2005). Figure 6.1 illustrates the structure of the Mamdani system. The fuzzy *IF-THEN* rules of the Mamdani system can be written as the following form

$$\boldsymbol{IF}\ x_1\ is\ A_{1j}\ and\ x_2 is\ A_{2k}\ and\ \dots and\ x_n\ is\ A_{nl}\ \boldsymbol{THEN}\ y\ is\ B_r$$

Where $x_1, x_2, \dots, x_n$ are the input of neurofuzzy system, y is the output of the neurofuzzy system, $A_{1j}, A_{2k}, \dots, A_{nl}$ are the linguistic values of the input variables and B_r is the linguistic value of the output variable.

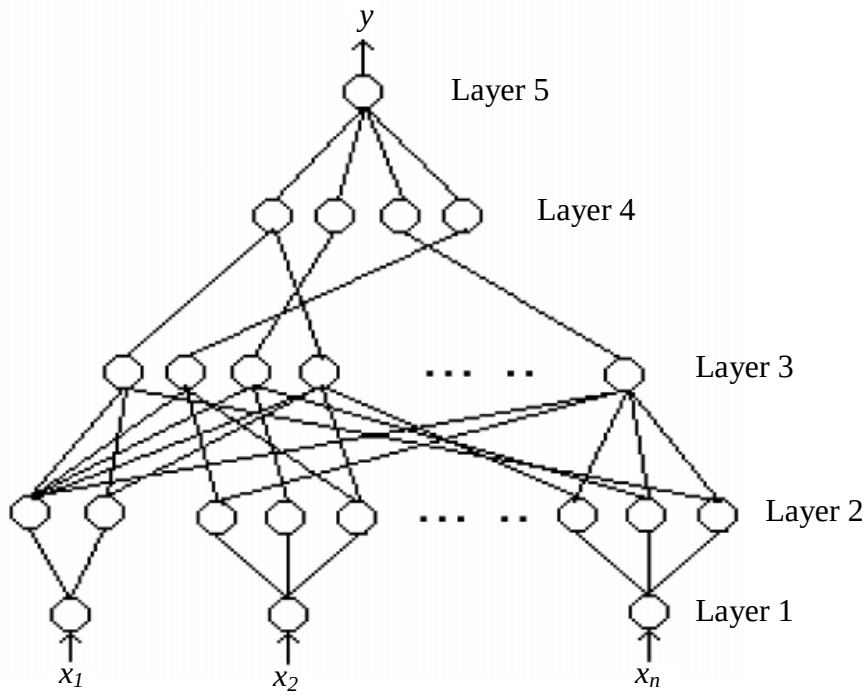


Figure 6.1 Mamdani System (Negoita et al. (2005))

The Mamdani system has five layers; each layer performs a specific task as shown in the following

Layer1: called input layer, in which the input variables applied to the system.

Layer2: called membership function layer. Each input variable is mapped to any values between 0 and 1, depend on the shape of member function in this layer.

Layer3: called rule-antecedent layer. Each node in this layer performs the precondition matching of the fuzzy rule.

Layer4: called rule-consequent layer. The conjunction operator (such as AND, OR) is performed to combine the fuzzy rules with the same consequent.

Layer5: called defuzzification layer. The fuzzy conclusion values are mapped to crisp outputs.

Figure 6.2 depicts the structure of the Takagi and Sugeno system. The first three layers are the same as those in the Mamdani system. In layer 4, input variables are used as input to this layer. Each node in this layer implements a linear model and the parameters of the linear model will be adjusted by using training algorithm. Layer 5 consists of single node that performs the defuzzification.

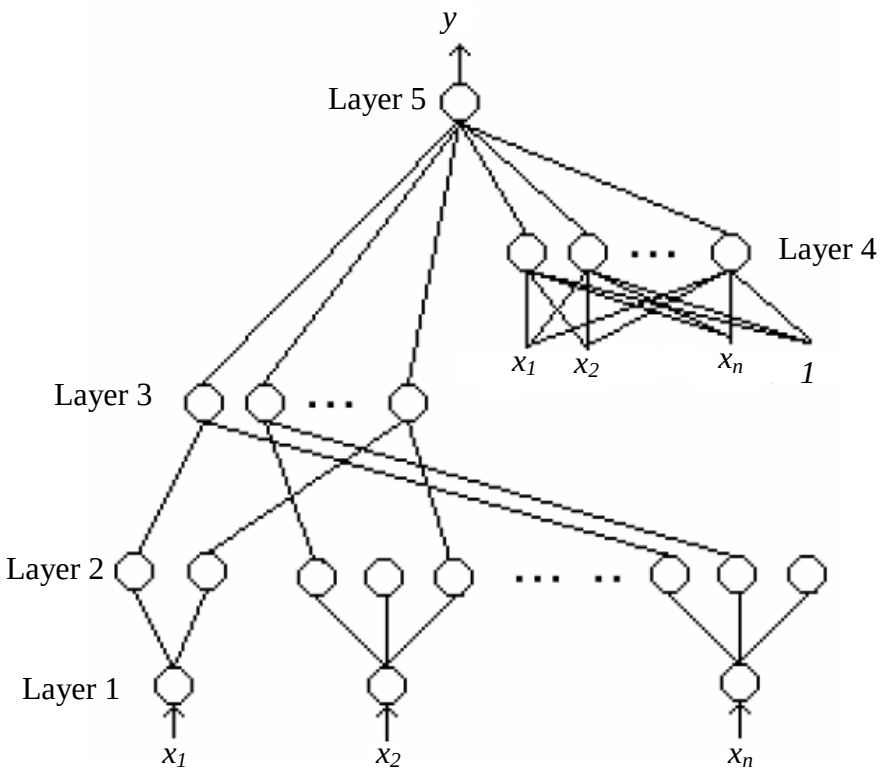


Figure 6.2 Takagi and Sugeno System (Negoita et al. (2005))

The fuzzy *IF-THEN* rules of the Takagi and Sugeno system can be written as the following form:

$$\boldsymbol{IF}\ x_1\ is\ A_{1j}\ and\ x_2 is\ A_{2k}\ and\ \dots and\ x_n\ is\ A_{nl}\ \boldsymbol{THEN}\ y = a_0 + \sum_{r=1}^{n} b_r x_r$$

where a_0 and b_r $(r = 1,2, \dots, n)$ are consequent parameters.

Usually crisp function is a polynomial in the input variables $\boldsymbol{x}$, but it can be any other functions that can appropriately described as the output of the system within the fuzzy region specified by the antecedent of the rule. Depending on the order of the polynomial; many types of Takagi and Sugeno system can be obtained. If the crisp function is a constant, the system is called zero-order Takagi and Sugeno model. If the crisp function is first order polynomial, the system is called first-order Takagi and Sugeno model and so on.

Some researchers have used the Mamdani System to design an effective neurofuzzy model. Zhang and Kandel used both control-oriented fuzzy neurons and decision-oriented fuzzy neurons to adjust the fuzzy membership functions adaptively by using a compensatory learning algorithm (Zhang and Kandel 1998). In Reference Kumar and Garg (2004), the fuzzy parameters were adjusted based on neural networks and genetic algorithm to design a control system for inverted pendulum. The neurofuzzy system has been used for early fault detection to help in minimizing quality and productivity offsets and to assist in averting hazardous consequences in abnormal situations (Gabbar et al. 2007). Figueiredo et al. (2004) introduced a class of neurofuzzy network and a constructive learning method based on competition of groups of neurons when the network receives inputs. Given a set of training data, the learning procedure automatically adjusts the input space portion to cover the whole space and finds the parameters of the membership functions for each input variable.

On the other hand, the Takagi and Sugeno systems have been used by several researchers to design neurofuzzy systems. Multivariable control of low head hydropower plants by using

neurofuzzy system has been presented in Reference Djukanovic et al. (1997). In Reference Rashidi and Vahedi (2004), the authors formulated a sensorless adaptive neurofuzzy speed controller for induction motor derives.

The Takagi and Sugeno systems show very good performances in the nonlinear function approximation problem, therefore, they are widely used as an approximation for the given input and output data. The Adaptive NeuroFuzzy Inference System (ANFIS) is a good example for Takagi and Sugeno system. Many researchers used ANFIS to design an intelligent control for different applications such as Areed et al. 2010, Gupta et al. 2009, Jang and Sun 1995 and Rezaeeian et al. 2008.

In this chapter, ANFIS based intelligent control will be designed for the full vehicle nonlinear active suspension systems with hydraulic actuators.

6.2 ANFIS structure

The ANFIS is a class of adaptive networks that acts as a fundamental framework for adaptive fuzzy inference systems (Jang and Sun 1995). It is one of the various methods that is used to organize the fuzzy inference system with given input and output data pairs. The drawbacks of fuzzy logic system and neural networks can be solved by using ANFIS which is a combination of a fuzzy logic controller and a neural network that makes the controller self-tuning and adaptive. By composing these two intelligent approaches, it will be achieve a good reasoning in quality and quantity. Since both fuzzy and neural systems are universal function approximators, therefore, their combination, the ANFIS, is also a universal function approximator.

The main concept of a ANFIS is derived from the human learning process, where an initial knowledge of a function is first setup by fuzzy rules and then the degree of function

approximation is iteratively improved by the learning capabilities of the neural network (Gupta et al. 2009).

The ANFIS is based mainly on the Takagi and Sugeno system (Jang and Mizutani 1996). It is used to design the fuzzy inference system with given input and output data pairs.

By consideration a first-order Takagi and Sugeno fuzzy inference system that contains *n* inputs variables, *K* rules and one output variable. The k^{th} rule of this system can be given as

$$R_k: \boldsymbol{IF}\ x_1 is\ A_{1k}\ and\ x_2 is\ A_{2k}\ and\ \dots and\ x_n\ is\ A_{nk}\ \boldsymbol{THEN}\ y = f_k(\boldsymbol{x})$$

where $f_k(\boldsymbol{x}) = p_{0k} + p_{1k}x_1 + p_{2k}x_2 + \cdots + p_{nk}x_n$.

Accordingly, in order to process a fuzzy rule by neural networks, it is necessary to modify the standard neural network structure. Figure 6.3 depicts the structure of multi-input single-output ANFIS.

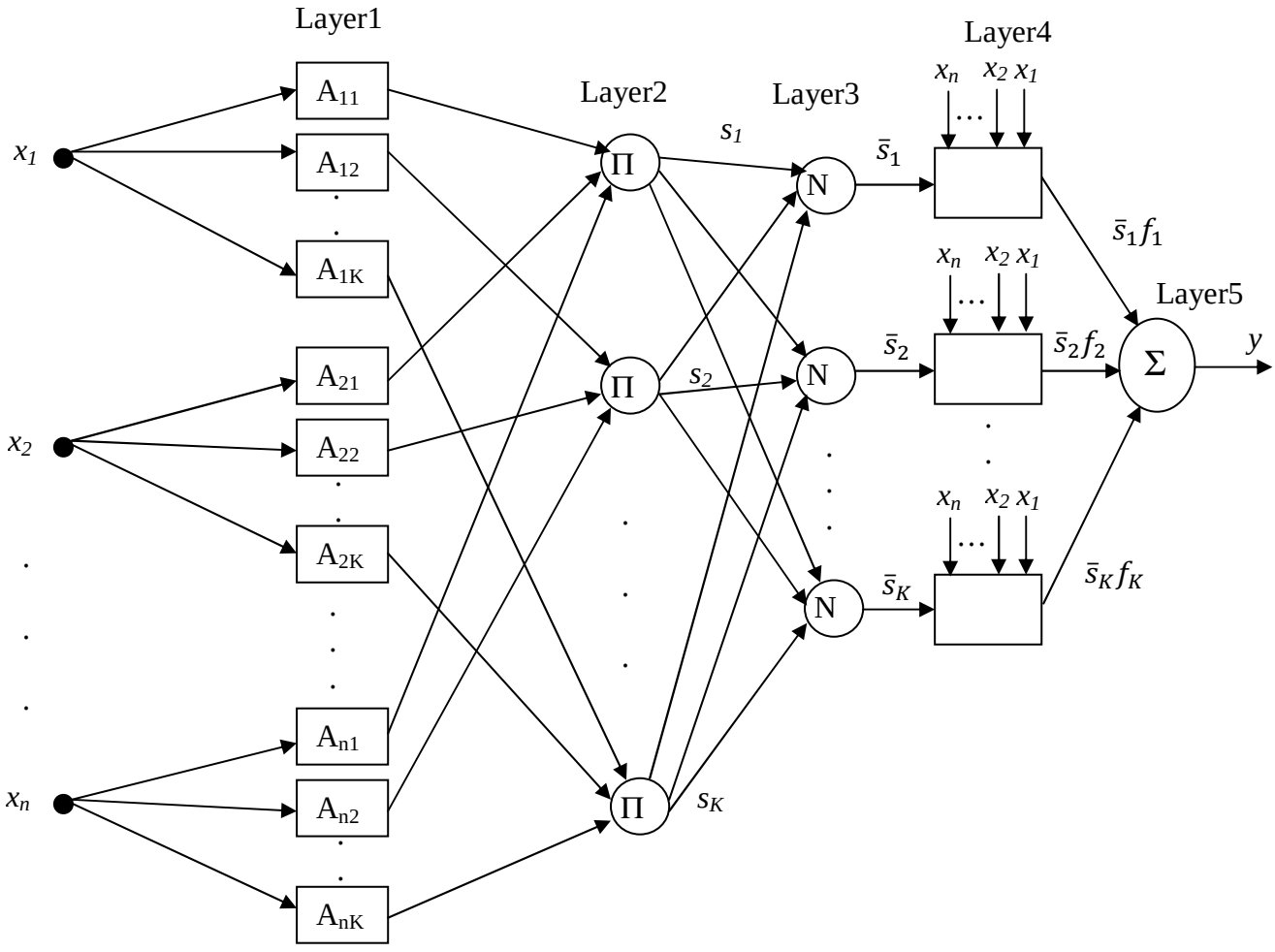


Figure 6.3 Multi Input Signal Output ANFIS Structure

Like the fuzzy models, the ANFIS has two main parts: premise part and consequent part. From Figure 6.3 the ANFIS has five layers, like the multilayer neural networks, each layer performs a specific task. The square nodes have adaptable parameters that will be adjusted during the training phase of the ANFIS while the circle nodes have fixed parameters. The output of the i^{th} node in the l^{th} layer is denoted by O_i^l, where every node in the same layer performs the same function as described below:

Layer 1: This layer is called fuzzification layer. Every node in this layer is a square node with the following node function:

$$O_r^1 = \mu_{A_{ik}}(x_i) \quad i = 1,2,\dots,n \ , k = 1,2,\dots,K \ and \ r = 1,2,\dots,n * K \tag{6.1}$$

where $\mu_{A_{ik}}(x_i)$ is the membership function of the r^{th} node . The $\mu_{A_{ik}}(x_i)$ has been chosen as the Bell-Shape membership function

$$\mu_{A_{ik}}(x_i) = \frac{1}{1 + \left[\left(\frac{x_i - c_{ik}}{a_{ik}}\right)^2\right]^{b_{ik}}}$$

where $\{a_{ik}, b_{ik}, c_{ik}\}$ are the parameters of membership function. They are called the premise parameters that will be adjusted in the training phase of the ANFIS.

Layer 2: This layer is called rule-antecedent layer. Every node in this layer is circle node (have no adaptive parameters) labelled Π which multiplies the incoming signals. The output of each node in this layer can be written as:

$$O_k^2 = s_k = \prod_{i=1}^{n} \mu_{A_{ik}}(x_i) \tag{6.2}$$

Layer 3: This layer is called normalization layer. Every node in this layer is a circle node labelled N. The output of the k^{th} node is the normalized k^{th} firing strength s_k. The output of any node in this layer can be given as:

$$O_k^3 = \bar{s}_k = \frac{s_k}{\sum_{l=1}^{K} s_l} \tag{6.4}$$

Layer 4: This layer is called rule-consequent layer. Every node in this layer is square node with linear function

$$f_k = p_{0k} + \sum_{i=1}^{n} p_{ik} x_i \tag{6.5}$$

where $\{p_{0k}, p_{1k}, p_{2k}, \ldots, p_{nk}\}$ ($k=1,2,\ldots,K$) are the set parameters of the polynomial function. They are called consequent parameters that will be adjusted in the training phase. The k^{th} output of this node is given as

$$O_k^4 = \bar{s}_k f_k = \bar{s}_k \left(p_{0k} + \sum_{i=1}^{n} p_{ik} x_i \right) \tag{6.6}$$

Layer 5: this layer called output layer or defuzzification layer. It computes the output of the ANFIS by summing up the outputs of layer 4. The node in this layer is a circle node labelled Σ and the output of this layer is given by the following equation:

$$y = O_1^5 = \sum_{l=1}^{K} \bar{s}_l f_l = \frac{\sum_{l=1}^{K} s_l f_l}{\sum_{l=1}^{K} s_l} \tag{6.7}$$

The trainable parameters of ANFIS, i.e. premise parameters and consequent parameters, should be adjusted to minimize the following performance function (Mean Squared Error):

$$MSE = 0.5 \sum_{m=1}^{M} E_m^2 \tag{6.8}$$

where M is the total number of training data set and E_m is the error signal between the desired output of m^{th} data and the actual output of ANFIS model of m^{th} data. The error signal E_m is given as

$$E_m = T_m - y_m \tag{6.9}$$

where T_m the m^{th} desired output and y_m the m^{th} actual output of the ANFIS model.

The main purpose of the ANFIS modelling is to achieve a set of local input-output relations that describe a process.

Commonly, the problem of system modelling requires two main stages: the structure identification and the parameter optimization. The structure identification deals with the problem of determining the input-output space partition and how many rules must be used by the ANFIS model. The parameter optimization is responsible for finding the optimum values of the trainable parameters of the ANFIS model, i.e., the parameters of the membership function in the premise part and the coefficients of the polynomial function in the consequent part (Kim and Lee 2003). A number of algorithms are proposed for adjusting the trainable parameters of the ANFIS such as the Gauss-Network algorithm, the Levenberg-Marquardt algorithm, the extended Kalman filter algorithm, Hybrid Training algorithm and Kohonene's map algorithm (Jang 1996; Jang and Mizutani 1996). In this chapter Hybrid Training algorithm will be used to adjust the trainable parameters of the ANFIS model.

6.3 Hybrid training algorithm

The output equation of the ANFIS can be rewritten as follows:

$$y = \bar{s}_1 f_1 + \bar{s}_2 f_2 + \cdots + \bar{s}_K f_K = (\bar{s}_1)p_{01} + (\bar{s}_1 x_1)p_{11} + (\bar{s}_1 x_2)p_{21} + \cdots + (\bar{s}_1 x_n)p_{n1} + (\bar{s}_2)p_{02} + (\bar{s}_2 x_1)p_{12} + (\bar{s}_2 x_2)p_{22} + \cdots + (\bar{s}_2 x_n)p_{n2} + \cdots + (\bar{s}_K)p_{0K} + (\bar{s}_K x_1)p_{1K} + (\bar{s}_K x_2)p_{2K} + \cdots + (\bar{s}_K x_n)p_{nK} \quad (6.10)$$

From Eq. (6.10), the output equation is linear in the consequent parameters. Therefore the hybrid training algorithm can be applied to adjust the trainable parameters of the ANFIS model (Jang 1993).

Eq. (6.10) can be written as other form as below:

$$y = \theta_1 \Psi_1(\boldsymbol{X}) + \theta_2 \Psi_2(\boldsymbol{X}) + \cdots + \theta_\rho \Psi_\rho(\boldsymbol{X}) \quad (6.11)$$

where $\boldsymbol{X}=[x_1, x_2,\ldots,x_n]^T$, Ψ_1 , Ψ_2, …,Ψ_ρ are nonlinear functions; $\theta_1, \theta_2,\ldots, \theta_\rho$ are unknown parameters, i.e. consequent parameters, that will be adjusted; and ρ is the total number of the unknown parameters.

If the values of the premise parameters are known, the nonlinear function in Eq. (6.11) can be calculated for specific input data vector.

To identify the unknown parameters θ_i, a training data set $\{(\boldsymbol{X}_m, y_m), m = 1,2, \dots, M\}$ should be obtained. For each data pair, the Eq. (6.11) can be written as follows:

$$\begin{cases} \theta_1\Psi_1(\boldsymbol{X_1}) + \theta_2\Psi_2(\boldsymbol{X_1}) + \cdots + \theta_\rho\Psi_\rho(\boldsymbol{X_1}) = y_1 \\ \theta_1\Psi_1(\boldsymbol{X_2}) + \theta_2\Psi_2(\boldsymbol{X_2}) + \cdots + \theta_\rho\Psi_\rho(\boldsymbol{X_2}) = y_2 \\ \vdots \\ \theta_1\Psi_1(\boldsymbol{X_M}) + \theta_2\Psi_2(\boldsymbol{X_M}) + \cdots + \theta_\rho\Psi_\rho(\boldsymbol{X_M}) = y_M \end{cases} \tag{6.12}$$

By using matrix notation, Eq. (6.12) can be given as

$$\boldsymbol{H\theta} = \boldsymbol{Y} \tag{6.13}$$

where H is an $M \times \rho$ matrix it is given as:

$$\boldsymbol{H} = \begin{bmatrix} \Psi_1(\mathbf{X_1}) & \dots & \Psi_\rho(\mathbf{X_1}) \\ \vdots & \dots & \vdots \\ \Psi_1(\mathbf{X_M}) & \dots & \Psi_\rho(\mathbf{X_M}) \end{bmatrix},$$

$\boldsymbol{\theta}$ and Y are $\rho \times 1$ vector and $M \times 1$ vector respectively, and can be given as:

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_\rho \end{bmatrix} \text{ and } \boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}.$$

If the premise parameters and the training data set are known, then the matrix $\boldsymbol{H}$ and vector $\boldsymbol{Y}$ will be simultaneously known.

Eq. (6.13) should be modified by incorporating an error vector $\boldsymbol{e}$ to account for random noise or modelling error, as follows:

$$\boldsymbol{H\theta} + \boldsymbol{e} = \boldsymbol{Y} \tag{6.14}$$

The estimation values of the unknown parameters $\widehat{\boldsymbol{\theta}}$ which minimize the sum of squared error (SSE) should be obtained. The SSE equation is given as:

$$SSE(\theta) = \sum_{m=1}^{M} (y_m - h_m^T\theta)^2 = \boldsymbol{e}^T\boldsymbol{e} = (\boldsymbol{Y} - \boldsymbol{H\theta})^T(\boldsymbol{Y} - \boldsymbol{H\theta}) \tag{6.15}$$

By using the least squares estimator (LSE) method the optimum consequent parameters $\widehat{\boldsymbol{\theta}}$ can be given the following equation (Hsia 1977):

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{T}\boldsymbol{Y} \tag{6.16}$$

The hybrid training algorithm is used to modify the parameters of the ANFIS model as follows: The gradient descent method as in neural network can be applied to modify the premise parameters while the least square estimate method can be applied to adapt the consequent parameters (Jang 1993). In the forward pass of the hybrid learning algorithm, functional signals go forward (under the condition that the premise parameters are fixed) until layer four and the consequent parameters are identified by the least squares estimate. In the backward pass, the error rates propagate backward and the premise parameters are updated by the back propagation error algorithm that was described in Chapter 5.

6.4 Design of the neurofuzzy controller for active suspension system

As mentioned before, the neurofuzzy control structure is a combination of a fuzzy logic system and a neural network, which enables the controller to be self-tuning and adaptive. Figure 6.4 depicts the neurofuzzy controller with controlled system. The structure of the NF controller in this figure is described in Section 6.2.

Four neurofuzzy controllers will be designed in this chapter, one for each suspension system. Each controller has three inputs and one output. The input signals are error, derivative of the error and integral of the error are scaled by using input scaling gains. The output of the controller is scaled by using output scaling gain which has been applied as input to the suspension system.

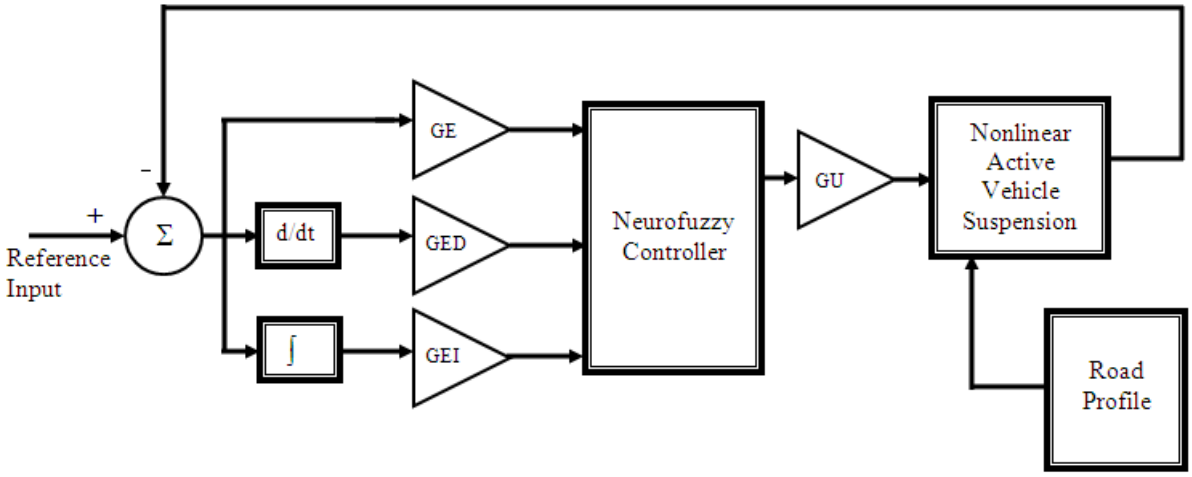


Figure 6.4 Neurofuzzy controller for full vehicle model

To find the optimal values of the NF parameters, i.e. premise and consequent parameters, to press the plant to meet all control objectives, the following steps have been carried out to design the NF controller:

1. The FOPID controllers are designed.
2. The input and output data that have been obtained from the FOPID controllers design should be used to train the trainable parameters of the NF controller using the hybrid training algorithm.
3. The input and the output scaling gains of the NF controller should be adjusted to improve the performance of the NF controller.

Firstly, the evolutionary algorithm has been used to design the FOPID controllers as discussed in Chapter 3. Then the input and output data that are obtained from the FOPID controllers should be used to design the parameters of the NF controller using the hybrid learning algorithm. Figure 6.5 shows the training phase of the NF controller. The random inputs have been applied as road profiles to obtain the input and output data set of the FOPID controller.

After obtaining the optimal trainable parameters of the NF, the NF controller design should be improved by adjusting the input and the output scaling gains (GE, GED, GEI and GU). To adjust the values of scaling gains, four-dimensional Golden Section Search (4-D GSS) method has been used.

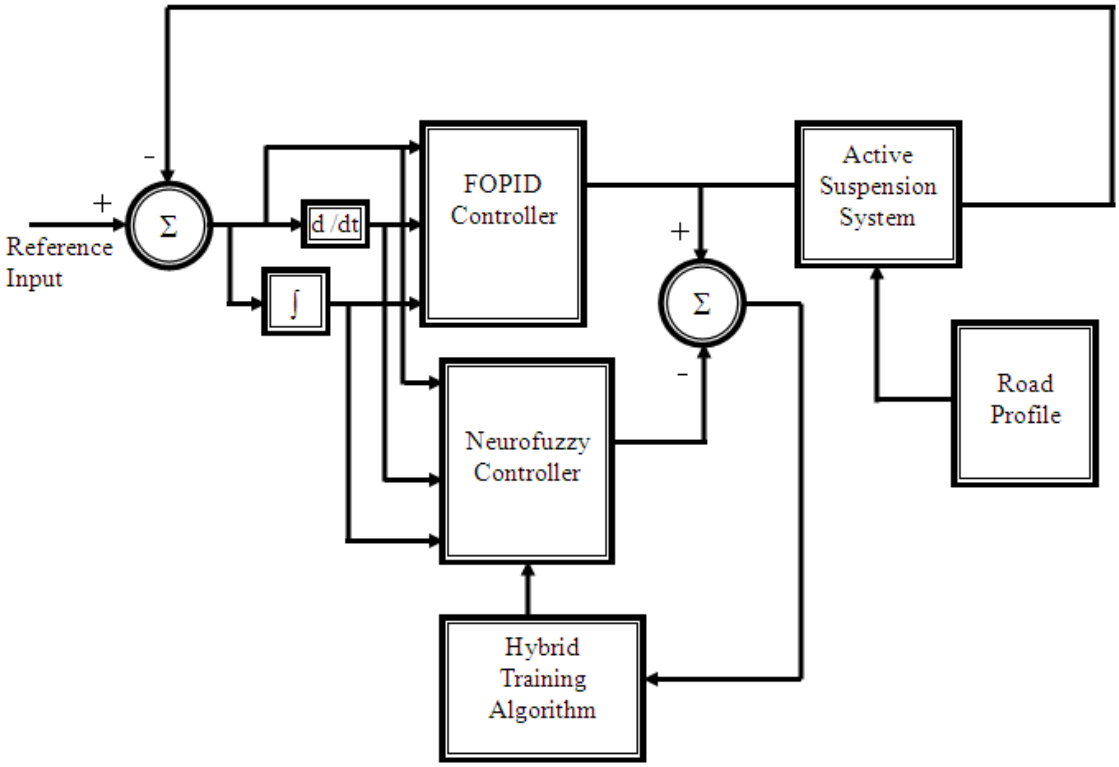


Figure 6.5 Training phase of neurofuzzy controller

6.5 Using MATLAB program to design the neurofuzzy controller for the controlled system

The ANFIS toolbox of MATLAB program has been used to adjust the trainable parameters of the neurofuzzy controllers. As shown in Chapter 3, the Evolutionary Algorithm applied to obtain the optimum values of the FOPID controller. The inputs and outputs data of the FOPID controllers has been applied as a training data set to train the parameters of the neurofuzzy controllers.

In this work, four NF controllers are designed, one for each suspension system. Each NF sub-controller has three inputs. The first input (Input 1) is the error between the derivative of the vertical displacement of the suspension where the NF sub-controller exists and the desired derivative of the vertical displacement. The second and the third input (Input 2 and Input 3) are the derivative and integral of this error respectively. The output of the NF sub-controller will be applied as a control signal for the hydraulic actuator. From the response of the input and output data of the FOPID controller with the time, the following rules have been obtained and they will be used as linguistic rules for the neurofuzzy controller:

R1: IF Input1 is NB and Input2 is NB and Input3 N Then y=f_1

R2: IF Input1is NS and Input2 is NS and Input3 P Then y=f_2

R3: IF Input1is ZE and Input2 is ZE and Input3 NCH Then y=f_3

R4: IF Input1is PS and Input2 is PS and Input3 P Then y=f_4

R5: IF Input1is PB and Input2 is PB and Input3 N Then y=f_5

where f_k is a the first order polynomial function and it can be given as

$f_k = p_{0k} + p_{1k} * Input1 + p_{2k} * Input2 + p_{3k} * Input3$ for $k=1,2,\ldots,5$. The coefficients of the polynomial function are called consequent parameters.

The Bell-Shape functions have been used as input membership functions for each NF sub-controller inputs. The parameters of input membership functions are called the premise parameters. Input1 has five grades: the negative big (NB), negative small (NS), zero (ZE), positive small (PS) and positive big (PB). Whereas Input2 has five grades: the negative big (NB), negative small (NS), zero (ZE), positive small (PS) and positive big (PB). While Input3 has three grades: the negative (N), no change (NCH) and positive (P).

From the ANFIS toolbox, the Hybrid Training Algorithm has been used as the optimisation method to obtain the parameters of neurofuzzy controllers. After 400 iterations, the errors between the outputs of the neurofuzzy controller and the outputs of the FOPID controllers reached to very small values as shown in Table 6.1.

Neurofuzzy Controller	Front-right	Front-left	Rear-right	Rear-left
Error	0.00361	0.00836	0.00539	0.00139

Table 6.1 The error between the output of the NF controller and FOPID

It means that for the same inputs, the output of the NF controller tracks the output of the FOPID controller. After the training task of the NF controllers is completed, the inputs and output scaling gains should be adjusted to improve the performance of the NF controllers. The 4-D Golden Section Search method has been used to adjust the scaling gains (i.e. the GE, GED, GEI and GU). As a result, the optimum values of scaling gains are 26, 22, 10 and 1.5, respectively.

After the full designing of the four neural controllers are completed, the Simulink model shown in Figure 6.6 is used to test the robustness of the proposed controller.

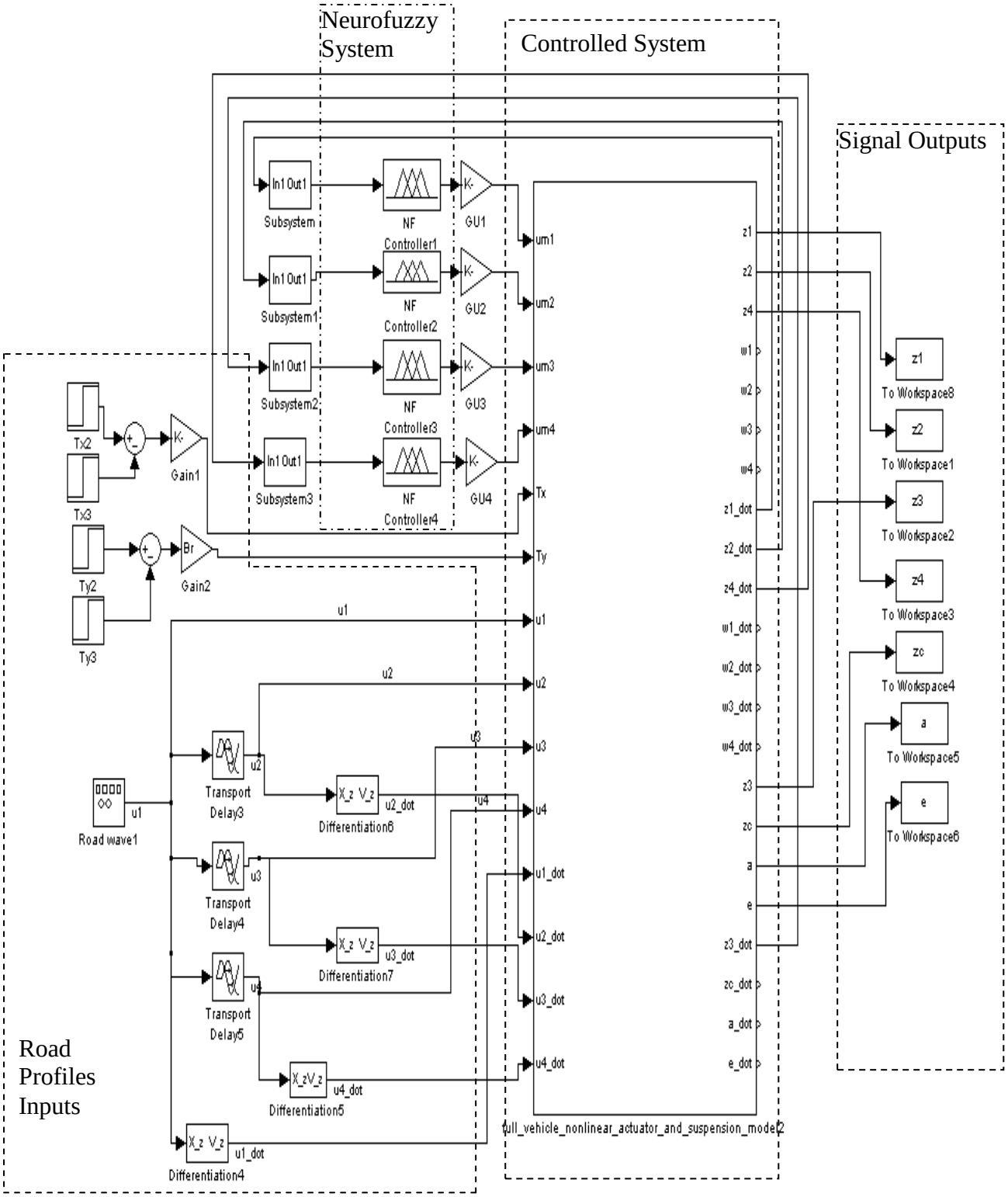


Figure 6.6 MATLAB SIMULINK model of controlled system with neurofuzzy controllers

The construction of the input subsystem is shown in Figure 6.7. This subsystem consists of the input scaling gains of the neurofuzzy controller and the reference input signal.

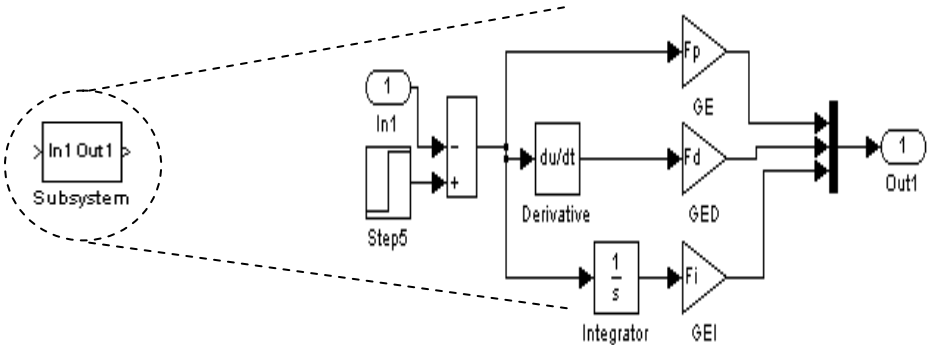


Figure 6.7 The structure of the input subsystem

As shown in Figure above, the neurofuzzy controller has three scaled inputs and one scaled control signal output. Suitable control signals of the neurofuzzy controllers have been applied as inputs to the suspension systems to control the position of the spool valves of hydraulic actuators. The hydraulic actuators generate appropriate forces to improve the performance of the vehicle system, i.e. riding comfort and road handling.

6.6 Simulation and results

The parameters of full vehicle nonlinear active suspension systems and hydraulic actuators (listed in Tables 2.1 and 2.2) have been used here to simulate the controlled system. Four neurofuzzy controllers have been designed, one for each suspension systems. To test the effectiveness of the neurofuzzy controllers, the outputs of the full vehicle system with proposed neurofuzzy controllers have been compared with the corresponding outputs of the full vehicle system without controller. Figures 6.8-6.14 show the comparison between the outputs of controlled process with neural controllers and the outputs of the passive system.

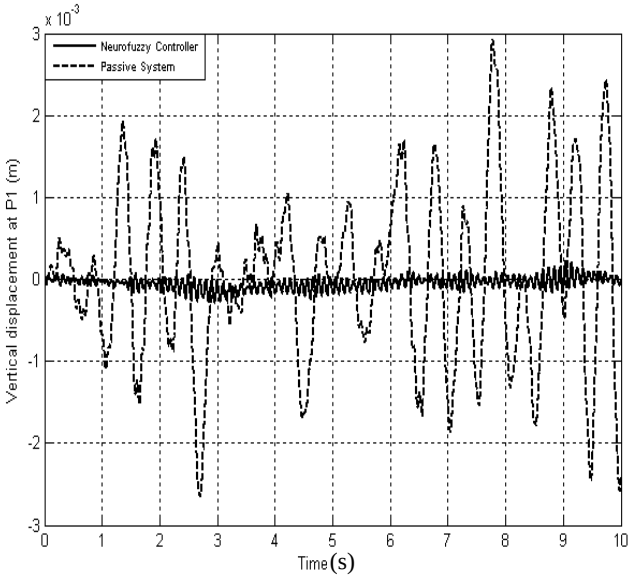


Figure 6.8 Time response of a vertical displacement at P_1

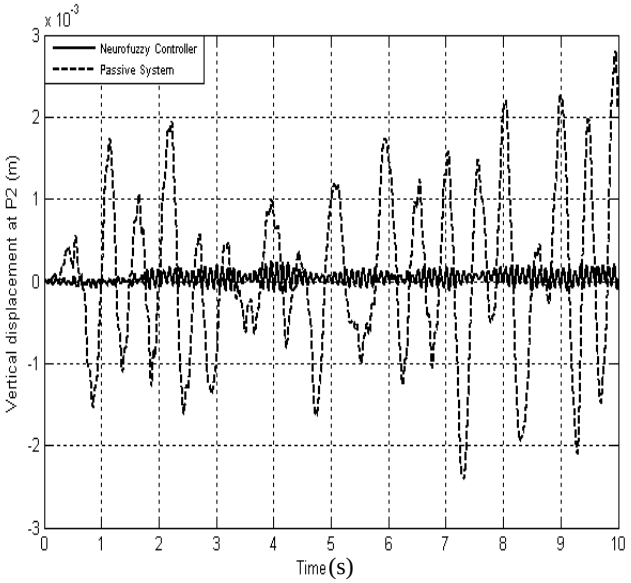


Figure 6.9 Time response of a vertical displacement at P_2

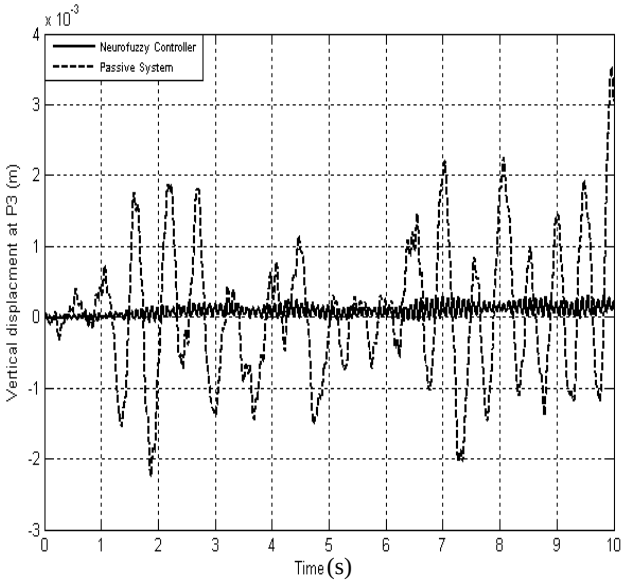


Figure 6.10 Time response of a vertical displacement at P_3

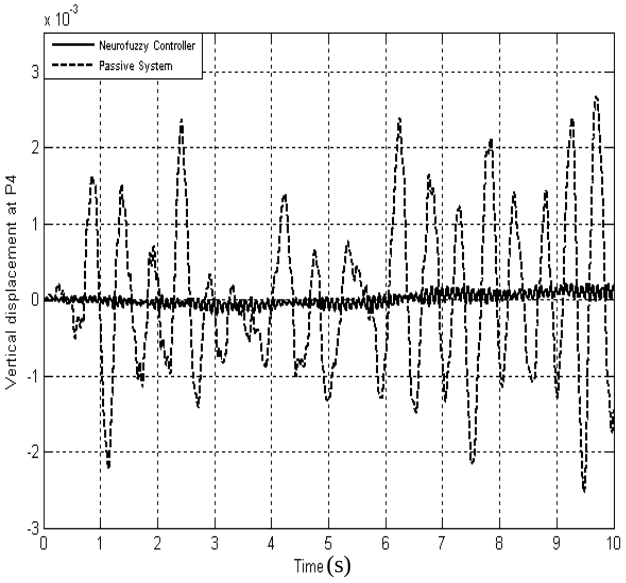


Figure 6.11 Time response of a vertical displacement at P_4

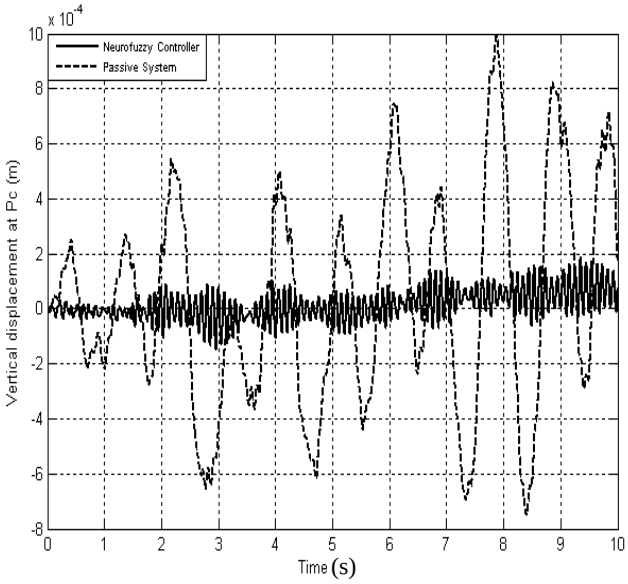


Figure 6.12 Time response of a vertical displacement at P_c

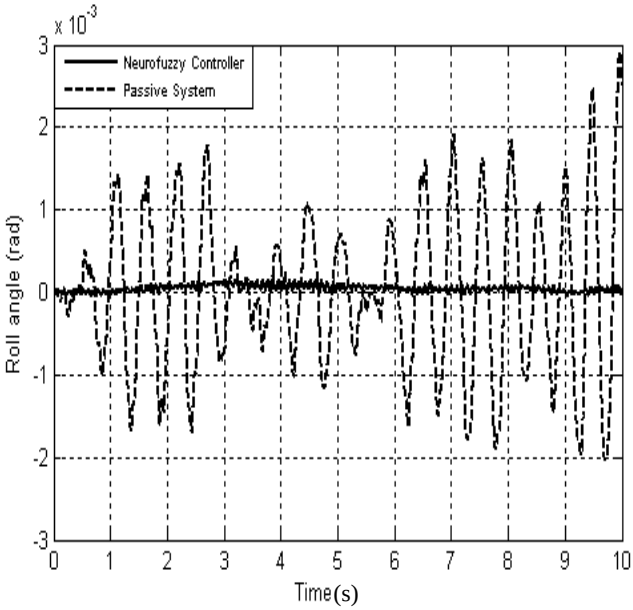


Figure 6.13 Time response of a roll angle

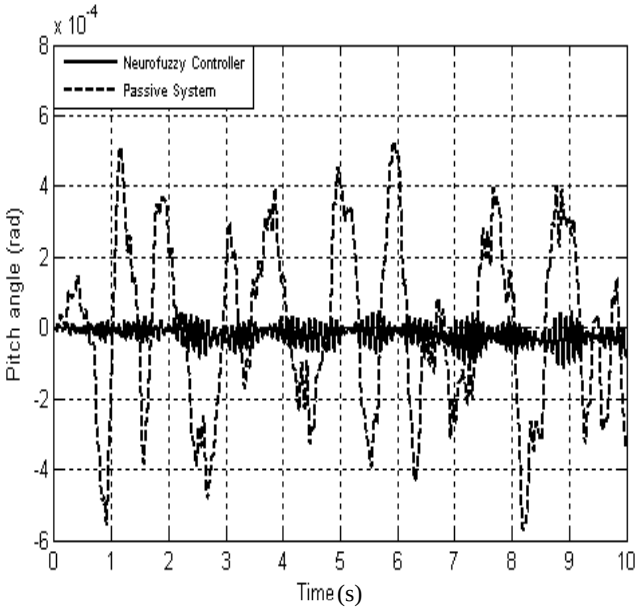


Figure 6.14 Time response of a pitch angle

Figures above show that the output responses of the controlled system have been improved which means the vibrations sensed by the passengers have been nearly omitted and the contact forces between the tyers and the road surfaces have been maximized.

The bending torque has been applied, as shown in Figure 6.15, with white a noise random input as the road profile to test the effectiveness of the neurofuzzy controller during sharp cornering. The Figures 6.16-6.22 show the comparison between the outputs of the suspension systems with neurofuzzy controllers, the outputs of the suspension systems with neural controllers and the corresponding outputs of the suspension systems without controllers.

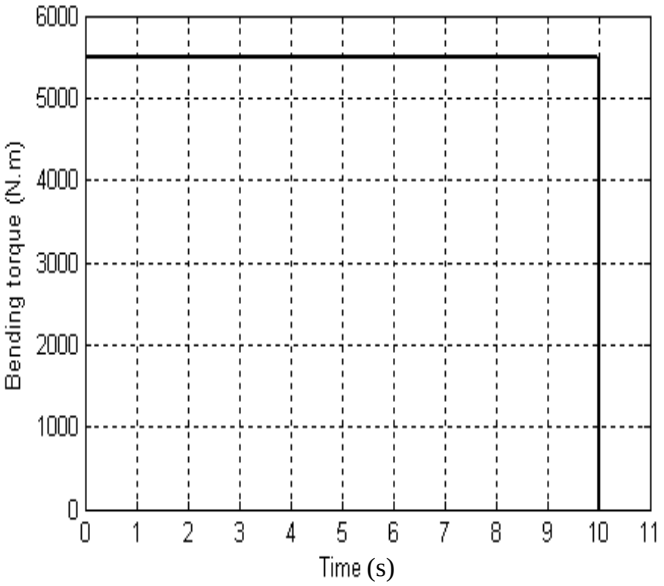


Figure 6.15 Bending torque

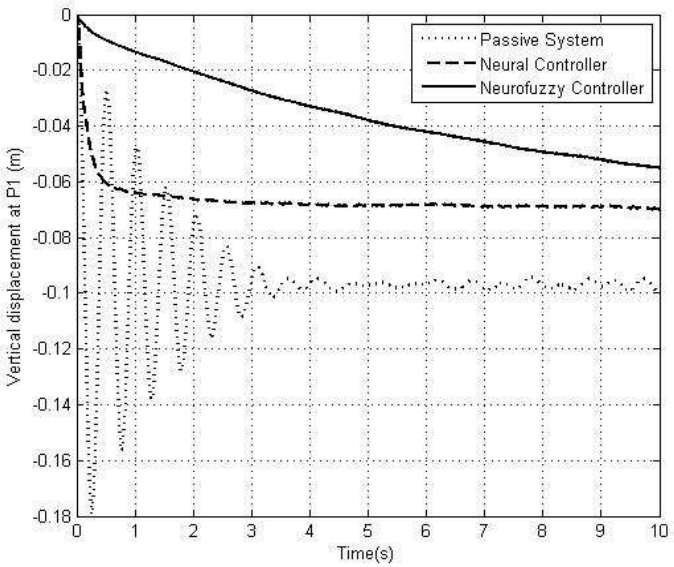


Figure 6.16 Time response of a vertical displacement at P_1

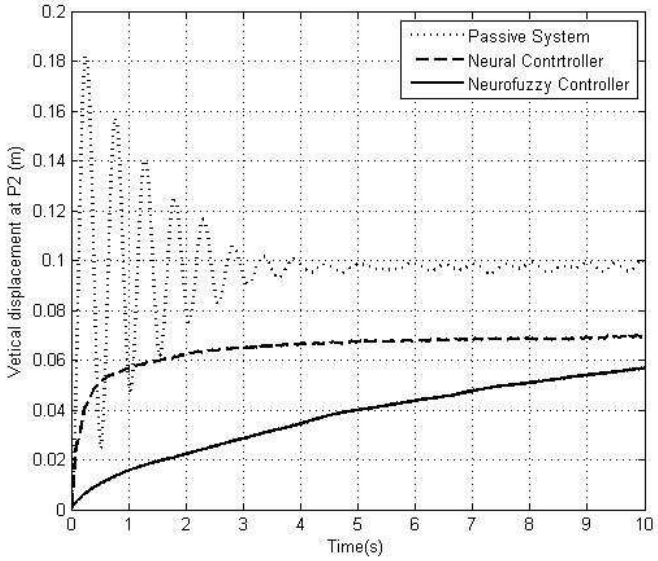


Figure 6.17 Time response of a vertical displacement at P_2

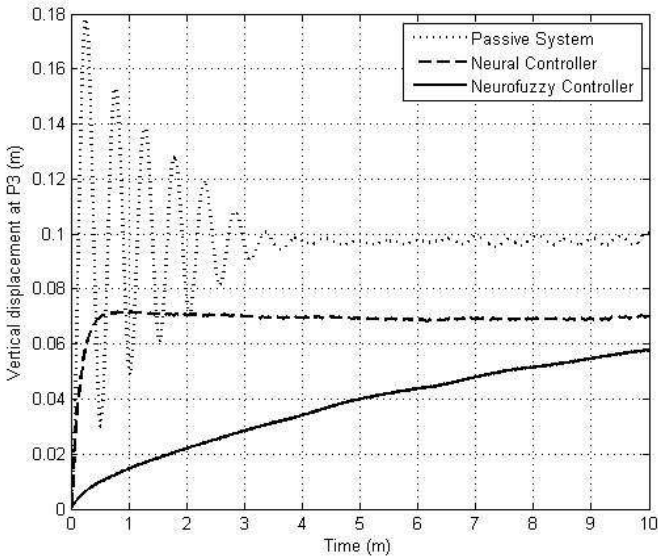


Figure 6.18 Time response of a vertical displacement at P_3

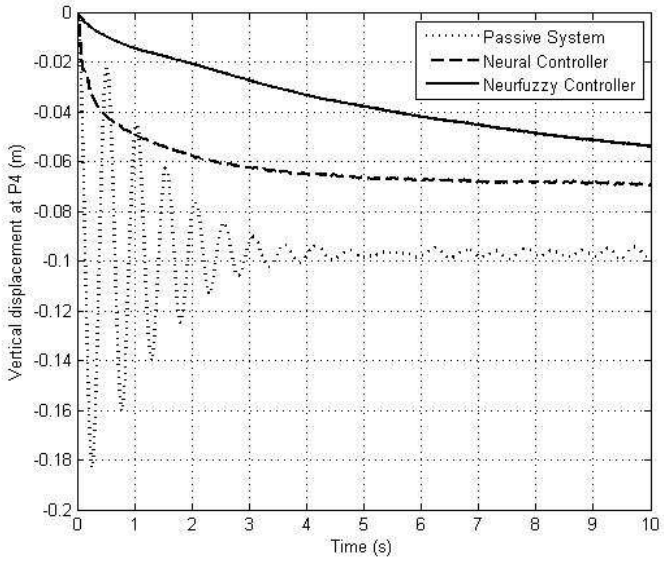


Figure 6.19 Time response of a vertical displacement at P_4

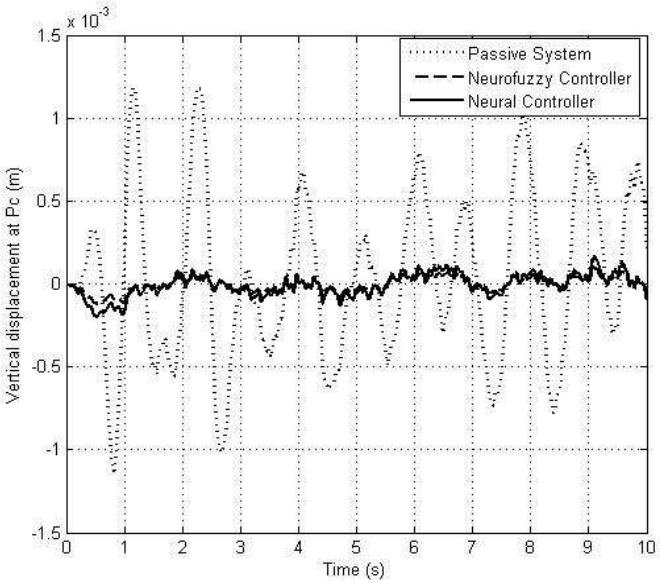


Figure 6.20 Time response of a vertical displacement at P_c

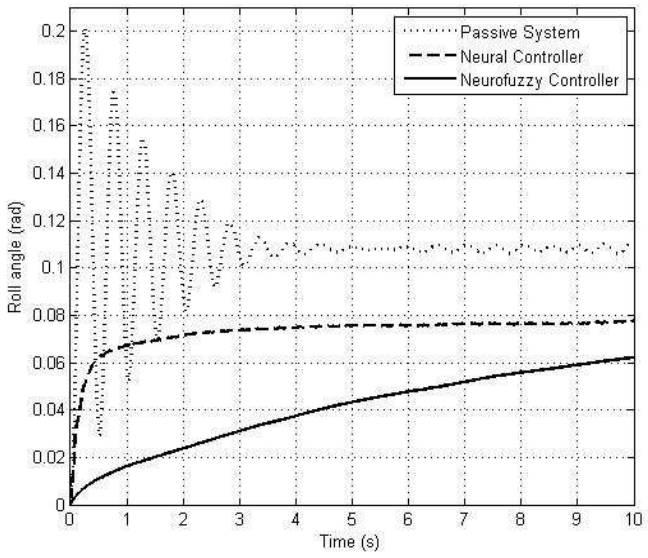


Figure 6.21 Time response of a roll angle

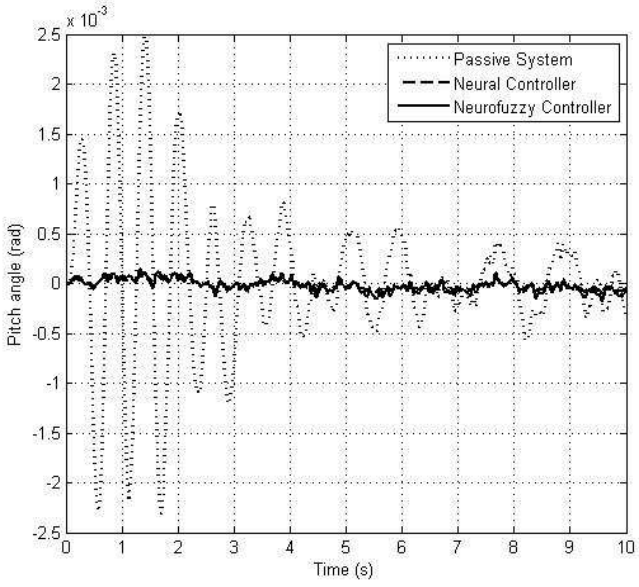


Figure 6.22 Time response of a pitch angle

Figures above show that the oscillations of the outputs responses of the controlled system have been eliminated and the final steady state of the outputs responses have been pressed to be small values.

To test the effectiveness of the neurofuzzy controller during sharp braking, the braking torque (shown in Figure 6.23) with a white noise random input has been applied as a road profile. The comparison has been made between the outputs of the suspension systems with neurofuzzy controller, the outputs of the suspension systems with neural controllers and the corresponding outputs of the suspension systems without controller as shown in Figure 6.24-6.30.

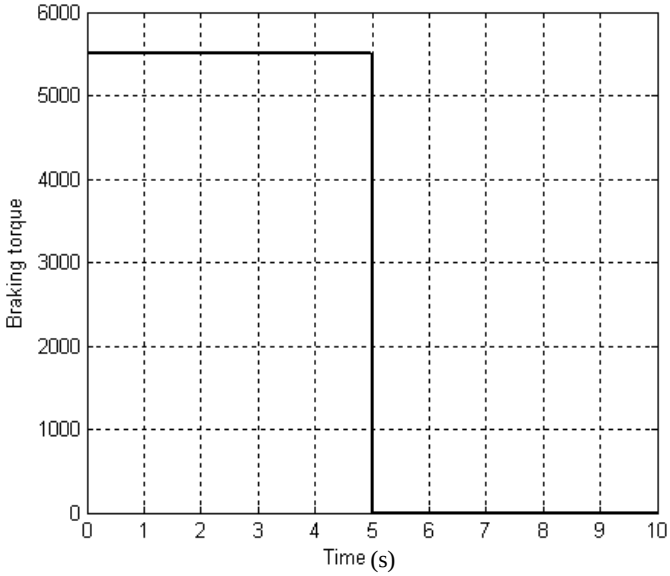


Figure 6.23 Braking torque

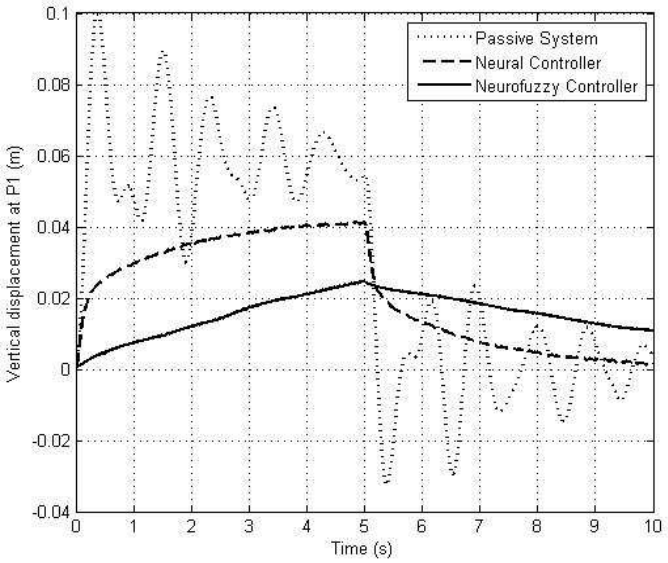


Figure 6.24 Time response of a vertical displacement at P_1

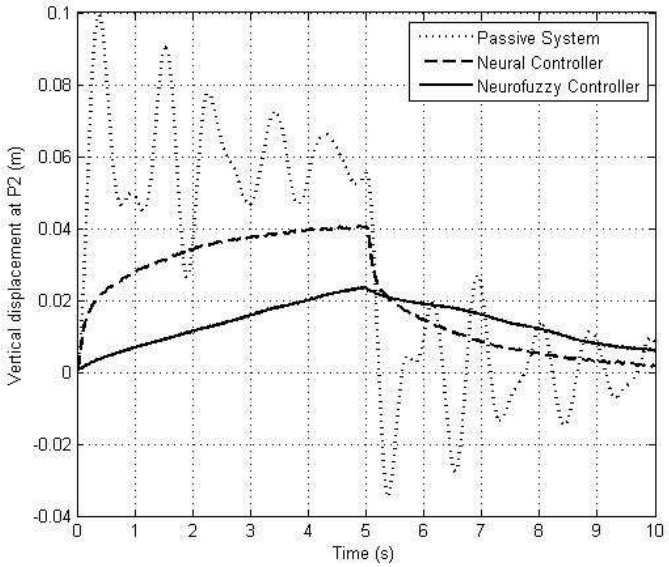


Figure 6.25 Time response of a vertical displacement at P_2

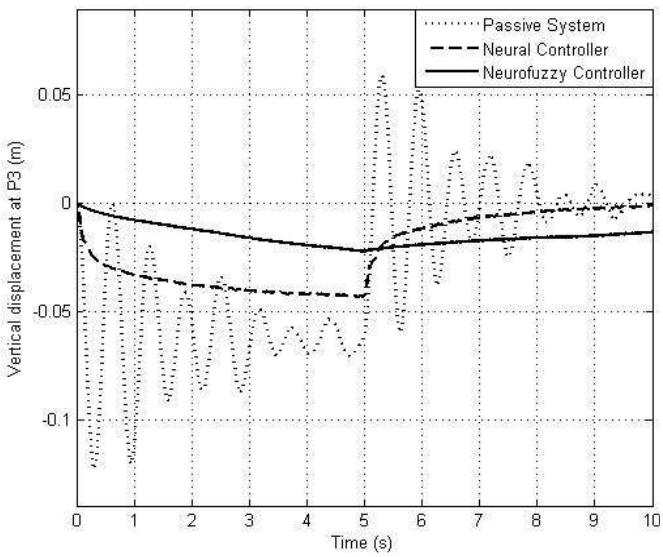


Figure 6.26 Time response of a vertical displacement at P_3

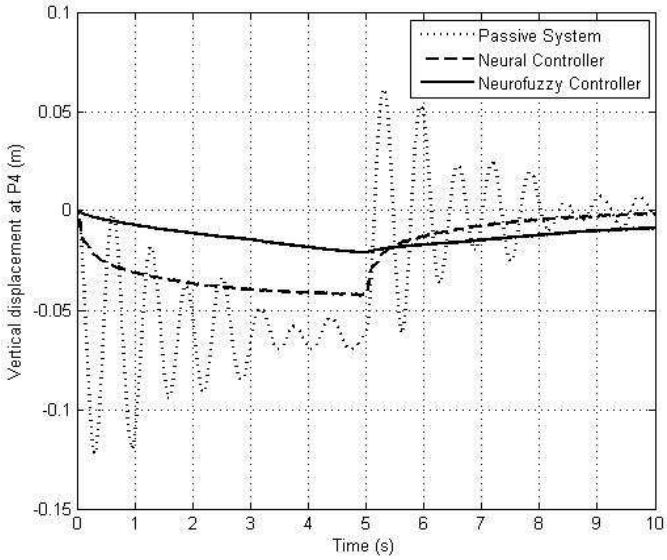


Figure 6.27 Time response of a vertical displacement at P_4

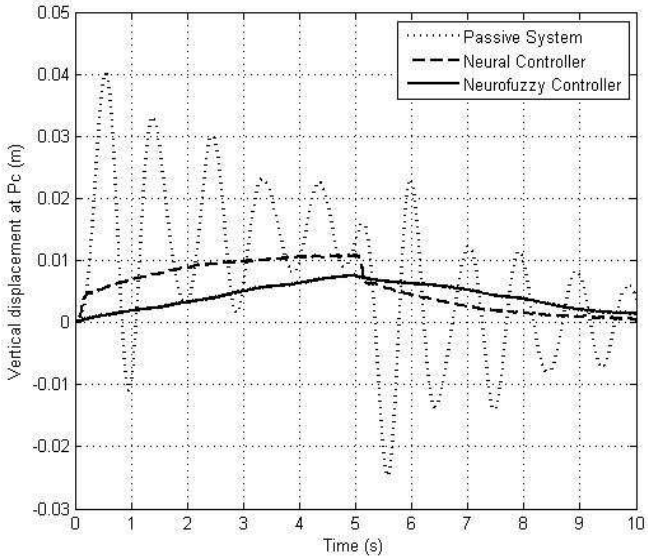


Figure 6.28 Time response of a vertical displacement at P_c

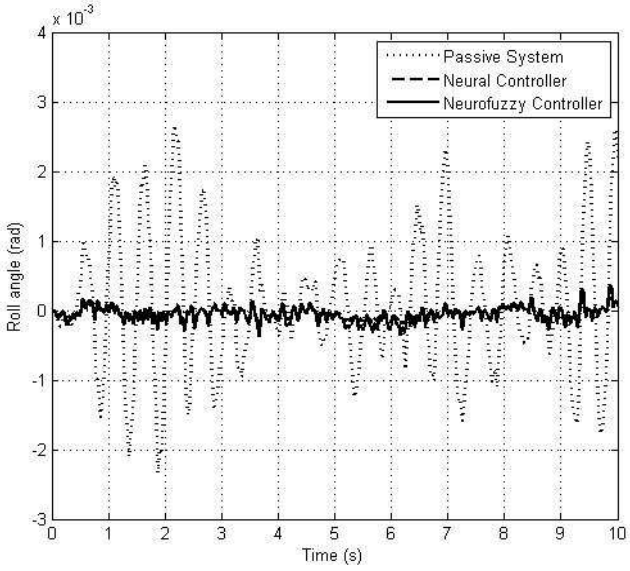


Figure 6.29 Time response of a roll angle

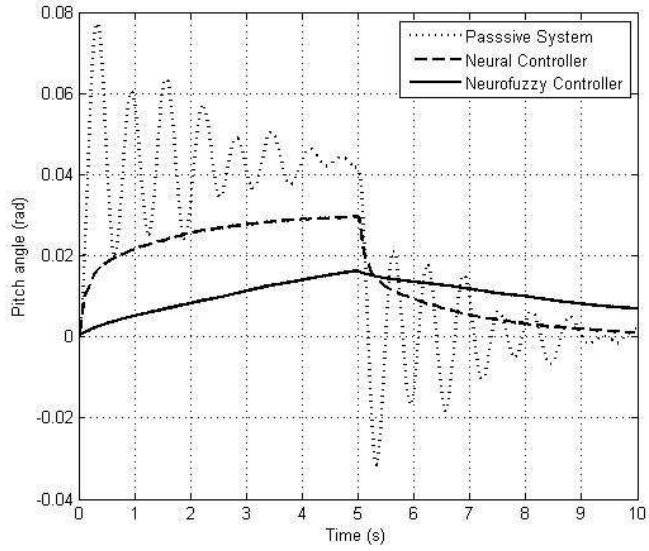


Figure 6.30 Time response of a pitch angle

Figures above show that the oscillations of the outputs responses of the controlled system have been eliminated and the final steady state of the outputs responses have been forced to small values.

By comparing between the outputs response of the controlled system with NF controller and the corresponding outputs response of the controlled system with other controller that are

designed in previous chapters, it can be seen that the NF controller is more effective and more robust than the other control types.

6.7 Robustness of the neurofuzzy controller

To test the robustness of the proposed system against the disturbances, six different types of disturbances have been applied individually. The same disturbances types that are used in previous chapters have been used in this chapter as listed below.

- *Square wave with varying amplitude is applied as the input of road profile.*
- *Sine wave input signal with varying amplitude is applied as the input of road profile.*
- *Square wave with varying frequency is applied as the input of road profile.*
- *Sine wave input signal with varying frequency is applied as the input of road profile.*
- *Different values of bending inertia torque (T_x) are applied.*
- *Different values of breaking inertia torque (T_y) are applied.*

Figures 6.31-6.36 show the comparison between the cost function of the controlled system with neurofuzzy controller and the passive system against the disturbances.

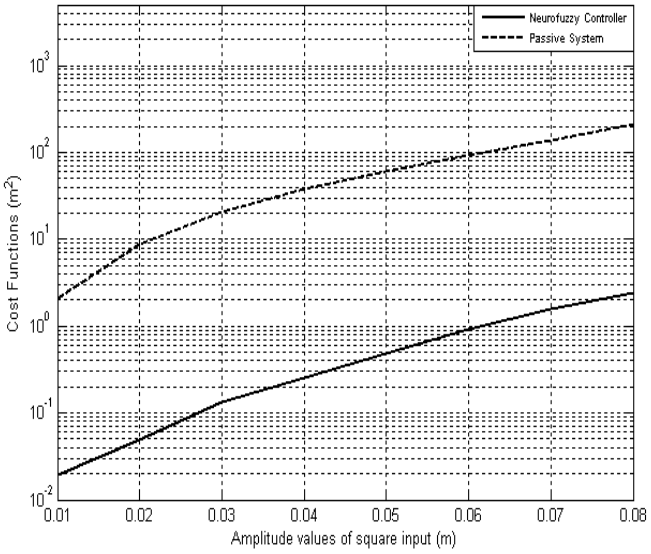


Figure 6.31 Time response of the cost functions against the different amplitude values of square wave

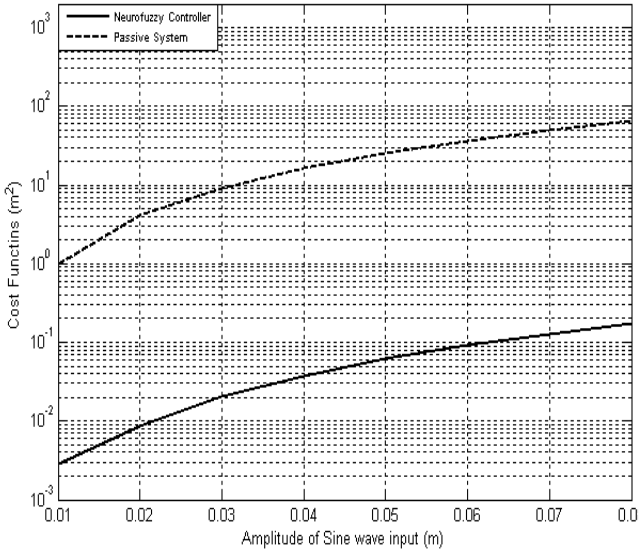


Figure 6.32 Time response of the cost functions against the different amplitude values of sine wave

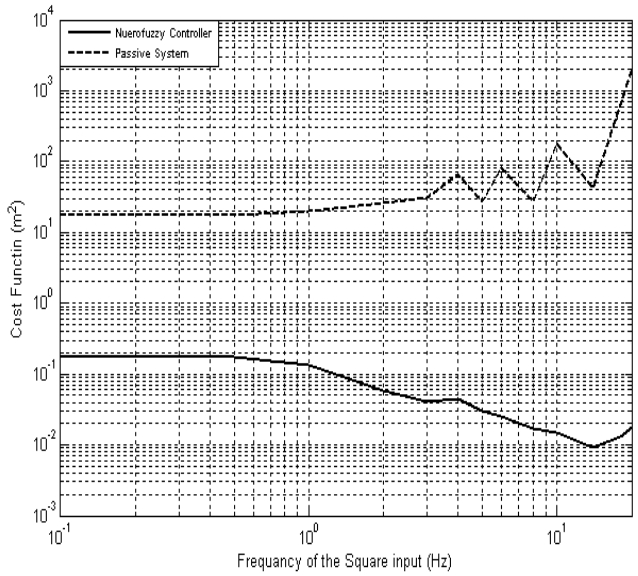


Figure 6.33 Time response of the cost functions against the different frequency values of square wave

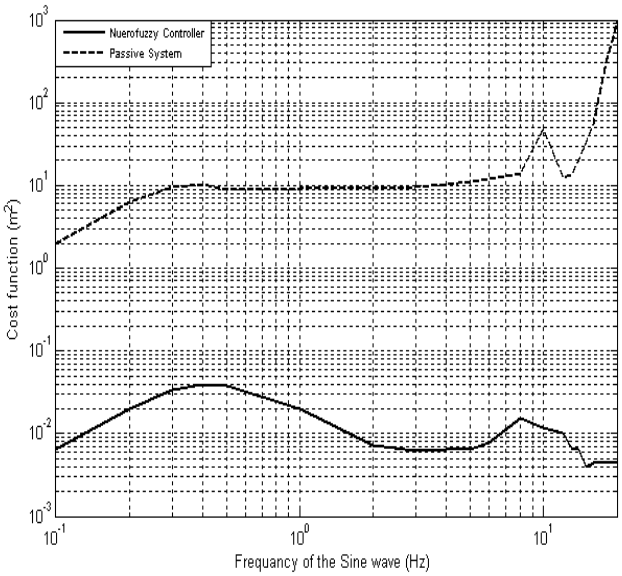


Figure 6.34 Time response of the cost functions against the different frequency values of sine wave

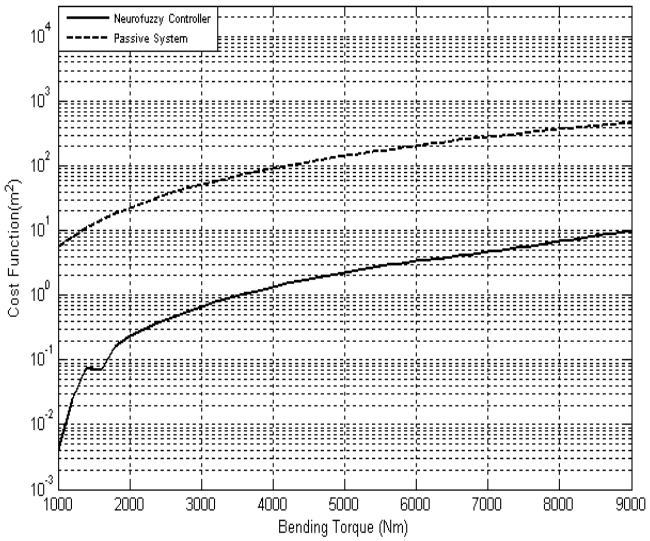


Figure 6.35 Time response of the cost functions against different values of bending torque (T_x)

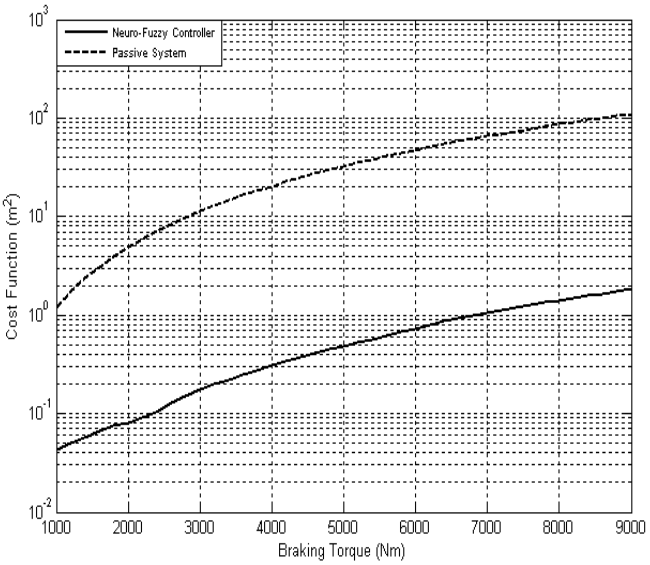


Figure 6.36 Time response of the cost functions against different values of braking torque (T_y)

6.8 The comparison among the robustness of the FOPID controller, FMRL controller, Neural controller and Neurofuzzy controller

In this project, different types of control system have been designed for full vehicle nonlinear active suspension systems with nonlinear hydraulic actuators, i.e. FOPID controller, FMRL controller, Neural controller and Neurofuzzy controller. In this section, the comparison among the robustness of those controllers against different disturbances types, which are listed in previous section, are investigated to select the best controller among them. Figures 6.37-6.42 show the response of the cost function against different disturbances types.

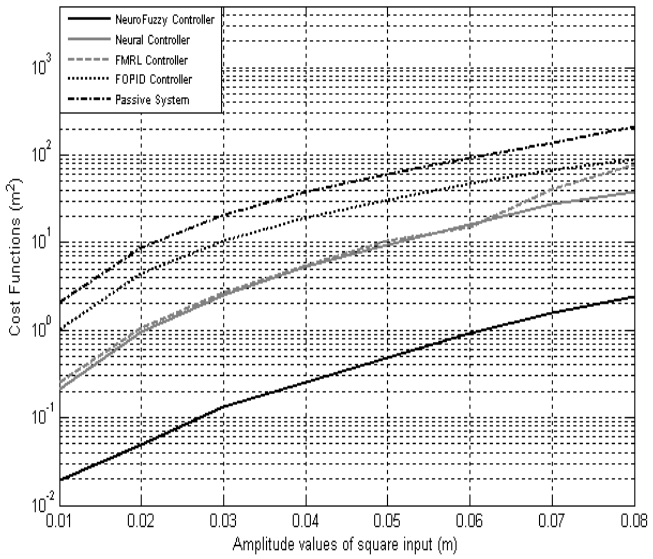


Figure 6.37 Time response of the cost functions against the different amplitude values of square wave

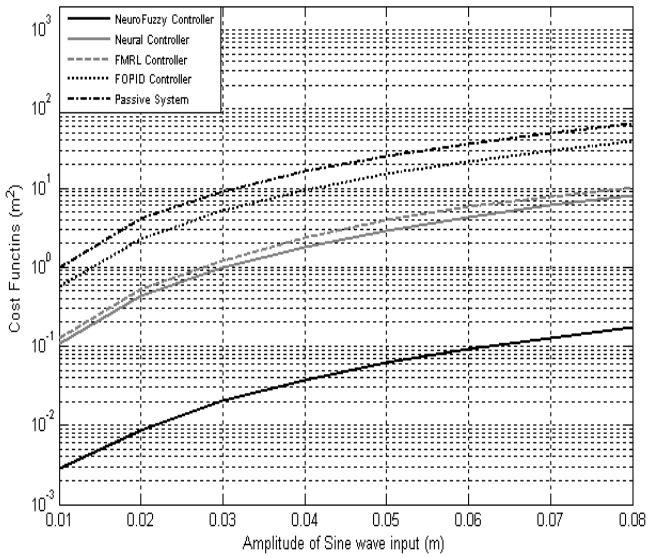


Figure 6.38 Time response of the cost functions against the different amplitude values of sine wave

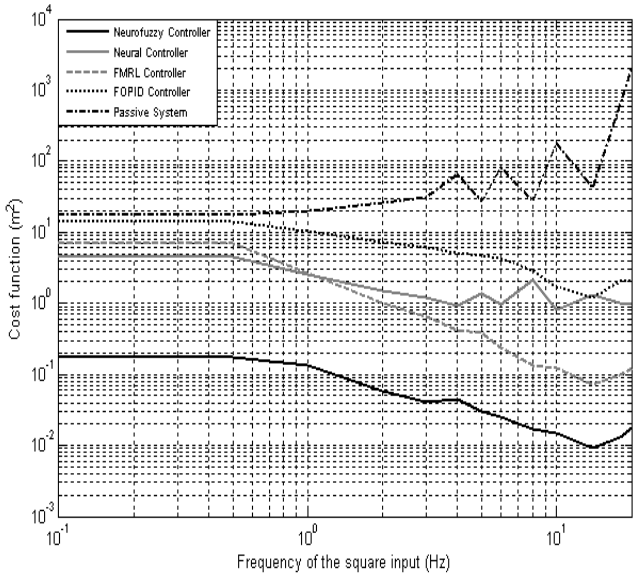


Figure 6.39 Time response of the cost functions against the different frequency values of square wave

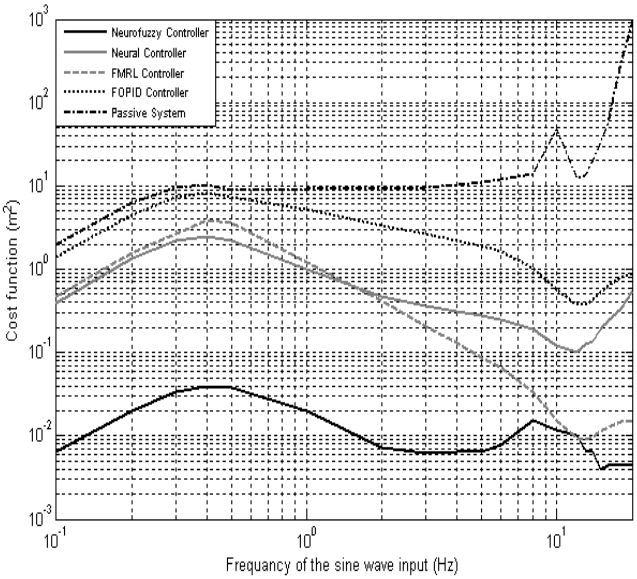


Figure 6.40 Time response of the cost functions against the different frequency values of sine wave

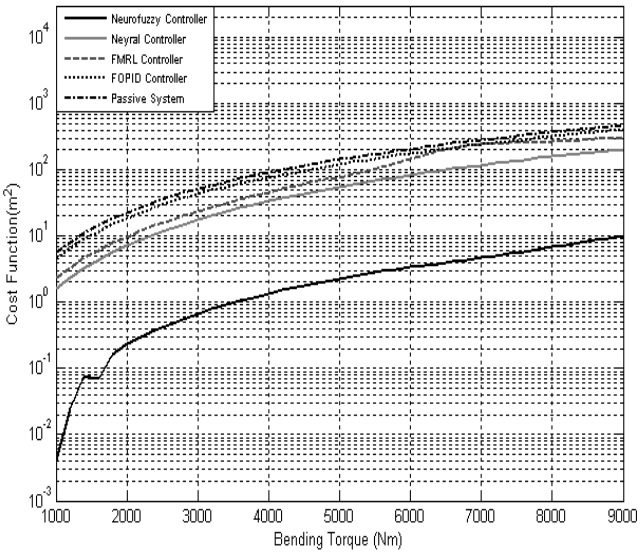


Figure 6.41 Time response of the cost functions against different values of bending torque (T_x)

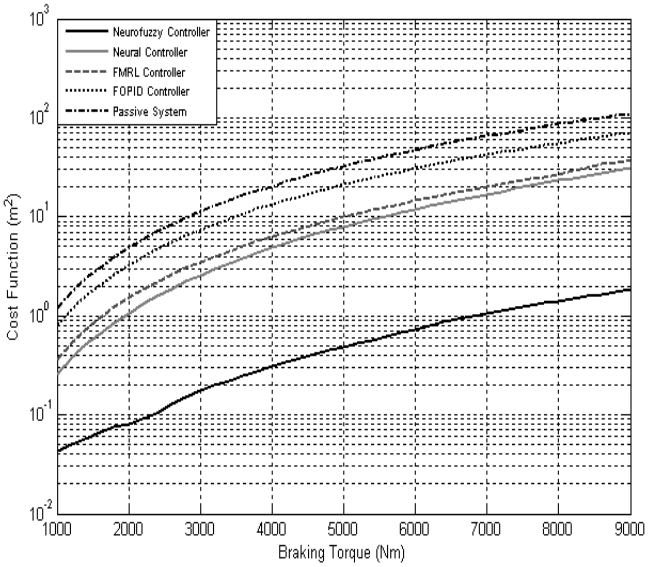


Figure 6.42 Time response of the cost functions against different values of braking torque (T_y)

The comparison illustrated that the neurofuzzy controller is the best controller for any disturbance type which means that the neurofuzzy controller is most effective and robust than the other controller systems to control the full vehicle nonlinear active suspension systems with nonlinear hydraulic actuators.

The hardware implementation of the neurofuzzy controller using FPGA board will be described in the next chapter.

6.9 Summary

As mentioned before, the main objectives of designed controller for a vehicle suspension system are to reduce the discomfort sensed by passengers which arises from road roughness and to improve the road handling associated with the pitching and rolling movements. This necessitates a very fast and accurate controller to meet as much control objectives; as possible. For these reasons, four novel neurofuzzy controllers have been successfully developed for the full vehicle nonlinear active suspension system. The approach of the proposed controller is to minimize the vibration on each suspended corner of the vehicle by supplying control forces to suspension system when travelling on a rough road. The other purpose of use the NF controller for the vehicle model is to reduce the body inclinations that happen during intensive manoeuvres including braking and cornering. It is believed that this is pioneer work to use the neurofuzzy method to design the controller for full vehicle nonlinear active suspension systems with nonlinear hydraulic actuators.

The input and output data sets that were obtained from the FOPID controllers have been used as training data to design the neurofuzzy controllers. The hybrid learning algorithm has been applied as optimisation method to train the trainable parameters of the neurofuzzy controllers. After the optimum parameters of the neurofuzzy controller have been obtained, the input and output scaling gains have been adjusted by using four-dimensional golden section search to

improve the performance of the proposed controllers. Four NF controllers have been designed, one for each suspension system. The performances of the proposed controllers have been tested and the results illustrated that the riding comfort and road handling have been greatly improved. Six different types of the disturbances have been applied individually to test the robustness of the proposed controllers. The test of the robustness proves that the NF controller is still stable and it forces the cost function to be minimum even significant disturbances occurred.

7. Implementation of Neurofuzzy Controllers Using an FPGA Platforms

7.1 Micro-Controller hardware implementation of adaptive systems

The conventional controller such as the PID controller requires an exact mathematical model of the controlled system to meet as much control objectives as possible. If the mathematical model for any system is difficult to establish, the fuzzy logic controller can be a good alternative to design a robust controller. Fuzzy logic systems, which can deal with imprecise information, are good at explaining their decisions but they cannot automatically acquire the rules used to make those decisions. On the other hand, artificial neural networks are good at recognizing patterns. The Neural networks have the ability to train the parameters of the control system. However, they are not good at explaining how they reach their decisions. These limitations in both systems have been a central driving force behind the creation of the neurofuzzy system where the two techniques are combined in a manner that overcomes the limitations of the individual techniques.

In the last three decades, many researchers focused on the development of a hardware implementation for fuzzy logic systems and neural networks. An analogue circuit used to implement each part of a fuzzy system includes: Fuzzification, Fuzzy Inference and Defuzzification based on different circuit techniques, such as CMOS, BiCMOS and bipolar

(Catania et al. 1994; Gilbert 1984; Ishizuka et al. 1992; Tasaka 1989; Yamakawa 1992). The structure of a fuzzy system is complex so that the analogue circuit has to be very complicated to implement the analogue fuzzy system. Therefore, most researchers proposed digital circuits to implement the fuzzy logic system using either Application Specific Integrated Circuits (ASICs) or general purpose software microprocessor or microcontroller. The ASICs are faster than the general purpose microprocessor, but the disadvantage of the ASIC is that they do not offer re-configurability by the user. On the other hand, the microprocessor is more economical and flexible than the ASIC, but often faces difficulties in dealing with control systems that require high processing and input and output handling speeds. Therefore, higher density programmable logic devices such as Programmable Logic Devices (PLDs) and Field Programmable Gate Arrays (FPGAs) have been developed to overcome the problems of microprocessors. The FPGA is suitable for fast implementation and quick hardware verification. The systems based on it are flexible and can be reprogrammed with an unlimited number of times. The rapid evaluation of silicon technologies has helped to reduce the size of FPGA integrated circuits and their cost. Therefore, the FPGAs can be used as final solutions to implement many systems (Freeman 1988; Sanchez-Solano et al. 2007).

Many papers have reported FPGA technology to design Fuzzy Logic Controller (FLC) for different applications. Such publications include these by Manzoul and Jayabharathi (1992) who developed software for synthesizing fuzzy controllers into Boolean equations. Also, a hardware implementation of a fuzzy controller on an FPGA has been described. The developed software together with the FPGA development system provides a complete design automation tool for fuzzy controllers. Obaid et al. (2010) presented analysis and performance evaluation of the proportional-derivative (PD) fuzzy logic controller designs using Matlab and an FPGA. Some authors proposed an embedded run-time reconfigurable architecture for the design of a fuzzy logic PID controller (Economakos and Economakos 2007; Hu and Li 1996;

Tipsuwanpornm et al. 2004). Singh et al. (2009) presented and implemented a fuzzy controller on an FPGA using VHDL for a motor system. Here, fuzzy controller was implemented on an FPGA board to control a shunt motor used for controlling the speed of an electrical vehicle (Poorani et al. 2005). In Reference Cirstea et al. (2001), simulation and implementation of a fuzzy logic controller for a diesel driven stand alone synchronous generator system had been designed. This controller was developed using VHDL and is implemented in an FPGA. Barriga et al. (2006) depicted a fuzzy logic modelling style based on two strategies: behavioural modelling using VHDL and structural VHDL based on specific architecture of fuzzy processor.

Other researchers used an FPGA to implement the neural network controller. In Reference Jung and Kim (2007), the authors implemented the intelligent neural network controller hardware with an FPGA based general purpose chip and a digital signal processing board to solve an nonlinear system control problems. Yu and Dent (1994) dealt with efficient implementation of feedforward networks in table-lookup FPGAs for a portable digital system in classification of balliscardiogram (BCG) for automatic heart disease monitoring. Muthuramalingam et al. (2007) discussed the issues involved in implementation of a multi-input neuron with linear and nonlinear excitation functions using an FPGA. Blake et al. (1998) described the hardware implementations of fuzzy systems, neural networks and fuzzy neural networks using a Xilinx FPGA.

As mentioned in Chapter 6, the neurofuzzy controller is more effective and robust than other controller systems for full vehicle nonlinear active suspension systems with nonlinear hydraulic actuators. Therefore, in this chapter, the implementation of neurofuzzy controller using an FPGA will be explained. Very High speed integrated circuit Hardware Description Language (VHDL) codes are used to describe the operation of the neurofuzzy controller that has been designed in Chapter 6. The Xilinx Integrated Software Environment (Xilinx ISE

10.1) has been used as the environment to type and synthesis the VHDL codes. Xilinx ISE 10.1 allows taking designs through several steps: Analysis and Synthesis, Implement Design, Generated Programming Files and Configure Target Device.

After successfully compiling the design, the generated programming files (configuration files) are downloaded using a USB port to program the FPGA. For the simulation results, ModelSim XE III 6.4b simulation program is used to simulate the operation of the VHDL codes and obtained the expected output data of the FPGA board. In order to compare the expected responses of the FPGA design with the MATLAB Simulink design, an M-file will be used to plot the data that is collected from the ModelSim program and the data that is collected from the MATLAB Simulink design to confirm that the FPGA boards operate as neurofuzzy controllers do.

7.2 Field programmable gate arrays (FPGA) architecture

A field programmable gate array (FPGA) is a logic device that contains a two-dimensional array of generic logic cells and programmable switches that can realize any digital system with low cost and reduced time. The FPGA consists of three major configurable elements (Brown and Vranesic 2005):

1) Configurable Logic Block (CLBs) arranged in an array that provides the functional elements and implements most of the logic in an FPGA. Each logic block has two flip-flops and it can realize any 5-input combination logic function.
2) Programmable Interconnect Resources (PIRs) that provides a routing path to connect between the rows and columns of CLBs, and between CLBs and input-output blocks.
3) Input-Output Blocks (IOBs) that provide the interface between the package pins and internal signal lines. It can be configured as an input, output or bidirectional port.

The CLBs, IOBs and their interconnectors are controlled by a configuration program store in a chip memory.

The structure of FPGA device is illustrated in Figure 7.1 (from Economakos and Economakos (2007)). A custom design can be implemented by specifying the function of each logic cell and selectively setting the connection of each programmable switch. FPGAs are programmed using support software and a download cable connected to a host computer. Once they are programmed, they can be disconnected from the computer and will retain their functionality until the power is removed from the chip. Since this process can be done by the user rather than by the fabrication facility, the device is known as field programmable.

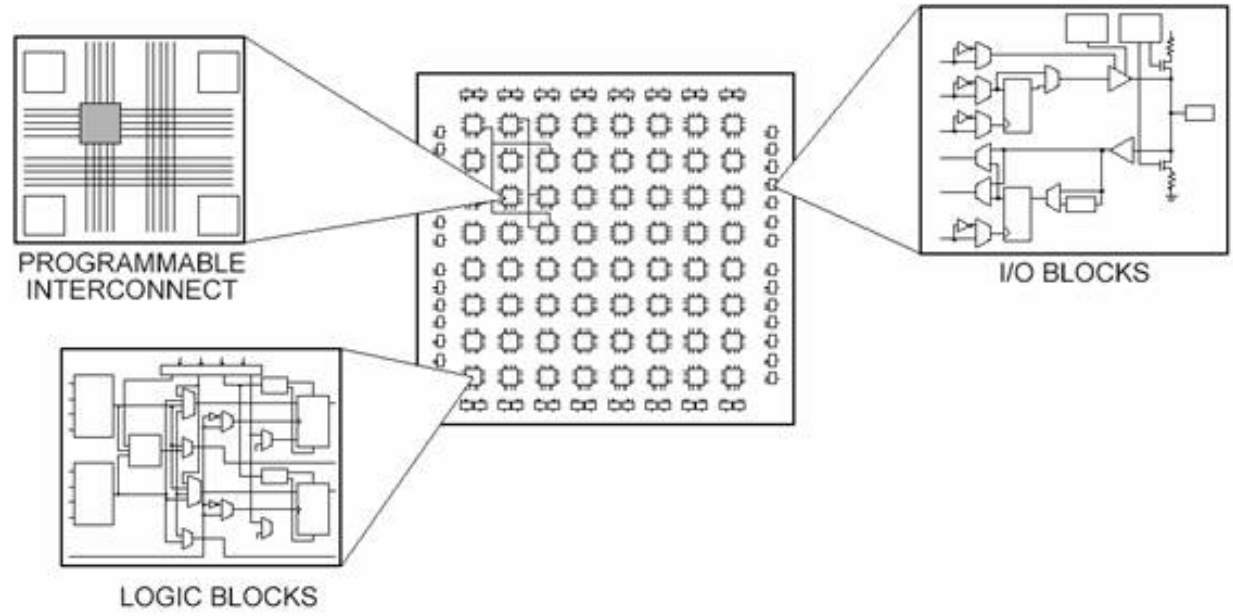


Figure 7.1 Overall Schematic of a traditional FPGA

7.3 Hardware description languages (HDL)

Hardware description languages (HDLs) have become widely use to design digital systems. HDLs can describe the digital circuit's operation, its design and organisation, and tests to verify its operation by means of simulation. A digital design can be created using any HDL

such as Verilog language and Very High speed integrated circuit Hardware Description Language (VHDL). The use of VHDL for modelling and simulation is especially appealing since it provides a formal description, simulation, verification and synthesis of the digital systems. In this chapter, VHDL will be used to design the neurofuzzy controller rather than Verilog language. Some authors have used the VHDL as a description language to support the data structure and function required for fuzzy logic systems (Barriga et al. 2006; Chen et al. 2009; Cobo et al. 1998; Gonzalez et al. 2007; Kim 2000; Rani et al. 2005; Salapura and Hamann 1996; Singh et al. 2009; Vuong et al. 2006). While some authors developed VHDL codes to implement the neural network controller (Jung and Kim 2007; Muthuramalingam et al. 2007; Yu and Dent (1994).

From the background research, it is evident that this is the first time that VHDL codes have been used to implement the neurofuzzy controller type described by Takagi and Sugeno system.

7.4 Description of neurofuzzy controller using VHDL codes

The main purpose behind implementation of the NF controller in VHDL is to minimise the hardware implementation cost of the generic NF controller for use in industrial applications. In this section, VHDL codes are presented for the neurofuzzy controllers which have two inputs one output. The first input is Error (E) and the second input is Change in Error (CE) and the output of control system is Control output (Con). The Bell Shape function has been used for each input.

For simplification, we assume that each input of the neurofuzzy controller has three grades: Negative (N), Zero (Z), and Positive (P) as illustrated in Figure 7.2 and Figure 7.3 for the first and second input respectively.

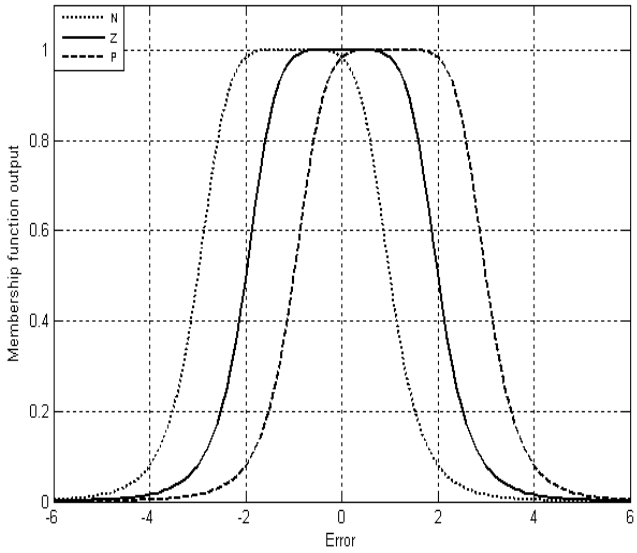


Figure7.2 Bell shape Membership Function for the first input

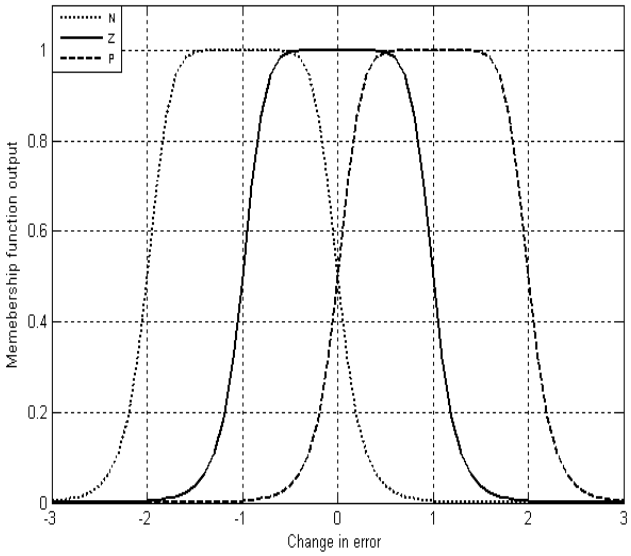


Figure7.3 Bell shape Membership Function for the second input

The VHDL codes are used to define the name, the start point and end point of each membership function for the first and the second input using record type declaration as follow:

```
--FIRST INPUT--
type input1 is (N1,Z1,P1);
type membership1 is record
                    term:input1;
                    point1:integer;
                    point2:integer;
end record;
type membership_function1 is array (natural range<>) of membership1;
constant mfs1:membership_function1:=
                    ((term=> N1,point1=>-6,point2=>4),
                     (term=> Z1,point1=>-5,point2=>5),
```

```
        (term=>P1,point1=>-4,point2=>6));
--SECOND INPUT--
type input2 is (N2,Z2,P2);
type membership2 is record
        term:input2;
        point1:integer;
        point2:integer;
end record;
type membership_function2 is array (natural range<>) of membership2;
constant mfs2:membership_function2:=
        ((term=> N2,point1=>-6,point2=>4),
         (term=> Z2,point1=>-5,point2=>5),
         (term=>P2,point1=>-4,point2=>6));
```

Three polynomial functions are assumed in the consequent part of the neurofuzzy controller as follows:

$f1 = a1E + b1CE + c1$

$f2 = a2E + b2CE + c2$

$f3 = a3E + b3CE + c3$

where *E* is the first input (Error); *CE* is the second input (Change in Error);

a1, *b1*, *c1*, *a2*, *b2*, *c2*, *a3*, *b3* and *c3* are the coefficient of *f1*, *f2* and *f3* (consequent parameters) respectively .

The VHDL codes are used to define the name and the coefficient of the polynomial functions using record type declaration as follows:

```
--OUTPUT--
type output is (f1,f2,f3);
```

```
type linear is record
        term: output;
        P1:integer;
        P2:integer;
        P3:integer;
end record;
type linear_function is array (natural range<>) of linear;
constant zeta: linear_function:=
        ((term=>f1,P1=>a1,P2=>b1,P3=>c1),
         (term=>f2,P1=>a2,P2=>b2,P3=>c2),
         (term=>f3,P1=>a3,P2=>b3,P3=>c3));
```

As mentioned before, the Xilinx ISE 10.1 program has been used as an environment to type and synthesis the VHDL codes. Unfortunately, that program does not support the type real of the inputs and outputs signals. Since, the output of the membership function is in a range from 0 to 1 (floating point), therefore, a new method should be suggested to solve this problem.

From the equation of calculation of the neurofuzzy output that was illustrated in Chapter 6, it can be seen that if the each output of the membership function is multiplied by an arbitrary integer number, the output value of the normalization node will not be affected. The following example can explain that:

If we assume that the neurofuzzy network shown in Chapter 6, has two inputs, a single output and two membership functions for each input as demonstrated in Figure 7.4, then the normalization node outputs can be given as:

$$\bar{s}_1 = \frac{A_1(x_1) * B_1(x_2)}{A_1(x_1) * B_1(x_2) + A_2(x_1) * B_2(x_2)}$$

$$\bar{s}_2 = \frac{A_2(x_1) * B_2(x_2)}{A_1(x_1) * B_1(x_2) + A_2(x_1) * B_2(x_2)}$$

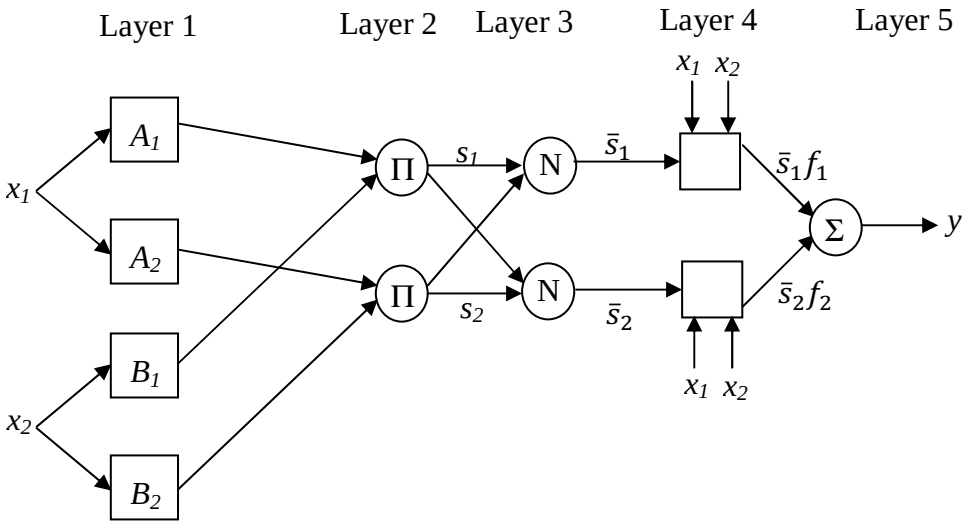


Figure 7.4 Neurofuzzy Network with two inputs and signal output

If the outputs of the membership functions are multiplied by *K* then the normalization node outputs will be given by

$$\bar{s}_1 = \frac{K * A_1(x_1) * K * B_1(x_2)}{K * A_1(x_1) * K * B_1(x_2) + K * A_2(x_1) * K * B_2(x_2)}$$

$$= \frac{K(A_1(x_1) * B_1(x_2))}{K(A_1(x_1) * B_1(x_2) + A_2(x_1) * B_2(x_2))}$$

$$\bar{s}_1 = \frac{A_1(x_1) * B_1(x_2)}{A_1(x_1) * B_1(x_2) + A_2(x_1) * B_2(x_2)}.$$

$$\bar{s}_2 = \frac{K * A_2(x_1) * K * B_2(x_2)}{K * A_1(x_1) * K * B_1(x_2) + K * A_2(x_1) * K * B_2(x_2)}$$

$$= \frac{K(A_2(x_1) * B_2(x_2))}{K(A_1(x_1) * B_1(x_2) + A_2(x_1) * B_2(x_2))}$$

$$\bar{s}_2 = \frac{A_2(x_1) * B_2(x_2)}{A_1(x_1) * B_1(x_2) + A_2(x_1) * B_2(x_2)}.$$

For example, if each output of the membership function is multiplied by 100, the accuracy will be two digits after the floating point and the outputs of the normalization nodes will remain as it is.

The inputs of the neurofuzzy system and the coefficients of the polynomial function in the consequent part are not integer numbers; they are a floating point numbers. As mentioned before, Xilinx ISE 10.1 program does not support the type real of the inputs and output signals. Therefore, two steps have been suggested to solve this problem:

1) Each value of the inputs and corresponding value of the bell-shape membership functions have been store in the look-up table.
2) The values of the neurofuzzy network inputs and the coefficient of the polynomial function are multiplied by a specific parameter Φ (for example 100), the absolute coefficients are multiplied by Φ^2, and the over all output of the neurofuzzy network is divided by Φ^2.

By using those two steps, the accuracy of the inputs of the neurofuzzy system and the coefficients of the polynomial function will be two digits after the floating point and the overall output of the neurofuzzy network will remain as it is.

To find the output of the Bell-shape membership function (Fuzzification step), the following VHDL codes should be used:

```
--FUZZIFICATION OF FIRST INPUT--
        if (E>= mfs1(0).point1) and (E<= mfs1(0).point2)then
            NE (E, error (0));
            else error(0)<=0;
        end if;
        if (E>= mfs1(1).point1) and (E<= mfs1(1).point2)then
            ZE (E, error (1));
            else error(1)<=0;
        end if;
        if (E>= mfs1(2).point1) and (E<= mfs1(2).point2)then
            PE (E, error (2));
```

```
            else error1(2)<=0;
        end if;

--FUZZIFICATION OF SECOND INPUT--
        if (CE>= mfs2(0).point1) and (CE<= mfs2(0).point2)then
            NCE (CE, chan_error (0));
            else chan_error (0)<=0;
        end if;
        if (CE>= mfs2(1).point1) and (CE<= mfs2(1).point2)then
            ZCE (CE, chan_error (1));
            else chan_error (1)<=0;
        end if;
        if (CE>= mfs2(2).point1) and (CE<= mfs2(2).point2)then
            PCE (CE, chan_error (2));
            else chan_error (2)<=0;
        end if;
```

where *NE, ZE, PE, NCE, ZCE* and *PCE* are subroutines which call the look up table to obtain the corresponding output of the membership function for a specific value of the input signal. Consideration a fuzzy rule base consists of only three rules:

R_1: $IF\ E\ is\ EN\ and\ CE\ is\ CEN\ THEN\ y = f_1(E, CE)$

R_2: $IF\ E\ is\ EZ\ and\ CE\ is\ CEZ\ THEN\ y = f_2(E, CE)$

R_3: $IF\ E\ is\ EP\ and\ CE\ is\ CEP\ THEN\ y = f_3(E, CE)$

where $f_1 = a_1E + b_1CE + c_1$, $f_2 = a_2E + b_2CE + c_2$ and $f_3 = a_3E + b_3CE + c_3$.

The "*and*" is a product operation between the two antecedents. The following VHDL codes are used to obtain the result of each rule evaluation

```
--RULE EVALUATION--
for i in 0 to 2 loop
        position(i)<=error(i)*chan_error(i);
end loop;
```

The following VHDL codes are used to calculate the value of the polynomial function:

```
for i in 0 to 2 loop
        f(i) <=E*zeta(i).P1+CE*zeta(i).P2+zeta(i).P3;
end loop;
```

To find the output of the i^{th} normalization nodes, the output of i^{th} node in rule-antecedent layer (Layer 3) must be divided by the sum of all nodes in this layer for example:

$$\bar{s}_1 = \frac{A_1(x_1) * B_1(x_2)}{A_1(x_1) * B_1(x_2) + A_2(x_1) * B_2(x_2)}$$

Unfortunately, the VHDL language does not support the division operation. Therefore, a subroutine program should be written to archive the division operation as follows:

```
--DIVISION--
    procedure divid_fun(num1,den1,n:in integer;
        signal result1: out std_logic_vector(n downto 0)) is
                variable temp1: integer;
                variable temp2: integer;
        begin
                temp1:= num1;
                temp2:= den1;
                if (temp1=0) or (temp2=0) then result1<=(others=>'0');
                else
                        for i in n downto 0 loop
```

```
                    if (temp1>=temp2*2**i) then
                        result1(i)<= '1';
                        temp1:=temp1-temp2*2**i;
                    else result1(i)<='0';
                    end if;
                end loop;
            end if;
        end divid_fun;
```

The output of this subroutine program is binary type; therefore, the output of this subroutine program should be converted from binary type to integer type using the following code

Out_integer<=conv_integer (unsigned (out_binary)).

Therefore, the outputs of the normalization nodes of the Neurofuzzy controller can be evaluated using the following VHDL codes:

```
--NORMALIZATION NODES OUTPUT EVALUATION--
constant itr: integer:=17;
sum1:=position (0) +position (1) +position (2);
for i in 0 to 2 loop
        divid_fun (abs(position(i)),abs(sum1),itr, result);
        result1 (i) <=conv_integer (unsigned (result));
       if position(i)=abs(position(i)) and sum1=abs(sum1) then sign<='0';
      end if;
     if position(i)=abs(position(i)) and sum1/=abs(sum1) then sign<='1';
    end if;
   if position(i)/=abs(position(i)) and sum1=abs(sum1) then sign<='1';
  end if;
```

```
 if position(i)/=abs(position(i)) and sum1/=abs(sum1) then sign<='0';
end if;
if sign='0' then result2(i) <=result1(i);
    else  result2(i)<=-1*result1(i);
end if;
end loop;
```

The output of each node in the layer 4 can be calculated by multiplying the output of the i^{th} normalization node by the i^{th} of polynomial function. The following VHDL codes are used to obtain the outputs of layer 4:

```
--LAYER4 OUTPUTS—
 for i in 0 to 2 loop
        product (i)<= result2(i)*f(i);
 end loop;
```

Finally, the over all output of the Neurofuzzy controller can be obtained using the following VHDL code:

```
output <=product(0)+ product(1)+ product(2);
```

7.5 Implementation of neurofuzzy controller using FPGA platforms

Higher density programmable logic devices such as FPGAs can be used to integrate large amounts of logic in a single integrated circuit. The Field Programmable Gate Array (FPGA) places fixed logic cells on the wafer and the FPGA designer constructs more complex functions from these cells. As mentioned before, Xilinx ISE 10.1 program has been used as programming environment to type and synthesis the VHDL codes that described the neurofuzzy controller and to generate a configuration file which is used to program the FPGA

board. Xilinx ISE consists of a set of software tools that allows us to implement the development flow of an FPGA system. The simplified development flow of an FPGA based system is shown in Figure 7.5. This development flow has two portions; the left portion of the flow is the refinement and programming process, while the right portion is represents the simulation of VHDL codes.

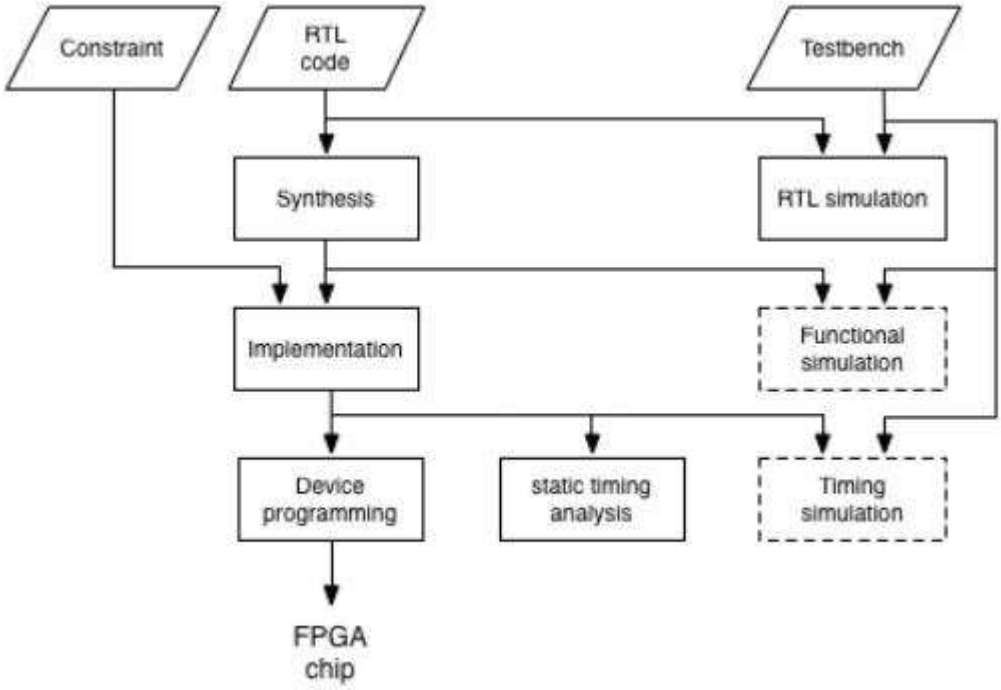


Figure 7.5 Development flow (coped from Reference Chu (2008))

The left portion blocks in Figure 7.5 involve the following basic steps:

- Register Transfer Level (RTL) code: the RTL code can be a result of direct VHDL coding. In this step the VHDL codes is entered to the developed flow.
- Synthesis: that block checks the syntax of VHDL codes.
- Implementation: determination the placement of the Logic Elements (LEs) in an actual FPGA chip and selecting the routing wires in the chip to make the required connections between specific LEs .

- Static timing analysis: propagation delays along the various paths in the implementing circuit are analyzed to provide an indication of the expected performance of the circuit.
- Timing Simulation: the implementing circuit is tested to verify both its functional correctness and timing.
- Device Programming or Configuration: the designed circuit is implemented in a physical FPGA chip by generating a configuration file. This configuration file is ready to download into the FPGA board using a USB port.

ModelSim XE III 6.4b simulation program is used to simulate the operation of the VHDL codes and obtain the expected output data of the FPGA board. The neurofuzzy controller is designed and simulated using MATLAB software tool (as explained in Chapter 6). The simulation data obtained from the ModelSim simulation program is compared with the simulation data obtained from the MATLAB software in order to confirm that the expected response of the FPGA-based controller is close to the response of the NF controller, which was designed by MATLAB software. Figure 7.6 illustrates the comparison between the simulation data.

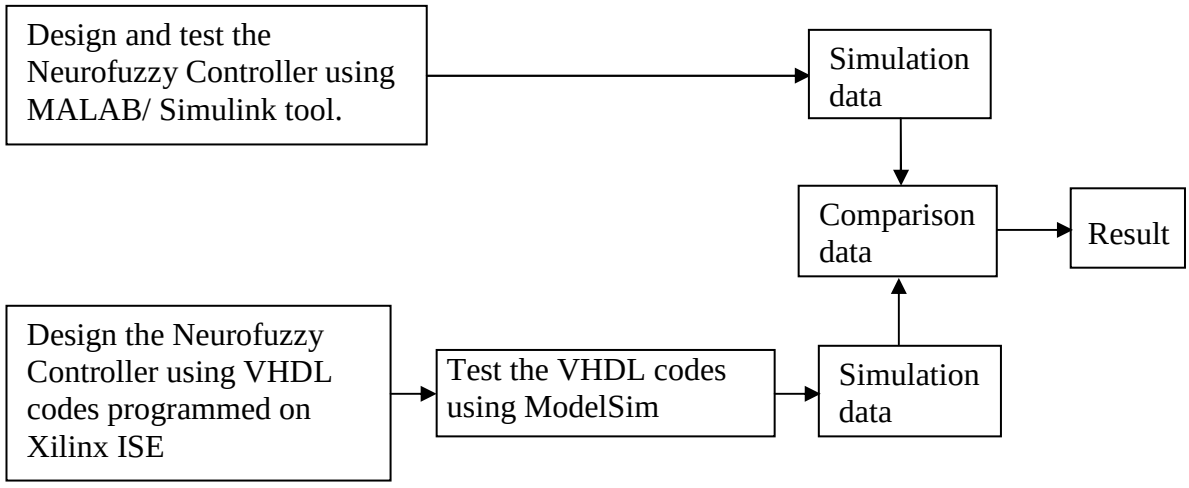


Figure 7.6 Comparison between the simulation data

Figure 7.7 shows the proposed connection of the FPGA board with the process system. First, a VHDL codes are downloaded from the host computer into the FPGA chip (XILINX Spartan XC3S500E) using a USB cable. Then, the Hirose 100-pin FX2 Edge connector is used to interface the board (NF controller) with model of the suspension system. The error between the reference input and the system output has been applied as an input to the A/D converter. The digital output of the A/D converter has been applied as input data to the FPGA boards. The FPGA board generates the digital inputs to the NF controller (error and error rate). The NF controller generates a suitable digital control signal based on the rules that were stored in the FPGA chip. The digital control signal will be delivered to the D/A convertor to generate an analogue control signal that will be applied as an input to the suspension system.

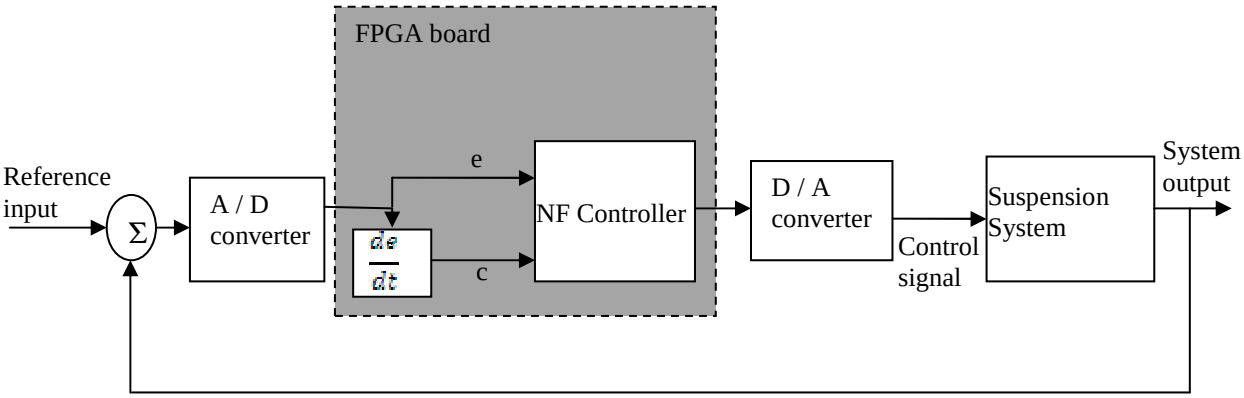


Figure7.7 Layout of FPGA board with suspension system

7.6 Simulation and results

Four intelligent Neurofuzzy controllers were design as shown in Chapter 6. The optimal trainable parameters of the Neurofuzzy controller will be used to implement and design the FPGA boards. Four VHDL codes have been implemented, one for each suspension system unit.

The Xilinx Spartan XC3S500E FPGA board can be used to implement the NF controller. The Hirose 100-pin FX2 Edge connector (J3) is used as the input and output port to receive the digital input error data from the A/D converter, where the error signal is the difference between the output of the controlled system and desired output, and to deliver the digital control signal to the D/A converter. The analogue control signal is used to force the output of the controlled system to follow the desired output.

Firstly, the VHDL codes are typed in the Xilinx ISE 10.1 program to describe the neurofuzzy controller which consists of two inputs, in which each input has five Bell-shape membership functions, one output and five rules. Then, the testbench file has been created to test the operation of VHDL codes. Finally, a constraint file is written to map the input and output signals of the top level module to physical pins of the FPGA board. The constraint files used in this work are shown in Appendix 4. The general layout of the design is shown in Figure 7.7. The designed controller accepts the error and the error rate both as digital signals, and delivers digital control action signal as an output. In this project, the error digital signal is represented as an 8-bit vector, the error rate digital signal is represented as a 13-bit vector and the digital control signal is represented as a 19-bit vector. The digital control signal is applied as an input to the A/D converter which delivers the analogue control signal for the hydraulic actuator of the suspension system.

As mentioned before, the ModelSim XE III 6.4b program has been used for the purposes of simulation and testing the VHDL codes to confirm that the FPGA boards used as the neurofuzzy inference systems operate as expected. The control signals obtained form the simulation program (ModelSim XE III 6.4b) have been compared with the control signals obtained from a SIMULINK design using MATLAB software. An M-file (Matlab-file) will be used to display the comparison.

Figures 7.8-7.11 show the comparison between the control signals which are obtained from each design (Matlab softwer and ModelSim software) for each controller.

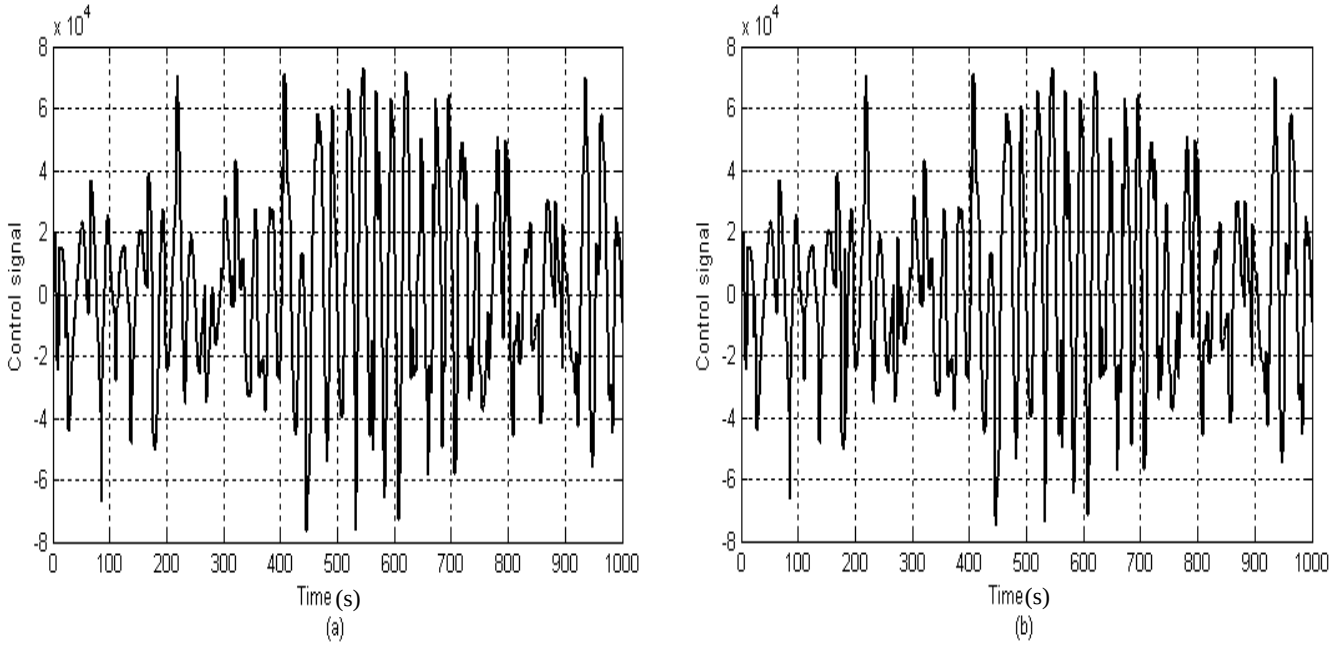


Figure 7.8 Control Signal Output for the First Controller
(a) From Matlab, (b) From ModelSim.

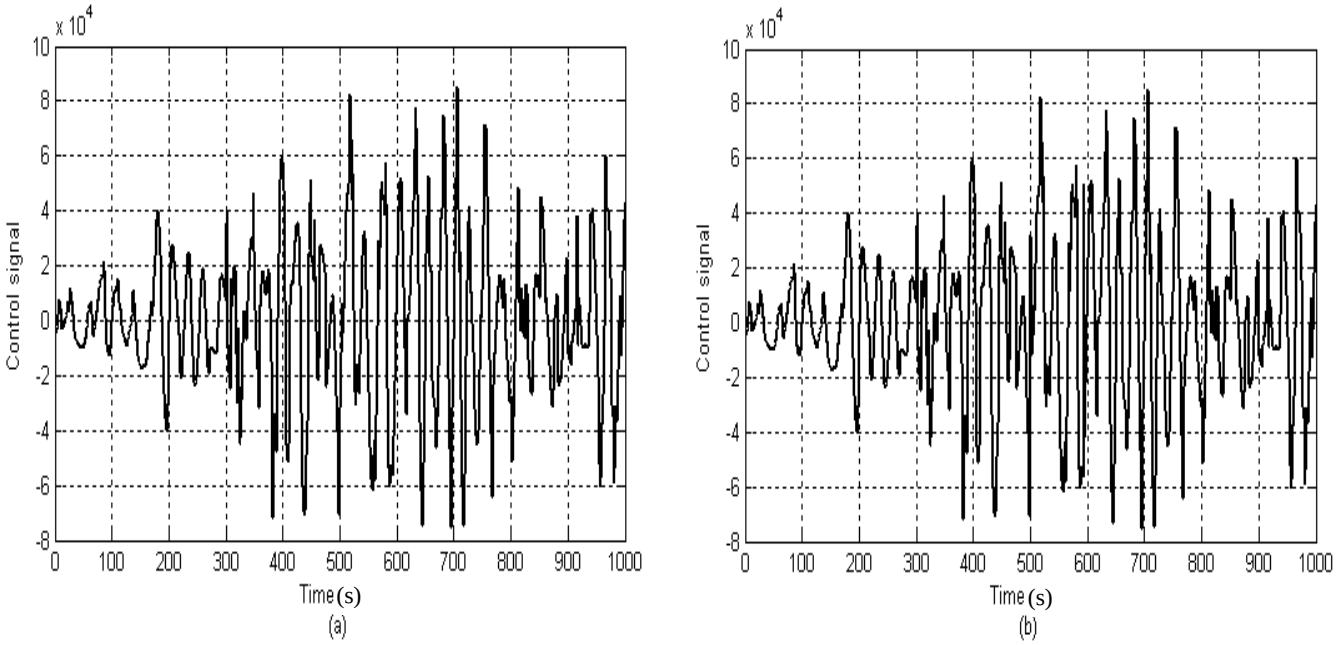


Figure 7.9 Control Signal Output for the Second Controller
(a) From Matlab, (b) From ModelSim..

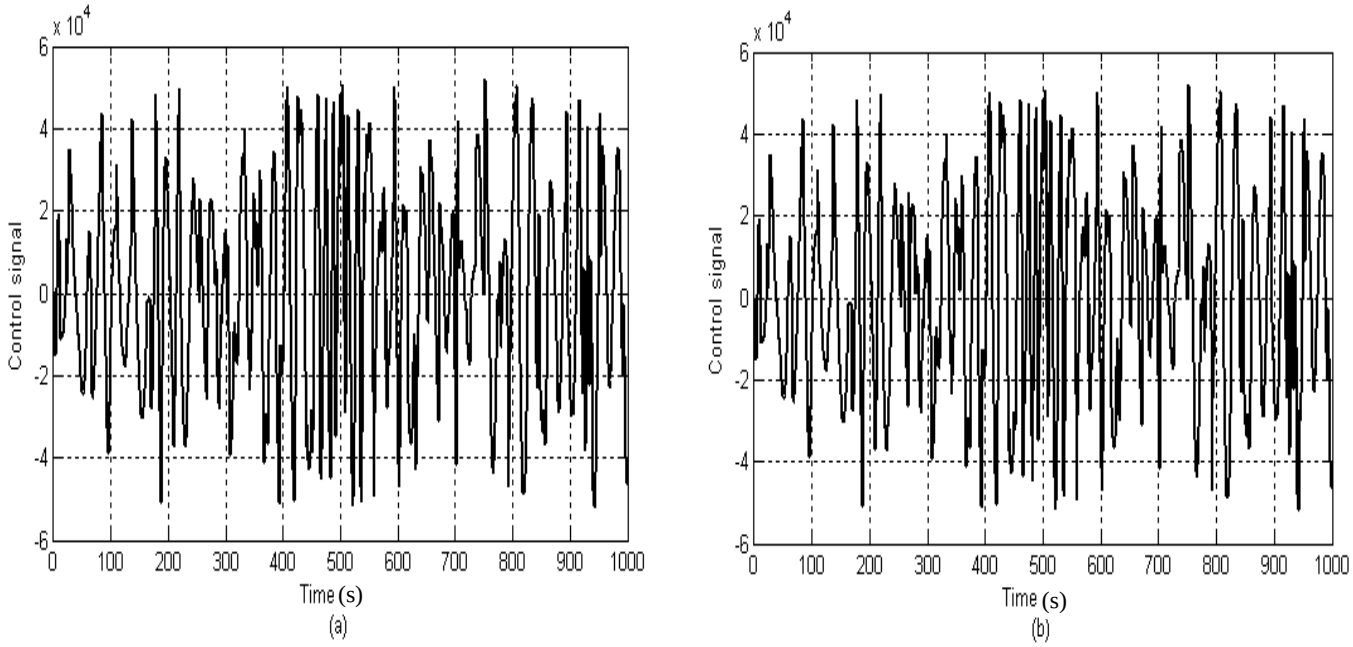


Figure 7.10 Control Signal Output for the Third Controller
(a) From Matlab, (b) From ModelSim.

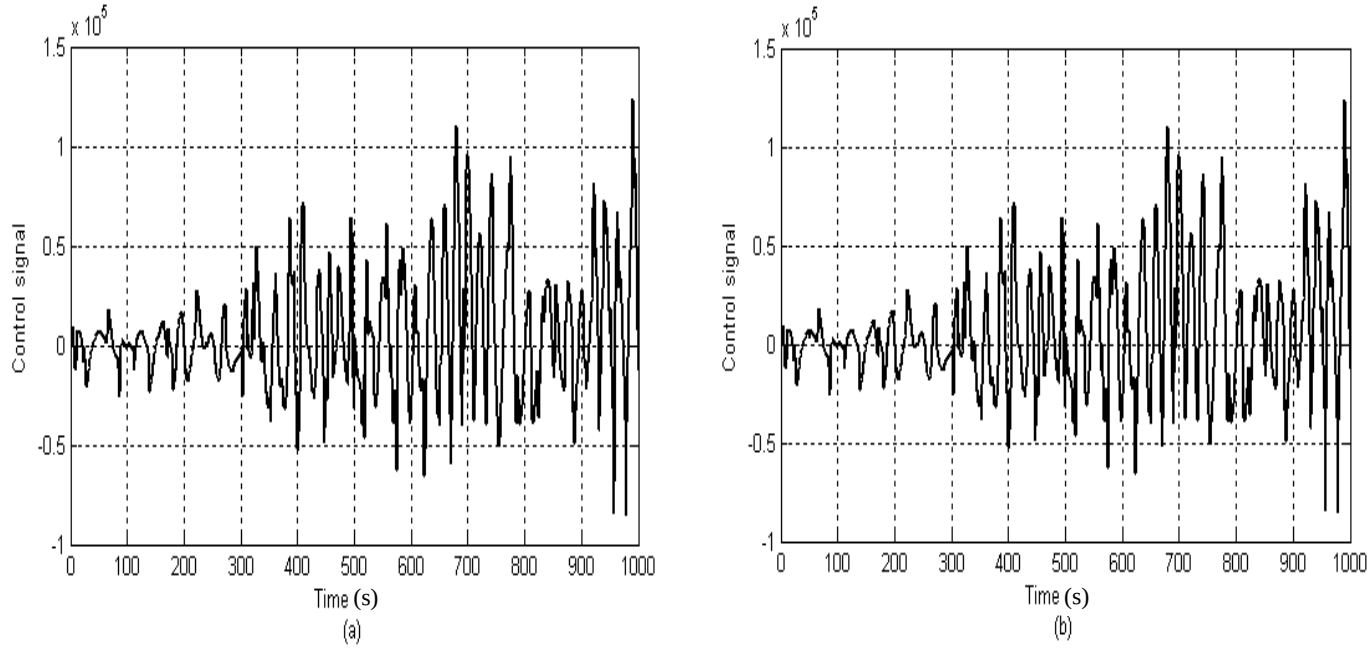


Figure 7.11 Control Signal Output for the Fourth Controller
(a) From Matlab, (b) From ModelSim.

Figures 7.12-7.15 show the samples of the simulation results that are obtained from ModelSim program. The red plot lines show the values of the first input, second input and the control signal output in the binary data type, respectively. While, the blue plot lines show the values of the first input, the second input and the control signal output in the decimal data type, respectively.

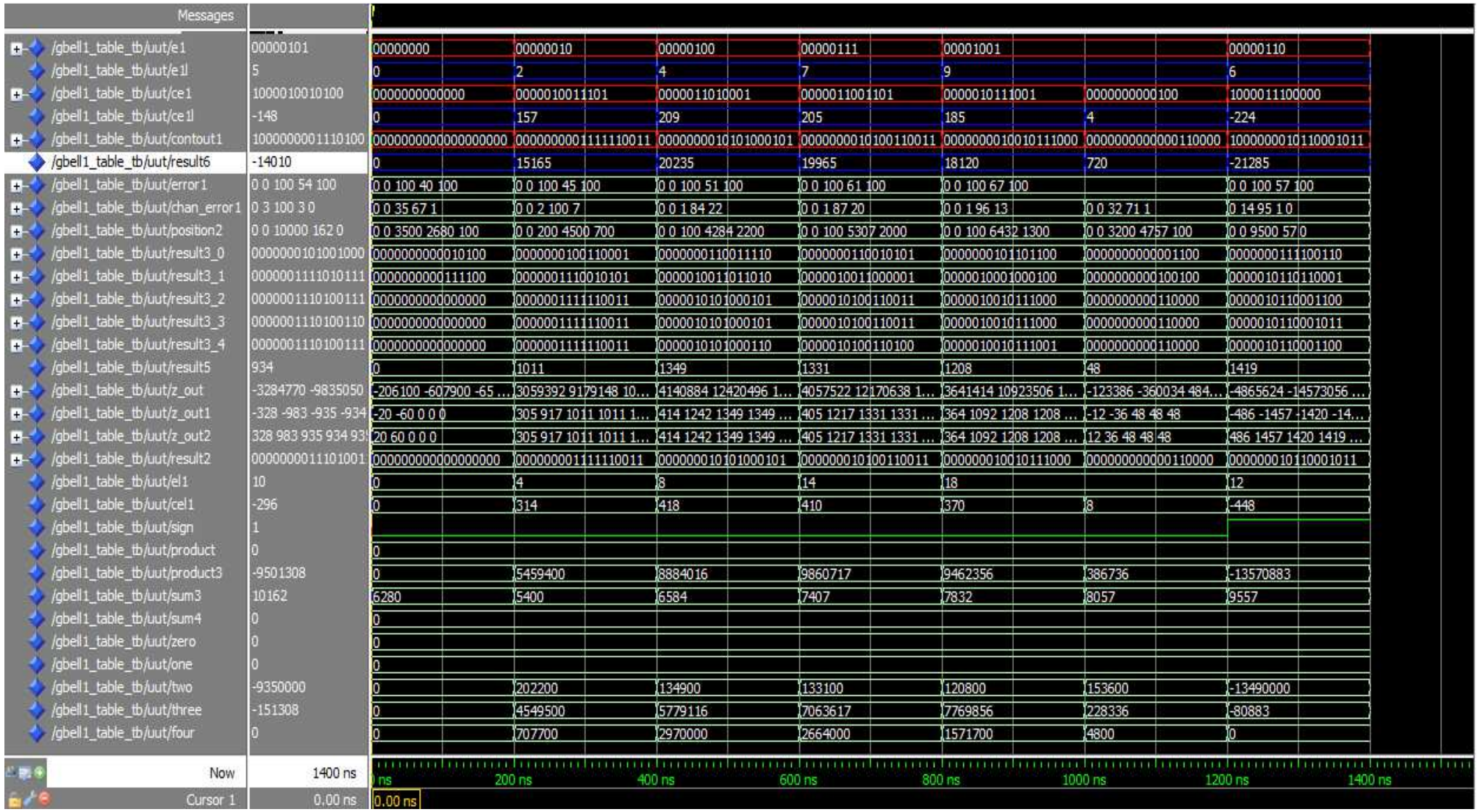


Figure 7.12 Samples of the Simulation Result of First Neurofuzzy Controller

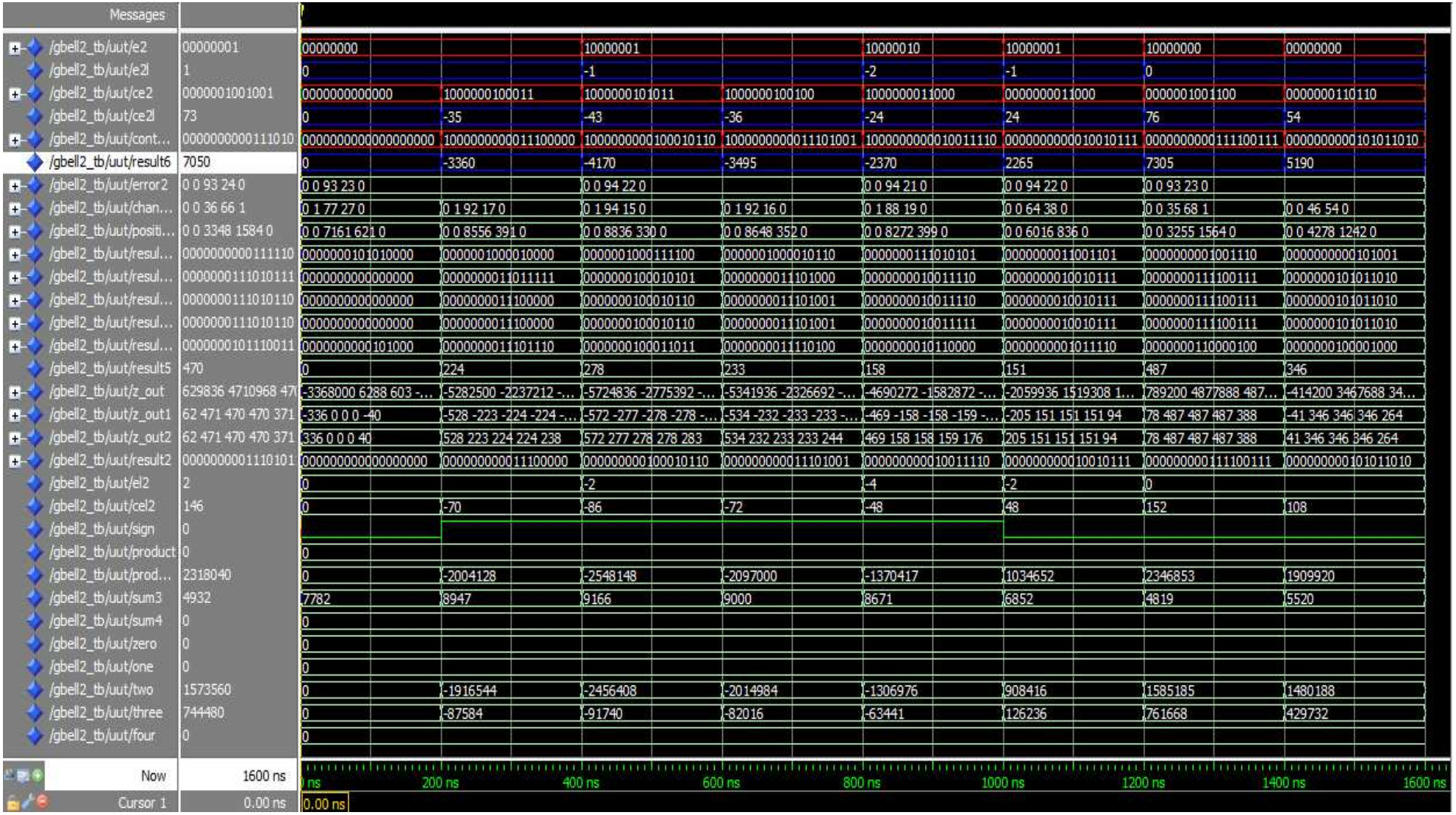


Figure 7.13 Samples of the Simulation Result of Second Neurofuzzy Controller

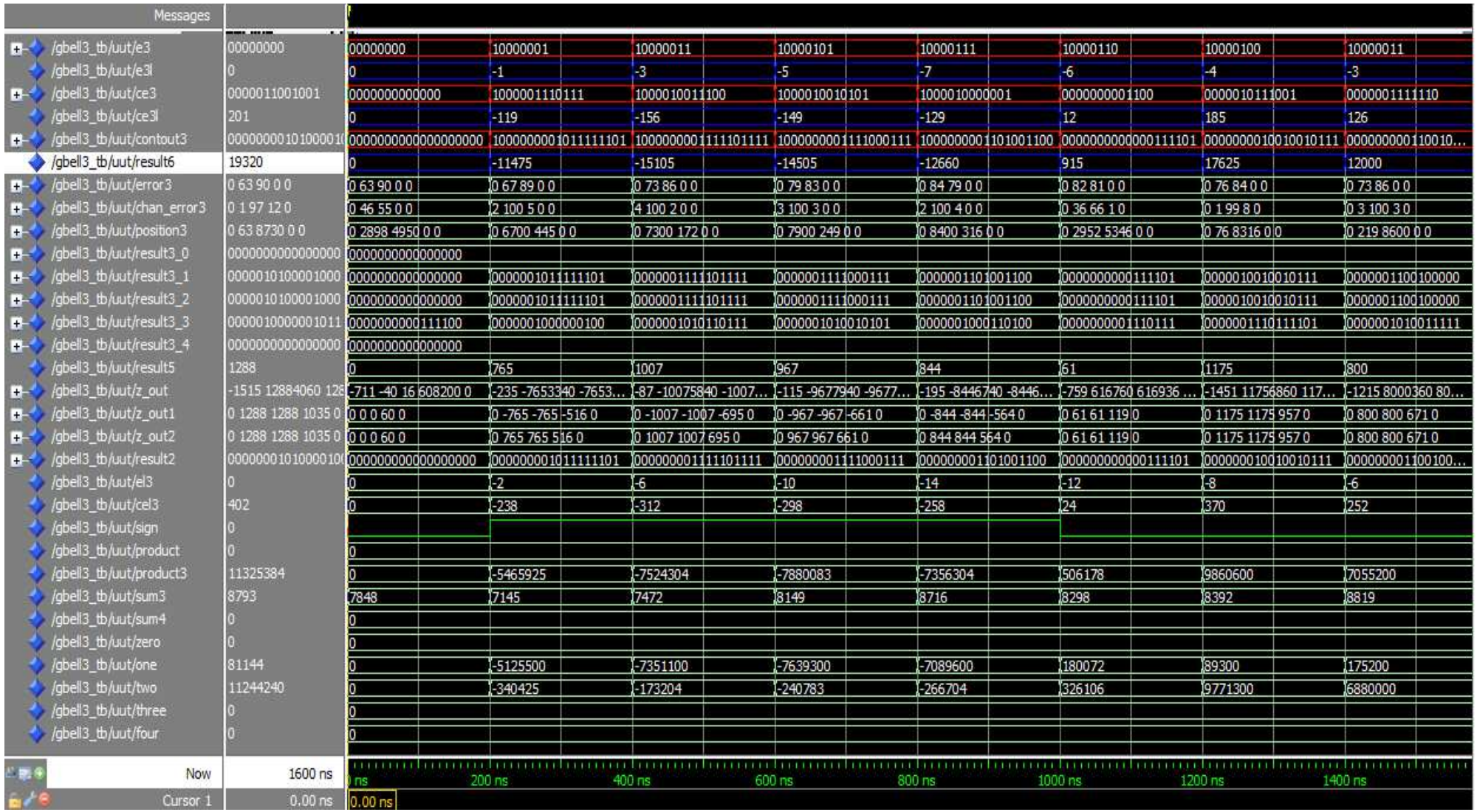


Figure 7.14 Samples of the Simulation Result of Third Neurofuzzy Controller

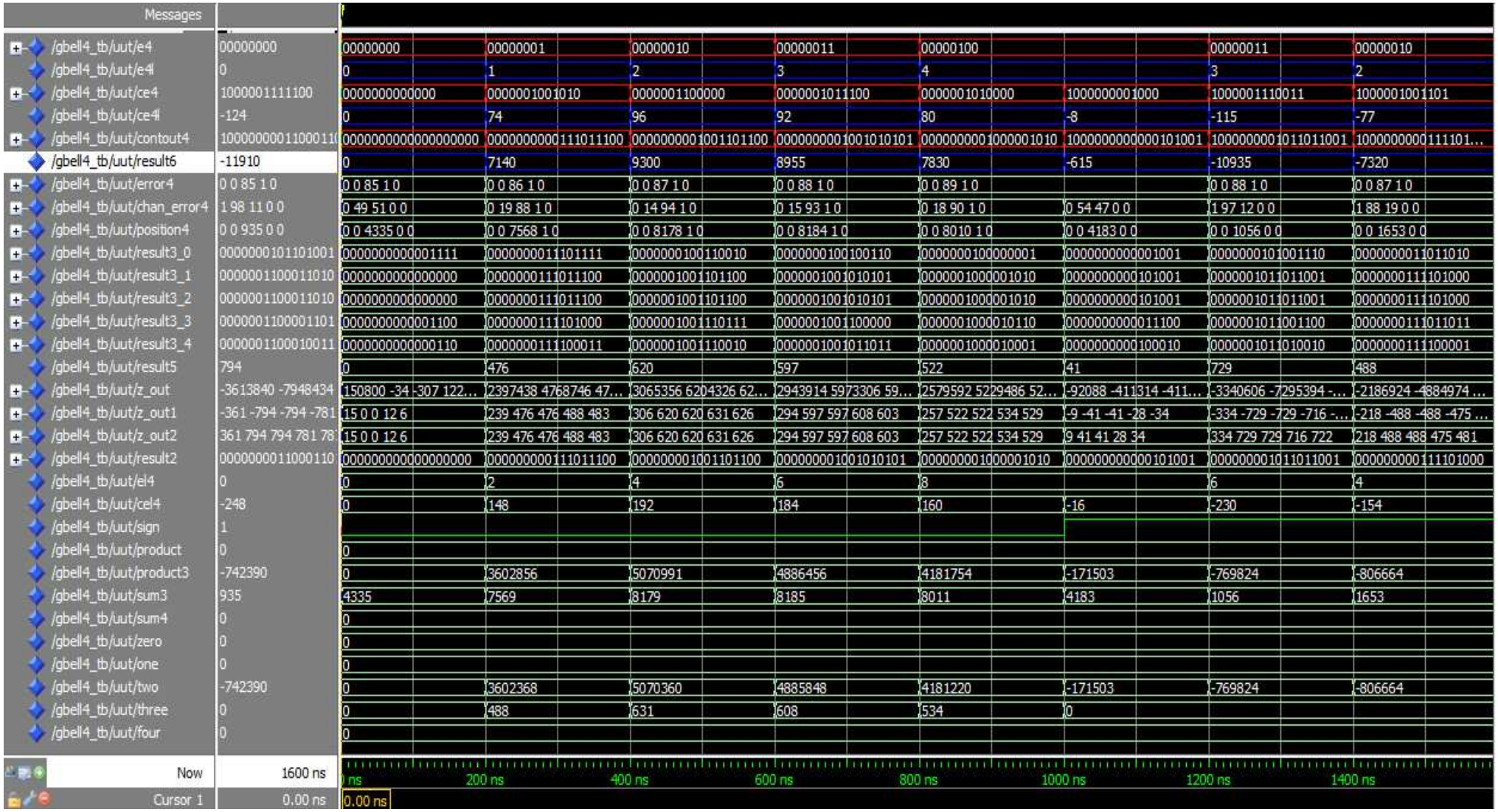


Figure 7.15 Samples of the Simulation Result of Fourth Neurofuzzy Controller

As mentioned before, the design of the Neurofuzzy logic controller is implemented using VHDL codes and simulated using Xilinx Spartan XC3S500E FPGA board. The VHDL codes are synthesized for converting into RTL view of the neurofuzzy controller architecture as shown in Figure 7.16-7.19.

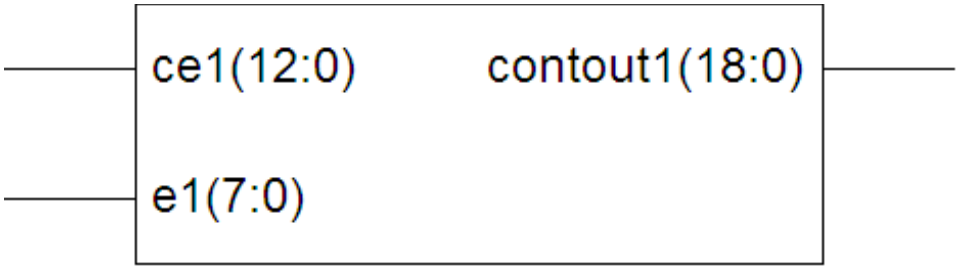


Figure 7.16 RTL View of the First Neurofuzzy Controller

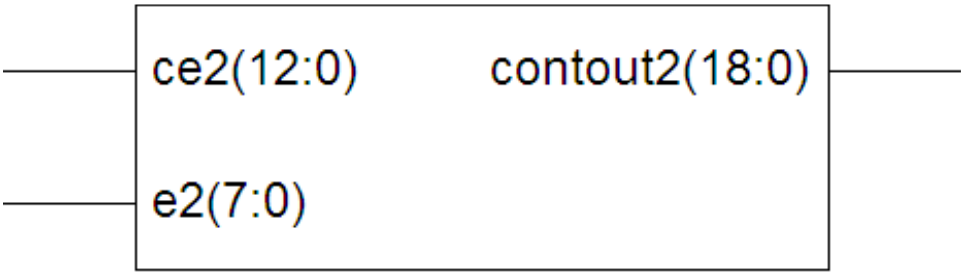


Figure 7.17 RTL View of the Second Neurofuzzy Controller

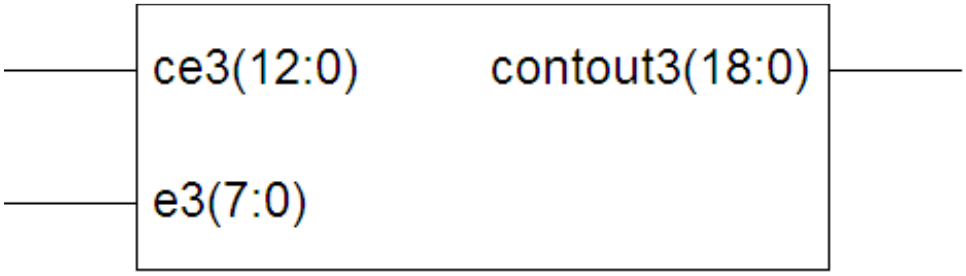


Figure 7.18 RTL View of the Third Neurofuzzy Controller

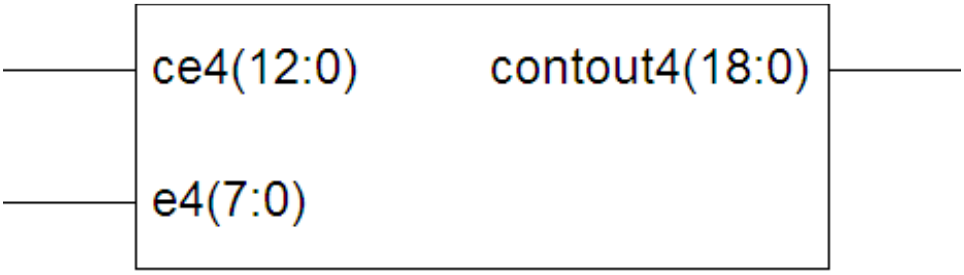


Figure 7.19 RTL View of the Fourth Neurofuzzy Controller

The technology mapping was chosen in this work as the Spartan XC3S500E with 4FG320 package and a speed grade of -4. The summary of the synthesize results are given in Table 7.1-7.4 for the first NF controller, second NF controller, third NF controller and fourth NF controller, respectively.

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	8728	9312	93%
Number of occupied Slices	4585	4656	98%
Number of Slices containing only related logic	4585	4585	100%
Number of Slices containing only unrelated logic	0	4585	0%
Number of bonded IOBs	40	232	17%
Number of MULT 18×18SIOs	15	20	75%

Table 7.1 Device Utilization Summary for First Neurofuzzy Controller

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	8582	9312	92%
Number of occupied Slices	4493	4656	96%
Number of Slices containing only related logic	4493	4493	100%
Number of Slices containing only unrelated logic	0	4493	0%
Number of bonded IOBs	40	232	17%
Number of MULT 18×18SIOs	18	20	90%

Table 7.2 Device utilization summary for second neurofuzzy controller

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	7569	9312	81%
Number of occupied Slices	3976	4656	85%
Number of Slices containing only related logic	3976	3976	100%
Number of Slices containing only unrelated logic	0	3976	0%
Number of bonded IOBs	40	232	17%
Number of MULT 18×18SIOs	16	20	80%

Table 7.3 Device utilization summary for third neurofuzzy controller

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	8727	9312	93%
Number of occupied Slices	4564	4656	98%
Number of Slices containing only related logic	4564	4564	100%
Number of Slices containing only unrelated logic	0	4585	0%
Number of bonded IOBs	40	232	17%
Number of MULT 18×18SIOs	17	20	85%

Table 7.4 Device utilization summary for fourth neurofuzzy controller

7.7 Summary

Four neurofuzzy controllers are used to design the control system for a full vehicle nonlinear active suspension system with nonlinear hydraulic actuators. The results of chapter 6 shown that the neurofuzzy controllers are more robust and more effective than the three types of other controllers (FOPID Controller, FMRL Controller and Neural Controller). Therefore, the neurofuzzy controllers have been implemented using an FPGA platform. The FPGA has been selected to design the neurofuzzy controller for many reasons, such as: it is suitable for fast implementation and quick hardware verification; the systems based on them are flexible and can be reprogrammed with unlimited number of times; and the rapid evaluation of silicon technologies has helped to reduce the size of FPGA integrated circuits and cost.

After the optimal trainable parameters of the neurofuzzy controllers are obtained, the VHDL codes are typed in the Xilinx ISE program. This program has many levels, such as: analysis and synthesis of VHDL codes, determination of the placements of the LEs in an actual FPGA chip, and generate of the configuration files to program the FPGA boards. The ModelSim program has been used to test the performance of the VHDL codes to make sure that the FPGA board which is used as the neurofuzzy controller system is working properly. Comparisons are made between the data obtained from the software designing of neurofuzzy controller using MATLAB program and the data obtained from the ModelSim software. The results illustrated that the control signals obtained from the ModelSim software are identical to the control signals which are obtained from the design simulation. Therefore, the FPGA boards have been demonstrated very effective in controlling the full vehicle nonlinear active suspension systems with hydraulic actuators.

8. Active Control of Electromagnetic Suspensions for Vehicle

8.1 Energy regenerative suspension

The main objective of vehicle suspension systems is to isolate the vehicle body from shock and vibration excited by road irregularities in order to maximize passenger riding comfort and also to secure continuous road wheel contact to the road, ensuring the vehicle's handling quality. As mentioned before, there are three types of suspension systems used to absorb the road vibration: the passive suspension system, semi-active suspension system and active suspension system.

The active suspension system is very effective to offer significant performance of vibration isolation, but they are high energy consumptive and complex. The active suspension is an essential technology for high performance machine system; therefore, many researches are making attempt to solve this energy problem.

When a vehicle is driving on a bumpy road, plus driver's acceleration and deceleration operations, there will be shocks between the sprung mass and the unsprung mass. This part of the mechanical power is normally converted into heat power by dampers and is dissipated in a natural way. If the wasted energy can be reclaimed in a proper way, the overall energy consumption required for the vehicle can be reduced.

Recently, the researchers have carried out research to reduce the fuel consumption required for the vehicle and decrease the harmful emission to protect the environment. In this direction, it has been considered that the energy used to be wasted as heat energy dissipated by the dampers can be recovered by a suitable energy regenerative device. In fact, there are several types of energy regenerative suspension designs already under investigation such as: hydraulic storage suspension (Shian et al. 2007), battery coil induction suspension (Ren et al. 2005), rack and pinion suspension (Shian et al. 2006), ball screw suspension (Yong-chao et al. 2008) and linear motion suspension (Zhengquan and Yu 2007).

In the recent years, worldwide attempts concentrated on vehicle vibration control and energy regeneration, theoretical and experimental progresses (Karnopp 1978; Karnopp 1989; Sada and Shiiba 1996; Velinsky and White 1980) have been achieved. In order to improve the performance of vehicle suspension systems, the developments achieved in power electronics, permanent magnet materials and microelectronic systems enable the possibility of actual implementation of electromagnetic actuators (Martins et al. 2006). In the reference by Montazeri and Soleymani (2010), the authors investigated the idea of an energy regeneration of active suspension systems in petrol-electric hybrid vehicles. A hybrid energy storage system, which consists of electrochemical batteries and super-capacitors, is proposed. In Reference Xu et al. (2010), a new hydraulic electromagnetic energy regenerative suspension design is proposed and a comparison is made with the other two energy regenerative suspension designs based on their structural and working principles, and an evaluation using the fuzzy comprehensive judgement is made. A retrofit regenerative shock absorber is designed that is characterized and tested to recover the vibration energy within a compact space in an efficient way (Zuo et al. 2010).

More recently, a new energy regenerative suspension prototype has been designed and developed by Xue Chun et al. at Shanghai Jiao Tong University (Xue-chun et al. 2008). This

prototype consists of two main parts: a ball screw mechanism and a brushless DC motor. The ball screw part converts the vertical vibration motion into rotational motion to rotate the DC motor's rotor. Depending on the direction of the current which flows through the motor's coils and stroke velocity ($\dot{z} - \dot{w}$), the DC motor may operate as an electromotor as well as a generator. When the DC motor operates as electromotor, the energy will be consumed from the battery to suppress the vehicle vibration so that the performance of the electromagnetic suspension behaves as an active suspension system. On the other hand, when it operates as a generator, the energy will be stored in the battery. The energy consumed by the active suspension system will be compensated by the energy generated from the electromagnetic actuator.

In this chapter, the nonlinear dampers are replaced by the electromagnetic actuators to solve the energy problem of the active suspension system.

Neurofuzzy controllers were designed to generate suitable control signals will be used in this chapter. The control signals will be applied as a control input signals to govern the hydraulic actuators to generate suitable damping forces for improving the vehicle performance. In this case, the DC motor will operate as a generator only and the reclaimed vibration energy will be converted into electrical energy to supply the electrical pumps of the hydraulic actuators.

8.2 Mathematical model of the electromagnetic actuator

The electromagnetic actuator consists of a permanent-magnet DC motor with a ball screws and a nut. Figure 8.1 shows the prototype of the electromagnetic actuator (from Reference Zhang et al. (2007)).

The electromagnetic actuator converts the vibration energy into electrical energy through the rotation of the DC motor and stores it in the battery. Therefore, the generated electrical energy can contribute to run the electrical pumps of the hydraulic actuators.

Figure 8.1 Prototype of the Electromagnetic Actuator

The DC motor can operate here in the electromotor mode or generator mode. In general, the power of the motor (P_e) can be represented as

$$P_e = \frac{4\pi\Phi}{P_H} vI \tag{8.1}$$

where Φ is the flux linkage; P_H is the lead of the ball-screw, v is the relative velocity ($\dot{z} - \dot{w}$); and I is the electric current flow through the motor's coils.

If the demand power is positive, the motor that operates in the electromotor mode and the current flows from the battery into the positive terminal of the motor and the energy of battery will be consumed. On the other hand, if the power is negative, the motor operates inversely and the current flows to the positive electrode of the battery and the motor charges the battery as a generator with reclaimed energy from vibration of the vehicle.

The torque of the DC motor (T_e) can be written as

$$T_e = C_e I \tag{8.2}$$

where the equivalent torque constant C_e can be expressed as

$$C_e = 2\Phi \tag{8.3}$$

There are two types of forces which can be generated by using an electromagnetic actuator. The first force is called damping force (F_m) which comes from the mechanism friction and inertia and it is given by

$$F_m = C_m(\dot{z} - \dot{w}) \tag{8.4}$$

The second force is called the vertical force which is given by

$$F_a = F_t \cot\varphi \tag{8.5}$$

where φ is the screw lead angle and F_t is the tangential force which is given by

$$F_t = \frac{T_e}{r} = \frac{C_e I}{r} \tag{8.6}$$

where r is the effective radius for force conversion.

The total force generated by the electromagnetic mechanism can be written as

$$F_c = F_m + F_a \tag{8.7}$$

Substituting Eq. 8.4 and 8.5 in Eq. 8.7 is yeilded

$$F_c = C_m(\dot{z} - \dot{w}) + \left(\frac{2\Phi cot\varphi}{r}\right) I \tag{8.8}$$

8.3 Simulation and results

As mentioned before, the nonlinear damper models shown in Figure 2.5 are replaced by the electromagnetic actuators. The design parameters of the electromagnetic actuator are listed in Table 8.1.

The neurofuzzy controllers designed in Chapter 6 have been used in this chapter to generate the control signals for the hydraulic actuators to improve the vehicle performance. The current flows through the coil of the DC motor of the i^{th} electromagnetic actuator is given by (Xue-chun et al. 2008)

$$I_i = k_p(z_i - w_i) - k_d(\dot{z}_i - \dot{w}_i) \tag{8.9}$$

where k_p and k_d are constants.

The MATLAB SIMULINK model illustrated in Figure 2.7 is modified to generate the damping force shown in Eq. (8.8) as shown in Figure 8.2.

In case of the electromagnetic actuators are used instead of the nonlinear damper models, then the responses of the vertical displacement at each corner, the vertical displacement at the center of gravity, roll angle and pitch angle will be improved.

Notation	Description	Values	Units
Cm	Damping force constant	200	N.s/m
Φ	Flux linkage	0.0759	V.s
P_H	Lead of the ball-screw	0.2	m
r	Effective radius for force conversion	0.008	m
φ	Thread lift angle	21.95	degree

Table 8.1 Design parameters of the electromagnetic actuator Huang et al. (2010)

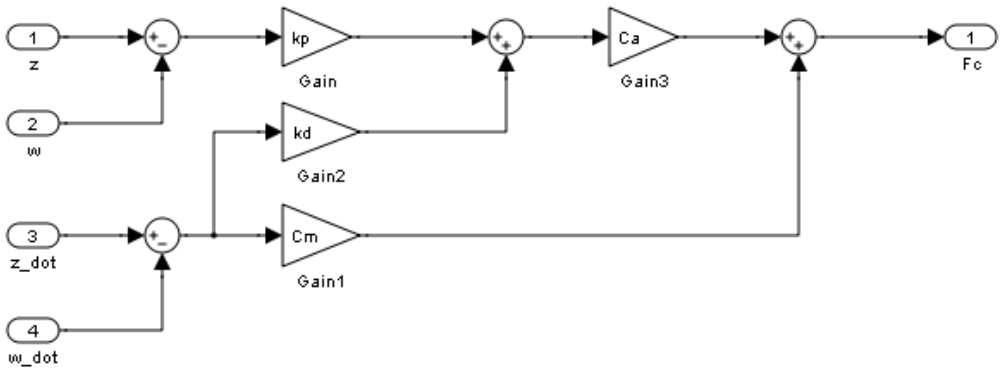


Figure 8.2 MATLAB Simulink model to generate the force of the electromagnetic actuator

Figures 8.3-8.9 show the comparison between the time outputs response of controlled system with the neurofuzzy controller and the corresponding time outputs response of the passive system.

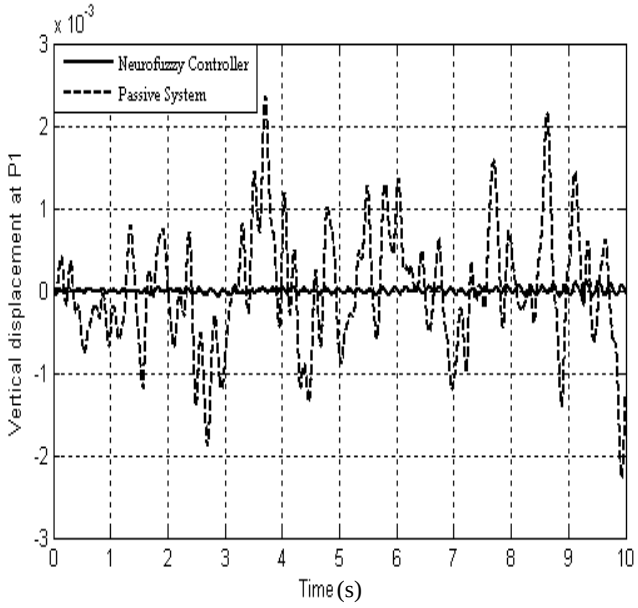


Figure 8.3 Time response of a vertical displacement at P_1

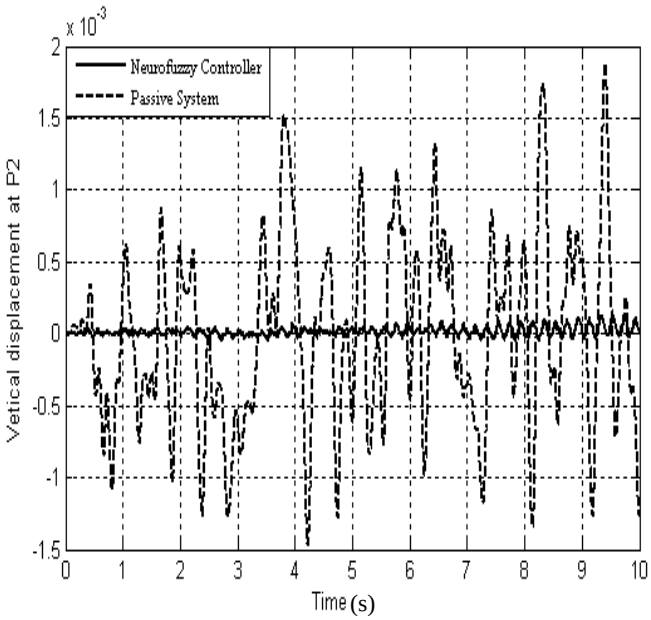


Figure 8.4 Time response of a vertical displacement at P_2

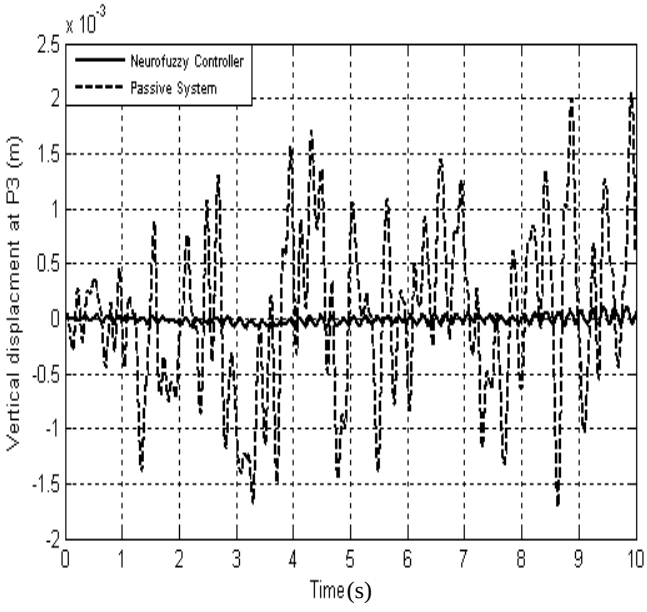


Figure 8.5 Time response of a vertical displacement at P_3

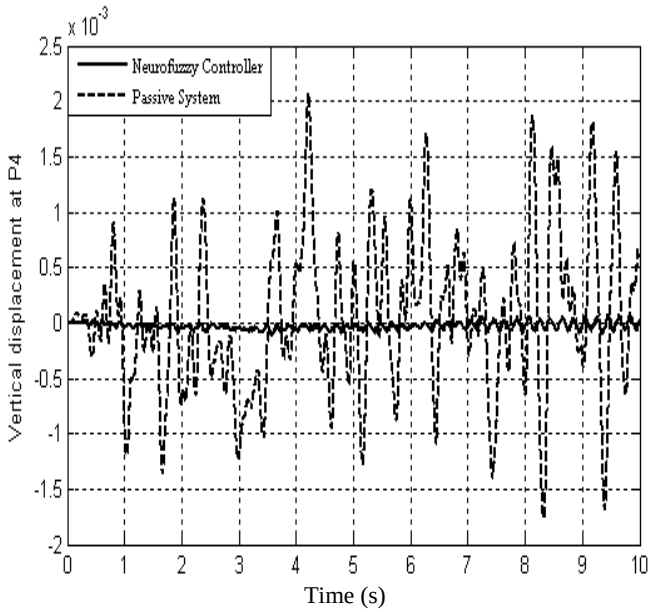


Figure 8.6 Time response of a vertical displacement at P_4

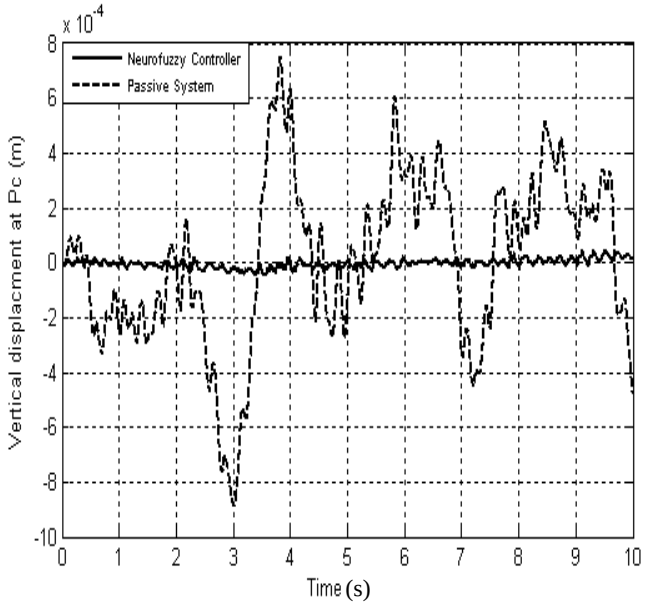


Figure 8.7 Time response of a vertical displacement at P_c

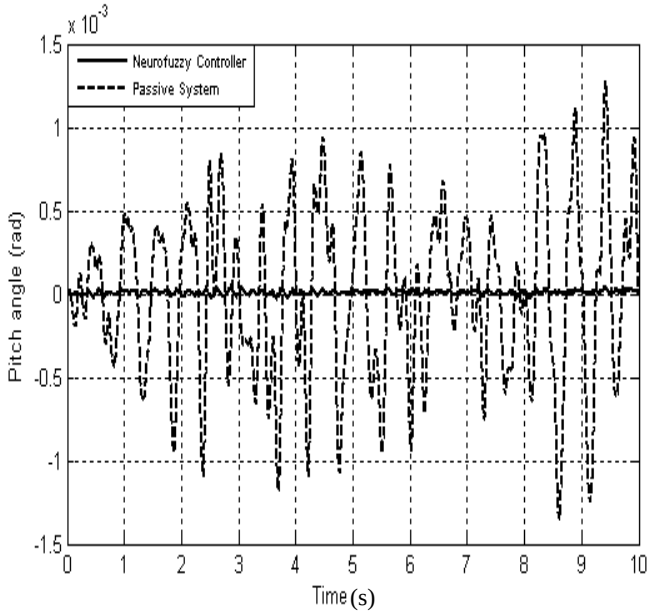


Figure 8.8 Time response of a pitch angle

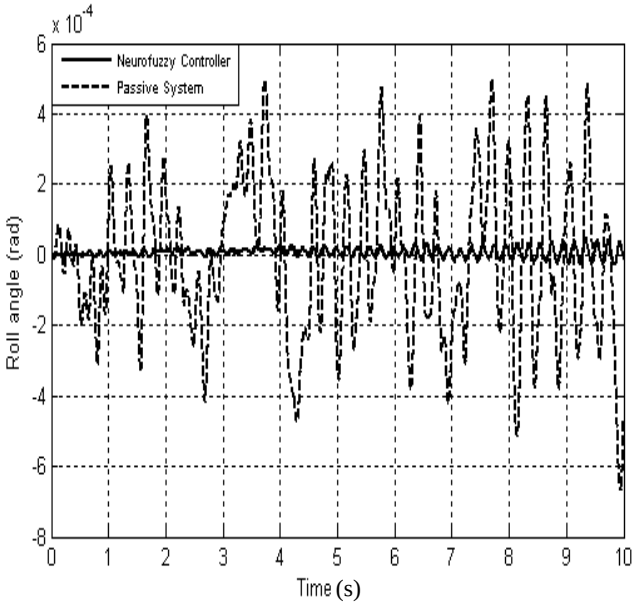


Figure 8.9 Time response of a roll angle

If the figures above are compared with Figures (6.8-6.18), it can be seen that the responses of the output signals of the controlled system with the neurofuzzy controller have been improved when the electromagnetic actuators are used. In another word by using the electromagnetic

actuators, not just the energy consumption is reduced but the riding comfort and road handling are improved as well.

The j^{th} consumption power by i^{th} hydraulic actuators can be calculated from the following equation:

$$P_{ci}(j) = F_i(j) * \left(\left(\dot{z}_i(j) - \dot{w}_i(j)\right) - (\dot{z}_i(j-1) - \dot{w}_i(j-1)\right) \quad (8.7)$$

where F_i is the output force of the i^{th} hydraulic actuator,

z_i is the vertical displacement at the i^{th} corner,

w_i is the vertical displacement at the i^{th} unsprung mass.

The total consumption power by the i^{th} hydraulic actuator can be calculated from the following equation:

$$P_{ti} = \sum_j F_i(j) * \left(\left(\dot{z}_i(j) - \dot{w}_i(j)\right) - (\dot{z}_i(j-1) - \dot{w}_i(j-1)\right) \quad (8.8)$$

Figures 8.10-8.13 show the consumption power by the front-right hydraulic actuator, front-left hydraulic actuator, rear-right hydraulic actuator and rear-left hydraulic actuator, respectively.

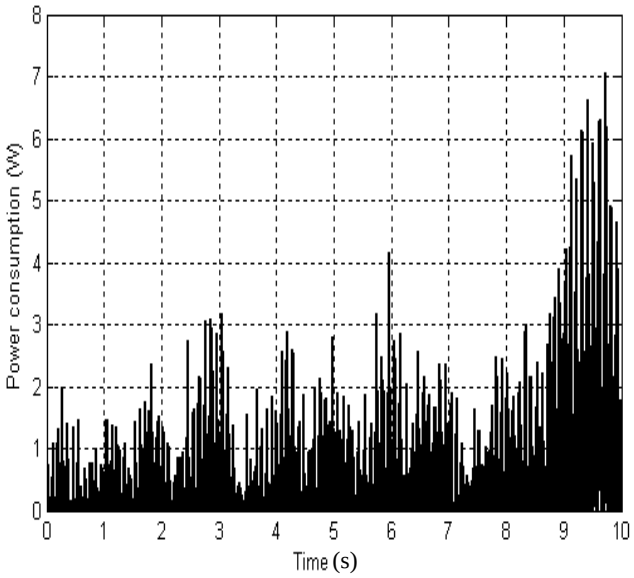


Figure 8.10 Power consumption by the front-right hydraulic actuator

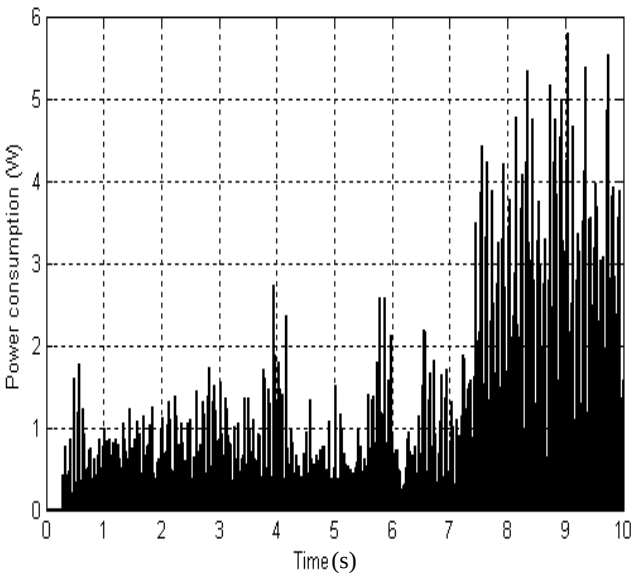


Figure 8.11 Power consumption by the front-left hydraulic actuator

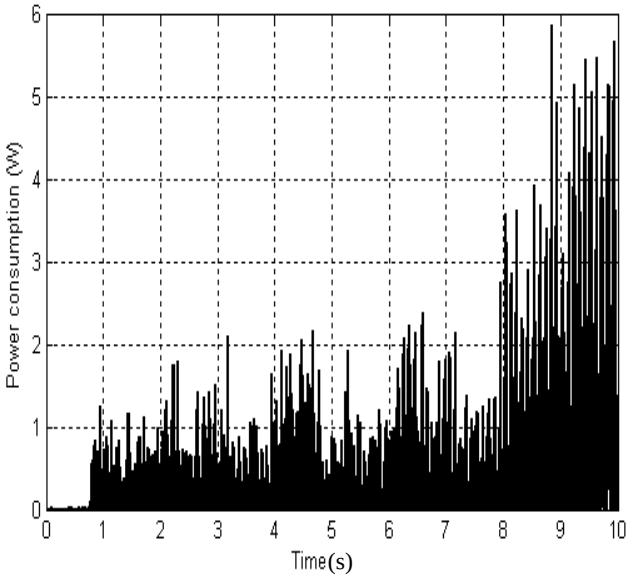


Figure 8.12 Power consumption by the rear-right hydraulic actuator

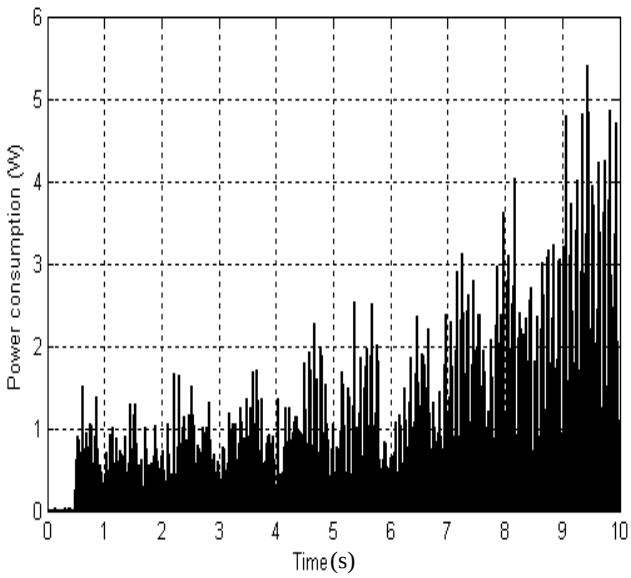


Figure 8.13 Power consumption by the rear-left hydraulic actuator

The power generated from the i^{th} electromagnetic actuator can be calculated from Eq. (8.1). Figures 8.14-18.17 show the power generated from the DC motors of the electromagnetic active suspension device.

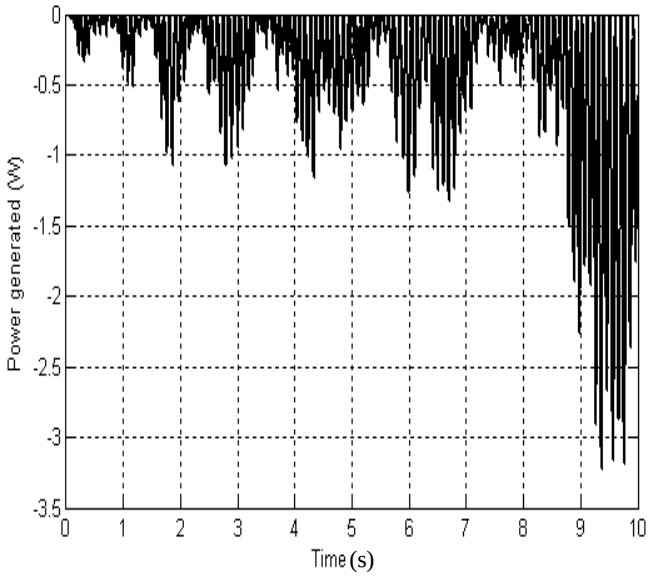


Figure 8.14 Output power from front-right suspension

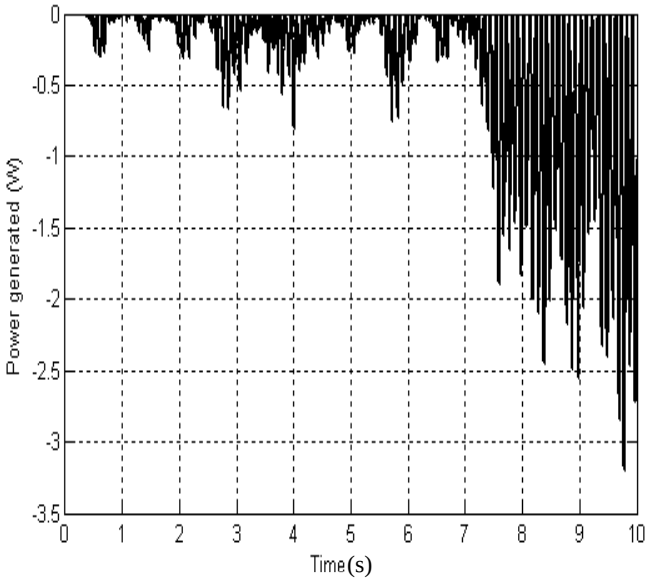


Figure 8.15 Output power from front-left suspension

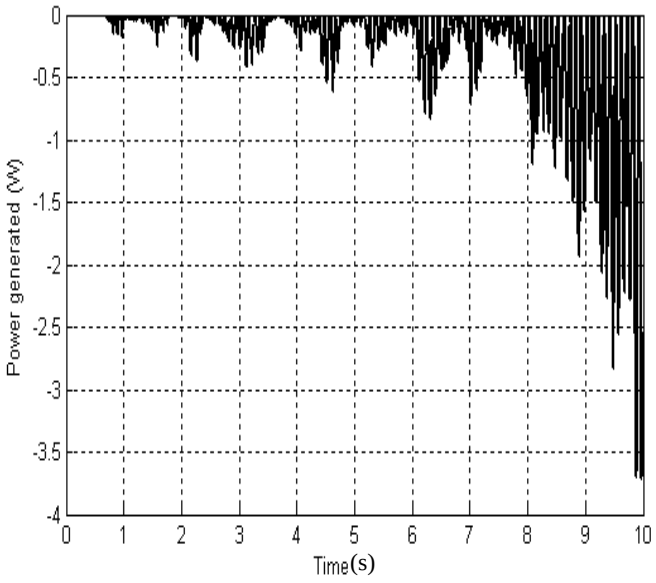


Figure 8.16 Output power from rear-right suspension

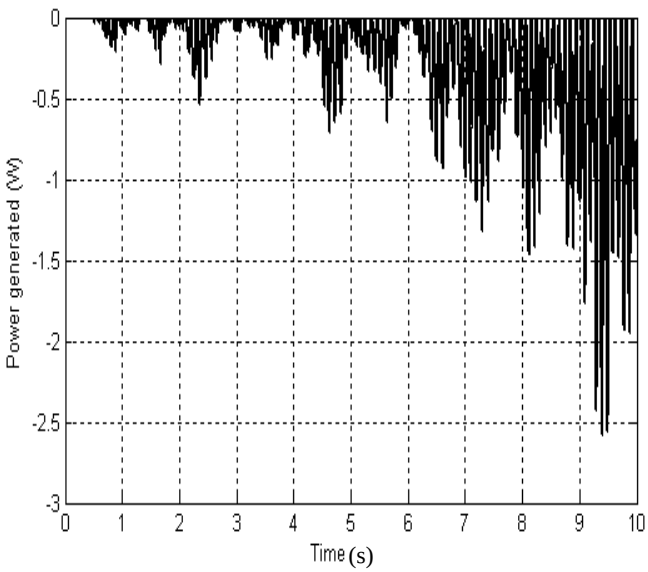


Figure 8.17 Output power from rear-left suspension

These figures illustrate that the DC motors operate as a generator only because the generated powers are negative. The energy will contribute to drive the pumps of the hydraulic actuators to generate damping forces as a part of the effort of improving the vehicle performance.

The results illustrate that about 50% of the consumed energy by the active suspensions can be saved by using the electromagnetic actuators. It means that the energy consumption by the active suspension systems has been reduced and the problem of the energy consumption resulting for driving the actuators in the active suspension has been solved.

8.4 Summary

The fuel consumption and harmful emission of the vehicles should be reduced to protect the environment. The active suspension systems, which have been used in this work, have great performance of vibration isolation i.e. they are better than the other two types of the suspension systems. But the main disadvantage of the active suspension systems is the highly energy consumption. Therefore, the nonlinear damper models have been replaced by

electromagnetic actuators which convert the vibration energy that arises from the rough road into electrical energy to reduce the energy consumption by the active suspension system. This electrical energy is used to run the electrical pumps of the hydraulic systems to generate the control forces which are applied between sprung mass and unsprung masses to improve the riding comfort and road handling

By applying the electromagnetic devices, the optimistic results have been found: the overall power consumption of the active suspension system has been reduced and the previous heat energy dissipated to the viscous fluid of the damper has been converted to useful electrical energy used to run the electrical pump in the hydraulic actuator device. When the electromagnetic device act as a generator, the power generated and stored in the battery will be used to run the pump of the hydraulic actuators to generate appropriate forces to improve the riding comfort and road handling. The results illustrate that the electromagnetic device acts as a power generator. In another word, the vibration energy excited by the rough road surface has been converted to useful electrical energy. Furthermore, when the nonlinear damper models are replaced by the electromagnetic actuators, the vertical displacement at each suspension, the vertical displacement at center of gravity, pitch angle response and roll angle response have been attenuated. As a result, by using hydraulic actuators with electromagnetic suspension systems, two targets have been met: increasing fuel economy and improving the vehicle performance.

9. Conclusions and Recommendations

9.1 Conclusions

In this work, full vehicle nonlinear active suspension model has been investigated, in which three motions, i.e. the vertical, pitching and rolling, are taken in to account. The intelligent control design problems for full vehicle nonlinear active suspension systems with hydraulic actuators have been investigated. The effects of the nonlinear hydraulic actuators with frictional forces are considered. The mathematical model for full vehicle nonlinear active suspension systems with nonlinear hydraulic actuators has been derived. The results have shown that the passive suspension systems without control are not sufficient to effectively reduce the vibrations arisen from travelling on the rough road or to minimize the rolling and pitching movements when sharp manoeuvres take place due to braking and cornering. Therefore, robust control systems should be designed.

Fractional order $PI^{\lambda}D^{\mu}$ controllers are proposed to design as control systems, where all output variables including vertical displacement at the centre of gravity, vertical displacement, roll angle and pitch angle, depended on the vertical displacements at the corner points are under control. The suspension deflection and body acceleration are used to evaluate the road handling and riding comfort of the passengers, respectively. By supplying the control signal just the vertical displacements at each corner and the body acceleration are reduced while the vertical displacements of unsprung masses will be unaffected. The results illustrate that the output responses of the controlled system have been slightly improved when the FOPID

controllers are used. The vertical displacement at each suspended corner and at the centre of gravity are still not negligible, i.e. they may still be sensed by the passengers. On the other hand, when the bending torque is applied, or the braking torque applied, with random inputs as road profile, the oscillation of the output responses are eliminated, but the steady state outputs are still unchanged. Furthermore, when the typical disturbances are applied, the results show that the performance of the cost function of the system with FOPID controller only has been slightly improved. However when different values of bending torque are applied as the output disturbance, the performance of the cost function for the system with proposed controller is close to the performance of the cost function for the passive system. It follows that when different values of the bending torque are applied, the proposed controllers will lose their robustness.

Fuzzy model reference learning controllers are designed. The results of the controller implementation indicated that the proposed controllers pressed the outputs of the controlled system responses to be smaller than the corresponding outputs of the passive system. On the other hand, when the bending effect is applied as an input to the controlled system, the oscillations of the system outputs are eliminated but the steady state errors are still away from the zero. However, in this case, the response of the vertical displacement at the centre of gravity and the pitch angle response have not been improved. Furthermore, the results illustrated that when the braking effect is applied as the input road profile to the controlled system, the oscillations of the system outputs are eliminated but the steady state errors still exist. In this case, just the roll angle response has not been improved.

Six different road profiles acted as the system inputs were applied as disturbances to assess the robustness of the proposed controllers. The performances of the cost functions of the proposed controller, when different types of disturbances applied, have been compared with the performances of the cost functions of the corresponding passive system. The results

illustrated that the proposed controller only slightly improves the performance of the cost functions.

Neural controllers are also designed. The results illustrated that the neural controllers have capability of minimizing the control objectives better than the two previous controllers. The riding comfort and the road handling have been improved using the neural controller. When the bending or braking torque is applied with random inputs as the road profile, the oscillations of the controlled system outputs are eliminated and the output steady state errors become very small.

Six types of disturbances have been applied individually to test the robustness of the proposed neural controller. The results demonstrated that the neural controller is still stable and it suppresses the cost function to be minimum even significant disturbances have occurred.

Neurofuzzy controllers are designed. The results have shown that the responses of the vertical displacements at each suspension corner, the vertical displacement at the centre of gravity, the pitching and rolling movement have been reduced to very small value when the NF controllers are employed. In addition, the results illustrated that when the NF controllers are used, the vertical displacements at each corner and body acceleration are reduced to very small values which means that the riding comfort and road handling are improved. When the bending torque or braking torque is applied, the output responses of the controlled system are reduced more than the corresponding outputs responses of the controlled system when the other types of control are used. Six types of the disturbances have also been applied to test the robustness of the proposed controller. The results prove that the NF controller is still stable and it suppresses the cost function to be minimum even significant disturbances have occurred. The results have been confirmed that when the NF controller has been used, the cost functions has a small value while when the other types of control systems are used the cost functions are much bigger.

The NF controllers have been implememted using FPGA platforms. The results show that the control signals obtained from ModelSim program are identical to the control signals obtained from design simulation. Therefore, it can be arrived at that the FPGA boards are effective for neurofuzzy controllers of the full vehicle nonlinear active suspension systems.

To solve the energy problem of the active suspension system, the nonlinear dampers are replaced by the electromagnetic actuators. The results illustrated that when the electromagnetic actuators are used, about 50% of the energy consumption by the active suspension system can be reclaimed. Furthermore, the vertical displacement at each suspension; the vertical displacement at centre of gravity, the pitch angle response and the roll angle response all have been improved. As a result, two targets have been met by using hydraulic actuators with electromagnetic suspension systems: increasing the fuel economy and improving the vehicle performance.

9.2 Recommendations for future work

The aim of any work in control field is to improve the performance of a controller or to propose a new control scheme. Several recommendations about future works are listed below:

1) In this work a bell shape function is used as a membership function for accomplishment of the fuzzification process. Other membership functions can also be used, such as Gaussian function, triangle function or trapezoid function. The formulas of those functions are simpler than that of the bell shape function. Therefore, the update equations used to upgrade the parameters of the neurofuzzy controller will be simpler.
2) To validate the robustness of the proposed neurofuzzy controller, the cases with six types of disturbances with dry asphalts road have been investigated. The

performance of the proposed neurofuzzy controller can be demonstrated by simulation under other road conditions, e.g. snowy road, wet asphalt or transitions between such conditions (the road switches from snowy road to wet asphalt) to assess the robustness of the neurofuzzy controller.

3) The numerical values of the hydraulic actuators and full vehicle model used in the simulation are taken from pervious data referenced by Alleyne and Hedrick (1995) and Park and Kim (1998) respectively. Just the data referencd by first reference are obtained experimentally. Therefore, the response of the real active suspension with proposed neurofuzzy controller might not match the simulation one. In this case, the parameters of the neurofuzzy controller should be refined. Therefore, the proposed neurofuzzy controller should move to the direction of real active suspension systems to obtain the real values of the neurofuzzy controller. The real system can be tested on real road profiles with different disturbances and conditions to assess its effectiveness.

4) In this work, neural network has been employed to obtain the optimal values of the parameters fuzzy logic systems. Other techniques can be proposed, such as genetic algorithm, to obtain the parameters of the fuzzy system. It might also yield good results.

References

[1] Adams, W. B. (1971). *English Pleasure Carriages*, London: Charles Knight and Co.

[2] Alleyne, A., and Hedrick, J. K. (1995). "Nonlinear Adaptive Control of Active Suspensions " *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, 3(1), 94-101.

[3] Ando, Y., and Suzuki, M. (1996). "Control of Active Suspension Systems Using the Singular Perturbation Method." *Control Eng. Practice*, 4(3), 287-293.

[4] Areed, F., Haikal, A., and Mohammed, R. (2010). "Adaptive Neuro-fuzzy Control of an Induction Motor." *Ain Shams Engineering Journal*, 1-8.

[5] Astrom, K. J., and Hagglund, T. (1995). *PID controller: Therory, Design and Tuning* USA: Instrument Society of America, Research Triangle Park.

[6] Back, T. (1996). *Evolutionary Algorithms in Theory and Practice*: Oxford University Press, London, UK.

[7] Barriga, A., Sanchez-Solano, S., Brox, P., Cabrera, A., and Baturone, I. (2006). "Modelling and Implementation of Fuzzy Systems based on VHDL." *International Journal of Approximate Reasoning*, 41, 164-178.

[8] Biglarbegian, M., Melek, W., and Golnaraghi, F. (2008). "A Novel Neuro-fuzzy Controller to Enhance the Performance of Vehicle Semi-active Suspension Systems." *Vehicle System Dynamics*, 46(8), 691-711.

[9] Biswas, A., Das, S., Abraham, A., and Dagupta, S. (2008). "Design of Fractional-order PID Controllers with an Improved Differential Evolution." *Engineering Applications of Artificial Intelligence*, 22, 343-350.

[10] Blake, J., Maguire, L., McGinnity, T., Roche, B., and McDaid, L. (1998). "The implementation of fuzzy systems, neural networks and fuzzy neural networks using FPGAs." *Information Science*, 112, 151-168.

[11] Braghin, F., Resta, F., and Sabbioni, E. (2007). "AModal Control for Active/Semi-Active Suspension Systems " *IEEE*.

[12] Brown, S., and Vranesic, Z. (2005). *Fundamentals of Digital Logic with VHDL Design* New York: The McGraw-Hill Companies, Inc.

[13] Buckley, J. J., and Ying, H. (1989). "Fuzzy Controller Theory: Limit Theorems for Linear Fuzzy Control Rules." *Automatica*, 25(3), 469-472.

[14] Bui, T. H., Suh, J. H., and B., K. S. (2002). "Hybrid Control of an Active Suspension System with Full-Car Model Using Hoo and Nonlinear Adaptive Control Methods." *KSME International Journal*, 16(12), 1613-1626.

[15] Campos, J., Lewis, F. L., Davis, L., and Ikenaga, S. (2000). "Backstepping Based Fuzzy Logic Control Of Active Vehicle Suspension Systems " *Proceedings of the American Control Conference Chicago, Illinois.*, 4030-4035.

[16] Canete, J. F., Perez, S., and Orozco, P. (2008). "Software Tools for System Identification and Control Using Neural Networks in Process Engineering " *World Academy of Science, Engineering and Technology* 47, 59-63.

[17] Cao, J., and Cao, B. (2006). "Design of Fractional Order Controller Based on Particle Swarm Optimization " *International Journal of Control, Automotion, and systems*, 4(6), 775-781.

[18] Cao, J., Liang, J., and Cao, B. (2005). "Optimization of Fractional Order PID Controllers Based on Genetic Algorithms " *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou*, 5686-5689.

[19] Catania, V., Puliafito, A., Russo, M., and Vita, L. (1994). "A VLSI Fuzzy Inference Processor Based on a Discrete Analoge Approch." *IEEE Trans. Fuzzy System*, 2, 93-106.

[20] Chamseddine, A., Noura, H., and Raharijaona, T. (2006). "Control of Linear Full Vehicle Active Suspension System Using Sliding Mode Techniques." *Proceedings of the 2006 IEEE International Conference on Control Applications Munich, Germany.*, 1306-1311.

[21] Chang, Y. (2009). "N-Dimension Golden Section Search: Its Variants and Limitations." *2nd International conference on Biomedical Engineering and Informatics, BMEI'09*, 1-6.

[22] Chen, P., and Huang, A. (2006). "Adaptive Sliding Control of Active Suspension Systems with Uncertain Hydraulic Actuator Dynamics." *Vehicle System Dynamics*, 44(5), 357-368.

[23] Chen, W., Yuan, H., and Wang, Y. (2009). "Design and implementation of Digital Fuzzy-PID controller based on FPGA "*4th IEEE Conference on Industrial Electronics and Applications, ICIEA' 09.* 393-397.

[24] Chen, Y., and Teng, C. (1995). "A Model Reference Control Structure Using a Fuzzy Neural Network." *Fuzzy Sets and Systems*, 73, 291-312.

[25] Cheng, C. P., and Li, T. S. (2007). "EP-based Fuzzy Control Design for an Active Suspension System with Full-car Model." *IEEE International Conference on Systems, Man and Cybernetics.* , 3288-3293.

[26] Chiang, C., Chung, H., and Lin, J. (1997). "A Self-Learning Fuzzy Logic Controller Using Genetic Algorithms with Reinforcements." *IEEE Transactions on fuzzy systems*, 5(3), 460-467.

[27] Choi, S.-B., Lee, H.-S., and Park, Y.-P. (2002). "H_{oo} Control Performance of a Full-Vehicle." *Vehicle System Dynamics*, 38(5), 341-360.

[28] Chu, P. (2008). *FPGA Prototyping by VHDL Examples Xilinx Sparten-3 Version*, New Jersey: John Wiley and Sons, Inc.

[29] Cirstea, M., Khor, J., and McCormick, M. (2001). "Fpga Fuzzy logic Controller for variable Speed Generator." *IEEE Inernational Conference on Control Application*, 5-7.

[30] Cobo, J., Van Noije, W., and Gualberto, L. "VHDL models for high level synthesis of fuzzy logic controllers." *Presented at XI Brazilian Symposium on Integrated Circuit Design, 1998. Proceedings.* , Rio de Janeiro.

[31] D'Amato, F. J., and Viassolo, D. E. (2000). "Fuzzy Control for Active Suspensions." *Mechatronics* 10, 897-920.

[32] Damousis, I. G., Satsios, K. J., Labridis, D. P., and Dokopoulos, P. S. (2002). "Combined Fuzzy Logic and Genetic Algorithm Techniques Application to an Electromagnetic Problem." *Fuzzy Sets and Systems*, 192, 371-386.

[33] Djukanovic, M., Calovc, M., Vescvic, B., and Sobajic, D. (1997). "Neurofuzzy Controller of Low Head Hydropower Plants Using Adaptive Network Based Fuzzy Inference System." *IEEE Transaction on Energy Conversion*, 12(4), 375-381.

[34] Dorcak, L., Peteras, L., Kostial, I., and Terpak, J. (2001). "State-space Controller Design for Fractional Order Regulated System." *Proceedings of the International Carpathian Control Conference*, 15-20.

[35] Du, H., and Zhang, N. (2009). "Fuzzy Control for Nonlinear Uncertain Electrohydraulic Active Suspensions With Input Constraint." *IEEE Transactions on Fuzzy Systems*, 17(2), 343-356.

[36] Economakos, G., and Economakos, C. (2007). "A Run-Time Recongurable Fuzzy PID Controller Based on Modern FPGA Devices"*Mediterranean Conference on Control and Automation*. City, pp. 1-6.

[37] Esmailzadeh, E., and Fahimi, F. (1997). "Optimal Adaptive Active Suspensions for a Full Car Model " *Vehicle System Dynamics*, 27, 89-107.

[38] Feng, J. Z., Li, J., and Yu, F. (2003). "GA-Based PID and Fuzzy Logic Control for Active Vehicle Suspension System." *International Journal of Automotive Technology*, 4(4), 181-191.

[39] Fialho, I. J., and Balas, G. J. (2000). "Design of Nonlinear Controllers for Active Vehicle Suspensions Using Parameter-Varying Control Synthesis." *Vehicle System Dynamics*, 33, 351-370.

[40] Figueiredo, M., Ballini, S., Soares, S., Andrade, M., and Gomide, F. (2004). "Learning Algorithm for a Class of Neurofuzzy Network and Application." *IEEE Transaction on Systems,Man, and Cybernetics*, 34(3), 293-301.

[41] Foda, S. G. (2001). "Neuro-Fuzzy Control of a Semi-active Car Suspension System " *IEEE*, 2, 686-689.

[42] Freeman, R. (1988). "Uers programmable Gate Array " *IEEE Spectrum*, 32-35.

[43] Gabbar, H., Akinlade, D., Sayed, H., and Osunleke, A. (2007). "Neurofuzzy-Based Learning Algorithm for Fault Detection & Simulation "*SICE Annual Conference 2007*. Japan, 2286-2291.

[44] Ghartemani, M. K., Zamani, M., Sadati, N., and Parniani, M. (2007). "An Optimal Fractional Order Controller for an AVR System Using Particle Swarm Optimization Algorithm " *Conference on Power Engineering, Large Engineering Systems* 244-249.

[45] Gilbert, B. (1984). "*A monolithic 16-channel Analogy array Normaliser*." *IEEE Journal of solid state circuits*, 19(6), 956 -963.

[46] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*: Reading, MA: Addison-Wesley

[47] Gonzalez, J., Castillo, O., and Aguilar, L. "FPGA as a Tool for Implementing Non-fixed Structure Fuzzy Logic Controllers." *Presented at Proceedings of the 2007 IEEE symposium on Foundations of Computational Intelligence, FOCI'07*.

[48] Guclu, R., and Gulez, K. (2008). "Neural Network Control of Seat Vibrations of a Non-linear Full Vehicle Model Using PMSM." *Mathematical and Computer Modelling*, 47, 1356-1371.

[49] Gupta, R., Kumar, R., and Surjuse, R. (2009). "ANFIS Based Intelligent Control of Vector Controlled Induction Motor Drive"*Second International Conference on Emerging Trends in Engineering and Technology ICETET '09*. City, pp. 674-680.

[50] Gurocak, H. B., and Lazaro, A. S. (1994). "A Fine Tuning Method for Fuzzy Logic Rule Bases " *Fuzzy Sets and Systems*, 67, 147-161.

[51] Hoffmann, F. (2001). "Evolutionary Algorithms for Fuzzy Control System Design." *Proceedings of the IEEE* . 89(9), 1318-1333.

[52] Hopfield, J. J. (1982). "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of the National Academy of sciences* 79, 2554-2558.

[53] Hsia, T. C. (1977). *System Identification Least Squares Methods*: D.C. Heath and Company.

[54] Hu, S., and Li, J. (1996). "The Fuzzy PID Gain Conditioner: Algorithm, Architecture and FPGA Implementation." *IEEE Inernational Conference on Industrial Technology*, 621-624.

[55] Huang, C., Li, T. S., and Chen, C. (2009). "Fuzzy Feedback Linearization Control for MIMO Nonlinear System and Its Application to Full-Vehicle Suspension System." *Circuits Syst Signal Process*, 28, 959-991.

[56] Huang, K., Fan, Y., and Zhang, Y. (2010). "Study on Active Control of Electromagnetic Suspension Based on Energy Flow Analysis"*10th International Symposium on Advanced Vehicle Control AVEC'10*. City: Loughborugh, UK, pp. 315-320.

[57] Huang, S. J., and Chen, H. (2006). "Adaptive Sliding Controller with Self-tuning Fuzzy Compensation for Vehicle Suspension Control." *Mechatronics*, 16, 607-622.

[58] Huang, S. J., and Lin, W. C. (2007). "A Neural network Based Sliding Mode Controller for Active Vehicle Suspension." *Proc. IMechE Automobile Engineering*, 221 part D, 1381-1397.

[59] Hussain, M. (1999). "Review of the Applications of Neural Networks in Chemical Process Control. Simulation and On-line Implementations." *Artificial Intelligence engineering* 13, 55-68.

[60] Ikenaga, S., Lewis, F. L., Campos, J., and Davis, L. (2000). "Active Suspension Control of Ground Vehicle based on a Full-Vehicle Model." *Proceedings of the American Control Conference*, 4019-4025.

[61] Ikenaga, S., Lewis, F. L., Davis, L., Campos, J., Evans, M., and Scully, S. (1999). "Active Suspension Control Using a Novel Strut and Active Filtered Feedback: Design and Implementation." *Proceedings of the 1999 LEEE International Conference on Conuol Applications Kohala Coast-Island of Hawai. Hawai, USA.*, 1502-1508.

[62] Ishibuchi, H., Nozaki, K., and Tanaka, H. (1994). "Empirical Study on Learning in Fuzzy Systems by Rice Taste Analysis " *Fuzzy Sets and Systems*, 64, 129-144.

[63] Ishikamn, T., Tsukui, Y., and Matsunami, M. (1996). "Optimization of Electromagnetic Device Using Artificial Neural Network with Quasi Newton Algorithm." *IEEE Transaction on Magnetics*, 32(3), 1226-1229.

[64] Ishizuka, O., Tanno, K., Tang, Z., and Matsumoto, H. (1992). "Design of a fuzzy Controller with Normalization Circuit"*IEEE International Conference on Digital Objective Identifier*. 1303-1308.

[65] Ismail, A. (1998). "Fuzzy Model Reference Learning Control of Multi-stage Flash Desalination Plants " *Desalination*, 116, 157-164.

[66] Jain, K. K., and Asthana, R. B. (2006). *Autombile Engineering* London: TaTa Mc Graw-Hill.

[67] Jang, J. (1993). "ANFIS: Adaptive-Network-Based Fuzzy Inference System " *IEEE Transaction on Systems,Man, and Cybernetics*, 23(3), 665-685.

[68] Jang, J. (1996). "Input Selection for ANFIS Learining "*Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*. 1493-1499.

[69] Jang, J., and Mizutani, E. "Levenberg-Marquardt Method for ANFIS Learning " *Presented at Conference of the North American Fuzzy Information Processing Society NAFIPS*, Berkeley, CA, USA.

[70] Jang, J., and Sun, C. "Neuro-Fuzzy Modeling and Control " *Presented at Proceeding of the IEEE*.

[71] Jianmin, S., and Yang, Q. (2007). "Automotive Suspension System with an Analytic Fuzzy Control Strategy " *IEEE International Conference on Vehicular Electronics and Safety ICVES*, 1-4.

[72] Joo, D. S., AL-Holou, N., Weaver, J. M., Lahdhirf, T., and Al-Abbas, F. (2000). "Nonlinear Modeling of Vehicle Suspension System " *Proceedings of the American Control Conference Chicago, Illinois*, 115-119.

[73] Jung, S., and Kim, S. (2007). "Hardware Implementation of a Real-Time Neural Network Controller With a DSP and an FPGA for Nonlinear Systems." *IEEE Transactions on Industral Electronics*, 54(1), 265-271.
[74] Kamnik, R., Matko, D., and Banjd, T. (1998). "Application of Model Reference Adaptive Control to Industrial Robot Impedance Control." *Journal of Intelligent and Robotic Systems*, 22, 153-163.
[75] Karkoub, M. A., and Zribi, M. (2001). "Optimal control of suspension systems using smart actuator." *IEEE*, 87-94.
[76] Karnopp, D. (1978). "Power requirements for traversing uneven roadways." *Vehicle System Dynamics*, 7, 135-152.
[77] Karnopp, D. (1989). "Permanent Magent Linear Motors used as Variable Mechanical Dampers for Vehicle Suspensions." *Vehicle System Dynamics*, 18, 187-200.
[78] Kiani, A., and Pariz, N. (2007). "Fractional PID Controller Design based on Evolutionary Algorithms for Robust two-inertia Speed Control " *First Joint on Fuzzy and Intelligent Systems*.
[79] Kim, C., and Lee, J. (2003). "Adaptive Network based Fuzzy Inference System with Pruning"*SICE Annual Conference in Fukui*. City: Fukui University, Japan 140-143.
[80] Kim, D. (2000). "An Implementation of Fuzzy Logic Controller on the Reconfigurable FPGA System." *IEEE Transactions on Industral Electronics*, 47(3), 703-715.
[81] Kohonen, T. (1977). *Associative Memory: A System Theoretical Approach*, Berlin: Springer-Verlag.
[82] Kovacic, Z., Bogdan, S., and Balenovic, M. (1997). "A Sensitivity-Based Self-Learning Fuzzy Logic Controller as a Solution for a Backlash Problem in a Servo System." *IEEE International Electric Machines and Drives Conference Record* TC2/11.1 - TC2/11.3
[83] Kovacic, Z., Bogden, S., and Crnosija, P. (1995). "Design and Stability of Self-organizing Fuzzy Control of High-order Systems " *IEEE International Symposium on Intelligent Control*, 389-394.
[84] Krtolica, R., and Hrova, D. (1990). "Optimal Active Suspension Control Based on a Half-Car Model." *Proceedings of the 29th IEEE Conference on Decision and Control*, 4, 2238-2243.
[85] Kumar, M., and Garg, D. "Intelligent Learning of Fuzzy Logic Controllers Via Neural Network and Genetic Algorithm." *Presented at Japan-USA Symposium on Flexible Automation JUSFA*, Denver, Colorado.
[86] Kumar, M. S. (2007). "Genetic Algorithm-Based Proportional Derivative Controller for the Development of Active Suspension System " *Information Technology and Control*, 36(1), 58-65.
[87] Kuo, Y., and Li, T. S. (1999). "GA-Based Fuzzy PI/PD Controller for Automotive Active Suspension System." *IEEE Transactions on Industrial Electronics*, 46(6), 1051-1056.
[88] Layne, J. R., and Passino, K. M. (1992). "Fuzzy Model Reference Learning Control." *Journal of Intelligent and Fuzzy Systems*, 4, 33-47.
[89] Layne, J. R., and Passino, K. M. (1993). "Fuzzy Model Reference Learning Control for Cargo Ship Steering " *Proceedings of the 1993 International Symposium Conference on Intelligent Control Chicago,USA.*, 457-462.
[90] Lera, G., and Pinzolas, M. (2002). "Neighborhood Based Levenberg Marquardt Algorithm for Neural Network Training." *IEEE Transaction on neural Networks*, 13(5), 1200-1203.
[91] Lin, J.-S., and Huang, C.-J. (2004). "Nonlinear Backstepping Active Suspension Design Applied to a Half-Car Model." *Vehicle System Dynamics*, 42(6), 373-393.

[92] Liu, H., Nonami, K., and Hagiwara, T. (2005). "Semi-active Fuzzy Sliding Mode Control of Full Vehicle and Suspensions." *Journal of Vibration and Control* 11, 1025-1042.

[93] Ma, C., and Hori, Y. (2004). "Fractional Order Control and its Application of Fractional Order PI^{a}D Controller for Robust Two-Inertia Speed Control." *Proc. of the 4th International Power Electronics and Motion Control Conference*, 1477-1482.

[94] Maiti, D., Biswas, S., and Konar, A. (2008). "Design of a Fractional Order PID Controller Using Particle Swarm Optimization Technique " *2nd National Conferenace on Recent Trends in Information Systems*.

[95] Manzoul, M. A., and Jayabharathi, D. (1992). "Fuzzy Controller on FPGA Chip." *Ieee Inernational Conference on Digital Objective Identifier*, 1309-1316.

[96] Martins, I., Esteves, J., Marques, G., and Silva, F. (2006). "Permanent-Magnets Linear Actuators Applicability in Automobile Active Suspensions." *IEEE Transaction on Vehicular Technology,*, 55(1), 86-94.

[97] Mayhan, P., and Washington, G. (1998). "Fuzzy Model Reference Learning Control: a New Control Paradigm for Smart Structures." *Smart Mater. Struct.*, 7, 874-884.

[98] Meng, L., and Xue, D. (2009). "Design of an Optimal Fractional-order PID Controller Using Multi-Objective GA Optimization." *Control and Decision Conference. CCDC '09. Chinese*, 3849-3853.

[99] Merritt, H. (1969). *Hydraulic Control Systems*, USA: John wiley and Sons, Inc.

[100] Montazeri, M., and Soleymani, M. (2010). "Investigation of the Energy Regeneration of Active Suspension Systemin Hybrid Electric Vehicles." *IEEE Transaction on Industral Electronics*, 57(3), 918- 925.

[101] Moon, S. Y., and Kwon, W. H. (1998). "Genetic-based Fuzzy Control for Half-car Active Suspension Systems." *International Journal of Systems Science*, 29(7), 699-710.

[102] Muthuramalingam, A., Himavathi, S., and Srinivasan, E. (2007). "Neural Network Implementation Using FPGA: Issues and Application." *International Journal of Information Technology*, 4(2), 86-92.

[103] Narendra, K., and Parthasarathy, K. (1990). "Identification and Control of Dynamical Systems Using Neural Network" *IEEE Transaction on Neural Network*, 1(1), 4-27.

[104] Negoita, M., Neagu, D., and Palade, V. (2005). *Computational Intelligence: Engineering of Hybrid Systems*, Germany: Springer-Verlag Berlin Heidelberg.

[105] Norgaard, M., Ravn, O., and Poulsen, N. (2001). "NNSYSID and NNCTRL Tools for System Identification and Control with Neural Networks." *Computing and Control Engineering Journal* 23, 29-36.

[106] Obaid, A., Sulaiman, N., Marhaban, M., and Hamidon, M. (2010). "Analysis and Performance Evaluation of PD-like Fuzzy Logic Controller Design Based on Matlab and FPGA." *International Journal of Computer Science*, 37(2).

[107] Onat, C., Kucukdemiral, I. B., Sivrioglu, S., and Yuksek, I. (2007). "LPV Model Based Gain-scheduling Controller for a Full Vehicle Active Suspension System" *Journal of Vibration and Control*, 13, 1626-1666.

[108] Park, J. H., and Kim, Y. S. (1998). "Decentralized Variable Structure Control for Active Suspensions Based on a Full-Car Model " *Proceedings of the 1998 IEEE International Conference on Control Applications Trieste, Italy.*, 383-387.

[109] Passino, K. M. (1998). *Fuzzy Control*, Canda: Addison Wesley Longman.

[110] Peng, C., and Magoulas, D. (2007). "Adaptive Non monotone Conjugate Gradient Training Algorithm For Recurrent Neural Networks"*19th IEEE International Conference on Tool with Artificial Intelligence.* 374-381.

[111] Peters, L., Beck, K., and Camposano, R. (1993). "Adaptive Fuzzy Controller Improves Comfort." *Second IEEE International Conference on Fuzzy Systems*, 1, 512-516.

[112] Poorani, S., Priya, T., Kumar, K., and Renganarayanan, S. (2005). "FPGA Based Fuzzy Logic Controller for Electric Vehicle." *Journal of the Institution of Engineers* 45(5), 1-14.

[113] Qiao, W. Z., Zhuang, W. P., and Heng, T., H. . (1992). "A Rule Self-regulating Fuzzy Controller " *Fuzzy Sets and Systems*, 47, 13-21.

[114] Rajamani, R., and Hedrick, J. K. (1995). "Adaptive Observers for Active Automotive Suspensions: Theory and Experiment " *IEEE Transactions on Control Systems Technology*, 3(1), 86-93.

[115] Rani, S., Kanagasabapathy, P., and Kumar, A. (2005). "Digital Fuzzy Logic Controller using VHDL". *IEEE Indicon 2005 Conference*: India, pp. 463-466.

[116] Rashidi, F., and Vahedi, A. (2004). "Sensorless Speed Control of Induction Motor Derives Using a Robust and Adaptive Neuro-Fuzzy Based Intelligent Controller"*IEEE International Conference on Industrial Technology ICIT '04*. Iran, 617-627.

[117] Ren, H., Shian, C., and Senlin, L. (2005). "A Permanent Magnetic Energy Regenerative Suspension." *ZL 200520072480.9*.

[118] Renn, J., and Wu, T. (2007). "Modeling and Control of a New 1/4 Servo-Hydraulic Vehicle Active Suspension System." *Journal of Marine Science and Technology*, 15(3), 265-272.

[119] Rezaeeian, A., Koma, A., Shasti, B., and Doosthoseini, A. (2008). "ANFIS Modeling and Feedforward Control of Shape Memory Alloy Actuators." *International Journal of Mathematical Models and Methodes in Applied Sciences*, 228-235.

[120] Rigler, A. K., Irvine, J. M., and Vogl, T. P. (1991). "Rescaling of Variables in Back Propagation Learning." *Neural Network*, 4, 225-229.

[121] Rumelhart, D., Hinton, G., and Willams, R. (1986). "Learning Represatations by Back-propagation Error." *Nature*, 533-536.

[122] Sada, Y., and Shiiba, T. (1996). "A New Hybrid Suspension System with Active Control and Energy Regeneration " *Vehicle System Dynamics*, 25, 641-654.

[123] Salapura, V., and Hamann, V. (1996). "Implementing fuzzy control systems using VHDL and statecharts"*Design Automation Conference, 1996, with EURO-VHDL '96 and Exhibition, Proceedings EURO-DAC '96* City, pp. 53-59.

[124] Samarasinghe, S. (2007). "Neural Network for Applied Sciences and Engineering." *Taylar and Francis Group LLC*.

[125] Sanchez-Solano, S., Cabrera, A., Baturone, I., Moreno-Valo, F., and Brox, M. (2007). "FPGA Implementation of Embedded Fuzzy Controllers for Robotic Applications." *IEEE Transation on Industrial Electronics*, 54(4), 1937-1945.

[126] Shariati, A., Taghirad, H. D., and Fatehi, A. (2004). "Decentralized Robust H∞ Controller Design for a Half-Car Active Suspension System." *Control 2004*.

[127] Sharkaway, A. (2005). "Fuzzy Control for the Automobiles Active Suspension System." *Vehicle System Dynamics*, 43(11), 795-806.

[128] Sharkawy, A. B. (2005). "Fuzzy and Adaptive Fuzzy Control for the Automobiles' Active Suspension System." *Vehicle System Dynamics*, 43(11), 795-806.

[129] Sharp, R. S., and Goodall, J. R. (1969). "A Mathematical Model for the Simulation of Vehicle Motions " *Journal of Engineering Mathematics*, 3(3), 219-237.

[130] Shian, C., Ren, H., and Senlin L. (2006). "Operation Theory and Structure Evaluation of Reclaiming Energy Suspension." *Transactions of the Chinese Society for Agricultural Machinery*, 37(5), 5-9.

[131] Shian, C., Ren, H., and Senlin L. (2007). "New Reclaiming Energy Suspension and its Working Principle " *Chines Journal of Mechanical Engineering*, 13(11), 177-182.

[132] Singh, B., Goyal, R., Kumar, R., and Singh, R. (2009). "Design and VLSI implementation of Fuzzy Logic Controller." *International Journal of Computer and Network Security*, 1(3), 77-81.

[133] Spentzas, K. N., and Kanarachos, S. A. (2002). "A Neural Network Approach to the Design of a Vehicle's Non-linear Hybrid Suspension System." *Proc Instn Mech Engrs*, 216 part B, 833-838.

[134] Sumathi, S., Hamsapriya, T., and Surekha, P. (2008). *Evolutionary Intelligence An Introduction to Theory and Applications with Matlab*: Spring Verlag Berlin Heidelberg, German.

[135] Sung, K.-G., Han, Y.-M., Lim, K.-H., and Choi, S.-B. (2007). "Discrete-time fuzzy sliding mode control for avehicle suspension system featuring an electrorheological fluid damper." *Smart Materials and Structures*, 16, 798-808.

[136] Suratgar, A. A., Tavakoli, M. B., and Hoseinabadi, A. "Modified Levenberg Marquardt Method for Neural Networks Training." *Presented at Proceedings of World Academy of Science, Engineering and Technology*.

[137] Szaszi, I., Gaspar, P., and Bokor, J. (2002). "Nonlinear Active Suspension Modelling Using Linear Parameter Varying Approach." *Proceedings of the 10th Mediterranean Conference on Control and Automation - MED2002 Lisbon, Portugal*.

[138] Tahboub, K. A. (2005). "Active Nonlinear Vehicle-Suspension Variable-Gain Control " *Proceedings of the 13th Mediterranean Conference on Control and Automation Limassol, Cyprus.*, 569-574.

[139] Tasaka, Y. (1989). "Hybrid Bus Type Fuzzy Controller with Analog Fuzzy Chips." *Proc. IFSA Word Congr. Seattle, WA*, 280-293.

[140] Tipsuwanpornm, V., Runghimmawan, T., Intajag, S., and Krongratana, V. (2004). "Fuzzy logic PID controller based on FPGA for process control." *IEEE International Symposium on Industrial Electronics*, 2(1495-1500).

[141] Tollenaere, T. (1990). "SuperSAB: Fast Adaptive Back Propagation with Good Scaling Properties." *Neural Network*, 3, 561-573.

[142] Tusset, A. M., Rafikov, M., and Balthazar, J. M. (2008). "An Intelligent Controller Design for Magnetorheological Damper Based on a Quarter-car Model." *Journal of Vibration and Control*, 15, 1907-1920.

[143] Velinsky, S., and White, R. (1980). "Vehicle Energy Dissipation due to Road roughness." *Vehicle System Dynamics*, 9, 359-384.

[144] Vinagre, B. M., Podlubny, L., Dorcak, L., and Feliu, V. (2000). "On Fractional PID Controllers: A Frequency Domain Approach." *Proc. of IFAC Workshop on Digital Control - Past Presentand Future of PID Control*, 53-58.

[145] Vogl, T. P., Mangis, J. K., Rigler, A. K., Zink, W. T., and Alkon, D. L. (1988). "Accelerating the Convergence of the Backpropagation Method." *Biological Cybernetics*, 59, 257-263.

[146] Vuong, P., Madni, A., and Vuong, J. (2006). "VHDL Implementation For a Fuzzy Logic Controller "*Automation Congress, 2006. WAC '06. World* City, pp. 1-8.

[147] Wilamowski, B. M., Iplikci, S., Kaynak, O., and Efe, M. O. (2001). "An Algorithm for Fast Convergence in Training Neural Networks"*International Joint Conference on Neural Networks*. Washington, DC. 1778-1782.

[147] Wn, S. J., Chiang, H., Chen, J., and Lee, T. (2004). "Optimal Fuzzy Control Design for Half-Car Active Suspension Systems " *Proceedings of the 2004 IEEE International Conference on Networking. Sensing & Control Taipei, Taiwan.*, 583-588.

[148] Wu, S.-J., Wu, C.-T., and Lee, T.-T. (2005). "Neural-Network-Based Optimal Fuzzy Control Design for Half-Car Active Suspension Systems " *Intelligent Vehicles Symposium. Proceedings. IEEE*, 376 - 381

[149] Xu, L., Guo, X., and Liu, J. (2010). "Evaluation of Energy-regenerative Suspension Structure Based on Fuzzy Comprehensive Judgment." *Advanced Materiral Research*, 139-141, 2636-2642.

[150] Xue-chun, Z., Fan, U., and Yong-chao, Z. (2008). "A Novel Energy Regenerative Active Suspension for Vehicles." *J. Shanghai Tiaotong University*, 13(2), 184-188.

[151] Xue, D., Chen, Y., and Atherton, D. P. (2007). *Linear Feedback Control Analysis and Design with MATLAB*, USA: The soceity for Industrial and Applied Mathematics

[152] Xue, D., Zhao, C., and Chen, Y. (2006). "Fractional Order PID Control of a DC-Motor with Elastic Shaft: a Case Study." *Proceedings of the 2006 American Control Conference Minneapolis, Minnesota, USA.*, 3182-3187.

[153] Yagiz, N., and Sakman, L. E. (2005). "Robust Sliding Mode Control of a Full Vehicle Without Suspension Gap Loss." *Journal of Vibration and Control*, 11(11), 1357-1374.

[154] Yagiz, N., and Sakman, L. E. (2006). "Fuzzy Logic Control of a Full Vehicle without Suspension Gap Degeneration " *Int. J. Vehicle Design*, 42, 198-212.

[155] Yamakawa, T. "*A Fuzzy Programmable Logic Array (Fuzzy PLA).*" *Presented at EEE International Conference on Digital Objective Identifier*.

[156] Yong-chao, Z., Fan, U., Yong-hui, G., and Xue-chun, Z. (2008). "Isolation and Energy regenarative Performance Experimental Verification of Automotive Electrical Suspension." *Journal of Shanghai Jiaotong University*, 42(6), 874-877.

[157] Yoshimura, T., Nakaminami, K., Kurimoto, M., and Hino, J. (1999). "Active suspension of passenger cars using linear and fuzzy-logic controls." *Control Engineering Practice*, 7, 41-47.

[158] Yu, X., and Dent, D. (1994). "Implementing neural networks in FPGAs."*IEE Colloquium on Hardware Implementation of Neural Networks and Fuzzy Logic* London, Uk, 1-5.

[159] Yue, L., Tang, C., and Li, H. (2008). "Research on Vehicle Suspension System Based on Fuzzy Logic Control"*International Conference on Automation and Logistics*. Qingdao, China, 1817-1821.

[160] Zhang, Y., Huang, K., Yu, F., Gu, Y., and Li, D. (2007). "Experimental Verification of Energy-regenerative Feasibility for an Automotive Electrical Suspension System"*IEEE International Conference on Vehicular Electronics and Safety*. 1-5.

[161] Zhang, Y., and Kandel, A. (1998). "Compensatory Neurofuzzy System with Fast Learning Algorithm." *IEEE Transaction on Neural Networks*, 9(1), 83-105.

[162] Zheng, L., Li, Y. N., Shao, J., and Sun, X. S. (2007). "The Design of a Fuzzy-sliding Mode Controller of Semi-active Suspension Systems with MR Dampers " *IEEE International Conference on Fuzzy Systems and knowledge Discovery.*, 4, 514-518.

[163] Zhengquan, W., and Yu, C. (2007). "Brief Introduction to Structure and Principle of Electromagnetic Shock Absorber." *Motor Technology*, 8, 56-59.

[164] Zuo, L., Scully, B., Shestani, J., and Zhou, Y. (2010). "Design and Characterization of an Electromagnetic Energy Harvester for Vehicle Suspensions." *Smart Mater. Structure.*, 19, 1-10.

Appendices

Appendix 1 Structure of the proposed system model

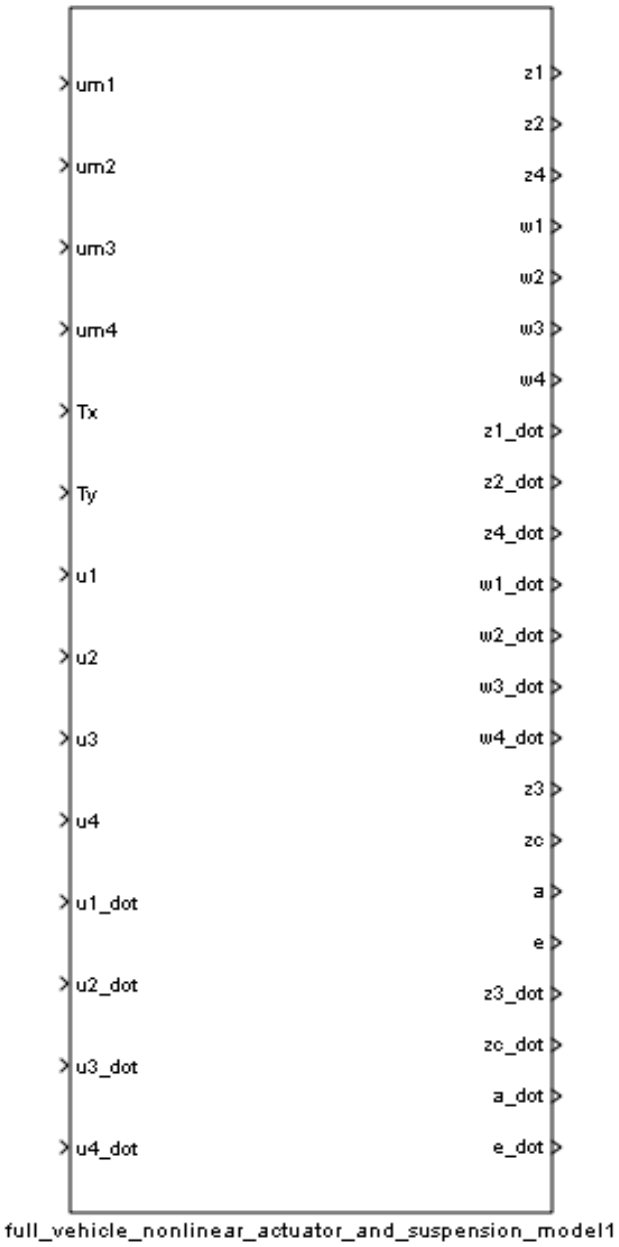


Figure A1.1 MATLAB SIMULINK model of full vehicle nonlinear active suspensions systems

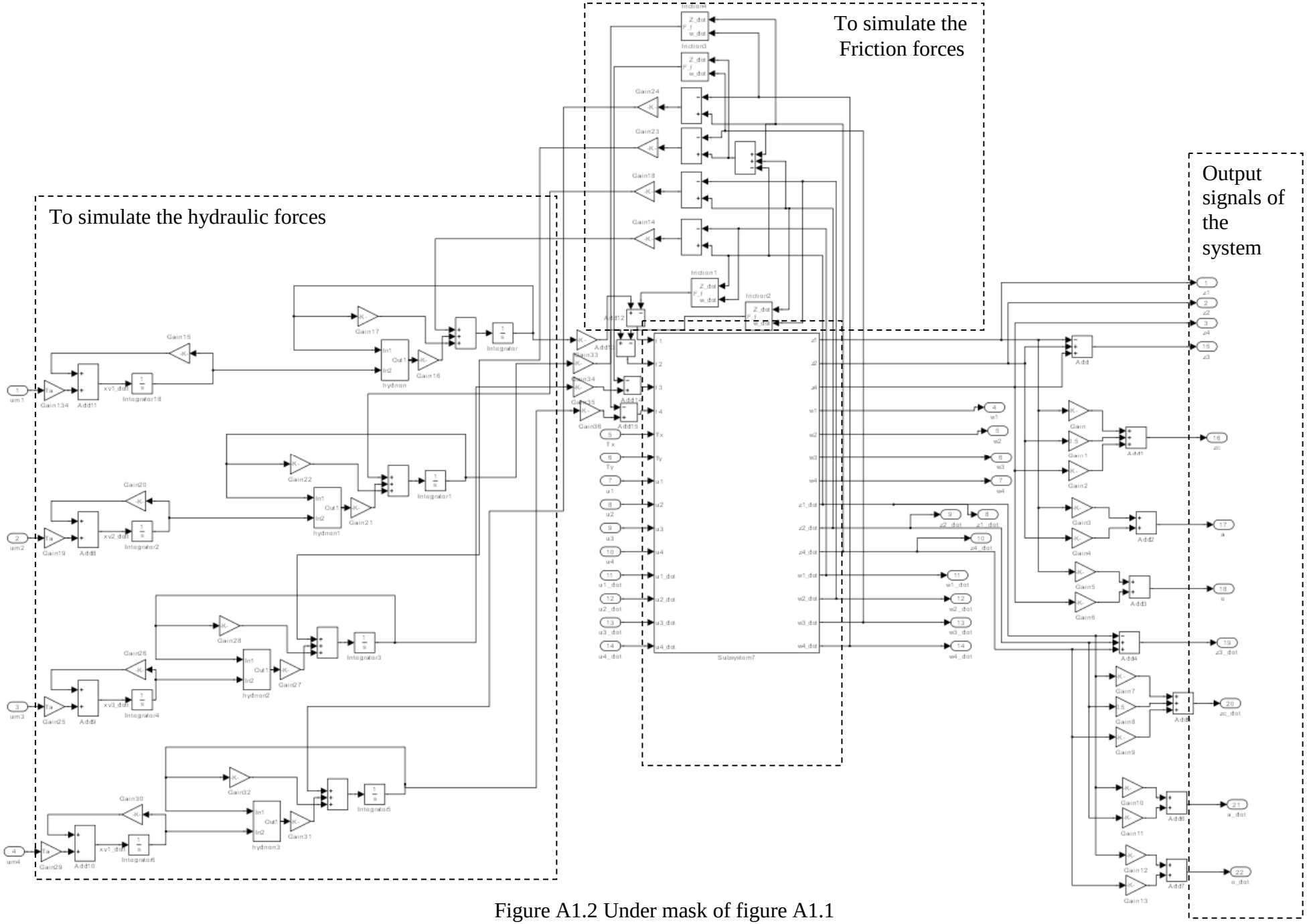


Figure A1.2 Under mask of figure A1.1

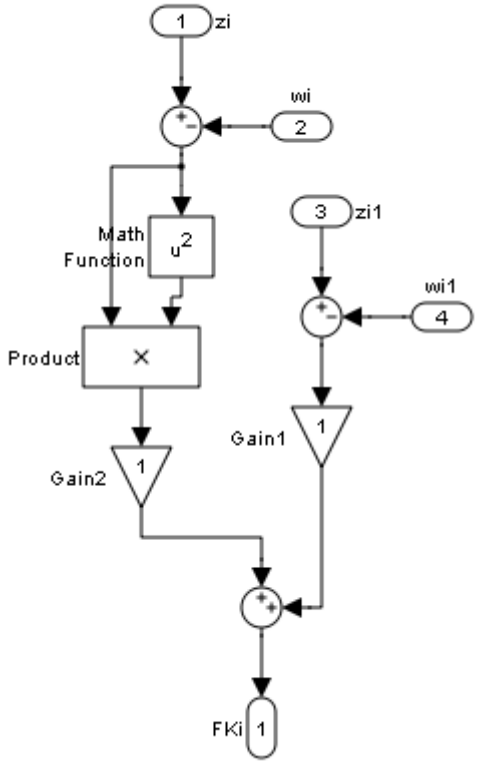


Figure A1.3 MATLAB SIMULINK model of spring's nonlinear force

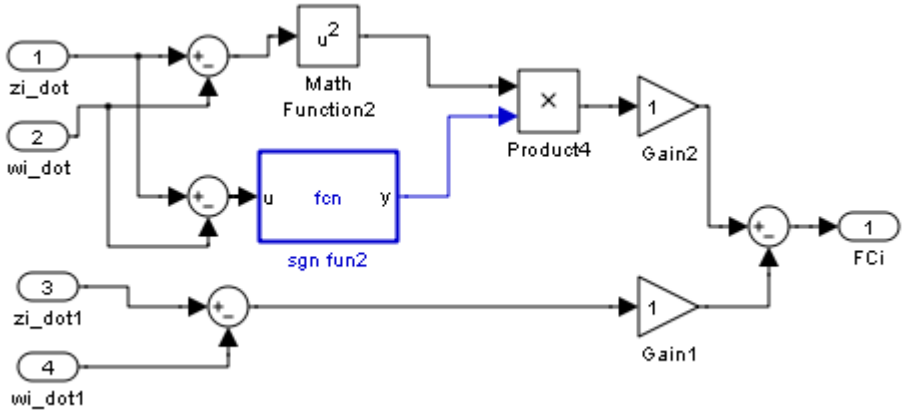


Figure A1.4 MATLAB SIMULINK model of damper's nonlinear

The MATLAB script file of the S_Function (the sigmoid function) which is used in Figure A1.4 is shown below

```
function y = fcn(u)
if u>0
   y=1;
elseif u==0
        y=0;
    else  u<0
            y=-1;
end
end
```

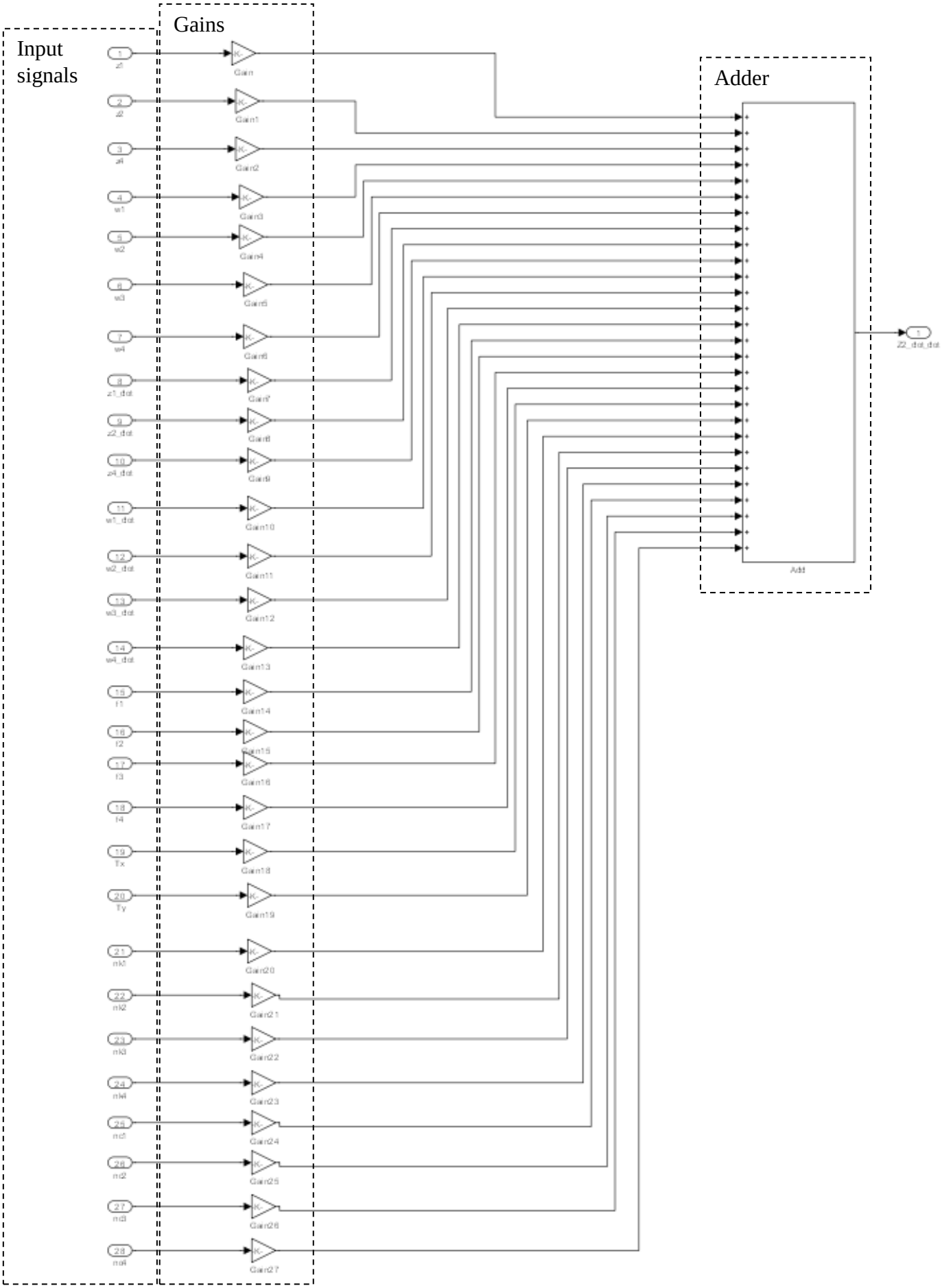


Figure A1.5 MATLAB SIMULINK model of $\ddot{z}_i$

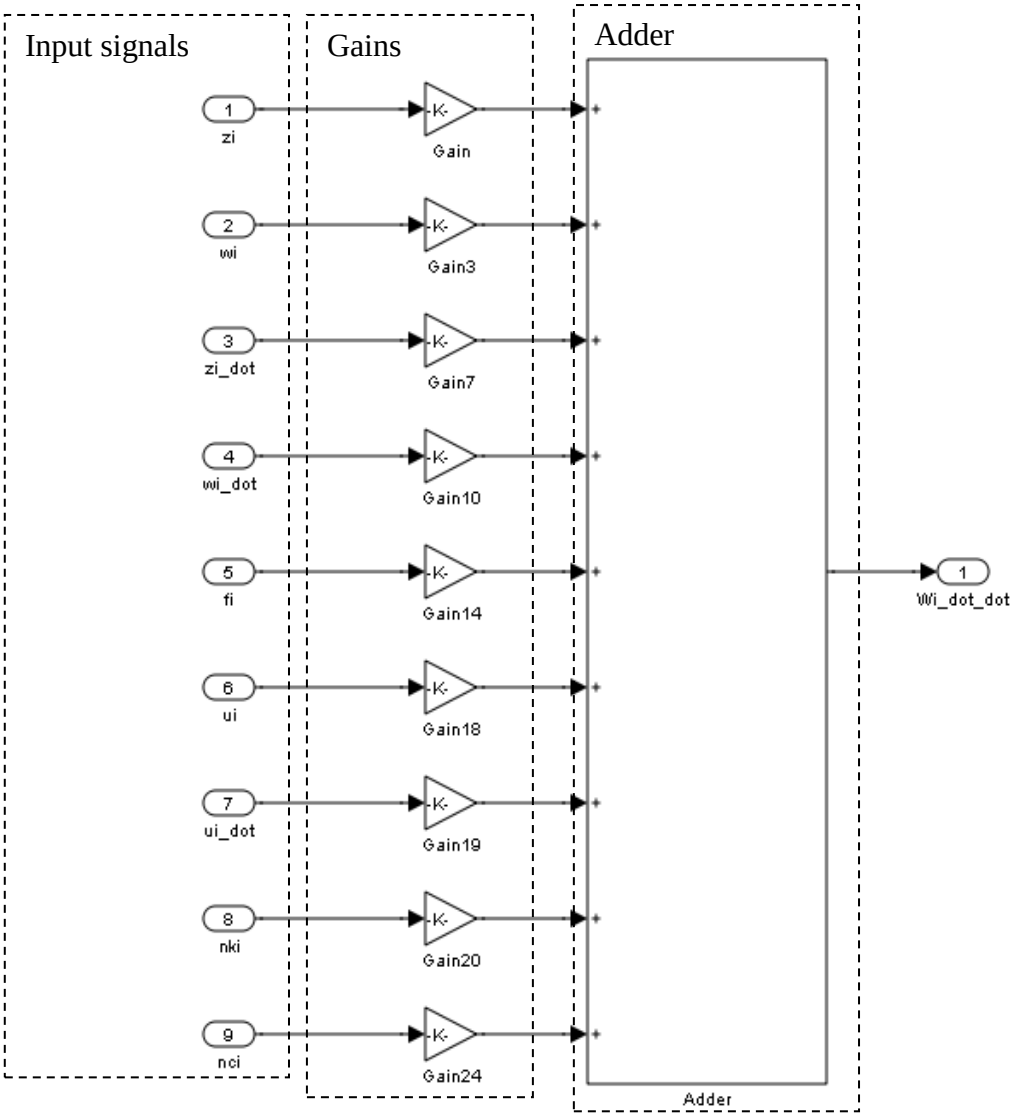


Figure A1.6 MATLAB SIMULINK model of $\ddot{w}_i$

Appendix 2

Appendix 2.1 S-function MATLAB script file for FMRLC

S-function MATLAB script file to train the center of the output membership functions of the fuzzy controller.

```
% start
function [sys,x0,str,ts] =
auto_tunning_FLCsfunction(t,x,u,flag,m,k,ke,kec,ku)

% The following outlines the general structure of an S-function.
%
switch flag,

case 0,
    [sys,x0,str,ts]=mdlInitializeSizes;
case 2,
sys=mdlUpdate(u,m,k,ke,kec,ku);
case 3,
    sys=mdlOutputs(x);
case{1,4,9},
    sys=[];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end

function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 1; % x
sizes.NumOutputs     = 1;  % sys
sizes.NumInputs      = 2;   %  u
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;   % at least one sample time is needed
sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0  =[0];

%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times
%
ts  = [-1  0];
```

```
% end mdlInitializeSizes

function sys=mdlUpdate(u,m,k,ke,kec,ku);

[sys(1)] =auto_tuning_FLC(ke,kec,ku,u(1),u(2),k,m)

function sys=mdlOutputs(x)

sys=x;

function  [fd]=auto_tuning_FLC(ke,kec,ku,e,ec,k,m)
    %find the output Fuzzy controller.
fd=Standard_FLC(ke,kec,ku,e,ec,k,m);

function [u]=Standard_FLC(ke,kec,ku,e,ec,k,m)
         % Standard FLC function
% inputs of this function:
% 1) [ke,kec] are scales of inputs of FLC
% 2) [ku] is output scale of FLC
% 3) e,ec are error and change in error
% 4) Rule_Base is a matrix ,it is a m file in the
%output of this function:
%1) u is output of standard FLC

%find the output of fuzzy inverse model
 p=addition_FLC(ke,kec,ku,e,ec);

     load('bdata.mat','b');
     %The Knowledge-Base Modifier(modify the centers of fuzzy controller
     %outputs membership function for the active set only)

     if -22.5<=e && e<=-15 && -22.5<=ec && ec<=-15
         b(2)=b(2)+m*k*p;
         b(3)=b(3)+m*k*p;

     end
     if -22.5<=e && e<=-15 && -15<=ec && ec<=-7.5
         b(2)=b(2)+m*k*p;
         b(3)=b(3)+m*k*p;

     end
     if -22.5<=e && e<=-15 && -7.5<=ec && ec<=0
         b(2)=b(2)+m*k*p;
         b(3)=b(3)+m*k*p;

     end
     if -22.5<=e && e<=-15 && 0<=ec && ec<=7.5
         b(4)=b(4)+m*k*p;
         b(3)=b(3)+m*k*p;
         b(5)=b(5)+m*k*p;

     end
     if  -22.5<=e && e<=-15 && 7.5<=ec && ec<=15
         b(4)=b(4)+m*k*p;
         b(5)=b(5)+m*k*p;
         b(6)=b(6)+m*k*p;
     end
```

```
if -22.5<=e && e<=-15 && 15<=ec && ec<=22.5
    b(6)=b(6)+m*k*p;
    b(5)=b(5)+m*k*p;
end
 if -15<=e && e<=-7.5 && -22.5<=ec && ec<=-15
    b(2)=b(2)+m*k*p;
    b(1)=b(1)+m*k*p;

end
if -15<=e && e<=-7.5 && -15<=ec && ec<=-7.5
    b(2)=b(2)+m*k*p;



end
if -15<=e && e<=-5 && -7.5<=ec && ec<=0
    b(2)=b(2)+m*k*p;
    b(3)=b(3)+m*k*p;

end
if -15<=e && e<=-7.5 && 0<=ec && ec<=7.5
    b(5)=b(5)+m*k*p;
    b(3)=b(3)+m*k*p;
    b(6)=b(6)+m*k*p;

end
if -15<=e && e<=-7.5 && 7.5<=ec && ec<=15
    b(5)=b(5)+m*k*p;
    b(6)=b(6)+m*k*p;
    b(7)=b(7)+m*k*p;
end
if -15<=e && e<=-7.5 && 15<=ec && ec<=22.5
    b(6)=b(6)+m*k*p;
    b(7)=b(7)+m*k*p;
end
 if -7.5<=e && e<=0 && -22.5<=ec && ec<=-15
    b(2)=b(2)+m*k*p;
    b(1)=b(1)+m*k*p;



end
if -7.5<=e && e<=0 && -15<=ec && ec<=-7.5
    b(2)=b(2)+m*k*p;
    b(1)=b(1)+m*k*p;



end
if -7.5<=e && e<=0 && -7.5<=ec && ec<=0
    b(4)=b(4)+m*k*p;
    b(3)=b(3)+m*k*p;
    b(2)=b(2)+m*k*p;

end
if -7.5<=e && e<=0 && 0<=ec && ec<=7.5
    b(4)=b(4)+m*k*p;
    b(6)=b(6)+m*k*p;
    b(3)=b(3)+m*k*p;
end
if  -7.5<=e && e<=-0 && 7.5<=ec && ec<=15
    b(6)=b(6)+m*k*p;
```

```
    b(7)=b(7)+m*k*p;
end
if -7.5<=e && e<=-0 && 15<=ec && ec<=7.5
    b(7)=b(7)+m*k*p;
end
 if -0<=e && e<=7.5 && -22.5<=ec && ec<=-15
    b(1)=b(1)+m*k*p;


end
if 0<=e && e<=7.5 && -15<=ec && ec<=-7.5
    b(1)=b(1)+m*k*p;
    b(2)=b(2)+m*k*p;

end
if 0<=e && e<=7.5 && -7.5<=ec && ec<=0
    b(4)=b(4)+m*k*p;
    b(5)=b(5)+m*k*p;
    b(2)=b(2)+m*k*p;


end
if 0<=e && e<=7.5 && 0<=ec && ec<=7.5
    b(4)=b(4)+m*k*p;
    b(6)=b(6)+m*k*p;
    b(5)=b(5)+m*k*p;
end
if  0<=e && e<=7.5 && 7.5<=ec && ec<=15
    b(6)=b(6)+m*k*p;
    b(7)=b(7)+m*k*p;
end
if 0<=e && e<=7.5 && 15<=ec && ec<=22.5
    b(6)=b(6)+m*k*p;
    b(7)=b(7)+m*k*p;
end

if 7.5<=e && e<=15 && -22.5<=ec && ec<=-15
    b(1)=b(1)+m*k*p;
    b(2)=b(2)+m*k*p;

end
if  7.5<=e && e<=15 && -15<=ec && ec<=-7.5
    b(1)=b(1)+m*k*p;
    b(2)=b(2)+m*k*p;
    b(3)=b(3)+m*k*p;

end
if  7.5<=e && e<=15 && -7.5<=ec && ec<=0
    b(2)=b(2)+m*k*p;
    b(5)=b(5)+m*k*p;
    b(3)=b(3)+m*k*p;


end
if  7.5<=e && e<=15 && 0<=ec && ec<=7.5
    b(5)=b(5)+m*k*p;
    b(6)=b(6)+m*k*p;

end
```

```
    if  7.5<=e && e<=15 && 7.5<=ec && ec<=15
        b(6)=b(6)+m*k*p;

    end
    if  7.5<=e && e<=15 && 15<=ec && ec<=22.5
        b(6)=b(6)+m*k*p;
        b(7)=b(7)+m*k*p;
    end
    if 15<=e && e<=22.5 && -22.5<=ec && ec<=-15
        b(2)=b(2)+m*k*p;
        b(3)=b(3)+m*k*p;


    end
    if  15<=e && e<=22.5 && -15<=ec && ec<=-7.5
        b(2)=b(2)+m*k*p;
        b(3)=b(3)+m*k*p;
        b(4)=b(4)+m*k*p;

    end
    if  15<=e && e<=22.5 && -7.5<=ec && ec<=0
        b(4)=b(4)+m*k*p;
        b(5)=b(5)+m*k*p;
        b(3)=b(3)+m*k*p;


    end
    if  15<=e && e<=22.5 && 0<=ec && ec<=07.5
        b(5)=b(5)+m*k*p;
        b(6)=b(6)+m*k*p;


    end
    if  15<=e && e<=22.5 && 07.5<=ec && ec<=15
        b(6)=b(6)+m*k*p;
        b(5)=b(5)+m*k*p;
    end
    if 15<=e && e<=22.5 && 15<=ec && ec<=22.5
        b(6)=b(6)+m*k*p;
        b(5)=b(5)+m*k*p;
    end


 save('bdata.mat','b');%the centers of output membership function.
 a=newfis('ST_FLC');%FIS=NEWFIS(FISNAME) creates a new Mamdani-style FIS
structure
 %%%to add input parameter of e into FIS
a=addvar(a,'input','e',[-22.5 22.5]);
%a = addvar(a,varType,varName,varBounds)

%%% fuzzify e to E
a=addmf(a,'input',1,'NB','trimf',[-30 -22.5 -15]);
%a = addmf(a,varType,varIndex,mfName,mfType,mfParams)
a=addmf(a,'input',1,'NM','trimf',[-22.5 -15 -7.5]);
a=addmf(a,'input',1,'NS','trimf',[-15 -7.5 0]);
a=addmf(a,'input',1,'ZE','trimf',[-7.5 0 7.5]);
a=addmf(a,'input',1,'PS','trimf',[0 7.5 15]);
a=addmf(a,'input',1,'PM','trimf',[7.5 15 22.5]);
a=addmf(a,'input',1,'PB','trimf',[15 22.5 30]);
```

```
 %%%to add input parameter of ec into FIS
a=addvar(a,'input','ec',[-22.5 22.5]);
%%% fuzzify ec to EC
a=addmf(a,'input',2,'NB','trimf',[-30 -22.5 -15]);
a=addmf(a,'input',2,'NM','trimf',[-22.5 -15 -5]);
a=addmf(a,'input',2,'NS','trimf',[-15 -7.5 0]);
a=addmf(a,'input',2,'ZE','trimf',[-7.5 0 7.5]);
a=addmf(a,'input',2,'PS','trimf',[0 7.5 15]);
a=addmf(a,'input',2,'PM','trimf',[7.5 15 22.5]);
a=addmf(a,'input',2,'PB','trimf',[15 22.5 30]);
lower=min(b);
upper=max(b);
%%%to add output parameter of u into FIS
a=addvar(a,'output','Fd',[lower-5 upper+5]);
%%% fuzzify u to U
a=addmf(a,'output',1,'NB','trimf',[b(1)-7.5     b(1)    b(1)+7.5]);
a=addmf(a,'output',1,'NM','trimf',[b(2)-7.5     b(2)    b(2)+7.5]);
a=addmf(a,'output',1,'NS','trimf',[b(3)-7.5     b(3)    b(3)+7.5]);
a=addmf(a,'output',1,'ZE','trimf',[b(4)-7.5     b(4)    b(4)+7.5]);
a=addmf(a,'output',1,'PS','trimf',[b(5)-7.5     b(5)    b(5)+7.5]);
a=addmf(a,'output',1,'PM','trimf',[b(6)-7.5     b(6)    b(6)+7.5]);
a=addmf(a,'output',1,'PB','trimf',[b(7)-7.5     b(7)    b(7)+7.5]);

%Part II: Rule-bases
[Rule_Base]=[         1 1 2 1 1
                      1 2 3 1 1
                      1 3 3 1 1
                      1 4 3 1 1
                      1 5 4 1 1
                      1 6 5 1 1
                      1 7 6 1 1
                      2 1 2 1 1
                      2 2 2 1 1
                      2 3 2 1 1
                      2 4 3 1 1
                      2 5 5 1 1
                      2 6 6 1 1
                      2 7 6 1 1
                      3 1 1 1 1
                      3 2 2 1 1
                      3 3 2 1 1
                      3 4 3 1 1
                      3 5 6 1 1
                      3 6 7 1 1
                      3 7 7 1 1
                      4 1 1 1 1
                      4 2 1 1 1
                      4 3 2 1 1
                      4 4 4 1 1
                      4 5 6 1 1
                      4 6 7 1 1
                      4 7 7 1 1
                      5 1 1 1 1
                      5 2 1 1 1
                      5 3 2 1 1
                      5 4 5 1 1
                      5 5 6 1 1
                      5 6 6 1 1
                      5 7 7 1 1
                      6 1 2 1 1
```

```
                    6 2 2 1 1
                    6 3 3 1 1
                    6 4 5 1 1
                    6 5 6 1 1
                    6 6 6 1 1
                    6 7 6 1 1
                    7 1 2 1 1
                    7 2 3 1 1
                    7 3 4 1 1
                    7 4 5 1 1
                    7 5 5 1 1
                    7 6 5 1 1
                    7 7 6 1 1];
    %ruleList=[
    %   1(mf1_in1) 1(mf1_in2) 1(mf1_out1) 1(wieght_in1) 1(weight_in2)
    %   1 2 2 1 1];
    %If the above system a has two inputs and one output, the first rule
can be
    %interpreted as: *If input 1 is MF 1 and input 2 is MF 1, then output 1
is
    %MF 2. with wieght eqyal 1 for each input

   %%% add Rule_base into FIS
    a=addrule(a,Rule_Base);
    %a = addrule(a,ruleList)

%Part III Fuzzify

%%% from e to E , ec to EC
E=ke*e;
EC=kec*ec;
%%%confine E
 if E >4
    E=4;
 elseif E<-4
          E=-4;
 end
 %%% confine EC
  if EC >5
    EC=5;
 elseif EC<-5
          EC=-5;
 end

 %  Part IV Fuzzy Inference
  FLC_input=[E,EC];
  U=evalfis(FLC_input,a);
 % Y = EVALFIS(U,FIS) simulates the Fuzzy Inference System FIS for the
 % input data U and returns the output data Y

 %%%% Part V Defuzzify
  u=ku*U;
 %%%Fuzzy inverse model
 function [p]=addition_FLC(ke,kec,ku,e,ec)
%Part_I :Member-ship Functions
 %%Creates a new Mamdani-style FIS structure
 b=newfis('ADD_FLC');
 %%%to add input parameter of ye into FIS
b=addvar(b,'input','e',[-22.5 22.5]);
%%% fuzzify ye to yE
```

```
b=addmf(b,'input',1,'NB','trimf',[-30 -22.5 -15]);
b=addmf(b,'input',1,'NM','trimf',[-22.5 -15 -5]);
b=addmf(b,'input',1,'NS','trimf',[-15 -7.5 0]);
b=addmf(b,'input',1,'ZE','trimf',[-7.5 0 7.5]);
b=addmf(b,'input',1,'PS','trimf',[0 7.5 15]);
b=addmf(b,'input',1,'PM','trimf',[7.5 15 22.5]);
b=addmf(b,'input',1,'PB','trimf',[15 22.5 30]);


 %%%to add input parameter of yc into FIS
b=addvar(b,'input','ec',[-22.5 22.5]);
%%% fuzzify yc to yC
b=addmf(b,'input',2,'NB','trimf',[-30 -22.5 -15]);
b=addmf(b,'input',2,'NM','trimf',[-22.5 -15 -5]);
b=addmf(b,'input',2,'NS','trimf',[-15 -7.5 0]);
b=addmf(b,'input',2,'ZE','trimf',[-7.5 0 7.5]);
b=addmf(b,'input',2,'PS','trimf',[0 7.5 15]);
b=addmf(b,'input',2,'PM','trimf',[7.5 15 22.5]);
b=addmf(b,'input',2,'PB','trimf',[15 22.5 30]);

%%%to add output parameter of u into FIS
b=addvar(b,'output','p',[-30 30]);
%%% fuzzify p to P
b=addmf(b,'output',1,'NB','trimf',[-30 -22.5 -15]);
b=addmf(b,'output',1,'NM','trimf',[-22.5 -15 -7.5]);
b=addmf(b,'output',1,'NS','trimf',[-15 -7.5 0]);
b=addmf(b,'output',1,'ZE','trimf',[-7.5 0 7.5]);
b=addmf(b,'output',1,'PS','trimf',[0 7.5 15]);
b=addmf(b,'output',1,'PM','trimf',[7.5 15 22.5]);
b=addmf(b,'output',1,'PB','trimf',[15 22.5 30]);

%Part II: Rule-bases
[Rule_Base_2]=[       1 1 2 1 1
                      1 2 3 1 1
                      1 3 3 1 1
                      1 4 3 1 1
                      1 5 4 1 1
                      1 6 5 1 1
                      1 7 6 1 1
                      2 1 2 1 1
                      2 2 2 1 1
                      2 3 2 1 1
                      2 4 3 1 1
                      2 5 5 1 1
                      2 6 6 1 1
                      2 7 6 1 1
                      3 1 1 1 1
                      3 2 2 1 1
                      3 3 2 1 1
                      3 4 3 1 1
                      3 5 6 1 1
                      3 6 7 1 1
                      3 7 7 1 1
                      4 1 1 1 1
                      4 2 1 1 1
                      4 3 2 1 1
                      4 4 4 1 1
                      4 5 6 1 1
                      4 6 7 1 1
                      4 7 7 1 1
```

```
                    5 1 1 1 1
                    5 2 1 1 1
                    5 3 2 1 1
                    5 4 5 1 1
                    5 5 6 1 1
                    5 6 6 1 1
                    5 7 7 1 1
                    6 1 2 1 1
                    6 2 2 1 1
                    6 3 3 1 1
                    6 4 5 1 1
                    6 5 6 1 1
                    6 6 6 1 1
                    6 7 6 1 1
                    7 1 2 1 1
                    7 2 3 1 1
                    7 3 4 1 1
                    7 4 5 1 1
                    7 5 5 1 1
                    7 6 5 1 1
                    7 7 6 1 1 ];
   %%% add Rule_base into FIS
    b=addrule(b,Rule_Base_2);


%Part III Fuzzify

%%% from e to E , ec to EC
E=ke*e;
EC=kec*ec;
%%%confine E
 if E >4
    E=4;
 elseif E<-4
          E=-4;
 end
 %%% confine EC
  if EC >5
    EC=5;
 elseif EC<-5
          EC=-5;
 end

 %  Part IV Fuzzy Inference
  add_FLC_input=[E,EC];
  P=evalfis(add_FLC_input,b);
 % Part V Defuzzify
  p=ku*P;
```

Appendix 2.2 N-Dimension golden section search method

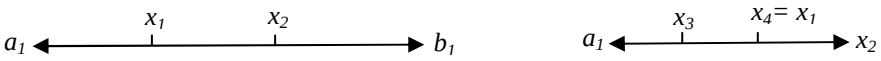
The Golden Section Search (GSS) method is used to optimize functions of N-variables provided upper and lower bounds. This method is very suitable to adjust a set of variables without derivative for the extreme of objective functions. For simplification, the structure of 1-D GSS method is explained first in the following subsection and then in the next subsection the method is expanded to 2-D GSS method.

- **One-dimension GSS method**

Consider an objective function $J(x)$ over the interval $[a_1, b_1]$. It can be assumed that $J(x)$ has only one minimum in $[a_1,b_1]$ at the point x^*. For the first iteration, the golden section points are selected at x_1 and x_2 as shown in Figure A2.1a where x_1 and x_2 can be given as: $x_1 = a_1+(1-r)(b_1-a_1)$ and $x_2=a_1+r\ (b_1-a_1)$, the variable r is defined as golden ratio. There are two cases

1. $J(x_1) < J(x_2)$, It means point x^* must be in the interval $[a_1, x_2]$. The other interval $[x_2,b_1]$ can be eliminated .
2. $J(x_1) \geq J(x_2)$, It means point x^* must be in the interval $[x_1, b_1]$. The other interval $[a_1,x_1]$ can be eliminated .

Without losing of generality, we focus on the case $J(x_1) < J(x_2)$. For the second iteration, the new golden section points are selected at x_3 and x_4 as shown in Figure A2.1b where $x_3 = a_1+(1-r)r(b_1-a_1)$ and $x_4=a_1+r^2\ (b_1-a_1)$. Note that $x_4=x_1$, where $r^2= (1-r)$, therefore, $J(x_1) = J(x_4)$. Just the value of $J(x_3)$ is evaluated at this iteration. One of the subintervals $[x_4, x_2]$ or $[a_1, x_3]$ is eliminated depending on the result of the comparison of two objective function values ($J(x_3)$ and $J(x_4)$) evaluated at the two inner points (x_3 and x_4). This pattern is continued until the point x^* (at which the objective function J is minimum) is obtained.



(a) (b)

Figure 6 One-dimension GSS

- **Two-dimension GSS method**

Similar to the 1-D GSS method, assume that the objective function is denoted as $J(x,y)$, the interval of uncertainty are $[a_1, b_1]$ and $[a_2,b_2]$, respectively and the minimum objective function happens at (x^*,y^*). For the first iteration, the golden section points are selected at x_1 and x_2 on interval $[a_1, b_1]$ and at y_1 and y_2 on interval $[a_2,b_2]$ where $x_1= a_1+(1-r)(b_1-a_1)$, $x_2=a_1+r\,(b_1-a_1)$, $y_1= a_2+(1-r)(b_2-a_2)$ and $y_2=a_2+r\,(b_2-a_2)$. There are four cases

1. $J(x_1,y_1)$ smaller than $J(x_1,y_2)$, $J(x_2,y_1)$ and $J(x_2,y_2)$: it means point (x^*,y^*) is in the region intersected by the two section $x\in[a_1,x_2]$ and $y\in[a_2,y_2]$,so that the other intervals can be eliminated.
2. $J(x_1,y_2)$ smaller than $J(x_1,y_1)$, $J(x_2,y_1)$ and $J(x_2,y_2)$: it means point (x^*,y^*) is in the region intersected by the two section $x\in[a_1,x_2]$ and $y\in[y_1,b_2]$, so that the other intervals can be eliminated.
3. $J(x_2,y_1)$ smaller than $J(x_1,y_1)$, $J(x_1,y_2)$ and $J(x_2,y_2)$: it means point (x^*,y^*) is in the region intersected by the two section $x\in[x_1,b_1]$ and $y\in[a_2,y_2]$, so that the other intervals can be eliminated.
4. $J(x_2,y_2)$ smaller than $J(x_1,y_1)$, $J(x_1,y_2)$ and $J(x_2,y_1)$: it means point (x^*,y^*) is in the region intersected by the two section $x\in[x_1,b_1]$ and $y\in[y_1,b_2]$, so that the other intervals can be eliminated.

We focus on the first case. At the second iteration, the new golden section points are selected at x_3 and x_4 on interval $[a_1,x_2]$ and y_3 and y_4 on the interval $[a_2,y_2]$ where $x_3= a_1+(1-r)r(b_1-a_1)$, $x_4=a_1+r^2\,(b_1-a_1)$, $y_3= a_2+(1-r)r(b_2-a_2)$, $y_4=a_2+r^2\,(b_2-a_2)$. Note that $x_4=x_1$ and $y_4=y_1$. Therefore, $J(x_1,y_1) = J(x_4,y_4)$. Only three evaluations of $J(x_3,y_3)$, $J(x_3,y_4)$ and $J(x_4,y_3)$ are made

in this iteration. Another subintervals are eliminated depending on the result of comparison of these four objective function values evaluated at the four inner points. This pattern is continued until the point (x^*,y^*), at which the objective function J is minimum, is obtained.

- **Three-dimension GSS method**

According to the same logic as in the 2-Ds, it is easy to implement a 3-D GSS. Suppose that the objective function is denoted as $J(x,y,z)$, the intervals of uncertainty are $[a_1,b_1]$, $[a_2,b_2]$ and $[a_3,b_3]$, respectively and the unique minimum objective function happens at (x^*,y^*,z^*). For first iteration, the golden section points are chosen as *1-r* and *r* relative to intervals of *x*, *y*, *z*, i.e. the first two points on *x* are $x_1 = a_1 + (1-r)(b_1-a_1)$ and $x_2 = a_1 + r(b_1-a_1)$; the first two points on *y* are $y_1 = a_2 + (1-r)(b_2-a_2)$ and $y_2 = a_2 + r(b_2-a_2)$; the first two points on *z* are $z_1 = a_3 + (1-r)(b_3-a_3)$ and $z_2 = a_3 + r(b_3-a_3)$. Suppose for example that $J(x_1,y_1,z_1)$ is smaller than $J(x_1,y_1,z_2)$, $J(x_1,y_2,z_1)$, $J(x_1,y_2,z_2)$, $J(x_2,y_1,z_1)$, $J(x_2,y_2,z_1)$, $J(x_2,y_1,z_2)$, and $J(x_2,y_2,z_2)$, therefore, the point (x^*,y^*,z^*) intuitively is in the region intersected by the three sections $x\in[a_1,x_2]$, $y\in[a_2,y_2]$ and $z\in[a_3,z_2]$.

At the second iteration, the new golden section points are selected at x_3 and x_4 on the interval $[a_1,x_2]$; y_3 and y_4 on the interval $[a_2,y_2]$ and z_3 and z_4 on the interval $[a_3,z_2]$, where $x_3 = a_1+(1-r)r(b_1-a_1)$, $x_4=a_1+r^2(b_1-a_1)$, $y_3= a_2+(1-r)r(b_2-a_2)$, $y_4=a_2+r^2(b_2-a_2)$, $z_3= a_3+(1-r)r(b_3-a_3)$ and $z_4=a_3+r^2(b_3-a_3)$. Note that $x_4=x_1$, $y_4=y_1$ and $z_4=z_1$.

Therefore, $J(x_1,y_1,z_1) = J(x_4,y_4,z_2)$. Only seven evaluations of $J(x_3,y_3,z_3)$, $J(x_3,y_3,z_4)$, $J(x_3,y_4,z_3)$, $J(x_3,y_4,z_4)$, $J(x_4,y_3,z_3)$, $J(x_4,y_4,z_3)$ and $J(x_4,y_3,z_4)$are made in this iteration. Another subintervals are eliminated depending on the result of comparison of these eight objective function values evaluated at the eight inner points. This pattern is continued until the point (x^*,y^*,z^*), at which the objective function *J* is minimum, is obtained.

Appendix 3 M-file programs to train neural controllers

- M-file program to train neural controller at front-right suspension

```
clear all;
close all;
clc;
time=1:1:10001;
load('error1data.mat','error1');
load('errordot1data.mat','errordot1');
load('errorI1data.mat','errorI1');
load('force1data.mat','force1');
timet=time';
error1t=error1';
errordot1t=errordot1';
errorI1t=errorI1';
force1t=force1';
P=[error1t;errordot1t;errorI1t];
T=force1t;
S1=20;
[S2,Q]=size(T);
mmnet=newff(minmax(P),[S1 S2],{'tansig' 'purelin'},'trainlm');
mmnet.dividefcn='';
[trainP,valP,testP,trainInd,valInd,testInd] = dividerand(P);
[trainT,valT,testT] = divideind(T,trainInd,valInd,testInd);
mmnet.trainparam.mu_max=1e15;
mmnet.trainParam.show = 5;
mmnet.trainParam.epochs = 3000;
mmnet.trainParam.goal = 1e-5;
mmnet=train(mmnet,P,T);
a = sim(mmnet,P);
Iput_Weight_a=mmnet.IW{1,1};
Iput_biase_a=mmnet.b{1};
Layer_weight_a=mmnet.LW{2,1};
Layer_biase_a=mmnet.b{2};
figure(1);
plot(timet,force1,timet,a);
```

- M-file program to train neural controller at front-left suspension

```
clear all;
close all;
clc;
time=1:1:10001;
load('error2data.mat','error2');
load('errordot2data.mat','errordot2');
load('errorI2data.mat','errorI2');
load('force2data.mat','force2');
timet=time';
error2t=error2';
errordot2t=errordot2';
errorI2t=errorI2';
force2t=force2';
P=[error2t;errordot2t;errorI2t];
T=force2t;
S1=15;
[S2,Q]=size(T);
mmnet=newff(minmax(P),[S1 S2],{'tansig' 'purelin'},'trainlm');
mmnet.dividefcn='';
[trainP,valP,testP,trainInd,valInd,testInd] = dividerand(P);
[trainT,valT,testT] = divideind(T,trainInd,valInd,testInd);
mmnet.trainparam.mu_max=1e15;
mmnet.trainParam.show = 5;
mmnet.trainParam.epochs = 10000;
mmnet.trainParam.goal = 1e-5;
mmnet=train(mmnet,P,T);
a = sim(mmnet,P);
Iput_Weight_a=mmnet.IW{1,1};
Iput_biase_a=mmnet.b{1};
Layer_weight_a=mmnet.LW{2,1};
Layer_biase_a=mmnet.b{2};
figure(1);
plot(timet,force2,timet,a);
```

- M-file program to train neural controller at rear-right suspension

```
clear all;
close all;
clc;
time=1:1:10001;
load('error3data.mat','error3');
load('errordot3data.mat','errordot3');
load('errorI3data.mat','errorI3');
load('force3data.mat','force3');
timet=time';
error3t=error3';
errordot3t=errordot3';
errorI3t=errorI3';
force3t=force3';
P=[error3t;errordot3t;errorI3t];
T=force3t;
S1=15;
[S2,Q]=size(T);
mmnet=newff(minmax(P),[S1 S2],{'tansig' 'purelin'},'trainlm');
mmnet.dividefcn='dividerand';
[trainP,valP,testP,trainInd,valInd,testInd] = dividerand(P);
[trainT,valT,testT] = divideind(T,trainInd,valInd,testInd);
mmnet.trainparam.mu_max=1e15;
mmnet.trainParam.show = 5;
mmnet.trainParam.epochs = 3000;
mmnet.trainParam.goal = 1e-5;
mmnet=train(mmnet,P,T);
a = sim(mmnet,P);
Iput_Weight_a=mmnet.IW{1,1};
Iput_biase_a=mmnet.b{1};
Layer_weight_a=mmnet.LW{2,1};
Layer_biase_a=mmnet.b{2};
figure(1);
plot(timet,force3,timet,a);
```

- M-file program to train neural controller at rear-right suspension

```
clear all;
close all;
clc;
time=1:1:10001;
load('error4data.mat','error4');
load('errordot4data.mat','errordot4');
load('errorI4data.mat','errorI4');
load('force4data.mat','force4');
timet=time';
error4t=error4';
errordot4t=errordot4';
errorI4t=errorI4';
force4t=force4';
P=[error4t;errordot4t;errorI4t];
T=force4t;
S1=20;
[S2,Q]=size(T);
mmnet=newff(minmax(P),[S1 S2],{'tansig' 'purelin'},'trainlm');
mmnet.dividefcn='dividerand';
[trainP,valP,testP,trainInd,valInd,testInd] = dividerand(P);
[trainT,valT,testT] = divideind(T,trainInd,valInd,testInd);
mmnet.trainparam.mu_max=1e15;
mmnet.trainParam.show =5;
mmnet.trainParam.epochs = 10000;
mmnet.trainParam.goal = 1e-5;
mmnet=train(mmnet,P,T);
a = sim(mmnet,P);
Iput_Weight_a=mmnet.IW{1,1};
Iput_biase_a=mmnet.b{1};
Layer_weight_a=mmnet.LW{2,1};
Layer_biase_a=mmnet.b{2};
figure(1);
plot(timet,force4,timet,a);
```

Appendix 4 Constrains File for the neurofuzzy controller

- Constrains File for Right-Front Neurofuzzy Controller

```
NET "e1<0>" LOC="C15";
NET "e1<1>" LOC="C3";
NET "e1<2>" LOC="B15";
NET "e1<3>" LOC="A15";
NET "e1<4>" LOC="C12";
NET "e1<5>" LOC="D12";
NET "e1<6>" LOC="G9";
NET "e1<7>" LOC="A8";
NET "ce1<0>" LOC="A11";
NET "ce1<1>" LOC="B11";
NET "ce1<2>" LOC="C4";
NET "ce1<3>" LOC="E13";
NET "ce1<4>" LOC="B16";
NET "ce1<5>" LOC="A16";
NET "ce1<6>" LOC="D14";
NET "ce1<7>" LOC="C14";
NET "ce1<8>" LOC="B14";
NET "ce1<9>" LOC="A14";
NET "ce1<10>" LOC="B13";
NET "ce1<11>" LOC="A13";
NET "ce1<12>" LOC="F12";
NET "contout1<0>" LOC="E12";
NET "contout1<1>" LOC="E11";
NET "contout1<2>" LOC="F11";
NET "contout1<3>" LOC="C11";
NET "contout1<4>" LOC="D11";
NET "contout1<5>" LOC="E9";
NET "contout1<6>" LOC="F9";
NET "contout1<7>" LOC="E8";
NET "contout1<8>" LOC="F8";
NET "contout1<9>" LOC="C7";
NET "contout1<10>" LOC="D7";
NET "contout1<11>" LOC="F7";
NET "contout1<12>" LOC="E7";
NET "contout1<13>" LOC="B6";
NET "contout1<14>" LOC="A6";
NET "contout1<15>" LOC="C5";
NET "contout1<16>" LOC="D5";
NET "contout1<17>" LOC="A4";
NET "contout1<18>" LOC="B4";
```

- Constrains File for Left-Front Neurofuzzy Controller

```
NET "e2<0>" LOC="C15";
NET "e2<1>" LOC="C3";
NET "e2<2>" LOC="B15";
NET "e2<3>" LOC="A15";
NET "e2<4>" LOC="C12";
NET "e2<5>" LOC="D12";
NET "e2<6>" LOC="G9";
NET "e2<7>" LOC="A8";
NET "ce2<0>" LOC="A11";
NET "ce2<1>" LOC="B11";
NET "ce2<2>" LOC="C4";
NET "ce2<3>" LOC="E13";
NET "ce2<4>" LOC="B16";
NET "ce2<5>" LOC="A16";
NET "ce2<6>" LOC="D14";
NET "ce2<7>" LOC="C14";
NET "ce2<8>" LOC="B14";
NET "ce2<9>" LOC="A14";
NET "ce2<10>" LOC="B13";
NET "ce2<11>" LOC="A13";
NET "ce2<12>" LOC="F12";
NET "contout2<0>" LOC="E12";
NET "contout2<1>" LOC="E11";
NET "contout2<2>" LOC="F11";
NET "contout2<3>" LOC="C11";
NET "contout2<4>" LOC="D11";
NET "contout2<5>" LOC="E9";
NET "contout2<6>" LOC="F9";
NET "contout2<7>" LOC="E8";
NET "contout2<8>" LOC="F8";
NET "contout2<9>" LOC="C7";
NET "contout2<10>" LOC="D7";
NET "contout2<11>" LOC="F7";
NET "contout2<12>" LOC="E7";
NET "contout2<13>" LOC="B6";
NET "contout2<14>" LOC="A6";
NET "contout2<15>" LOC="C5";
NET "contout2<16>" LOC="D5";
NET "contout2<17>" LOC="A4";
NET "contout2<18>" LOC="B4";
```

- Constrains File for Right-Rear Neurofuzzy Controller

```
NET "e3<0>" LOC="C15";
NET "e3<1>" LOC="C3";
NET "e3<2>" LOC="B15";
NET "e3<3>" LOC="A15";
NET "e3<4>" LOC="C12";
NET "e3<5>" LOC="D12";
NET "e3<6>" LOC="G9";
NET "e3<7>" LOC="A8";
NET "ce3<0>" LOC="A11";
NET "ce3<1>" LOC="B11";
NET "ce3<2>" LOC="C4";
NET "ce3<3>" LOC="E13";
NET "ce3<4>" LOC="B16";
NET "ce3<5>" LOC="A16";
NET "ce3<6>" LOC="D14";
NET "ce3<7>" LOC="C14";
NET "ce3<8>" LOC="B14";
NET "ce3<9>" LOC="A14";
NET "ce3<10>" LOC="B13";
NET "ce3<11>" LOC="A13";
NET "ce3<12>" LOC="F12";
NET "contout3<0>" LOC="E12";
NET "contout3<1>" LOC="E11";
NET "contout3<2>" LOC="F11";
NET "contout3<3>" LOC="C11";
NET "contout3<4>" LOC="D11";
NET "contout3<5>" LOC="E9";
NET "contout3<6>" LOC="F9";
NET "contout3<7>" LOC="E8";
NET "contout3<8>" LOC="F8";
NET "contout3<9>" LOC="C7";
NET "contout3<10>" LOC="D7";
NET "contout3<11>" LOC="F7";
NET "contout3<12>" LOC="E7";
NET "contout3<13>" LOC="B6";
NET "contout3<14>" LOC="A6";
NET "contout3<15>" LOC="C5";
NET "contout3<16>" LOC="D5";
NET "contout3<17>" LOC="A4";
NET "contout3<18>" LOC="B4";
```

- Constrains File for Left-Rear Neurofuzzy Controller

```
NET "e4<0>" LOC="C15";
NET "e4<1>" LOC="C3";
NET "e4<2>" LOC="B15";
NET "e4<3>" LOC="A15";
NET "e4<4>" LOC="C12";
NET "e4<5>" LOC="D12";
NET "e4<6>" LOC="G9";
NET "e4<7>" LOC="A8";
NET "ce4<0>" LOC="A11";
NET "ce4<1>" LOC="B11";
NET "ce4<2>" LOC="C4";
NET "ce4<3>" LOC="E13";
NET "ce4<4>" LOC="B16";
NET "ce4<5>" LOC="A16";
NET "ce4<6>" LOC="D14";
NET "ce4<7>" LOC="C14";
NET "ce4<8>" LOC="B14";
NET "ce4<9>" LOC="A14";
NET "ce4<10>" LOC="B13";
NET "ce4<11>" LOC="A13";
NET "ce4<12>" LOC="F12";
NET "contout4<0>" LOC="E12";
NET "contout4<1>" LOC="E11";
NET "contout4<2>" LOC="F11";
NET "contout4<3>" LOC="C11";
NET "contout4<4>" LOC="D11";
NET "contout4<5>" LOC="E9";
NET "contout4<6>" LOC="F9";
NET "contout4<7>" LOC="E8";
NET "contout4<8>" LOC="F8";
NET "contout4<9>" LOC="C7";
NET "contout4<10>" LOC="D7";
NET "contout4<11>" LOC="F7";
NET "contout4<12>" LOC="E7";
NET "contout4<13>" LOC="B6";
NET "contout4<14>" LOC="A6";
NET "contout4<15>" LOC="C5";
NET "contout4<16>" LOC="D5";
NET "contout4<17>" LOC="A4";
NET "contout4<18>" LOC="B4";
```

List of published papers:

1. Ammar A. Aldair and Weiji J. Wang. "*Evolutionary Algorithm Based Fractional Order $PI^{\lambda}D^{\mu}$ Controller Design for a Full Vehicle Nonlinear Active Suspension System*", ***10th International Symposium on Advanced Vehicle Control AVEC'10***, August 22-26 2010 Loughborough, UK, pp. 18-25. http://www.lboro.ac.uk/departments/tt/avec10/programs.html.

2. Ammar A. Aldair and Weiji J. Wang. "*Adaptive Neuro Fuzzy Inference Controller for Full Vehicle Nonlinear Active Suspension Systems*", ***1st International Conference on Energy, Power and Control,*** **the University of Basrah, Basrah, Iraq, November 30 to December 02, 2010, pp. 97- 106**. (***Note: Conference proceeding has been published by the IEEE organization at the IEEE Explore digital library***). http://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Adaptive+Neuro+Fuzzy+Inference+Controller+for+Full+Vehicle+Nonlinear+Active+Suspension+Systems&x=17&y=16.

3. Ammar A. Aldair and Weiji J. Wang." *FPGA Based Adaptive Neuro Fuzzy Inference Controller for Full Vehicle Nonlinear Active Suspension Systems*", ***International Journal of Artificial Intelligence & Applications (IJAIA),*** Vol. 1, No. 4, October 2010, pp. 1-15. http://airccse.org/journal/ijaia/currentissue.html.

4. Ammar A. Aldair and Weiji J. Wang." *Design of Fractional Order Controller Based on Evolutionary Algorithm for a Full Vehicle Nonlinear Active Suspension Systems*"

International Journal for Control and Automation (IJCA), Vol. 3, No. 4, December, 2010, pp.33-46.
http://www.sersc.org/journals/IJCA/vol3_no4.php.

5. Ammar A. Aldair and Weiji J. Wang." *The Energy Regeneration of Electromagnetic Active Suspension in Full Vehicle with Neurofuzzy Controller*", ***International Journal of Artificial Intelligence & Applications (IJAIA)***, Vol.2, No.2, April 2011, pp. 32-43. http://airccse.org/journal/ijaia/current2011.html.

6. Ammar A. Aldair and Weiji J. Wang." *Design an Intelligent Controller for Full Vehicle Nonlinear Active Suspension Systems"* ***International Journal on Smart Sensing and Intelligent Systems***, Vol. 4, No. 2, June 2011, pp. 224-243.
http://www.s2is.org/Issues/v4/n2/papers/paper5.pdf.

7. Ammar A. Aldair and Weiji J. Wang." *Fuzzy model reference learning controller based full vehicle nonlinear active suspension system*" ***International Journal of Research and Reviews in Computer Science (IJRRCS)***, for Vol. 2, No. 3, 2011, June 2011 issue, pp.864-873.
http://scholarlyexchange.org/ojs/index.php/IJRRCS/article/viewFile/8584/6112.

8. Ammar A. Aldair and Weiji J. Wang." *Neural Controller Based Full Vehicle Nonlinear Active Suspension Systems with Hydraulic Actuators*" ***International Journal for Control and Automation (IJCA),*** for Vol. 4, No. 2, 2011 (June 2011 issue).
http://www.sersc.org/journals/IJCA/vol4_no2/7.pdf

9. Ammar A. Aldair and Weiji J. Wang. "*A Neurofuzzy Controller based Full Vehicle Nonlinear Active Suspension System*" ***Journal of Vibration and Control***, Manuscript accepted for online first.

http://jvc.sagepub.com/content/early/recent