Analysis and Development of Control Methodologies for Semi-active Suspensions

Omid Ghasemalizadeh

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

Saied Taheri, Chair

Mehdi Ahmadian

Corina Sandu

Michael Roan

Amandeep Singh

August 30, 2016

Blacksburg, Virginia

Keywords: Vehicle Dynamics, Semi-active suspension, Control Methodology, Simulations, Fuzzy Logic, Neural Netwo

Analysis and Development of Control Methodologies for Semi-active Suspensions

Omid Ghasemalizadeh

ABSTRACT

Semi-active suspensions have drawn particular attention due to their superior performance over the other types of suspensions. One of their advantages is that their damping coefficient can be controlled without the need for any external source of power. In this study, a handful of control approaches are implemented on a car models using MATLAB/Simulink. The investigated control methodologies are skyhook, groundhook, hybrid skyhook-groundhook, Acceleration Driven Damper, Power Driven Damper, H_∞ Robust Control, Fuzzy Logic Controller, and Inverse ANFIS. $H\infty$ Robust Control is an advanced method that guarantees transient performance and rejects external disturbances. It is shown that H_∞ with the proposed modification, has the best performance although its relatively high cost of computation could be potentially considered as a drawback. Also, the proposed Inverse ANFIS controller uses the power of fuzzy systems along with neural networks to help improve vehicle ride metrics significantly.

In this study, a novel approach is introduced to analyze and fine-tune semi-active suspension control algorithms. In some cases, such as military trucks moving on off-road terrains, it is critical to keep the vehicle ride quality in an acceptable range. Semi-active suspensions are used to have more control over the ride metrics compared to passive suspensions and also, be more cost-effective compared to active suspensions. The proposed methodology will investigate the skyhook-groundhook hybrid controller. This is accomplished by conducting sensitivity analysis of the controller performance to varying vehicle/road parameters. This approach utilizes sensitivity analysis and one-at-a-time methodology to find and reach the optimum point of vehicle suspensions. Furthermore, real-time tuning of the mentioned controller will be studied. The online tuning will help keep the ride quality of the vehicle close to its optimum point while the vehicle parameters are changing. A quarter-car model is used for all simulations and analyses.

Analysis and Development of Control Methodologies for Semi-active Suspensions

Omid Ghasemalizadeh

GENERAL AUDIENCE ABSTRACT

Passenger safety and comfort have always been two major concerns in designing and engineering vehicles. Suspensions play a vital role in this regard. They are there to ensure a very smooth and comfortable ride experience. Many technologies have been developed to increase performance of suspension and customize their functionality. However, only a few developments led to a new family of suspensions and opened a broad field in automotive engineering for researchers to do their twist and tweaks. One fascinating technology that was developed a few decades ago, was semi-active suspensions. Their advantage over conventional ones is that its stiffness can be adjusted on the fly. This property can be combined with a control methodology in order to improve the ride experience further more compared to conventional suspensions.

In this dissertation, some novel control methodologies are developed and compared with existing ones. The results are discussed exclusively for each controller.

To my dear family, my father, my mother, my sister, and my brother,

MOHAMMAD

FERESHTEH

FARZANEH

FARID

Table of Contents

Table of Figures

Table of Tables

1 Introduction

Suspensions have significant impact on vehicle dynamics and consequently, on its ride quality. Conventional suspensions are designed based on parameters of a vehicle as well as the application that the vehicle will be used for. A drawback with the conventional suspensions is that they are designed for a limited range of road/ride characteristics. Other technologies have been developed in last few decades for the purpose of improving ride quality such as Active Suspensions and Semi-active Suspensions. In active suspensions, an actuator is used along with a low stiffness conventional suspension to apply required amount of force to the sprung mass in order to enhance ride experience of the passengers. However, the equipment is expensive, heavy compared to other types of suspensions, and difficult to maintain. On the other hand, semi-active suspension are proven to work nearly as good as active suspensions but they do not cost as much. The hardware and control algorithms for such suspensions are being developed and fine-tuned on an ongoing basis in the industry which leads to them being used in vehicles even more than before. Margolis modified two active damper control strategies for a motorcycle-like vehicle to be semi-active and concluded that similar performance to active systems is feasible[1]. Margolis, in another work, showed that semi-active suspensions perform nearly as good as active systems even for a system with two degrees of freedom[2].

1.1 Skyhook and Groundhook

In skyhook theory, the sprung mass is connected to the world reference system with an imaginary damper to ensure that the human comfort index is within the acceptable range. It similar for groundhook but, this time the wheel is connected to the world reference frame with an imaginary damper.

In 1974, Karnopp used semi-active damper along with skyhook damper concept to isolate vibrations of a sprung mass from ground motions via computer simulation[3]. He showed that performance of the semi-proposed active damper is comparable to that of fully active vibration control systems. Choi field-tested four ER semi-active suspensions using a passenger car and implemented skyhook theory to show that ride comfort and steering stability are improved[4]. Hong used a human-in-the-loop tuning control algorithm to develop a modified skyhook-groundhook control scheme[5]. Caponetto developed a soft computing to control semi-active skyhook suspension[6]. They used Genetic Algorithm to optimize a fuzzy controller to change the characteristic parameters of the skyhook suspension. Ahmadian implemented two variation of skyhook theory, skyhook function no-jerk skyhook[7]. They were able to reduce the jerk that usually is present in conventional skyhook controller. Later on, Emura developed a simple semi-active suspension according to the skyhook damper concept[8]. He used a single-point tire model along with a quarter-car model in his study. Later, Ahmadian analyzed performance three well-known controller skyhook, groundhook, and hybrid on a quarter-car rig experimentally [9]

Groundhook was another semi-active suspension control law that was developed mainly to reduce vehicle damages due to interaction with road[10]. Same authors published another work that optimized parameters of groundhook control policy for a nonlinear model[11]. They took many aspects of damage into account for their optimization study.

1.2 Other methodologies

Bellizzi studied adaptive control methodology for semi-active suspensions with dry-friction and stochastic excitation[12]. Being easy to implement was one the main goals in designing and developing the class of sub-optimal feedback laws. In 1989, Cheok developed neuro controller that used Lyapunov Control Theory and Model Reference Adaptive Control to improve performance of semi-active suspensions[13]. This neural model did not require a complete knowledge of the system dynamics. In 1994, Venhovens introduced a skyhook adaptive controller along with two methods to either measure or estimate the states of the vehicle[14]. Later, Gordon developed an optimal controller for semi-active suspensions[15]. He used quarter-car model to implement his controller. Valasek proposed a skyhook-groundhook concept for enhancing truck ride comfort and road holding metrics[16]. Rao proposed a novel tunable fuzzy logic controller to improve human comfort of vehicles equipped with active suspensions[17]. In 1999, Yi developed an observer-based controller for semi-active suspensions[18]. They proposed a methodology that simplified design process of observer. Same authors in the same year published another work went through details of designing an observer for semi-active suspensions[19]. The proposed observer uses acceleration signal and estimates suspension deflection and velocity with guaranteed convergence.

In 2000, Jansen published a work that he compared control algorithms of MR damper in application of structure health and safety[20].

In 2005, Savaresi introduced Acceleration-Driven-Damper (ADD) that aims at reducing the sprung mass acceleration[21]. A significant advantage of his control method was its simplicity. Another simple control policy was proposed that reduces the sprung mass acceleration by controlling the power stored or dissipated in the system[22]. Also, Chen was able to combine the concept of skyhook damper along with sliding mode controller to introduce a new semi-active control policy that focused on passenger's ride comfort[23]. Later in 2014, Ghasemalizadeh proposed a new semi-active suspension algorithm by modifying H_∞ Robust Control theory to suit the nature of limited force in semi-active suspensions[24].

In this document, first, different car models with various degrees of freedom will be introduced and the dynamics equations will be written. In chapter 3, Simple yet widely used controllers will be discussed along with their control laws and results on different car model simulations. In the next chapter, a novel method will be proposed to analyze and fine-tune semi-active suspension control methods. Chapter 4 will be dedicated to a novel modified version of H_∞ Robust Control and its implementation results on quarter-car and a 6-axle full-car models. In chapter 6, Fuzzy Logic Controllers family will be discussed as a different method I nature to design controllers for semi-active suspension. Later in that chapter, Inverse ANFIS controller will be proposed as a novel and flexible controller that significantly improves ride metrics compared to other conventional controllers.

2 Dynamic Modeling

Different car model from a point contact quarter-car model to an n-axle model were developed. The models were developed in MATLAB. It should be mentioned that point contact tire model was used in developing of the car models. In contact point tire model, a mass (called unsprung mass) and a spring represent the tire and the free end of the spring is assumed to move vertically according to given road-profile.

2.1 Quarter-car

Lin, Padovan, and Lu [25] recommend a procedure of starting from a quarter vehicle model for a single wheel heave-only suspension and then expanding and studying full vehicle suspension systems. A quarter-car model consists of two masses, two springs, and a semi-active damper. Figure below shows the described model.

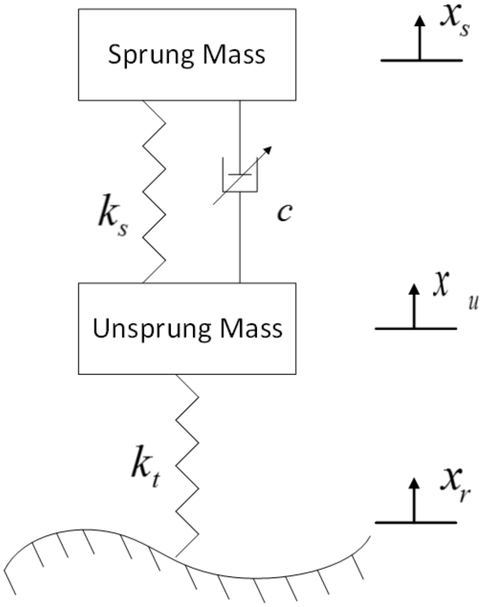


Figure 2-1 – Point contact quarter-car model

As shown in Figure 1, the semi-active suspension damper and spring are represented as ks and c, respectively. Moreover, k_t is known as the tire stiffness. One can write the governing equations of the quarter-car model as following:

$$m_s\ddot{x_s} + c(\dot{x}_s - \dot{x}_u) + k_s(x_s - x_u) = 0 \qquad \text{Eq. 1}$$

$$m_u\ddot{x_u} - c(\dot{x}_s - \dot{x}_u) - k_s(x_s - x_u) + k_t(x_u - x_z) = 0 \qquad \text{Eq. 2}$$

A quarter-car model has been developed in MATLAB and Table 1 explains the parameters used in the equations.

Table 1 – Quarter-car terms definition

x_s	Sprung Mass Displacement	x_u	Unsprung Mass Displacement
m_s	Sprung Mass	m_u	Unsprung Mass
c	Suspensions Damping	k	Suspension Stifness
k_t	Tire Stiffness	x_z	Road Profile

2.2 Half-car

A half-car model has two corners that each corner contains a suspension and a tire. This model has four degrees of freedom (DOF) in total; two degrees of freedom are associated with the vertical motion of the unsprung masses, pitch angle of the sprung mass adds another degree of freedom to the systems, and vertical motion of the sprung mass is the fourth DOF. Figure 2 shows the described half-car model. Four separate equations can be written describing dynamics of each degree of freedom.

A half-car model with two axles has been developed in MATLAB.

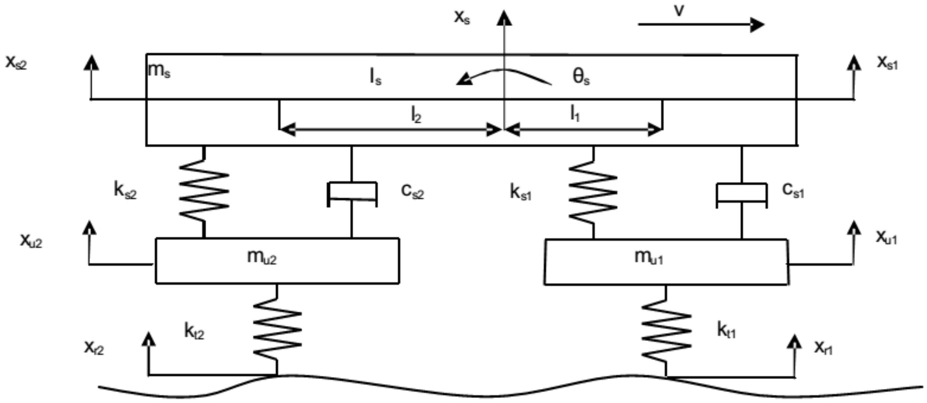


Figure 2-2 - Half-car model with two corners

According to Figure above, the governing equations of the half-car model can be formed as below:

$$m_s\ddot{x_s} + c_{s1}(\dot{x}_{s1} - \dot{x}_{u1}) + c_{s2}(\dot{x}_{s2} - \dot{x}_{u2}) + k_{s1}(x_{s1} - x_{u1}) + k_{s2}(x_{s2} - x_{u2}) = 0 \qquad \text{Eq. 3}$$

$$I_s\ddot{\theta}_s + l_1(c_{s1}(\dot{x}_{s1} - \dot{x}_{u1}) + k_{s1}(x_{s1} - x_{u1})) - l_2(c_{s2}(\dot{x}_{s2} - \dot{x}_{u2}) + k_{s2}(x_{s2} - x_{u2})) = 0 \qquad \text{Eq. 4}$$

$$m_{u1}\ddot{x_{s1}} - c_{s1}(\dot{x}_{s1} - \dot{x}_{u1}) - k_{s1}(x_{s1} - x_{u1}) + k_{t1}(x_{u1} - x_{z1}) = 0 \qquad \text{Eq. 5}$$

$$m_{u2}\ddot{x_{s2}} - c_{s2}(\dot{x}_{s2} - \dot{x}_{u2}) - k_{s2}(x_{s2} - x_{u2}) + k_{t1}(x_{u2} - x_{z2}) = 0. \qquad \text{Eq. 6}$$

Table 2 shows the parameters and symbols used in half-car dynamics equations.

Table 2 – Half-car terms definition

x_s	Center of Gravity Displacement	m_s	Sprung Mass
x_{s1}	Front Corner Displacement	x_{s2}	Rear Corner Displacement
x_{u1}	Front Wheel Displacement	x_{u2}	Rear Wheel Displacement
c_{s1}	Front Suspension Damping	c_{s2}	Rear Suspension Damping
k_{s1}	Front Suspension Stiffness	k_{s2}	Rear Suspension Stiffness
k_{t1}	Front Tire Stiffness	k_{t2}	Rear Tire Stiffness
x_{z1}	Front Road Profile	x_{z2}	Rear Road Profile
a	CG to Front Axle Distance	b	CG to Rear Axle Distance

2.3 Full-car with 2 axles

Full-car suspension models attempt to capture entire vehicle dynamics. Elimination of turning motion, or yaw, simplifies the investigation. Full-car models investigate vehicle heave motion, pitch motion, and roll motion. Four wheeled vehicle full-car suspension models combine four quarter-car suspension models (one for each wheel station) joined by a rigid frame.

A full-car model with 2 axles contains four corners creating seven degrees of freedom for the system. Similar to the half-car model, vertical motion of each unsprung mass counts as one degree of freedom, pitch and roll angles of the sprung mass are two other degrees of freedom, and vertical motion of the sprung mass is the 7th and the last degree of freedom of the full-car model shown in Figure 2-3.

A full-car model with two axles has been developed in MATLAB and delivered as one of the deliverables of the project.

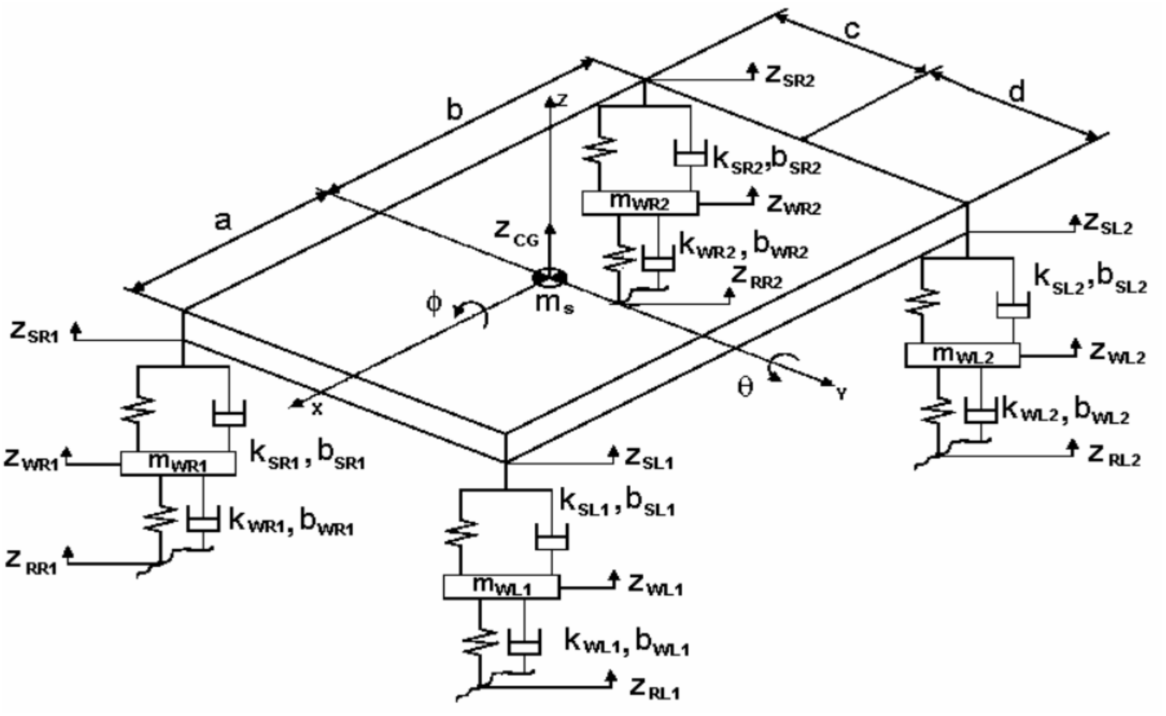


Figure 2-3 - Full-car model with two axles

Similar to quarter-car and half-car models, one can write the dynamics equations of the full-car model:

$$\begin{aligned} m_s\ddot{x}_s + c_{sr1}(\dot{x}_{sr1} - \dot{x}_{ur1}) + c_{sr2}(\dot{x}_{sr2} - \dot{x}_{ur2}) + k_{sr1}(x_{sr1} - x_{ur1}) + k_{sr2}(x_{sr2} \\ - x_{ur2}) + c_{sl1}(\dot{x}_{sl1} - \dot{x}_{ul1}) + c_{sr2}(\dot{x}_{sl2} - \dot{x}_{ul2}) + k_{sr1}(x_{sl1} - x_{ul1}) \\ + k_{sl2}(x_{sl2} - x_{ul2}) = 0 \end{aligned} \qquad \text{Eq. 7}$$

$$\begin{aligned} I_s\ddot{\theta}_s + a(c_{sr1}(\dot{x}_{sr1} - \dot{x}_{ur1}) + k_{sr1}(x_{sr1} - x_{ur1}) + c_{sl1}(\dot{x}_{sl1} - \dot{x}_{ul1}) \\ + k_{s1}(x_{sl1} - x_{ul1})) - b(c_{sr2}(\dot{x}_{sr2} - \dot{x}_{ur2}) + k_{sr2}(x_{sr2} - x_{ur2}) \\ + c_{sr2}(\dot{x}_{sr2} - \dot{x}_{ur2}) + k_{sr2}(x_{sr2} - x_{ur2})) = 0 \end{aligned} \qquad \text{Eq. 8}$$

$$\begin{aligned} I_s\ddot{\varphi}_s + c(c_{sr1}(\dot{x}_{sr1} - \dot{x}_{ur1}) + k_{sr1}(x_{sr1} - x_{ur1}) + c_{sr2}(\dot{x}_{sr2} - \dot{x}_{ur2}) \\ + k_{sr2}(x_{sr2} - x_{ur2})) - d(c_{sr1}(\dot{x}_{sl1} - \dot{x}_{ul1}) + k_{s1}(x_{sl1} - x_{ul1}) \\ + c_{sr2}(\dot{x}_{sr2} - \dot{x}_{ur2}) + k_{sr2}(x_{sr2} - x_{ur2})) = 0 \end{aligned}$$

$$m_{ur1}\ddot{x}_{sr1} - c_{sr1}(\dot{x}_{sr1} - \dot{x}_{ur1}) - k_{sr1}(x_{sr1} - x_{ur1}) + k_{tr1}(x_{ur1} - x_{zr1}) = 0 \qquad \text{Eq. 9}$$

$$m_{ul1}\ddot{x}_{s1} - c_{sl1}(\dot{x}_{sl1} - \dot{x}_{ul1}) - k_{sl1}(x_{sl1} - x_{ul1}) + k_{tl1}(x_{ul1} - x_{zl1}) = 0 \qquad \text{Eq. 10}$$

$$m_{ur2}\ddot{x_{sr2}} - c_{sr2}(\dot{x}_{sr2} - \dot{x}_{ur2}) - k_{sr2}(x_{sr2} - x_{ur2}) + k_{tr2}(x_{ur2} - x_{zr2}) = 0 \qquad \text{Eq. 11}$$

$$m_{ul2}\ddot{x_{s2}} - c_{sl2}(\dot{x}_{sl2} - \dot{x}_{ul2}) - k_{sl2}(x_{sl2} - x_{ul2}) + k_{tl2}(x_{ul2} - x_{zl2}) = 0. \qquad \text{Eq. 12}$$

Parameters and symbols used in full-car model equations are brought in Table 3.

Table 3 – Half-car terms definition

x_s	Center of Gravity Displacement	m_s	Sprung Mass
θ	Pitch Angle	φ	Roll Angle
x_{sr1}	Front Right Corner Displacement	x_{sr2}	Rear Right Corner Displacement
x_{ur1}	Front Right Wheel Displacement	x_{ur2}	Rear Right Wheel Displacement
x_{sl1}	Front Left Corner Displacement	x_{sl2}	Rear Left Corner Displacement
x_{ul1}	Front Left Wheel Displacement	x_{ul2}	Rear Left Wheel Displacement
c_{sr1}	Front Right Suspension Damping	c_{sr2}	Rear Right Suspension Damping
k_{sr1}	Front Right Suspension Stiffness	k_{sr2}	Rear Right Suspension Stiffness
c_{sl1}	Front Left Suspension Damping	c_{sl2}	Rear Left Suspension Damping
k_{sl1}	Front Left Suspension Stiffness	k_{sl2}	Rear Left Suspension Stiffness
k_{tr1}	Front Right Tire Stiffness	k_{tr2}	Rear Right Tire Stiffness
k_{tl1}	Front Left Tire Stiffness	k_{tl2}	Rear Left Tire Stiffness
x_{zr1}	Front Right Road Profile	x_{zr2}	Rear Right Road Profile
x_{zl1}	Front Left Road Profile	xl_{z2}	Rear Left road Profile
a	CG to Front Axle Distance	b	CG to Rear Axle Distance
c	CG to Right Corners Distance	d	CG to Right Corners Distance

2.4 Full-car with desired number of axles

This model is a general car model that describes dynamics of a full-car model with any number of axles. Assuming number of axles is “n”, number of degrees of freedom would be 2×n+3 therefore, the same number of equations is required to have the dynamics of the model fully constrained. For further explanation, for a car with “n” axles, 2×n degrees of freedom are associated with vertical motion of the unsprung masses, two degrees of freedom are associated with pitch and roll angles, and vertical motion of the sprung mass is the last degree of freedom. A schematic view of the full-car model with n number of axles is depicted in

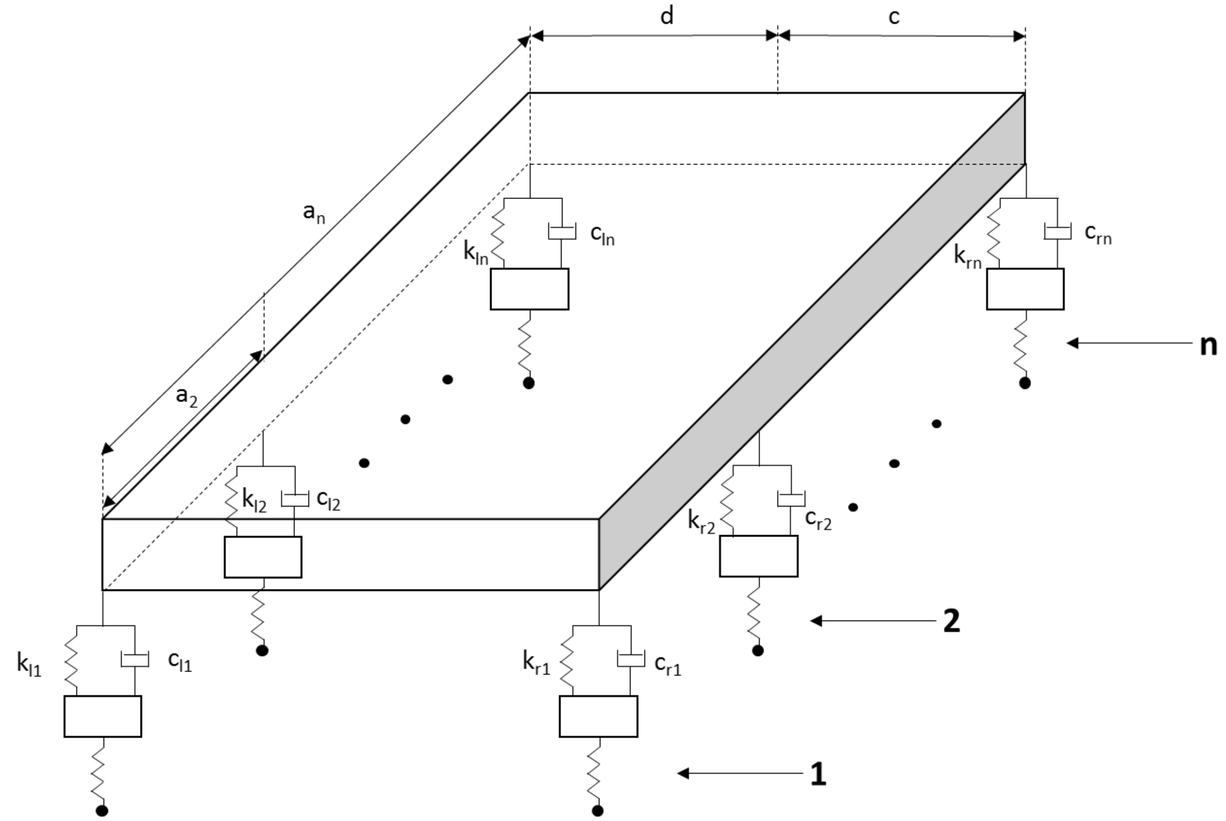


Figure 2-4 – n-axles-car model

For this model with arbitrary number of axles one can write the heave equations as follows:

$$m_s\ddot{x_s} + \sum_{i=1}^{n}(c_{sri}(\dot{x}_{sri} - \dot{x}_{uri}) + c_{sri}(\dot{x}_{sri} - \dot{x}_{uri}) + k_{sr1}(x_{sli} - x_{uri}) + k_{sr2}(x_{sl2} - x_{ul2})) = 0 \qquad \text{Eq. 13}$$

Similarly, the pitch equation can be written as below:

$$I_s\ddot{\theta_s} + \sum_{i=1}^{n}\Big((a - a_i)\big(c_{sri}(\dot{x}_{sri} - \dot{x}_{uri}) + k_{sri}(x_{sri} - x_{uri}) + c_{sli}(\dot{x}_{sli} - \dot{x}_{uli}) + k_{si}(x_{sli} - x_{uli})\big)\Big) = 0. \qquad \text{Eq. 14}$$

Also, the roll equation can be written similar to pitch equation:

$$
\begin{aligned}
I_s\ddot{\varphi}_s + \sum_{i=1}^{n} \Big(c\big(c_{sri}(\dot{x}_{sri} - \dot{x}_{uri}) + k_{sri}(x_{sri} - x_{uri})\big) \\
- d\big(c_{sli}(\dot{x}_{sli} - \dot{x}_{uli}) + k_{si}(x_{sli} - x_{uli})\big)\Big) = 0.
\end{aligned}
\tag{Eq. 15}
$$

Now that the sprung mass governing equations have been derived, all sprung mass equations can be formed as below:

$$m_{uri}\ddot{x_{sri}} - c_{sri}(\dot{x}_{sri} - \dot{x}_{uri}) - k_{sri}(x_{sri} - x_{uri}) + k_{tri}(x_{uri} - x_{zri}) = 0 \tag{Eq. 16}$$

$$m_{uli}\ddot{x_{sl\imath}} - c_{sli}(\dot{x}_{sli} - \dot{x}_{uli}) - k_{sli}(x_{sli} - x_{uli}) + k_{tli}(x_{uli} - x_{zli}) = 0 \tag{Eq. 17}$$

where i varies between 1 and n. An algorithm that generates governing equations of a full-car model with desired parameters and number of axles has been developed in MATLAB and delivered as one of the deliverables of the project.

Table 4 shows the parameters and symbols for the n-axle-car model

Table 4 – Half-car terms definition

x_s	Center of Gravity Displacement	m_s	Sprung Mass
x_{sri}	ith Axle Right Corner Displacement	x_{sli}	ith Axle Left Corner Displacement
x_{uri}	ith Axle Right Wheel Displacement	x_{uli}	ith Axle Left Wheel Displacement
c_{sri}	ith Axle Right Suspension Damping	c_{sli}	ith Axle Left Suspension Damping
k_{sri}	ith Axle Right Suspension Stiffness	k_{sli}	ith Axle Left Suspension Stiffness
k_{tri}	ith Axle Right Tire Stiffness	k_{tli}	ith Axle Left Tire Stiffness
x_{zri}	ith Axle Right Road Profile	x_{zli}	ith Axle Left Road Profile
a_i	ith Axle to Front Axle Distance	a	CG to Front Axle Distance
c	CG to Right Corners Distance	d	CG to Right Corners Distance

3 Primary Controllers

Many control design problems have investigated control of semi-active suspensions although, most of the cases considered control design of active suspensions since they are not dealing with dissipative nature suspensions and only the applied force on the masses by the suspension is controlled.

3.1 Controllers

According to the literature, the semi-active suspension control problem is considered a widely investigated topic as many control methodologies have been developed in this area. In this section, most well-known and used ones are introduced and analyzed with their references. In other words, this chapter contains some of the existing control designs along with their evaluations. Rather than lengthy explanation of each controller, the main goal of this chapter is evaluation and analysis of each controller on different vehicle models. In this chapter some of well-known strategies will be discussed and a set of simulations will be run to analyze the functionality of each controller.

3.1.1 Skyhook

This approach is mainly developed for active suspensions where the chassis is connected to the world reference frame (sky) and the goal is to reduce the chassis and axles vertical oscillations [3]. Therefore, an imaginary damper connects the sprung mass to the sky as shown in Figure 3-1. By isolating the sprung mass from the road profile, vertical oscillation of sprung mass reduces. The control input is F_{sky} that allows us control vertical oscillation of sprung mass but not unsprung mass. Semi-active suspensions allow us mimic behavior of skyhook by clipping the damping stiffness within the physically feasible values [8].

Figure 3-1 shows the skyhook concept and how the sprung mass is connected to the world reference frame via the imaginary damper.

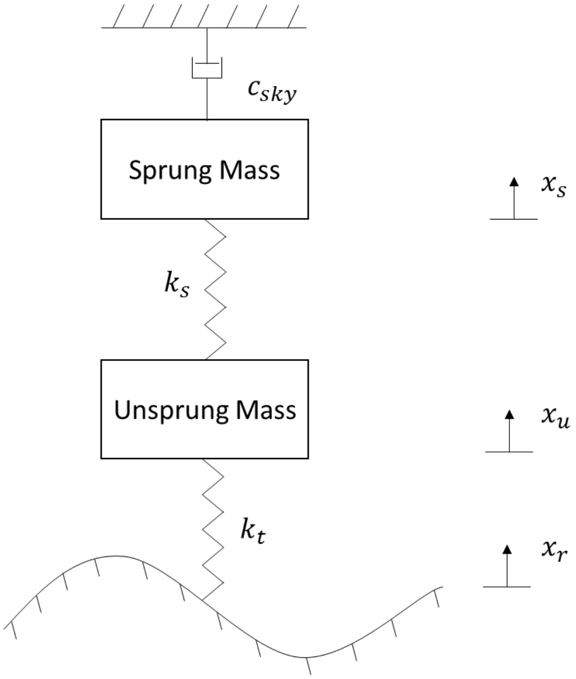


Figure 3-1 - Skyhook damper concept

As shown above, the imaginary damper force being applied on the sprung mass is calculated from Eq. 18:

$$F_{sky} = c_{sky}\dot{x}_s \qquad \text{Eq. 18}$$

where, F_{sky} is the force in imaginary damper and c_{sky} is its coefficient.
In order to have a semi-active damper emulate a skyhook damper, the damper force should be equal to what is calculated from Eq. 19. In other words, the semi-active damper coefficient should be:

$$c_s = c_{sky}\frac{\dot{x}_s}{(\dot{x}_s - \dot{x}_u)} \qquad \text{Eq. 19.}$$

Due to nature of semi-active suspensions and their hardware limitations, c can only be between two positive values. Saying above, the skyhook policy can be written as:

$$\begin{cases} \dot{x}_s(\dot{x}_s - \dot{x}_u) > 0 & c_s = c_{sky}\dfrac{\dot{x}_s}{(\dot{x}_s - \dot{x}_u)} \\ \dot{x}_s(\dot{x}_s - \dot{x}_u) < 0 & c_s = c_{min} \end{cases} \qquad \text{Eq. 20.}$$

where, c_{min} is the minimum value that the semi-active damper can acquire. Note that the value obtained from above control law has to be clipped between minimum and maximum damping coefficients of the damper, namely, c_{min} and c_{max}.

3.1.2 Groundhook

Groundhook control scheme is similar to skyhook except that the imaginary damper connects the unsprung mass to the world reference frame. Similar to skyhook, the semi-active damper coefficient is set such that it applies the same force on the unsprung mass as the imaginary damper would.

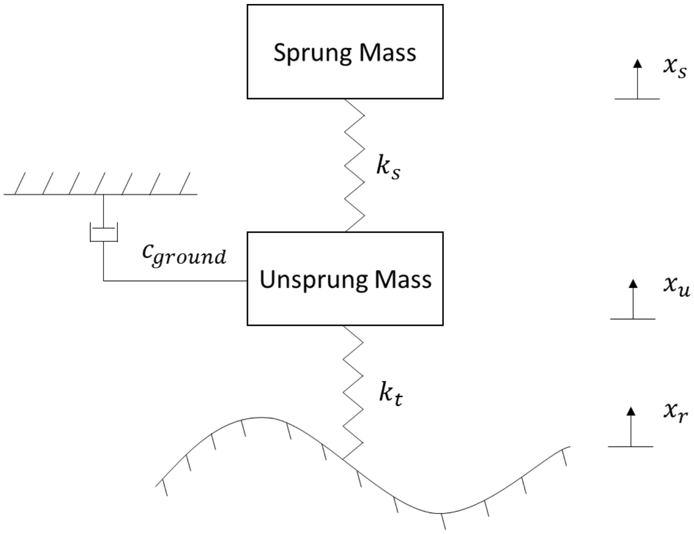


Figure 3-2 – Groundhook damper concept

Again, similar to skyhook, the force being applied by the imaginary damper is set equal to the semi-active damper force:

$$c_g(\dot{x}_s - \dot{x}_u) = c_{ground}\dot{x}_u \qquad \text{Eq. 21}$$

where, c_{ground} is the imaginary damper coefficient which is set by user.

Now, the control policy for ground hook can be formed as:

$$\begin{cases} \dot{x}_u(\dot{x}_s - \dot{x}_u) < 0 & c_g = c_{ground}\dfrac{\dot{x}_u}{(\dot{x}_s - \dot{x}_u)} \\ \dot{x}_u(\dot{x}_s - \dot{x}_u) > 0 & c_g = c_{min} \end{cases} \qquad \text{Eq. 22.}$$

As there are limitations with the nature of semi-active dampers, the value of c_g is undergone a saturation process that limits it off between the minimum and maximum feasible damping coefficients.

3.1.3 Hybrid Skyhook-Groundhook

As explained before, skyhook minimizes the sprung mass vibrations whereas, groundhook reduces unsprung mass fluctuations. However, they both act at the cost of increasing the other masses motions. In order to take advantage of positive aspects of both controllers, a hybrid concept could be used. Figure 3-3 shows schematic view of the hybrid controller.

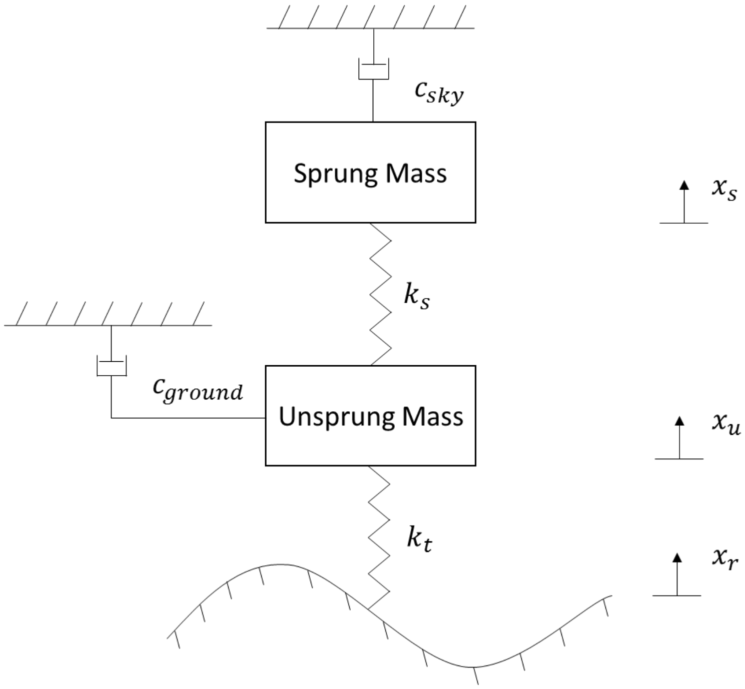


Figure 3-3 – Hybrid skyhook-groundhook controller

Figure above shows how the imaginary dampers connect the unsprung and sprung masses to the world reference frame. In this control scheme, the final damper coefficient c is calculated from below equation:

$$c = \alpha c_s + (1 - \alpha)c_g \qquad \text{Eq. } 23$$

where, α varies between zero and one and is the relative ratio between skyhook and groundhook control. At $\alpha = 1$ it becomes a pure skyhook control policy as opposed to $\alpha = 0$, where it is purely groundhook.

3.1.4 Acceleration Driven Damper (ADD)

This approach was first introduced in [10]. This strategy is known as a simple yet effective control algorithm and shown to be optimal. It minimizes vertical acceleration of sprung mass by adjusting the damping coefficient. The ADD method adjusts damping coefficient as described in Eq. 24:

$$c_{in} = \begin{cases} c_{min} \;\; if\; \ddot{x_s}\dot{x}_{def} \leq 0 \\ c_{max} \;\; if\; \ddot{x_s}\dot{x}_{def} > 0 \end{cases} \qquad \text{Eq. 24}$$

where c_{in}, c_{min}, and c_{max} are the semi-active suspension damping coefficient, minimum damping coefficient and maximum damping coefficient, respectively. The suspension force f is calculated using Eq. 25:

$$f = c_{in}\dot{x}_{def} \qquad \text{Eq. 25}$$

This control approach is very well adapted for human comfort but the switching of damper coefficient values affects the closed-loop performance. In other words, ADD requires fast switching of damping coefficient, which is not as practical.

3.1.5 Power Driven Damper (PDD)

PDD approach, introduced in [11], controls the energy stored and the power dissipated in a semi-active suspension. The results are comparable to those of ADD but the chattering effect of the control input is resolved to some extent. The proposed control law is shown in below equation:

$$c_{in} = \begin{cases} c_{min} & if\; k_s x_{def}\dot{x}_{def} + c_{min}\dot{x}_{def} \geq 0 \\ c_{max} & if\; k_s x_{def}\dot{x}_{def} + c_{min}\dot{x}_{def} < 0 \\ \dfrac{c_{min} + c_{max}}{2} & if\; x_{def} \neq 0\; and\; \dot{x}_{def} = 0 \\ -\dfrac{k_s x_{def}}{\dot{x}_{def}} & Otherwise \end{cases} \qquad \text{Eq. 26}$$

Again, the suspension force is calculated from equation Eq. 25. The advantage of this approach is its low-chattering control input, although it needs knowledge of suspension stiffness value. The damping coefficient can acquire any value between the maximum and minimum values in PDD as opposed to ADD.

3.2 Simulations and Results

In this section, the mentioned algorithms are implemented on different car models in MATLAB/Simulink environment and the results are discussed analytically.

3.2.1 Quarter-Car

The control algorithms discussed above was implemented on a MATLAB/Simulink quarter-car model. The quarter-car parameters were selected to represent a heavy truck. The sprung mass was assumed to be $m_s = 2250kg$, unsprung mass was assumed to be $m_u = 200kg$, the minimum damping coefficient was $c = 2000\, {}^{N.s}/_{m}$, the maximum damping coefficient was $c = 40000\, {}^{N.s}/_{m}$, suspension stiffness was $k_s = 180000\, {}^{N}/_{m}$, and the tire stiffness assumed to be $k_t = 500000\, {}^{N}/_{m}$. Also, a passive system was considered for comparing the semi-active suspension algorithms with a passive suspension case. The passive damper value was assumed $c = 5000\, {}^{N.s}/_{m}$. Also, the road-profile shown in Figure 3-4 was used.

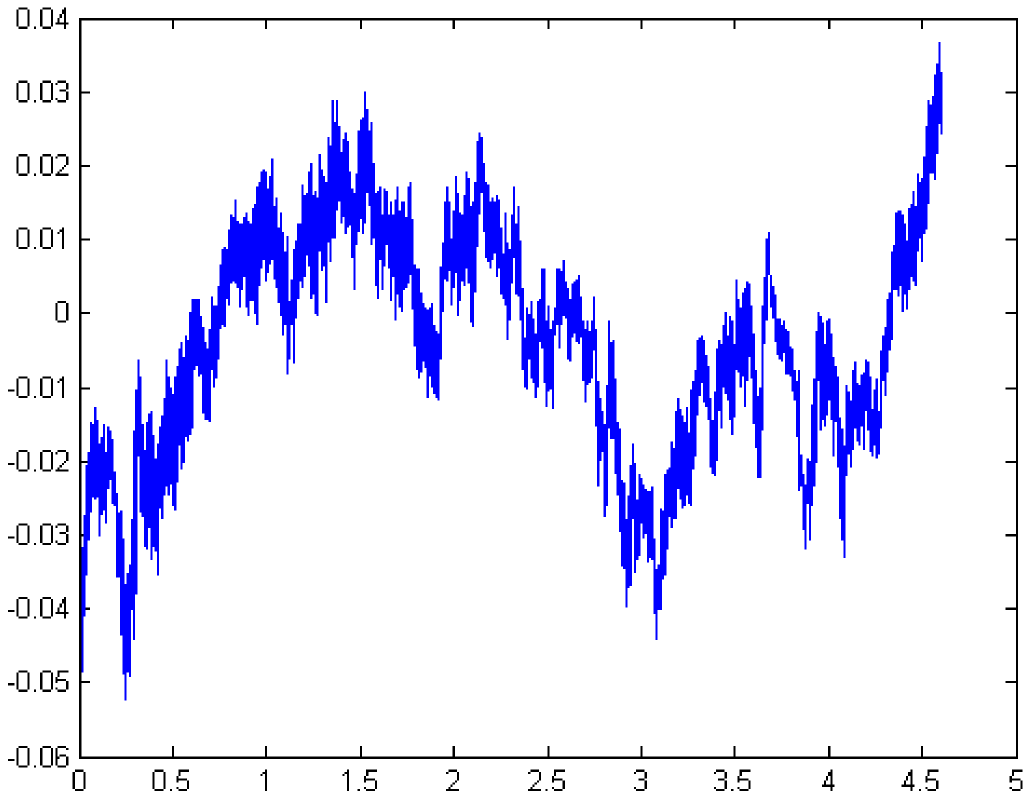


Figure 3-4 – Road profile used in quarter-car semi-active suspension control

A set of simulations was run for ADD, PDD, and a passive case and following results were obtained. Sprung mass displacement for each control methodology is shown in Figure 3-5.

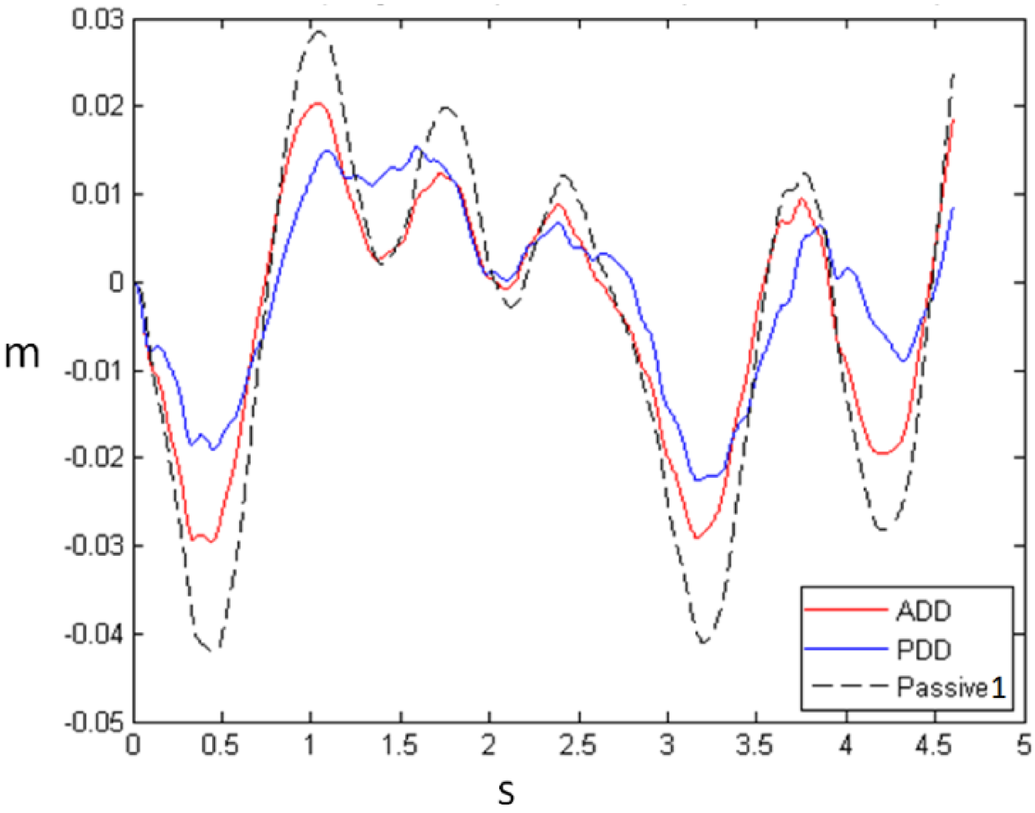


Figure 3-5 – Sprung mass displacement of ADD, PDD, and passive 1 cases

The sprung mass acceleration root-mean-squared (RMS) and suspension deflection RMS values are brought in Table 5.

Table 5 - Sprung mass acceleration RMS and suspension deflection RMS

Approach	**Acceleration RMS**	**Suspension Deflection RMS**
PDD	1.014	0.0094
ADD	1.033	0.0102
Passive 1	1.4613	0.0149

The discussion of results can be classified as follows:

ADD: This algorithm is easy to implement and showed improvement on acceleration of sprung mass. The disadvantage of this method is the chattering effects in calculated damping coefficients as mentioned in section 3.1.4.

PDD: This is algorithm is also easy to implement but it has less chattering effect compared to ADD because it can acquire any damping value between the minimum and maximum values. The previous statement is the main reason that PDD performed better than ADD.

Passive Case: As expected, the passive damper returned poorer human comfort and road holding values. It should be noted that not all the time a passive damper would perform worse than ADD and PDD and that highly depends on the road profile but generally those controllers are expected to perform better. In other words, a proper control algorithm softens and stiffens the damper appropriately in order to increase ride quality indices.

3.2.2 Half-car – ADD, PDD, and Passive

In this section, different semi-active suspension control algorithms were implemented on a half-car model. Furthermore, a bump stop is incorporated into the suspension model. A bump stop protects the suspension and vehicle (as well as the occupants) from violent "bottoming" of the suspension. By this simple modeling of bump stop, the suspension stiffness suddenly becomes ten times larger once the suspension hits the travel limit.

The control methods used in this study are ADD and PDD in addition to a passive case. Following Parameters were used in the half-car model:

Table 6 - Half-car model parameters

Parameter	**Value**	**Parameter**	**Value**
Sprung Mass m_s	4500 kg	Unsprung Mass m_u	200 kg
Front Suspension Stiffness k_s	130000 N/m	Front Tire Stiffness k_t	500000 N/m
Rear Suspension Stiffness k_s	180000 N/m	Passive Damping	20000 N.m/s
Minimum Damping c_{min}	2000 N.m/s	Maximum Damping c_{max}	40000 N.m/s

Also, four different road profiles with different RMS values were tested on 5 velocities of the vehicle; 5m/s, 7.5m/s, 10m/s, 12.5m/s, 15m/s. The road profiles are depicted in Figure 3-6 where the RMS of each one is mentioned above it.

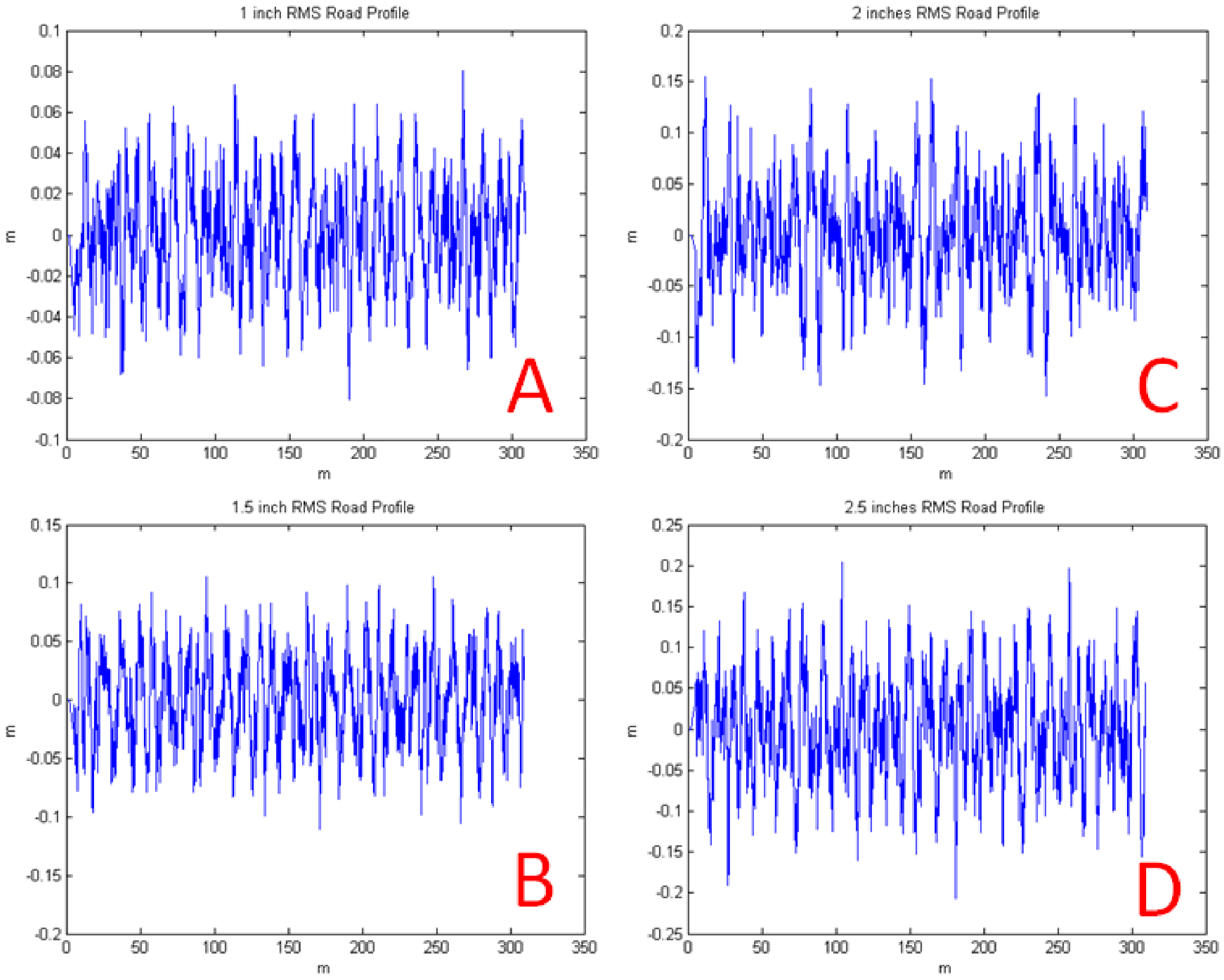


Figure 3-6 - Road profiles

Following tables show results of twenty simulations (5 velocities on 4 road profiles) for each control methodology (4 cases). Each table corresponds for a velocity value from 5m/s to 15m/s with 2.5m/s increment.

Table 7 - Results of the half-car simulation with 5m/s velocity

Method	**Road Profile**	**Front Axle Acc. RMS**	**Rear Axle Acc. RMS**	**CG Acc RMS**	**Absorbed Power**
ADD	A	1.563	1.861	1.119	1.293
	B	2.527	3.475	1.904	3.170
	C	3.367	4.118	2.419	4.671
	D	3.917	4.953	2.912	6.642
PDD	A	1.236	1.835	0.955	0.578
	B	2.395	3.961	1.719	1.657
	C	3.599	4.640	2.299	2.892
	D	4.468	7.666	3.623	4.477
Passive	A	2.382	2.720	1.757	2.951
	B	3.550	3.748	2.489	5.626
	C	4.378	5.232	3.150	8.158
	D	6.002	6.748	4.290	13.877

Table 8 - Results of the half-car simulation with 7.5m/s velocity

Method	**Road Profile**	**Front Axle Acc. RMS**	**Rear Axle Acc. RMS**	**CG Acc RMS**	**Absorbed Power**
ADD	A	2.031	2.622	1.473	2.201
	B	2.799	3.765	2.128	4.233
	C	3.868	6.321	3.259	7.662
	D	4.674	6.849	3.546	10.054
PDD	A	1.851	2.923	1.497	1.170
	B	2.573	4.663	1.973	2.291
	C	3.755	7.572	3.705	4.753
	D	4.909	17.260	7.534	11.906
Passive	A	3.035	3.338	2.046	4.630
	B	4.356	4.768	2.772	7.468
	C	5.584	6.791	3.793	12.513
	D	7.168	8.203	4.712	21.315

Table 9 - Results of the half-car simulation with 10m/s velocity

Method	**Road Profile**	**Front Axle Acc. RMS**	**Rear Axle Acc. RMS**	**CG Acc RMS**	**Absorbed Power**
ADD	A	2.188	2.879	1.560	2.800
	B	3.451	0.055	2.610	4.700
	C	4.242	6.117	3.054	8.724
	D	6.337	8.284	4.411	14.065
PDD	A	2.109	3.577	1.764	1.636
	B	3.269	4.216	2.616	3.084
	C	9.650	25.003	10.190	7.885
	D	10.338	29.725	13.516	14.531
Passive	A	3.448	4.161	2.235	6.038
	B	4.892	6.039	3.015	9.648
	C	6.254	8.889	4.284	17.346
	D	7.966	10.184	5.064	28.225

Table 10 - Results of the half-car simulation with 12.5m/s velocity

Method	Road Profile	Front Axle Acc. RMS	Rear Axle Acc. RMS	CG Acc RMS	Absorbed Power
ADD	A	2.508	3.267	1.788	3.720
	B	3.687	5.821	3.034	8.683
	C	4.759	6.401	3.442	10.775
	D	6.590	7.833	4.514	17.856
PDD	A	2.130	3.708	1.767	1.671
	B	3.464	6.227	3.133	5.274
	C	6.074	8.293	4.213	6.412
	D	7.467	16.327	7.338	15.923
Passive	A	3.631	4.918	2.294	6.405
	B	5.371	7.506	3.579	14.106
	C	7.141	10.277	4.435	21.298
	D	8.954	12.239	5.524	34.568

Table 11 - Results of the half-car simulation with 15m/s velocity

Method	**Road Profile**	**Front Axle Acc. RMS**	**Rear Axle Acc. RMS**	**CG Acc RMS**	**Absorbed Power**
ADD	A	2.648	3.233	1.952	4.512
	B	3.887	5.477	2.906	8.213
	C	5.054	6.504	3.690	13.915
	D	6.194	9.063	4.895	24.431
PDD	A	2.097	3.620	1.801	1.842
	B	3.315	6.757	3.008	4.728
	C	4.450	10.799	5.007	8.273
	D	5.833	16.734	7.546	16.282
Passive	A	3.787	5.415	2.419	5.616
	B	5.617	8.514	3.741	12.396
	C	7.678	10.973	4.747	20.964
	D	9.451	14.059	6.077	31.995

As seen in tables above, PDD is performing well especially for the lower speeds. However, as the speed of the vehicle increases, road profile becomes rougher and chance of hitting bump stop increases as well. As a conclusion, the algorithm that performs very well in minimizing the sprung mass acceleration RMS when a bump stop does not exist. It would not be guaranteed to perform the same in existence of a bump stop.

3.2.3 Full-car with two axles

In this section, ADD and PDD are implemented on a full-car model and compared to a passive case. The vehicle assumed to have 9000kg of sprung mass ($m_s = 9000kg$), each unsprung mass is $m_u = 200kg$, the minimum damping coefficient was $c = 2000\, {}^{N.s}/_{m}$, the maximum damping coefficient was $c = 40000\, {}^{N.s}/_{m}$, suspension stiffness for the front axle was $k_s = 130000\, {}^{N}/_{m}$and it was $k_s = 180000\, {}^{N}/_{m}$ for the rest of the axles, and the tire stiffness assumed to be $k_t = 500000\, {}^{N}/_{m}$ for all the axles. Moreover, the road profiles depicted in Figure 3-7 were used for the simulation.

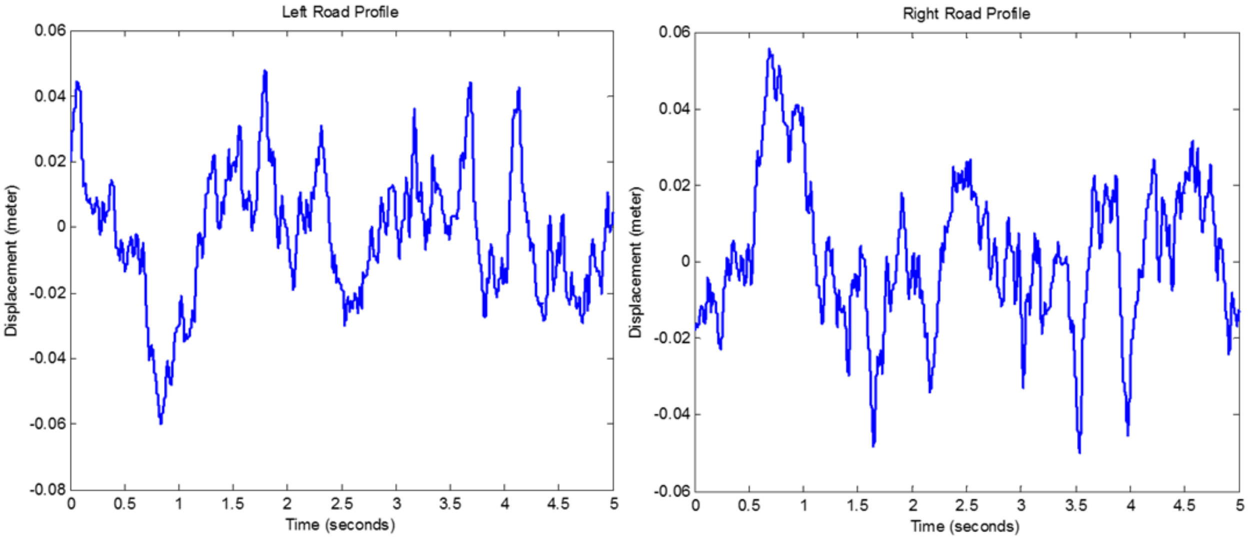


Figure 3-7 - Road profiles used for simulation of the full-car model

Having above road profiles for the left and right axles, a set of simulations was run and following results were obtained.

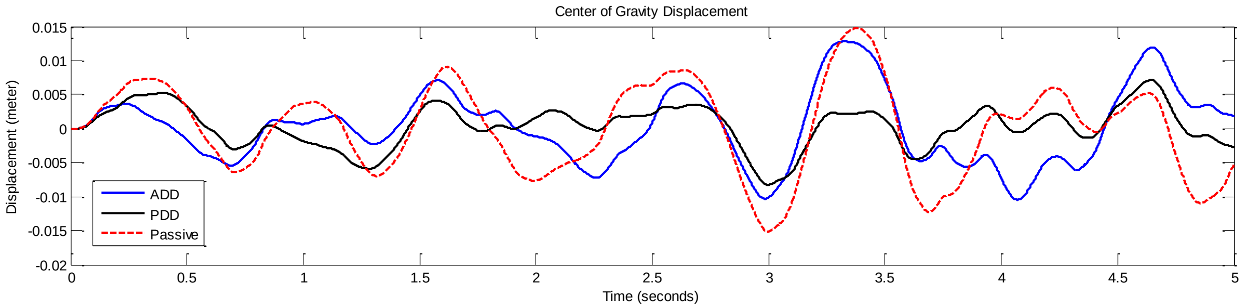


Figure 3-8 - Center of gravity displacement for full-car model with 2 axles

Figure above shows that both strategies based on modified H∞ are performing better than the other cases in terms of magnitudes of peaks. Figure below shows the center of gravity acceleration plots of each case versus the passive suspension.

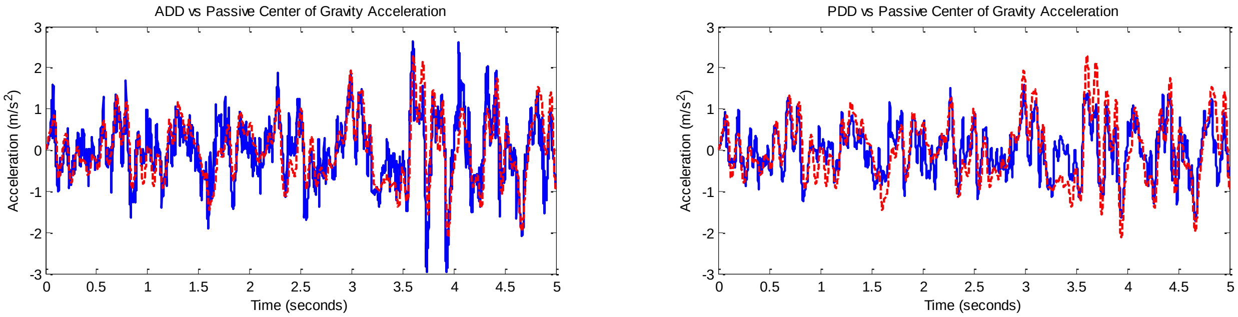


Figure 3-9 - Center of gravity acceleration for full-car model with 2 axles

Also, the pitch angle plot versus time is shown in Figure 3-10.

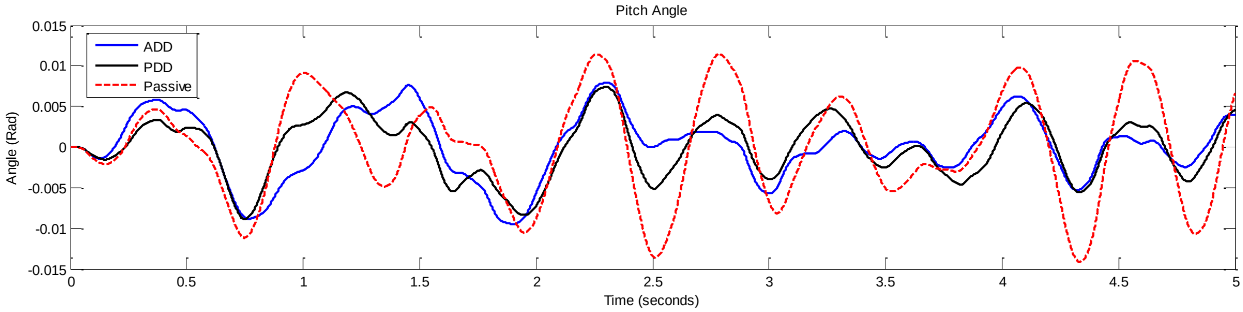


Figure 3-10 - Pitch angle for full-car model with 2 axles

Furthermore, pitch angular acceleration is shown in figure below.

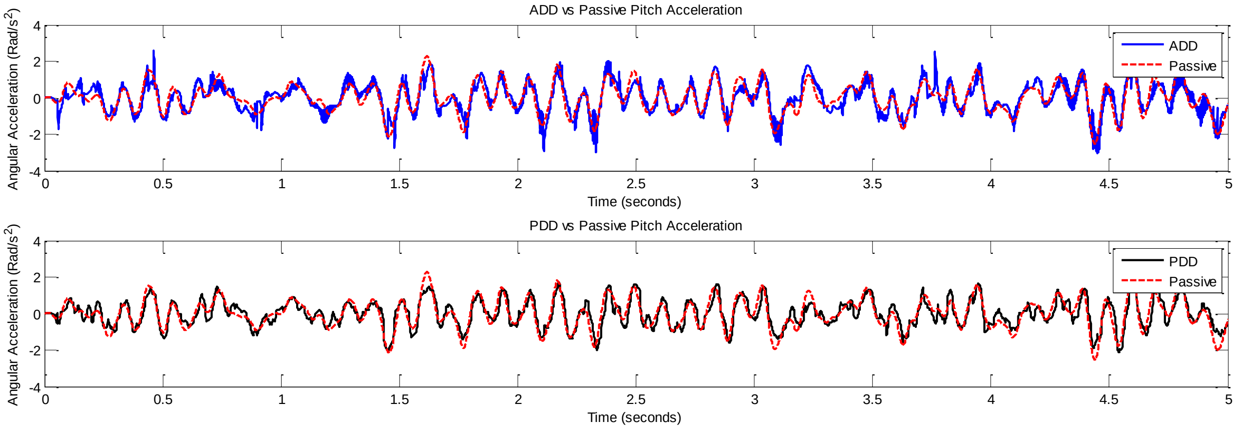


Figure 3-11 - Pitch angular acceleration for full-car model with 2 axles

Additionally, the roll angle is shown in Figure 3-12.

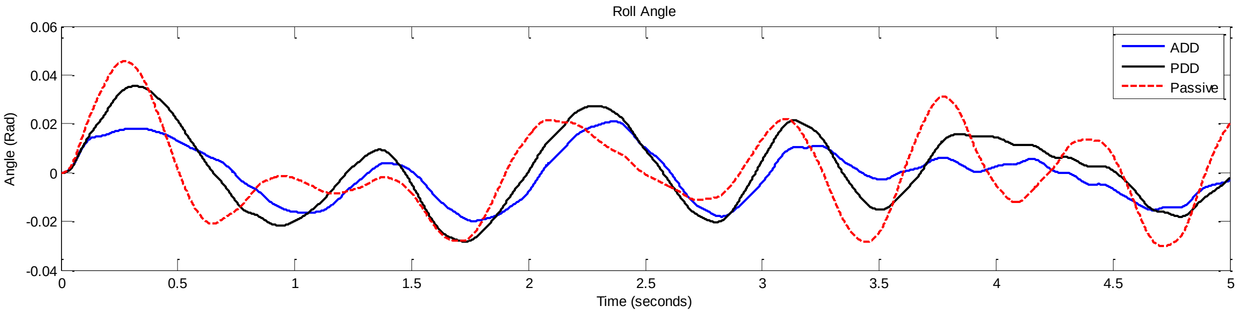


Figure 3-12 - Roll angle for full-car model with 2 axles

Roll angular acceleration is depicted in Figure 3-13.

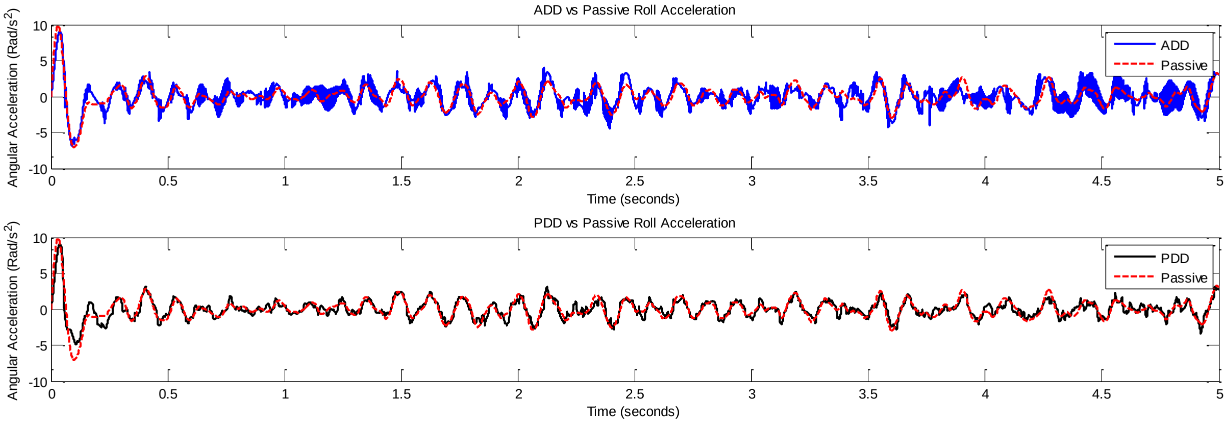


Figure 3-13 - Pitch angular acceleration for full-car model with 2 axles

Figures above can also be presented in form of Table 12 containing acceleration RMS values the center of gravity and all four corners.

Table 12 – Acceleration RMS values for full-car model

Method	**CG Acceleration RMS**	**Front Right Acceleration RMS**	**Front Left Acceleration RMS**	**Rear Right Acceleration RMS**	**Rear Left Acceleration RMS**

ADD	0.77986	1.9782	1.9403	1.8011	1.8782
PDD	0.59976	1.9163	1.868	1.8541	1.8979
Passive	0.76511	2.4377	2.502	2.7747	2.419

Table above shows PDD is the best algorithm and ADD is the last control scheme in terms of acceleration RMS. This can be interpreted as the nature of ADD and PDD since they are corner dependent algorithms. They only control each corner without knowing what is happening to the other corners. Because the corners of a car are dynamically interconnected, an algorithm that controls the entire dynamic system is potentially a better one.

4 One-at-a-time analysis and tuning of semi-active suspension controllers

In this chapter analyzing previously introduced controllers will be discussed and tuning of their performances will be investigated. Tuning of the controllers can be done in two ways, either by switching between controllers due to the situation that the car is going through or by adjusting tuning variables that some certain controllers have in their control laws.

As mentioned earlier, ride metrics of a vehicle can be divided to two general indices, human comfort and road holding. As explained before, human comfort can be represented by sprung mass acceleration RMS while road holding can be quantified by suspension deflection RMS. The indices can be different in various literatures and resources but the ones that are widely used by researchers are chosen here.

Saying above, performance of a suspension system can be represented by a point on a plot where the vertical axis is sprung mass acceleration RMS and the horizontal axis shows the suspension deflection RMS. IT is obvious that the goal is to get as close as possible to the origin where the two indices are equal to zero meaning the passenger is experiencing a theoretical (not feasible) perfect ride. Therefore, by “tuning” in this document, moving the performance point of a suspension system closer to the origin of above mentioned is referred.

4.1 Choosing the right controller

A methodology is proposed on how to pick the controller that outperforms the best, while one or more ride variables are varying. This methodology helps engineers compare and evaluate ride metrics of a control algorithm and choose the one that suits the situation.

The proposed methodology is implemented on ADD, PDD, Skyhook, and Groundhook shows what ride properties to look at when choosing the best controller amongst those four. The quarter-car model with parameters described in 5.2.1 is investigated using simulations programmed in MATLAB environment. Two different cases were considered. One, where the sprung mass is varying. This case happens when a military vehicle is launching missiles or when and agricultural vehicle is spraying s land. The other case simulated a situation when the car is going on a road where changes of the road profile RMS is significant. In the first simulation the sprung mass was varied between 0.1 and 2 times of the nominal value. The road profile variation was the same as sprung mass. In other words, the road profile RMS was changing between 0.1 and 2 times of the nominal value.

4.1.1 Results

For the first case the performance graph that is shown in Figure 4-1 was obtained. Circles represent the performance where the sprung mass is at its nominal value. Starts are representing the minimum sprung mass value, 10% of the nominal sprung mass value.

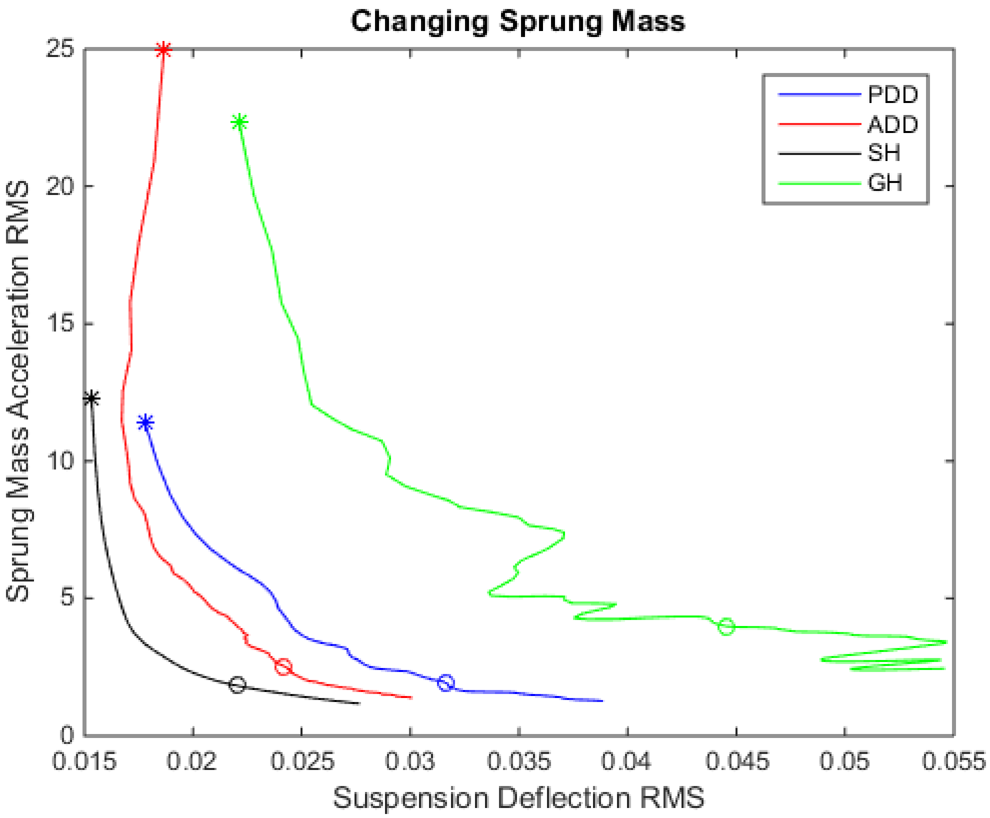


Figure 4-1 - Performance graph of ADD, PDD, Skyhook, and Groundhook when sprung mass was varying

Undoubtedly, Skyhook (SH) has the best performance in terms of both of the ride metrics. PDD is slightly behind the SH. GH is not a good option in this case due to having an unpredictable trend.

The second case, where road profile RMS was varying, returned the following figure.

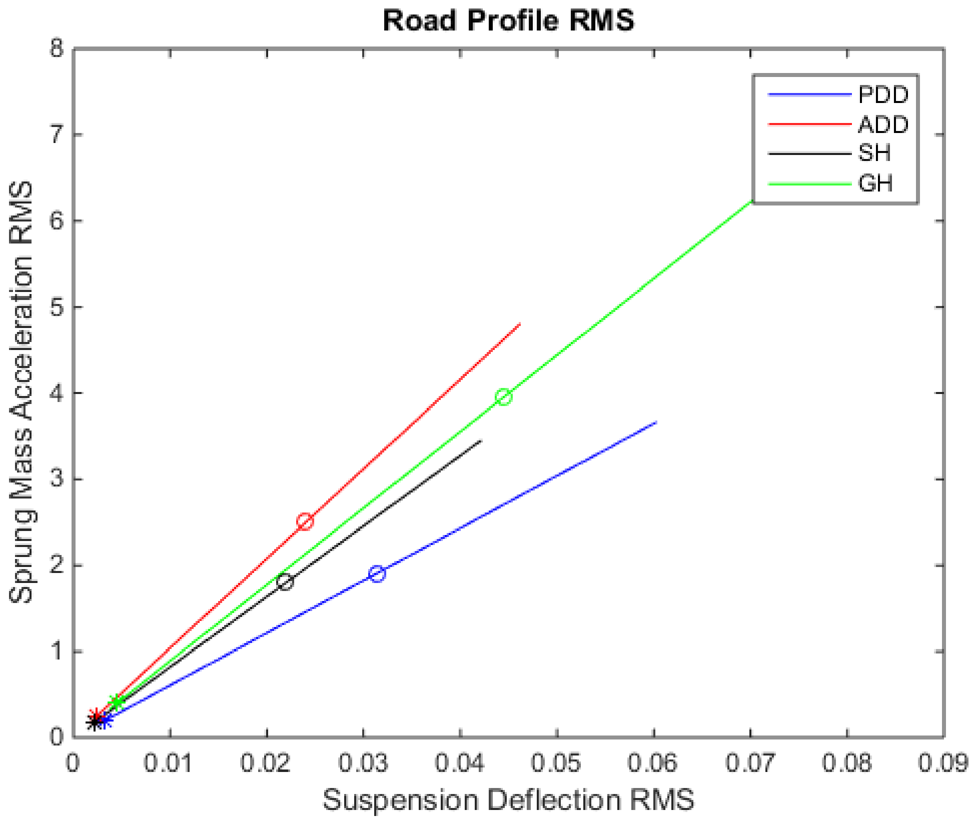


Figure 4-2 - Performance graph of ADD, PDD, Skyhook, and Groundhook when road profile RMS was varying

As seen in Figure 6-1, when the sprung mass is at its minimum all the four controller perform relatively similarly. However, as the sprung mass start to increase, there are two different controllers that can be picked based on the driver's need. If the road holding is of importance, PDD is the right choice however, for having a better human comfort SH is the controller that one would pick.

4.2 Fine-tuning controllers by adjusting the tuning variable

As the title says, this option is only available when one or more tuning variables show up in the controller equations. Among the introduced controllers only SH, GH, and shybrid SH-GH have this option of fine-tuning.

4.2.1 Skyhook Fine-tuning

To investigate the effect of adjusting c_{sky} in Eq. 20, it was varied between 5000 N.s/m and 100000 N.s/m. Again, two cases of varying sprung mass and varying road profile RMS were considered to see the effect of adjusting c_{sky} in both case. The variation range of both value were the same as previous section.

Result of the simulation corresponding to the varying sprung mass case is shown in Figure 4-3.

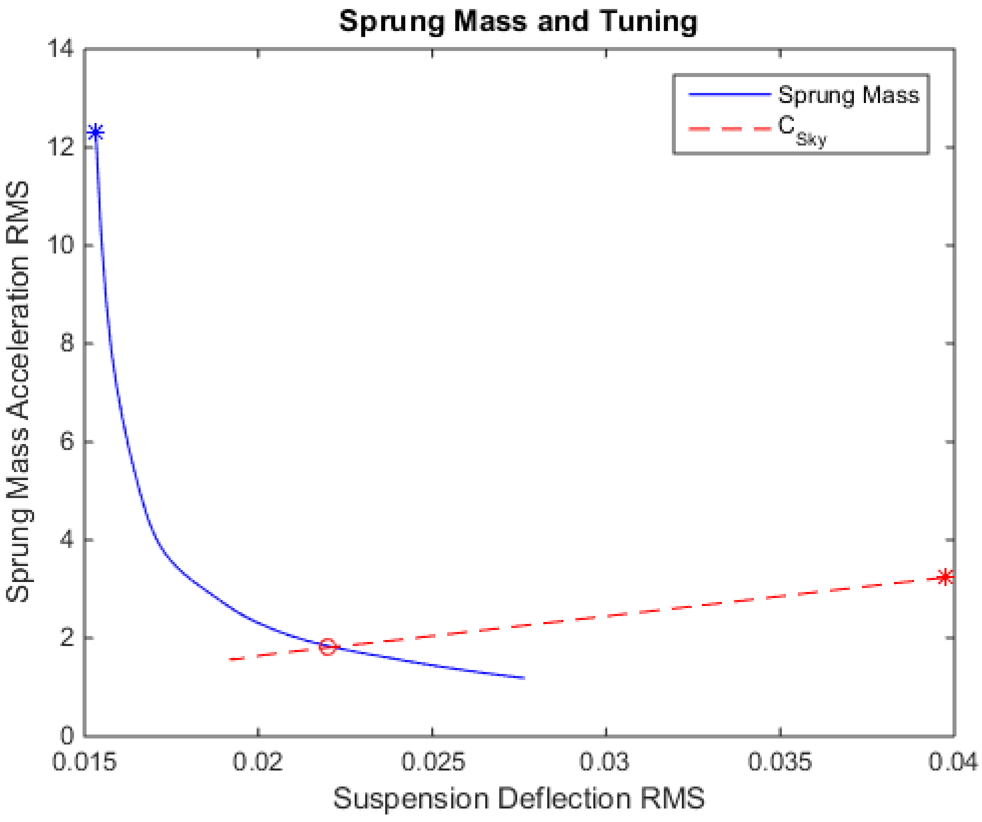


Figure 4-3 - Performance graph Skyhook when sprung mass was varying

As seen, by increasing c_{sky} value the blue curve would shift closer to the origin meaning, the performance of the controller would improve. It should be noted that the star point corresponds to $c_{sky} = 20000\ N.s/m$.

Moreover, Figure 4-4 depicts the simulation result when the road profile RMS varied.

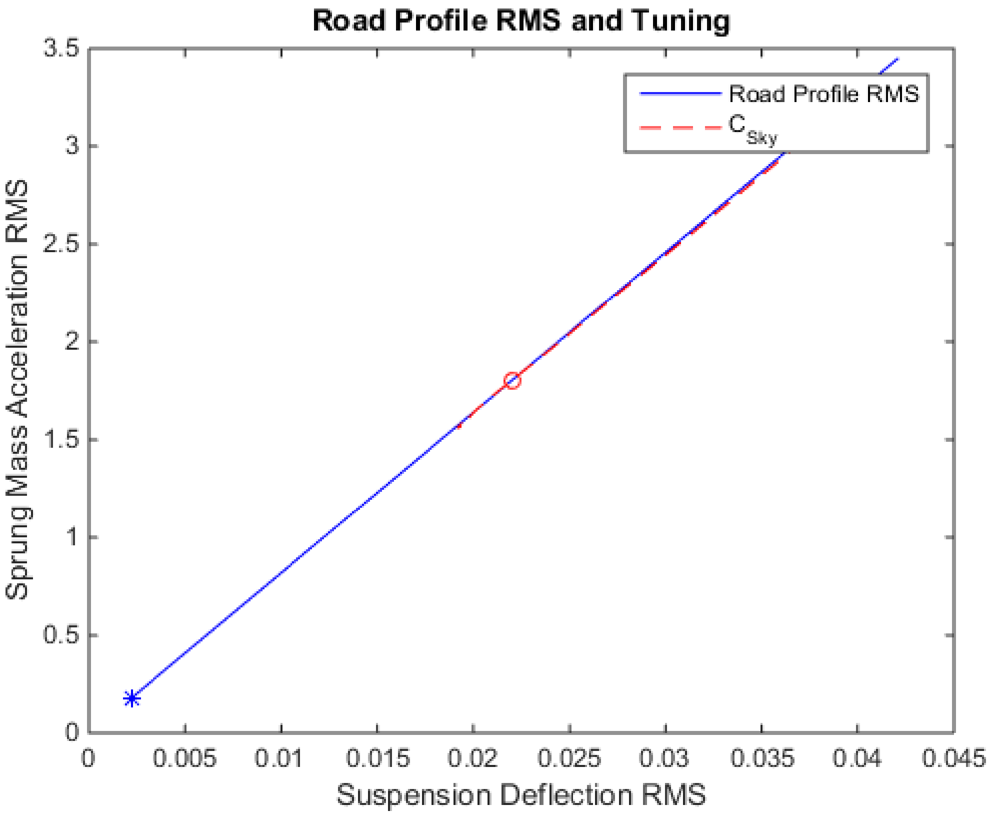


Figure 4-4 - Performance graph Skyhook when road profile RMS was varying

Figure above shows that increasing c_{sky} helps improving the ride metrics but since the direction of dashed line is align with the direction changes in road profile RMS, much enhancement is not expected. In other words, adjusting c_{sky} does not have as much effect as it does in the previous case (varying sprung mass).

4.2.2 Groundhook Fine-tuning

Similar to skyhook, effect of adjusting c_{ground} on ride metrics is the goal of this section. c_{ground} was varied between 5000 N.s/m and 100000 N.s/m with the same scenario as in 4.2.1.

Result of the simulation corresponding to the varying sprung mass case is shown in Figure 4-5.

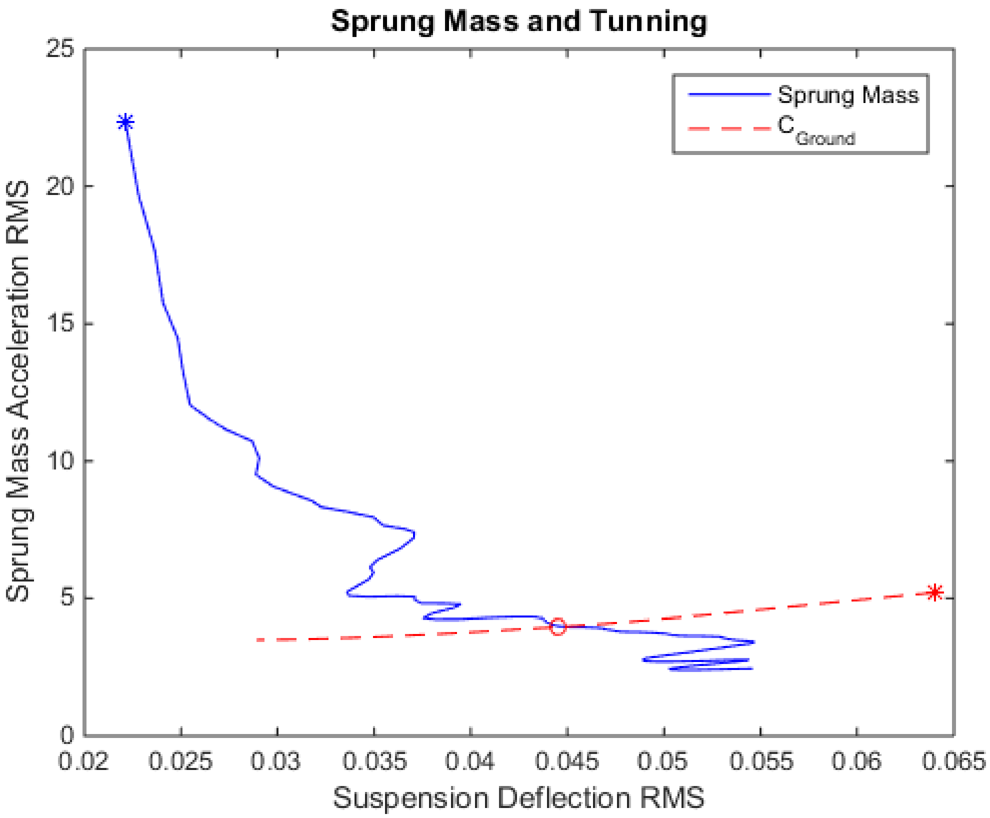


Figure 4-5 - Performance graph Groundhook when sprung mass was varying

As seen, by increasing c_{ground} value the blue curve would shift closer to the origin meaning, the performance of the controller would improve. It should be noted that the star point corresponds to $c_{ground} = 20000\ N.s/m$.

Moreover, Figure 4-6 depicts the simulation result when the road profile RMS varied.

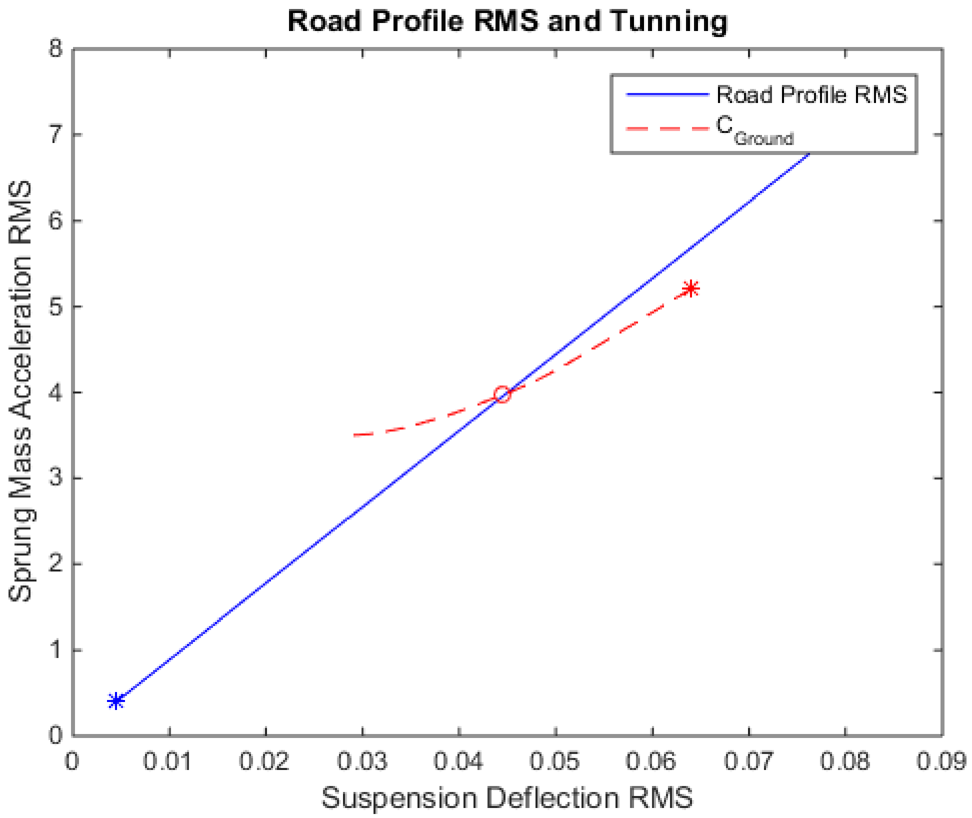


Figure 4-6 - Performance graph Groundhook when road profile RMS was varying

Figure above shows that increasing c_{ground} helps improving the ride metrics but since the direction of dashed curve is roughly align with the direction changes in road profile RMS, much enhancement is not expected. In other words, adjusting c_{ground} does not have as much effect as it does in the previous case (varying sprung mass) although comparted to the Skyhook case, the red curve has an angle with the blue curve. This angle makes the effect of adjusting the c_{ground} more significant than the Skyhook case.

4.2.3 Sensitivity Analysis

In general, sensitivity analysis (SA) determines how robust a dependent variable is to changes in a set of independent variables. In other words, using sensitivity analysis one can determine how much the output of a system is affected from perturbations in the variables/parameters of that system. SA can be used to understand the relationship between inputs and outputs of a system. This concept can be used in controls to find the parameters that have the highest level of influence on a system metrics and fine tune those parameters in order to reach better performance.

Sensitivity analysis can be performed in two ways: (1) local method (2) general method. In local method, variables are varied one by one around a fixed point and the effect of them is analyzed. However, in general method, parameters are varied in their feasible space and effect of parameters and their interactions is obtained. In this study, we are interested in the latter type. A space of parameters is assumed and the sensitivity of the system within that space is calculated.

4.2.3.1 Variance-based Methods

Variance-based SA is a general method that decomposes variance of the output into parts that each correspond to variance of an input. In this approach, output and inputs are quantified as probability distributions. This can be written as conditional expectations:

$$S_j = \frac{V[E[Y|X_j]]}{V[Y]} \qquad \text{Eq. } 27$$

where S_j is the first order effect of input variable X_j and Y is the system output. $E\left[Y\middle|X_j\right]$ denotes conditional expectation of Y given X_j and operator $V[\cdot]$ is the variance of a random variable.

Obviously, Eq. 27 only helps for a small range of systems where the relationship between the output Y and the input X_j is given and simple. For complex systems with nonlinearities, usually an estimation algorithm is used to calculate S_j. Each algorithm has its pros and cons and chosen based on the system.

4.2.3.2 Estimation

As mentioned above, numerous estimation algorithms have been developed by researchers to estimate the “main effect” of an input variable on the system output. Discussing about the history behind each algorithm and comparing their strengths and weaknesses is beyond the scope of this article. Because of simplicity, reasonable efficiency and low computational load, fourier amplitude sensitivity testing (FAST) method is used in this study.

In FAST, the conditional variances are represented by coefficients from multiple Fourier series expansion from the output function. Afterwards, a set of operations are performed to transform the multiple Fourier series expansion to a one-dimensional integral. The resulted integral is evaluated at particular sample points that are generated from incommensurate and mostly irrational frequencies. To facilitate the computation, instead of irrational frequencies, integer frequencies that are incommensurate to some order are chosen. It should be noted that is efficient compared to other variance-based global sensitivity estimation but, on the other side, it is limited to "first order" sensitive analysis.

4.2.4 Sensitivity Analysis on Skyhook-Groundhook

Sensitivity analysis was run for $\alpha = 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875$. Absorbed power, Sprung Mass Acceleration root-mean-squared (RMS), and Suspension Deflection RMS were considered as the outputs where c_{sky} and c_{ground} were inputs of the Sensitivity Analysis (SA).

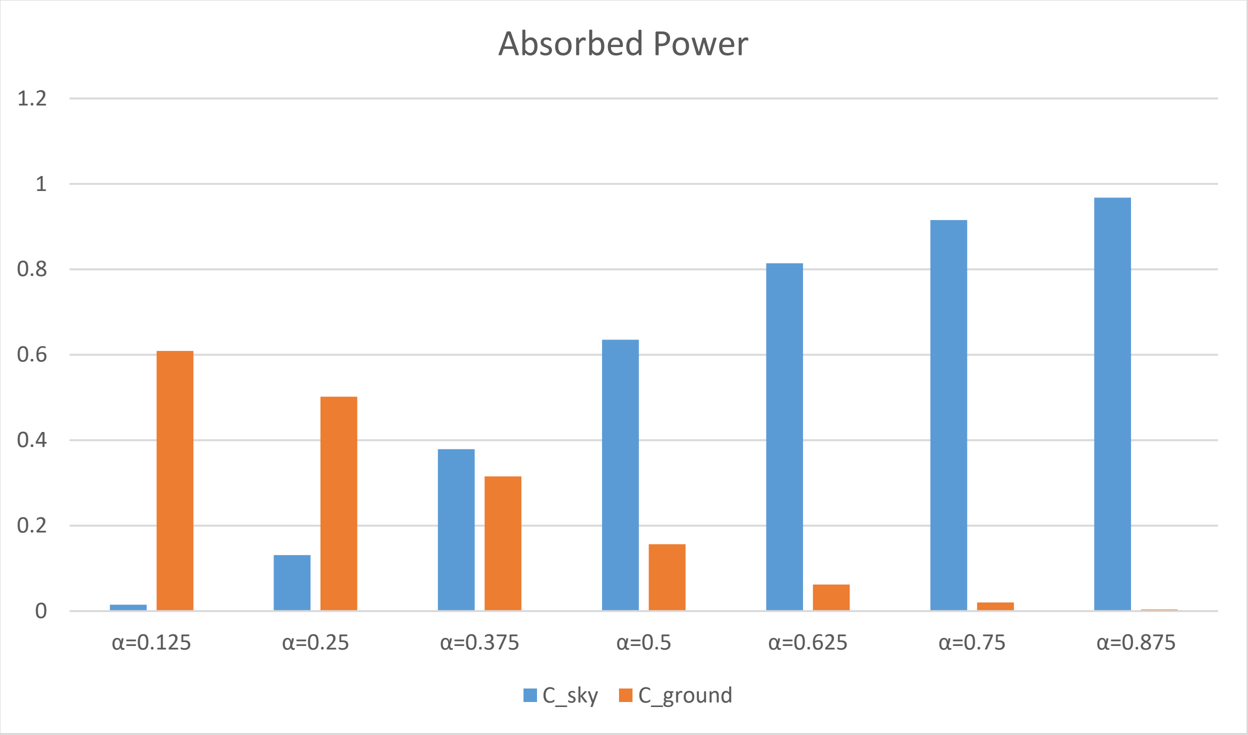


Figure 4-7 – Absorbed power sensitivity to c_{sky} and c_{ground}

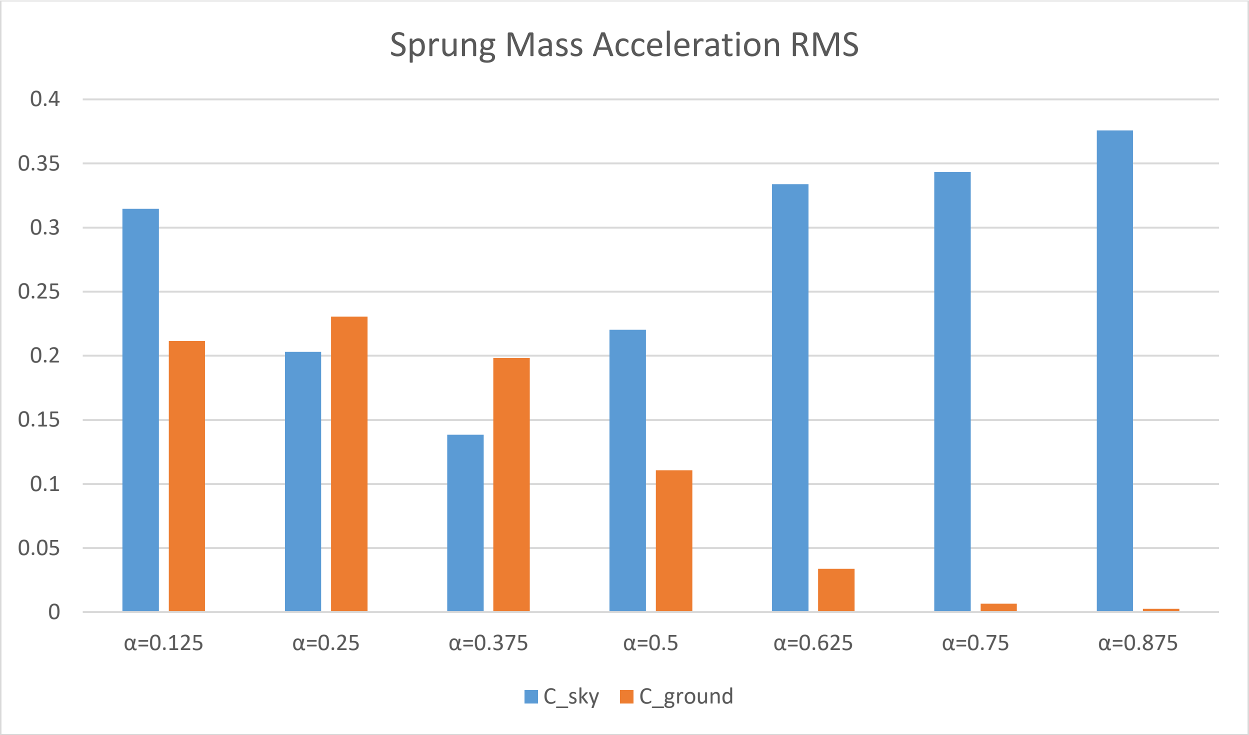


Figure 4-8 – Sprung mass acceleration RMS sensitivity to c_{sky} and c_{ground}

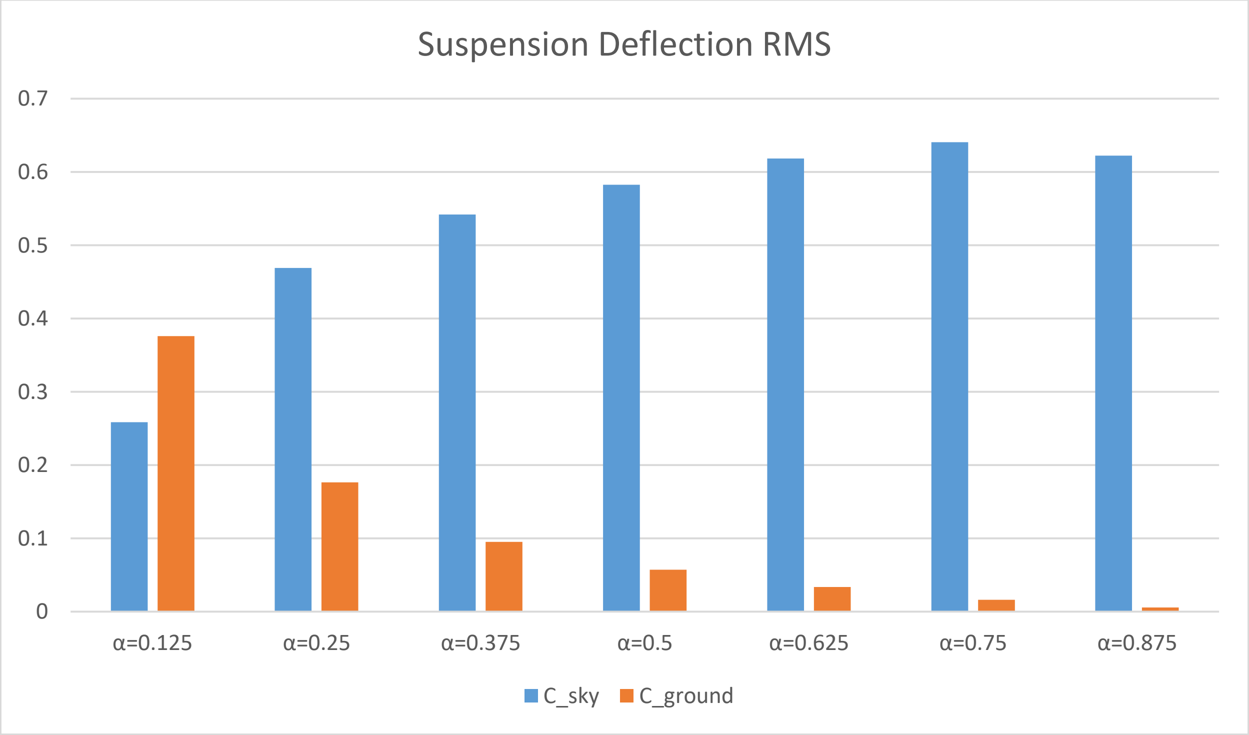


Figure 4-9 – Suspension deflection RMS sensitivity to c_{sky} and c_{ground}

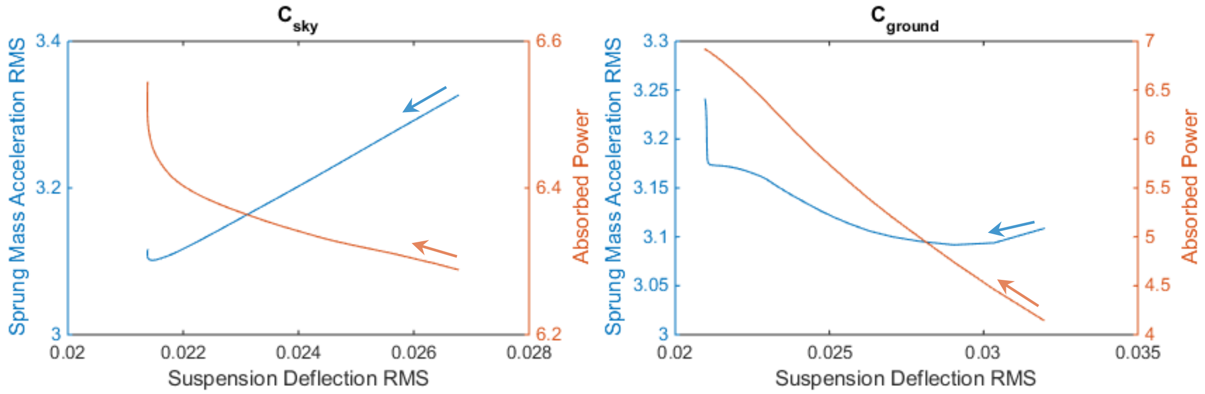
In literature, absorbed power [26] and sprung mass acceleration RMS are both considered as human comfort indices. Also, suspension deflection RMS is known as an index for road holding ride metric. As shown in Figure 4-7, Figure 4-8, and Figure 4-9, robustness of each ride metric index to c_{ground} reduces as α increases. This is due to the role of α in Eq. 23 where by increasing α, the controller tends to behave more similar to Skyhook controller rather than Groundhook.

Opposite trend is seen for c_{sky} in absorbed power and suspension deflection RMS. However, sprung mass acceleration RMS is the least sensitive to c_{sky} variation at $\alpha = 0.375$. It should be noted that this value of α, where the robustness is at its highest for c_{sky}, highly depends on the vehicle parameters.

4.2.5 Hybrid Skyhook-Groundhook Fine-tuning

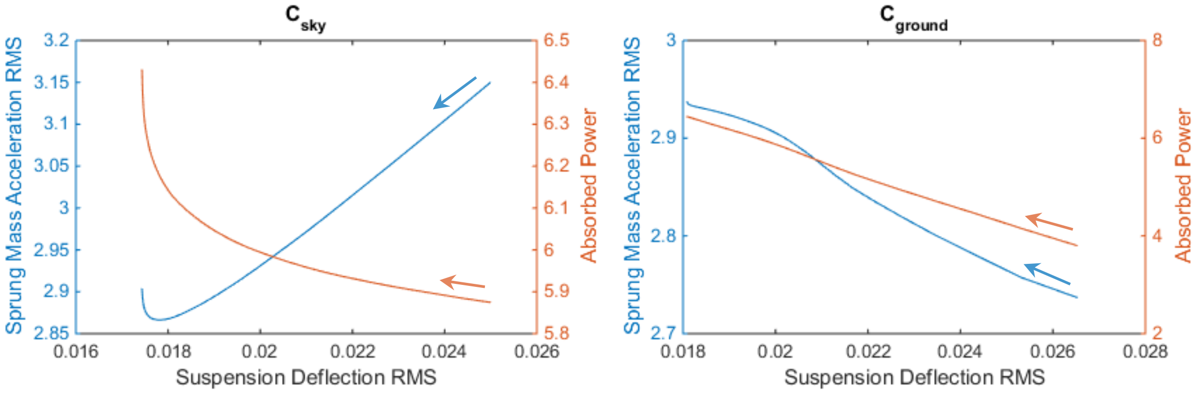
In this section, c_{sky} and c_{ground} varied within the same minimum and maximum values as in the previous section, to investigate how ride metrics are affected. A set of simulations was run for each values of parameter α (same as previous section) to study its effect on ride metrics as well, α=0.125, 0.25, 0.375, 0.5, 0.625, 0.75, and 0.875.

Both human comfort indices, Sprung Mass Acceleration RMS and Absorbed Power, were plotted versus road holding metric, Suspension Deflection RMS. It was done for both c_{sky} and c_{ground} separately at all seven values of α. Figure 4-10-Figure 4-16 show results of the aforementioned simulations. The plots demonstrate Sprung Mass Acceleration RMS and Absorbed Power versus Suspension Deflection RMS for both c_{sky} and c_{ground}. The figures marked with (a) are corresponding to the case where c_{sky} changes and c_{ground} is kept at the nominal value (100000 $^{N.s}/_{m}$). The plots indicated by (b) have the nominal value of c_{sky} at 20000 $^{N.s}/_{m}$ and c_{sky} is varying. The arrows on each curve show the direction of either c_{sky} or c_{ground} increasing, depending which variable is varying.



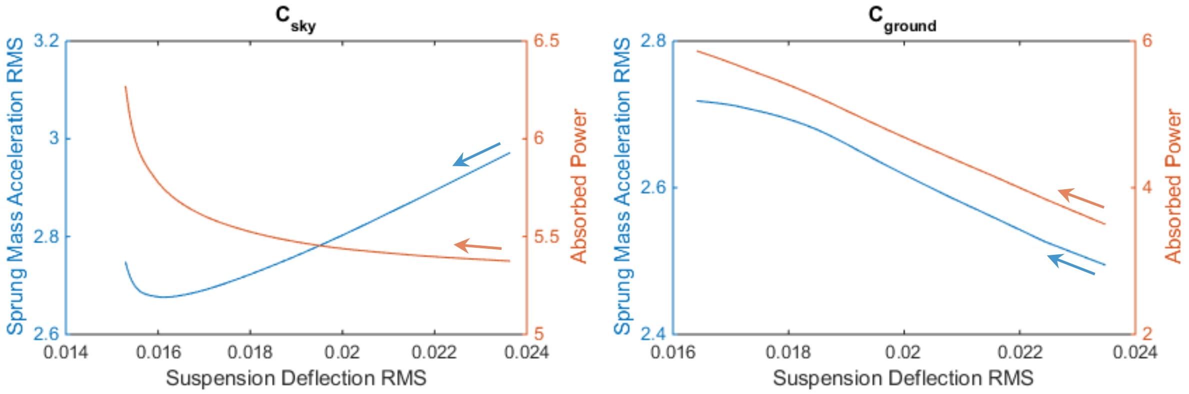
(a) (b)

Figure 4-10 - Effect of varying c_{sky} and c_{ground} at $\alpha = 0.125$ on the ride metrics



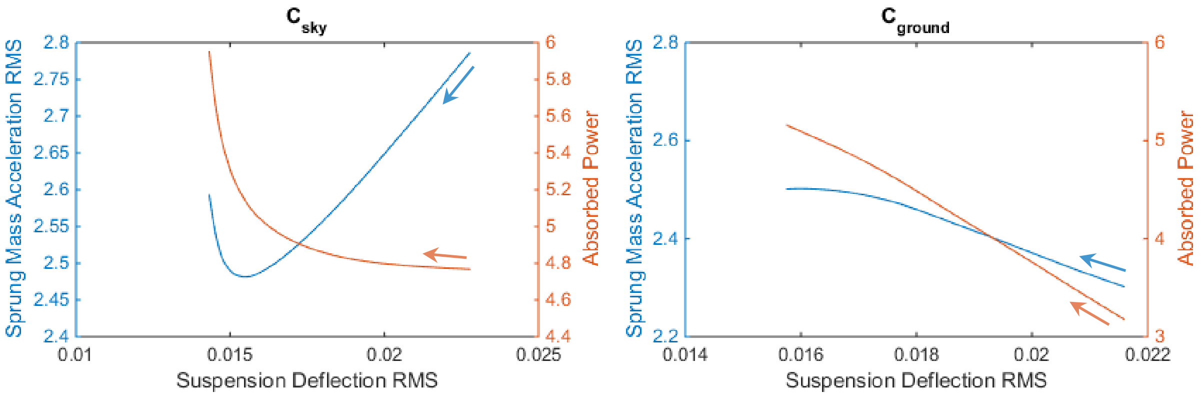
(a) (b)

Figure 4-11 - Effect of varying c_{sky} and c_{ground} at $\alpha = 0.25$ on the ride metrics



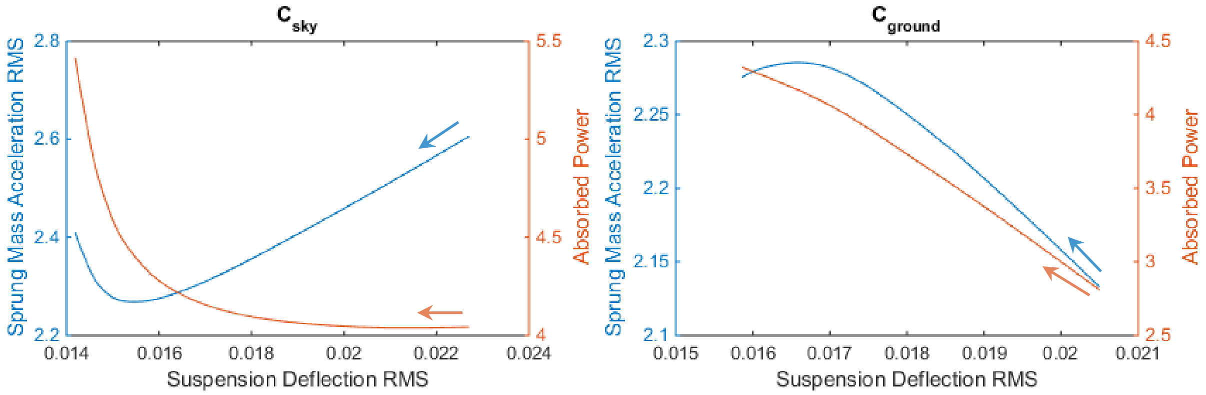
(a) (b)

Figure 4-12 - Effect of varying c_{sky} and c_{ground} at $\alpha = 0.375$ on the ride metrics



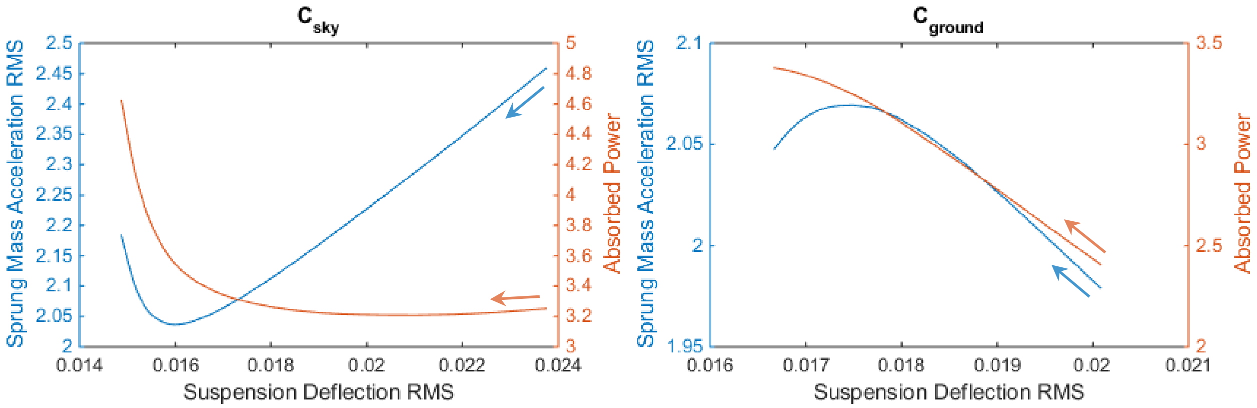
(a) (b)

Figure 4-13 - Effect of varying c_{sky} and c_{ground} at $\alpha = 0.5$ on the ride metrics



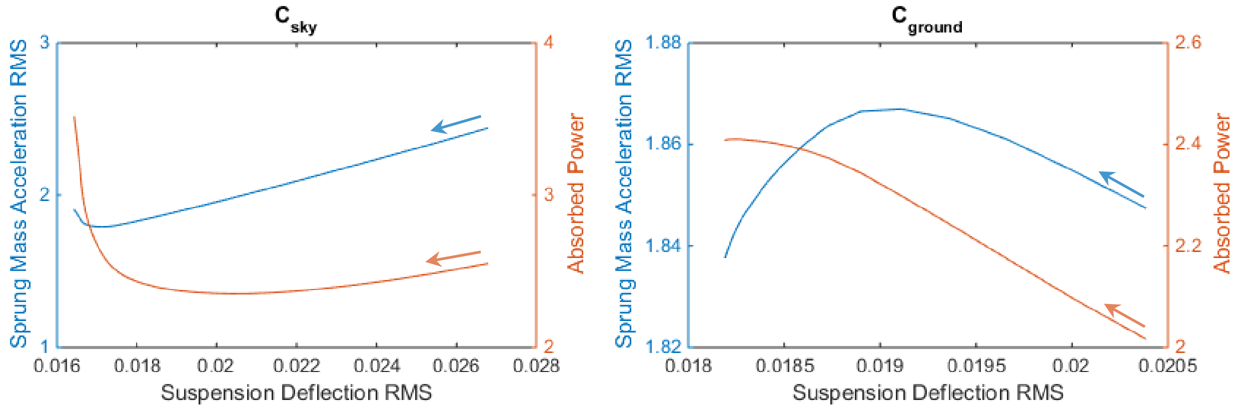
(a) (b)

Figure 4-14 - Effect of varying c_{sky} and c_{ground} at $\alpha = 0.625$ on the ride metrics



(a) (b)

Figure 4-15 - Effect of varying c_{sky} and c_{ground} at $\alpha = 0.75$ on the ride metrics



(a) (b)

Figure 4-16 - Effect of varying c_{sky} and c_{ground} at $\alpha = 0.875$ on the ride metrics

As seen in Figure 4-10 to Figure 4-16, when increasing c_{sky} (plots indicated by (a)), the road holding characteristic (Suspension Deflection RMS) enhances, so does Sprung Mass Acceleration RMS but, only up to a minimum point. After that minimum point, it starts getting worse meaning there is a an optimum point for human comfort while increasing c_{sky}. But if the human comfort is measured by Absorbed Power, the above statement would not hold. Except the case where $\alpha = 0.875$, there is no minimum point for human comfort. In other words when $\alpha = 0.125, 0.25, 0.375, 0.5, 0.625,$ or 0.75, we lose human comfort by increasing c_{sky}.

However, the difference between those two human comfort metrics is not as much when c_{ground} increases and c_{sky} is kept at its nominal value. For $\alpha = 0.625, 0.75,$ and 0.875, Sprung Mass Acceleration RMS faces a maximum value but it does not seen in Absorbed Power.

Saying above, the two human comfort metrics do not follow the same trend especially when c_{sky} changes. In other words, when tuning or evaluating a control methodology, it is very important to choose which human comfort index to investigate on, since depending which one is picked, significant difference in the final numbers will be resulted.

4.2.6 Further Tuning

Figure 4-10 to Figure 4-16 give insight on how the controller performs when one parameter is kept at it nominal value and the other one changes. Although this method is helpful to spot the trends of ride metrics but to truly find the best values of c_{sky} and c_{ground}, one needs to consider searching c_{sky}-c_{ground} space. In order to fulfill this task, a global optimization method is used in this study to find the best set of c_{sky} and c_{ground} that corresponds to optimum ride metrics. However, in order to find the best values of c_{sky} and c_{ground}, an objective function must be defined. Since the goal is to optimize both human comfort and road holding at the same time, the objective function was assumed as distance of a performance point from origin on Human Comfort Vs. Road Holding plot. Note that both metrics are positive numbers. In other words, the objective function is:

$$Objective\ Function = \sqrt{Human\ Comfort^2 + Road\ Holding^2} \qquad \text{Eq. 28.}$$

In Eq. 28, Human Comfort can be quantified as either Sprung Mass Acceleration RMS or Absorbed Power. Two sets of simulations were run, one with Sprung Mass Acceleration RMS taken as the Human Comfort and another with Absorbed Power.

Genetic Algorithm was used to minimize the objective function. Genetic Algorithm (GA) is known as a heuristic Global Optimization method. It is a non-gradient optimization where genomes are mutated over each generation according to their objective function values. Details of GA is beyond the scope of this article and one can refer to many published resources in Optimization topic. One important point to keep in mind when using GA is that it does not necessarily return the global minimum point. If the genomes get trapped in a local minima, the resultant will be the minimum point of that local minima.

Table 13 and Table 14 show the results of optimization. It should be noted that Table 13 contains the simulations where Sprung Mass acceleration RMS was taken as the Human Comfort index. On the other hand,
Table 14 corresponds to the case where Absorbed Powered was considered as Human Comfort index in Eq. 28.

Table 13 - Hybrid controller tuning with Sprung Mass Acceleration RMS taken as Human Comfort index

Alpha α	c_{sky}	c_{ground}	Suspension Deflection RMS	Sprung Mass Acceleration RMS	Absorbed Power	Objective Function
0.125	67845.15	20000.50	0.0266	2.8596	4.2820	2.8596

0.25	39723.10	26954.03	0.0210	2.6257	4.5158	2.6257
0.375	45987.11	20000	0.0187	2.3313	3.8760	2.3313
0.5	39668.37	20000	0.0177	2.1737	3.5233	2.1737
0.625	39901.30	24906.91	0.0166	2.0875	3.5104	2.0875
0.75	25930.30	470760.42	0.0160	2.0320	3.5250	2.0320
0.875	57825.96	20000	0.0176	1.7144	2.4824	1.7145

Table 14 - Hybrid controller tuning with Absorbed Power taken as Human Comfort index

Alpha α	c_{sky}	c_{ground}	Suspension Deflection RMS	Sprung Mass Acceleration RMS	Absorbed Power	Objective Function
0.125	32165.85	20000	0.0290	2.9466	4.1238	4.1230
0.25	17786.00	20000	0.0274	2.7878	3.8054	3.8054
0.37	14371.76	20000	0.0262	2.6457	3.4899	3.4900
0.5	13210.79	20000	0.0251	2.5002	3.1556	3.1557
0.62	14025.52	20000	0.0234	2.2999	2.7860	2.7861
0.75	11964.51	20000	0.0243	2.2472	2.4251	2.4252
0.87	14597.12	27968.91	0.0219	2.0023	2.1403	2.1405

As seen in tables above, for both cases Objective Function value reduces as α increases meaning the higher α is, the better controller performs at its tuned parameters, namely c_{sky} and c_{ground}. As expected, values of tuned c_{sky} and c_{ground} are quite different for the same α value but different metrics of Human Comfort. One fact to point out here is that value of c_{ground} is at its minimum for most of the tuned parameter pairs. Therefore, increasing the span space may lead to better results in terms of vehicle ride metrics.

5 H∞ Robust Control and Modified H∞ Controller

H∞ algorithm guarantees stabilization and robustness by modeling the system as an optimization problem. It is considered as a modern control technique that was developed in late 1970s – early 1980s. The name comes from the mathematical H∞ norm that represents the maximum singular value of a matrix function in Laplace space that is bounded in the right-half plane. The only drawback with this method is handling of non-linear constraints such as saturation. The controller formulation will be briefly discussed in next paragraphs.

Assume the closed loop system shown in Figure 5-1, where P is the plant, K is the controller feedback, w is the external disturbance, z is the variable to be minimized, y is the plant output, and u is control input. It should be noted that w, z, y, and u are vectors but P and K are marices.

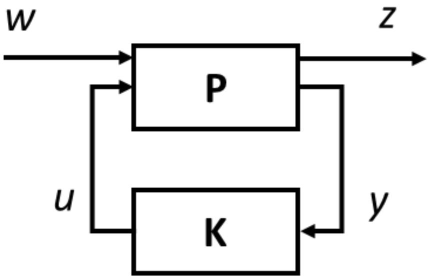


Figure 5-1 - H∞ closed loop system

In system showed above, the goal is minimizing of the error variable (z). Mathematics representation of the closed loop model is as following:

$$\begin{bmatrix} z \\ y \end{bmatrix} = P(s) \begin{bmatrix} w \\ u \end{bmatrix} \qquad \text{Eq. 29}$$

and from Figure 5-1 it can be interpreted that $u = K(s)y$. Therefore, the transfer function $F_l(P, K)$ from exogenous input w to minimized output z can be written as

$$z = F_l(P, K)w \qquad \text{Eq. 30}$$

The objective of H∞ algorithm is to find a feedback controller K that minimizes the H∞ norm of $F_l(P, K)$[14]. As a side note, the idea is very similar to H_2 control design.

In this study the plant P is a quarter-car model equipped with a semi-active suspension as described in section 2.1. Also, w is the road profile and u is the control output which is the calculated damping coefficient for the semi-active suspension. In reality, the damping value is bounded between two positive minimum and maximum values. In other words:

$$0 < c_{min} \ll c \ll c_{max} \qquad \text{Eq. } 31$$

where c_{min} and c_{max} are the minimum and maximum values that the semi-active damper can acquire, respectively.

According to Eq. 31, a saturation operator is required to clip the calculated damping value and feed it into the acceptable range. Thus, the modified schematic block diagram becomes as shown in figure below:

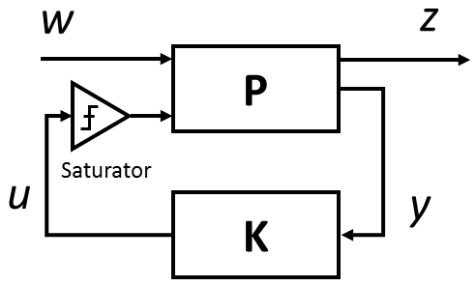


Figure 5-2 - Modified H∞ algorithm block diagram

5.1 Modified H∞ Control

As mentioned previously, H∞ cannot guarantee stability in presence of a saturation constraint. Therefore, a tweak to the above control scheme is introduced to resolve the instability issue. The proposed modification assumes a mathematical model similar to the main quarter-car model but without any constraints, and implements the H∞ control on that model. On the side, the main model is fed with the clipped control input that comes from the controller. The proposed H∞ control scheme is shown in Figure 5-3 below.

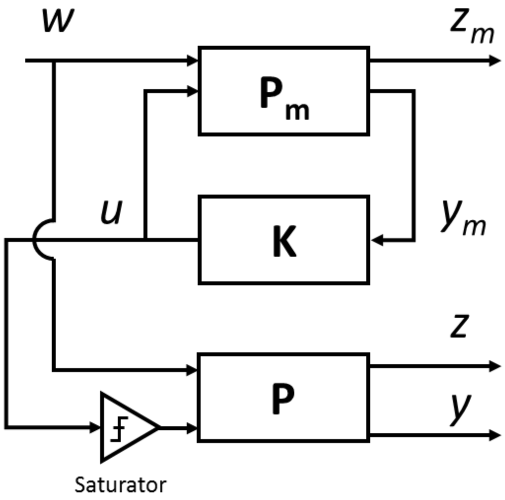


Figure 5-3 - Proposed modification for H∞ Control

In above control method, a mathematical model of the quarter-car without a saturation constraint for the semi-active damper (P_m) is controlled by feedback control according to H∞ control algorithm. Furthermore, the control input is saturated and fed to the main plant P (with saturation constraint). Using introduced method, the system does not involve any instability issue although the performance of the main plant with saturation constraint is compromised. It is a sacrifice of performance to resolve singularity issue of the closed loop system.

5.2 ADD, PDD, Modified H∞, and Passive Cases

5.2.1 Quarter-car simulation

In this section, the proposed modified H∞ is compared to ADD, PDD, and a passive case. As discussed in sections 4 and 5.1, H∞ can potentially cause singularity issue if the control input is saturated but a solution is to create another model of the plan without saturated input and design the controller for that model. The main plant works in parallel with the synthesized model and is fed with the same control input. However, the control input that is fed to the main model passes through a saturation operator. This way the system does not become singular and the controller and the plant work flawlessly.

The control algorithms discussed above was implemented on a MATLAB/Simulink quarter-car model. The quarter-car parameters were selected to represent a heavy truck. The sprung mass was assumed to be $m_s = 2250kg$, unsprung mass was assumed to be $m_u = 200kg$, the minimum damping coefficient was $c = 2000\, {}^{N.s}/_{m}$, the maximum damping coefficient was $c = 40000\, {}^{N.s}/_{m}$, suspension stiffness was $k_s = 180000\, {}^{N}/_{m}$, and the tire stiffness assumed to be $k_t = 500000\, {}^{N}/_{m}$. Also, a passive system was considered for comparing the semi-active suspension algorithms with a passive suspension case. The passive damper value was assumed $c = 5000\, {}^{N.s}/_{m}$. Also, the road-profile shown in Figure 5-4 was used.

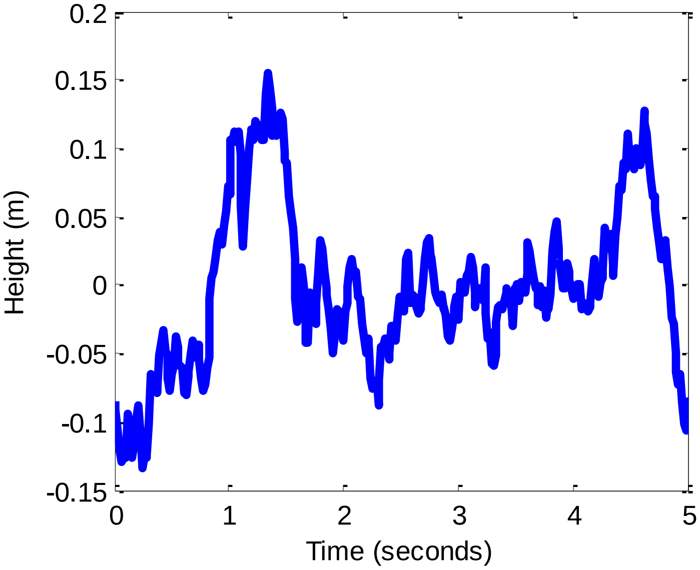


Figure 5-4 - Road profile used in 5.2

The quarter-car simulation model was implemented using MATLAB/Simulink and control algorithms were implemented using the same tool. Figure 5-5 shows the sprung mass displacement vs. time for the passive system and the three algorithms discussed above.

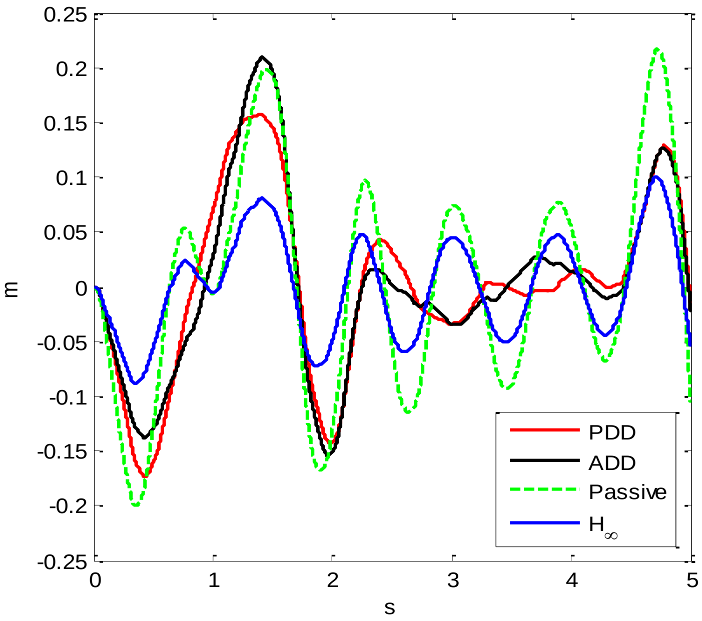


Figure 5-5 – Sprung mass displacement for different control approaches.

As shown in Figure 5-5, PDD and ADD methods keep almost the same curve, but PDD has smaller magnitudes in some times. Moreover, H_∞ shows lower magnitude of sprung mass displacement compared to the other three. PDD and ADD show some chattering effect that has a negative impact on vertical acceleration of sprung mass. Note that, as mentioned before, PDD reduces chattering effect in control input (i.e. damping coefficient), not sprung mass displacement.

Figure 5-6 shows the acceleration plot of each of the cases. The plots are separated for better visual display. Passive case has the highest acceleration magnitude as expected. H_∞ has the lowest perturbations of acceleration. ADD and PDD are similar but the advantage of PDD is having zero acceleration in some time intervals. This can be realized from the PDD control law equation (6). When the *otherwise* condition holds, the spring force applied on the sprung mass is neutralized by the damper force. In other words, the total amount force applied on the sprung mass is zero and consequently, the acceleration becomes as well.

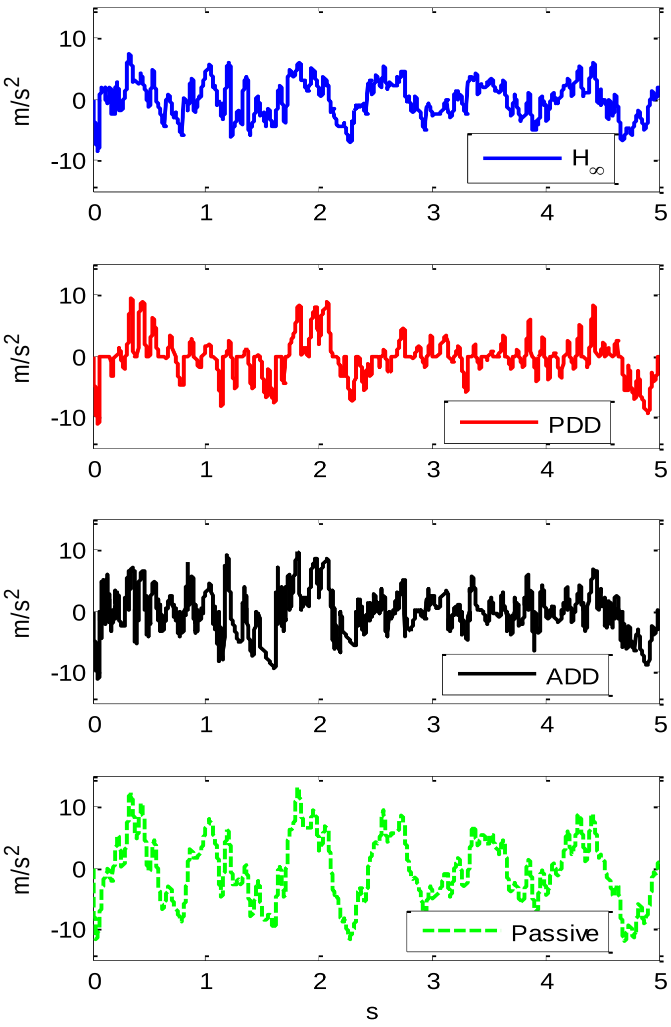


Figure 5-6 - Sprung mass acceleration for different control algorithms

To study performance of each method the absorbed power of each case was calculated over the 5 seconds simulation time. As shown in Figure 5-7, passive case builds up the highest absorbed power. ADD and PDD are showing similar behavior up to 1s and after that PDD is settling down to a lower value. Also, H$_\infty$ has the lowest absorbed power compared to the other cases.

To investigate human-comfort metric of each approach, quantified values are required along with the shown graphs. For acceleration of sprung mass, root mean square (RMS) values of vertical acceleration is widely used in the literature as the human comfort metric. Absorbed power is another quantity that expresses human-comfort metric. In this study, a procedure was used that averages the absorbed power from the initial moment till present. Thus, the final absorbed power, over the course of 5 seconds simulation, can be realized better from the final data points of graphs in Figure 5-7.

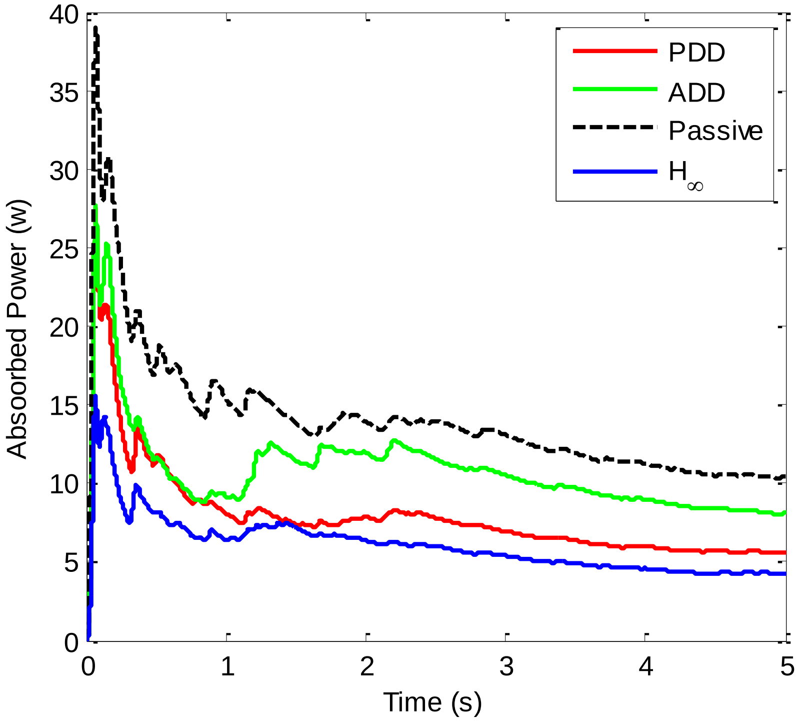


Figure 5-7 - Absorbed power of different control algorithms

In order to have a quantified value of the absorbed power, the average of its values over the last 0.5s of simulation was calculated and shown in Table 15.

Table 15 – RMS values of sprung mass acceleration and absorbed power (average of over last 0.5s)

Method	**RMS($\ddot{x}_s$)**	**Absorbed Power**
ADD	3.764	8.2293
PDD	3.3885	5.6376
H∞	3.0487	4.3078
Passive	5.5165	10.4842

According to the results summarized in Table 15, H∞ is by far the best approach. Roughly speaking, it out performs the second best method, PDD, by a 10% improvement on sprung mass acceleration RMS and 23% improvement on absorbed power. Note that the absorbed power mentioned in the table, is the average of values shown in the Figure 5-7 over last 0.5s of the simulation

As a modern method, H∞ needs more computational time and its implementation is more complicated compared to the other approaches. Note that this simulation has been performed only for a quarter-car model. The computation load will be much more for a full-vehicle where at least 4 suspensions are to be controlled simultaneously.

ADD has the worse RMS value of sprung mass acceleration than PDD does. This fact also can be realized from the sprung mass displacement curve, Figure 5-5. The corresponding curve to ADD has slightly more chattering effect and that leads to higher RMS value of acceleration, which is the second derivative of displacement with respect to time.

PDD performs better than ADD and its low-chattering control input is considered as an advantage over ADD. In real-time applications, too much chattering in input signal cannot be performed by the actuators with relatively smaller bandwidths.

In section 5.2 a quarter-car model was used to implement the aforementioned approaches. To compare the capability of those methods, further investigation on a full-car model is required. ADD and PDD are corner independent approaches but on the other hand, H∞ controls the whole model as one system. In other words, for a full-vehicle model with “n” corners/suspensions, “n” ADD/PDD controllers are needed whereas only one H∞ controller is required to control the ride quality. Saying above, pitch and roll angles are taken care of implicitly in H∞ but it is not the case for ADD/PDD.

5.2.2 6-axle vehicle simulation

In this section, results of implementing the controllers on a 6-axle car model (12 corners/suspensions) will be presented. A simulation in MATLAB/Simulink environment was performed using below parameters. The vehicle assumed to have 9000kg of sprung mass ($m_s = 9000kg$), each unsprung mass is $m_u = 200kg$, the minimum damping coefficient was $c = 2000 \, {}^{N.s}/_{m}$, the maximum damping coefficient was $c = 40000 \, {}^{N.s}/_{m}$, suspension stiffness for the front axle was $k_s = 130000 \, {}^{N}/_{m}$and it was $k_s = 180000 \, {}^{N}/_{m}$ for the rest of the axles, and the tire stiffness assumed to be $k_t = 500000\,{}^{N}\!/_{m}$ for all the axles. The distances between the second, third, fourth, fifth, and sixth axles to the front axle are 1, 1.9, 2.4, 3, and 3.6 meters, respectively. Also, the center of gravity is located 1.7m away from the front axle. The track width is 2m and the center of gravity is assumed to be at the middle of the vehicle i.e. 1m away from both left and right suspensions. A schematic view of the vehicle is shown in Figure 5-8.

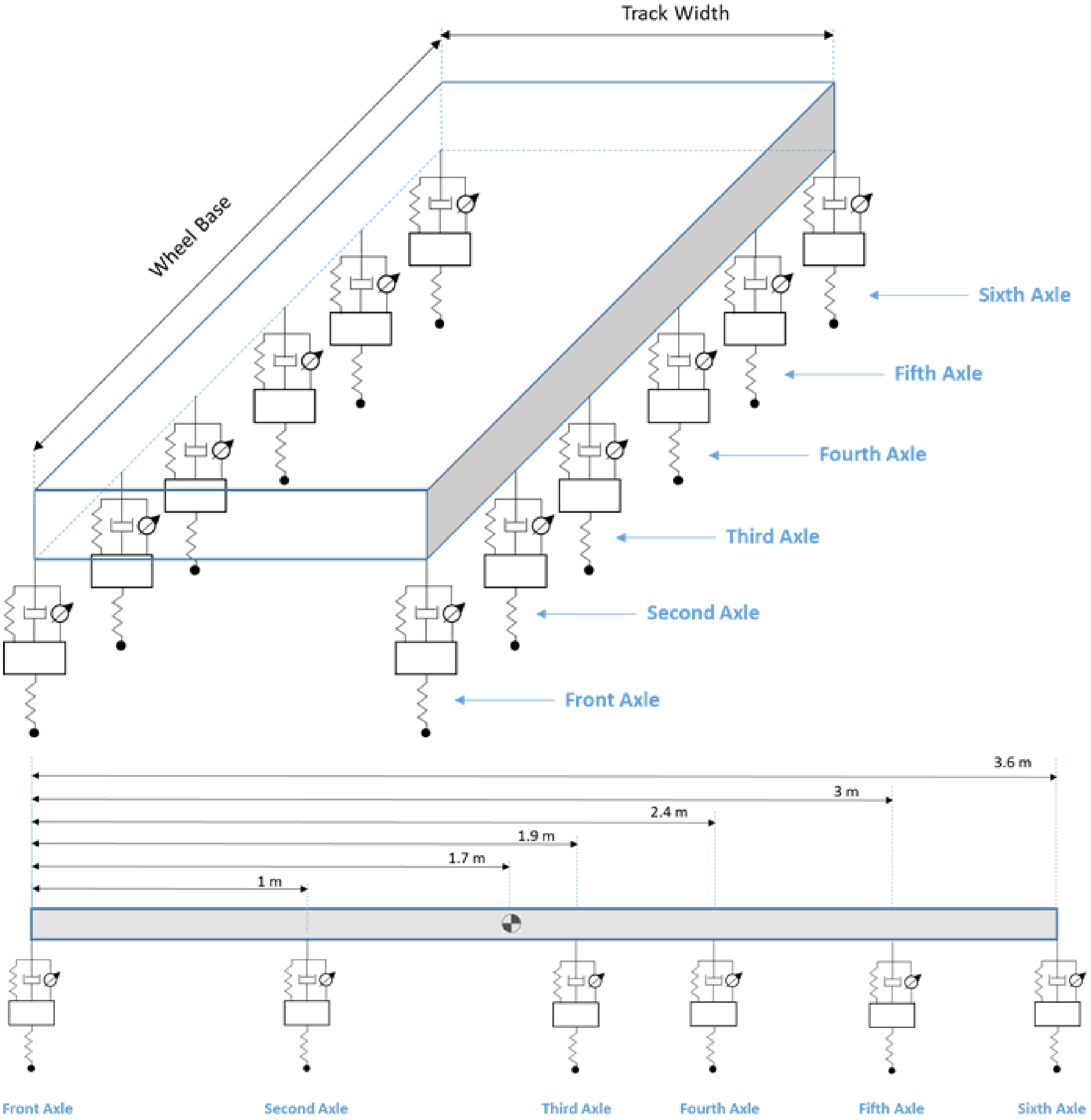


Figure 5-8 - Schematic view of a full-car with six axles

Below road profiles (left and right wheels road profiles) were used in the simulation.

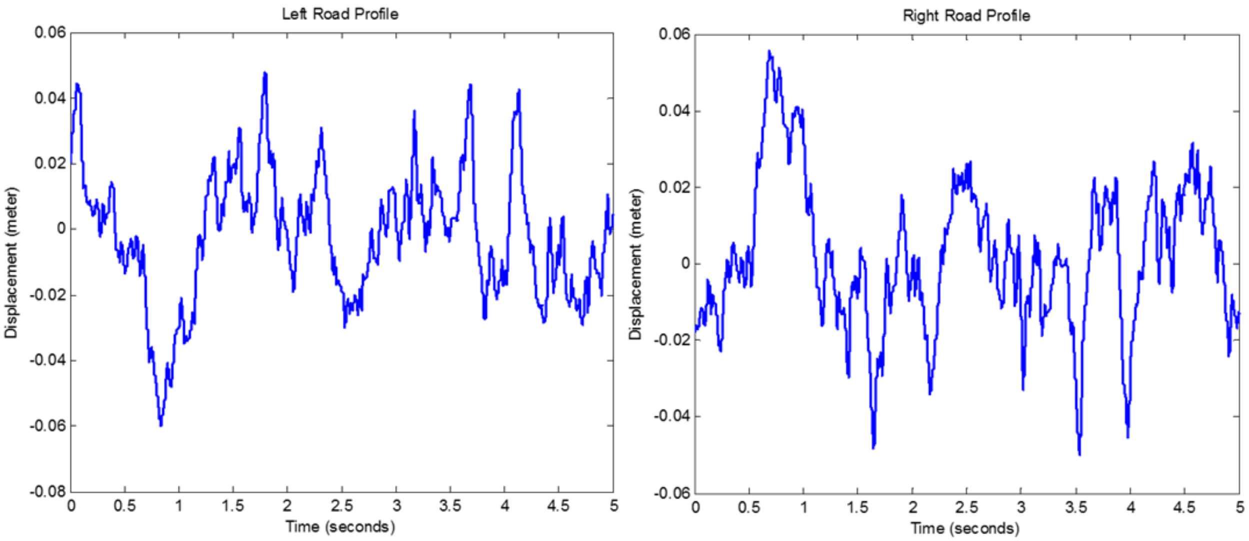


Figure 5-9 - Left and right road profiles used for the 6-axle vehicle model simulation

The three controllers were implemented on the 6-axle model and a set of simulations were run to compare their performances. Figure 5-5 show the center of gravity displacement.

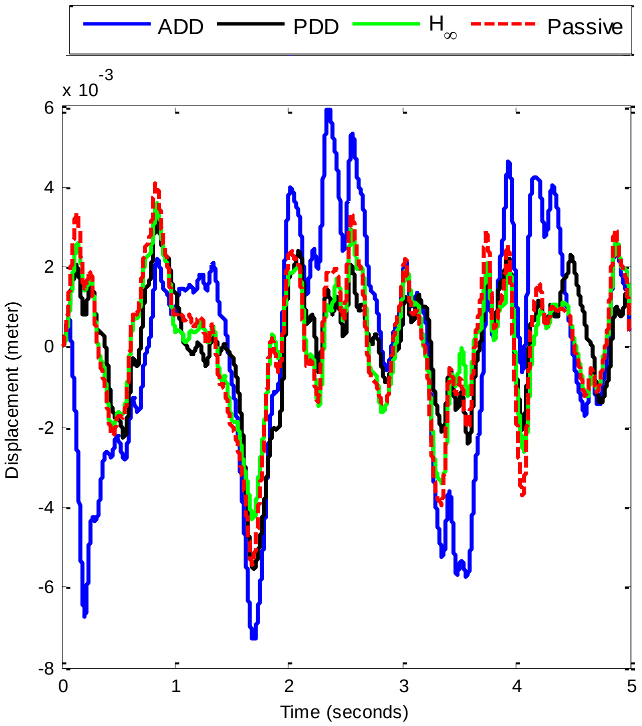


Figure 5-10 - 6-axle vehicle center of gravity displacement

Visually speaking, ADD has the highest amplitude of center of gravity displacement, PDD and H_∞ have lowest amplitudes and passive is in between. Since center of gravity displacement does not show the ride quality, acceleration plots are needed for further investigations.

Acceleration plots are shown in Figure 5-6. Again, PDD and H_∞ have lowest acceleration amplitudes and ADD has the highest spikes. In other words, human-comfort index (acceleration RMS) is expected to be highest for ADD. Passive suspension is working better than ADD in the 6-axle model and that is because ADD controls each corner independently and does not take care of the roll and pitch angles. PDD controls each corner separately too according to the acceleration plots its control law is good enough for a full-car model. Accurate comparison of PDD and H_∞ is not possible yet as more information such as absorbed power is needed.

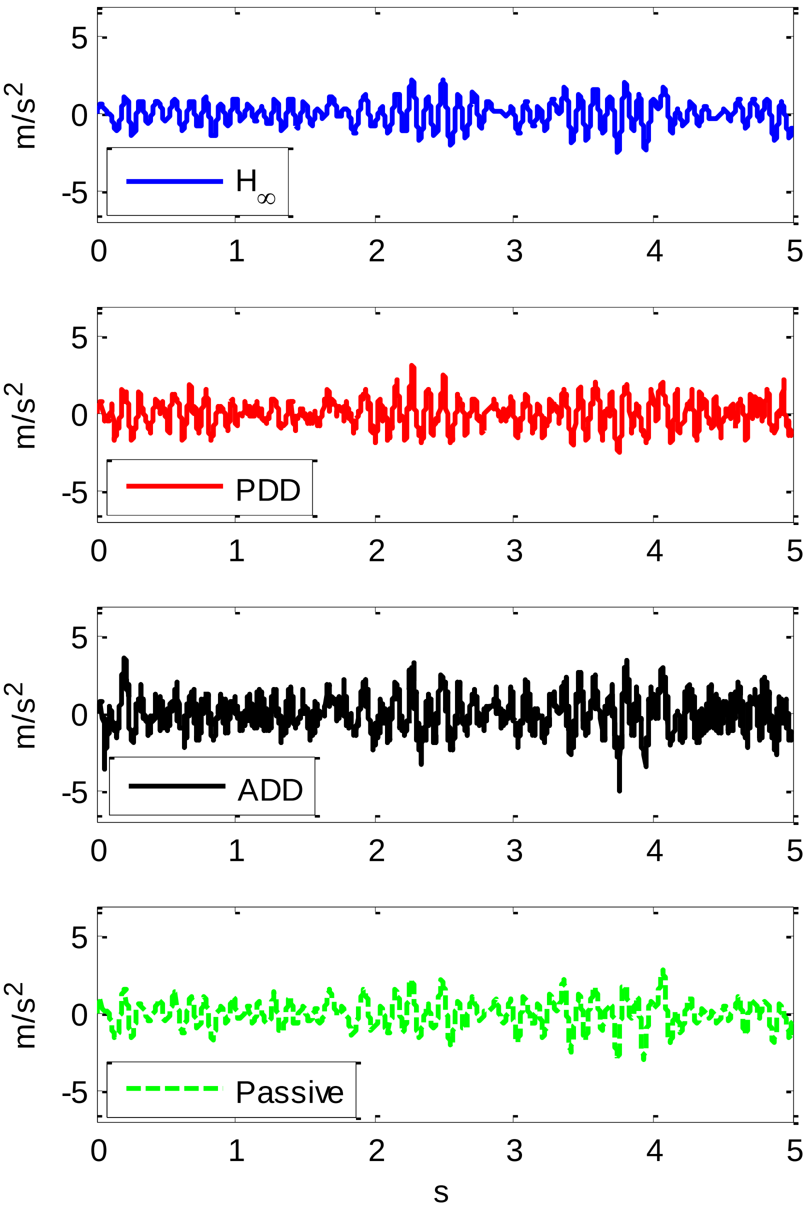


Figure 5-11- Center of gravity acceleration for 6-axle vehicle model

Figure 5-12 shows the absorbed power plots for the controllers. As seen, ADD builds up the highest amount of absorbed power over the 5-second simulation. H∞ has best performance in terms of absorbed power. Moreover, PDD is slightly worse than H∞ and passive has the third performance and is better than ADD.

As discussed previously, since ADD is a corner independent method, it only controls the acceleration of the point that it is connected to the sprung mass. It is a drawback of that concept because dynamics of all the corners are interconnected to each other by pitch and roll effects and performing independently from the other corners reduces chance of performing as well as expected. PDD has the same issue but apparently its concept of reducing the absorbed power as mentioned in 3.1.5 increases its efficiency.

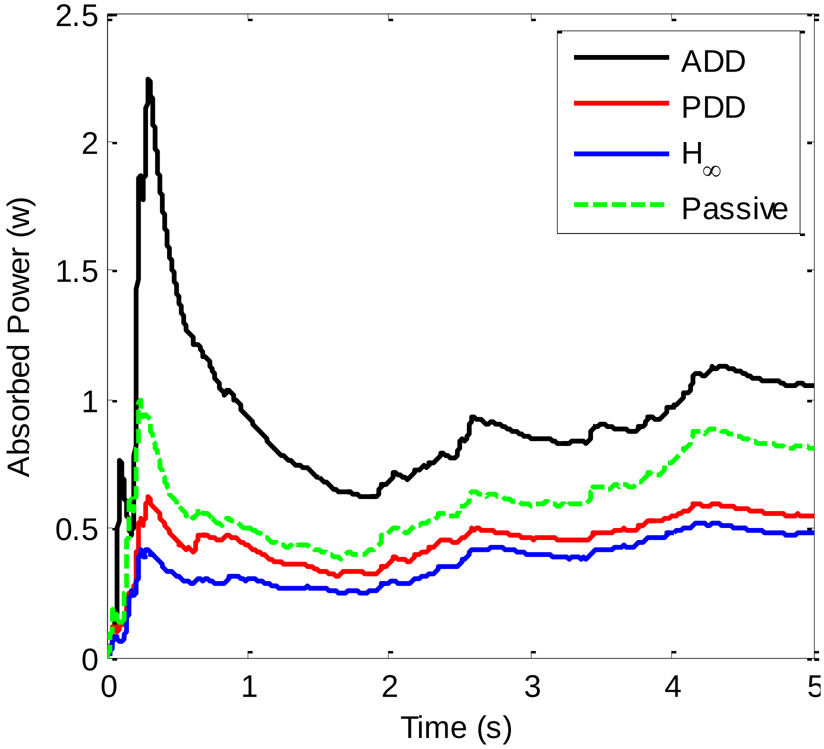


Figure 5-12 - Absorbed power for 6-axle vehicle model

The same as quarter-car model, a table of quantified values of center of gravity acceleration and absorbed power plots provided below. Once again, the absorbed power value in the table is the average of last 0.5s values from the graph.

Table 16 - RMS values of sprung mass acceleration and absorbed power (average of over last 0.5s)

Method	**RMS($\ddot{x}_s$)**	**Absorbed Power**
ADD	1.1208	1.0701
PDD	0.8836	0.5585
H_∞	0.8319	0.4841
Passive	0.9212	0.8246

In this case, H_∞ has improved the RMS of acceleration by 6% compared to the second best approach PDD. Moreover, the absorbed power of H_∞ has been improved by 13% with respect to PDD.

Studying of vehicle corners, and roll and pitch effects can be performed as the future of works of this investigation. Also, using different road profiles would give a better understanding of how each method works.

6 Fuzzy Controller and Inverse ANFIS

6.1 Fuzzy Logic

Fuzzy logic is developed based on Boolean logic by Lotfi Zadeh in 1965 [27] considering the mathematical concept of fuzzy sets, which is a more general concept of classical sets. Providing an environment that a state can be some value other than true or false, fuzzy has an extensive flexibility for logical procedures and reasoning that allows taking into account uncertainties and small perturbations.

Linguistic variables are another advantage of fuzzy logic that allows human level reasoning being possible for implantation in mathematics world. For instance, here is a rule that a driver follows when driving: If the light is red and if my speed is high and if the light is close, then I brake hard.

6.1.1 Fuzzy Sets

Fuzzy logic is defined on the theory of fuzzy sets, which is a generalization of the classical set theory. Saying that the theory of fuzzy sets is a generalization of the classical set theory means that the latter is a special case of fuzzy sets theory. To make a metaphor in set theory speaking, the classical set theory is a subset of the theory of fuzzy sets, as shown in

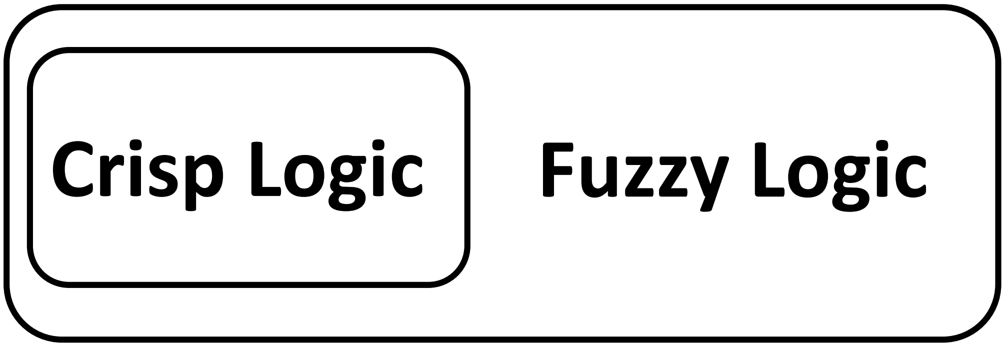


Figure 6-1 - Classical logic theory is a subset of Fuzzy logic

In this dissertation, Boolean logic will be called classical logic for the sake of convenience.

Membership function determines degree of membership in a fuzzy set. Figure 6-2 shows the membership function chosen to determine degree of membership for “good” quality of service.

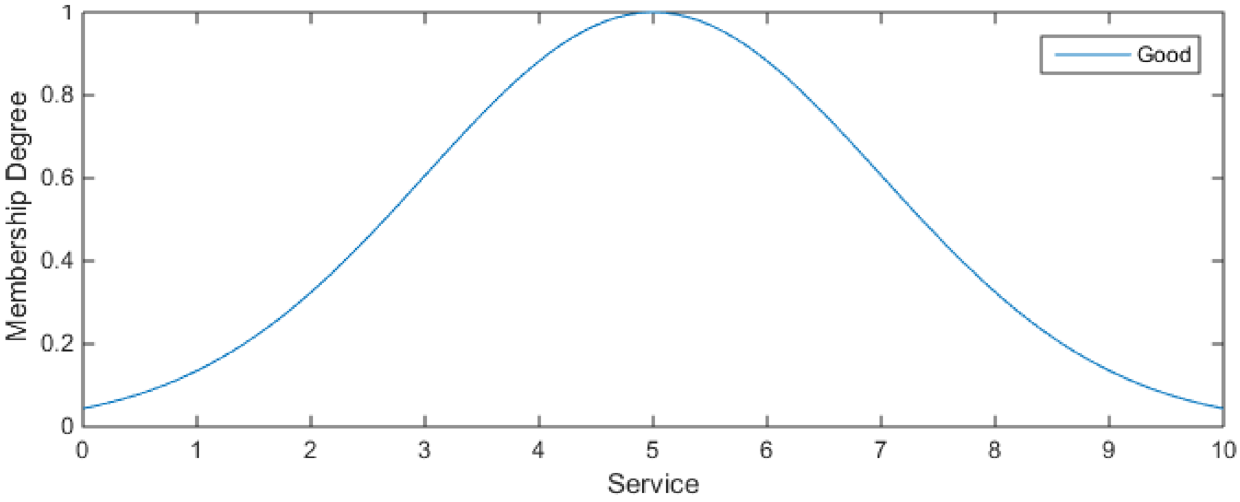
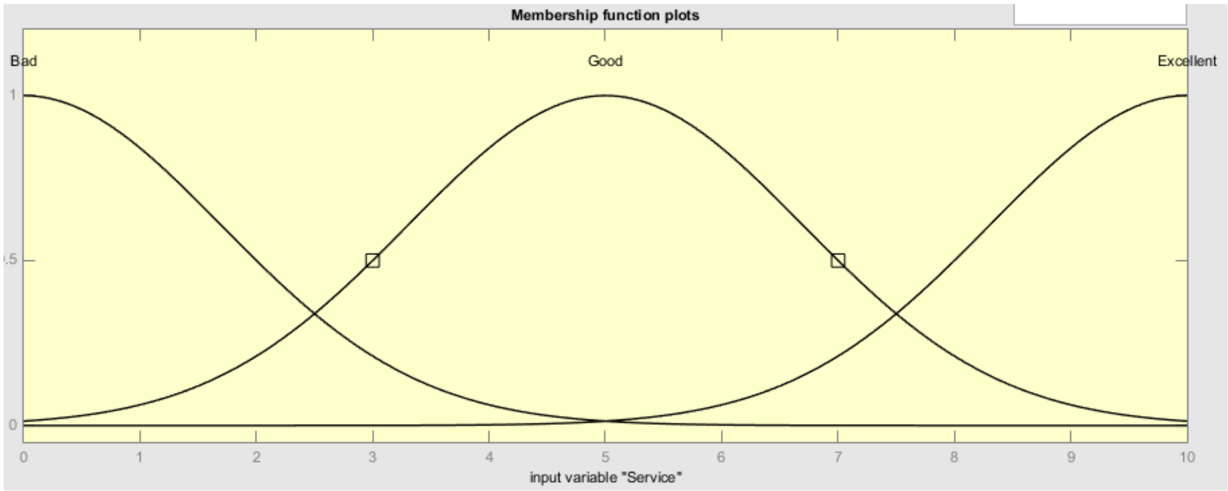


Figure 6-2 - Membership function for "good service quality"

The shape of the membership function is defined based on the advice of the knowledgeable expert or by statistical studies: sigmoid, hyperbolic, tangent, exponential, Gaussian or any other form are usually used.

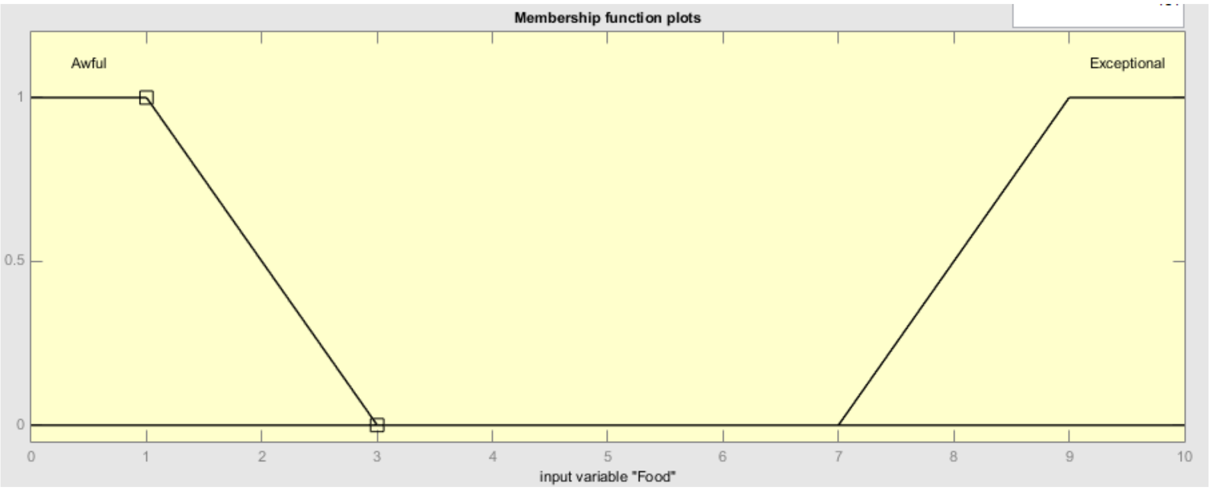
6.1.2 The linguistic variables

The concept discussed above lets us define fuzzy systems in human language easily and without loss of accuracy, since the membership function relates fuzzy logic with human language that will be defined in following. Assume V is a linguistic (quality of service, quality of food, taste of food, etc.), X the interval where variable V can vary within and TV is a set of fuzzy sets. A linguistic variable is defined based on the triplet (V, X, TV). Figure 6-3, Figure 6-4, and Figure 6-5 show 3 linguistic variables quality of service, quality of food, and tip amount, respectively. As shown, the shape and position of the functions are arbitrary and up to the expert's knowledge.



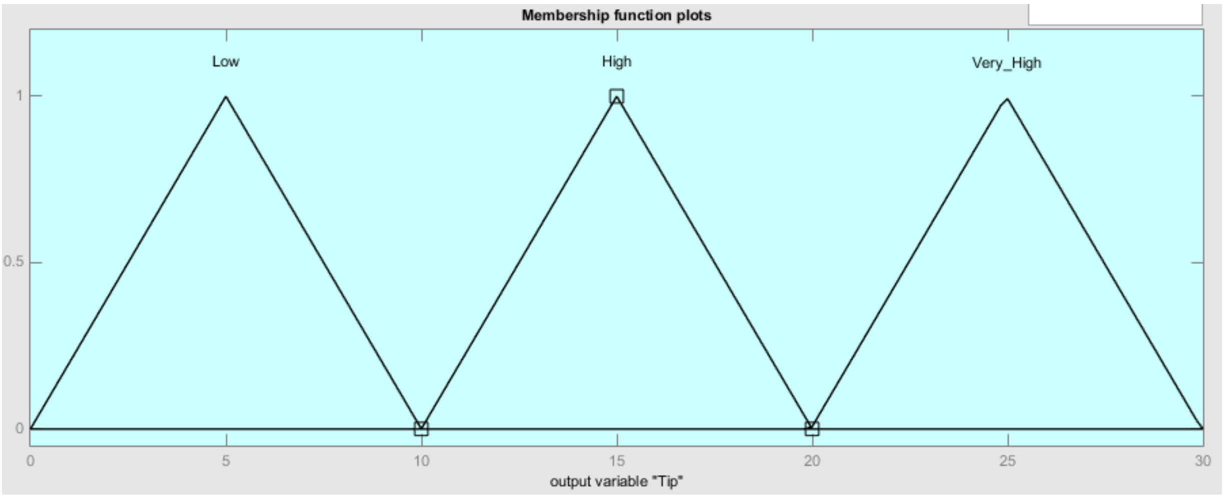
$$V = \mathit{service}$$
$$X = \Re^+$$
$$T_V = \{\ \underline{\text{Poor}}\ , \underline{\text{Good}}\ , \underline{\mathit{excellent}}\}$$

Figure 6-3 - Linguistic variable “quality of service”



$$V = \text{Quality of food}$$
$$X = \Re^+$$
$$T_V = \{\ \underline{\text{Awful}}\ ,\ \underline{\text{Delicious}}\ \}$$

Figure 6-4 - Linguistic variable “quality of food”



$$
\begin{aligned}
V &= \text{Tip}\\
X &= \Re^{+}\\
T_V &= \{\ \underline{\text{Low}}\ ,\underline{\text{Medium}}\ ,\underline{\text{High}}\ \}
\end{aligned}
$$

Figure 6-5 - Linguistic variable “tip amount”

6.1.3 The fuzzy operators

To easily work with fuzzy sets, we need to redefine the operators in classical sets theory in fuzzy logic space to correlate membership functions of fuzzy logic for values within 0 and 1. Although the definitions of fuzzy sets properties are always the same, definition for operators of fuzzy sets is determined by the expert such as membership functions.

6.1.4 Reasoning in fuzzy logic

In classical logic, the conditions are in form of: *If p then q, p true then q true.*

In fuzzy logic, fuzzy reasoning (approximate reasoning) is expressed by fuzzy rules in human language using linguistic variables that was previously defined in previous sections of this dissertation. Fuzzy rules have the form: If a ∈ A and b ∈ B then c ∈ C, with A, B and C fuzzy sets. For example: “If (the quality of the food is ‘exceptional’), then (tip is ‘very high’)”.

The linguistic variable ’tip’ is a member of fuzzy set “very high” and the degree of its membership is determined by membership function, i.e. the membership degree of the variable “food quality” to the fuzzy set “exceptional“. Idea behind fuzzy rules are that the more

In order to come up with the final output and degree of “Very high tip” more input actions have to be checked and it requires more fuzzy rules and an operation called implication. Fuzzy implications determines the degree of membership for output variable through is series of fuzzy operations.

Fuzzy implication is chosen by the designer (expert) and there is no strict definition of the operation but there is a wide choice of options and one definition is usually taken. Similar to the fuzzy operators AND, OR and NOT, the proposition of a fuzzy rule can be defined based on a mixture of other fuzzy propositions. A table consisting of all rules of a fuzzy system is called fuzzy decision matrix. The decision matrix for our tip instance is shown below:

Table 17 - Fuzzy logic decision matrix for the tip example

1	If the service is bad or the food is awful	then the tip is low
2	If the service is good	then the tip is high
3	If the service is excellent or the food is exceptional	then the tip is very high

Figure 6-6 shows the implication of rule 2.

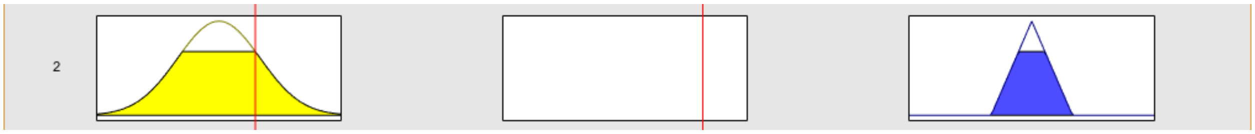


Figure 6-6 – Fuzzy Implication for rule 1

Figure 6-7 shows what we output as this fuzzy rule “If (the service is excellent or the food is exceptional), then (tip is very high)” where the quality of service is rated 6.5 out of 10 and the quality of food 8.2 out of 10 if we choose the Mamdani implication and the translation of OR by MAX.

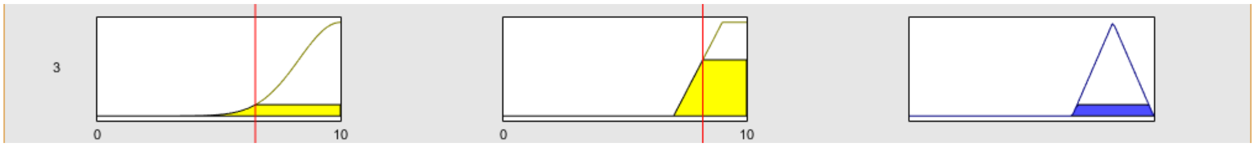


Figure 6-7 - Fuzzy Implication for rule 3

Now we go our decision matrix and use that for implication. But, fuzzy sets need to be defined first before taking any other action: the summation is done by utilizing the operator MAX. Operator MAX is usually used for aggregation. The Figure 6-8 shows this aggregation.

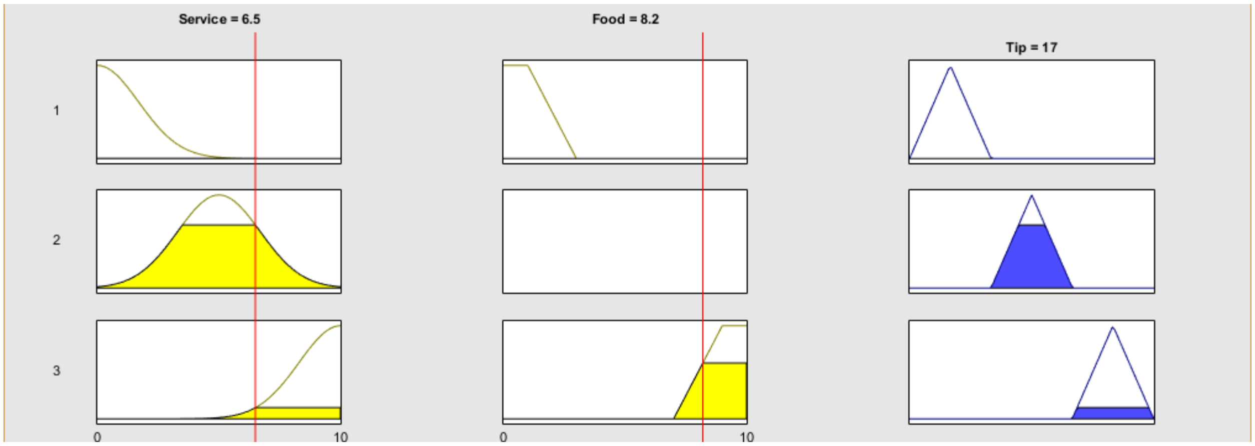


Figure 6-8 - Fuzzy Implication using the decision matrix

Now a procedure has to be defined to finalize the decision and output the final value of tip. This is done by knowing the value of linguistic variables and the implication operation. This step is called the defuzzification where a fuzzy variable is converted to a crisp variable i.e. fuzzy value to crisp value transformation.

6.1.5 Defuzzification

Similar to other steps of fuzzy inference, the expert has to pick from a variety of defuzzification operations. More information regarding this step can be found in [28]. Two main methods defuzzification are explained here: the mean of maxima (MeOM) and center of gravity (COG).

The MeOM defuzzification takes the average of fuzzy set resulted by the aggregation procedure and outputs that value as the final value of the fuzzy inference.

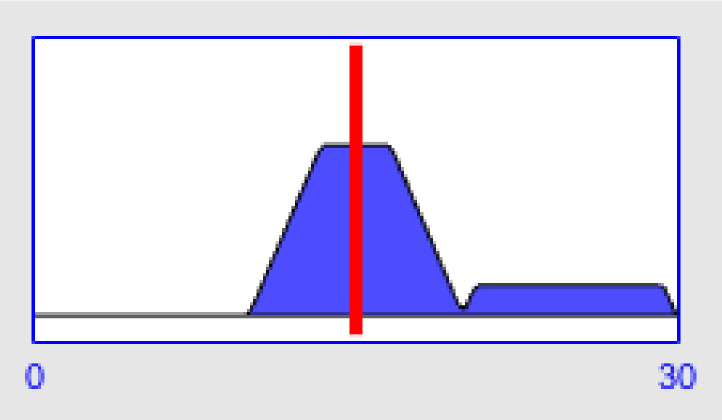


Figure 6-9 - Defuzzification with the method of the mean of maxima (MeOM)

The COG defuzzification is mostly used compared to MeOM. It calculates the center of gravity of the membership functions fuzzy set resulted by the aggregation of fuzzy implications.

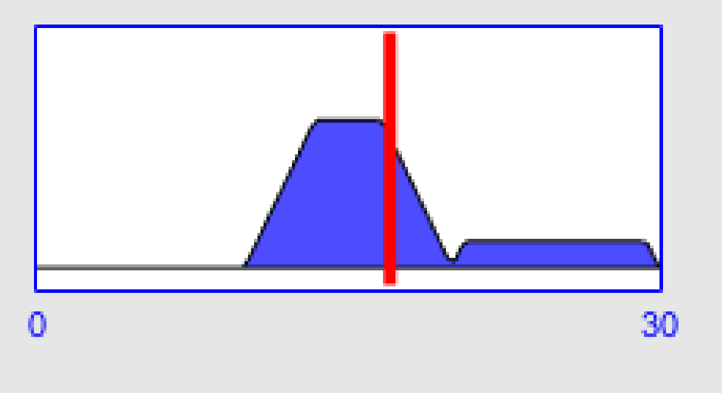


Figure 6-10 - Defuzzification with the method of center of gravity (COG)

COG does not have the discontinuities show up in MeOM defuzzification however, com[plexity is much higher and the procedure becomes less computationally effective. Some work as [29] seek to improve performance by searching other methods as effective but with a lower computational complexity. As we see in the two figures showing the MeOM and COG defuzzifications applied to our example, the choice of this method can have a significant effect on the final decision.

6.2 Semi-active Suspension Fuzzy Logic Controller (FLC)

Now that the fuzzy logic and fuzzy inference systems have been briefly explained, one can design a controller based on the theory of fuzzy systems. In order to do so, a set of inputs, membership functions for each input, fuzzy rules, and membership functions for the output of the controller must be defined by an expert.

This approach has been used to develop FLC's for semi-active suspensions by researchers. The choice of inputs and membership function along by reasoning rules can affect the performance of the controller and not every FLC is guaranteed to improve the performance of the semi-active suspension. That is the reason that an expert in the field is needed to choose the inputs, membership functions, and rules properly based on the expert's prior knowledge.

In this section, an approach is introduced to design an FLC for semi-active suspension to improve ride metrics of the vehicle.

6.2.1 Quarter-car Fuzzy Logic Controller for Semi-active suspensions

A quarter-car model was studied as described in section 2.1 and the parameters used are the same as the ones in section 5.2.1. A FLC was designed with two inputs, suspension deflection and sprung mass velocity. Each input had three triangle membership functions and the output has five triangle membership functions assigned to. The parameters of all those membership functions were determined using a global optimization process. The initial shape and position of membership functions for inputs as well as the output are shown in Figure 6-11, Figure 6-12, and Figure 6-13.

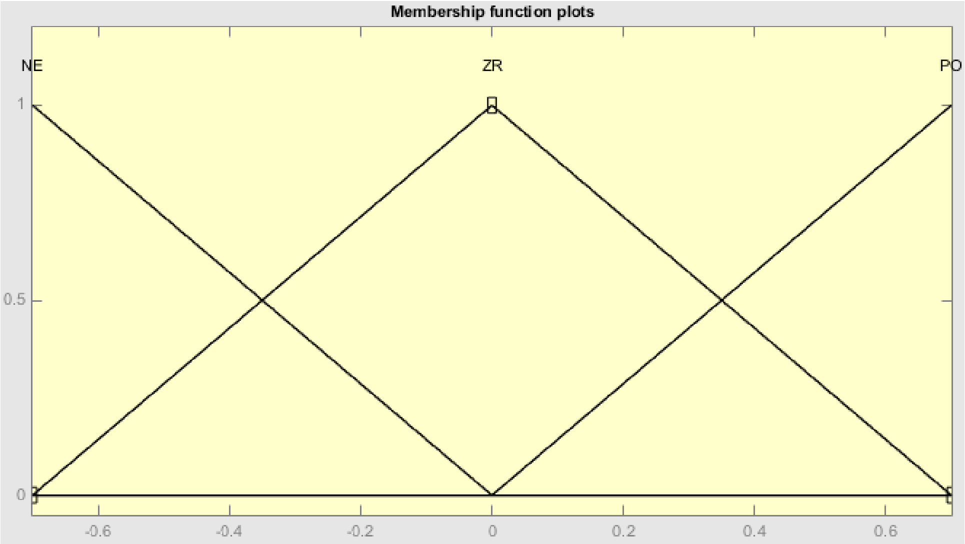


Figure 6-11 - Suspension deflection membership functions (input)

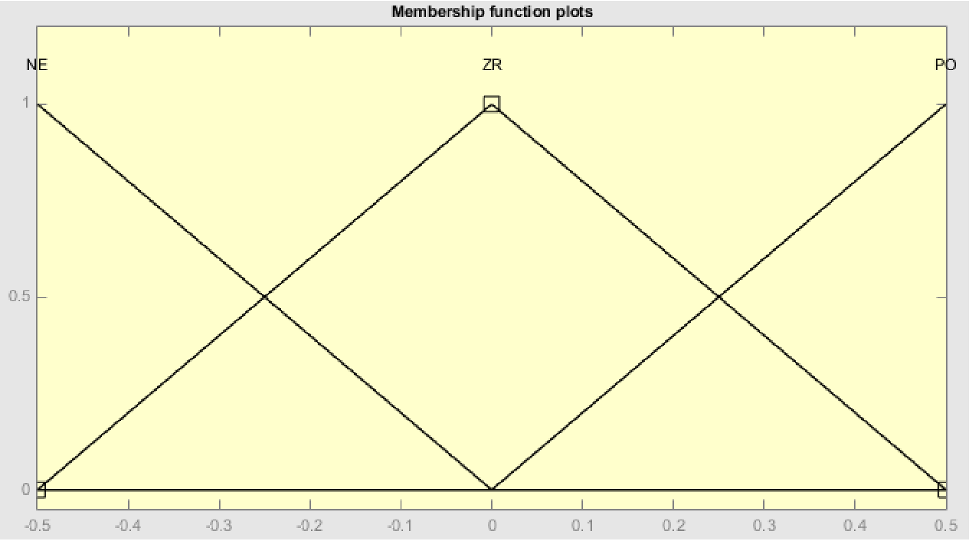


Figure 6-12 – Sprung mass velocity membership functions (input)

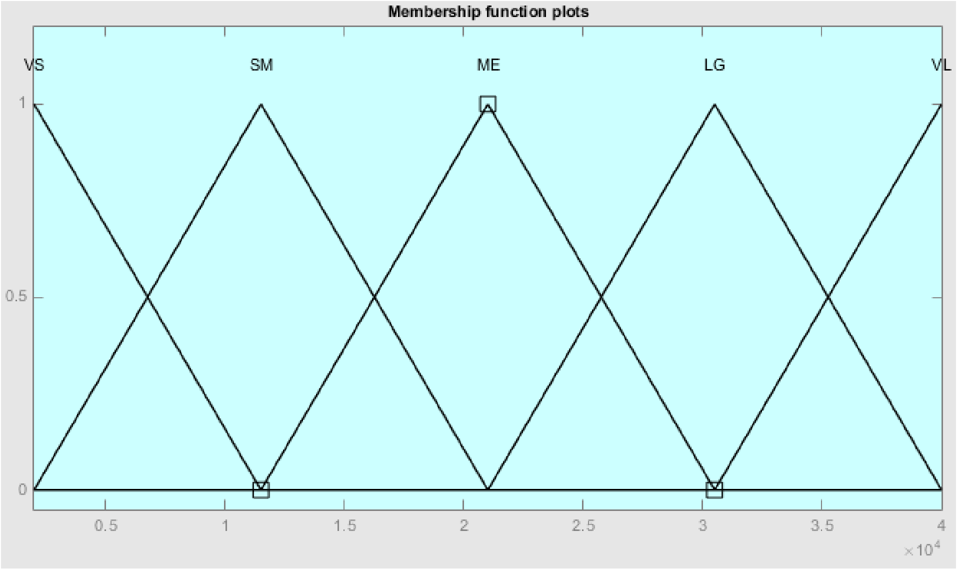


Figure 6-13 – Suspension damping (output)

Next step was to assign the rules. The decision matrix was as follows:

Table 18 - Decision matrix for semi-active suspension FLC

		Sprung Mass Velocity		
		NE	ZR	PO
Suspension	NE	VL	LG	ZR

	ZR	SM	VS	SM
	PO	ZR	LG	VL

After the general form Fuzzy Inference System (FIS) was determined, Genetic Algorithm was used to find the best parameters of membership functions to minimize sprung mass acceleration RMS (a human comfort index). 50 genomes and 100 generations was set for the GA and it was run with 1e-4 tolerance of optimization. The minimum objective function value (sprung mass acceleration RMS here) at each generation is shown in Figure 6-14.

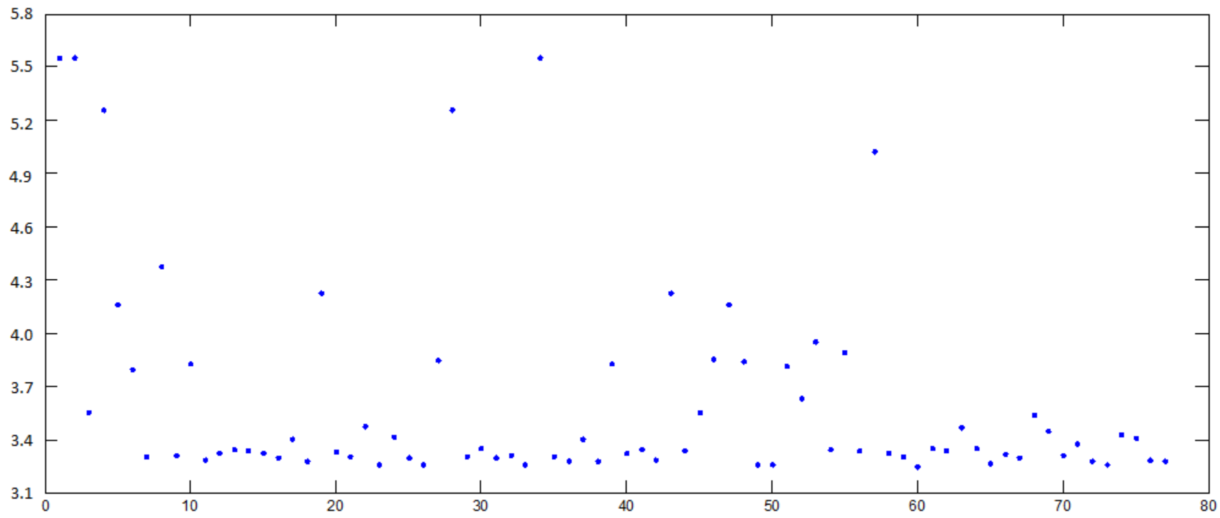


Figure 6-14 - Minimum objective function value at each generation of GA

Is shown that the GA was terminated before at the 77th generation due to the optimization tolerance being met. The minimum objective function value was 3.28 m/s^2.

The designed FLC was compared to a PDD controller as long a passive suspension case same as studied in 5.2.1. The sprung mass displacement plot is shown in Figure 6-15.

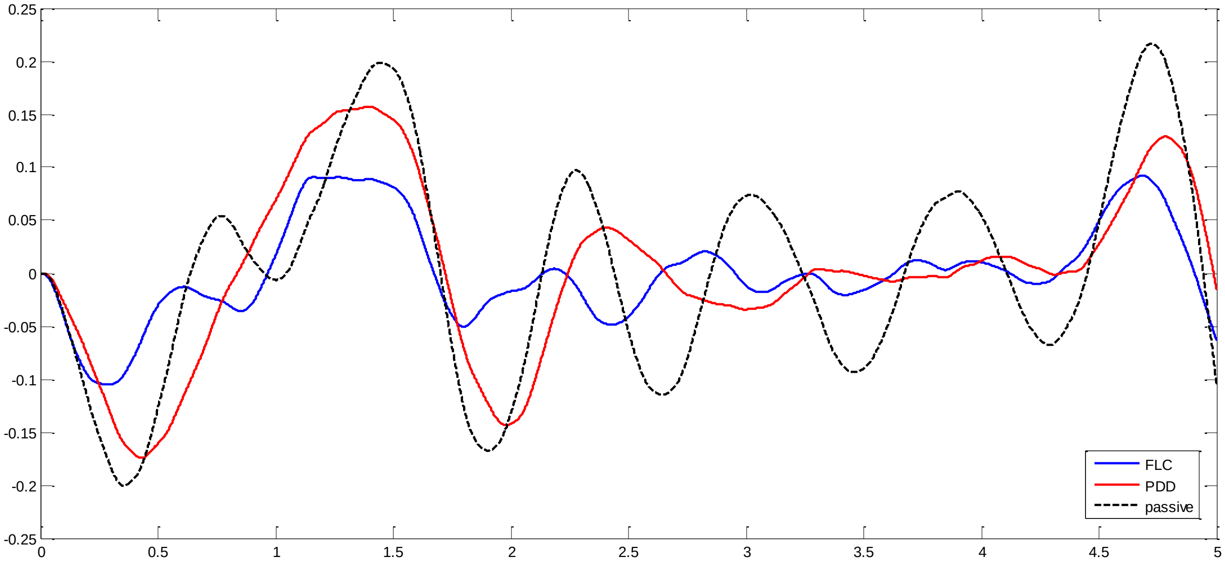


Figure 6-15 - Sprung mass displacement

As seen in figure above, the FLC shows a better performance in terms of sprung mass displacement even though the optimization was performed on sprung mass acceleration.

Figure 6-16 depicts the sprung mass acceleration of proposed FLC, PDD, and passive suspensions.

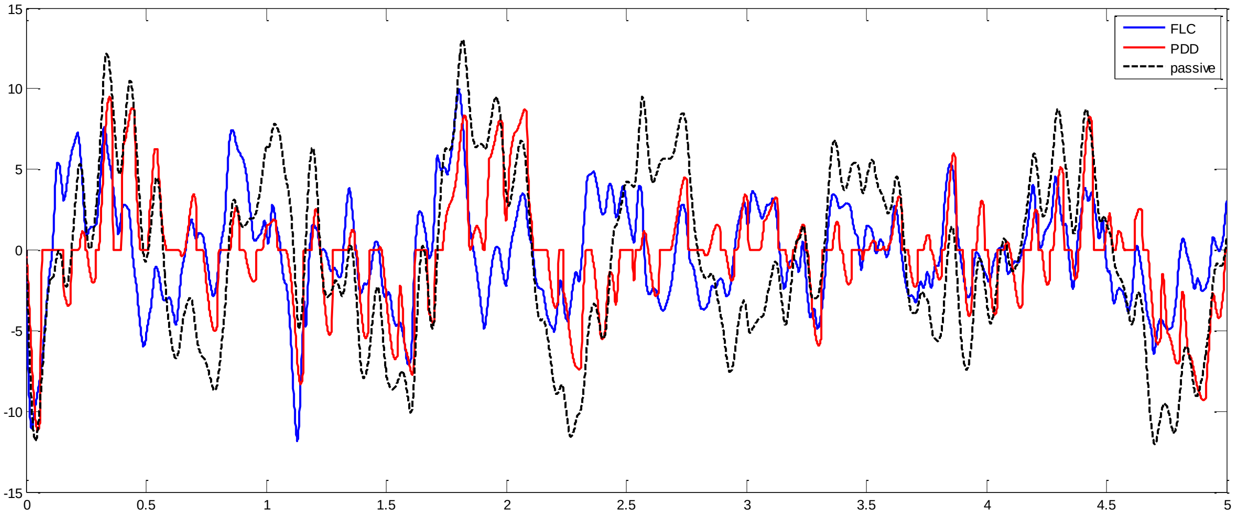


Figure 6-16 - Sprung mass acceleration comparison of proposed FLC, PDD, and passive case

Since the goal of the optimization was to minimize the sprung mass RMS the difference between PDD and the proposed FLC is not quite obvious from Figure 6-16.

To further investigate performance of FLC, PDD, and passive case, absorbed power graphs of them plotted in Figure 6-17.

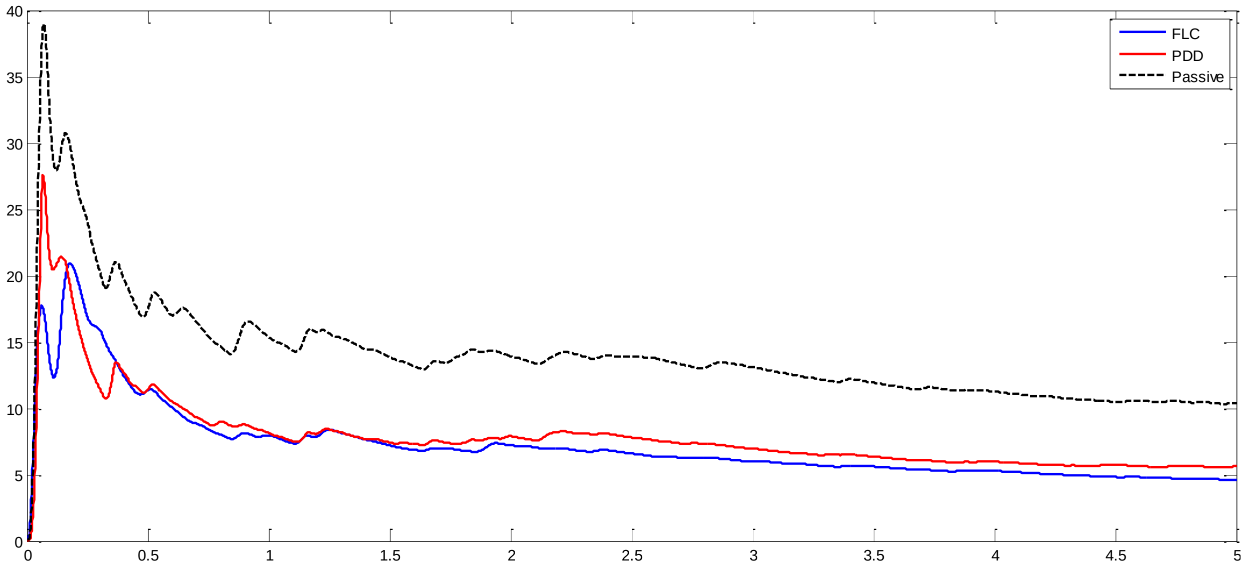


Figure 6-17 – Absorbed power comparison of proposed FLC, PDD, and passive case

From figure above is it obvious that the proposed FLC is a better controller than PDD however, they are both far better choices than the passive.

Table 19 contains the human comfort indices of the aforementioned semi-active suspensions control methodologies.

Table 19 - Human comfort indices

Method	Acceleration RMS	Absorbed Power
PDD	3.3885	5.6376
FLC	3.2879	4.7321
Passive	5.5159	10.4817

6.3 Inverse Adaptive Neuro Fuzzy Inference System (ANFIS) Controller

Adaptive Neuro Fuzzy Inference System (ANFIS) is an approach to design and train neuro-fuzzy systems by having sample input and output data points of system. This method is usually used to train an unknown model when input and output signals are known.

6.3.1 Neuro-Fuzzy Systems

Parameters of a fuzzy inference system is obtained from a series of neural network learning algorithms in neuro-fuzzy systems. These integrated systems have similar scheme of data and demonstration of knowledge behind the model. Human expertise has a significant role in a fuzzy system where the necessary components of the rules are derived, fuzzy reasoning is utilized to come up with the final output. Thus the if-then rules are extensively dependent of the designer’s knowledge of the system and the same is for the membership functions if the fuzzy system. It becomes more difficult when there is no mechanism to generate the rules and membership functions of a fuzzy system from experiences and knowledge of the designer. Similar to other optimization and learning algorithms, adaptability is another need to guarantee precision of the system within an acceptable tolerance. On the other hand, learning mechanism of neural networks is not dependent of human knowledge or expertise of the designer. Because of unified structure of data in neural networks, it is very uncommon to derive data from the parameters of neural networks. The hyper-planes in a neural network divide the input space into regions where the output values are differentiated. If there was any way of visualizing these hyperplanes, we would be able to reduce the learning mechanism from the training data. However, practically, the knowledge is transformed from human designers and it more convenient, both in terms of being comprehensive and straight-forward, to express the knowledge in form of fuzzy systems. In other words, the if-then rules are the best way to demonstrate such knowledge.

To a large extent, the drawbacks pertaining to these two approaches seem complementary. Therefore, it seems natural to consider building an integrated system combining the concepts of FIS and ANN modeling. A common way to apply a learning algorithm to a fuzzy system is to represent it in a special neural network like architecture. However the conventional neural network learning algorithms (gradient descent) cannot be applied directly to such a system as the functions used in the inference process are usually non differentiable. This problem can be tackled by using differentiable functions in the inference system or by not using the standard neural learning algorithm.

6.3.2 Mamdani Integrated Neuro-Fuzzy Systems

In Mamdani neuro-fuzzy systems, backpropagation learning technique is used to train the membership functions and their parameters. Mamdani neuro-fuzzy system schematic view is shown in Figure 6-18.

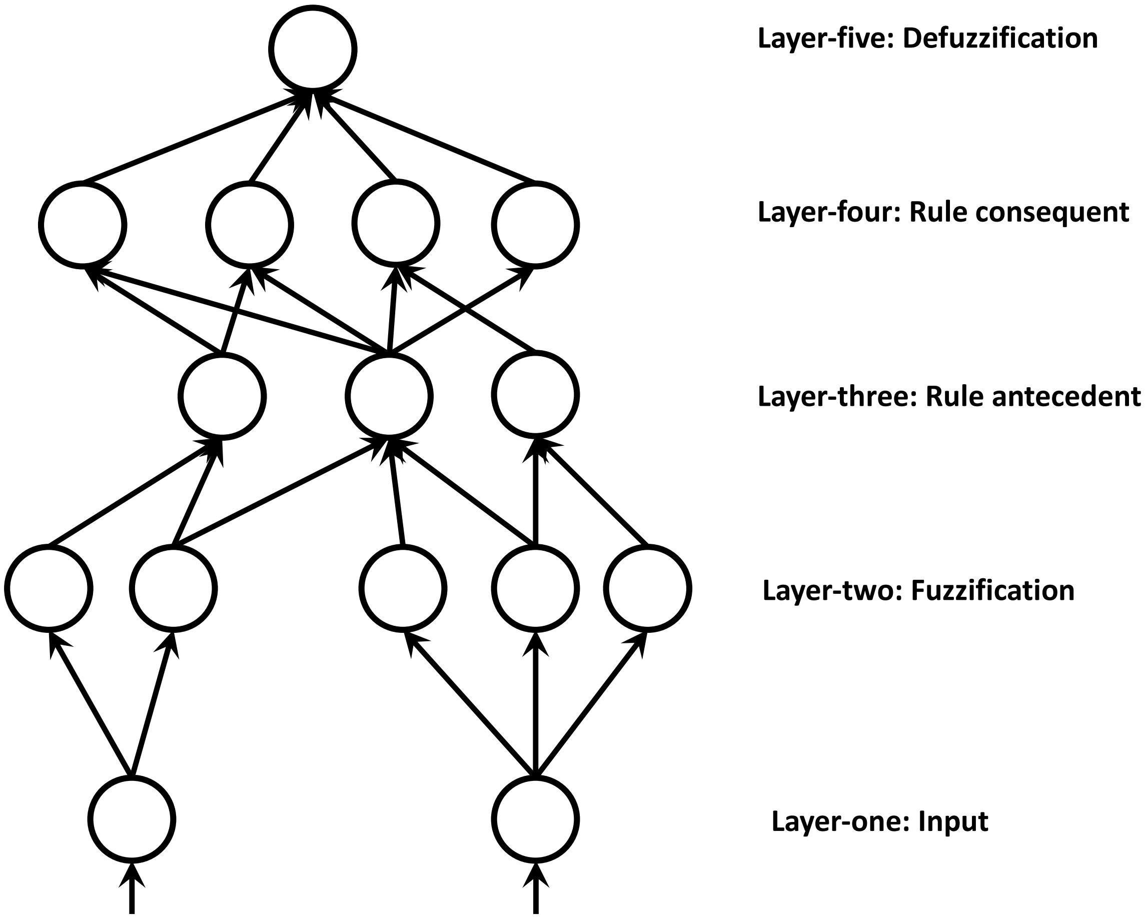

Layer-five: Defuzzification
Layer-four: Rule consequent
Layer-three: Rule antecedent
Layer-two: Fuzzification
Layer-one: Input

Figure 6-18 - Neuro-fuzzy system schematic diagram

Explanation of each layer follows: Layer-one (the input layer): The name itself is pretty self-explanatory. No calculation is performed in this layer and only the input values is passed to the next layer without any change or modification. The link weight in layer one is unity meaning the input value is passed the way it is and it not scaled by any value. Layer-two (the fuzzification layer): every node on layer two points one linguistic variable (low, medium, high, etc.) to one of the input variables in layer one. Meaning, the membership degree and that is the membership value of inputs to the fuzzy set of each linguistic variable where it is calculated in this layer. A learning algorithm determines what parameters will be associated to each membership function and how they will be oriented. Moreover, the final decision making about membership functions will be done based on the network learning procedure. Layer-three (the rule antecedent layer): this is where the multiple inputs are evaluated based on the rules and output of each rule is calculated (a T-norm most of the times). This output implies the impact of the rules and the amount of a particular rule in the final output. Layer-four (the rule consequent layer): In this layer the conclusion of each rule combine together to be passed to the next layer. Layer four determine the degree of membership of each output and usually utilizes S-norm to come up with the output of this layer. Layer-five (the combination and defuzzification layer): This layer defuzzifies the final output using the defuzzifing methods commonly used.

6.3.3 Takagi-Sugeno Integrated Neuro-Fuzzy System

Takagi-Sugeno (TSK) neuro-fuzzy systems are known for their capability of being used in mathematical learning procedures. TSK systems utilize a family of backpropagation training algorithms to come up with the membership functions coefficients as well as least mean square estimation to obtain the rule conclusion procedure. Learning algorithm can be broken down to two major parts: propagation of input patterns are done in the first part, and the optimization of parameters are done though an iterative procedure where the least mean square estimation method is the major player, while the membership functions are kept fixed intentionally during this training algorithm. The membership function will be optimized later on. In the next part, the patterns are propagated one more time, and at this step a supervised learning technique (backpropagation) is used to come up with the membership function coefficients, while the conclusion parameters are kept fixed. By iteration of this procedure the algorithm continues. Functionality of each node is shown in details and (as depicted in Figure 6-18) is as follows: Layers one, two and three works similar to Mamdani FIS. Layer four (the rule strength normalization): all nodes layer four calculate the following ratio:

$$\overline{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2 \qquad \text{Eq. } 32$$

Layer-five (the rule consequent layer): All nodes have the following function:

$$\overline{w}_i f_i = \overline{w}_i(p_i x_i + q_i x_2 + r_i) \qquad \text{Eq. 33}$$

where $\overline{w}_i$ is the output of layer four, and {pi,qi,ri} is the coefficient set. A well-known method of finding the coefficients is the least mean square estimation algorithm. Layer-six (rule inference layer), the final node, layer-six, computes the final crisp output or in other words, it defuzzifies the final output of the system:

$$Overall\ output = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \qquad \text{Eq. 34}$$

6.3.4 Adaptive Network Based Fuzzy Inference System (ANFIS)

This chapter goes over an architecture and learning methodology of adaptive networks that is a family of all neural networks that utilize feedforward concept for backpropagation learning algorithms. In adaptive networks nodes and directional links play important roles and ae connected to each other. Furthermore, those nodes that are adaptive (can be a part of nodes or all of them) contain a set of parameters meaning the learning algorithm is responsible for training parameters of those nodes and that means eventually the error index will be minimized.

Fuzzy logic only is implemented on systems that the rules are set by the designer and the linguistic variables along with the membership functions are properly lined up according to the desired characteristics of the system. However, it is almost impossible to come up with proper rules and membership functions (parameters in general) that fulfil design requirement or show desired performance. In other words, sometimes (significant number of situations) data is too raw to conclude something from it. This way, the parameters of membership function are chosen so that the final input/output relationship works as required and the membership functions are design based on the constraints set by the user. Saying above, an adaptive mechanism to come up with the parameters is needed where neuro-fuzzy adaptive algorithms are very well suited for this situation. The so-called neuro-fuzzy adaptive mechanism works very similar to the neuro-adaptive training procedures. The aforementioned mechanism is suitable for training parameters of fuzzy systems from variations in data sets. For a given input/output set of data points, it automatically calculates the membership function parameters that let the fuzzy system follow them. A very neural network-like can be implemented here to analyze input/output data to map the input through membership functions of the fuzzy modeling system into the given output data points. The above discussed parameters are varied and suited during this learning process. A gradient vector is developed to process those parameters and train them according to the constraints. The gradient vector is a measure of how the neuro-fuzzy system is behaving and can be an index of the performance of the system. Once the gradient vector is calculated, any training and optimization method can be utilized in order to find the best parameters with lowest possible error. Most of the times the error measure is just the squared of the error between the obtained value and the given output. Adaptive Neuro Fuzzy Inference System (ANFIS) utilizes back propagation and least mean square method for estimation of membership parameter . An ANFIS system is shown in Figure 6-19 schematically.

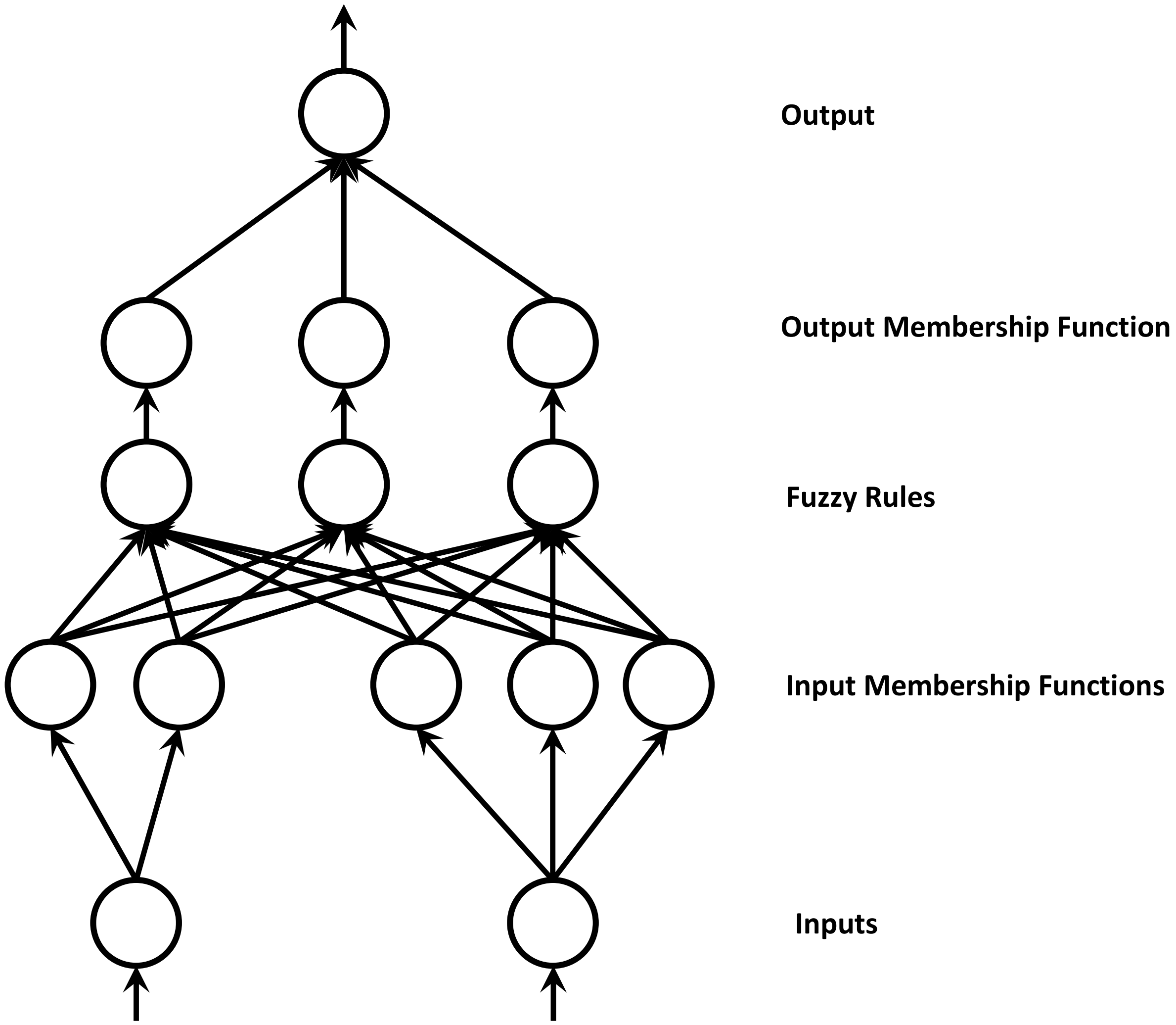


Figure 6-19 - ANFIS diagram

In general, ANFIS is a procedure to train a TSK neuro-fuzzy system with known inputs and output in order to follow an unknown system. As shown in Figure 6-19, first, the crisp inputs are fuzzified and processed using the fuzzy rules. Next, the implication and inference mechanisms turn the outputs of the rules to crisp values through difuzzification process.

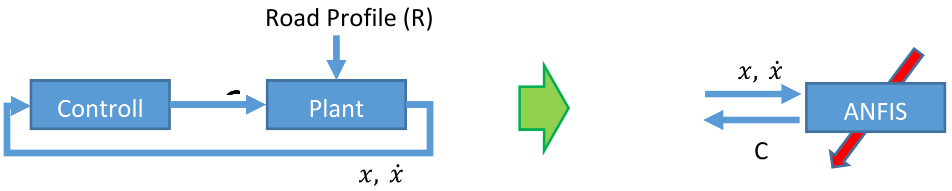
6.3.5 Inverse ANFIS Controller for semi-active suspensions on quarter-car model

Ideally, when an inverse of a plant is available, the easiest and most straightforward way to control the system is to input the desired output and feed the inverse plant output signal as the control input to the main plant. In general invers of the plant is not available and one has to find a way to either train the inverse of the system or estimate it[30]. This method of control is done by two steps, learning step and application step. In learning step, a learning algorithm is used to train the inverse of the system using input/output given data. The trained inverse model makes the control input to be fed to the main plant.

To implement this control scheme on quarter-car semi-active suspensions system, first the outputs and inputs of the plant must be determined. In this study, states of the system and their derivatives are taken as the plant outputs. In other words, displacement, velocity, and acceleration of both sprung and unsprung masses are considered as the outputs of the system. On the other hand, the damping coefficient is considered as the input of the plant. Note that road profile, despite of being an external signal, is considered as disturbance signal due to being unknown, hence, is not considered as the plant input.

In order to train the ANFIS system, a set of data is required to be fed to the training algorithm. Thus, a simulations was run to collect input and output data of plant. Next, the collected data was given to the ANFIS training algorithm. Note that in order to train a system that gives damping coefficient when states are given, the damping must be treated as the output and states as the inputs of the trained ANFIS. The reason that it is called Inverse ANFIS controller is due to this swapping of inputs and output.

Figure 6-20 shows the ANFIS performance after training.



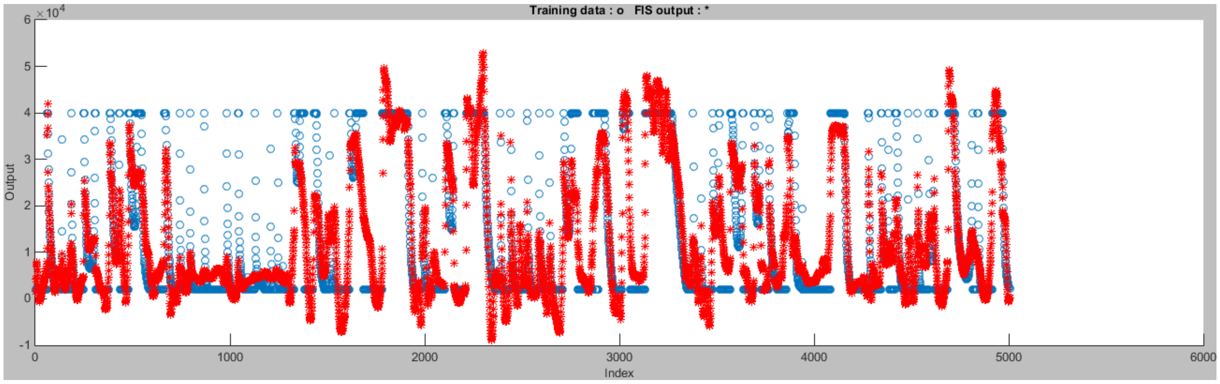
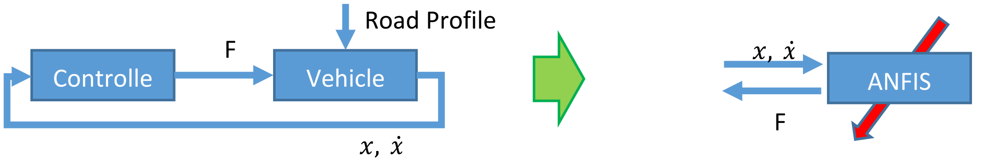


Figure 6-20 - ANFIS training results with damping coefficient as the input

As seen, the training result did not come as expected because damping coefficient is constrained to be between a lower and an upper limits whereas, there is no such constraint in the ANFIS training algorithm. One away around this issue is to replace damping coefficient with damping force which, in theory, does not have any limitation. Training algorithm was run but, this time with the damping force as the output of the ANFIS.



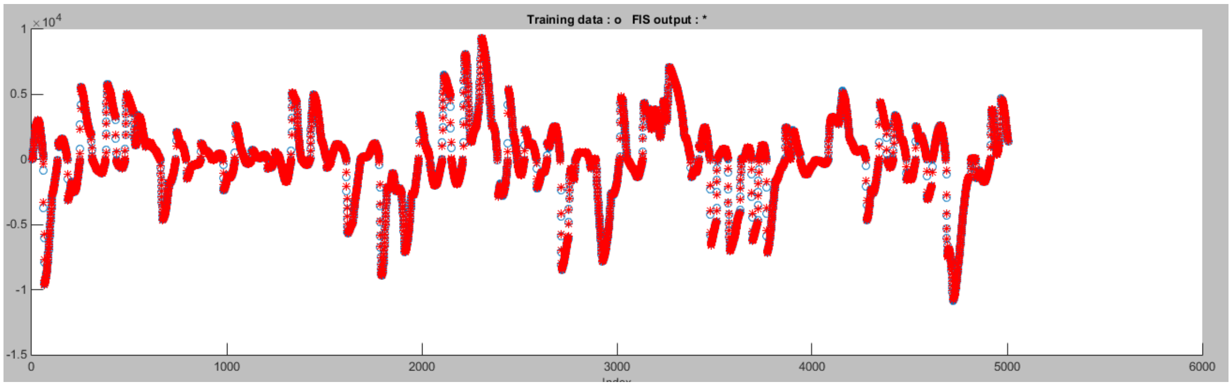


Figure 6-21 - ANFIS training results with damping force as the input

Figure 6-21 shows that choosing damper force as the output of ANFIS was effective and trained ANFIS returned estimated damping force with accepted accuracy.

Next step is establishing a procedure that feeds the Inverse ANFIS controller with desired states so that the corresponding damping force will be estimated by the controller. Once the desired damping force is given, it will be easy to calculate the damping coefficient which will be fed to the main plant, the quarter semi-active suspension system.

Ideally one would want to reduce all states as much as possible so ewthat all ride metrics would be reduced. Therefore, adjusting the desired states all equal to zero would be an ideal choice. However, there are two facts that make such assumption impossible. It has been seen before that reducing one ride metric such as human comfort, causes the other increase, roughly speaking. Another reason, if all desired states are set to zero, then the controller will be fighting against the system dynamics and that causes unwanted chattering and noises in the actual states of the system.

On the other hand if all desired states are set equal to the current state of the system (full feedback), then the effect of controller will be neutral.

One approach to find the best desired states is to feed all the current states back to the controller but with a correction factor. Figure 6-22 explains how the desired states are chosen based on the actual states of the system.

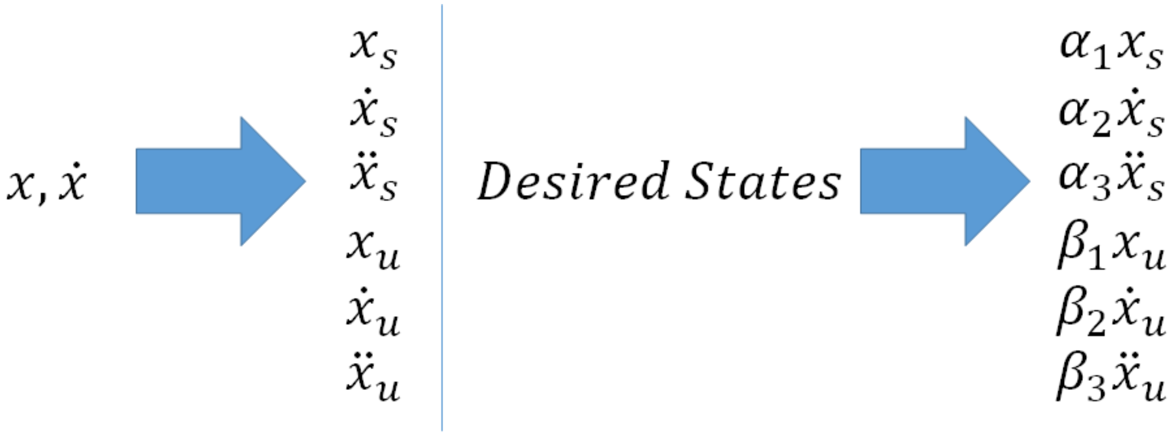


Figure 6-22 - Desires states with correction factor

The goal here is to find the best values for the correction factors that maximize performance of the controller. It is obvious that not all of correction factors have the same effect on the controller performance. Therefore, it is desired to reduce the number of correction factors and only optimize those that have significant effect on the ride metrics.

In order to analyze the effect of correction factors One-at-a-time method was used. The nominal value of each correction factor was set to 1 and they varied from 0.5 to 1.5.

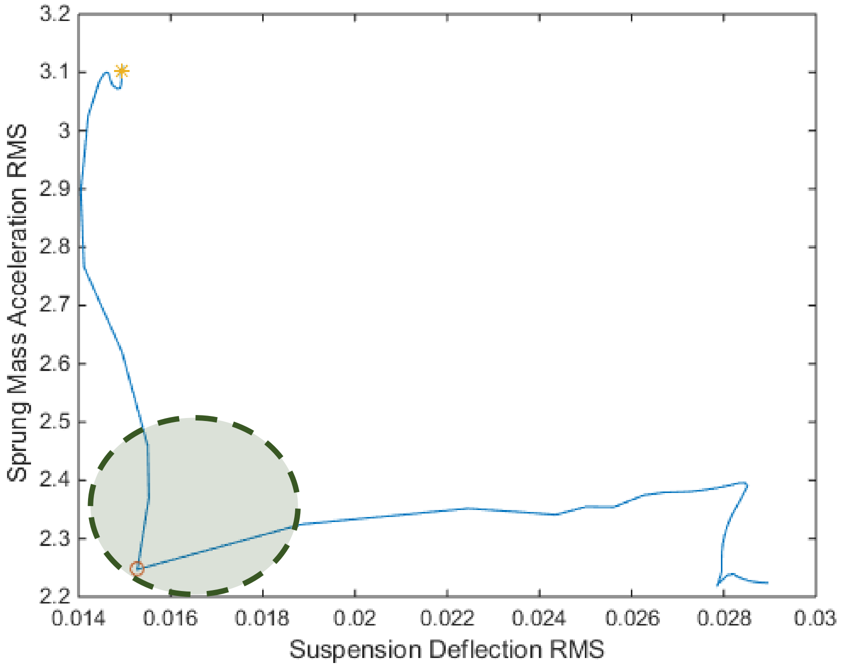


Figure 6-23 - Performance plot when α_1 was changing

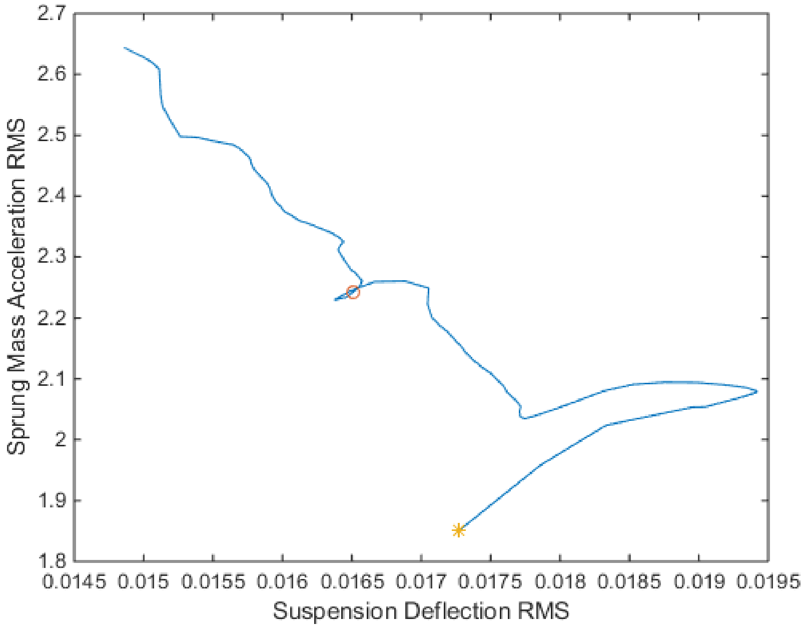


Figure 6-24 - Performance plot when α_2 was changing

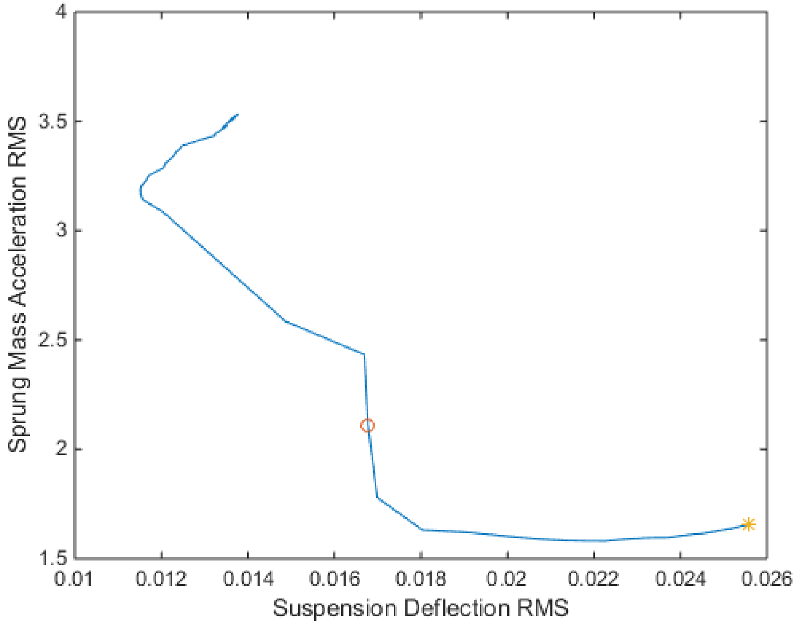


Figure 6-25 - Performance plot when α_3 was changing

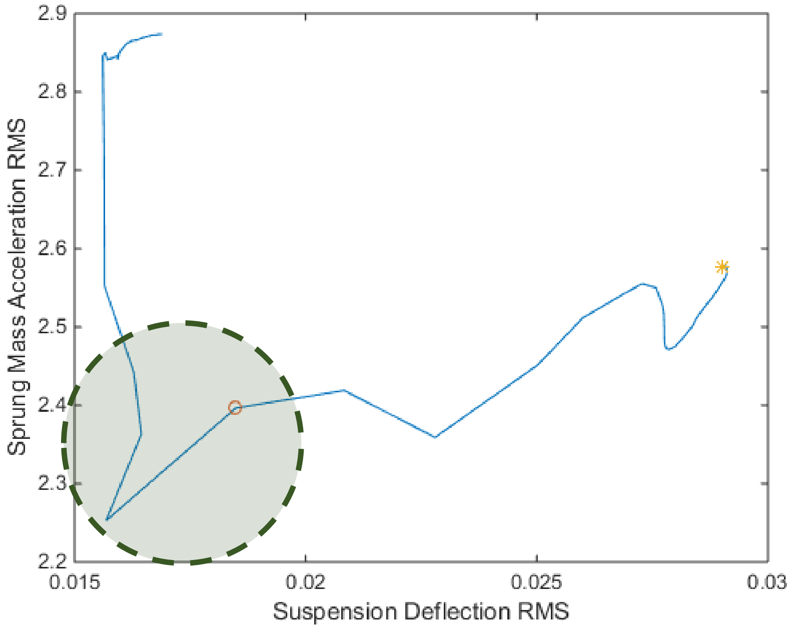


Figure 6-26 - Performance plot when β_1 was changing

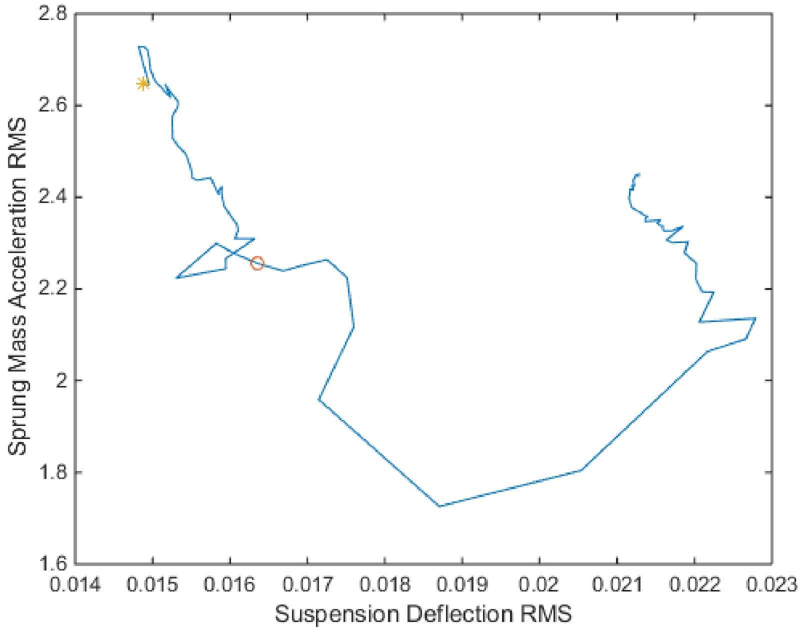


Figure 6-27 - Performance plot when β_2 was changing

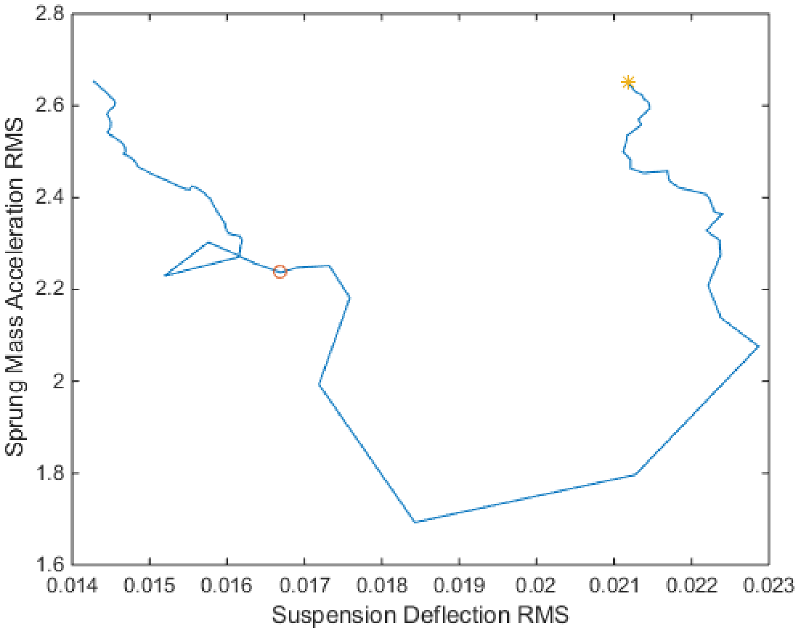


Figure 6-28 - Performance plot when β_3 was changing

Again the start indicates where correction factor is at it minimum value, 0.5 and circle indicates the nominal value of correction factor. From Figures above it can be interpreted that α_1 and β_1 have the most effect on the ride metrics and with the choice of their values an optimum performance of the vehicle cane be reached. Other correction factors either do not reach an optimum performance point or do not follow a promising trend that a better performance can be seen.

According to analysis above, α_1 and β_1 were chosen as the most effective correction factors and the rest were set to 1 meaning, the real states of the system are fed back to the controller without a correction factor except α_1 and β_1.

6.3.5.1 Sensitivity Analysis

To investigate role of each correction factor using sensitivity analysis (explained in details in previous chapters), all factors were considered as inputs with nominal values of 1 and range of variation between 0.5 and 1.5. The results of the FAST method are shown in Figure 6-29, it can be observed that α_1, β_1, and α_3 have the highest influence on the performance indices among all correction factors.

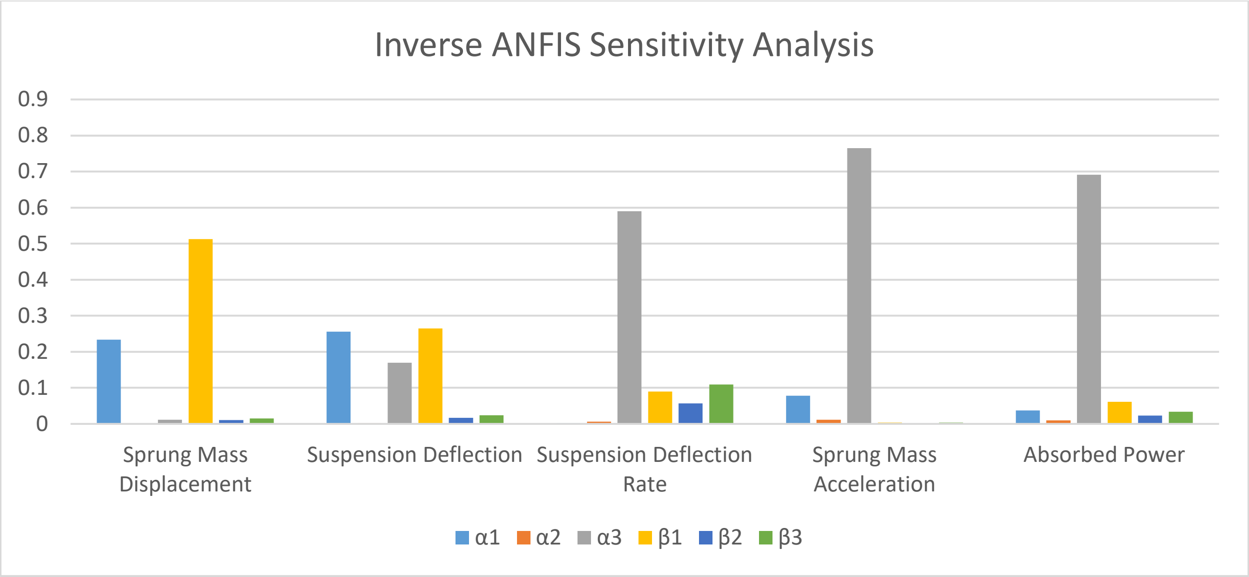


Figure 6-29 - Inverse ANFIS sensitivity analysis of correction factors

However, since the goal is to use minimal number of correction factors and fix the rest at 1, once has to decide whether a correction factor among α_1, β_1, and α_3 can be ignored or not. Considering the fact that α_3 is the second derivative of α_1, it is not unreasonable to fix α_3 at 1. In other words, including both α_3 and α_1 in final fin-tuning would be redundant.

In addition to reasoning above, previous section showed that tuning α_1 and β_1 can lead to a better performance as opposed to α_3.

6.3.5.2 Fine-tuning and Optimization

Genetic algorithm was used to find the optimum values of α_1 and β_1. Table 20 shows the values of correction values based on the provided objective function.

Table 20 - Correction factor optimum values

Objective Function	α_1	β_1
Sprung Mass Acceleration RMS	0.7879	0.7071
Suspensions Deflection RMS	0.7879	0.9091
As described in Eq. 28	0.7879	0.8641

Moreover, Figure 6-30 and Figure 6-31 illustrate how ride metrics change when the correction factors vary between 0.5 and 1.5.

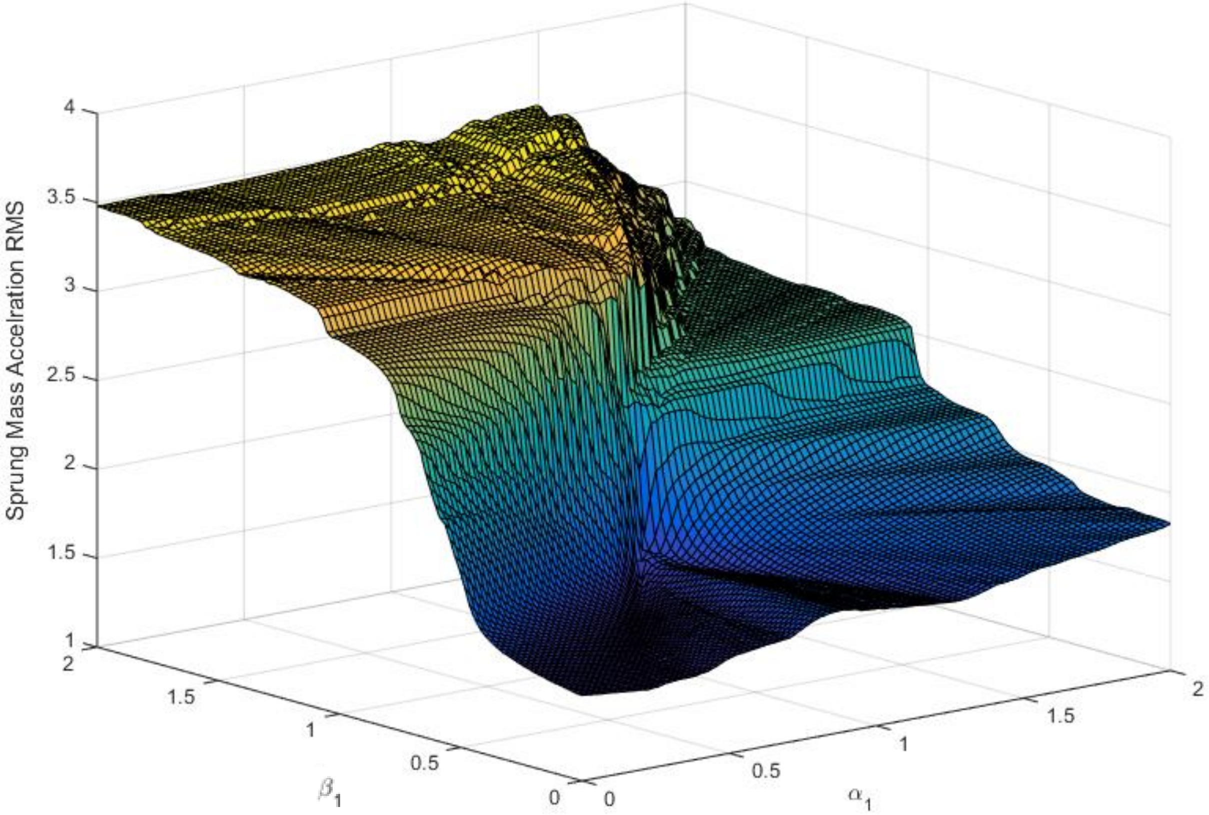


Figure 6-30 – Sprung mass acceleration RMS surface when α_1 and β_1 were changing

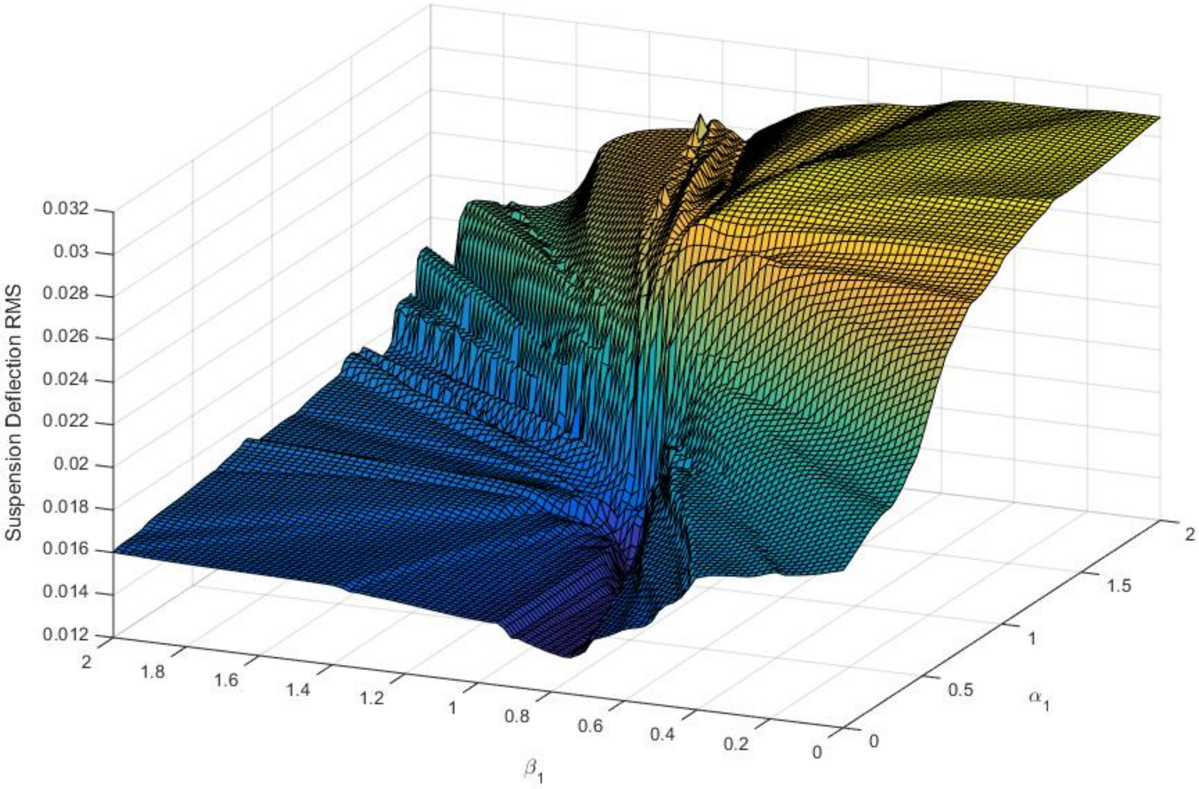


Figure 6-31 – Suspension deflection RMS surface when α_1 and β_1 were changing

The fact that optimal values are smaller than 1 is because the effort is to reduce the states magnitudes in order to reduce the ride metrics. Also, they are close to 1 since being much smaller, causes the controller fight against the model dynamics and that could introduce unwanted noises to the signals and worsen the ride metrics.

For the first case at Table 20 where the objective function is the sprung mass acceleration RMS, absorbed power plot is shown in Figure 6-32.

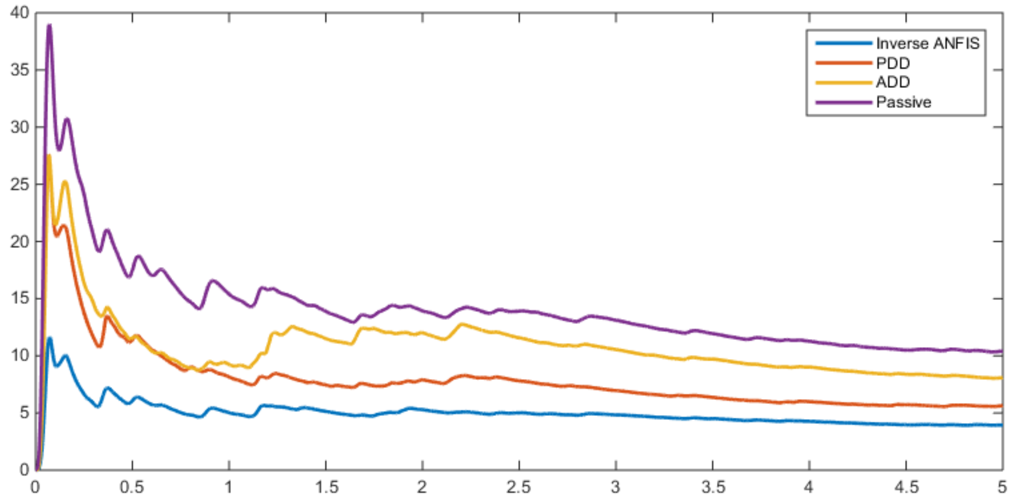


Figure 6-32 - Absorbed power plot for Inverse ANFIS vs other methodologies

As show above the absorbed power for the Inverse ANFIS method is lower than the other methods. Not that absorbed power is a measure that how much power would be absorbed by an average male internal organs at their natural frequencies.

7 Conclusion

Control and analysis of semi-active suspensions is what was studied in this document. The necessity to do so was introducing and developing car model where the semi-active suspensions are mounted and functioning. In the second chapter, the developed car models were discussed starting from a simple quarter-car model. Half-car model and Full-car model with axles were the other two car models investigated in this document. Finally, a car model was introduced with arbitrary number of axles. In this model, equations of a full-car model with *n* number of axles were developed so, one can automatically generate the equations by having the parameters of the car. This would be helpful to automate car modeling simulations.

In chapter 3, primary controllers that are widely used by researchers and in the automotive industry were introduced. Skyhook, Groundhook, hybrid Skyhook-Groundhook, ADD, and PDD were the controller schemes that were studied in this chapter. Eventually, performance of those controllers were evaluated by measuring the ride metrics of aforementioned car models while those particular control algorithms were implemented.

Chapter 4 was dedicated to analysis and fine-tuning of previously introduced controllers. There is not one sing le control scheme that works all the time for any vehicle/road condition. Picking the right controller using proposed approach was the goal of first part of this chapter. At the end of this part, a novel approach was developed on how to choose the best performing controller based on the driver/vehicle/road conditions. The second part of the same chapter proposed a new method to fine-tune a family of controllers. By a family of controllers, those algorithms are referred that contain a tuning variable in their control laws. Skyhook, Groundhook, and hybrid Skyhook-Groundhook were the algorithms that were studied in that sense.

H_∞ Robust Control was introduced and its equations were derived in chapter 5. It was mentioned that pure H_∞ Robust Control could not effectively enhance ride metrics of a vehicle although it is considered as powerful control scheme that cancels external disturbance effect. The reason is the hardware limitation that implicitly exists in semi-active suspensions. Basically, semi-active suspensions cannot acquire damping of beyond a positive range. That range is defined by two lower and upper damping values. Since H_∞ does not have a constraint for such limitation, the results of implementing it on a system with such limitation will cause singularity in the dynamics equations. Therefore, a slight modification was applied to the H_∞ Robust Control concept by assuming a dummy model and designing the controller for that dummy car model. By saturating the control input generated for the dummy model and feeding it to the real system, the controller did not cause any stiffening of the dynamics equations. Later in the same chapter the proposed novel controller was implemented on a quarter-car model as well as a full-car model with 6 axles. The simulations result showed that the modified H_∞ controller improved ride metrics of the vehicle compared to other well-known controllers and a passive suspension.

Finally, Fuzzy Logic was discussed in the 6th chapter as a new approach to control semi-active suspensions. Fuzzy Logic Controllers (FLC) demonstrate the ability of being easily adaptable to the controller design requirements by directly applying the expert's knowledge. This can be interpreted as downside for FLC's because without having a prior knowledge, the designer will not be able to set the membership functions and rule of the FLC properly. A simple FLC was implemented on a quarter-car model and the results showed improvement compared to other controllers. Later, novel Inverse ANFIS control scheme was proposed. This algorithms utilized the power ANFIS schemes to train a controller based on a set of collected data. This controller works based on a set of desired inputs. The desired inputs are set by the user and are usually the plant outputs that are minimized. For the next step, a methodology was proposed to reduce the number of desired inputs to only those ones that have significant impact on the controller performance. Eventually, the controller was implemented on a quarter-car model and it was shown that significant enhancement can be reached using Inverse ANFIS control scheme.

References

1. Margolis, D.L., *Semi-active heave and pitch control for ground vehicles.* Vehicle System Dynamics, 1982: p. 31-42.
2. Margolis, D.L., *Semi-active control of wheel hop in ground vehicles.* Vehicle System Dynamics, 1983. **12**(6): p. 317-330.
3. Karnopp, D., M.J. Crosby, and R.A. Harwood, *Vibration control using semi-active force generators.* Journal of Manufacturing Science and Engineering, 1974: p. 619-626.
4. Choi, S.B., H.K. Lee, and E.G. Chang, *Field test results of a semi-active ER suspension system associated with skyhook controller.* Mechatronics, 2001. **11**(3): p. 345-353.
5. Hong, K.-S., H.-C. Sohn, and J.K. Hedrick, *Modified skyhook control of semi-active suspensions: A new model, gain scheduling, and hardware-in-the-loop tuning.* ournal of Dynamic Systems, Measurement, and Control, 2002. **124**(1): p. 158-167.
6. Caponetto, R., et al., *A soft computing approach to fuzzy sky-hook control of semiactive suspension.* Transactions on Control Systems Technology, 786-798. **11**(6): p. 786-798.
7. Ahmadian, M., X. Song, and S.C. Southward, *No-jerk skyhook control methods for semiactive suspensions.* Journal of vibration and acoustics, 2004. **126**(4): p. 580-584.
8. Emura, J., et al., *Development of the semi-active suspension system based on the sky-hook damper theory.* SAE Technical Paper, 1994: p. 17-26.
9. Ahmadian, M. and A. Christopher, *A Quarter-Car Experimental Analysis of Alternative Semiactive Control Methods.* Journal of Intelligent Material Systems and Structures, 2001: p. 604-612.
10. ValSek, M. and M. Nova. *Ground Hook for Semi-Active Damping of Truck's Suspension*. in *Proc. of CTU Workshop 96, Engineering Mechanics*. 1996.
11. Valášek, M., et al., *Extended ground-hook-new concept of semi-active control of truck's suspension.* Vehicle system dynamics, 1997: p. 289-303.
12. Bellizzi, S. and a.R. Bouce. *Adaptive control for semi-active isolators*. in *Conference on Mechanical Vibration and Noise, ASME*. 1989.
13. Cheok, K.C. and N.J. Huang. *Lyapunov stability analysis for self-learning neural model with application to semi-active suspension control system*. in *IEEE International Symposium on Intelligent Control*. 1989.
14. Venhovens, P.T., *The Development and Implementation of Adaptive Semi-Active Suspension Control.* Vehicle System Dynamics, 1994. **23**(1): p. 211-235.
15. Gordon, T.J., *Non-linear optimal control of a semi-active vehicle suspension system.* Chaos, Solitons & Fractals, 1995. **5**(9): p. 1603-1617.
16. Valasek, M. and M. Novak, *Ground Hook for Semi-Active Damping of Truck's Suspension.* In Proc. of CTU Workshop, 1996. **96**: p. 467-468.
17. Rao, M.V.C. and V. Prahlad., *A tunable fuzzy logic controller for vehicle-active suspension systems.* Fuzzy sets and systems, 1997. **85**(1): p. 11-21.
18. Yi, K., B.S. Song, and J.H. Park, *Observer-based control of vehicle semi-active suspensions.* Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 1999. **213**(6): p. 531-543.
19. Yi, K. and B.S. Song, *Observer design for semi-active suspension control.* Vehicle System Dynamics, 1999. **32**(2): p. 129-148.

20. Jansen, L.M. and S.J. Dyke, *Semi-Active Control Strategies for MR Dampers.* Journal of Engineering Mechanics, 2000. **126**(8): p. 795-803.

21. Savaresi, S.M., E. Silani, and S. Bittanti, *Acceleration-Driven-Damper (ADD): An optimal control algorithm for comfort-oriented semiactive suspensions.* Journal of dynamic systems, measurement, and control, 2005: p. 218-229.

22. R, M. and Z. R. *Control of mechatronic systems by dissipative devices: application to semi-active vehicle suspensions*. in *American Control Conference*. 2006.

23. Chen, Y., *Skyhook surface sliding mode control on semi-active vehicle suspension system for ride comfort enhancement.* Engineering, 2009: p. 23-32.

24. Ghasemalizadeh, O., et al. *Semi-active Suspension Control using Modern Methodology: Comprehensive Comparison Study*. in *Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*. 2014. arXiv preprint arXiv:1411.3305.

25. Lin, et al. *Toward better ride performance of vehicle suspension system via intelligent control*. in *IEEE International Conference on Systems, Man and Cybernetics*. 1992.

26. Pradko, F., R. Lee, and V. Kaluza, *Theory of human vibration response*. 1966, DTIC Document.

27. Zadeh, L.A., *Fuzzy sets.* Information and control, 1965. **8**(3): p. 338-353.

28. Leekwijck, V., Werner, and E.E. Kerre, *Defuzzification: criteria and classification.* Fuzzy sets and systems, 1999. **180**(2): p. 159-178.

29. Madau, D.P. and L.A. Feldkamp. *Influence value defuzzification method*. in *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*. 1996.

30. Mouloud, D., F. Palis, and A. Zeghbib. *ANFIS based modelling and control of non-linear systems: a tutorial*. in *2004*. IEEE International Conference on Systems, Man and Cybernetics.