CRANFIELD UNIVERSITY

Yukio Watanabe

Learning Control of Automotive Active Suspension Systems

97.

School of Mechanical Engineering

Ph.D. Thesis

CRANFIELD UNIVERSITY

School of Mechanical Engineering

Ph.D. Thesis

Academic Year 1996-97

Yukio Watanabe

Learning Control of Automotive Active Suspension Systems

Supervisor: Professor Robin S. Sharp

September 1997

Abstract

This thesis considers the neural network learning control of a variable-geometry automotive active suspension system which combines most of the benefits of active suspension systems with low energy consumption.

Firstly, neural networks are applied to the control of various simplified automotive active suspensions, in order to understand how a neural network controller can be integrated with a physical dynamic system model. In each case considered, the controlled system has a defined objective and the minimisation of a cost function. The neural network is set up in a learning structure, such that it systematically improves the system performance via repeated trials and modifications of parameters. The learning efficiency is demonstrated by the given system performance in agreement with prior results for both linear and non-linear systems. The above simulation results are generated by MATLAB and the *Neural Network Toolbox*.

Secondly, a half-car model, having one axle and an actuator on each side, is developed via the computer language, AUTOSIM. Each actuator varies the ratio of the spring/damper unit length change to wheel displacement in order to control each wheel rate. The neural network controller is joined with the half-car model and learns to reduce the defined cost function containing a weighted sum of the squares of the body height change, body roll and actuator displacements. The performances of the neuro-controlled system are compared with those of passive and proportional-plus-differential controlled systems under various conditions. These involve various levels of lateral force inputs and vehicle body weight changes.

Finally, energy consumption of the variable-geometry system, with either the neuro-control or proportional-plus-differential control, is analysed using an actuator model via the computer simulation package, SIMULINK. The simulation results are compared with those of other actively-controlled suspension systems taken from the literature.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Robin S. Sharp, for his invaluable guidance during the course of my research and to my parents for their help and encouragement.

I am also pleased to acknowledge the Committee of Vice-Chancellors and Principals for their generous support through an Overseas Research Student award.

I am indebted to the following for their valuable comments and technical guidance: Mr. A. John Robertson, Dr. Bill J. Batty, Professor Ichiro Kageyama, Mr. Duncan Allison and Mr. Panos Kyratsis.

My warmest thanks are due to my English teacher, Mr. Rodney Eyre, for reading the draft of each chapter and making a number of helpful suggestions.

List of Contents

APPENDICES

List of Figures

List of Tables

Notation

$\mathbf{A}$, $\mathbf{A}_\mathrm{d}$	System-matrices for continuous and discrete-time systems, respectively
$\boldsymbol{a}^R, \hat{\boldsymbol{a}}^R$	Output vectors of neural network and lood-up data, respectively
$a^H{}_h, a^R{}_r$	Outputs of h th element at the hidden layer and r th element at the output layer of neural network, respectively
$a^{*H}{}_h, a^{*R}{}_r$	Net inputs of h th element at the hidden layer and r th element at the output layer of neural network, respectively
$\boldsymbol{B}$, $\boldsymbol{B}_d$	Distribution vectors for control inputs in continuous and discrete-time systems, respectively
b	Bias of a neural network element in Chapter 3
$\boldsymbol{b}^H$	Bias vector for hidden layer of neural network
b^R	Bias for output layer of neural network
$\boldsymbol{b}^R$	Bias vector for output layer of neural network
$\mathbf{C}$, $\mathbf{C}_\mathrm{d}$	Output matrices for continuous and discrete-time systems, respectively
c	Damping rate (N/(m/s))
c_s, c_t	Suspension and tyre damping rates, respectively (N/(m/s))
d_Y	Distance between the body centre and each side end (m) in Chapter 7
$D^H{}_h, D^R{}_r$	Learning signal of h th element at the hidden layer and r th element at the output layer of neural network, respectively
E	(i) Motor back-e.m.f. (V) in Chapter 9 (ii) Stochastic operator in Chapter 3, see equation (3.3)
e^R	Output error function in Chapter 3
$\mid d_S \mid, \mid d_P \mid$	Magnitudes of starboard and port potentiometers, respectively in Chapter 7
F	Output function of a neural network element in Chapter 3
F_D	Force on the lower wishbone (see Fig. 2.8 in Chapter 2)
F_{fric}	Friction force (N) in Chapter 9
F_{LPE}	Function of linear processing element
F_{NN}	Function of multi-layer neural network
F_{strut}	Suspension force (N) in Chapter 9

F_{Yb}	Lateral force through the vehicle body mass centre (N)
F_{YSW}	Lateral force through the starboard wheel mass centre (N)
F_{YPW}	Lateral force through the port wheel mass centre (N)
F_{ZS}	Vertical force through the starboard wheel/ground contact point (N)
F_{ZP}	Vertical force through the port wheel/ground contact point (N)
f_{rd}	Frequency of sinusoidal road velocity input (Hz)
f_{res}	Frequency resolution (Hz)
f_{YS}	Side force of the starboard wheel (N)
f_{YP}	Side force of the port wheel (N)
$\mathbf{G}$, $\mathbf{G}_d$	Distribution vectors for the disturbance inputs in continuous and discrete-time systems, respectively
G_1, G_2, G_3, G_4	Control gains in Chapter 7
I	Motor current (A) in Chapter 9
i	Count number
I_{nq}	Non-quadratic function
I_q	Quadratic function
J	Cost function
j	Motor inertia (kgm^2) in Chapter 9
$K(i)$	Weight as function of count number in cost function in Chapter 5
K_{crf}	D.C. current feedback transformer (V/A)
K_e	Motor back-e.m.f. constant (Vsec/rad)
K_P	Power amplifier gain (V/V)
K_S	Speed control gain (V/V)
K_T	Electromagnetic torque constant (Nm/A)
K_{tch}	D.C. tachogenerator (D.C. motor speed transformer) (Vsec/rad)
k_{coe}	Non-linear spring coefficient (N/m^3)
k_t	Tyre stiffness (N/m)
L	Linear output function of neural network element
$\mathbf{L}$	Vector of linear output functions of neural network elements
m	Discrete frequency step number
M_b, M_w	Body and wheel masses, respectively (kg)

n	Discrete time step number
N	Sample size of cost function
N_c, N_f, N_g	Neural network function and two non-linear functions of plant, respectively (see Fig. 3.10 in chapter 3).
N_{rd}	Sample size of random noise road input
P_1,P_2,.....	Coordinate points in the models in Chapter 7
P_P, P_S	Coordinate points of the vehicle body port and starboard ends, respectively, in Chapter 7
q	Total batches of training in Figures 5.3, 6.3 and 8.3
Q	Total discrete-time steps of training in Figures 4.4, 5.3, 6.3, 8.3 and 8.4
R	Motor resistance (Ω) in Chapter 9
r	Mean radius of leadscrew (m) in Chapter 9
q_1, q_2, q_3, q_4	Weighting constants in cost function
S	Tangent-sigmoid transfer function of neural network element
S	Vector of tangent-sigmoid transfer functions of neural network elements
S_p, S_h, S_r	The number of processing elements in each of input, hidden and output layers, respectively
t	Continuous time in seconds
T	Sampling time (second)
T_L	Load torque (Nm) in Chapter 9
T_M	Motor electromagnetic torque (Nm) in Chapter 9
u	(i) Actuator force (N) in Chapters 4, 5 and 6 (ii) Output of a neural network element in Chapter 3
u^*	Net input of a neural network element in Chapter 3
$\mathbf{u}$	Vector of actuator displacements from equilibrium (m) in Chapter 8
V	Motor driving voltage (V) in Chapter 9
v	External disturbance, road input vertical velocity (m/s) in Chapters 4,5 and 6
$\mathbf{v}$	External disturbance vector, lateral forces through the body and two wheels (N) in Chapter 8

V_S, V_P	Actuator velocity commands on the starboard and port sides, respectively, (m/s) in Chapter 7
V_m	Amplitude of each frequency component in random noise (m) in Chapter 5
$V_m(i)$	Amplitude function of count number in random noise in Chapter 5
W	Energy consumption (J) in Chapter 9
w	Weighting parameters in neural network controller (vector form)
$\mathbf{w}^{\mathrm{H}}$	Weighting parameters for hidden layer of network (matrix form)
w^R	Weighting parameters for output layer of network (vector form)
$\mathbf{w}^R$	Weighting parameters for output layer of network (matrix form)
w_h	h th weighting parameter in a neural network element
$w^H{}_{h,p}$	Weighting parameter between p th element at the input layer and h th element at the hidden layer
$w^H{}_{r,h}$	Weighting parameter between h th element at the hidden layer and r th element at the output layer
x	State vector of system
y	Output vector of system
y_{act}	Actuator displacement from equilibrium (m) in Chapter 9
z_b, z_r, z_w	Displacements of body, road and wheel, respectively (m)
Z_b, Z_S, Z_P	Displacements of the body centre, body starboard end and body port end, respectively (m) in Chapter 7
α	Leadscrew helix angle (rad) in Chapter 9
β	Leadscrew friction angle (rad) in Chapter 9
Δw	Weighting parameter update of neural network
ΔP	Load transfer (see Fig. 2.8 in Chapter 2)
δ	Learning rate
ϕ	Vehicle body roll angle from equilibrium (rad)
γ	Shaping parameter in the friction function (9.8)
φ	The angle (rad) between the direction of the strut force and the line, which is at a right angle to the track in Chapter 9

μ	Friction coefficient between the slider and track in Chapter 9
ξ	Load transfer ratio
θ	Motor armature displacement from equilibrium (rad) in Chapter 9
θj	Weighting parameters of neural network (see Fig. 3.10 in Chapter 3)
θ_m	Phase lag of each frequency component of random noise in Chapter 5
$\mathbf{\Phi}$, $\mathbf{\Psi}$	Vector functions in difference equations

D.C.	Direct Current
E.m.f.	Electro-motive force
G	Gravity
LQG	Linear Quadratic Gaussian
MP	pontryagin Maximum Principle
Neuro	Neural network(s)
P + D	Proportional-plus-Differential
R.m.s.	Root-mean-square
SOFLIC	Self-Organising Fuzzy Logic Intelligent Control

Chapter 1
Introduction

In recent years, many improvements have been made in the automotive field by the use of mechatronics and advanced control. In the context of vehicle stability, handling and ride comfort, for example, active suspension systems, i.e. those which use controlled actuators and energy supplies, can provide performance which is markedly superior to that possible with conventional passive suspensions involving springs and dampers. Although various types of actively-controlled suspension systems have been studied and developed in both academic research and industrial fields, their commercial impact has been very small due to a major problem - that of energy consumption (Hillebrecht *et al,* 1992).

The great interest in low-energy design of active suspension systems has led to a re-emergence of an old principle - i.e. variable geometry affecting the ratio of the spring/damper unit length change to wheel displacement - used with manual adjustment on a Velocette Thruxton Motorcycle in 1965 (Hicks, 1992). A type of variable geometry mechanism, containing pre-loaded springs with adjustable cranks, has been made the basis of a computer-controlled suspension system on a passenger car in Delft University of Technology (Venhovens, Knaap and Pacejka, 1992; Knaap, Venhovens and Pacejka, 1994). Another variable geometry mechanism was achieved by moving one end of a buckling spring element with an electro-mechanical actuator (Leighton, 1995). These types of suspension system, without actuation, are essentially passive suspensions and achieve levelling and body-attitude-control with relatively small actuators and low energy consumption, but an increase in mechanical complexity ensues. Although the present thesis proposes a much simpler variable geometry mechanism which involves sliding one end of a spring/damper unit on a mechanical track, the control system design problems are novel due to non-linearity and mechanical limitations.

One way of tackling the control problems is to employ learning techniques, which include automata (Gordon, Marsh and Wu, 1993; Howell, Frost and Gordon, 1996), self-organising fuzzy logic intelligent control and neural networks (Harris, Moore and Brown, 1992), neuro-fuzzy (Kiguchi and Fukuda, 1996) and genetic algorithms (Yeh, Lu and Chen, 1994). Vehicle system applications of neural networks and fuzzy logic have been reviewed very recently by Zadeh, Fahim and El-Gindy (1997).

The Aim of the Thesis

The aim of the present thesis is the establishment of the controller design for a variable geometry active suspension system using a learning method - namely, a neural network approach.

Chapter 2 discusses the general characteristics, potential benefits and problems of conventional passive suspensions and the active suspension domain. The discussion will lead us further into a consideration of low energy active suspension systems.

Chapter 3 deals firstly with a broad review of learning algorithms, and then it focuses attention on neural network learning control techniques.

The issue to be considered is how a neural network controller can be integrated with a physical model of a non-linear dynamic system. It is divided into three sub-problems which are dealt with in the following three chapters, respectively. In chapter 4, a very simple linear system is set up for the purpose of establishing a structure in which effective on-line learning can take place. In chapter 5, a standard restriction of linear optimal control theory, namely the quadratic cost function form, is removed. The learning technique is applied to the control of a non-linear dynamic system in chapter 6. All simulation results presented in these three chapters are generated by MATLAB and the *Neural Network Toolbox* (Demuth and Beale, 1994).

Chapter 7 describes the design and model development of the variable geometry active suspension system. A feedback control scheme which involves sensors and proportional-plus-differential control gains is set up. Levelling function, body-roll and jacking (bounce) responses, under vehicle body weight changes and lateral force inputs, are simulated via FORTRAN and the computer language, AUTOSIM.

In chapter 8, the neural network controller is integrated with the variable geometry active suspension system developed in chapter 7. The controller learns to reduce the defined cost function to its minimum value. The system performances are compared with those of the passive and proportional-plus-differential controlled systems.

Chapter 9 analyses energy consumption of the variable geometry active suspension system using an actuator model developed via the computer simulation package, SIMULINK. The actuator model involves four parts: motor speed control, motor current control, electric motor circuit and mechanical system. Each part is detailed using corresponding mathematical equations. The simulations provide power and energy consumption of the variable geometry system with either neural network control or proportional-plus-differential control. These results are compared with those of other actively-controlled suspension systems taken from the literature.

The thesis is concluded in chapter 10, which includes a consideration of what further work is necessary to gain a complete understanding of the system.

Chapter 2
Automotive Active Suspension Systems - a Review

To enable people to drive cars for hours without becoming tired, to prevent goods being damaged in transit and to ensure mechanical components remain in good condition, they must be isolated from road disturbance. Passive suspensions, which involve springs and dampers, can be reasonable vibration isolators. Suspensions need to be compliant for the isolation but they need to be stiff for body-attitude-control. Therefore, passive suspensions must involve compromise between vibration isolation and body-attitude-control.

On the other hand, springs and dampers can be replaced by actuator assemblies with feedback control loops. According to Segel (1993), such systems date back to a mechanical-hydraulic active ride-improvement system developed by Hanna and Osbon at the Westinghouse Research Laboratory in 1961. Since then, various types of actively-controlled suspension systems have been proposed and developed in both academic and commercial fields. Especially, commercial interest was stimulated by the application of active devices to race cars (Wright and Williams, 1984 and 1989). These have been reviewed and classified by Sharp and Crolla (1987); Wallentowitz and Konik (1991); and Elbeheiry *et al.* (1995).

The first section of this chapter is devoted to describing the general characteristics of conventional passive suspensions and to active, semi-active, slow-active suspensions and active roll-control systems, which involve actively-controlled anti-roll bars. The second section details potential benefits and problems of both passive and active systems. The third section concentrates on low energy system types, which may have considerable potential for solving one of the main problems of active systems - that of power consumption.

2.1 Passive and Active Systems

2.1.1 Passive Suspensions

To study vibrations in automotive suspension systems, various vehicle models have been developed. For a passenger car, a seven-degree-of-freedom representation (bounce, pitch and roll of the body and hop for each wheel) may be used as shown in Fig. 2.1 (a). On the other hand, a half-car model, as shown in Fig. 2.1 (b), may be used to investigate either the pitch and bounce motions or the roll and bounce motions of the vehicle body.

Nevertheless, the essence of the ride problem can be captured by a quarter-car model, which includes an un-sprung mass representing the wheel and associated components and a sprung mass representing the vehicle body, as shown in Fig. 2.1 (c). Frequency response function gains of body displacement, suspension deflection and tyre deformation for a road displacement input can be derived from the quarter-car model. These functions with different values of spring stiffness, damping and un-sprung mass are shown in Fig. 2.2 to Fig. 2.4 (Wong, 1993). From these figures, conventional passive suspensions have two principal modes with natural frequencies, from 0.5 Hz to 2 Hz and from 10 Hz to 12 Hz. The former mode involves the natural frequency primarily of the sprung mass and the latter involves mainly the natural frequency of the un-sprung mass. Both modes are excited by road disturbance. Especially, the former involves body motion during cornering, accelerating and braking. When the frequency of the road disturbance coincides with one of the natural frequencies, a resonance results. The resonances of the sprung and un-sprung masses are referred to as 'body resonance' and 'wheel-hop resonance', respectively.

As shown in Fig. 2.2, a soft suspension spring provides relatively good vibration isolation. However, the road-holding tends to be better with a stiff suspension spring. It can be seen that there are invariant points in these functions.

Fig. 2.3 shows that light damping provides good vibration isolation and road-holding in the mid-frequency range, good protection of the body and mechanical components from high frequency excitation but poor performances at each resonant condition.

The un-sprung mass is hardly influential near the body resonance, A light un-sprung mass is advantageous close to the wheel-hop resonance but it is disadvantageous at higher frequency range in terms of vibration isolation of the body as shown in Fig. 2.4.

Consequently, conventional passive suspensions with fixed characteristics of springs and dampers compromise over driving conditions.

2.1.2 Active Suspensions

In order to provide the cars with improved vibration isolation, road-holding, body-attitude-control and so on under various driving conditions, the concept of an adaptation of vehicle parameters, involving the spring stiffness and damper rate, emerged. The most effective way to achieve the concept is to replace the spring and damper with an actuator assembly with a feedback control loop. Hardware requirements of such systems are hydraulic pumps, cylinder/piston units, servo-valves, accumulators, filters, pipe-work, oil reservoirs, coolers and so on - there are drawbacks with them: reliability and maintainability, excessive weight, production cost, running cost, packaging and noise.

A fully-active suspension system presented by Sharp and Crolla (1987) is depicted in Fig. 2.5. The driving conditions are continuously monitored by sensors, i.e. the accelerometer mounted on the vehicle body and potentiometer measuring the suspension travel. The signals from the sensors are fed back through the signal conditioning and its output determines an actuator force demand signal. The actuator servo-valve is controlled in order to minimise the error between the demand signal and actuator force measured via the force transducer. The actuator control bandwidth extends to beyond the wheel-hop frequency.

Active suspensions are supervised by micro-computers with the prescribed control strategies. The control strategy employing linear optimal control theory has been studied by Wilson, Sharp and Hassan (1986). The optimal controller was designed to minimise the cost function, which involves the weighted sum of the squares of the vehicle body vertical acceleration, suspension travel and tyre dynamic deformation. The optimal control theory must involve full-states to be fed back to the controller. Therefore, un-measurable states (i.e. tyre deformation and vehicle body height) need to be obtained by some form of estimator such as a Kalman filter. Furthermore, the employment of such linear control theory is somewhat restrictive in connection with a problem containing essential non-linear features, i.e. a finite working space and non-linear characteristics of the tyres. These non-linearities were integrated into the controller design by Gordon, Marsh and Milsted (1990 and 1991).

Preview control, which involves the measurement of the road profile ahead of the front wheels, has been studied by Sharp (1995). Sharp mentioned that a fully-active suspension system with preview control can provide better performance but consumes higher energy than a non-preview active system. A slow-active system (this will be described in sub-section 2.1.4) with the preview control needs longer preview time, which is almost impractical for a contemporary car.

2.1.3 Semi-Active Suspensions

Semi-active suspensions commonly use springs to support the vehicle body in parallel with continuously variable dampers. The springs have fixed characteristics while the dampers are adjusted via electro-rheological fluids, solenoid valves or motors.

The system structure can be expressed as in Fig. 2.5 with the addition of a spring and the actuator replaced by a damper. The feedback control loop can be inherited from the fully-active system. The damper force signal is required to track a demand signal coming from the state measurements and signal conditioning. However, the damper force produced is limited since power can only be dissipated. Therefore, the performance may involve compromise between fully-active and passive systems.

Another type of semi-active suspension system, which involves a variable spring mechanism using interconnected gas/oil spheres, has been developed and fitted to the Citroen XM. The gas/oil spheres (acting as spring/damper units) on each side of one axle are interconnected by opening a firmness regulator valve with an additional gas/oil sphere (central sphere), when the vehicle is in the 'soft' mode. In the 'firm' mode, the firmness regulator valve is in the closed position and each wheel sphere is then independent, while the central sphere is inactive. A control computer collects suspension and vehicle state values and controls the opening and closing of the valves.

2.1.4 Slow-Active Suspensions

Slow-active suspensions are active suspensions with the actuator control bandwidth embracing the body resonant frequencies but not extending to the wheel-hop resonant frequencies.

A type of slow-active system has been presented by Sharp and Hassan (1987) as shown in Fig. 2.6. The system has a limited bandwidth actuator in series with a spring and has a damper in parallel with the two. The actuator becomes rigid beyond its control bandwidth and the system then works as an ordinary passive suspension. Another type of slow-active system has an actuator in series with a spring/damper unit, which is better than the type shown in Fig. 2.6 in energy consumption (Williams and Miller, 1994).

The system can include a D.C. motor with an irreversible leadscrew in series with an air-spring unit (Sharp, and Hassan, 1988). An alternative is a hydro-pneumatic suspension type, which involves gas-filled springs in a hydraulic actuator assembly (Williams, Best and Crawford, 1993).

2.1.5 Active Roll-Control Systems

Active roll-control systems involve rotary actuators mounted in conventional anti-roll bars, and a number of simulation studies of them have been reported (Lang and Walz, 1991; Sharp and Pan, 1993; and Darling and Ross-Martin, 1997).

A full-car model with actively-controlled anti-roll bars is depicted in Fig. 2.7 (presented by Sharp and Pan, 1993). The actuator in each of the front and rear axles twists through the appropriate angle in order to reduce or eliminate vehicle-body-roll during cornering. The control unit receives information from a lateral accelerometer and actuator rotation sensors, processes those signals and commands actuator rotor positions.

Since these systems retain springs and dampers, they work as conventional passive suspensions on a straight road which may be advantageous from an energy point of view. Furthermore, such anti-roll bars can be switched off via by-pass valves round the actuators - this function may provide a superior performance to that of conventional anti-roll bars, especially, on a severe cross-level road. However, the systems cannot influence body-pitch and bounce motions.

2.2 Potential Benefits and Problems

In this section, the various types of suspension described in the last section are summarised in the following features: performance, drawbacks, energy consumption and control problem.

Passive suspensions can be reasonable vibration isolators with soft springs but the spring stiffness must be high enough to control the body-attitude. Light damping provides good performance in the mid-frequency and high-frequency ranges (between body and wheel-hop resonant frequencies and above wheel-hop resonance) but not around resonant frequencies. Therefore, some compromise between vibration isolation and body-attitude-control must be involved.

On the other hand, fully-active suspensions can provide good vibration isolation, road-holding, body-attitude-control and levelling. Semi-active suspensions are able to achieve similar performance with controlled-dampers. In practice, however, both fully-active and semi-active suspensions provide poor performance in the high-frequency range (beyond wheel-hop resonant frequency) due to time-delay, friction and noise. The problem can be solved by employing slow-active suspensions, which isolate the bodies from road disturbance passively and control body-attitude actively within the actuator bandwidth of 3 - 5 Hz. However, they cannot achieve as good body-control as fully-active suspensions. Active roll-control systems can eliminate body-roll but body-pitch and bounce motion remain.

Fully-active, slow-active and active roll-control systems include drawbacks: complex arrangement, reliability and maintainability, expensive production costs, excessive weight and noise. Although semi-active suspensions require reduced hardware arrangement, the noise problem still remains.

Williams and Miller (1994) reported that a fully-active suspension consumes higher energy than a slow-active suspension on a straight road but lower energy during cornering. The fully-active suspension consumes 30 % more energy than the slow-active suspension under test conditions which involve driving on a straight road at 20 m/s and a 0.35 g cornering every 10 seconds. Hillebrecht *et al.* (1992) indicated that a slow-active suspension consumes 15 % more fuel than conventional passive suspension. These higher potential running costs have been a major contributing factor to the lack of commercial interest for active suspension systems.

Control problems include non-linearities (i.e. tyre characteristics), limited state feedback, selection of sensors, mechanical limitations (i.e. suspension working space) and specific external disturbance (i.e. road profiles or lateral force due to wind). Gordon, Marsh and Wu (1993) and Howell, Frost and Gordon (1996) have used learning techniques to cope with these problems.

The above features are summarised in Table 2.1 (a) for passive, fully-active and semi-active systems and (b) for slow-active and active roll-control systems. Consequently, some types of active suspension systems, which are economical in energy terms, are of great interest.

2.3 Low energy active suspension systems

A type of low energy active suspension system has been developed and tested in Delft University of Technology (Venhovens, Knaap and Pacejka, 1992; Knaap, Venhovens and Pacejka, 1994). This low energy concept involves a pre-loaded spring attached to an adjustable lever as shown in Fig. 2.8. The force, F_D, on the lower wishbone can be varied by adjusting the length of the lever in order to compensate the load transfer, ΔP, caused by body-pitch and body-roll. This concept was practically integrated into a cone mechanism (see Fig. 2.9), in which the adjustable crank varies the influence of the pre-loaded spring on the lower wishbone without changing the pre-load.

This system works passively for vibration isolation and actively for body-control with the actuator bandwidth of 5 Hz. The system consumed an average power of 770 W in a double lane change test with a maximum lateral acceleration of 7.5 m/s^2. This value seems to be very economical in comparison with the power consumption of 3300 W in a transient handling test (whose maximum lateral acceleration was 6.9 m/s^2) with a slow-active suspension system (Williams and Miller, 1994).

Another development of a low energy active suspension system, which involves a buckling spring element and an actively-controlled leverage ratio at each wheel, has been reported by Leighton and Pullen (1994) and Leighton (1995). The mechanical arrangement for a single wheel station is depicted in Fig. 2.10. The wheel force is generated by a spring element, which has a very low effective rate when the spring displacement exceeds the buckling point. The vehicle body was isolated from road disturbance passively and its attitude was controlled by moving one end of each spring element using D.C. motor/ballscrew units with the bandwidth of 4 Hz. The system requires power levels of 100 W per wheel as described by Leighton.

2.4 Conclusions

Low energy active suspension system types seem to provide a significant contribution to the reduction of running costs.

On the other hand, new mechanical design problems are posed by the system principle, which involves: variable geometry, mechanical components, materials, change of leverage ratio, position and direction of actuation, power requirement and bandwidth considerations.

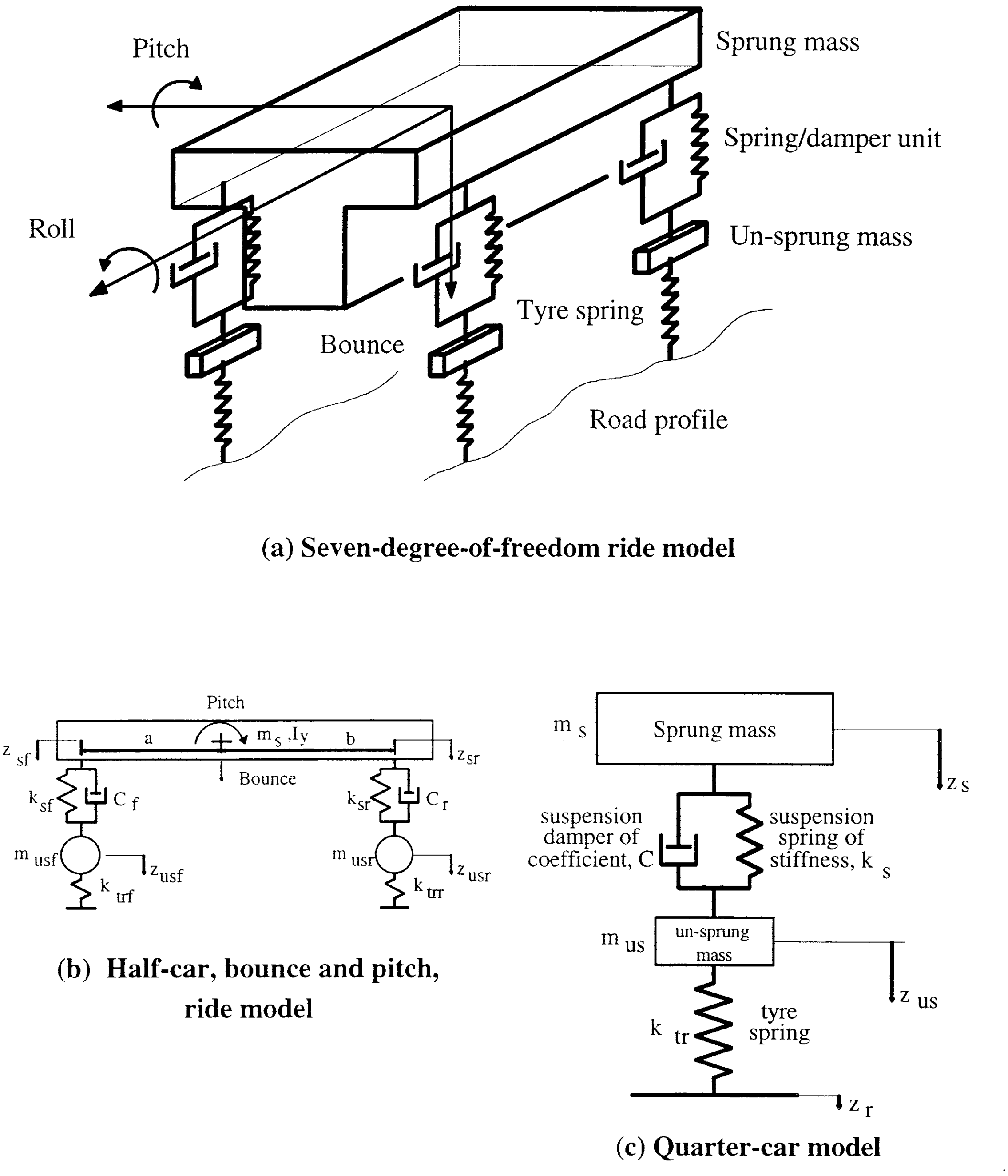
Furthermore, the control system design problems are novel, since variable geometry implies non-linearity and mechanical limitations. In order to deal with non-linear control system design, learning control is a valid approach and this is surveyed in the next chapter.

Table 2.1 (a) Comparison between passive, fully-active and semi-active suspension systems in the various features: performance and advantages, drawbacks, energy consumption and control problems

Systems / Features	Passive	Fully-active	Semi-active
Performance and advantages	• Economical; • Reasonable vibration isolation; • High reliability.	• Good vibration isolation and road-holding in theory but not in practice (especially in the high-frequency range) due to time-delay, friction and noise; • Good body-attitude-control; • levelling function.	• Good vibration isolation and road-holding in theory but not in practice (especially in the high-frequency range) due to time-delay, friction and noise; • Economical hardware arrangement.
Drawbacks	• Vibration isolation and body-attitude-control must be traded-off.	• Complex arrangement; • Excessive weight; • Expensive production cost; • Expensive running cost; • Low reliability and maintainability.	• Noise
Energy consumption	• Economical	• Very high for vibration isolation on a rough road; • Relatively low for body-attitude-control.	• Economical
Control problems	• Since, the characteristics are fixed, it does not suit wide range of driving conditions; • The characteristics can be adaptive via switches.	• Non-linearities; • Limited state-feedback; • selection of sensors; • mechanical limitations.	• Non-linearities; • Limited state-feedback; • selection of sensors; • mechanical limitations.

Table 2.1 (b) Comparison between slow-active suspension and active roll-control systems in the various features: performance and advantages, drawbacks, energy consumption and control problems

Systems Features	Slow-active	Active roll-control
Performance and advantages	• Vibration isolation with a passive manner; • Body-attitude-control but not as good as a fully-active system; • levelling function.	• Vibration isolation with a passive manner; • Body-roll reduction or elimination.
Drawbacks	• Complex arrangement; • Excessive weight; • Expensive production cost; • Expensive running cost; • Low reliability and maintainability.	• Complex arrangement; • Excessive weight; • Expensive production cost; • Relatively low running cost; • Low reliability and maintainability.
Energy consumption	• Economical for vibration isolation on a rough road; • Very high for body-attitude-control.	• Economical for vibration isolation on a rough road; • Relatively low for body-attitude-control.
Control problems	• Non-linearities; • Limited state-feedback; • selection of sensors; • mechanical limitations.	• Non-linearities; • selection of sensors; • mechanical limitations.

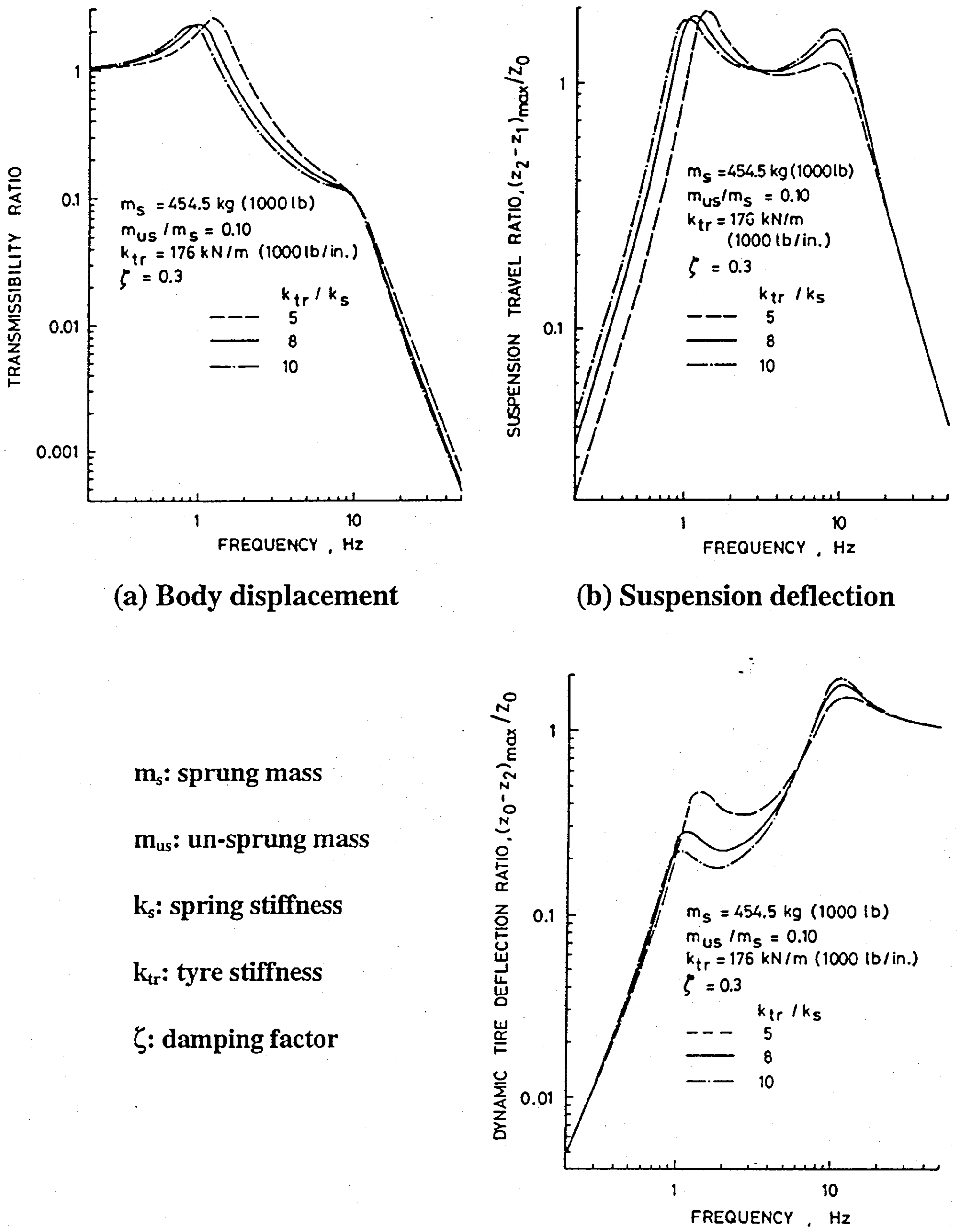


(a) Seven-degree-of-freedom ride model

(b) Half-car, bounce and pitch, ride model

(c) Quarter-car model

Fig. 2.1 Three basic ride models used to study vibrations in automotive suspensions: (a) Seven-degree-of-freedom ride model, (b) half-car, bounce and pitch, ride model and (c) quarter-car model.



(a) Body displacement

(b) Suspension deflection

m_s: sprung mass

m_{us}: un-sprung mass

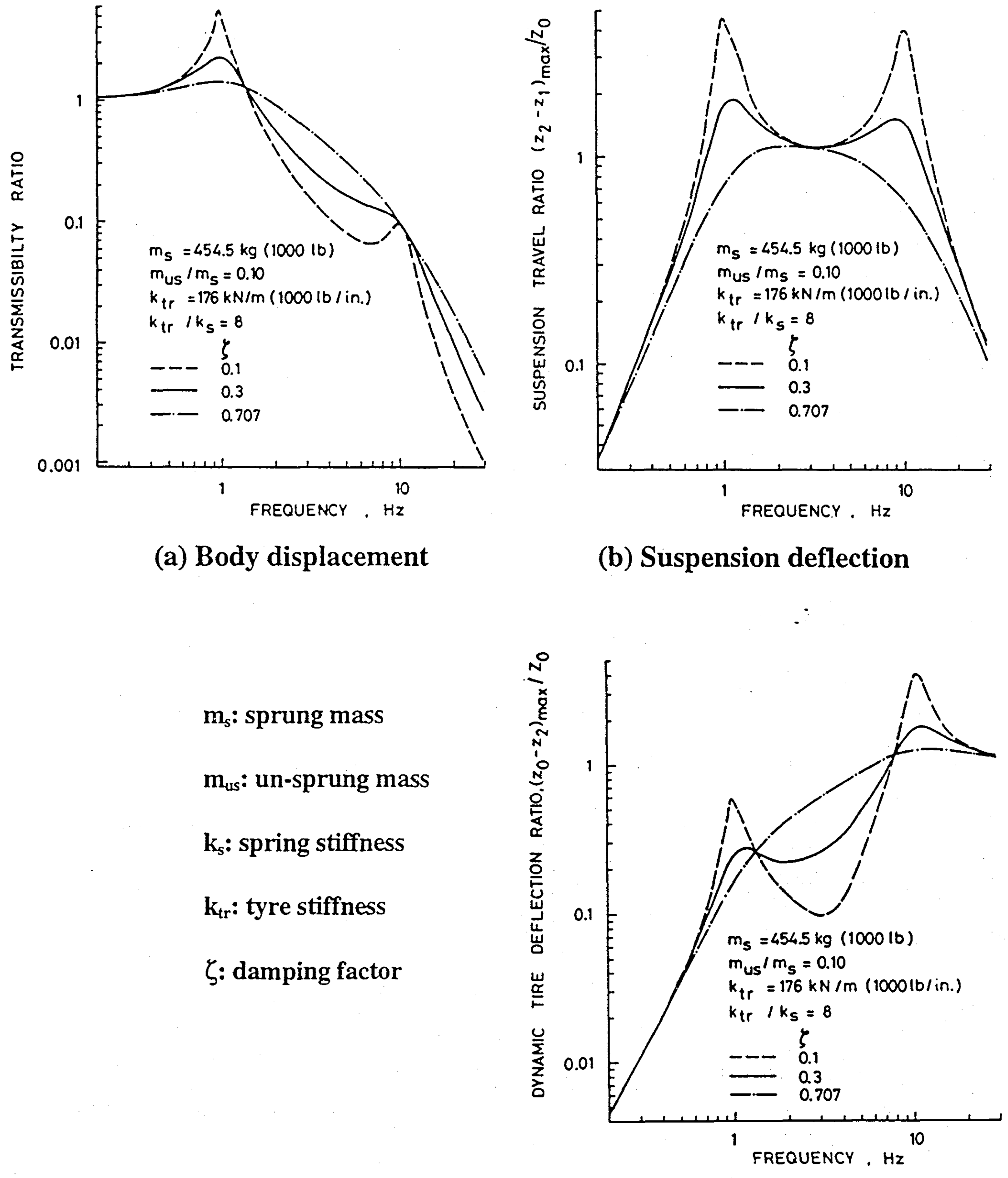
k_s: spring stiffness

k_{tr}: tyre stiffness

ζ: damping factor

(c) Tyre deformation

Fig. 2.2 Frequency response function gains of (a) body displacement, (b) suspension deflection and (c) tyre deformation for a road displacement input, with three different spring stiffness (presented by Wong, 1993).



(a) Body displacement

(b) Suspension deflection

m_s: sprung mass

m_{us}: un-sprung mass

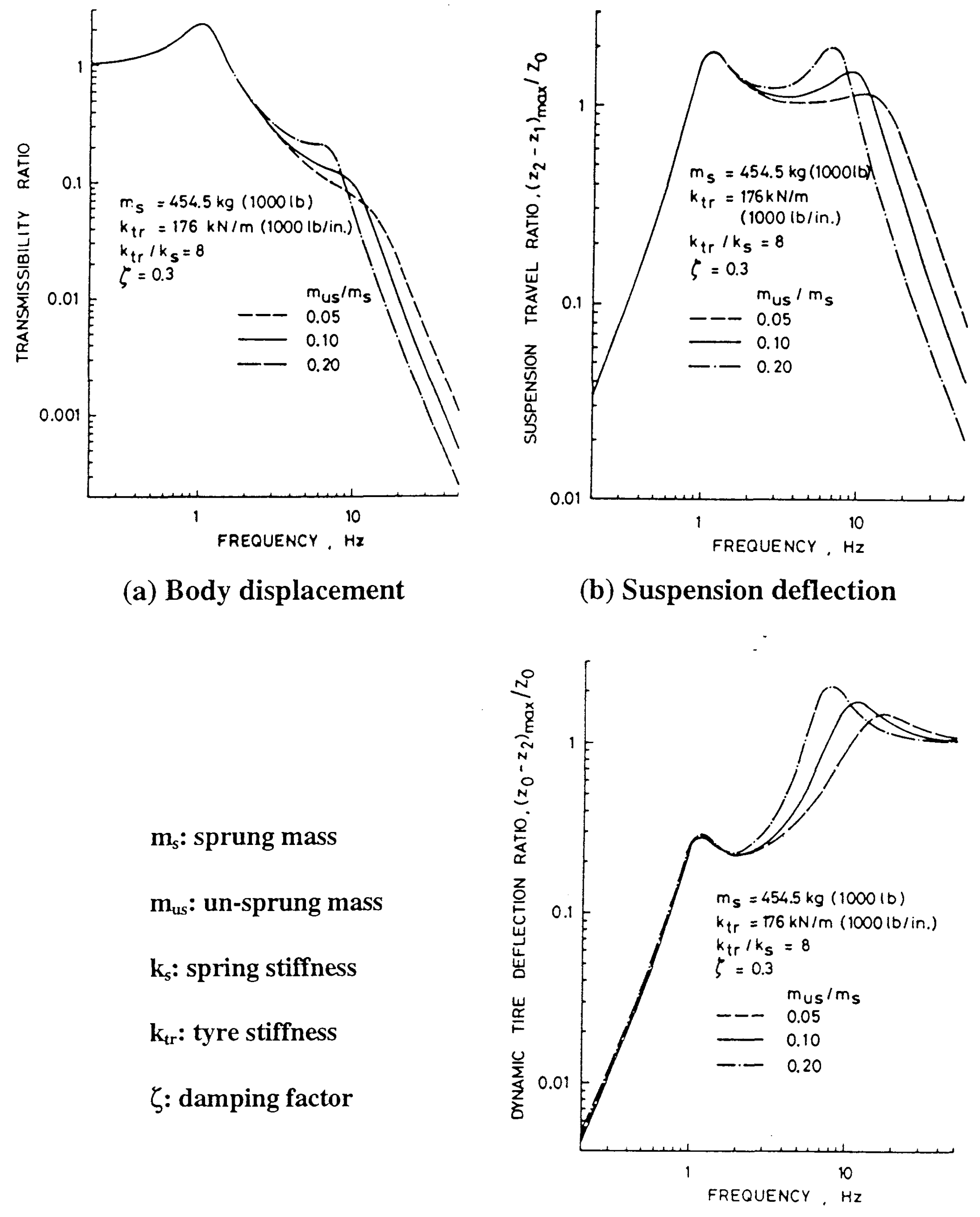
k_s: spring stiffness

k_{tr}: tyre stiffness

ζ: damping factor

(c) Tyre deformation

Fig. 2.3 Frequency response function gains of (a) body displacement, (b) suspension deflection and (c) tyre deformation for a road displacement input, with three different damping factors (presented by Wong, 1993).



(a) Body displacement

(b) Suspension deflection

m_s: sprung mass

m_{us}: un-sprung mass

k_s: spring stiffness

k_{tr}: tyre stiffness

ζ: damping factor

(c) Tyre deformation

Fig. 2.4 Frequency response function gains of (a) body displacement, (b) suspension deflection and (c) tyre deformation for a road displacement input, with three different un-sprung masses (presented by Wong, 1993).

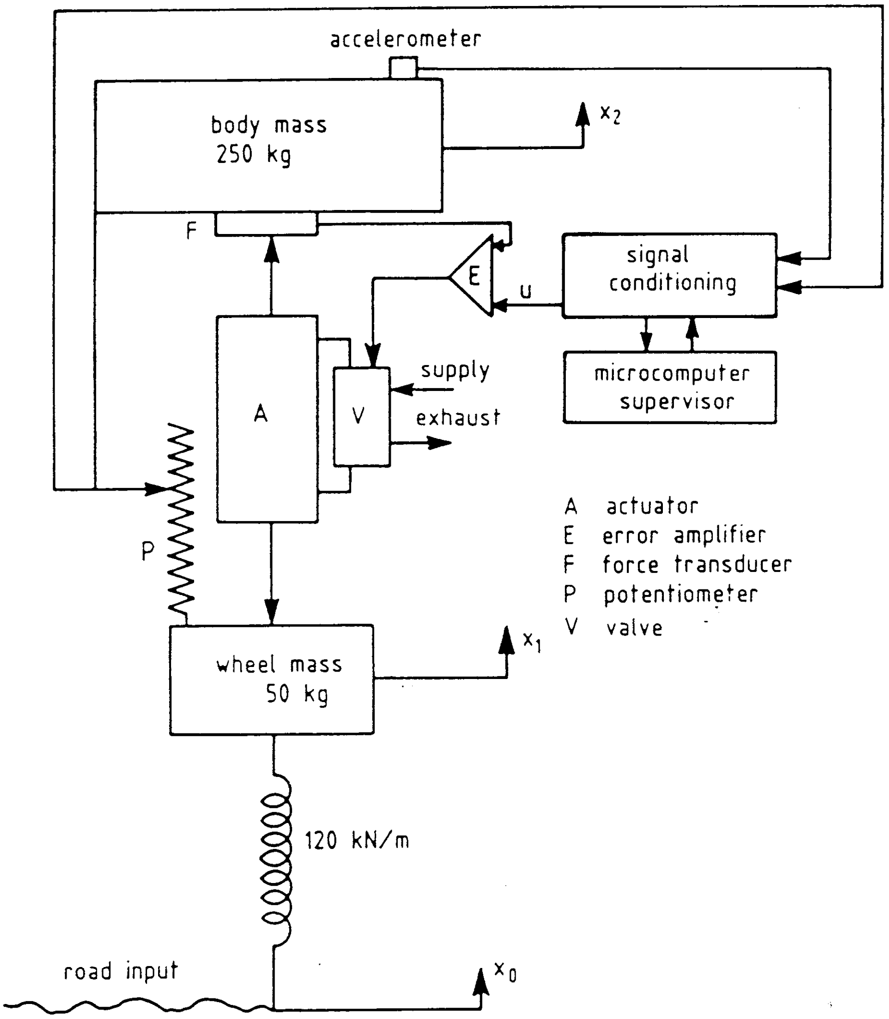


Fig. 2.5 Fully-active suspension system (presented by Sharp and Crolla, 1987)

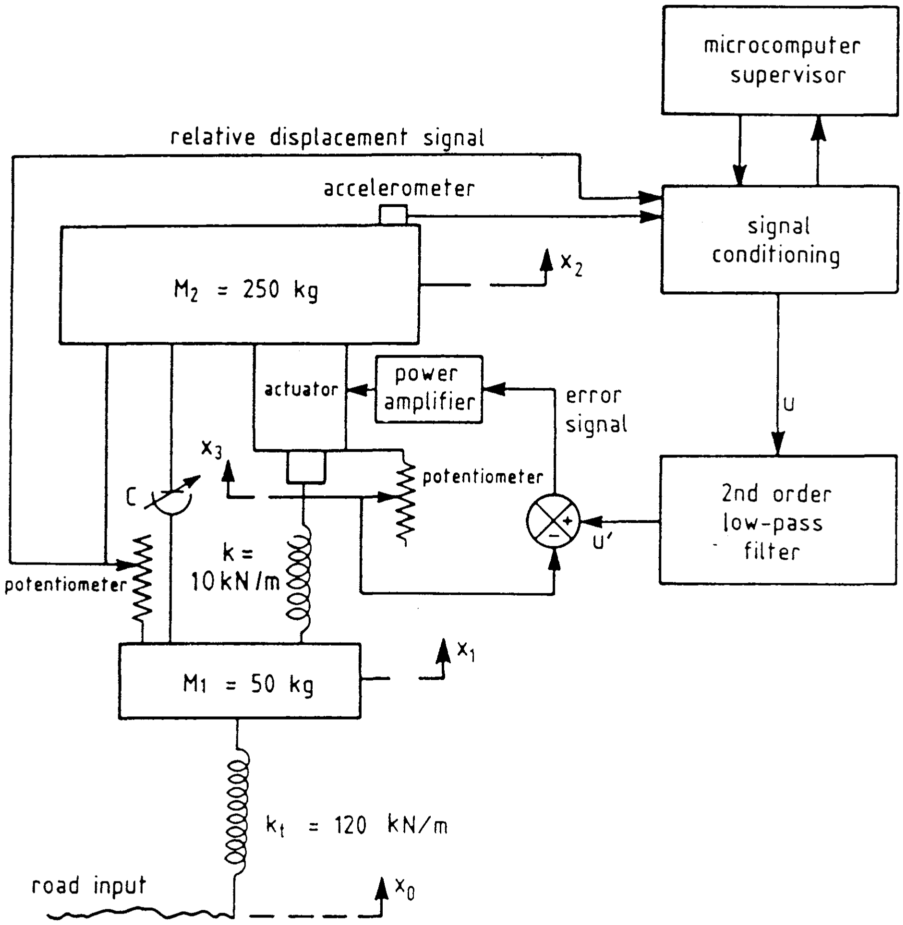


Fig. 2.6 Slow-active suspension system (presented by Sharp and Hassan, 1987)

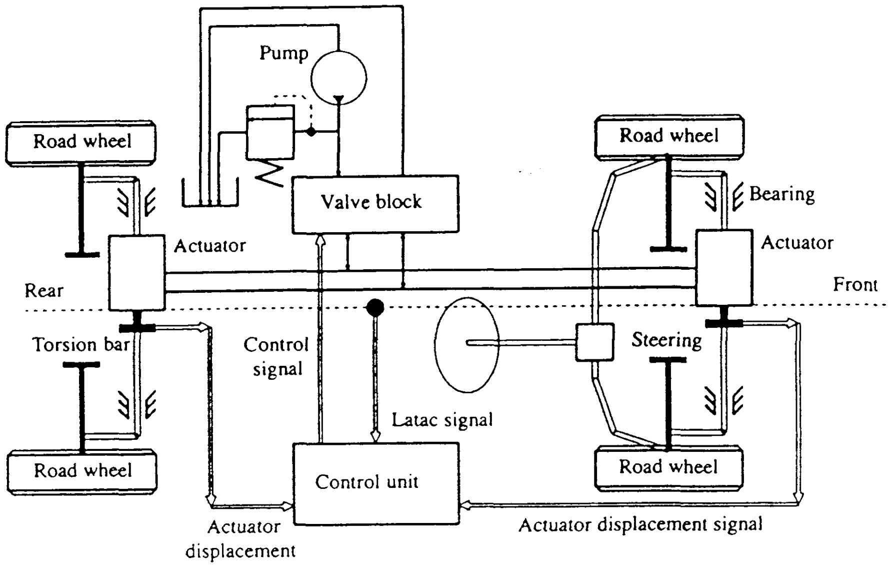


Fig. 2.7 Diagrammatic active roll-control system with interconnected actuators (presented by Sharp and Pan, 1993)

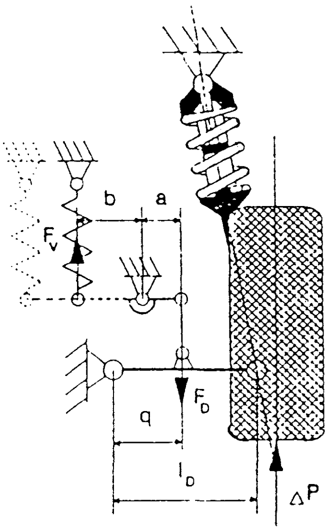


Fig. 2.8 Variable compensating force with elastic property by means of an adjustable transmission mechanism (presented by Venhovens, Knaap and Pacejka, 1992)

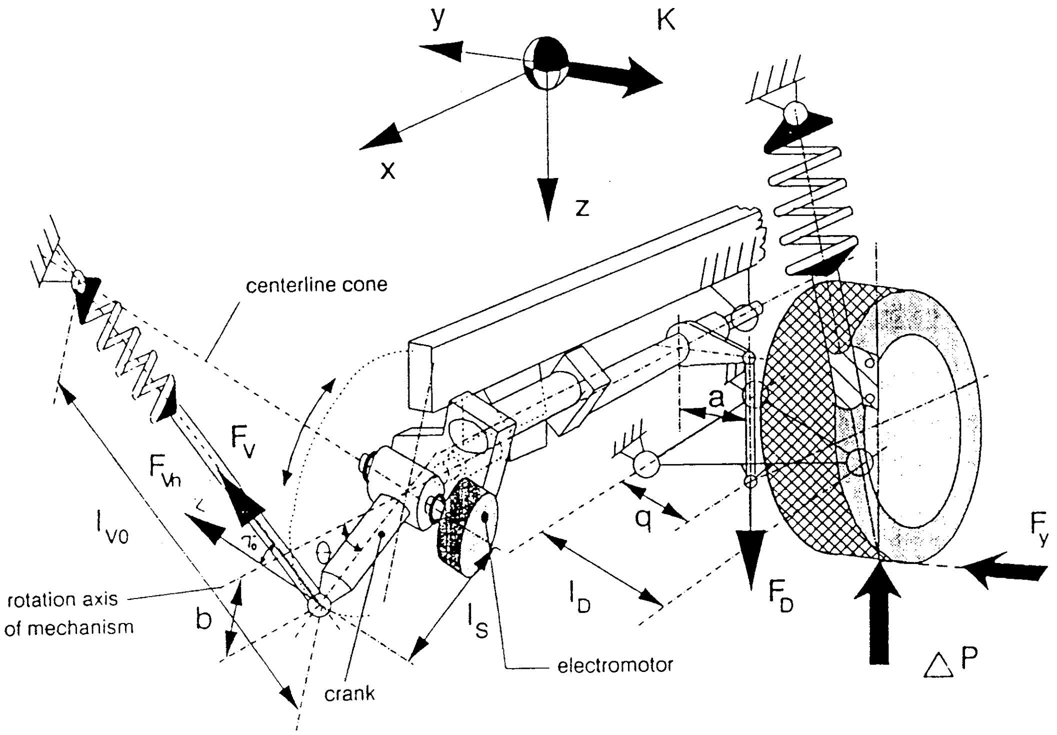


Fig. 2.9 Cone mechanism built into a left front suspension (presented by Venhovens, Knaap and Pacejka, 1992)

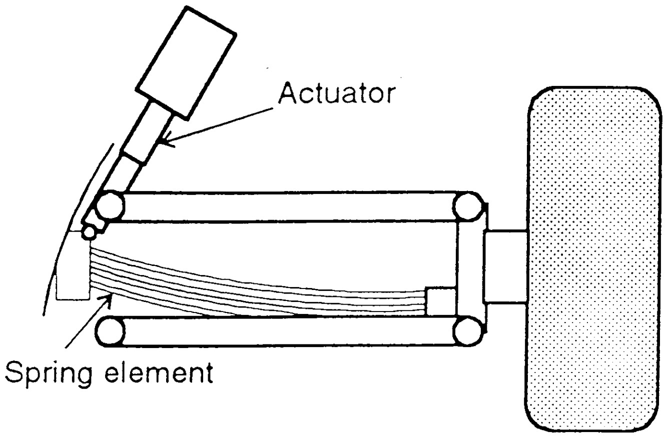


Fig. 2.10 Variable geometry wishbone linkage (presented by Leighton, 1995)

Chapter 3
Learning Method

This chapter deals with a survey of learning methods primarily from the point of view of non-linear optimal control of automotive active suspension. In the first section, a common learning process is described using a flow diagram. The second section presents five learning algorithms, automata, SOFLIC (Self-Organising Fuzzy Logic Intelligent Control), neural networks, neuro-fuzzy and genetic algorithms. The third section concentrates on several well-established neural network learning control techniques.

3.1 Background

Many control engineers have to deal with system complexities and non-linearities. Classical methods, such as linear control theory, have to compromise on highly idealised situations: the plants, which have to be controlled, and the controllers are linear without limitations; external disturbance inputs are expressed in the form of random white noise; and quadratic forms of cost function are used for performance criteria. Considerable study has been carried out into non-linear optimal control over the past thirty years based on non-linear plants, specified disturbance inputs, non-quadratic forms of cost function and non-linear input/output relationships of controllers via polynomial operators.

On the other hand, modern techniques in intelligent frameworks (automata, fuzzy logic, artificial neural networks and genetic algorithms) have been used extensively for control engineering. The control, which is so-called intelligent control, was introduced by Harris, Moore and Brown (1992), and several examples of intelligent control have been edited by Harris (1994). A review paper of neural network applications in the automotive field has been published by El-Gindy and Palkovics (1993), and a more recent review involving fuzzy logic applications was authored by Zadeh, Fahim and El-Gindy (1997).

Learning is one of the main properties of intelligent control and it enables an intelligent system to be optimised via iterative trials. If the plant is non-linear, learning may be one of the valid approaches to obtain an optimal controller.

There are many different algorithms within the learning method. Most algorithms have a common learning process, which involves the following stages:

(i) initialisation: setting up a framework (e.g. defining a network structure) with initial learning states (e.g. an initial set of weighting parameters);

(ii) trial: running with a set of training data;

(iii) evaluation: evaluation via a performance criterion (e.g. cost function);

(iv) modification: update of the learning states and convergence to a minimum error or cost;

(v) testing: running with sets of test data which are different from the set of training data;

(vi) refinement: the framework is refined (e.g. by changing of network structure).

The above process starts with 'initialisation'. Three stages, 'trial', 'evaluation' and 'modification', iterate with the training data until the system satisfies the given performance criteria. The system is then tested with the test data and evaluated. If the system satisfies the performance criteria, the process will finish; otherwise, the framework is refined and the process restarts with 'initialisation'. This process is illustrated in the flow diagram shown in Fig. 3.1.

3.2 Learning Algorithms

There are three basic learning algorithms, learning automata, SOFLIC (Self-Organising Fuzzy Logic Intelligent Control) and neural networks. Complex algorithms such as 'neuro-fuzzy' and other algorithms such as 'genetic algorithms' have also been studied over a wide area.

3.2.1 Learning Automata

Learning automata can be used to design an optimal controller without any explicit system model. The role of the automaton is to select one controller in a set of controllers having different control parameter values. The system runs with each controller under a given training environment and its performance is evaluated by a cost function. After each run, the probability factor of a particular controller being chosen is updated via a penalty-reward scheme, so that the best performing controller will be chosen in the end. Learning proceeds in stages, such that a set of different controllers is investigated at each stage and the set is then refined for the next learning stage until the cost function is converged to its minimum value.

Learning automata have been used for an optimal controller design of an automotive active suspension system by Gordon, Marsh and Wu (1993). More recently, the learning efficiency of the algorithm has been developed by Howell, Frost and Gordon (1996). For example, when a feedback controller has m parameters and each parameter has r possible discrete values, there are r^m different controllers. Therefore, if r and m are large, the learning time will be long. In order to cope with this problem, Howell *et al.* demonstrated interconnected automata, which involved more than one automaton assigned to the controller parameters.

3.2.2 SOFLIC (Self-Organising Fuzzy Logic Intelligent Control)

Fuzzy logic is a method for mimicking the human reasoning process. A fuzzy logic interface uses a small number of rules (based on *a priori* knowledge) and provides a smooth output via a process of interpolation using a set of membership functions. SOFLIC includes such a reasoning process and a learning process, which involves a defined performance index to evaluate the system output and to modify the rules and the membership functions. The algorithm was introduced in the literature mentioned above (Harris, Moore and Brown, 1992; Harris *ed.*, 1994).

Lin, Lu and Padovan (1993) have attempted to use a similar algorithm, which involves the modification of an output scaling factor, for an automotive active suspension controller. The controller, which is called a self-tuning fuzzy logic controller, consists of primary and secondary fuzzy interfaces as shown in Fig. 3.2. The former interface performs a basic control function to provide ride comfort by minimising the error between the body vertical acceleration and its reference signal, while the latter interface is used to tune the former. An absolute maximum error and its change within the observation time are fed through the secondary interface, whose output determines the modification of the scaling factor of the output membership function in the primary interface.

3.2.3 Neural Networks

Neural networks simply approximate to a human's information processing structure but actually the biological structure has not been clarified. Today, neurocomputing is incorporated in the computer package, MATLAB (Demuth and Beale, 1994) and the corresponding literature has been published (Moscinski and Ogonowski, 1995; Hagan, Demuth and Beale, 1996).

A neural network is a parallel-distributed information processing structure involving a number of processing elements. A processing element, having multiple-input and single output is illustrated in Fig. 3.3. The output of the element is given by

$$u^* = \sum_{h=1}^{S_h} w_h \; a_h + b, \tag{3.1}$$

$$u = F(u^*), \tag{3.2}$$

where a_h is h th input, w_h is h th weighting parameter of the element, b is the threshold, and $F(u^*)$ is the output function. There are four basic output transfer functions: linear, threshold, sigmoid and radial basis, as shown in Fig. 3.4 (Demuth and Beale, 1994).

A basic multi-layer neural network is depicted in Fig. 3.5. The network involves input, hidden and output layers, having S_p, S_h, and S_r processing elements, respectively. Each element is connected to the other elements in the next layer through the weighting parameters. A weighting parameter between p th element at the input layer and h th element at the hidden layer is denoted by $w^H{}_{h,p}$; similarly, $w^R{}_{r,h}$ denotes a weighting parameter between h th element at the hidden layer and r th element at the output layer.

When a set of training data $\{x(n), \hat{a}^R(n)\}_{n=1}^{N}$ is given, the network is assessed by a cost function of the form:

$$J = E(e^R(n)) = \frac{1}{N}\sum_{n=1}^{N} e^R(n), \tag{3.3}$$

$$e^R(n) = \frac{1}{2}\sum_{r=1}^{S_r} (\hat{a}_r^R(n) - a_r^R(n))^2, \tag{3.4}$$

where E denotes the time averaging within the training data. The weighting parameters are updated via a gradient descent method to reduce the cost, J. The weight update is expressed by

$$w(\text{new}) = w(\text{old}) + \Delta w, \tag{3.5}$$

$$\Delta w = -\delta \frac{\partial J}{\partial w}, \tag{3.6}$$

where δ is the learning rate to convergence of learning.

The weighting parameter update, Δw, involves the update between the hidden and output layers

$$\Delta w^R{}_{r,h} = -\delta E\left(\frac{\partial e^R}{\partial a*^R{}_r} \frac{\partial a*^R{}_r}{\partial w^R{}_{r,h}} \right) \tag{3.7}$$

$$= -\delta E(D^R{}_r \; a^H{}_h),$$

where the learning signal of the output layer, $D^R{}_r$, is given by

$$D^R{}_r = \frac{\partial e^R}{\partial a*^R{}_r} = \frac{\partial e^R}{\partial a^R{}_r} \frac{\partial a^R{}_r}{\partial a*^R{}_r} \tag{3.8}$$

$$= -(\hat{a}^R{}_r(n) - a^R{}_r(n))F'(a*^R{}_r),$$

where $F'(a*)$ denotes the derivative of $F(a*)$ with respect to $a*$.

In a similar manner, the weighting parameter update, Δw, involves the update between the input and hidden layers

$$\Delta w^H{}_{h,p} = -\delta E\left(\frac{\partial e^R}{\partial a*^H{}_h} \frac{\partial a*^H{}_h}{\partial w^R{}_{h,p}} \right) \tag{3.9}$$

$$= -\delta E(D^H{}_h \; x_p),$$

where the learning signal of the hidden layer, $D^H{}_h$, is given by

$$D^H{}_h = \frac{\partial e^R}{\partial a*^H{}_h} = \sum_{r=1}^{S_r} \frac{\partial e^R}{\partial a*^R{}_r} \frac{\partial a*^R{}_r}{\partial a^H{}_h} \frac{\partial a^H{}_h}{\partial a*^H{}_h}, \tag{3.10}$$

$$\therefore D^H{}_h = \sum_{r=1}^{S_r} D^R{}_r \; w^R{}_{r,h} \; F'(a*^H{}_h). \tag{3.11}$$

The learning signal at the hidden layer involves the signal from the output layer as described in equation (3.11). This expresses that the learning signal is processed from the end of the network backwards to the beginning. This is called back-propagation learning algorithm. In general, neural networks have fixed network structures and input/output relationships are created by learning.

3.2.4 Neuro-Fuzzy

As mentioned in the above two subsections, the main advantage of using fuzzy logic is that an *a priori* knowledge can be incorporated into the system but learning is difficult. On the other hand, neural networks are capable of learning, but representing knowledge and extracting knowledge from networks are difficult.

Neuro-fuzzy combines learning functions of neural networks and fuzzy knowledge representation. Kiguchi and Fukuda (1996) have used a fuzzy-neural force controller for a planar robot manipulator (see Fig. 3.6). The controller involves a fuzzy environment evaluator and a network controller. The network's architecture is based on a fuzzy interface, which includes fuzzifier, rule and defuzzifier layers. The number of elements at the rule layer is the same as the number of fuzzy rules.

3.2.5 Genetic Algorithms

Genetic algorithms are based on natural genetics, involving randomised information exchanges among survivors. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the survivors of the old (crossover); an occasional new part is tried for good measure (mutation). Yeh, Lu and Chen (1994) have utilised a genetic algorithm to optimise a fuzzy logic controller for a semi-active suspension system. In their work, each string is decoded into fuzzy membership functions and the corresponding rule table entries and the system performance is evaluated by a cost function. The better performing strings are more likely to survive for the next generation. Crossover and mutation among the survivors create a new population. The generations proceed until converged.

Overall, the above three basic learning algorithms, automata, SOFLIC and neural networks are specified through the learning process as shown in Table 3.1. The neuro-fuzzy and genetic algorithms are specified in a similar manner as shown in Table 3.2. From these tables, automata and genetic algorithms have a similar feature, that several candidates are compared, one or more than one good performing candidates are selected and the population is refined for the next trial. The other algorithms involve non-linear input/output functions. Neural networks adjust the functions via the gradient descent method, while SOFLIC and neuro-fuzzy are based on *a priori* knowledge.

Neural networks have non-linearities and learning capability. When we have a newly developed non-linear plant (i.e. variable geometry active suspension system), a neural network may be a good approach to controller design. The next section takes up several well-established neuro-control techniques, which relate to automotive active suspensions.

3.3 Neural Networks for Control

When a dynamic plant, which can be non-linear and/or unknown, lies between a neuro-controller and the plant output, the problem is how the learning signal is sent from the plant output to the controller. Nguyen and Widrow (1991) suggested the following technique, which might be one of the solutions to the problem. The control structure is illustrated in Fig. 3.7. Firstly, a neural network mimics the dynamic plant as a neuro-emulator. Secondly, the neuro-controller (it has not been trained) runs the neuro-emulator from an initial state to the desired state in the representative time steps. Thirdly, the learning signal, which involves the error between the output from the emulator and desired output, is back propagated from the end of the run to the beginning through time (see Fig. 3.8). The weighting parameters are updated for each run.

Moran and Nagai (1992) have utilised a similar technique for an optimal preview control of vehicle rear suspension. The training process is depicted in Fig. 3.9. When the neuro-controller is trained, the neuro-vehicle (it has already been trained to mimic the vehicle model) is used to back propagate the learning signal from the vehicle output to the neuro-controller in order to calculate the derivatives of the cost function with respect to the neuro-controller's weighting parameters.

In contrast to the above techniques, an alternative has been studied by Narendra and Parthasarathy (1991) to cope with the similar problem that a dynamic plant lies between a neuro-controller and the plant output. The system structure is illustrated in Fig. 3.10 (a). The plant is described by a difference equation of the form: $y_p(k+1) = N_f(y_p(k))+N_g(y_p(k))$, where N_f and N_g can be either non-linear functions (for known plant) or neural network functions (for unknown plant). A multi-layer neural network, N_c, is used as a feedback controller and its weighting parameters, θ_j, are updated in order to minimise the error between the plant output and the output from the reference model. The derivatives of the plant output with respect to the weighting parameters are calculated via the gradient generating circuits and general back-propagation as shown in Fig. 3.10 (b).

Feldkamp *et al.* (1992) have utilised such gradient computations to train a neuro-controller for an automotive active suspension system. The full system is depicted in Fig. 3.11. A road disturbance input is applied to the plant and its states are fed back through the neuro-controller. The output from the controller determines the actuator force, which is applied to both the plant and an identification network. The system performance is evaluated by a cost function, which involves the vehicle states. The neuro-controller learns to reduce the cost function using the derivatives of the states with respect to the weighting parameters. The derivatives are obtained by a process of gradient calculation.

Overall, it is reasonable to assume that neural networks with gradient computations have the following possibilities:

(a) the plants can be non-linear; and if the plants are known, the neuro-emulators are not required;

(b) general back-propagation is used when the controllers have multi-layered structures;

(c) cost functions, which have either quadratic or non-quadratic forms, must be used for training;

(d) specified external disturbance inputs are applied to the systems.

However, a negative point is that the number of the gradient generating circuits is the same as the number of weighting parameters. Therefore, we have to compromise between the number of parameters and the calculation time.

3.4 Conclusions

Neural network learning controllers are often developed using neural network emulators of the corresponding plants, in order to allow back-propagation of errors through the plants to the controllers, as basis of the learning process. Replacement by a neural network of a plant already known by its fundamental describing equations is considered disadvantageous, since the network is bound not to mimic the plant precisely over its full range of operating conditions. This will be avoided in the present work.

On the other hand, a neural network controller with gradient computations may be capable of learning to control a non-linear system, without employing a neural network emulator. However, this technique is essentially novel as an application to variable-geometry active suspension control.

Nevertheless, before turning to the subject, the problem can be separated into three case-studies in order to establish an understanding of how neural network controllers can be applied to optimise the performance of vehicle suspensions (either linear or non-linear suspensions), reducing general forms of cost functions to minimum values, under specified external disturbance inputs (e.g. road conditions, either sinusoidal or random; or lateral forces during cornering). The three case-studies are specified in Table 3.3. These case-studies are taken up in chapter 4, chapter 5 and chapter 6, respectively.

Table 3.1 Processes of learning algorithms: automata; SOFLIC; and neural networks

Algorithm / Process	**Automata**	**SOFLIC**	**Neural Networks**
Initialisation	• Setting up a training environment. • Setting up a set of controllers having different control parameter values.	• Setting up a training environment. • Setting up an initial fuzzy rules and membership functions defined by *a priori* knowledge.	• Setting up a set of training data. • Making a network structure. • Setting up an initial set of weighting parameters.
Trial	• Running the system with each controller under the training environment over a representative time.	• Running under the training environment over a representative time.	• Running with the set of training data over a representative time.
Evaluation	• Evaluation for each controller using a cost function. • Selecting the most probable controller using a penalty-reward scheme.	• Evaluation using a performance index defined by *a priori* knowledge.	• Evaluation using a cost function.
Modification	• Refinement of the set according to the best performing controller.	• Modification of the rules and shapes of the membership functions.	• Change of the weighting parameters via back-propagation algorithm.
Testing	• The best controller is tested under a testing environment.	• Running under a testing environment.	• Running with a set of testing data.
Refinement of Framework	• The size of set and the length of representative time are changed.	• The number of rules and the number of membership functions are changed.	• The network structure is changed. • The number of processing elements in the network is changed.

Table 3.2 Processes of learning algorithms: genetic algorithms and neuro-fuzzy

Algorithm Process	Genetic algorithms	Neuro-Fuzzy
Initialisation	• Setting up a training environment. • Setting up an initial population of strings.	• Setting up a set of training data. • Setting up initial fuzzy rules and membership functions defined by *a priori* knowledge. • Making a network structure based upon the rules and membership functions defined.
Trial	• Decode each string into the system. • Running the system under the training environment over a representative time.	• Running with the set of training data over a representative time.
Evaluation	• Evaluation using a cost function. • Selecting good performing strings according to the cost.	• Evaluation using a cost function.
Modification	• Crossover and mutation among the survived strings, creating new population for the next trial.	• Change of the weighting parameters via back-propagation algorithm.
Testing	• The converged population is tested under a testing environment.	• Running with a set of testing data.
Refinement of Framework	• The number of strings in a population and the number of pieces in a string are changed.	• The number of rules and the number of membership functions are changed.

Table 3.3 Specifications of case-studies

simulation cases factors	Case 1	Case 2	Case 3
Plant model	Linear 1/4 car suspension model	Linear 1/4 car suspension model	Non-linear mass-spring-damper system (including non-linear spring rate)
Controller	A Linear processing element	Three layer Neural Network	Three layer Neural Network
External disturbance	Road vertical velocity, cosine wave single frequency	Road vertical velocity, random white noise	Road vertical velocity, cosine wave single frequency
Cost function	Quadratic form	Non-quadratic form	Non-quadratic form
Optimisation process	Gradient computation	Gradient computation and back -propagation	Gradient computation and back -propagation

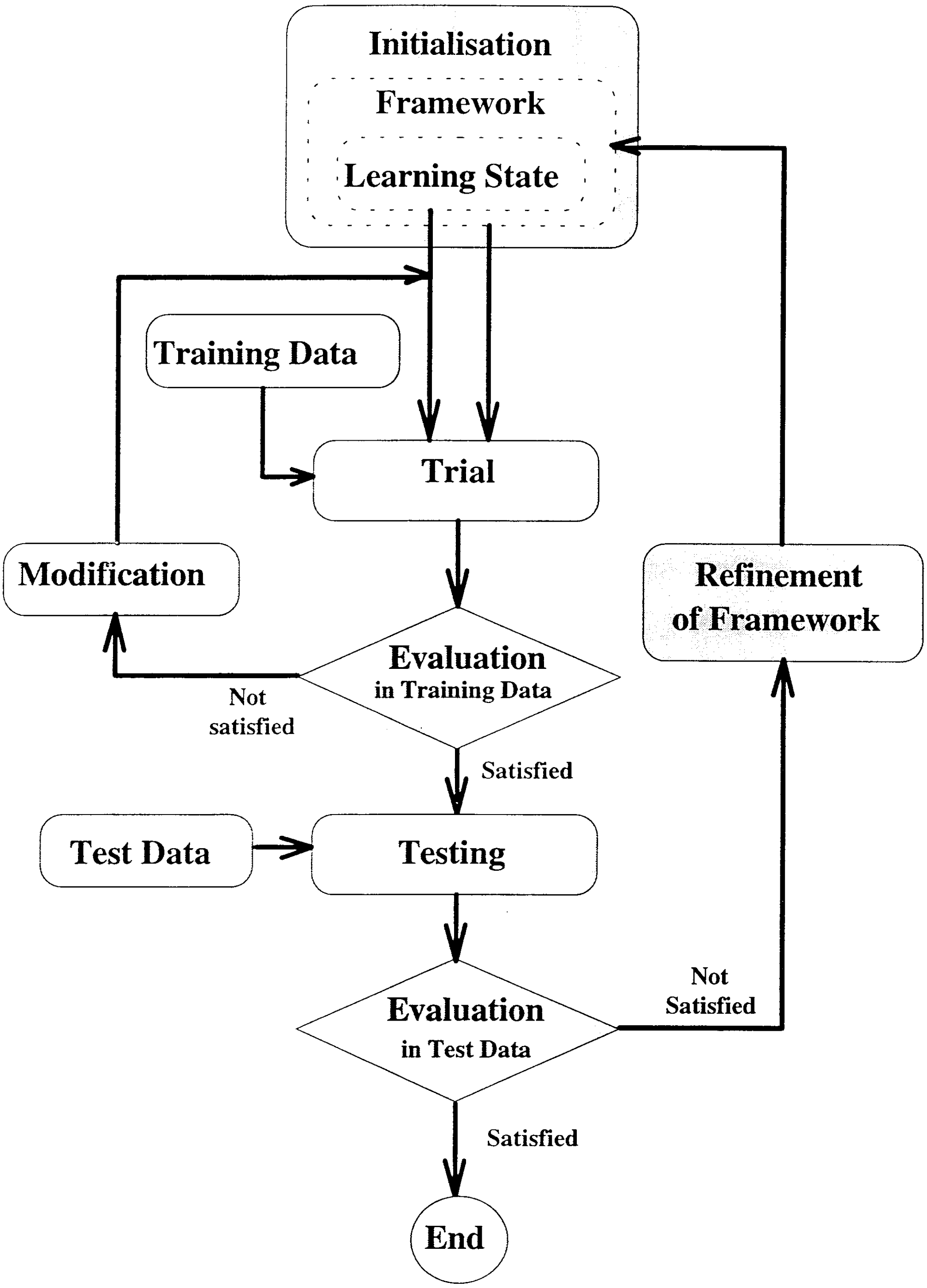


Fig. 3.1 Flow diagram of a common learning process

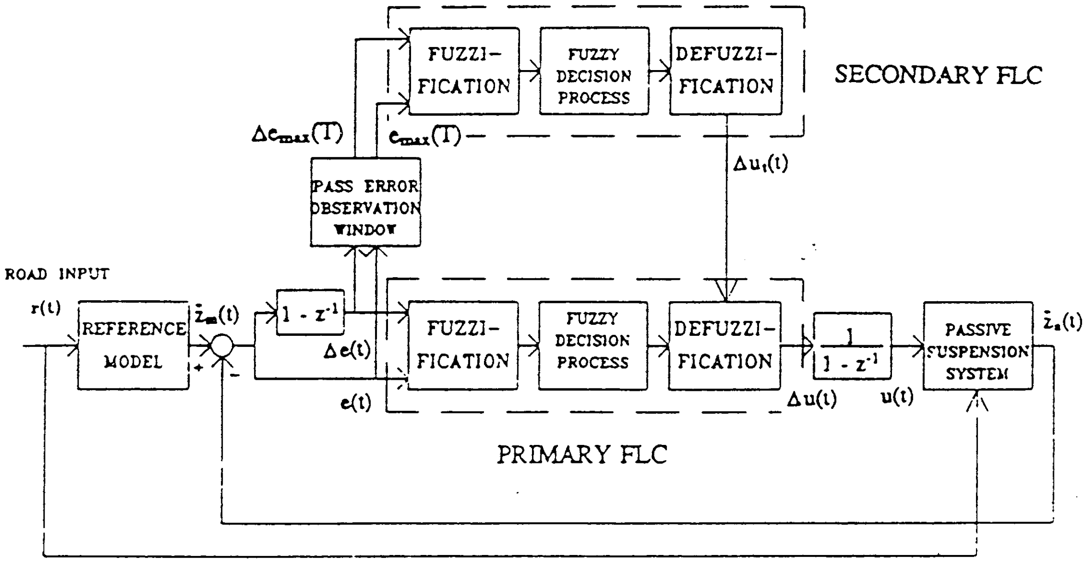


Fig. 3.2 The model reference self-tuning fuzzy logic controller (Lin *et al.,* 1993)

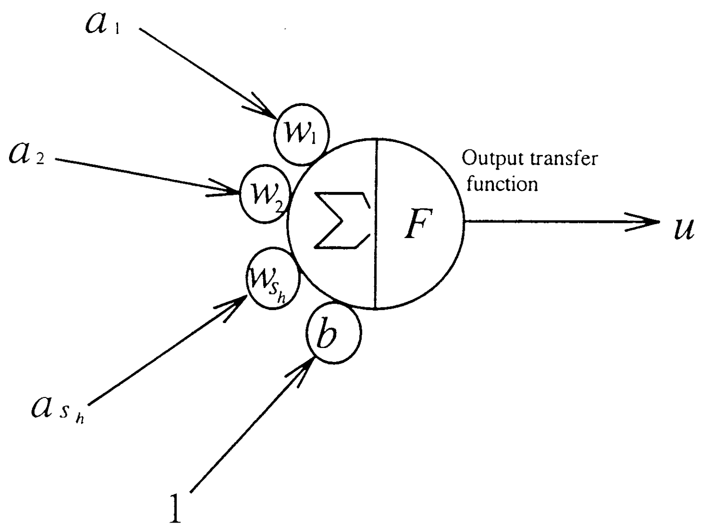
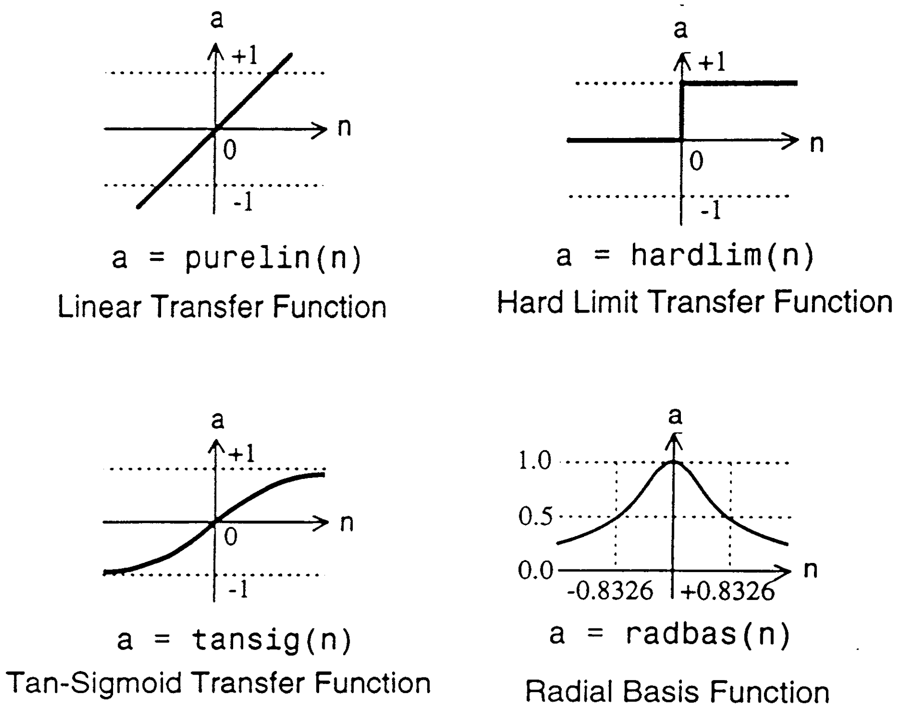


Fig. 3.3 Basic neural network processing element involving a linear combiner and an output transfer function



a = purelin(n)
Linear Transfer Function

a = hardlim(n)
Hard Limit Transfer Function

a = tansig(n)
Tan-Sigmoid Transfer Function

a = radbas(n)
Radial Basis Function

Fig. 3.4 Four basic output transfer functions of neural network processing elements (Demuth and Beale, 1994)

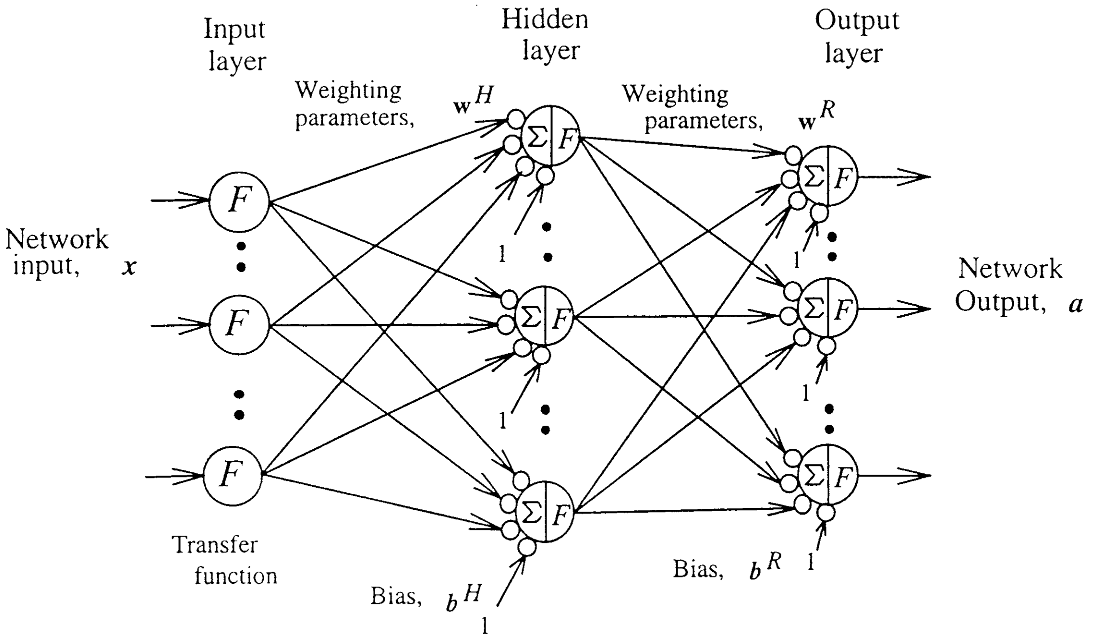


Fig. 3.5 Basic multi-layer neural network

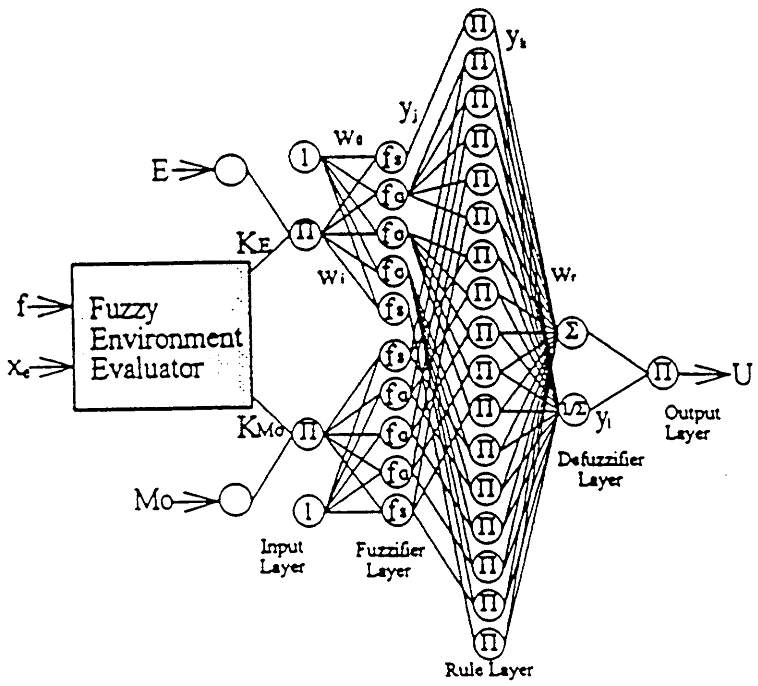


Fig. 3.6 Architecture of the fuzzy neural force controller (Kiguchi and Fukuda, 1996)

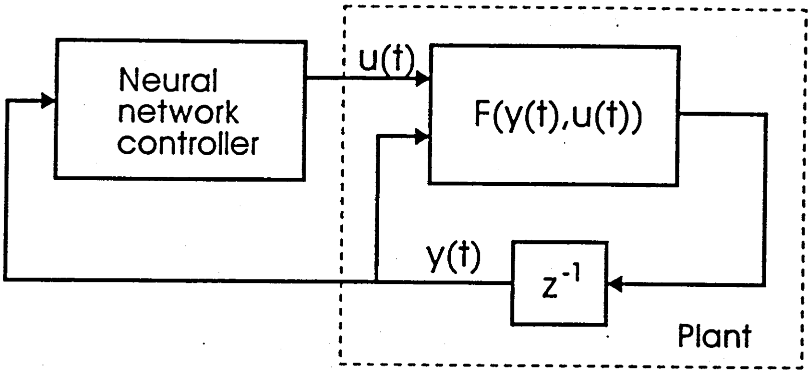


Fig. 3.7 Control structure for the self-learning controller (Nguyen and Widrow, 1991)

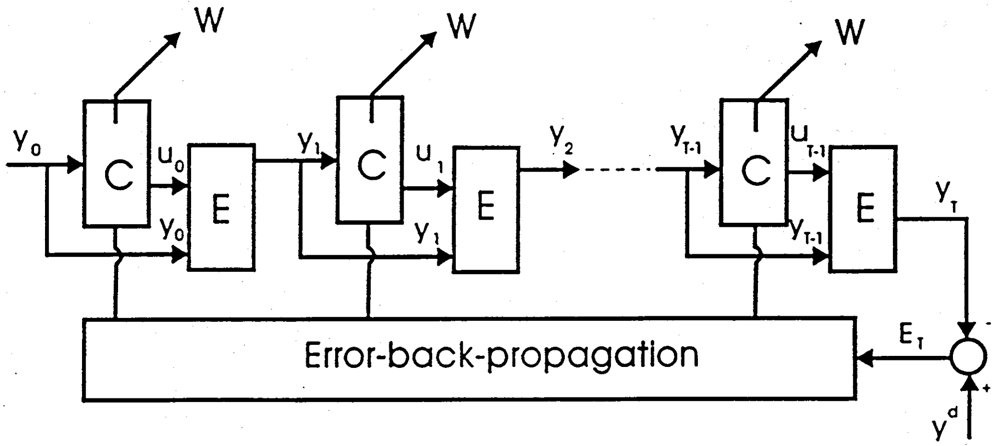


Fig. 3.8 Training with back-propagation. C - controller, E - plant emulator (Nguyen and Widrow, 1991)

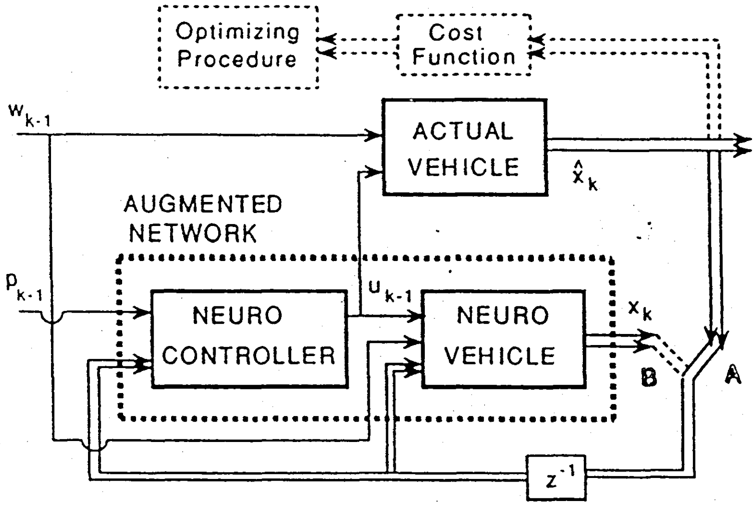


Fig. 3.9 The training process for cost function minimisation (Moran and Nagai, 1992)

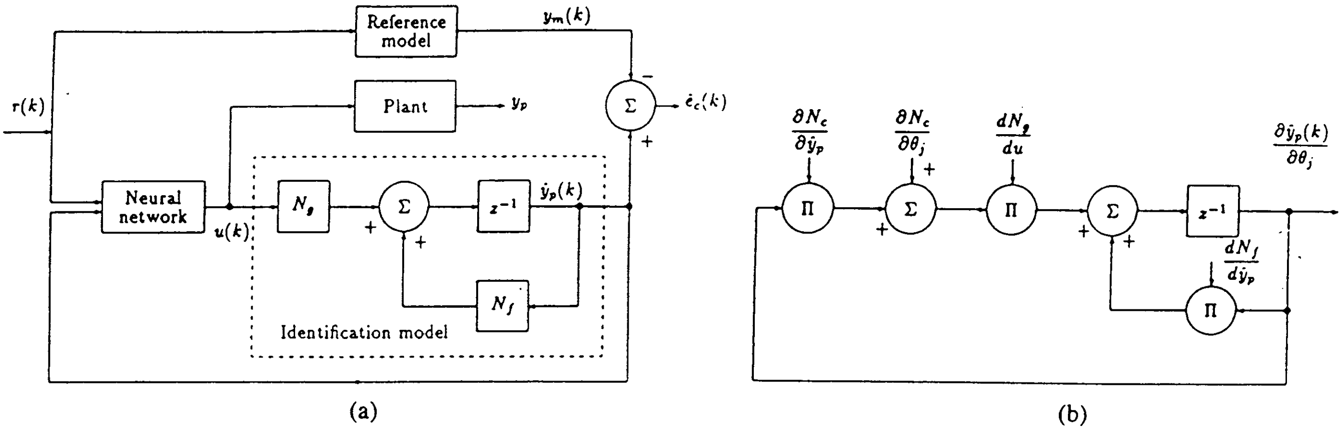


Fig. 3.10 (a) Structure of the controller; (b) The gradient generation circuit (Narendra and Parthasarathy, 1991)

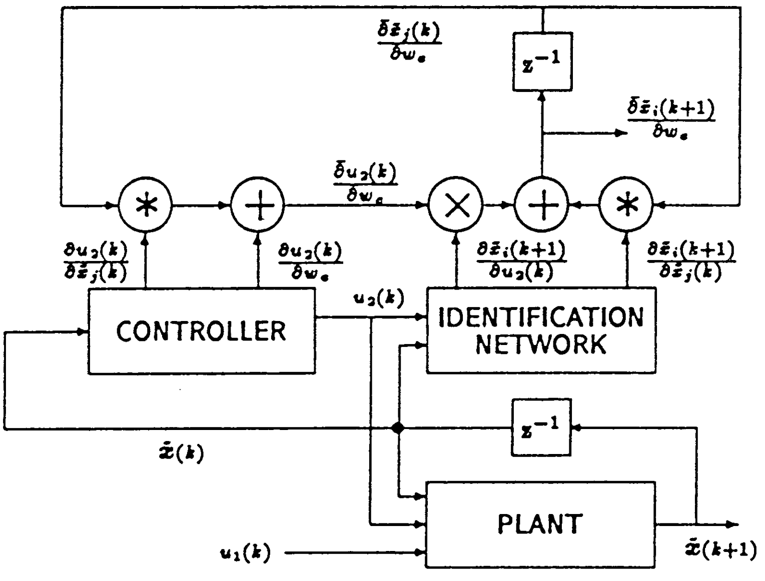

$\frac{\delta \bar{x}_j(k)}{\delta w_e}$
z^{-1}
$\frac{\delta \bar{x}_i(k+1)}{\delta w_e}$
$\frac{\delta u_2(k)}{\delta w_e}$
$\frac{\delta u_2(k)}{\delta \bar{x}_j(k)}$
$\frac{\delta u_2(k)}{\delta w_e}$
$\frac{\delta \bar{x}_i(k+1)}{\delta u_2(k)}$
$\frac{\delta \bar{x}_i(k+1)}{\delta \bar{x}_j(k)}$
CONTROLLER
$u_2(k)$
IDENTIFICATION NETWORK
z^{-1}
$\bar{x}(k)$
PLANT
$u_1(k)$
$\bar{x}(k+1)$

Fig. 3.11 Recurrent structure for controller training involving the derivative computations (Feldkamp *et al.*, 1992)

Chapter 4
Learning Control of Quarter-Car Suspension System

This chapter considers the application of a learning method to the control of a linear quarter-car suspension system. The controller consists of a linear processing element described in subsection 3.2.3 in the previous chapter. System outputs are fed back through the elements which learn, on-line, to reduce a conventional quadratic cost function to its minimum value, for a number of specified road disturbance input functions. The first section of the chapter gives the model description including system equations described in discrete-time, the cost function and the road disturbance function as a single frequency cosine wave. In the second section, the learning process is described. In the third section, the learning process is tracked and the results of the learning are compared with those coming from a more conventional off-line optimisation. The accuracy of the training is demonstrated by comparison of results with those from a standard optimisation procedure, the Linear Quadratic Gaussian (LQG) optimisation, involving a white noise disturbance input and full state feedback.

4.1 Quarter-Car Model with Sinusoidal Road Input

A standard two-degree of freedom quarter-car model is shown in Fig. 4.1. The model consists of body and wheel masses, a force generator, a tyre spring and a tyre damper. Suspension force is applied between the body and wheel masses and the force, u, is controlled by a feedback controller. Road input is applied at a single tyre contact point by a vertical velocity, v. The following representative model parameters have been used:

body mass :	$M_b =$ 250 kg;
wheel mass :	$M_w =$ 30 kg;
tyre stiffness :	$k_t =$ 200,000 N/m;
tyre damping rate :	$c_t =$ 100 N/(m/s).

The basic mechanical equations are:

$$
\begin{aligned}
M_w\,\ddot{z}_w &= k_t\,(z_r - z_w) + c_t\,(\dot{z}_r - \dot{z}_w) - u,\\
M_b\,\ddot{z}_b &= u,
\end{aligned}
\tag{4.1}
$$

where z_w, z_b and z_r are displacements of the wheel, body and road, respectively, measured from static equilibrium. The active suspension model can be converted to standard state-space form (Newland, 1989), given by:

$$
\begin{aligned}
\dot{\boldsymbol{x}}(t) &= \mathbf{A}\,\boldsymbol{x}(t) + \boldsymbol{B}\,u(t) + \boldsymbol{G}\,v\,(t),\\
y(t) &= \mathbf{C}\,\boldsymbol{x}(t),
\end{aligned}
\tag{4.2}
$$

where the four states, $\boldsymbol{x}$, the two outputs, $\boldsymbol{y}$, and the road disturbance input, v, are:

state vector, $\boldsymbol{x} = [x_1\ x_2\ x_3\ x_4]^{\mathrm{T}}$: x_1: tyre deformation ($z_r - z_w$);

x_2: suspension deflection ($z_w - z_b$);

x_3: wheel vertical velocity ($\dot{z}_w$);

x_4: body vertical velocity ($\dot{z}_b$);

output vector, $\boldsymbol{y} = [y_1\ y_2]^{\mathrm{T}}$: y_1: suspension deflection ($z_w - z_b$);

y_2: suspension velocity ($\dot{z}_w - \dot{z}_b$);

force input: u: actuator force between the body and wheel (u);

road disturbance input: v: vertical velocity at the tyre contact point ($\dot{z}_r$).

Matrix **A** is the system-matrix, vector ***B*** is the distribution vector for the actuator force, u, vector ***G*** is the input distribution vector and matrix **C** is the output matrix. They are given by:

$$
\mathbf{A} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ \dfrac{k_t}{M_w} & 0 & \dfrac{-c_t}{M_w} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\quad
\boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ \dfrac{-1}{M_w} \\ \dfrac{1}{M_b} \end{bmatrix},\quad
\boldsymbol{G} = \begin{bmatrix} 1 \\ 0 \\ \dfrac{c_t}{M_w} \\ 0 \end{bmatrix},\quad
\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}.
$$

In the next section of the chapter, the learning process will involve discrete-time operation. Then, the differential and output equations, (4.2), can be represented by the following standard discrete-time state-space form:

$$\begin{aligned} \boldsymbol{x}((n+1)T) &= \mathbf{A}_{\mathrm{d}}\, \boldsymbol{x}(nT) + \boldsymbol{B}_d\, u(nT) + \boldsymbol{G}_d\, v(nT), \\ \boldsymbol{y}(nT) &= \mathbf{C}_{\mathrm{d}}\, \boldsymbol{x}(nT), \end{aligned} \tag{4.3}$$

where n and T denote discrete step number and discrete sampling time, respectively. The discrete state matrices and vectors, $\mathbf{A}_{\mathrm{d}}$, $\boldsymbol{B}_d$, $\mathbf{C}_{\mathrm{d}}$ and $\boldsymbol{G}_d$, are obtained from the continuous state matrices and vectors, $\mathbf{A}$, $\boldsymbol{B}$, $\mathbf{C}$ and $\boldsymbol{G}$ as described by Franklin and Powell (1980).

The road velocity input, v, is considered as a single frequency cosine wave:

$$v(nT) = \cos 2\pi f_{rd} nT\,, \tag{4.4}$$

where f_{rd} is the frequency of the road velocity input.

The suspension force, u, is generated by the limited state feedback controller described by:

$$u(nT)=F_{LPE}(\boldsymbol{y}(nT),\boldsymbol{w}(nT)), \tag{4.5}$$

where $\boldsymbol{w}$ denotes a vector of weighting parameters of the controller and $F_{LPE}(\boldsymbol{y},\boldsymbol{w})$ is the function of a linear processing element which will be optimised by a learning method as described in the next section of the chapter. The system performance is assessed by a conventional quadratic cost function of the form:

$$J = \frac{1}{N}\sum_{n=1}^{N}\left(q_1\, x_1^2(nT) + q_2\, x_2^2(nT) + \dot{x}_4^2(nT)\right), \tag{4.6}$$

where $q_1 = 116000$ and $q_2 = 1190$, as used previously by Gordon *et al.* (1994).

4.2 Learning Process with Quadratic Cost Function

A linear processing element is applied to the control of the quarter-car suspension system. The element has two weighting parameters and a linear combiner. There are two inputs and one output from the element as shown in Fig. 4.2. The element output can be described by:

$$\begin{aligned} u(nT) &= F_{LPE}(y(nT), w(nT)) \\ &= w_1(nT)\, y_1(nT) + w_2(nT)\, y_2(nT). \end{aligned} \tag{4.7}$$

The vehicle performance is assessed, on-line, by the quadratic cost as equation (4.6). The cost is specified by the form:

$$J(nT) = \frac{1}{N} \sum_{i=n-N+1}^{n} I_q(x(iT), u(iT)), \tag{4.8}$$

where I_q is a quadratic function, in which the final term of (4.6) can be expressed as $u(nT)/M_b$. The weighting parameters, w, are updated, on-line, by the gradient method in order to reduce the cost. The gradient method for a dynamic system involving neural networks is described by Narendra and Parthasarathy (1991). A block diagram, showing the system structure, is shown in Fig. 4.3 (a). The update rule is:

$$w((n+1)T) = w(nT) + \Delta w(nT), \tag{4.9}$$

where

$$\begin{aligned} \Delta w(nT) &= -\delta \frac{\partial J(nT)}{\partial w(nT)} \\ &= -\frac{\delta}{N} \sum_{i=n-N+1}^{n} \left(\frac{\partial I_q((x(iT), u(iT))}{\partial x(iT)} \cdot \frac{\partial x(iT)}{\partial w(iT)} + \frac{\partial I_q((x(iT), u(iT))}{\partial u(iT)} \cdot \frac{\partial u(iT)}{\partial w(iT)} \right), \end{aligned}$$

δ is the learning rate and Δw is the weighting vector update. The gradient, $\partial J(nT)/\partial w(nT)$, is evaluated over the interval, $[n - N + 1,\ n]$: Then the weighting parameters are updated at the step, $n + 1$. The sequence of the weighting vector update in discrete-time is shown in Fig. 4.4. Strictly speaking, w is no longer a constant vector element because the cost function calculation is not re-executed in the on-line learning process. The learning rate can be chosen to give a suitable compromise between speed of learning and stability or it can be adaptive.

The generation of the sensitivity matrices and vectors is shown in Fig. 4.3 (b). The Jacobians, $\partial\, I_q(nT)/\partial\, x(nT)$ and $\partial\, I_q(nT)/\partial\, u(nT)$ are partial derivatives of the cost with respect to the state vector elements and control force at step, n, respectively. These are obtained by simple differentiation. The Jacobians, $\partial x(nT)/\partial w(nT)$ and $\partial u(nT)/\partial w(nT)$ are partial derivatives of the state vector elements and control force with respect to the weighting parameters, respectively. These are obtained by partial differentiation of equations (4.3) and (4.7) with respect to $\boldsymbol{w}$ as:

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{w}}((n+1)T) = \mathbf{A}_{\mathrm{d}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{w}}(nT) + \boldsymbol{B}_d\frac{\partial u}{\partial \boldsymbol{w}}(nT), \quad \text{since}\ \frac{\partial v}{\partial \boldsymbol{w}} = 0, \tag{4.10}$$

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{w}}(nT) = \mathbf{C}_{\mathrm{d}}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{w}}(nT),$$

$$\frac{\partial u}{\partial \boldsymbol{w}}(nT) = \frac{\partial F_{LPE}(\boldsymbol{y}(nT),\boldsymbol{w}(nT))}{\partial \boldsymbol{y}(nT)}\cdot\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{w}}(nT) + \frac{\partial F_{LPE}(\boldsymbol{y}(nT),\boldsymbol{w}(nT))}{\partial \boldsymbol{w}(nT)},$$

where $\partial F_{LPE}(\boldsymbol{y},\boldsymbol{w})/\partial\ \boldsymbol{y}$ and $\partial F_{LPE}(\boldsymbol{y},\boldsymbol{w})/\partial\ \boldsymbol{w}$ are row vectors, $[w_1\ w_2]$ and $[y_1\ y_2]$, respectively.

4.3 Simulation Results

Two outputs from the model, suspension deflection and suspension velocity, are fed back through the linear processing element and its output determines the actuator force. Road velocity inputs (cosine waves, 0.5, 1, 2, 4, 8, 12 and 16 Hz, amplitude, 1 m/s) were applied to the tyre contact point and the linear processing elements were optimised with each road velocity in order to reduce the cost function, (4.6). The update rule of the weighting parameters is given by (4.9), where the sampling time, T, is 0.005 second and the sampling size of the cost, N, is 50. The reduction of the cost during training is shown in Fig. 4.5, demonstrating how the performance of the suspension was improved by training for the 8 Hz road input.

Each training involved an adaptive learning rate (Demuth and Beale, 1994), in order to improve efficiency. The adaptive learning rate increases the learning rate, δ, if the cost ratio (present cost/previous cost) is less than 1 and decreases it if the cost ratio is more than 1.005. The adaptation parameters are selected by the analyst according to the case. In the training for the 8 Hz road input, the learning rate is increased by multiplying by 1.01 and decreased by multiplying by 0.7.

The system performances of the learning-controlled systems for road disturbance inputs, 0.5 Hz and 8 Hz, are compared with the Linear Quadratic Gaussian (LQG) controlled system (Gordon *et al*, 1994) in Fig. 4.6 and Fig. 4.7, respectively. In Fig. 4.6, although there are no differences between the learning and LQG controlled systems in amplitudes of body vertical acceleration and tyre deformation, the learning controlled system dramatically reduces suspension deflection for the 0.5 Hz road input. On the other hand, in Fig. 4.7, the learning-controlled system is advantageous in reducing body vertical acceleration and tyre deformation for the 8 Hz road input.

Fig. 4.8 shows an overall comparison between the learning control, the LQG and conventional off-line optimisation via the Nelder-Mead Simplex method (Nelder and Mead, 1964) for the road velocity inputs, 0.5, 1, 2, 4, 8, 12 and 16 Hz. (Detailed data of the results are shown in Table 4.1.) Fig. 4.8 shows that the learning control can improve on the LQG control for each road input. Of course, the LQG control involves compromise over the frequency components in the excitation, while the learning and simplex controllers deal specifically with one frequency at a time. Furthermore, the on-line learning control achieves as small a cost as the conventional off-line optimisation method.

4.4 Conclusions

It was shown how a single neural network element can be trained on-line to optimise the performance of a quarter-car suspension system. The learning technique arrived at the same results as the off-line optimisation device, the Nelder-Mead Simplex controller, for each specified road disturbance input function.

It was shown that even if a controller has a linear processing element, it can improve upon an LQG controller for a linear system if the disturbance input has only a known single frequency. This has potential application to an adaptive controller, which responds by adaptation to particularly strong frequency components in the road disturbance input to the vehicle.

The results obtained suggest that good performance will be obtainable from a variable-geometry active-suspension system when the road disturbance is sinusoidal. Continuation of the work in this direction requires consideration of cost functions of general form, randomly-profiled road disturbance inputs and a non-linear system. Probably a more elaborate neural network will be required. These are considered in chapter 5 and chapter 6.

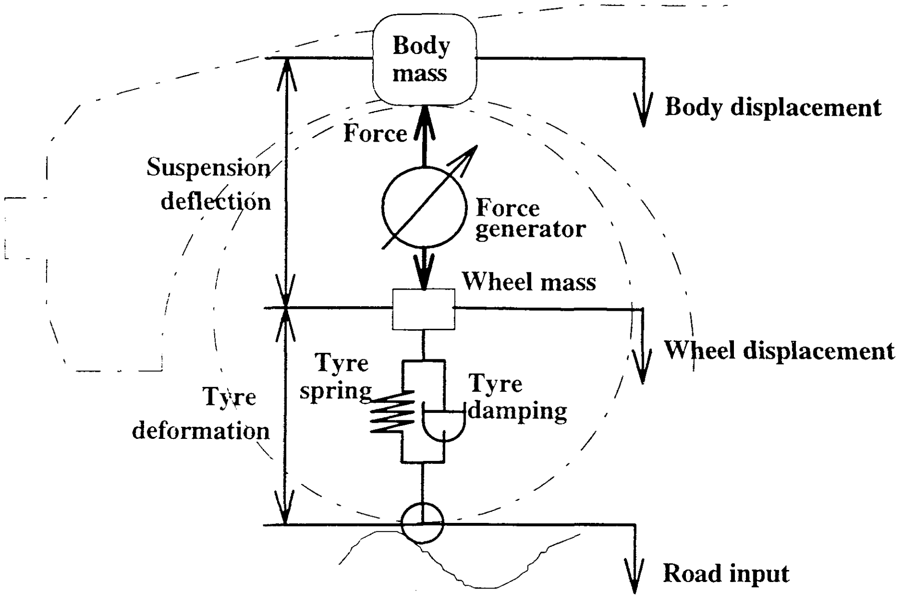


Fig. 4.1 Quarter-car active suspension model

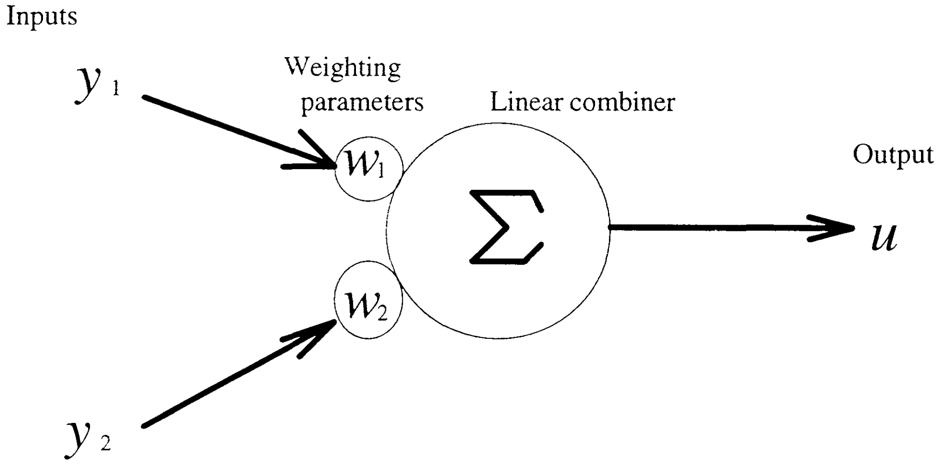
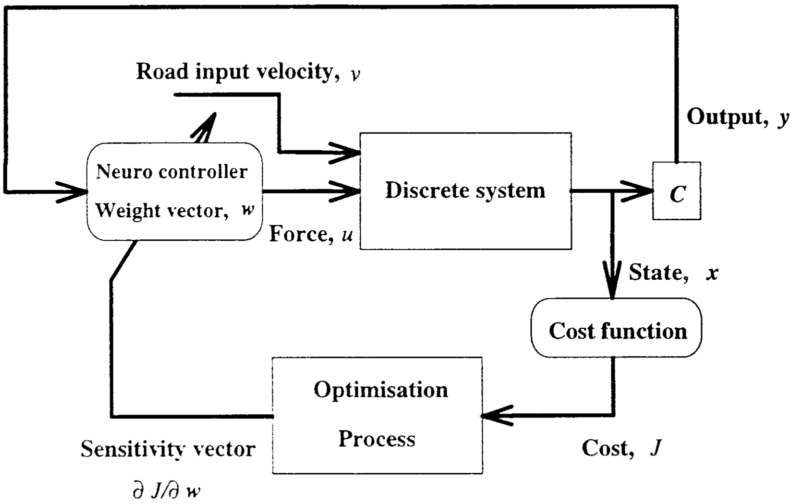
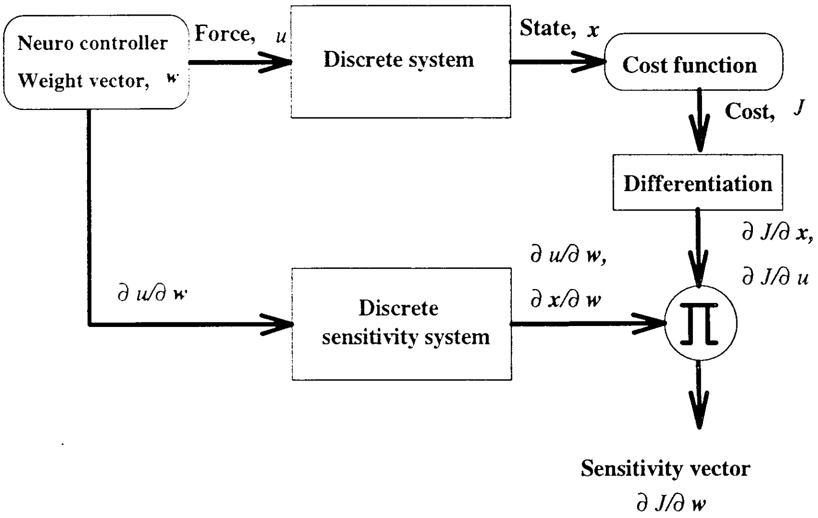


Fig. 4.2 Linear processing element



(a) System structure



(b) Optimisation process

Fig. 4.3 Diagrammatic representations of (a) system structure and (b) optimisation process

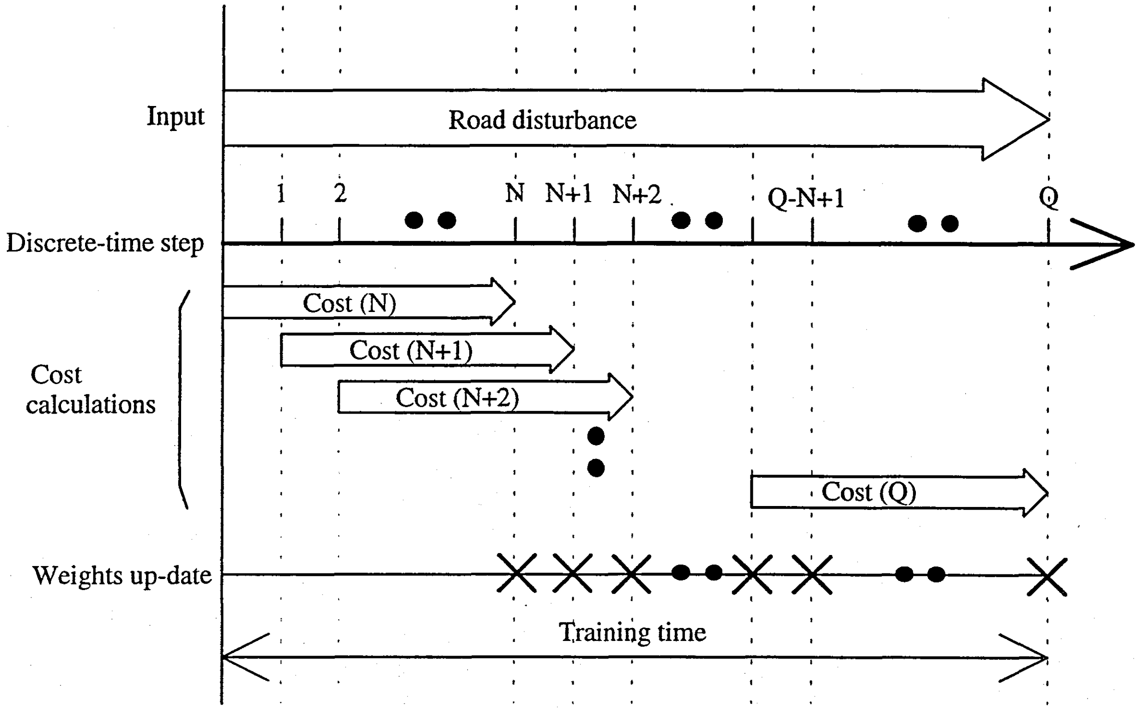


Fig. 4.4 Sequential representation of on-line learning system in discrete-time

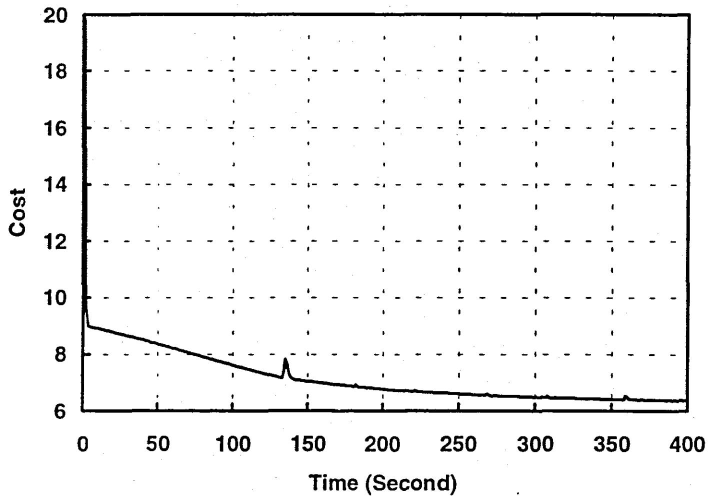
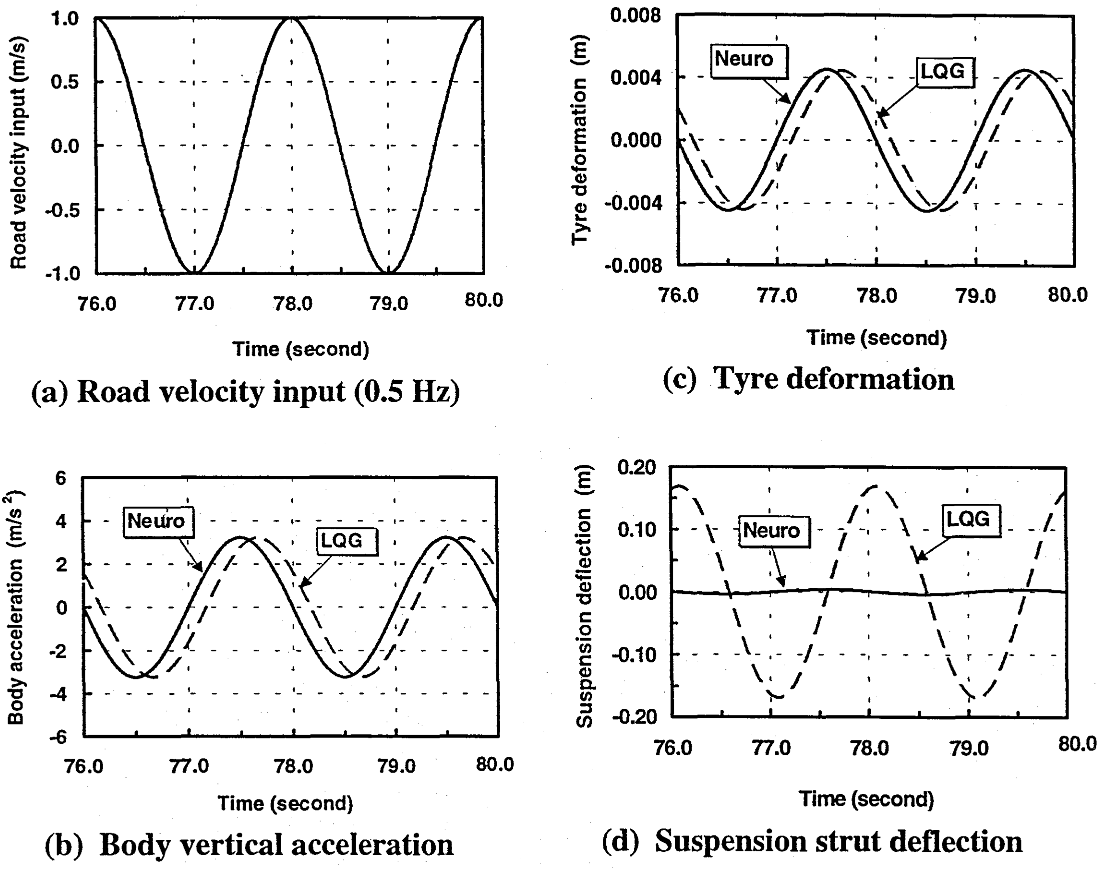


Fig. 4.5 The descending cost function, as learning proceeds; 8 Hz input frequency



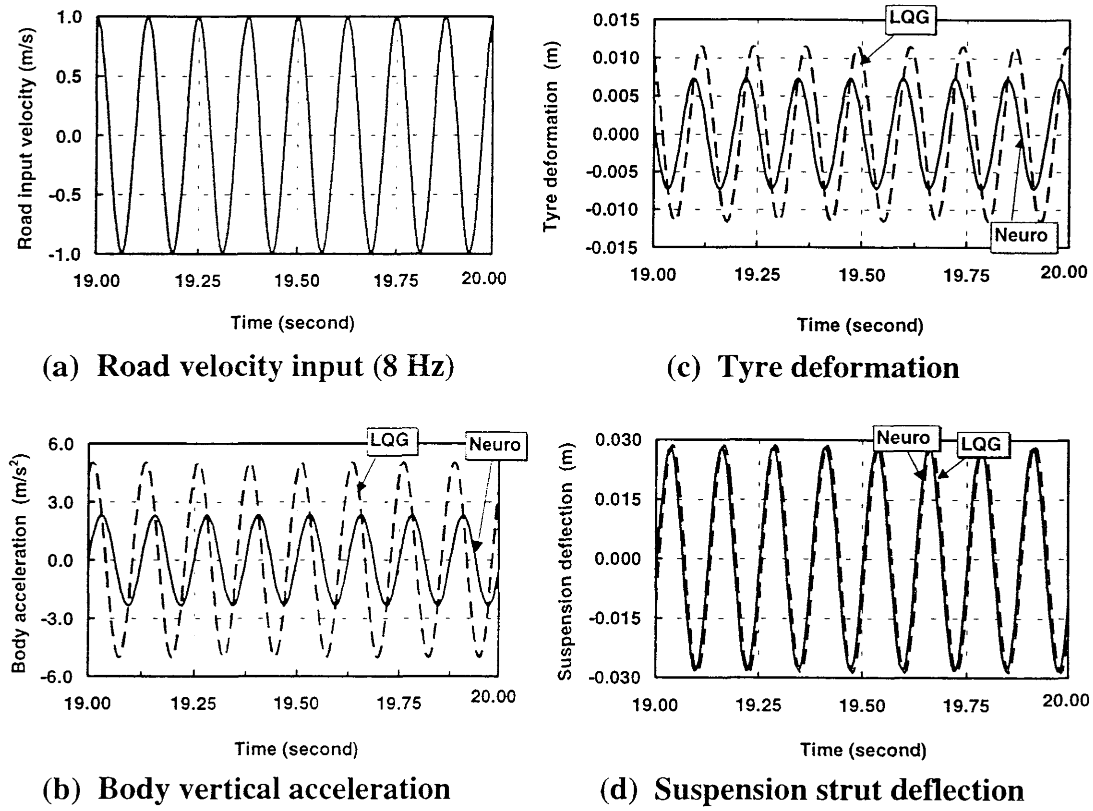
(a) Road velocity input (0.5 Hz)

(c) Tyre deformation

(b) Body vertical acceleration

(d) Suspension strut deflection

Fig. 4.6 Vehicle responses for 0.5 Hz road input



(a) Road velocity input (8 Hz)

(c) Tyre deformation

(b) Body vertical acceleration

(d) Suspension strut deflection

Fig. 4.7 Vehicle responses for 8 Hz road input

Table 4.1 Overall comparison of LQG, Nelder-Mead and neuro-controller results

Road input frequency		Gain [N/(m/s)] (force/input)	Phase lag[rad] (force/input)	cost
0.5 Hz	LQG	803	-1.02	23
	Nelder	793	-1.48	6.5
	neuro	806	-1.56	6.4
1 Hz	LQG	1223	- 0.38	29
	Nelder	591	- 1.51	10
	neuro	589	- 1.48	10
2 Hz	LQG	1079	0.06	16
	Nelder	22	0.06	4.2
	neuro	49	0.06	4.2
4 Hz	LQG	1000	0.25	11
	Nelder	9.0	0.13	2.1
	neuro	62	0.13	2.1
8 Hz	LQG	1263	0.50	21
	Nelder	536	1.63	6.3
	neuro	583	1.76	6.3
12 Hz	LQG	1999	1.13	77
	Nelder	2009	1.51	36
	neuro	2005	1.51	36
16 Hz	LQG	1376	2.26	48
	Nelder	1884	1.58	36
	neuro	1890	1.58	36

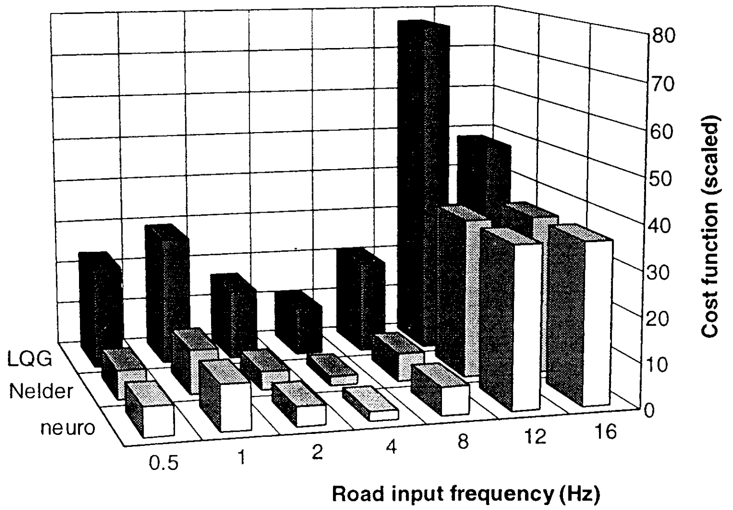


Fig. 4.8 Overall comparison of LQG, Nelder-Mead and neuro-controller results

Chapter 5
Neural Network Control of Quarter-Car Suspension System

In chapter 4, a single neural network element was applied to the control of the quarter-car suspension system. The element was optimised with a road disturbance input, which was specified as a single frequency cosine wave, in order to reduce a conventional quadratic cost function.

This chapter deals with a more realistic situation involving a randomly profiled road disturbance input and a non-quadratic cost function. A multi-layer neural network is applied to the control of the quarter-car suspension system. System states are fed back through the neural network which learns to reduce the non-quadratic cost function to its minimum value. The first section of the chapter gives the non-quadratic cost function and road disturbance function as a random noise. The second section describes the multi-layer neural network and its learning process. In the third section, the learning process is tracked. The performance of the neural network controller is compared with that of the non-linear optimal controller which is designed via the Pontryagin Maximum Principle (MP). Both the neuro and MP controllers are optimised using the same cost function and are tested for various levels of randomly profiled roads.

5.1 Quarter-Car Model with Random Noise Input

The quarter-car model, described in section 4.1 in the previous chapter, was represented by the discrete-time state-space form:

$$\boldsymbol{x}((n+1)T) = \mathbf{A}_{\mathrm{d}}\ \boldsymbol{x}(nT) + \boldsymbol{B}_d\ u(nT) + \boldsymbol{G}_d\ v(nT), \qquad (5.1)$$

where the four states of $\boldsymbol{x}$, control input, u, and road disturbance input, v, were described in the previous chapter.

In this chapter, the road disturbance input is considered as a random noise given by:

$$v(iT) = f_{res} \sum_{m=1}^{N_{rd}} V_m \cos\left(2\pi \frac{mi}{N_{rd}} + \theta_m \right), \qquad (5.2)$$

$$(i = 1 \cdots N_{rd})$$

where m denotes the discrete frequency step number, f_{res} denotes the frequency resolution and θ_m denotes the phase lag. The discrete Fourier spectrum, V_m, depends on the level of roughness of the random road represented. One batch of the random noise input has N_{rd} discrete data points.

The actuator force, u, is generated by a full-state feedback controller as follows:

$$u(nT)=F_{NN}(\boldsymbol{x}(nT),\boldsymbol{w}(nT)), \qquad (5.3)$$

where $\boldsymbol{w}$ denotes a vector of weighting parameters and $F_{NN}(\boldsymbol{x},\boldsymbol{w})$ is the function of a multi-layer neural network. The weighting vector, $\boldsymbol{w}$, will be optimised in order to reduce the non-quadratic cost function given by:

$$J = \frac{1}{N} \sum_{n=1}^{N} \left(q_1\, x_1^2(nT) + q_2\, x_1^6(nT) + q_3\, x_2^2(nT) + q_4\, x_2^{10}(nT) + \dot{x}_4^2(nT) \right), \qquad (5.4)$$

where q_1= 1600; q_2=2×10^{12} ; q_3=500; q_4=5×10^{11} as used previously by Gordon *et al.* (1994).

The contribution to the cost, (5.4), from the tyre deformation steeply increases when the tyre deformation exceeds the absolute value, 0.005 m, as shown in Fig. 5.1. The optimal controller design techniques, which employed the non-quadratic cost functions, were previously studied by Gordon, Marsh and Milsted (1990 and 1991) and Gordon *et al.* (1994). The following sections in this chapter will describe how the neural network learns to minimise the non-quadratic cost function and how the neural network performs.

5.2 Learning Process with Non-Quadratic Cost Function

Multi-layer neural network

The architecture of the multi-layer neural network involves three layers, input, hidden and output, and each layer includes a number of elements as shown in Fig. 5.2. Each output of an element is connected to the elements in the next layer through the weighting parameters. The input-output relationship is:

$$u = L(w^R S(\mathrm{w}^{\mathrm{H}} L(x) + b^H) + b^R), \tag{5.5}$$

where w^{H} and b^H are the weight matrix and bias vector for the hidden layer, respectively, and w^R and b^R are the weight vector and bias for the output layer, respectively. L denotes the linear output function of an element and S denotes the tangent-sigmoid transfer function of an element. In this case, there are 4 inputs, 6 hidden elements and 1 output. The network is included in the *Neural Network Toolbox for* MATLAB and more detailed equations are given in the users' guide (Demuth and Beale, 1994).

Learning process

The neural network controller output is described by the standard discrete-time form:

$$u(nT)=F_{NN}(x(nT),w(nT)) , \tag{5.6}$$

where the weighting vector, w, involves all components of weight and bias in the network. During trials, vehicle performance is assessed by the non-quadratic cost, (5.4), which is given by the form:

$$J(nT) = \frac{1}{N} \sum_{i=n-N+1}^{n} I_{nq}(x(iT), u(iT)) , \tag{5.7}$$

where I_{nq} is a non-quadratic function, in which the final term of (5.4) can be expressed as $u(nT)/M_b$. The weighting vector, w, is updated by:

$$w((n+1)T) = w((n-N+1)T) + \Delta w(nT), \tag{5.8}$$

where

$$\Delta w(nT) = -\delta \frac{\partial\ J(nT)}{\partial\ w(nT)}$$

$$= -\frac{\delta}{N} \sum_{i=n-N+1}^{n} \left(\frac{\partial\ I_{nq}(x(iT),u(iT))}{\partial\ x(iT)} \cdot \frac{\partial\ x(iT)}{\partial\ w(iT)} + \frac{\partial\ I_{nq}(x(iT),u(iT))}{\partial\ u(iT)} \cdot \frac{\partial\ u(iT)}{\partial\ w(iT)} \right).$$

The gradient, $\partial\ J(nT)\ /\partial\ w(nT)$, is evaluated over the interval, $[n - N + 1,\ n]$, for one batch of the random noise road input. The weighting vector is updated at the step, n +

1. The batch is repeated until the cost is minimised. The sequence of the learning process is shown in Fig. 5.3.

In (5.8), the Jacobians, $\partial\, I_{nq}(nT)/\partial\, x(nT)$ and $\partial\, I_{nq}(nT)/\partial\, u(nT)$ are obtained by simple differentiation. The Jacobians, $\partial\, x(nT)\, /\partial\, w(nT)$ and $\partial\, u(nT)\, /\partial\, w(nT)$, are obtained by partial differentiation of equations (5.1) and (5.6) with respect to w as:

$$\frac{\partial\, x}{\partial\, w}((n+1)T) = \mathbf{A}_{\mathrm{d}}\frac{\partial\, x}{\partial\, w}(nT) + \boldsymbol{B}_d\frac{\partial\, u}{\partial\, w}(nT), \quad \text{since}\,\frac{\partial\, v}{\partial\, w} = 0, \qquad (5.9)$$

$$\frac{\partial\, u}{\partial\, w}(nT) = \frac{\partial\, F_{NN}(x(nT),w(nT))}{\partial\, x(nT)}\cdot\frac{\partial\, x}{\partial\, w}(nT) + \frac{\partial\, F_{NN}(x(nT),w(nT))}{\partial\, w(nT)},$$

where $\partial F_{NN}(x(nT),w(nT))/\partial x(nT)$ and $\partial F_{NN}(x(nT),w(nT))/\partial\; w(nT)$ are specified in Appendix A and are calculated by the standard back-propagation algorithm taken from the literature (Hagan, Demuth and Beale, 1996).

5.3 Simulation Results

Firstly, a random noise velocity input, (5.2), was repeatedly applied to the tyre contact point until the neural network controller minimised the non-quadratic cost, (5.4). Secondly, the neuro controller was tested for various levels of random noise velocity inputs with r.m.s. values: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8 and 1 m/s. In the test, the neuro controller gave a good performance reducing the tyre deformation for the road velocity input with r.m.s. value, 1.0 m/s. However, it gave poor ride comfort through the range. A possible reason was that the high-amplitude motion dominated the cost function, which had contributed to updating the weighting parameters of the neuro controller. In order to avoid the problem, the neural network controller was trained by the following effective technique:

The amplitude, V_m, in the road disturbance, (5.2), was replaced by $V_m(i)$ as:

$$v(iT) = f_{res}\sum_{m=1}^{N_{rd}} V_m(i)\cos\left(2\pi\frac{mi}{N_{rd}} + \theta_m\right), \qquad (5.10)$$

$$(i = 1 \cdots N_{rd})$$

where the amplitude varied in discrete-time. A batch of the road velocity input, created by (5.10), is shown in Fig. 5.4. In the batch, the high and low amplitudes were separated. The cost function, (5.7), was modified as:

$$J(nT) = \frac{1}{N} \sum_{i=n-N+1}^{n} K(i)\, I_{nq}(x(iT), u(iT)), \qquad (5.11)$$

where $K(i)$ were selected to be bigger, the lower the road amplitude. The learning technique presented here would give the neural network more opportunities to learn for the low-amplitude motion. The batch of the road input, having 1600 data points in discrete-time, was repeatedly applied to the system and the cost function (5.11) was successfully reduced from the initial cost, 8.93×10^{26}, to the final cost, 1.42. The reduction of the cost during the training is shown in Fig. 5.5.

Finally, the neural network controller was tested for various levels of random noise velocity inputs with r.m.s. values: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8 and 1 m/s. Each input was independent of the others. The system performance was compared with that of the LQG controller (Gordon *et al.*, 1994) and the non-linear optimal controller designed by the Pontryagin Maximum Principle (MP) (Gordon, Marsh and Milsted, 1990 and 1991; Gordon *et al.*, 1994). The MP controller was optimised using the non-quadratic cost function, (5.4). The LQG involves the quadratic cost function, (4.6), which was given in section 4.1 in the previous chapter.

Fig. 5.6 shows an overall comparison between the neuro, LQG and MP controllers in the r.m.s. value of the tyre deformation through the range of road velocity input: 0.1-1.0 m/s. Fig. 5.7 shows an overall comparison between the neuro, LQG and MP controllers in the peak value of the tyre deformation through the range of road velocity input: 0.1-1.0 m/s. From each figure, in the LQG controlled system, both the r.m.s. and peak values of the tyre deformation increase linearly with respect to the road velocity input. On the other hand, the neuro and MP controllers have non-linear characteristics.

This example demonstrates that the non-linear control reduces the tyre deformation for the road velocity input with r.m.s. value, 1.0 m/s, and this provides an obvious advantage for improving ride comfort for the road velocity input with r.m.s. value, 0.1 m/s. Fig. 5.8 shows autospectra of body accelerations with the LQG and neuro controllers for the road velocity input with r.m.s. value, 0.1 m/s.

5.4 Conclusions

It was shown how the multi-layer neural network can be trained to optimise the performance of the quarter-car suspension system. The neural network successfully reduced a non-quadratic cost function under various levels of randomly profiled roads.

The results show that the learning neural network control produces the same performance as the MP optimal system. The learning neural network system provides an alternative and possibly simpler way of arriving at the optimal system. However, the selection of training data, network architecture and so on are still of interest to improve learning efficiency.

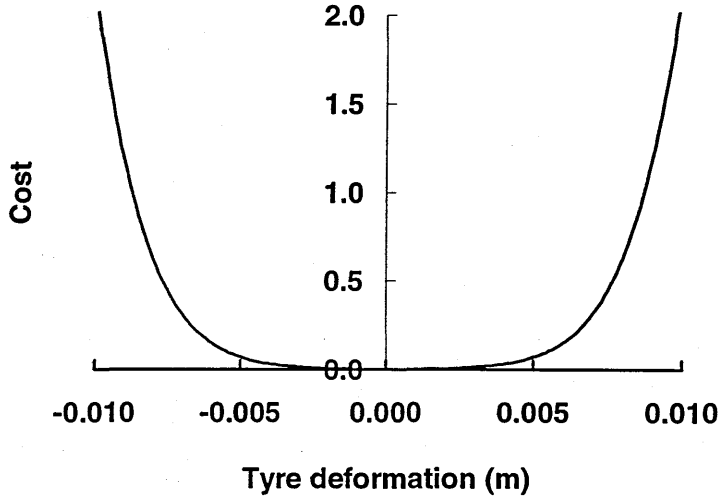


Fig. 5.1 Contribution of tyre deformation to non-quadratic cost

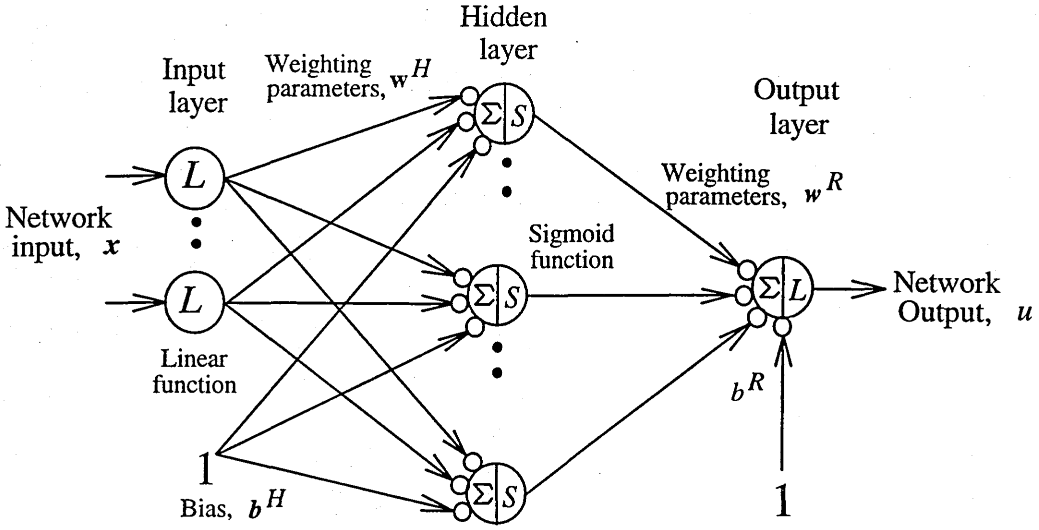


Fig. 5.2 Multi-layer neural network with single output

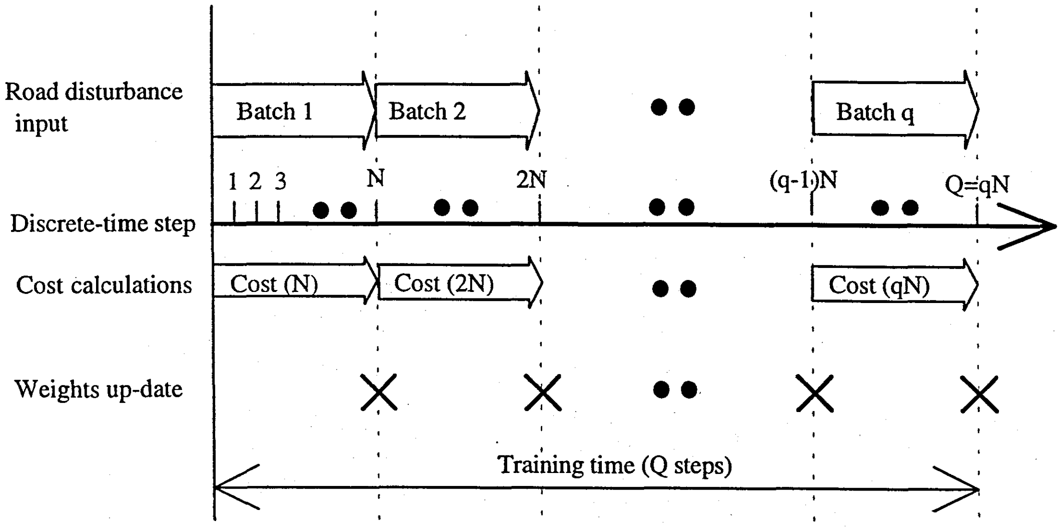


Fig. 5.3 Sequential representation of batch learning system in discrete-time

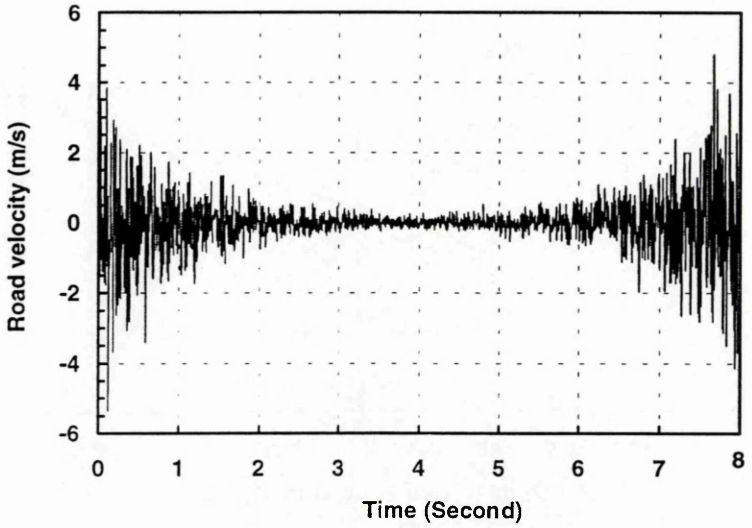


Fig. 5.4 One batch of the road velocity input used for training

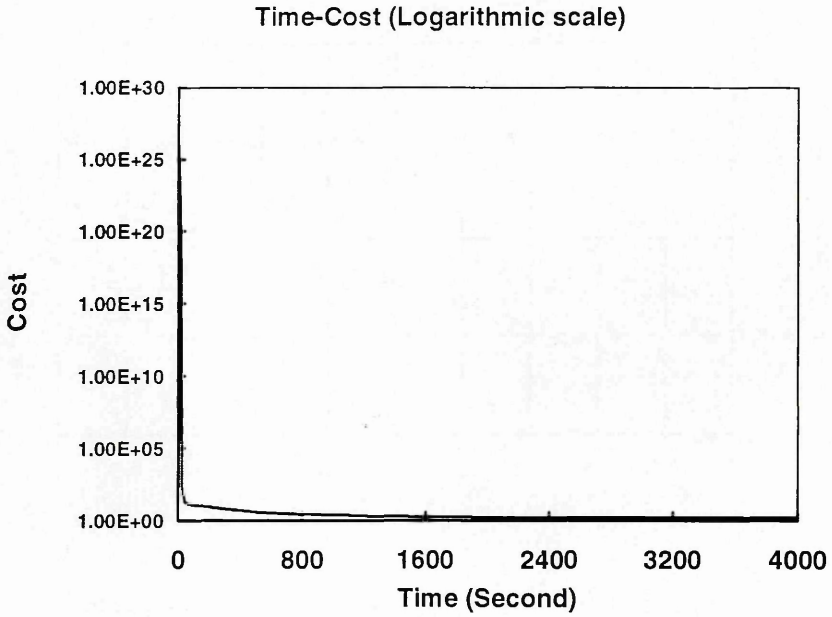


Fig. 5.5 The descending non-quadratic cost function

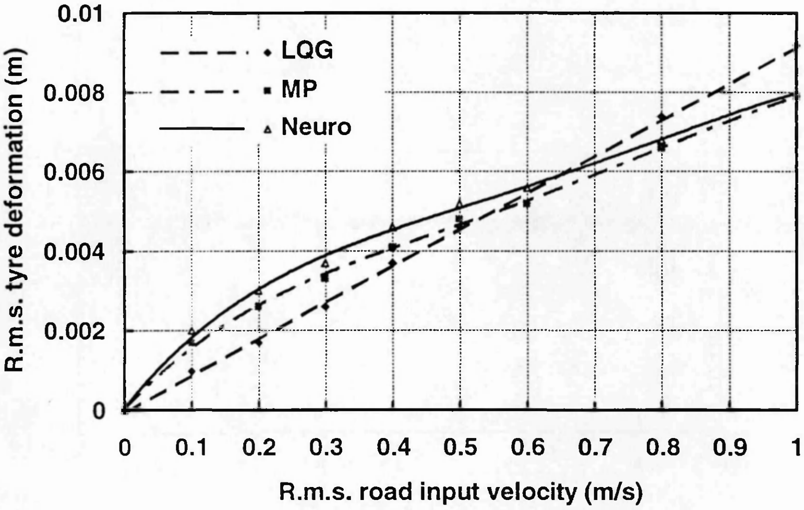


Fig. 5.6 R.m.s. values of tyre deformation as functions of r.m.s. values of random road velocity input: Comparison between LQG, MP and neuro-controllers

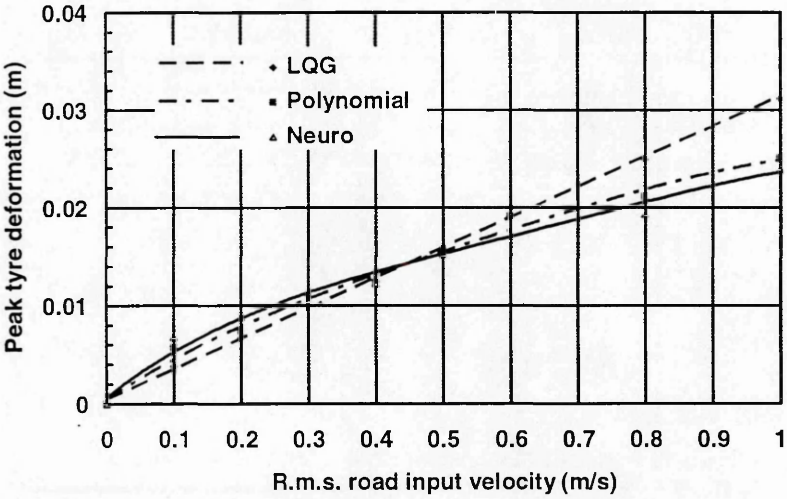
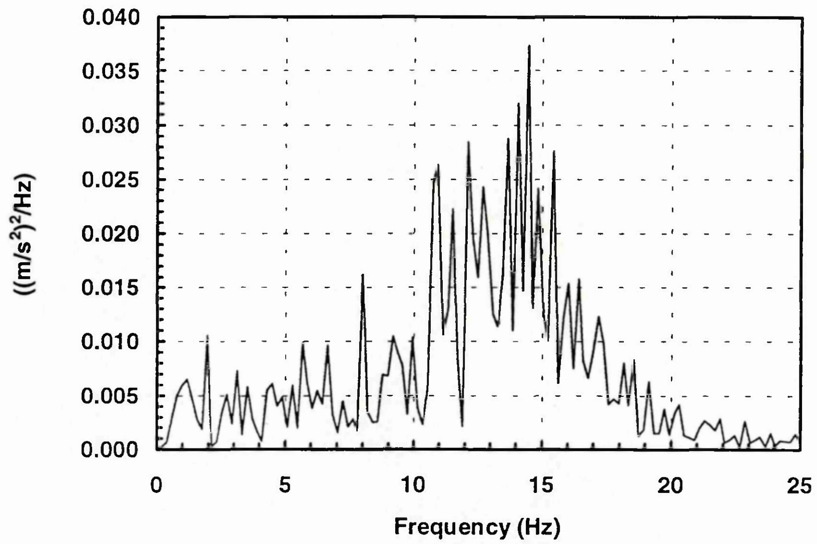
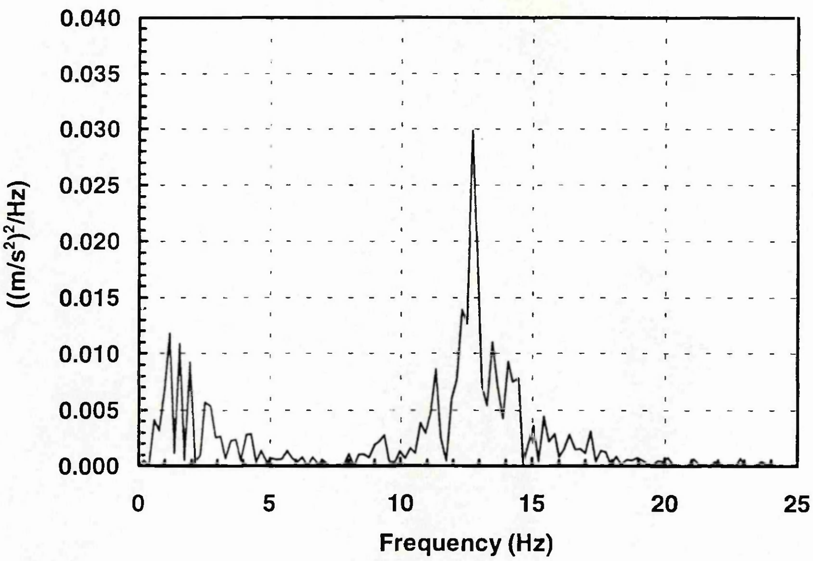


Fig. 5.7 Peak values of tyre deformation as functions of r.m.s. values of random road velocity input: Comparison between LQG, MP and neuro-controllers



(a) LQG controller



(b) Neuro-controller

Fig. 5.8 Autospectra of body accelerations under a random road input (velocity r.m.s. 0.1 m/s) for LQG and neuro-controllers

Chapter 6

Neural Network Control of Non-Linear System

Chapter 4 and chapter 5 showed how neural networks (a single neural network element and a multi-layer neural network, respectively) can be trained to optimise the performance of a linear suspension system. Each previous chapter confirms that neural networks can provide optimal control systems using cost functions of general form. This chapter deals with the unsettled question of how a neural network can be applied to the control of a non-linear system.

In the first section of the chapter, a non-linear single-mass vibration system is developed. A non-quadratic cost function, which evaluates the system performance, is provided. The second section describes the learning process regarding the control of the non-linear system. In the third section, three neural network controllers are trained, but separately, using three different cost functions according to each demand. The issue of how the neural network learns to control the non-linear system correctly, according to the cost function given, is confirmed by a comparison between the performances of these three controllers.

6.1 Non-Linear Single-Mass Vibration System

A one-degree of freedom single-mass vibration system is designed as shown in Fig. 6.1. The model consists of a body mass, a force generator, a non-linear spring and a damper. The force of the non-linear spring is given by:

$$\text{spring force} = -\, k_{coe}\,(\text{deflection})^3, \qquad (6.1)$$

where k_{coe} is the spring coefficient in N/m^3.

One end of the force generator, which is controlled by a feedback controller, is fixed at the sky and its vertical force, u, is applied to the body. An external disturbance input is applied at the lower end of the spring-damper through a vertical velocity, v. The following representative model parameters have been used:

body mass :	$M_b = 250$ kg;
spring coefficient :	$k_{coe} = 2{,}450{,}000$ N/m^3;
damping rate :	$c = 1500$ N/(m/s).

The basic mechanical equation is:

$$M_b\, \ddot{z}_b = -k_{coe}\left(z_b - z_r\right)^3 - c\left(\dot{z}_b - \dot{z}_r\right) + u. \tag{6.2}$$

where the gravity force is ignored. z_b and z_r are displacements of the body and external disturbance input, respectively, measured from static equilibrium. The vibration model can be converted to the standard state-space form given by:

$$\begin{aligned} &\dot{x}_1(t) = x_3(t), \\ &\dot{x}_2(t) = x_3(t) - v(t), \\ &\dot{x}_3(t) = -(k_{coe} / M_b)x_2{}^3(t) + (c / M_b)(-x_3(t) + v(t)) + u(t) / M_b, \\ &y_1(t) = x_2(t), \\ &y_2(t) = x_3(t), \\ &y_3(t) = \dot{x}_3(t), \end{aligned} \tag{6.3}$$

where the three states of $\boldsymbol{x}$, the three outputs of $\boldsymbol{y}$ and the external disturbance input, v, are:

$$\begin{aligned} \boldsymbol{x} &= \left[x_1\ x_2\ x_3\right]^{\mathrm{T}} = \left[z_b \quad (z_b - z_r) \quad \dot{z}_b\right]^{\mathrm{T}}, \\ \boldsymbol{y} &= \left[y_1\ y_2\ y_3\right]^{\mathrm{T}} = \left[(z_b - z_r) \quad \dot{z}_b \quad \ddot{z}_b\right]^{\mathrm{T}}, \\ v &= \dot{z}_r. \end{aligned}$$

In the next section of the chapter, the learning process will involve discrete-time operation. Using the following transform equation:

$$\dot{x}(nT) = \frac{x((n+1)T) - x(nT)}{T}, \tag{6.4}$$

the differential and output equations, (6.3), can be represented by the following discrete-time state-space form:

$$\begin{aligned}
x_1((n+1)T) &= x_1(nT) + T\,x_3(nT), \\
x_2((n+1)T) &= x_2(nT) + T\,x_3(nT) - T\,v(nT), \\
x_3((n+1)T) &= x_3(nT) - T(k_{coe}/M_b)x_2{}^3(nT) \\
&\quad + T(c/M_b)\{-x_3(nT) + v(nT)\} + T\,u(nT)/M_b, \\
y_1(nT) &= x_2(nT), \\
y_2(nT) &= x_3(nT), \\
y_3(nT) &= -(k_{coe}/M_b)x_2{}^3((n-1)T) + (c/M_b)\{-x_3((n-1)T) + v((n-1)T)\} \\
&\quad + u((n-1)T)/M_b,
\end{aligned} \tag{6.5}$$

where n and T denote discrete step number and discrete sampling time, respectively. Strictly speaking in (6.5), it is considered that the output of body acceleration is delayed one step in discrete-time as:

$$y_3(nT) = \dot{x}_3((n-1)T). \tag{6.6}$$

The equations, (6.5), can be written in the following standard difference equation form:

$$\begin{aligned}
\boldsymbol{x}((n+1)T) &= \Phi\ (\boldsymbol{x}(nT), u(nT), v(nT)), \\
y(nT) &= \Psi\ (\boldsymbol{x}(nT), \boldsymbol{x}((n-1)T),\ u((n-1)T),\ v((n-1)T)).
\end{aligned} \tag{6.7}$$

The external disturbance input, v, is considered as a single frequency cosine wave:

$$v(nT) = \cos 2\pi f_{rd} nT\ , \tag{6.8}$$

where f_{rd} is the frequency of the road velocity input.

The actuator force, u, is generated by the output feedback controller described by:

$$u(nT)=F_{NN}(y(nT),w(nT)), \tag{6.9}$$

where $\boldsymbol{w}$ denotes a vector of weighting parameters of the controller and $F_{NN}(\boldsymbol{y},\boldsymbol{w})$ is the function of a multi-layer neural network which will be optimised by the learning process as described in the next section of the chapter.

The system performance is assessed by a non-quadratic cost function of the form:

$$J = \frac{1}{N}\sum_{n=1}^{N}\left(q_1\, y_1^{\,6}(nT) + q_2\, y_2^{\,2}(nT) + q_3\, y_3^{\,2}(nT) + q_4\, u^2(nT)\right), \tag{6.10}$$

where weighting constants q_1, q_2, q_3 and q_4 will be chosen according to the priorities of the four factors: work space, body velocity, body acceleration and control force. These are described in Section 6.3.

6.2 Learning Process with Non-Linear System

A multi-layer neural network is applied to the control of the non-linear single-mass vibration system. The network has the same architecture as described in Section 5.2, having 3, 4 and 1 elements in the input, hidden and output layers, respectively.

The system states, outputs and actuator force are given by the standard discrete-time forms:

$$\begin{aligned} \boldsymbol{x}((n+1)T) &= \Phi(\boldsymbol{x}(nT), u(nT), v(nT)), \qquad (6.11)\\ y(nT) &= \Psi(\boldsymbol{x}(nT), \boldsymbol{x}((n-1)T), u((n-1)T), v((n-1)T)),\\ u(nT) &= F_{NN}(y(nT), \boldsymbol{w}(nT)), \end{aligned}$$

respectively.

The vehicle performance is assessed by the non-quadratic cost, (6.10), which is given by:

$$J(nT) = \frac{1}{N}\sum_{i=n-N+1}^{n} I_{nq}(y(iT), u(iT)), \tag{6.12}$$

where I_{nq} is a non-quadratic function. The weighting vector, $\boldsymbol{w}$, is updated by the gradient method in order to reduce the cost function. A block diagram, showing the system structure, is shown in Fig. 6.2 (a). The update rule is:

$$\boldsymbol{w}((n+1)T) = \boldsymbol{w}((n-N+1)T) + \Delta\, \boldsymbol{w}(nT), \tag{6.13}$$

where

$$\Delta w(nT) = -\delta \frac{\partial\ J(nT)}{\partial\ w(nT)}$$

$$= -\frac{\delta}{N} \sum_{i=n-N+1}^{n} \left(\frac{\partial\ I_{nq}(y(iT),u(iT))}{\partial\ y(iT)} \cdot \frac{\partial\ y(iT)}{\partial\ w(iT)} + \frac{\partial\ I_{nq}(y(iT),u(iT))}{\partial\ u(iT)} \cdot \frac{\partial\ u(iT)}{\partial\ w(iT)} \right).$$

δ is the learning rate and Δw is the weighting vector update. The gradient, $\partial\ J(nT)\ /\partial\ w(nT)$, is evaluated over the interval, $[n - N + 1,\ n]$. The weighting vector is updated at the step, $n + 1$. The sequence of the weighting vector updates in discrete-time is shown in Fig. 6.3.

The Jacobians, $\partial\ I_{nq}(nT)/\partial\ y(nT)$ and $\partial\ I_{nq}(nT)/\partial\ u(nT)$ are the partial derivatives of the cost function with respect to the output vector elements and control force at step, n, respectively. These are obtained by simple differentiation. The Jacobians, $\partial\ y(nT)\ /\partial\ w(nT)$ and $\partial\ u(nT)\ /\partial\ w(nT)$, are the partial derivatives of the output vector elements and control force with respect to the weighting parameters, respectively. These are obtained by the partial differentiation of equations (6.11) with respect to w as:

$$\frac{\partial x}{\partial w}((n+1)T) = \frac{\partial \Phi}{\partial x} \cdot \frac{\partial x}{\partial w}(nT) + \frac{\partial \Phi}{\partial u} \cdot \frac{\partial u}{\partial w}(nT), \quad \text{since } \frac{\partial v}{\partial w} = 0, \qquad (6.14)$$

$$\frac{\partial y}{\partial w}(nT) = \frac{\partial \Psi}{\partial x} \cdot \frac{\partial x}{\partial w}(nT) + \frac{\partial \Psi}{\partial x} \cdot \frac{\partial x}{\partial w}((n-1)T) + \frac{\partial \Psi}{\partial u} \cdot \frac{\partial u}{\partial w}((n-1)T),$$

$$\frac{\partial u}{\partial w}(nT) = \frac{\partial\ F_{NN}(y(nT),w(nT))}{\partial\ y(nT)} \cdot \frac{\partial\ y}{\partial\ w}(nT) + \frac{\partial\ F_{NN}(y(nT),w(nT))}{\partial\ w(nT)},$$

where $\partial\ \Phi(nT)\ /\partial\ x(nT)$ and $\partial\ \Phi(nT)\ /\partial\ u(nT)$ are the sensitivities of the functions, Φ, with respect to the state vector elements and control force, respectively. $\partial\ \Psi(nT)\ /\partial\ x(nT)$ and $\partial\ \Psi(nT)\ /\partial\ u(nT)$ are the sensitivities of the functions, Ψ, with respect to the state vector elements and control force, respectively. Strictly speaking, both $\partial\ \Phi(nT)\ /\partial\ x(nT)$ and $\partial\ \Psi(nT)\ /\partial\ x(nT)$ involve x_2^2. Therefore, they vary according to the value of x_2^2; while both $\partial\ \Phi(nT)\ /\partial\ u(nT)$ and $\partial\ \Psi(nT)\ /\partial\ u(nT)$ are constant. A block diagram, showing the generation of the sensitivity matrices and vectors, is shown in Fig. 6.2 (b). $\partial F_{NN}(y(nT),w(nT))/\partial y(nT)$ and $\partial F_{NN}\ (y(nT),w(nT))/\partial\ w(nT)$ are specified in Appendix A and are calculated by the standard back-propagation algorithm taken from the literature (Hagan, Demuth and Beale, 1996).

6.3 Simulation Results

Three outputs from the non-linear single-mass vibration system, deflection of the suspension, body velocity and body acceleration, are fed back through the multi-layer neural network and its output determines the actuator force.

Three neural network controllers were optimised using a non-quadratic cost function (6.10), but separately, for three different characters. In particular, 'controller 1' compromises between the body motion and deflection of the suspension, 'controller 2' minimises the body motion and 'controller 3' minimises the deflection of the suspension. Each controller was developed using one of three sets of the weighting constants in the cost function as described in Table 6.1.

The update rule of the weighting parameters of the neuro-controller is given by (6.13), where the sampling time, T, is 0.005 second and sampling size of the cost, N, is 100. Each training successfully minimised the cost function under the external velocity input as a single frequency cosine wave (frequency, 2 Hz, and amplitude, 2 m/s). Fig. 6.4 shows the reduction of the cost function during training of 'controller 1'.

The external input and control force response with each controller for each of three cost functions is shown in Fig. 6.5. The suspension deflection and body displacement responses with each controller for each of the three cost functions are shown in Fig. 6.6. From these figures, 'controller 2' can be seen to dramatically reduce body vertical displacement while 'controller 3' reduces suspension deflection. This shows that the three neural networks learned to control the non-linear vibration system correctly according to each cost function.

6.4 Conclusions

It was shown how a multi-layer neural network can be trained to optimise the performance of a non-linear system. The neural network can learn to control the non-linear system correctly according to the given cost function.

Overall, a new path has opened to the final objective of the study, i.e. effective neuro-control of a variable geometry active suspension system, the action of which is essentially non-linear.

Table 6.1 Cost function weighting constants for the three controllers

controlle	Weighting constants			
	q_1	q_2	q_3	q_4
controller 1	1×10^{6}	0	1×10^{-1}	2.5×10^{-7}
controller 2	0	1×10^{2}	0	0
controller 3	1×10^{9}	0	0	0

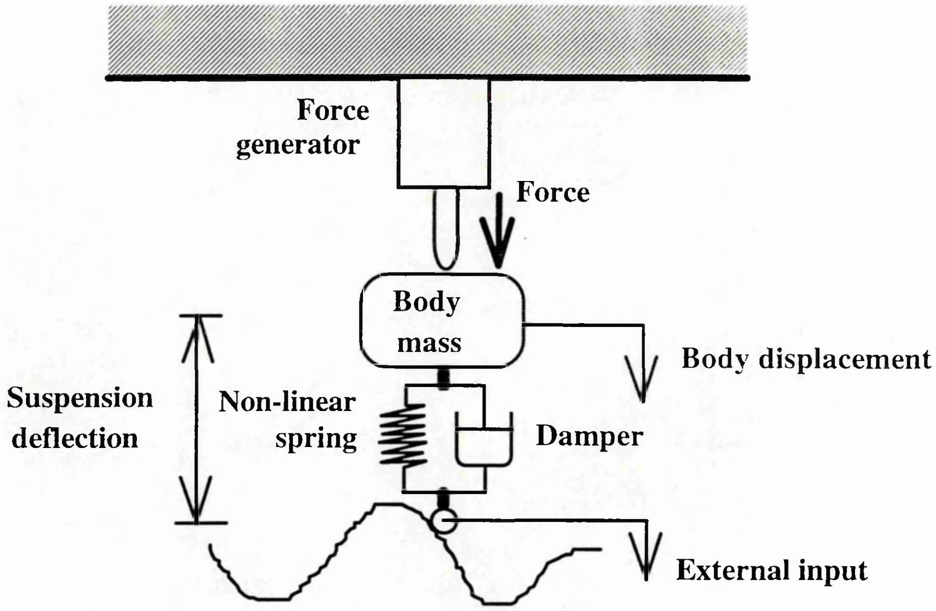
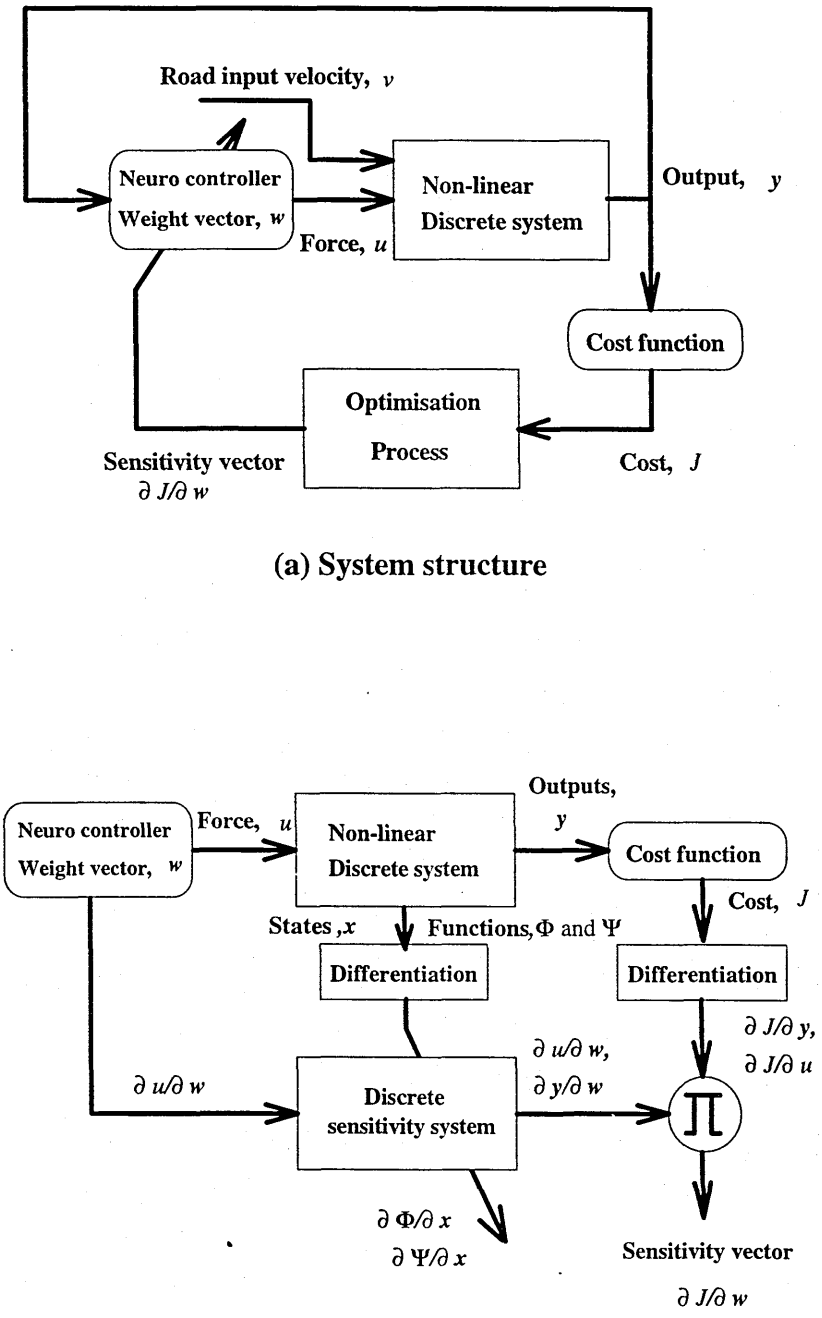


Fig. 6.1 Non-linear single-mass vibration system



(a) System structure

(b) Optimisation process

Fig. 6.2 Diagrammatic representations of (a) system structure and (b) optimisation process, involving non-linear systems Ψ

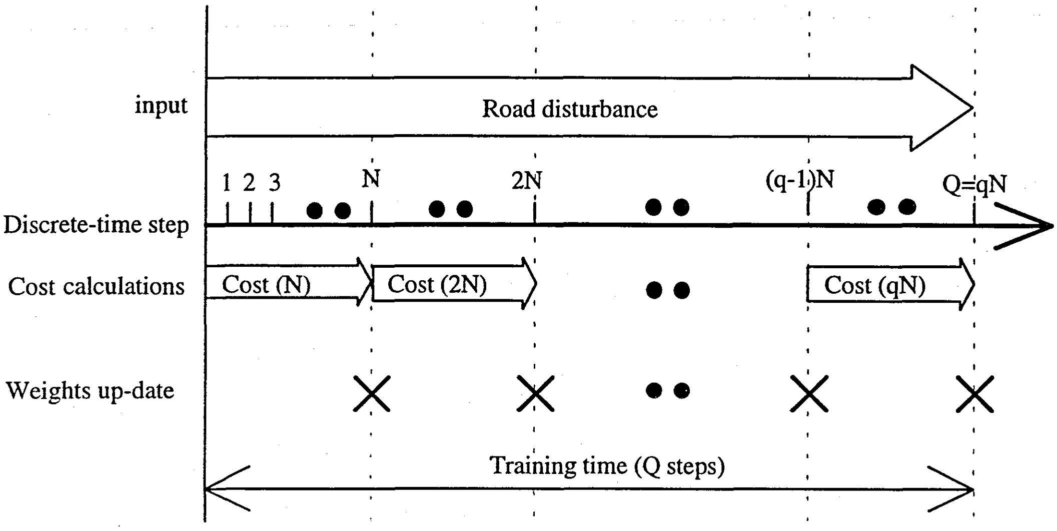


Fig. 6.3 Sequential representation of the learning process in discrete-time

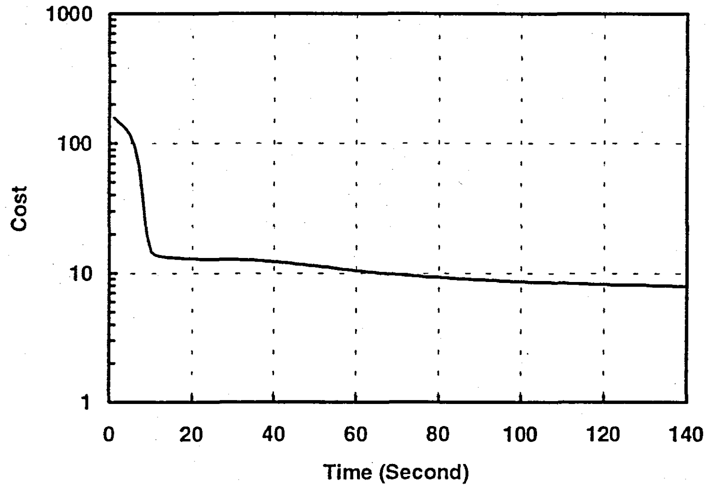
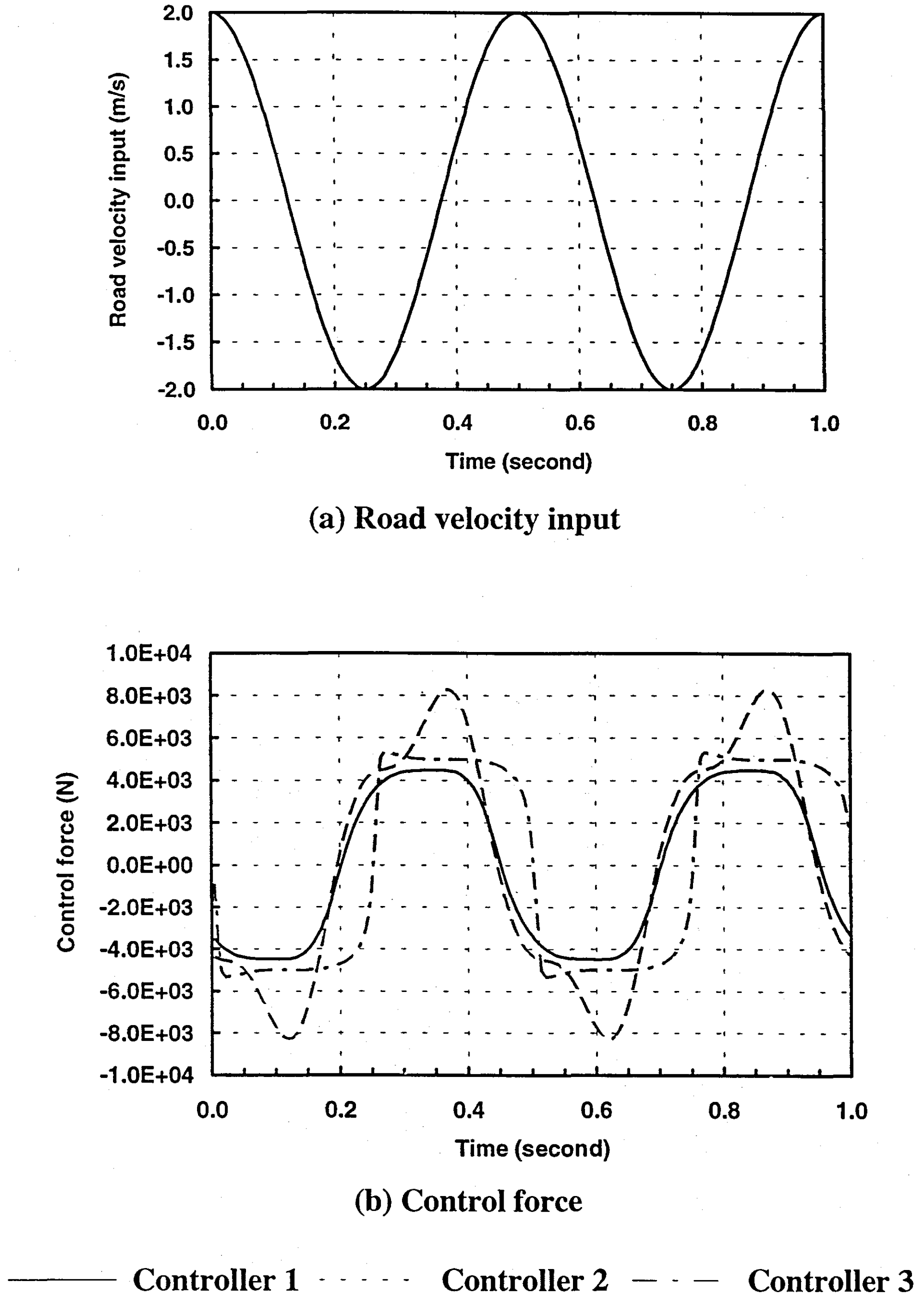


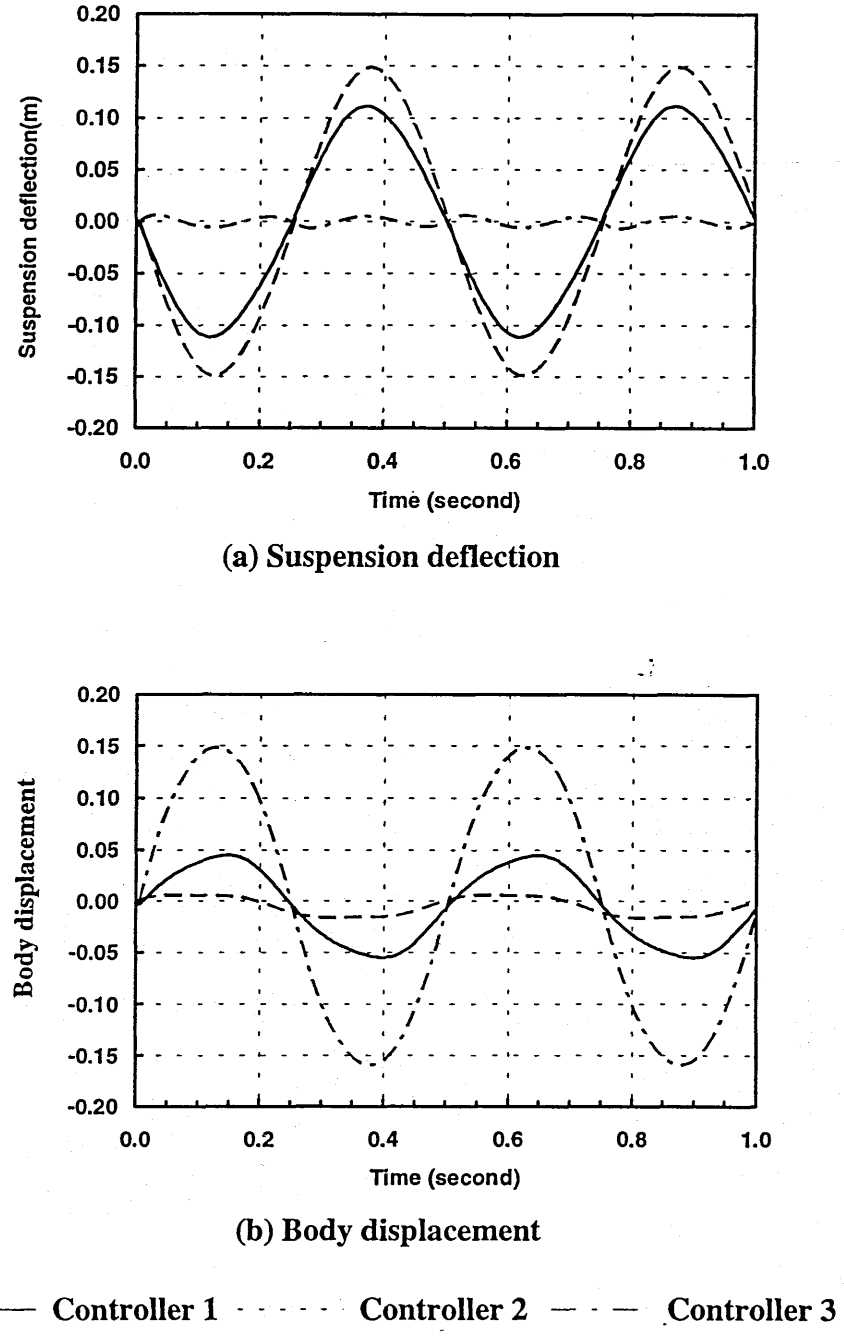
Fig. 6.4 The descending cost function, as learning proceeds with 'controller 1'



(a) Road velocity input

(b) Control force

Fig. 6.5 Road input and control force response with each controller for each of three cost functions: Controller 1 gives a compromise; controller 2 prioritises body control; controller 3 conserves working space.



(a) Suspension deflection

(b) Body displacement

Fig. 6.6 Suspension deflection and body displacement responses with each controller for each of three cost functions: Controller 1 gives a compromise; controller 2 prioritises body control; controller 3 conserves working space.

Chapter 7

Modelling and Control of Variable Geometry Active Suspension System

This chapter considers the modelling and control of a variable geometry active suspension system, whose mechanical design is expected to have considerable potential for solving one of the main problems of automotive active suspensions, that of power consumption. (The power consumption of the system will be described in chapter 9.)

The first section of the chapter deals with a quarter-car suspension model, which involves variable leverage ratio; that of the spring/damper unit length change to the wheel displacement. The leverage ratio may be varied by actuation, which is substantially perpendicular to the suspension force. The developed quarter-car model is made one side of a half-car model in the second section of the chapter. Levelling of the vehicle body with symmetrical actuations on the starboard and port sides is introduced. The third section describes a control scheme, which involves proportional plus differential controls of body height and body roll. (I will use the term 'P + D control' to refer to the proportional plus differential control.) These models and control schemes are composed via the computer language, AUTOSIM, which automatically provides computer simulation programmes. The performance of the P + D controlled half-car system is demonstrated by a comparison with a passive system under ramp lateral force inputs, which involve a maximum steering wheel input velocity of 270 degree/second.

7.1 Quarter-Car Model

The variable geometry active suspension system involves an actively controlled leverage ratio for each wheel, that can be described by:

$$\text{leverage ratio} = \frac{\text{spring/ damper unit deflection}}{\text{wheel displacement}}. \qquad (7.1)$$

Generally speaking, the higher the leverage ratio is - the stiffer the suspension. A possible variable geometry suspension design is shown in Fig. 7.1 (a). In this design, one end of a preloaded spring/damper unit slides on the lower arm via an actuation, which is perpendicular to the suspension unit line of action. However, the problem with this design is that stiffening of suspension and loss of preload may work against each other - generally, the loss of preload is more influential than the stiffening of the suspension. In order to avoid this problem, this design can be improved as shown in Fig. 7.1 (b). The improved design has a circular track for the slideway of a spring/damper unit. The design will not cause any loss of preload but it may be difficult to develop practically.

A quarter-car suspension model, which involves variable geometry without the above problems, is shown in Fig. 7.2. This model includes a lower arm, an upper arm, a hub carrier, a wheel, a spring/damper unit and an inclined track. The lower and upper arms are connected to the ground (*N*) by pin joints at point 1 and point 2, respectively. The hub carrier is connected to the lower arm by a pin joint at point 6, and to the upper arm by a pin joint at point 5. The wheel moves together with the hub carrier and a vertical force is applied to the wheel through point 7. The spring/damper unit acts between point 3 and point 4, where point 3 is connected to the ground while point 4 is movable on the track. The nominal ground coordinates of the seven points are shown in Table 7.1.

An actuation works at point 4, horizontally, and it makes one end of the spring/damper unit move along the track which, in turn, makes an angle of 15 degrees to the lower arm, as shown in Fig. 7.3.

Simulations

Computer simulation programmes of the quarter-car suspension model were developed via the computer language, AUTOSIM, and simple simulations have been done.

Firstly, point 7 was fixed at the nominal point and point 4 was moved along the track. A kinematic relationship between the spring/damper unit length and actuator movement, showing that an actuator movement in an outwards direction increases the preload of the spring/damper unit until the movement of 0.116 m, is shown in Fig. 7.4.

Secondly, vertical forces were applied to the wheels through point 7 with various positions of point 4. Relationships between the wheel forces and spring forces for the various positions of point 4 are shown in Fig. 7.5. (The spring stiffness was 60000 N/m.) The leverage ratios at the various positions of point 4 are shown in Table 7.2. An actuator movement in an outwards direction makes the suspension stiffer (increasing the leverage ratio).

Overall, both preload and leverage ratio work together in the same direction of actuator movement. The presented kinematic design will be used for development of a half-car model, which will be described in the next section of this chapter.

7.2 Development of Half-Car Model

A half-car suspension model, which involves a variable leverage ratio for each wheel, is shown in Fig. 7.6. The model includes a vehicle body and one wheel on each of the starboard and port sides. Each side consists of a lower arm, an upper arm, a hub carrier, a wheel, a spring/damper unit and an inclined track. The model is specified in the nominal ground coordinates as shown in Table 7.3.

A computer simulation programme of the half-car model was developed via the computer language, AUTOSIM, and the following representative model parameters have been used:

Vehicle body mass :	be specified later;
Vehicle body inertia :	100 kg m^2;
Wheel mass :	25 kg;
Wheel camber inertia :	40 kg m^2 *;
Suspension spring stiffness :	60000 N/m;
Suspension damping rate :	6000 N/(m/s);
Tyre stiffness :	2000000 N/m **;
Tyre damping rate :	50 N/(m/s);

Weights and levelling of vehicle bodies

Body weight conditions: normal, light and heavy, are specified in Table 7.4. When a vehicle body mass is specified in the simulation programme, all positions of the mechanical components converge to the equilibrium states.

Table 7.5 shows the vehicle body height change for each of the three body weight conditions. (Both starboard and port actuators stay at their nominal positions on the tracks.)

* It was too late to find the mistake of the parameter setting of the wheel camber inertia. The parameter should be 1 kg m^2. However, the mistake does not affect seriously the following simulation results, as described in Appendix B. ** The parameter was adopted in order to avoid unnecessary body roll caused by tyre deformations.

The positive displacement of the vehicle body is downwards and negative is upwards. As shown in Table 7.5, the vehicle body sags with the normal and heavy body weight conditions and it jacks up with the light body weight condition.

Presumably, the body height is controlled by symmetrical actuations on the starboard and port sides. The two actuations work at point 4 and point 12, horizontally, and each point moves along the inclined tracks on the starboard and port sides, respectively, as shown in Fig. 7.7.

Table 7.6 shows the starboard actuator movements for the three body weight conditions, maintaining the nominal body height. It was shown that in order to maintain the nominal body height, each actuator moves in an outwards direction for the normal and heavy body weight conditions, while they move in an inwards direction for the light weight condition.

Free vibrations

Free vertical vibrations of the vehicle bodies with the three body weight conditions were investigated. Both starboard and port actuators were fixed at the equilibrium positions, which maintaining the nominal body height, for each body weight condition (see Table 7.6). An external vertical force was applied to each body mass for 0.5 second and was then released. Body height responses for the three body weight conditions are shown in Fig. 7.8. Each natural frequency was calculated via a Fourier analysis and the range of the natural frequencies obtained, i.e. 0.82-0.90 Hz, might be of potential benefit for ride comfort (see Table 7.7). The frequency pattern is the reverse of ordinary suspension characteristics. There may be design possibilities of a constant natural frequency.

Lateral and side forces

Forces acting on the half-car model during cornering are shown in Fig. 7.9. Three lateral forces act in the same direction through the body mass centre, starboard wheel mass centre and port wheel mass centre, respectively, while two side forces act in the opposite direction through the ground touching points of the starboard wheel and port wheel, respectively. The lateral forces are given by:

$$F_{Yb} = M_b \cdot L_a \quad \text{(through the body mass centre),} \qquad (7.2)$$

$$F_{YSW} = M_W \cdot L_a \quad \text{(through the starboard wheel mass centre),} \qquad (7.3)$$

$$F_{YPW} = M_W \cdot L_a \quad \text{(through the port wheel mass centre),} \qquad (7.4)$$

where L_a is the lateral acceleration caused by cornering. The side forces of the starboard and port wheels are given by:

$$f_{YS} = -\xi \, (F_{Yb} + F_{YSW} + F_{YPW}), \tag{7.5}$$

$$f_{YP} = -(1-\xi) \, (F_{Yb} + F_{YSW} + F_{YPW}), \tag{7.6}$$

respectively. ξ is the load transfer ratio given by:

$$\xi = \frac{F_{ZS}}{F_{ZS} + F_{ZP}}, \tag{7.7}$$

where F_{ZS} and F_{ZP} are the vertical forces through point 7 and point 15, respectively. The division of the lateral forces between port and starboard sides would not, in practice, be exactly proportional to the vehicle loads. This is a simplification, made for convenience.

7.3 Control Scheme

As described in the previous section of this chapter, body height can be controlled by symmetrical actuations on the starboard and port sides. In a similar way, we can think that body roll can be controlled by anti-symmetrical actuations. The feedback control scheme here involves body roll and height controls, and we must be concerned with how to measure the body height and body roll in a practical way.

Sensing system

Fig. 7.10 shows a vehicle body, which involves a movement in the Z direction and a rotation in roll. The body movement and displacement of the body mass centre, Z_b, are identical but it can not be measured practically. (It can be measured via very expensive equipment, e.g., optical height sensor.) However, supposing there are two measurable displacements at points, P_S and P_P, which are located at a distance of d_Y from the mass centre on the starboard and port sides respectively, we can measure both the body movement and rotation as follows:

When the vehicle body moves and rotates, the displacements of P_S and P_P, will be:

$$Z_S = Z_b + d_Y \cdot \phi, \tag{7.8}$$

$$Z_P = Z_b - d_Y \cdot \phi, \tag{7.9}$$

respectively. Therefore, the displacement of the body mass centre, Z_b, and the body rotation, ϕ, can be obtained by:

$$Z_b = (Z_S + Z_P) / 2, \tag{7.10}$$

$$\phi = (Z_S - Z_P) / 2d_Y. \tag{7.11}$$

Z_S and Z_P can be measured by wheel/body displacement sensors on the starboard and port sides. The starboard displacement sensor is located between P_3 and P_6. The port displacement sensor is located between P_{11} and P_{14}. Z_S and Z_P are:

$$Z_S \cong |d_S|_0 - |d_S|, \tag{7.12}$$

$$Z_P \cong |d_P|_0 - |d_P|, \tag{7.13}$$

respectively. $|d_S|_0$ is the magnitude of the nominal distance between P_3 and P_6, and $|d_P|_0$ is the magnitude of the nominal distance between P_{11} and P_{14}.

P + D body height and roll controls

A detailed control scheme is shown in Fig. 7.11. Outputs of the wheel/body displacement sensors on the starboard and port sides, Z_S and Z_P, are fed back through both the height and roll controllers. The height change can be calculated by the sum of the sensor outputs, while the roll can be calculated by the difference between them. Each controller involves a proportional and a differential gain. The output from the height controller determines symmetrical actuator velocities on the starboard and port sides, while the output from the roll controller determines anti-symmetrical actuator velocities. The actuator velocity commands on the starboard and port sides are:

$$\begin{aligned} V_S &= G_1(Z_S + Z_P) + G_2(\dot{Z}_S + \dot{Z}_P) + G_3(Z_S - Z_P) + G_4(\dot{Z}_S - \dot{Z}_P), \\ V_P &= -G_1(Z_S + Z_P) - G_2(\dot{Z}_S + \dot{Z}_P) + G_3(Z_S - Z_P) + G_4(\dot{Z}_S - \dot{Z}_P), \end{aligned} \tag{7.14}$$

respectively.

The details of computer programming in AUTOSIM corresponding to the above system and the equations created are described in Appendix C.

7.4 Simulation Results

Ramp lateral force inputs were applied to the half-car model through the body (loaded with a body mass of 665 kg) and two wheels. The lateral accelerations corresponding to the force inputs, the maximum value of which is 7.848 m/s^2 (0.8 G), is shown in Fig. 7.12. These inputs may provide a similar circumstance to a maximum steering wheel input velocity of 270 degrees/second.

Sensor outputs were fed back through both height and roll controllers and each control signal was sent to a distributor circuit, where the outputs determined actuator velocity commands on the starboard and port sides. Control gains, which were chosen by trials, are: G_1 = 0.22 (m/s)/m, G_2 = 0.02 (m/s)/(m/s), G_3 = 2.89 (m/s)/m and G_4 = 0.78 (m/s)/(m/s).

Fig. 7.13 (a) and (b) show the starboard and port actuator movements of the P + D controlled system from each equilibrium position, respectively. Fig. 7.14 shows a comparison between body height responses of the P + D controlled and passive systems. In the passive system, both starboard and port actuators were fixed at the equilibrium positions. Fig. 7.15 shows a comparison between body roll responses of the P + D controlled and passive systems.

From the results obtained, the P + D controlled system removes steady state error of the body height change and body roll during 0.8 G cornering. In particular, the P + D controlled system dramatically improved on the roll response when compared with the passive system. However, it caused a severe overshoot of the body height change, which should be reduced.

7.5 Conclusions

It was shown how the leverage ratio, i.e. that of the spring/damper unit length change to the wheel displacement, can be varied by actuation which is substantially perpendicular to the suspension force. The developed half-car model, which involves the variable geometry, demonstrated that the nominal vehicle body height was maintained by symmetrical actuations on the starboard and port sides for the various vehicle weight conditions.

The P + D controlled system successfully removed steady state error of both the body height change and body roll for ramp lateral force inputs, which involved maximum lateral acceleration of 0.8 G. Although the P + D controller was one of the best performing controllers in the author's trials, it caused a severe transient response of the body height change. This weakness may be reduced by a learning control system, that will be described in the next chapter.

Table 7.1 The nominal ground coordinates of the points in the quarter-car variable geometry suspension model

Point	*Y* axis (m)	*Z* axis (m)	Description
O	0	0	Ground coordinate system origin
1	0.3	-0.15	Lower arm inner pivot
2	0.4	-0.35	Upper arm inner pivot
3	0.485	-0.55	Spring/damper unit top mounting point
4	0.485	-0.165	Spring/damper unit bottom mounting point
5	0.67	-0.5	Upper arm outer pivot
6	0.67	-0.18	Lower arm outer pivot
7	0.75	0	Tyre contact point

Table 7.2 Values of the leverage ratio for the various actuator movements

Actuator movement (m)	Leverage ratio (—)
-0.06	0.31
-0.03	0.39
-0.015	0.43
0	0.47
0.015	0.51
0.03	0.55
0.06	0.63

Table 7.3 The nominal ground coordinates of the points in the half-car variable geometry active suspension model; (S) and (P) denote the starboard and port sides, respectively

Point	*Y* axis (m)	*Z* axis (m)	Description
O	0	0	Ground coordinate system origin
1	0.3	-0.15	Lower arm inner pivot (S)
2	0.4	-0.35	Upper arm inner pivot (S)
3	0.485	-0.55	Spring/damper unit top mounting point (S)
4	0.485	-0.165	Spring/damper unit bottom mounting point (S)
5	0.67	-0.5	Upper arm outer pivot (S)
6	0.67	-0.18	Lower arm outer pivot (S)
7	0.75	0	Tyre contact point (S)
8	0.75	-0.34	Wheel mass centre (S)
9	-0.3	-0.15	Lower arm inner pivot (P)
10	-0.4	-0.35	Upper arm inner pivot (P)
11	-0.485	-0.55	Spring/damper unit top mounting point (P)
12	-0.485	-0.165	Spring/damper unit bottom mounting point (P)
13	-0.67	-0.5	Upper arm outer pivot (P)
14	-0.67	-0.18	Lower arm outer pivot (P)
15	-0.75	0	Tyre contact point (P)
16	-0.75	-0.34	Wheel mass centre (P)
17	0	-0.5	Vehicle body mass centre

Table 7.4 Vehicle body weight conditions

	Light (kg)	Normal (kg)	Heavy (kg)
Vehicle mass	1000	1000	1000
Petrol	0	60	60
Passengers	0	60 * 2 (people)	60 * 4 (people)
Luggage	0	0	30
Total	1000	1180	1330
Half mass	500	590	665

Table 7.5 Vehicle body height change without actuations for the three body weight conditions

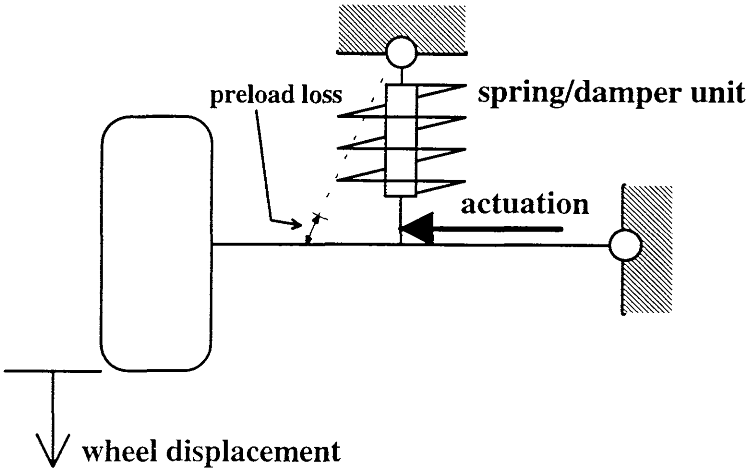
Body weight condition	Vehicle body mass (kg)	Vehicle body height change (m)
Light	500	-0.013
Normal	590	0.033
Heavy	665	0.084

Table 7.6 Starboard actuator movements for the three body weight conditions; maintaining the nominal body height (The port actuator works symmetrically)

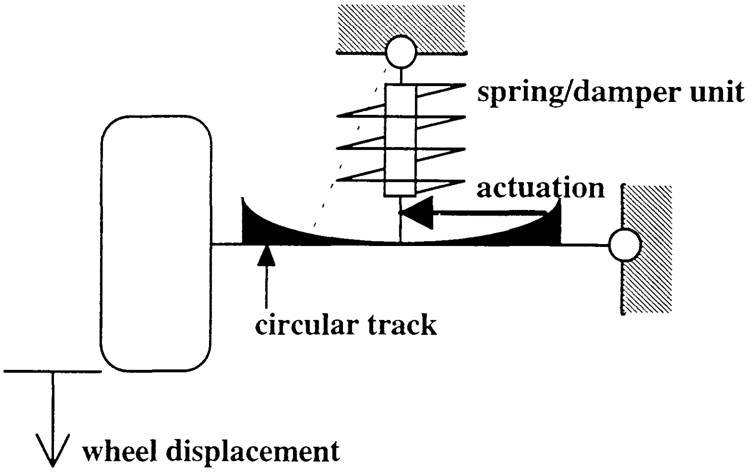
Body weight condition	Vehicle body mass (kg)	Starboard actuator movement (m)
Light	500	-0.0058
Normal	590	0.0129
Heavy	665	0.0282

Table 7.7 Natural frequencies of vehicle bodies for the three body weight conditions (Both starboard and port actuators are fixed at the equilibrium positions, which maintain the nominal body height, for each body weight condition)

Body weight condition	Vehicle body mass (kg)	Natural frequency of vertical vehicle body motion (Hz)
Light	500	0.82
Normal	590	0.86
Heavy	665	0.9



(a) Variable geometry design with a sliding spring/damper unit end on a lower arm



(b) Variable geometry design with a sliding spring/damper unit end on a circular track

Fig. 7.1 Alternative designs of variable geometry suspension systems

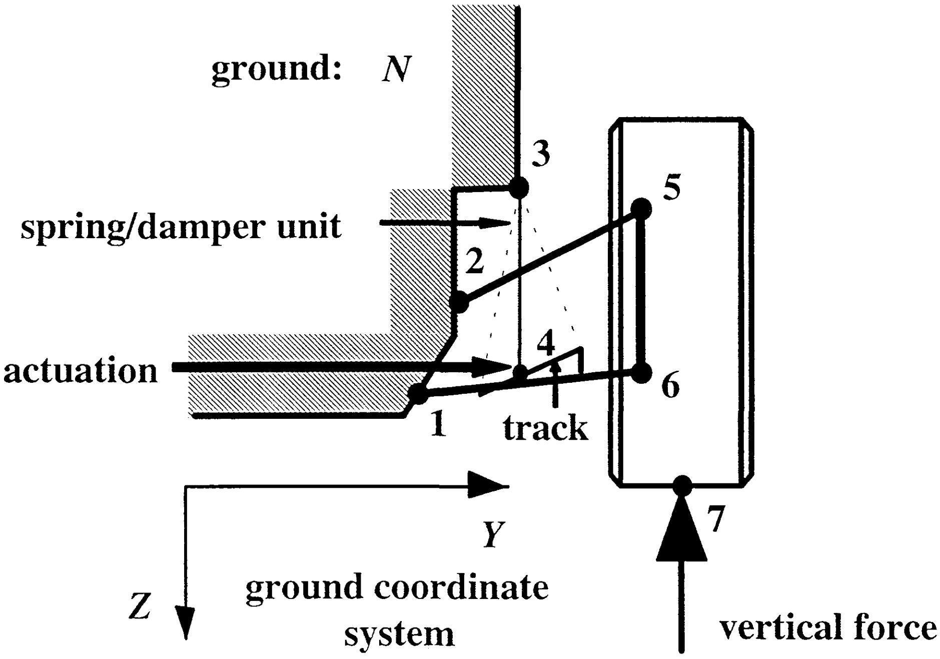


Fig. 7.2 Quarter-car variable geometry suspension model

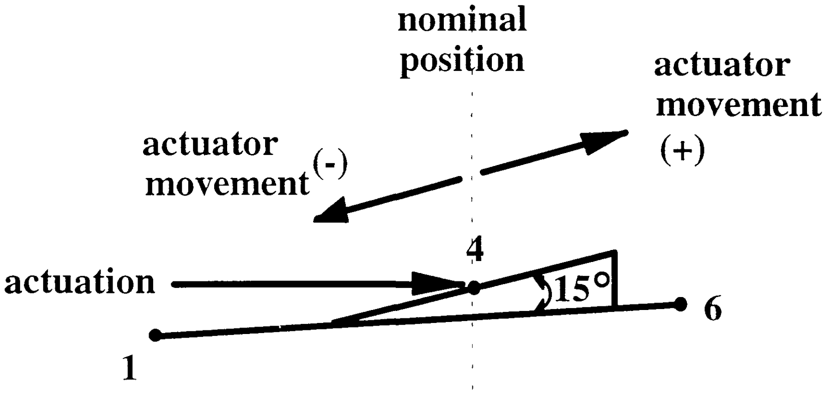


Fig. 7.3 Actuator movement on the inclined track

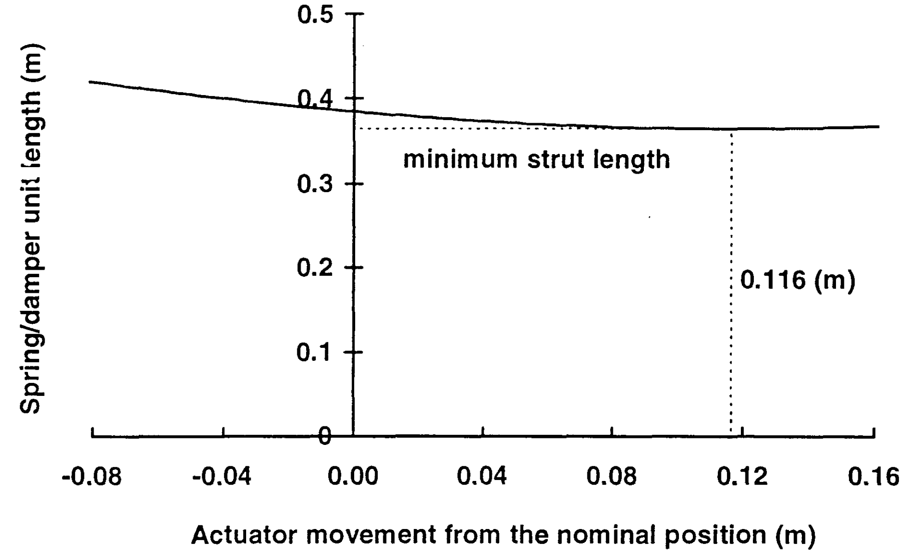


Fig. 7.4 Relationship between the spring/damper unit length and actuator movement without wheel displacement

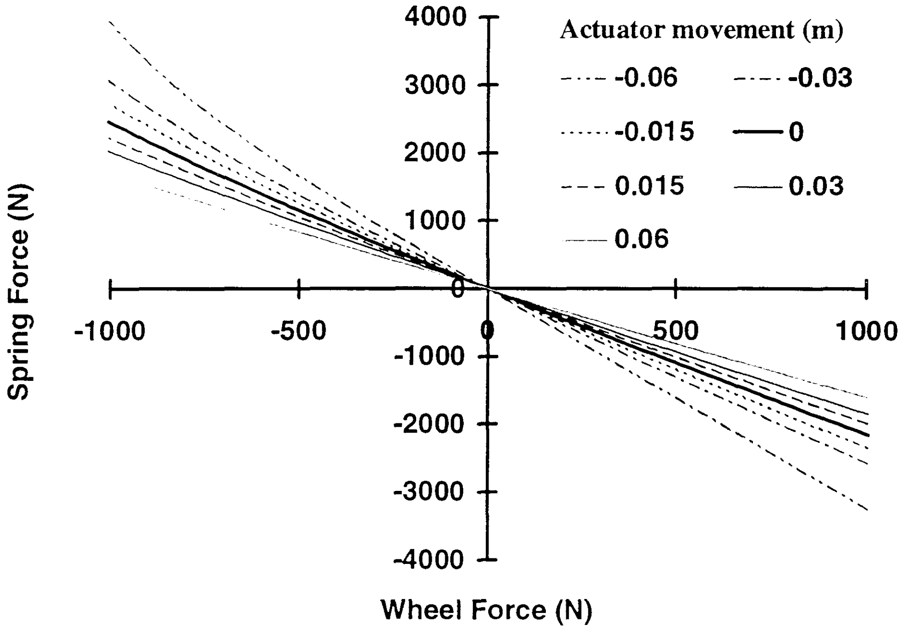


Fig. 7.5 Relationships between the wheel forces and spring forces for the various positions of one end of the spring/damper unit via actuation

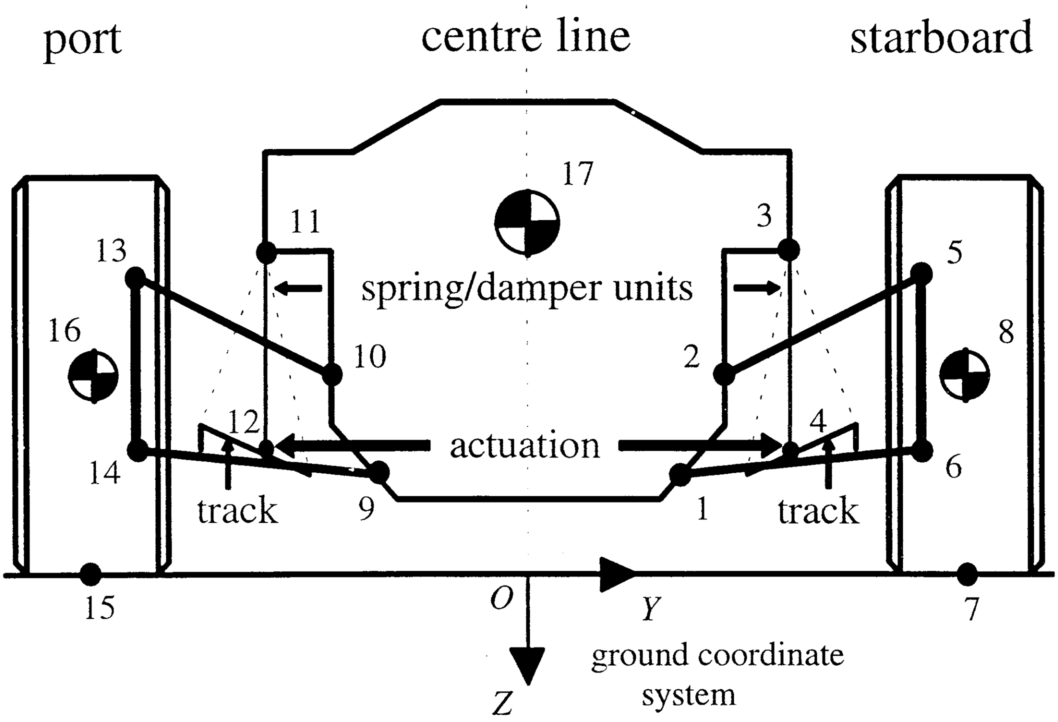


Fig. 7.6 Half-car variable geometry active suspension model (Vehicle front end viewed from rear)

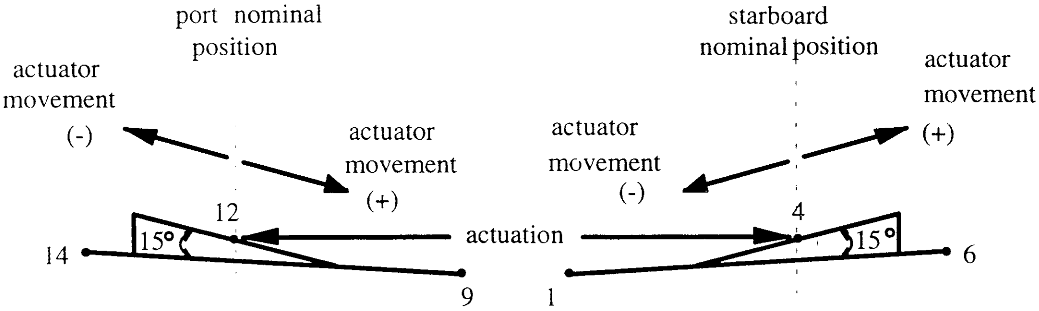


Fig. 7.7 Actuator movements on the inclined tracks on the starboard and port sides

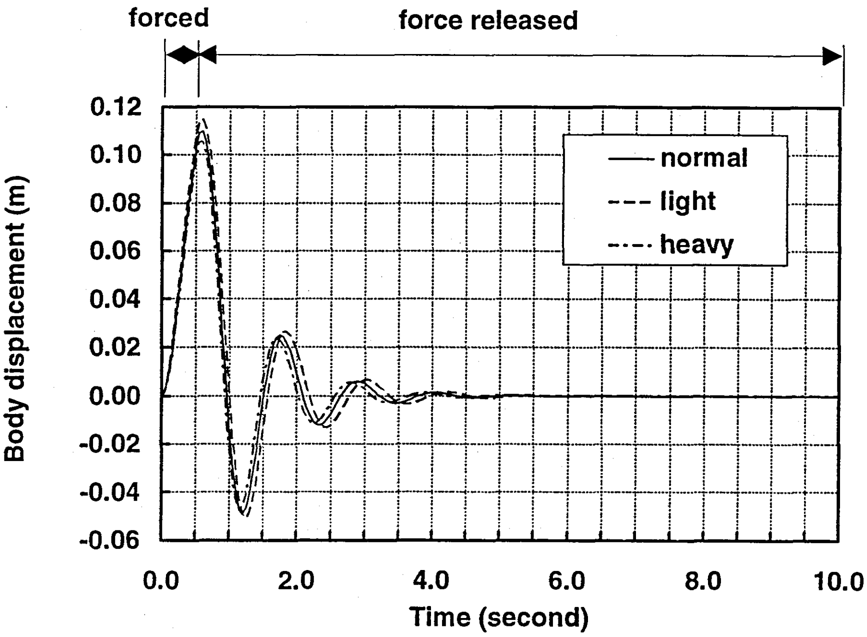


Fig. 7.8 Free vertical vehicle body motions with three body weight conditions: light (500 kg), normal (590 kg) and heavy (665 kg)

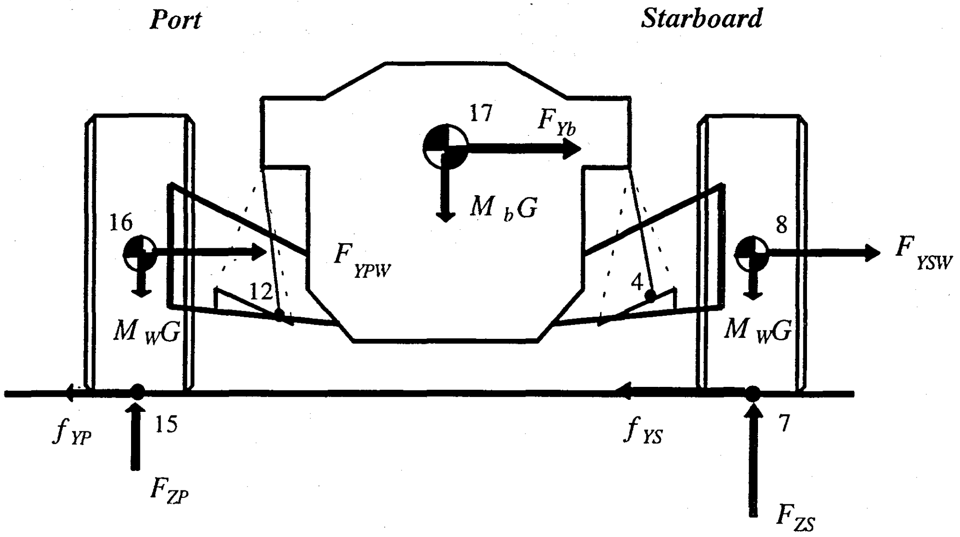


Fig. 7.9 Acting forces during cornering (Vehicle front end viewed from rear)

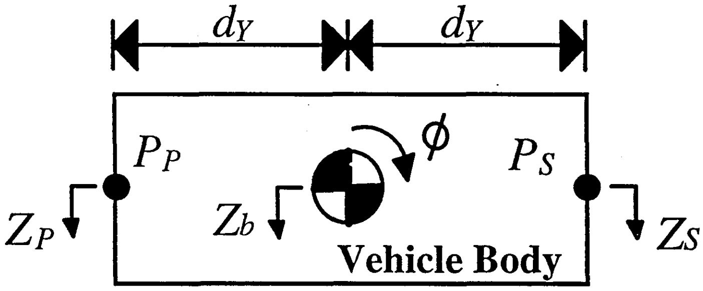


Fig. 7.10 Vehicle body, which moves in Z direction and rotates in roll

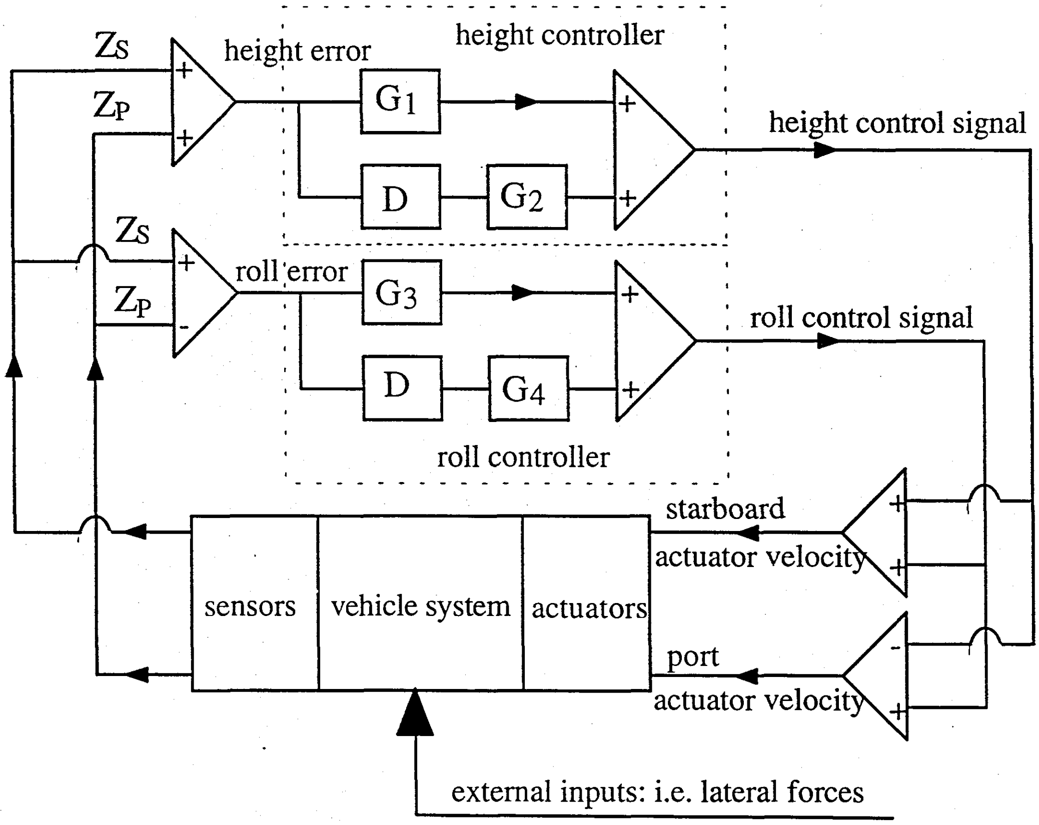


Fig. 7.11 Detailed feedback control scheme of variable geometry active suspension system

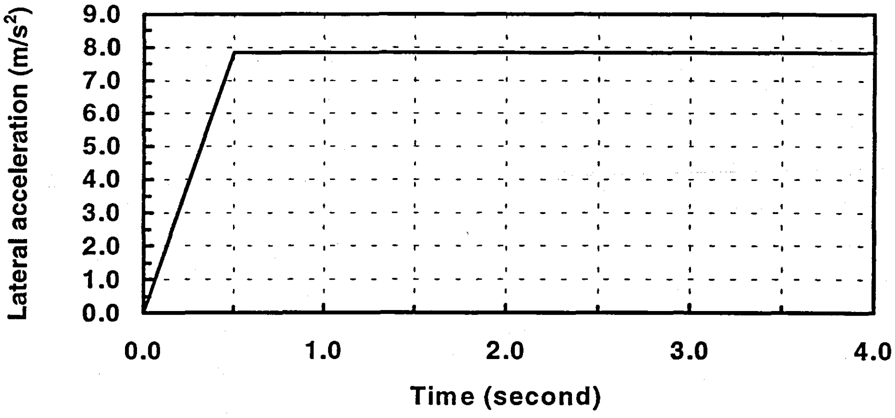
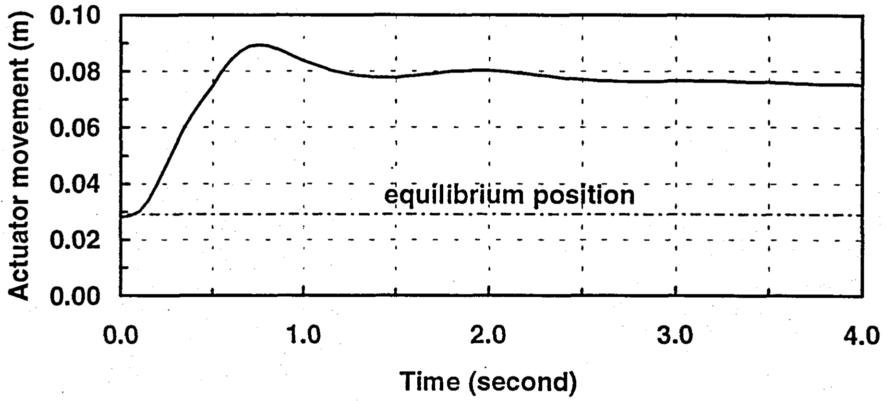
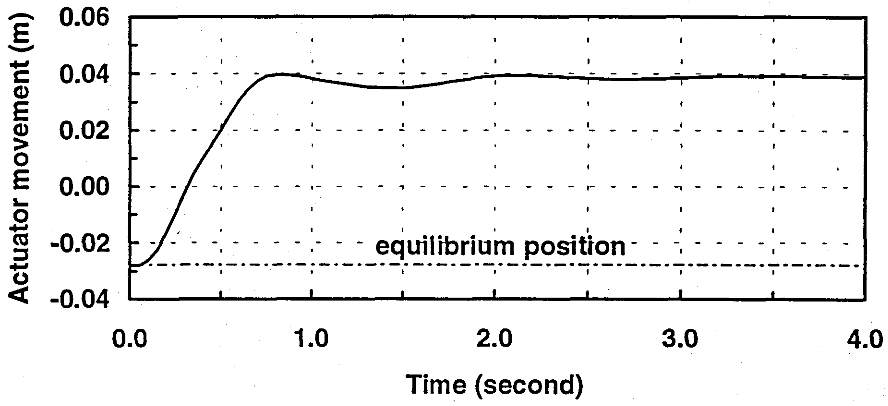


Fig. 7.12 Lateral acceleration corresponding to the ramp force input



(a) Starboard actuator movement



(b) Port actuator movement

Fig. 7.13 Actuator movements on the starboard and port sides from the equilibrium positions (vehicle body mass : 665 kg)

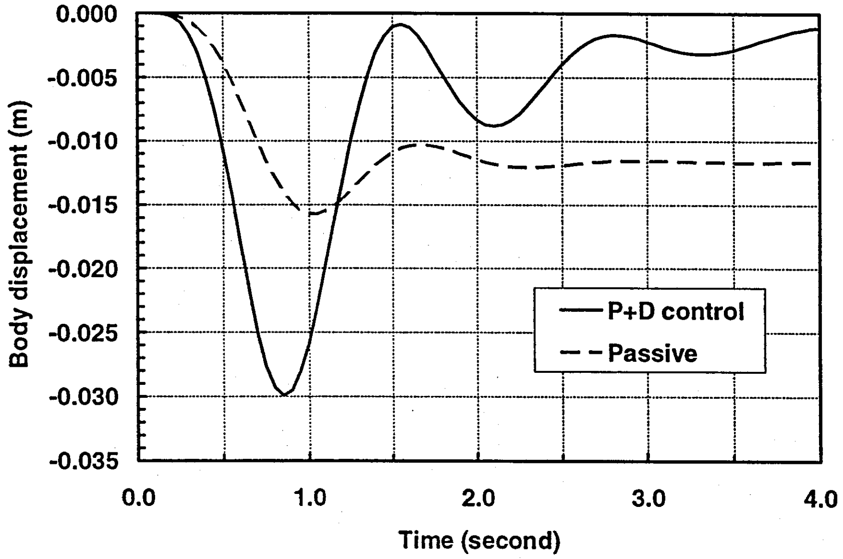


Fig. 7.14 Comparison between vehicle body height change responses of the P + D controlled and passive systems during 0.8 G cornering (vehicle body mass : 665 kg)

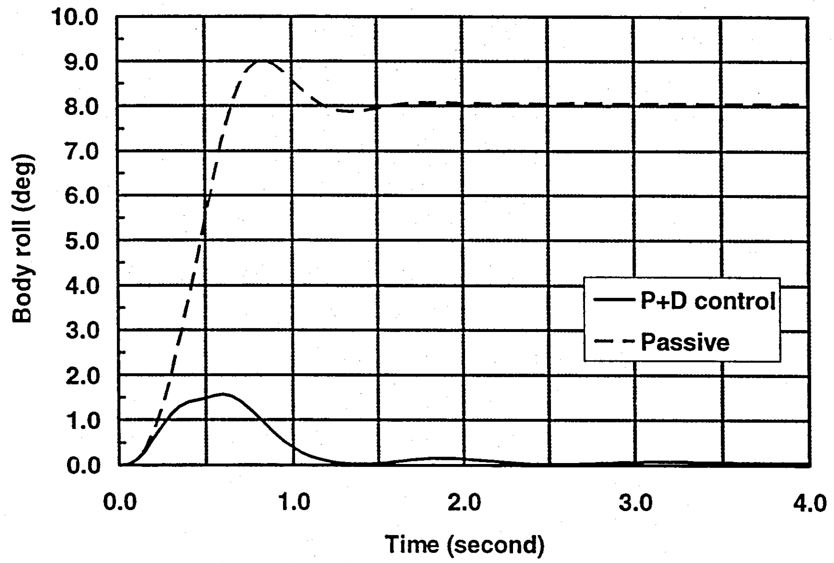


Fig. 7.15 Comparison between vehicle body roll responses of the P + D controlled and passive systems during 0.8 G cornering (vehicle body mass : 665 kg)

Chapter 8

Neuro-Control of Variable Geometry Active Suspension System

In chapter 7, the variable geometry active suspension system was developed and controlled by P + D body height and roll controllers. Control gains in each controller were chosen by hand-trials. In this chapter, two neural networks are applied in a similar manner, but they learn and optimise by themselves to reduce a quadratic cost function to its minimum value. (I will use the term 'neuro-control' or 'neuro-controller' to refer to the two neural networks for the body height and roll controls together.)

The first section of the chapter deals with the full system, which includes the vehicle suspension model and neuro-controller, and the learning process involving a cost function. The second section describes how a simulation programme can be provided using the two computer packages, AUTOSIM and MATLAB/Neural Network Toolbox. In the third section, the learning process is tracked. The performances of the neuro-controlled system are compared with those of passive and P + D controlled systems under various conditions, which involves various levels of lateral force inputs and vehicle body weight changes.

8.1 The System and Learning Process

The full system is depicted diagrammatically in Fig. 8.1 (a). External inputs, i.e. lateral forces, are applied to the vehicle system. Outputs from the vehicle system are fed back through the neuro-controller and its outputs determine actuator velocities on the starboard and port sides. Vehicle states and actuator velocities are used to form a cost function, which is used, through the optimisation process, to update the weighting parameters of the neuro-controller.

The vehicle system can be represented in the following standard difference equation form:

$$\begin{aligned} \boldsymbol{x}((n+1)T) &= \Phi\left(\boldsymbol{x}(nT),\boldsymbol{u}(nT),\boldsymbol{v}(nT)\right), \\ \boldsymbol{y}(nT) &= \Psi(\boldsymbol{x}(nT)), \end{aligned} \tag{8.1}$$

where n and T denote discrete step number and discrete sampling time, respectively. The ten states of $\boldsymbol{x}$, the four outputs of $\boldsymbol{y}$, the control inputs, $\boldsymbol{u}$, and the external disturbance inputs, $\boldsymbol{v}$, are:

state vector, $\boldsymbol{x}$:
x_1: vehicle body height change;
x_2: vehicle body roll;
x_3: lower arm (S) rotation relative to vehicle body;
x_4: wheel (S) rotation relative to lower arm (S);
x_5: lower arm (P) rotation relative to vehicle body;
x_6: wheel (P) rotation relative to lower arm (P);
x_7: vehicle body height speed;
x_8: vehicle body roll rate;
x_9: lower arm (S) rotation speed relative to vehicle body;
x_{10}: lower arm (P) rotation speed relative to vehicle body;

where (S) and (P) denote the starboard and port sides, respectively.

output vector, $\boldsymbol{y}$:
y_1: sum of wheel/body displacements on the starboard and port sides;
y_2: sum of wheel/body velocities on the starboard and port sides;
y_3: difference between wheel/body displacements on the starboard and port sides;
y_4: difference between wheel/body velocities on the starboard and port sides;

control input, $\boldsymbol{u}$:
u_1: starboard actuator displacement;
u_2: port actuator displacement;

disturbance inputs, $\boldsymbol{v}$: lateral forces applied to the body and wheels.

The vehicle parameters were described in section 7.2 in chapter 7. Each lateral force input, v_j (j = 1,2,3), is considered as one batch of a ramp input having N data points in discrete-time:

$$v_j(iT) = \begin{cases} \dfrac{v_{j\max} iT}{0.5}, & (0 \le iT \le 0.5) \\ v_{j\max}, & (0.5 < iT \le NT) \end{cases} \qquad (8.2)$$

where v_j rises linearly to the maximum value of $v_{j\max}$ in 0.5 second and then is constant.

The controller involves two multi-layer neural networks for body height and body roll control, respectively, as shown in Fig. 8.2. Each network includes; 2, 3 and 1 elements in the input, hidden and output layers, respectively. Two outputs from the vehicle system, y_1 and y_2, are fed back through the network for the height control and its output determines symmetrical actuator velocities on the starboard and port sides, while the other two outputs, y_3 and y_4, are fed back through the other network and its output determines anti-symmetrical actuator velocities.

The actuator velocities on the starboard and port sides can be represented by:

$$\dot{u}(nT) = F_{NN}(y(nT), w(nT)), \qquad (8.3)$$

where the weighting vector, w, involves all of the weighting parameters of the two neural networks.

Using the following transform equation:

$$\dot{u}(nT) = \frac{u((n+1)T) - u(nT)}{T}, \qquad (8.4)$$

the actuator displacements can be represented by:

$$u((n+1)T) = u(nT) + T\ F_{NN}(y(nT), w(nT)). \qquad (8.5)$$

Cost function and learning process

The system performance is assessed by a quadratic cost function of the form:

$$J = \frac{1}{N}\sum_{n=1}^{N}\left(q_1\, x_1^{\,2}(nT) + q_2\, x_2^{\,2}(nT) + q_3\, \dot{u}_1^{\,2}(nT) + q_4\, \dot{u}_2^{\,2}(nT)\right), \qquad (8.6)$$

where $q_1 = 26013$, $q_2 = 6096$, $q_3 = 300$ and $q_4 = 300$, as chosen by author's trials. The cost function can be written:

$$J(nT) = \frac{1}{N}\sum_{i=n-N+1}^{n} I_q(x(iT), \dot{u}(iT)), \qquad (8.7)$$

where I_q is a quadratic function. The weighting vector, $\boldsymbol{w}$, is updated by the gradient method in order to reduce the cost function. The update rule is:

$$w((n+1)T) = w((n-N+1)T) + \Delta w(nT), \qquad (8.8)$$

where

$$\Delta w(nT) = -\delta \frac{\partial\, J(nT)}{\partial\, w(nT)}$$

$$= -\frac{\delta}{N}\sum_{i=n-N+1}^{n}\left(\frac{\partial\, I_q(x(iT),\dot{u}(iT))}{\partial\, x(iT)}\cdot\frac{\partial\, x(iT)}{\partial\, w(iT)} + \frac{\partial\, I_q(x(iT),\dot{u}(iT))}{\partial\, \dot{u}(iT)}\cdot\frac{\partial\, \dot{u}(iT)}{\partial\, w(iT)}\right).$$

δ is the learning rate and Δw is the weighting vector update. The gradient, $\partial\, J(nT) /\partial\, w(nT)$, is evaluated over the interval, $[n - N + 1,\ n]$, i.e. one batch. The weighting vector is updated at the step, $n + 1$. The sequence of the weighting vector updates in discrete-time is shown in Fig. 8.3.

A block diagram, showing the generation of the gradient, $\partial\, J(nT) /\partial\, w(nT)$, is shown in Fig. 8.1 (b). The Jacobians, $\partial\, I_q(nT)/\partial\, x(nT)$ and $\partial\, I_q(nT)/\partial\, \dot{u}(nT)$ are the partial derivatives of the cost function with respect to the state vector elements and actuator velocities at step, n, respectively. These are obtained by simple differentiation. The Jacobians, $\partial\, x(nT) /\partial\, w(nT)$ and $\partial\, \dot{u}(nT) /\partial\, w(nT)$, are the partial derivatives of the state vector elements and actuator velocities with respect to the weighting parameters, respectively. These are obtained by the partial differentiation of equations (8.1), (8.3) and (8.5) with respect to $\boldsymbol{w}$ as:

$$\frac{\partial x}{\partial w}((n+1)T)=\frac{\partial \Phi}{\partial x}\cdot\frac{\partial x}{\partial w}(nT)+\frac{\partial \Phi}{\partial u}\cdot\frac{\partial u}{\partial w}(nT), \quad \text{since}\ \frac{\partial v}{\partial w}=0, \tag{8.9}$$

$$\frac{\partial y}{\partial w}(nT)=\frac{\partial \Psi}{\partial x}\cdot\frac{\partial x}{\partial w}(nT), \tag{8.10}$$

$$\frac{\partial \dot{u}}{\partial w}(nT)=\frac{\partial F_{NN}(y(nT),w(nT))}{\partial y(nT)}\cdot\frac{\partial y}{\partial w}(nT)+\frac{\partial F_{NN}(y(nT),w(nT))}{\partial w(nT)}, \tag{8.11}$$

$$\frac{\partial u}{\partial w}((n+1)T)=\frac{\partial u}{\partial w}(nT)+T\cdot\frac{\partial \dot{u}}{\partial w}(nT), \tag{8.12}$$

where $\partial\ \Phi(nT)\ /\partial\ x(nT)$ and $\partial\ \Phi(nT)\ /\partial\ \boldsymbol{u}(nT)$ are the sensitivities of the functions, Φ, with respect to the state vector elements and actuator displacements, respectively. $\partial\ \Psi(nT)\ /\partial\ x(nT)$ denotes the sensitivities of the functions, Ψ, with respect to the state vector elements. These three sensitivity matrices vary according to the values of $\boldsymbol{x}$ and $\boldsymbol{u}$ (In the simulation programme, these three sensitivity matrices are updated every time step, as described in the next section of this chapter). $\partial F_{NN}(y(nT),w(nT))/\partial y(nT)$ and $\partial F_{NN}\ (y(nT),w(nT))/\partial\ w(nT)$ are specified in Appendix D and are calculated in a similar manner to that of the standard back-propagation algorithm taken from the literature (Hagan, Demuth and Beale, 1996).

8.2 Simulation Programme

The half-car variable geometry active suspension model was developed via the computer language, AUTOSIM, which automatically provides a computer simulation programme. The provided simulation programme involved the vehicle equations (8.1) and it was written in FORTRAN. On the other hand, the neuro-controller was provided via the computer package, MATLAB/Neural Network Toolbox.

Nevertheless, the vehicle equations written in FORTRAN can be converted to a MATLAB function-file. (MATLAB function-files may be used in a similar manner to a subroutine of a FORTRAN programme.) Therefore, it was decided that the full system would be implemented via one simulation programme in MATLAB code.

A flowchart of the simulation programme of the full system is shown in Fig. 8.4. This simulation programme involves the following MATLAB function-files:

PCCOMP.M

This function-file contains the pre-computing of constants for the vehicle equations.

FCT.M

This function-file computes the vehicle equations using the pre-computed constants given by 'PCCOMP.M' and a numerical integrator.

Each of 'PCCOMP.M' and 'FCT.M' was converted from the corresponding subroutine of the FORTRAN programme, which was provided via AUTOSIM.

PCSENS.M

This function-file contains the pre-computing of constants for vehicle sensitivity matrices.

SENS.M

This function-file computes the vehicle sensitivity matrices: $\partial \Phi / \partial \boldsymbol{x}$, $\partial \Phi / \partial \boldsymbol{u}$ and $\partial \Psi / \partial \boldsymbol{x}$, according to the operating states and inputs, using the pre-computed constants given by 'PCSENS.M'.

AUTOSIM involves an option to automatically compute these sensitivity matrices, and write the elements of the matrices into a file in the form of MATLAB. Each of 'PCSENS.M' and 'SENS.M' is a part of such a file.

JACOB.M

This function-file computes equations (8.9) and (8.10).

NECON.M

This function-file contains both forward and back propagation of the neuro-controller. The former provides $\boldsymbol{u}$ and $\dot{\boldsymbol{u}}$ with equations (8.4) and (8.5) respectively, while the latter provides $\partial \boldsymbol{u} / \partial \boldsymbol{w}$ and $\partial \dot{\boldsymbol{u}} / \partial \boldsymbol{w}$ with equations (8.11) and (8.12) respectively. This function-file involves M-files from MATLAB/Neural Network Toolbox, e.g. 'TANSIG.M' (tangent sigmoid function).

ERR.M

This function-file calculates the cost function.

WUPDATE.M

This function-file contains the update of the neuro-controller weighting parameters.

LACC.M

This function-file generates the lateral force inputs given by equation (8.2).

8.3 Simulation Results

Firstly, each of two neural networks learned to mimic the relevant input-output relationship of each of the P + D height and roll controllers, which were described in chapter 7. The neuro-controller successfully achieved to perform the P + D control system as described in Appendix E.

Secondly, the neuro-controller was installed into the system and learned to reduce the cost function (8.6). The batch lateral force inputs, involving maximum lateral acceleration of 7.848 m/s^2 (0.8 G) depicted in Fig. 8.5, were repeatedly applied every 4 seconds to the vehicle system loaded with a body mass of 665 kg. The cost function was successfully reduced from the initial cost, 4.5, to the final cost, 1.8, in 4612 seconds (1153 batches).

Fig. 8.6 (a) shows a comparison between starboard actuator responses with the neuro-control and P + D control. Fig. 8.6 (b) shows a comparison between port actuator responses in a similar manner.

Fig. 8.7 (a) shows a comparison between vehicle body height responses with the neuro-control and P + D control. Fig. 8.7 (b) shows a comparison between vehicle body roll responses in a similar manner. From these figures, the neuro-control not only dramatically reduced overshoot of the vehicle body height change, but also improved on the body roll response.

The neuro-controller was tested for various conditions, which involve different levels of ramp lateral force inputs and vehicle body weights. Table 8.1 shows an overall comparison between neuro, P + D, and passive systems in the vehicle responses: body height change, body roll and actuator movements; and cost under the various conditions.

From the table, the neuro-control reduces the overshoots of body height change and body roll, and achieves the minimum cost function in each condition.

8.4 Conclusions

It was shown how the neuro-controller is trained to optimise the performance of the variable geometry active suspension system. The two computer packages, AUTOSIM and MATLAB/Neural Network Toolbox, can be used together for the purpose. (The computer package, SIMULINK, may be an alternative. It will be described in chapter 9.)

The neuro-controller successfully reduced the cost function and, particularly, it improved the transient response of the body height change under all conditions tried, which involve various levels of ramp lateral force inputs and vehicle body weight changes.

One final point which needs to be clarified is the power consumption. This will be taken up in the next chapter.

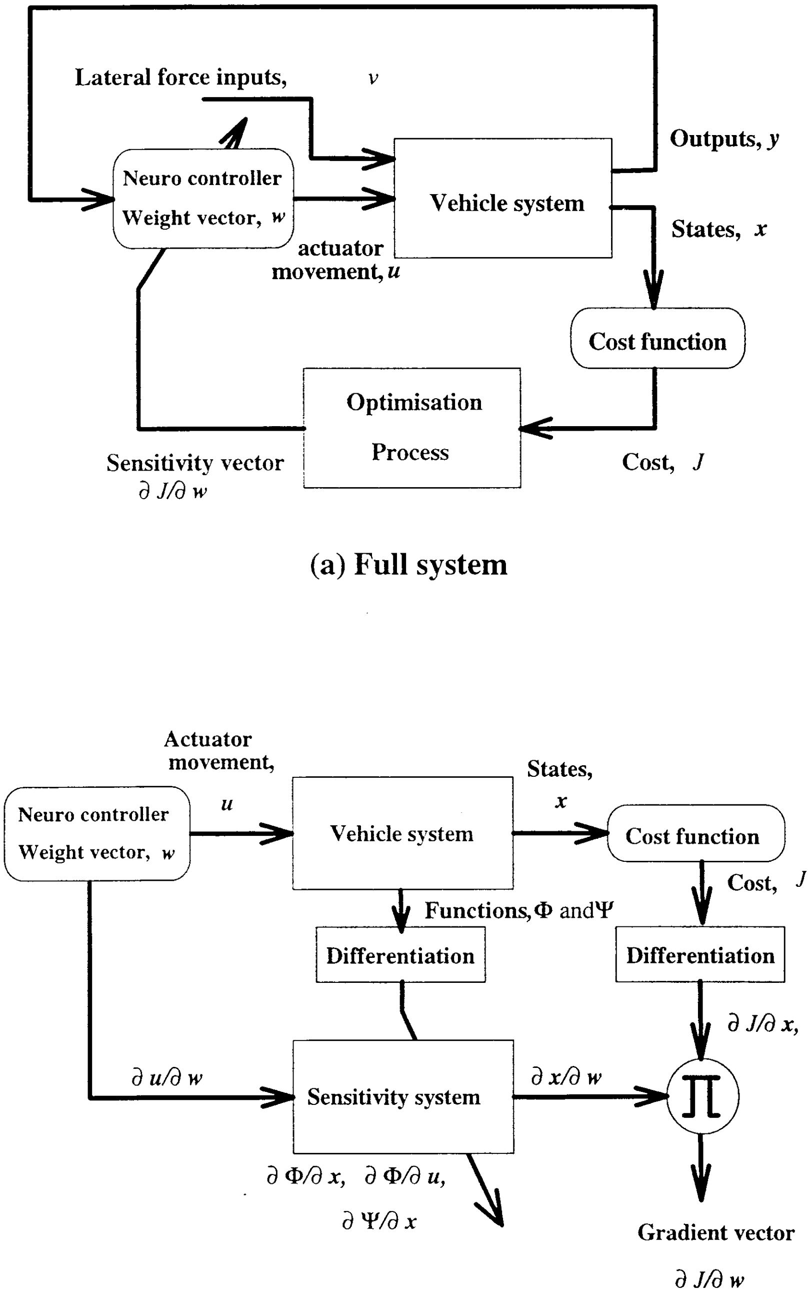
Table 8.1 Overall comparison between neuro-control, P + D control and passive systems in vehicle responses: vehicle body height change, body roll and actuator movements; and cost function

Condition	Control system	Body height change (mm)		Body roll (deg)	
		Overshoot*	Steady state	Overshoot*	Steady state
Condition 1	Neuro	-10.4	-0.0	1.0	0.1
0.8 G	P + D	-28.8	-1.1	1.5	0.1
665 kg	No-control	-4.0	-11.7	0.9	8.1
Condition 2	Neuro	-2.1	-0.0	0.5	0.0
0.4 G	P + D	-5.5	-0.4	0.7	0.0
665 kg	No-control	-0.8	-2.2	0.4	3.7
Condition 3	Neuro	-0.5	0.0	0.2	0.0
0.2 G	P + D	-1.3	-0.1	0.3	0.0
665 kg	No-control	-0.2	-0.5	0.2	1.8
Condition 4	Neuro	-10.4	-0.0	1.0	0.1
0.8 G	P + D	-31.9	-1.0	1.3	0.1
590 kg	No-control	-3.4	-9.5	1.2	8.6
Condition 5	Neuro	-2.1	-0.0	0.4	0.0
0.4 G	P + D	-6.4	-0.4	0.6	0.0
590 kg	No-control	-0.7	-1.8	0.5	3.9
Condition 6	Neuro	-0.5	0.0	0.2	0.0
0.2 G	P + D	-1.5	-0.1	0.3	0.0
590 kg	No-control	-0.2	-0.4	0.2	1.9

Condition	Control system	Starboard actuator (mm)		Port actuator (mm)		Cost
		Overshoot*	Steady state	Overshoot*	Steady state	
Condition 1	Neuro	1.8	46.1	8.1	67.5	1.8
0.8 G	P + D	14.2	47.1	0.8	67.1	4.5
665 kg	No-control	----------	----------	----------	----------	110
Condition 2	Neuro	0.2	25.2	1.2	30.0	0.32
0.4 G	P + D	1.2	25.4	No-over**	29.8	0.43
665 kg	No-control	----------	----------	----------	----------	22
Condition 3	Neuro	0.1	13.2	0.3	14.3	0.075
0.2 G	P + D	No-over**	13.2	No-over**	14.2	0.083
665 kg	No-control	----------	----------	----------	----------	5.3
Condition 4	Neuro	5.1	39.9	7.9	61.7	1.6
0.8 G	P + D	14.3	40.0	No-over**	61.5	5.0
590 kg	No-control	----------	----------	----------	----------	123
Condition 5	Neuro	0.7	22.0	1.6	27.1	0.28
0.4 G	P + D	1.8	22.2	No-over**	26.9	0.40
590 kg	No-control	----------	----------	----------	----------	25
Condition 6	Neuro	0.3	11.6	0.5	12.9	0.066
0.2 G	P + D	0.2	11.6	No-over**	12.7	0.072
590 kg	No-control	----------	----------	----------	----------	6.0

*Overshoot : difference between transient response and steady state.

**No-over : no-overshoot, where transient response is smaller than steady state.



(a) Full system

(b) Optimisation process

Fig. 8.1 Diagrammatic representations of (a) full system and (b) optimisation process

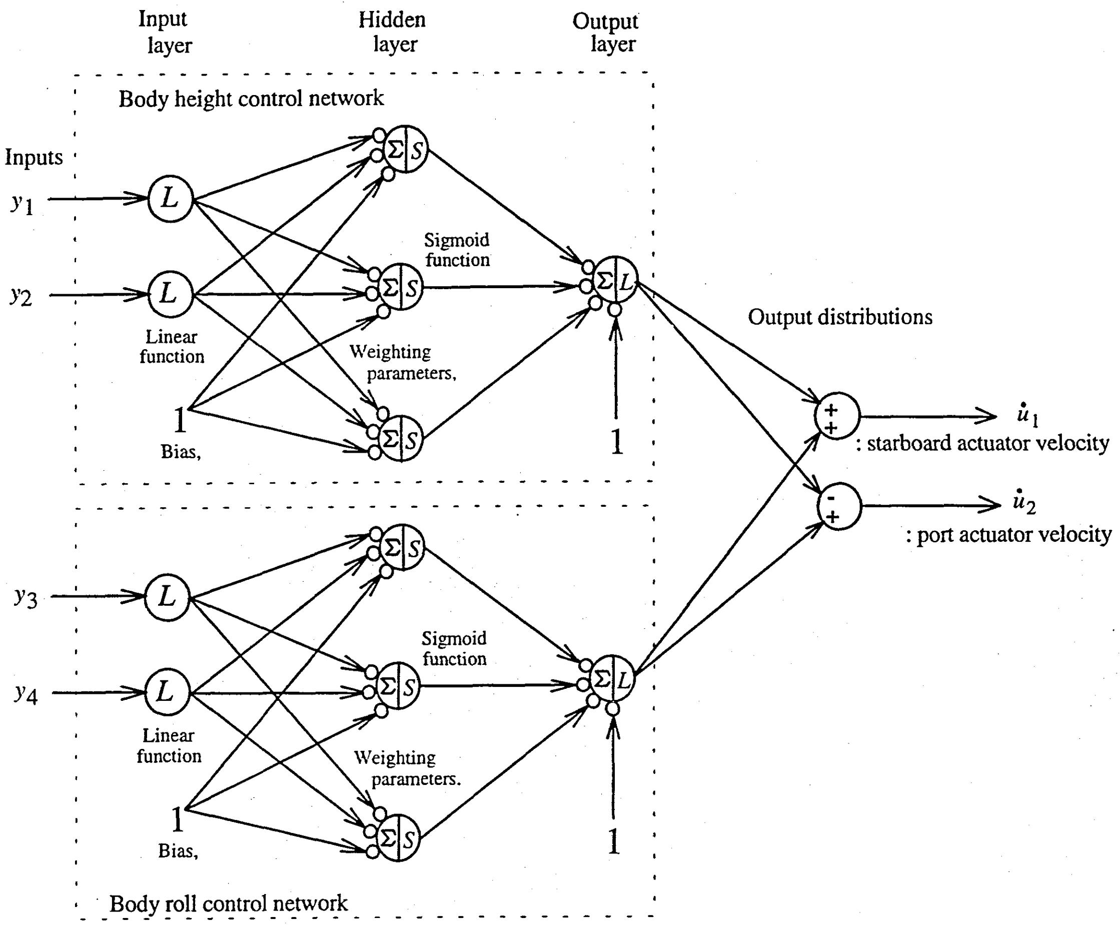


Fig. 8.2 Neuro-controller involving two networks for body height control and roll control

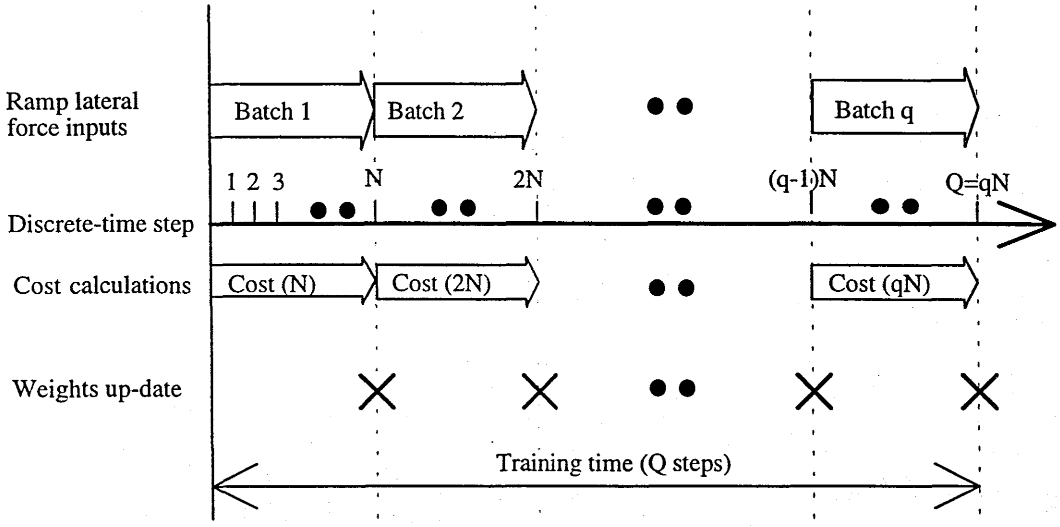


Fig. 8.3 Sequential representation of batch learning system in discrete-time

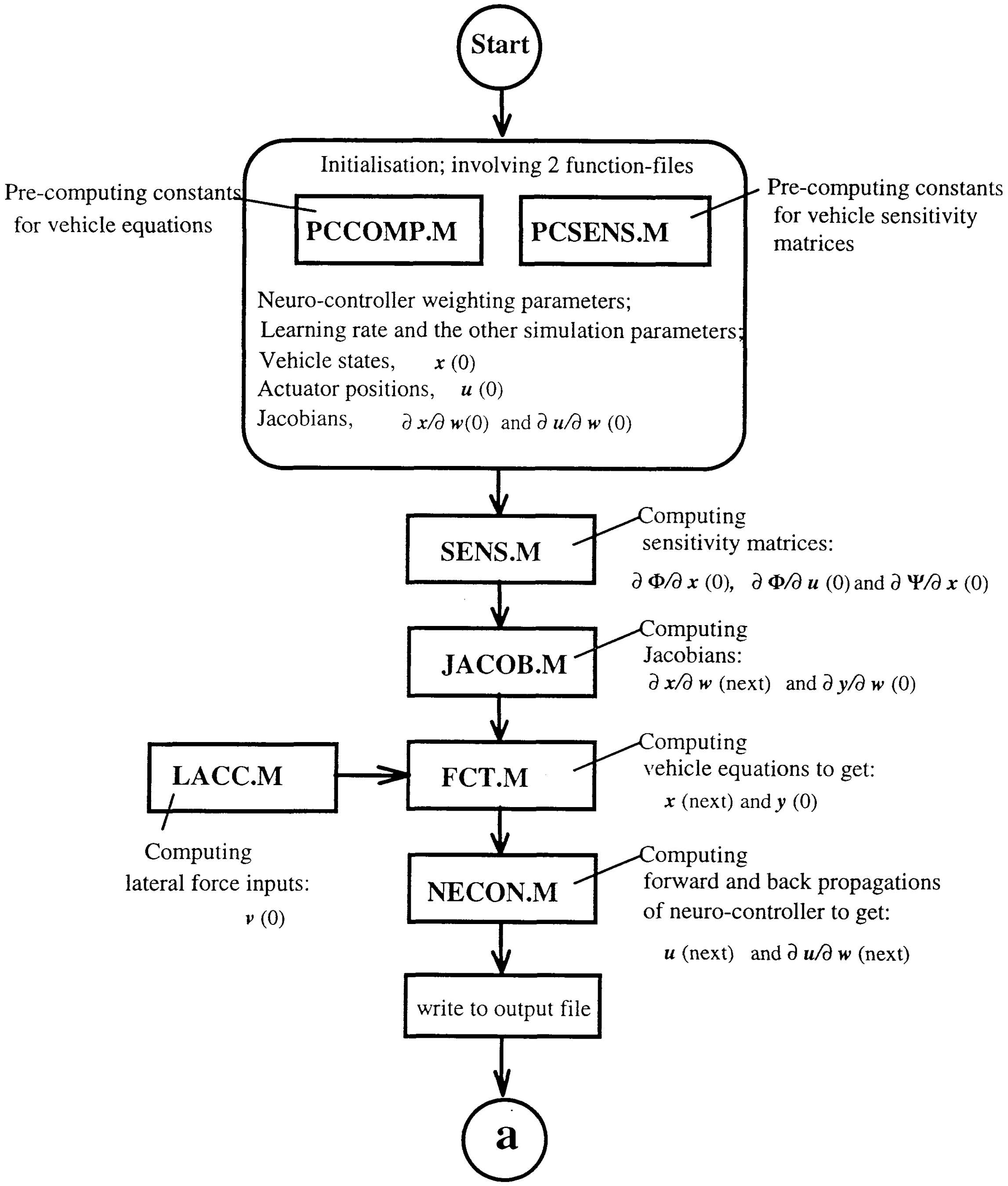

Start
Initialisation; involving 2 function-files
Pre-computing constants for vehicle equations
PCCOMP.M
PCSENS.M
Pre-computing constants for vehicle sensitivity matrices
Neuro-controller weighting parameters;
Learning rate and the other simulation parameters;
Vehicle states, x (0)
Actuator positions, u (0)
Jacobians, $\partial x/\partial w$(0) and $\partial u/\partial w$ (0)
SENS.M
Computing sensitivity matrices:
$\partial \Phi/\partial x$ (0), $\partial \Phi/\partial u$ (0) and $\partial \Psi/\partial x$ (0)
JACOB.M
Computing Jacobians:
$\partial x/\partial w$ (next) and $\partial y/\partial w$ (0)
LACC.M
Computing lateral force inputs:
v (0)
FCT.M
Computing vehicle equations to get:
x (next) and y (0)
NECON.M
Computing forward and back propagations of neuro-controller to get:
u (next) and $\partial u/\partial w$ (next)
write to output file
a

Fig. 8.4 Flowchart of the simulation programme
(This continues on the next page)

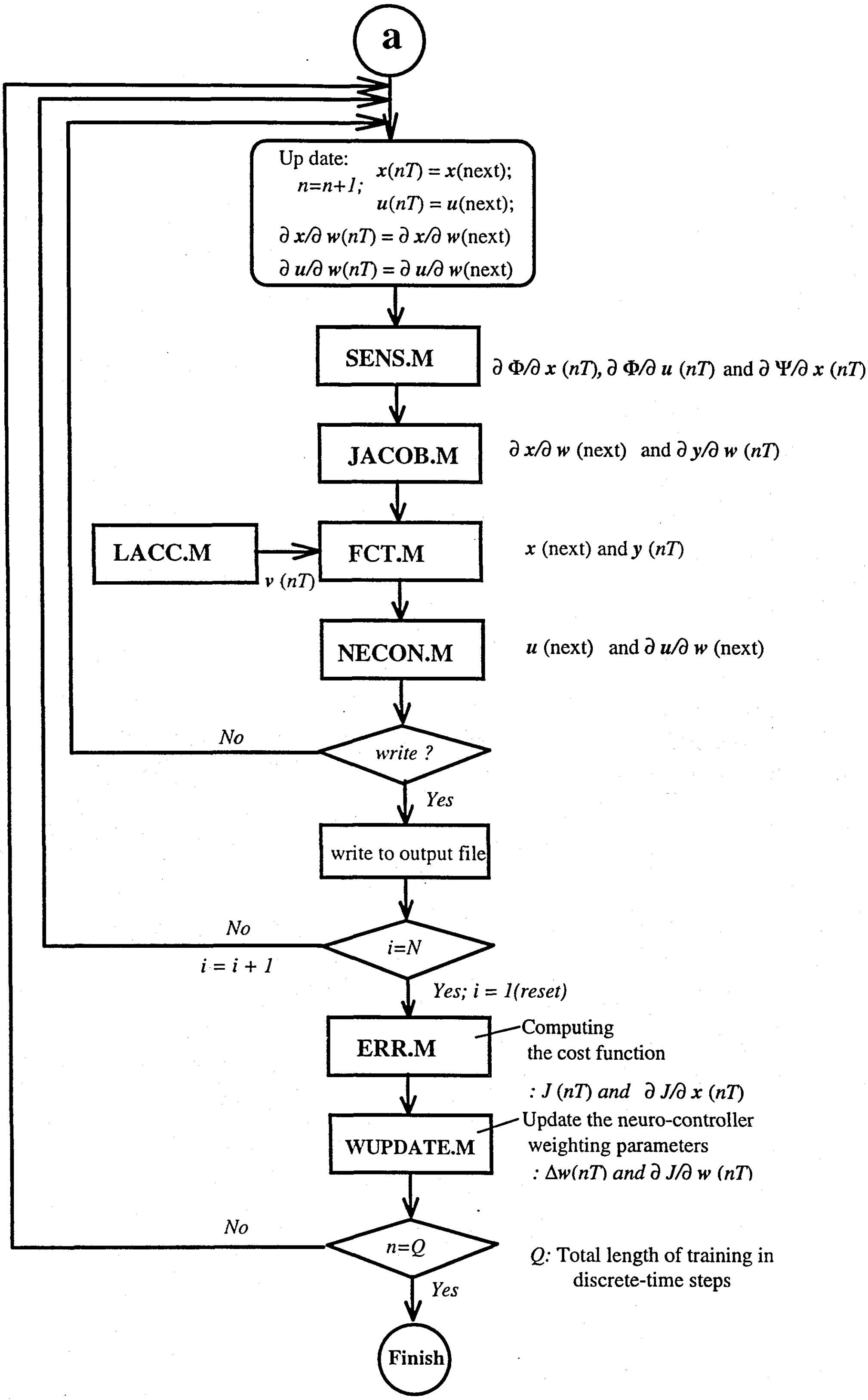

a
Up date:
n=n+1;
x(nT) = x(next);
u(nT) = u(next);
∂ x/∂ w(nT) = ∂ x/∂ w(next)
∂ u/∂ w(nT) = ∂ u/∂ w(next)
SENS.M
∂ Φ/∂ x (nT), ∂ Φ/∂ u (nT) and ∂ Ψ/∂ x (nT)
JACOB.M
∂ x/∂ w (next) and ∂ y/∂ w (nT)
LACC.M
v (nT)
FCT.M
x (next) and y (nT)
NECON.M
u (next) and ∂ u/∂ w (next)
No
write ?
Yes
write to output file
No
i = i + 1
i=N
Yes; i = 1(reset)
ERR.M
Computing
the cost function
: J (nT) and ∂ J/∂ x (nT)
WUPDATE.M
Update the neuro-controller
weighting parameters
: Δw(nT) and ∂ J/∂ w (nT)
No
n=Q
Yes
Q: Total length of training in
discrete-time steps
Finish

Fig. 8.4 Flowchart of the simulation programme

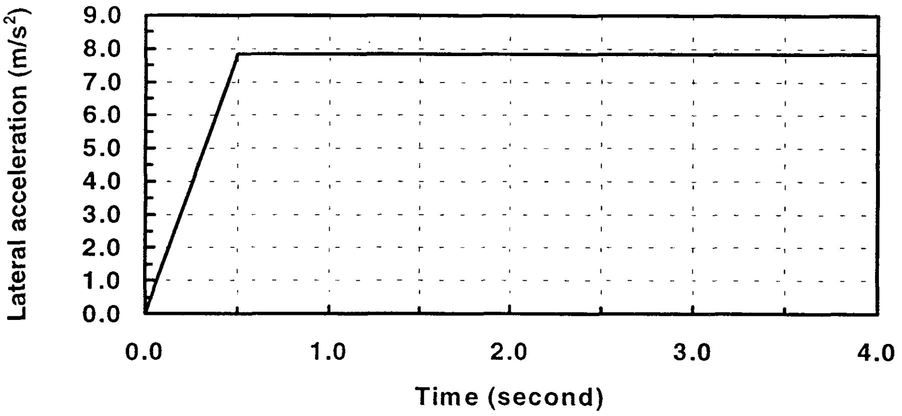
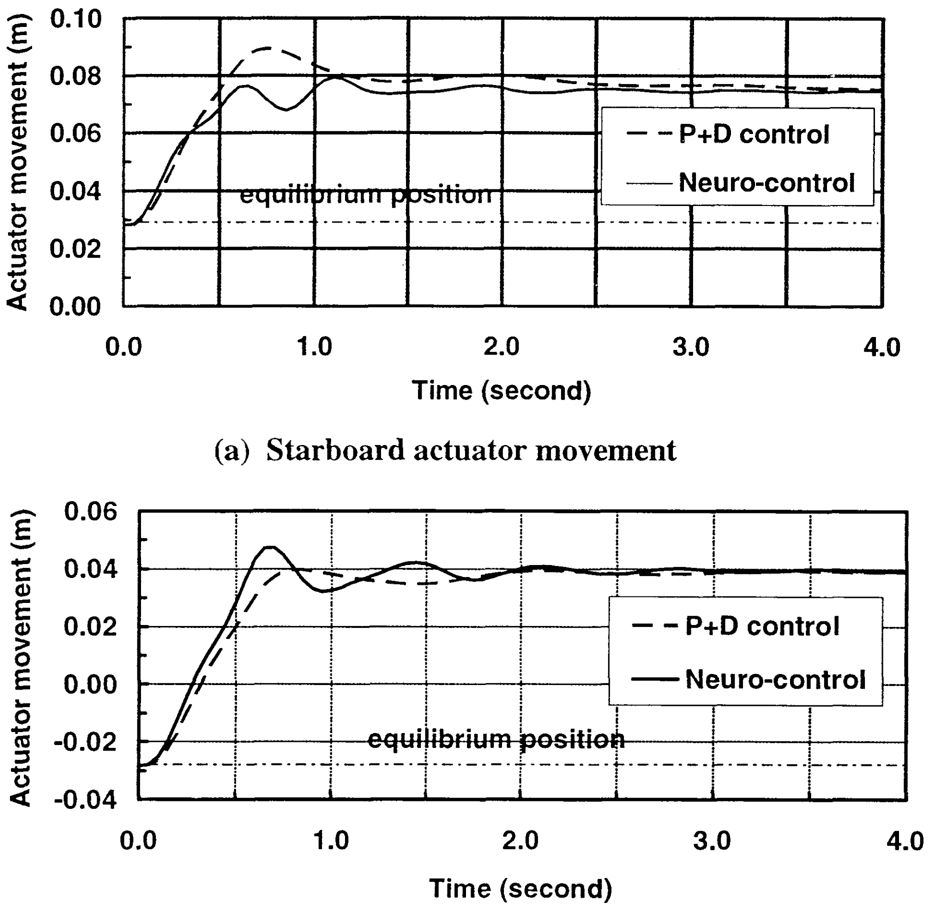


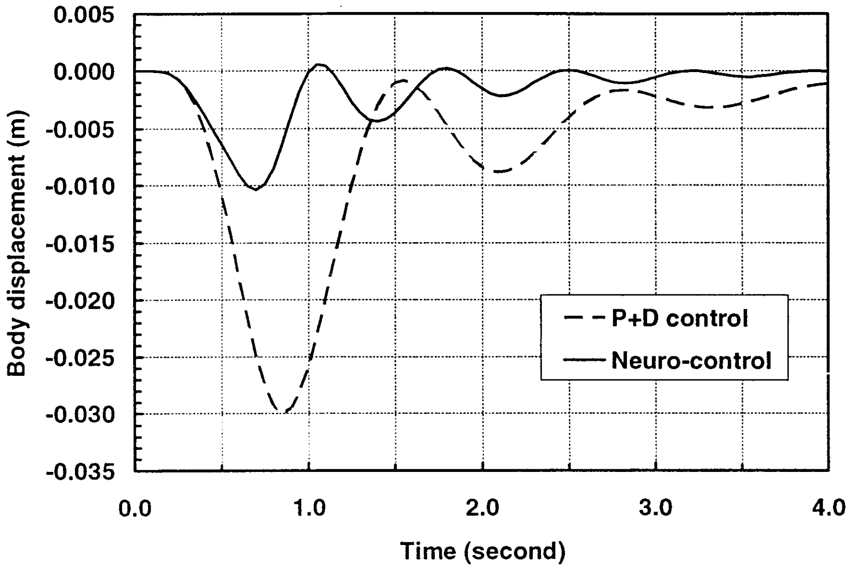
Fig. 8.5 Lateral acceleration corresponding to the force input



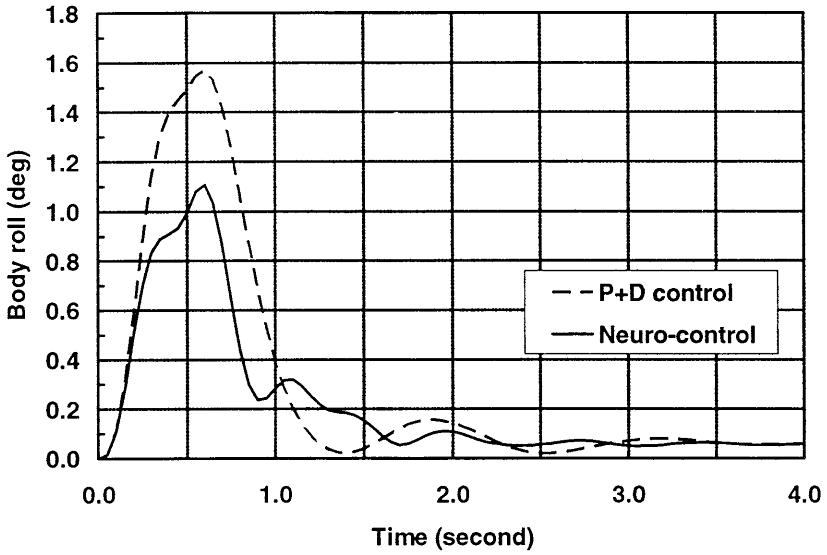
(a) Starboard actuator movement

(b) Port actuator movement

Fig. 8.6 Comparisons between actuator responses with neuro-control and P + D control on the starboard and port sides during 0.8 G cornering (vehicle body mass: 665 kg)



(a) Vehicle body height change response



(b) Vehicle body roll response

Fig. 8.7 Comparisons between the neuro-control and P + D control in vehicle body responses: (a) height change and (b) body roll; during 0.8 G cornering (vehicle body mass : 665 kg)

Chapter 9

Energy Consumption of Variable Geometry Active Suspension System

This chapter considers the modelling and simulation of an actuator system, which is expected to be installed at each wheel station of the variable geometry active suspension system; using the computer simulation package, SIMULINK.

The first section of the chapter presents general information on the model building and simulation via SIMULINK. In the second section, an actuator model is built in SIMULINK's block diagram window, and it is described through the four parts: motor speed control, motor current control, electric motor circuit and mechanical system. The third section provides simulation results showing power and energy consumption of the variable geometry system, with either the P + D control or the neuro-control system. These are compared with those of other actively-controlled suspension systems taken from the literature, under similar circumstances, i.e. that involving 0.8 G cornering.

9.1 SIMULINK

SIMULINK is a computer package for simulating dynamic systems as an extension to MATLAB. SIMULINK has two phases of use, model building and simulation.

SIMULINK facilitates the model building via block diagram windows, in which models are created and edited principally by mouse driven commands. Building a system is much like drawing a block diagram. Instead of drawing the individual blocks, they are copied from either block libraries supplied with SIMULINK or block libraries which we define ourselves.

The feedback control scheme of the variable geometry active suspension half-car model, which consisted of the P + D control system in chapter 7 or the neuro-control system in chapter 8, can be converted to SIMULINK's block diagram window, as shown in Fig. 9.1. This model involves three MATLAB function-files called: 'CONTROLLER.M', 'PCCOMP.M' and 'VEHICLE.M'. 'CONTROLLER.M' contains either the P + D control, which was represented by equation (7.14), or the neuro-controller with optimal weighting parameters, whose function form was represented by equation (8.3). 'PCCOMP.M' was already described in section 8.2 in the previous chapter. 'VEHICLE.M' involves differential equations of the suspension model. The differential equations were converted to a MATLAB function-file from the corresponding subroutine of the FORTRAN programme, which was automatically generated via AUTOSIM.

In simulations, such differential equations are integrated; SIMULINK provides a number of integration methods, e.g. Runge-Kutta third and fifth-order methods, Euler's method, etc. In practice, a simulation result, provided via SIMULINK's variable geometry suspension with a controller model using the Runge-Kutta third-order method, agrees with that of the corresponding FORTRAN programme using the Runge-Kutta second-order method. Therefore, it is not necessary to show these results in figures.

9.2 Actuator Model

SIMULINK's block diagram window showing the actuator model for a single wheel station is shown in Fig. 9.2. (There will be four actuator models in order to represent a full car system having four wheels.) The actuator model can be described through the following four parts: motor speed control, motor current control, electric motor circuit and mechanical system.

Motor speed control

An actuator velocity demand is used, through the inverted leadscrew gear ratio (this will be described later on) and a D.C. speed transformer, to provide the motor speed demand signal. The actual motor speed feedback signal is provided through a D.C. tachogenerator, and the feedback signal is compared with the demand signal in the speed-error amplifier. The resulting speed-error signal is amplified by the gain, K_S, and the output determines the motor current demand signal.

Hence, if the actual motor speed is less than the desired speed, the speed-error amplifier will demand current in proportion to the speed error; the motor will therefore accelerate in an attempt to minimise the speed error. As the motor speed comes up to the desired speed, the speed error reduces; and the final speed is approached smoothly.

Motor current control

A current feedback signal is obtained via a D.C. current transformer, and it is compared with the motor current demand signal in the current-error amplifier. The resulting current-error signal is amplified by the gain, K_P, and the output determines the motor driving voltage, V.

Electric motor circuit

The motor driving voltage, V, is combined with the motor back-e.m.f., E, which is generated in proportion to the motor speed, $\dot{\theta}$, to oppose the driving voltage. The back-e.m.f. is represented by:

$$E = K_e \dot{\theta}, \tag{9.1}$$

where K_e is the motor back-e.m.f. constant in Vsec/rad. The combined voltage governs the motor current, I, through the motor resistance, R. The voltage equation is:

$$I = \frac{V - E}{R} \quad \text{or} \quad V = E + IR. \tag{9.2}$$

Multiplying equation (9.2) by the current gives the power equation as:

$$\begin{aligned} &\text{Electrical Input Power } (VI) \\ &\qquad = \text{Mechanical Output Power } (EI) + \text{Power Loss } (I^2R). \end{aligned} \tag{9.3}$$

I will use the term 'power' to refer to the electrical input power, VI; which is the sum of the mechanical output power and the power loss as described in equation (9.3). The energy consumption can be represented by the integration of the power:

$$W = \int VI \, dt. \tag{9.4}$$

The electromagnetic torque, T_M, is generated in proportion to the motor current, I:

$$T_M = K_T I\,, \tag{9.5}$$

where K_T is the electromagnetic torque constant in Nm/A. The motor responds to the electromagnetic torque and the load torque, which opposes the electromagnetic torque, by accelerating the motor armature of inertia, j:

$$j\ddot{\theta} = T_M - T_L\,. \tag{9.6}$$

The load torque, T_L, is related to the mechanical load as described below.

Mechanical system

The mechanical arrangement of an actuator for a single wheel station is shown diagrammatically in Fig. 9.3. The actuator velocity, $\dot{y}_{act}$, is related to the motor speed, $\dot{\theta}$, through a leadscrew gear ratio:

$$\dot{y}_{act} = \dot{\theta}\, r \tan\alpha\,, \tag{9.7}$$

where r is the radius of the screw and α is the helix angle.
When the slider moves on the track, a friction force occurs to oppose the action. The friction force is:

$$F_{fric} = \mu\, F_{strut} \cos\varphi\, \tanh(\gamma\, \dot{y}_{act}), \tag{9.8}$$

where μ is the friction coefficient between the surfaces of the slider and the track; F_{strut} is the strut force; φ is the angle between the direction of the strut force and the line, which is at a right angle to the track; and γ is the shaping parameter. This friction function is illustrated in Fig. 9.4, where $\mu = 0.06$, $\varphi = 0$ and $\gamma = 1000$.

The mechanical load is represented by:

$$\text{Load} = \mu\, F_{strut} \cos\varphi\, \tanh(\gamma\, \dot{y}_{act}) + F_{strut} \sin\varphi\,, \tag{9.9}$$

where the first term on the right side of the equation always opposes the actuation. On the other hand, the second term (which is a component of the strut force acting parallel to the track) opposes the actuation when the actuation is towards the wheel, but they work together when the actuation is towards the vehicle body.

Nevertheless, a simplification can be made by the neglect of the angle, φ, for convenience. Therefore, the simplified load is:

$$\text{Load} = \mu\, F_{strut}\, \tanh(\gamma\, \dot{y}_{act})\,. \qquad (9.10)$$

The equation (9.10) expresses that multiplying the friction function, $\mu \tanh(\gamma\, \dot{y}_{act})$, by the strut force gives the load, i.e. the friction force. The equation (9.10) is involved in the SIMULINK model.

When the leadscrew is loaded, the work for one rotation of the load torque is:

$$2\pi\; T_L = 2\pi\; r \tan\alpha \times \text{Load} + 2\pi \times \text{Friction Torque}\,. \qquad (9.11)$$

The friction torque is:

$$\begin{aligned} \text{Friction Torque} &= r \times \text{Friction Coefficient of Leadscrew} \times \text{Load} \\ &= r \tan\beta \times \text{Load}, \end{aligned} \qquad (9.12)$$

where β is the friction angle of the leadscrew.

Substitution of (9.12) into the last term of (9.11) then yields:

$$\begin{aligned} 2\pi\; T_L &= 2\pi\; r \tan\alpha \times \text{Load} + 2\pi\; r \tan\beta \times \text{Load} \\ \therefore\; T_L &= r \tan(\alpha + \beta\,) \times \text{Load}. \end{aligned} \qquad (9.13)$$

Hence the load torque, T_L, is related to the mechanical load through the Torque/Load ratio, r tan(α +β); strictly speaking, the leadscrew is irreversible, if the helix angle and friction angle have the relationship, α ≤ β.

In SIMULINK's actuator model, the following representative model parameters have been used:

D.C. current feedback transformer : K_{crf} = 0.2 V/A ;

D.C. tachogenerator (D.C. motor speed transformer) : K_{tch} = 0.05 Vs/rad ;

Speed control gain : K_S = 20 V/V ;

Power amplifier gain: K_P = 20 V/V;

Motor back-e.m.f. constant :	$K_e = 0.09$ Vsec/rad ;
Electromagnetic torque constant :	$K_T = 0.09$ Nm/A ;
Motor inertia :	$j = 0.001$ kgm^2 ;
Motor resistance :	$R = 1\ \Omega$;
Mean radius of leadscrew :	$r = 0.005$ m ;
Leadscrew helix angle :	$\alpha = 0.087266$ rad (= 5 deg) ;
Leadscrew friction angle :	$\beta = 0.087266$ rad (= 5 deg) ;

Friction coefficient between the slider and the track : $\mu = 0.06$.

9.3 Simulation Results

Firstly, SIMULINK's half-car model, involving the variable geometry active suspension with control system, is simulated. (see Fig. 9.1) In this model, the actuator demand velocities govern the actual actuator velocity responses, i.e. the actuators respond, perfectly, to the demand. When the simulation is completed, the time histories of the actuator demand velocities and suspension strut forces are stored in the output data called 'data 1' and 'data 2', respectively. Each data consists of two column-vectors for the starboard and port sides.

Secondly, SIMULINK's actuator model is used for each side actuators via data 1 and data 2. (Here, one column-vector of each data is used.) The time-histories of the actuator velocity, the motor current signal and its demand, the power and energy consumption are stored as column-vectors for each simulation.

Fig. 9.5 shows the demand velocity and actual velocity response of the starboard actuator with the neuro-control system under ramp lateral force inputs, which involve maximum lateral acceleration of 7.848 m/s^2 (0.8 G); the half-car is loaded with the vehicle body mass of 665 kg. As shown in this figure, the actuator velocity response accords with the demand velocity.

Fig. 9.6 shows the motor current demand and the current feedback signals; Fig. 9.7 (a) and (b) show the power and energy consumption, respectively, in the similar manner.

Table 9.1 shows the power and energy consumption of the starboard and port actuators with each of the P + D control and neuro-control systems. Each system responds in a similar manner. From this table, the neuro-controlled system consumes more energy than the P + D controlled system on each side of the vehicle. The possible reason is that the neuro-controller seems to require faster actuator responses than the P + D controller, see Fig. 8.6 in the previous chapter. Thus, we can compromise the performance and energy consumption using the cost function (8.6), when we train the neuro-controller.

Table 9.2 shows a comparison between the r.m.s. value of the power with each of three hydraulic active suspension systems, involving direct actuations between bodies and wheels, and that of the variable geometry active suspension with either the P + D control or the neuro-control system. Each system relates to a full-car, having four wheel stations with four actuators; under a 0.8 G cornering. Data of the hydraulic active suspension systems are taken from the literature (Williams and Miller, 1994). From this table, the hydraulic high-bandwidth system requires the lowest power for cornering. However, it requires the highest power among the three hydraulic systems on a straight road as described by Williams and Miller. On the other hand, if the variable-geometry systems do not activate on a straight road, they are inherently passive suspensions. Overall, the variable-geometry systems require low power as compared with the hydraulic systems.

Table 9.3 shows a comparison between the energy consumption of a hydraulic active roll control system, which involves rotary actuators in the anti-roll bars at front and rear, and that of the variable geometry active suspension with either P + D control or neuro-control system. Each system relates to a full-car, having four wheel stations; under a 0.8 G cornering. Data of the active roll control system is taken from the literature (Sharp and Pan, 1993). From this table, the variable-geometry suspension with the P + D control system achieves lower energy consumption than the active roll control system, while the neuro-controlled system consumes higher energy than it. Hence, we can compromise the performance and energy consumption, as mentioned earlier.

Table 9.4 shows the power and energy consumption of the variable-geometry suspensions with neuro-control, involving each of three different friction levels between the sliders and the tracks. Each system relates to a full-car, having four wheel stations with four actuators; under a 0.8 G cornering. From this table, the system, involving the larger friction of the mechanical system, requires higher power and energy consumption. Therefore, when we design the mechanical arrangement of the actuators, it is necessary to consider the materials of the sliders and tracks.

9.4 Conclusions

It was shown how the actuator model can be built via the computer simulation package, SIMULINK.

It turned out that the neuro-control system, which demonstrated better performance than the P + D control system in the previous chapter, required higher power and energy consumption for a 0.8 G cornering. Therefore, the controller design may involve a compromise between the performance and energy cost.

Nevertheless, from the simulation results presented, the variable geometry active suspension systems require low power and energy consumption as compared with the other actively controlled suspension systems taken from the literature.

Table 9.1 Power and energy consumption of the starboard and port actuators of the variable-geometry active suspension half-car model with each of the P + D control and neuro-control systems. Each system responds to 0.8 G cornering; the half-car is loaded with the vehicle body mass of 665 kg

	Starboard actuator			Port actuator		
Control system	R.m.s. power (W)	Peak power (W)	Energy * consum. (J)	R.m.s. power (W)	Peak power (W)	Energy * consum. (J)
Neuro	293	1654	431	341	1796	473
P + D	191	1239	221	186	1210	182

* Energy consumption is calculated for 4 second.

Table 9.2 Comparison between the r.m.s. value of the power with each of three hydraulic active suspension systems, involving direct actuations between bodies and wheels, and that of the variable geometry active suspension with either the P + D control or neuro-control system. Each system relates to a full-car, having four wheel stations with four actuators; under 0.8 G cornering. Data of the hydraulic active suspension systems are taken from the literature (Williams and Miller, 1994)

	R.m.s. power (W)
Hydraulic high-bandwidth system	190
Hydraulic low-bandwidth system 1	3829
Hydraulic low-bandwidth system 2	3743
Variable geometry system (neuro)	1268
Variable geometry system (P + D)	754

Table 9.3 Comparison between the energy consumption of a hydraulic active roll control system, which involves rotary actuators in the anti-roll bars at front and rear, and that of the variable geometry active suspension with either P + D control or neuro-control system. Each system relates to a full-car, having four wheel stations; under 0.8 G cornering. Data of the active roll control system is taken from the literature (Sharp and Pan, 1993)

	Energy consumption (J)
Active roll control system	1194
Variable geometry system (neuro)	1808
Variable geometry system (P + D)	806

Table 9.4 Power and energy consumption of the neuro-controlled variable-geometry suspension systems, involving each of three different friction levels between the sliders and the tracks. Each system relates to a full-car, having four wheel stations with four actuators; under 0.8 G cornering

	R.m.s. power (W)	Energy consum. (J)
No-friction	1062	1368
Friction* : μ = 0.06, friction angle** : 5°	1268	1808
Friction* : μ = 0.1, friction angle** : 5°	1442	2274

* Friction between the sliders and the tracks,

** Friction angle of the leadscrews.

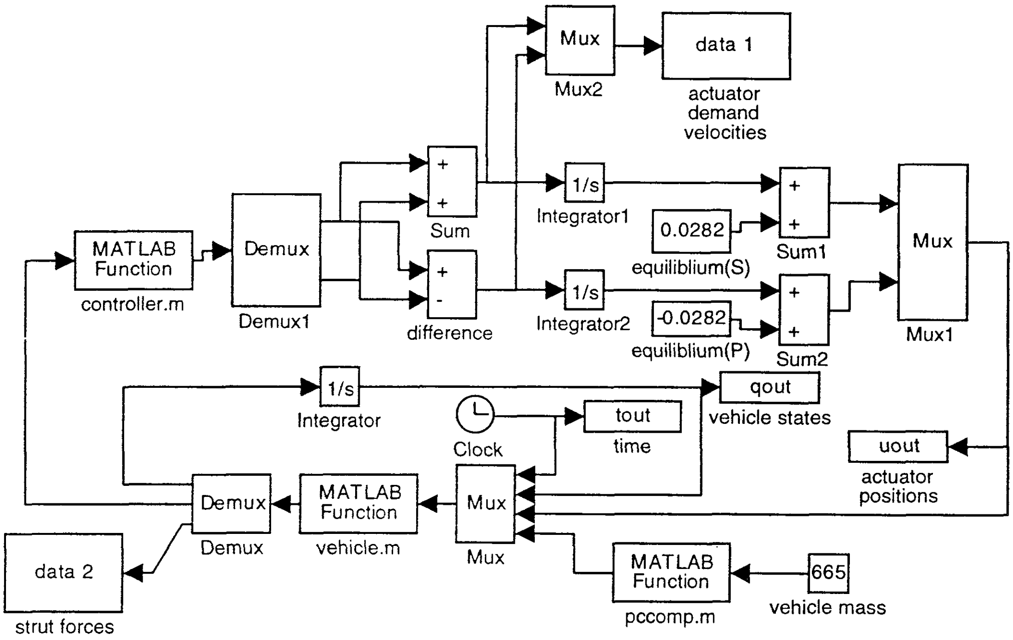


Fig. 9.1 Feedback control scheme of variable geometry active suspension half-car model on SIMULINK's block diagram window

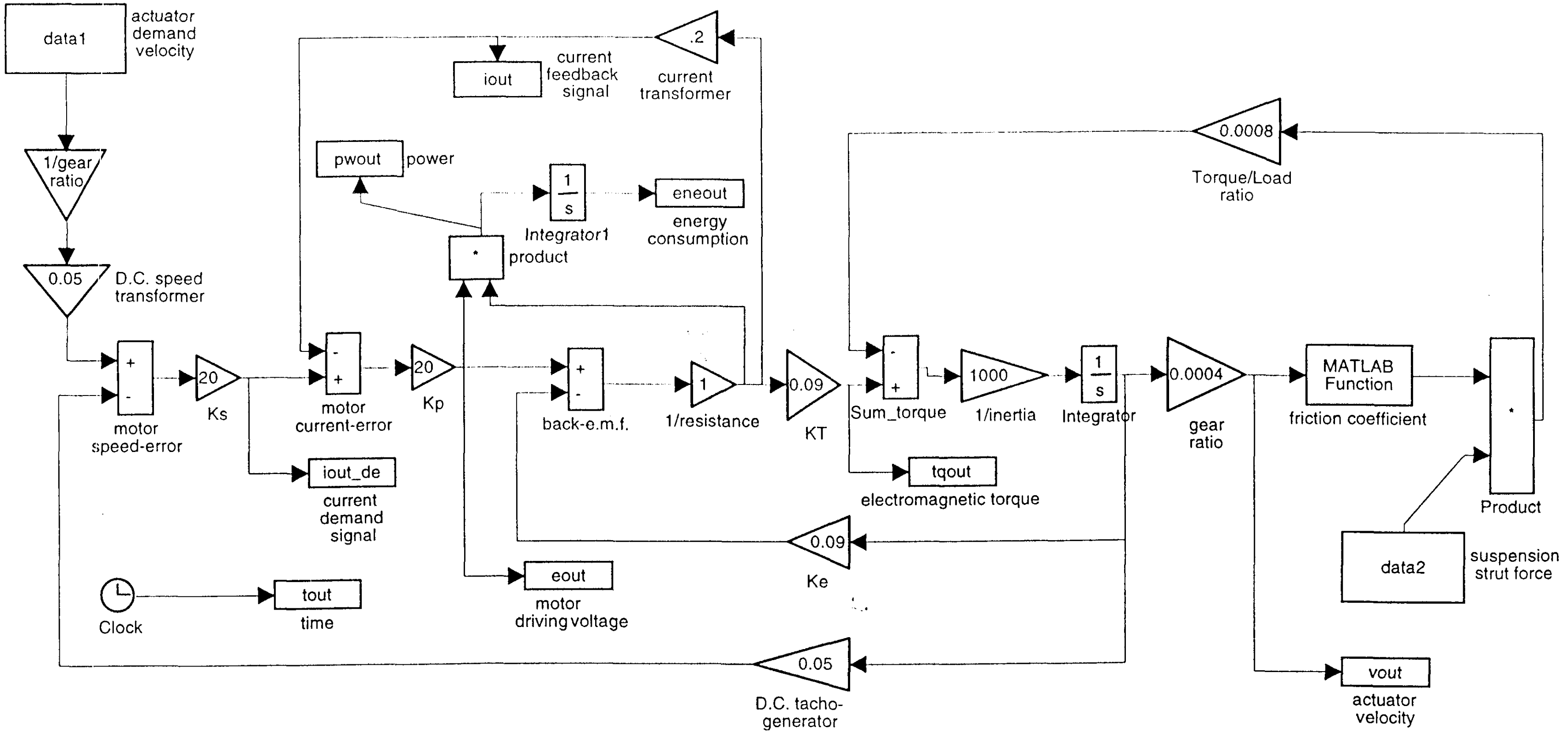


Fig. 9.2 Actuator model, for a single wheel station, on SIMULINK's block diagram window

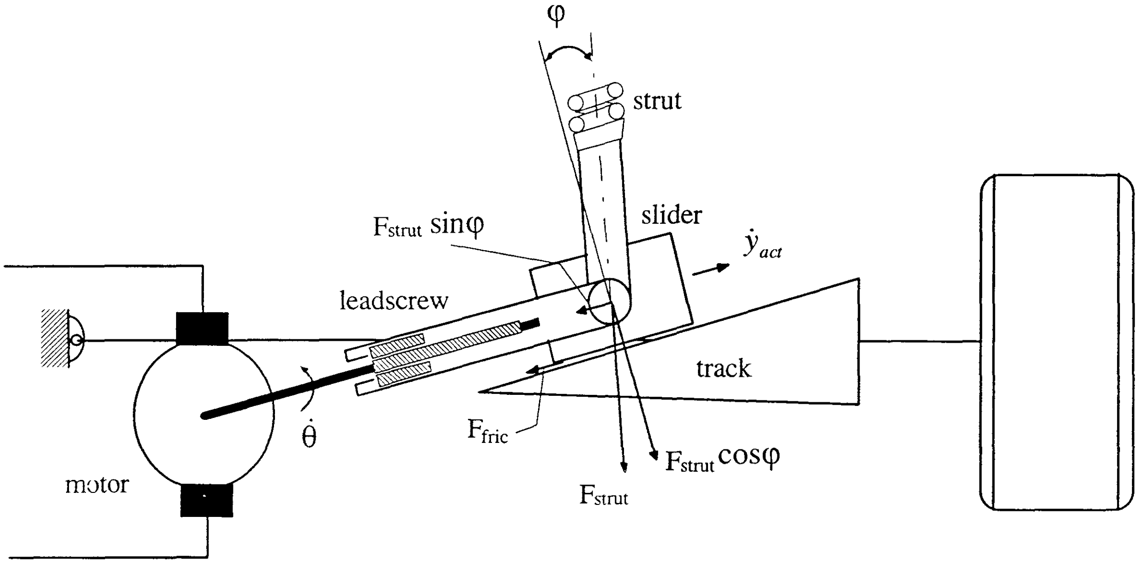


Fig. 9.3 Diagrammatic representation of mechanical arrangement of an actuator for a single wheel station

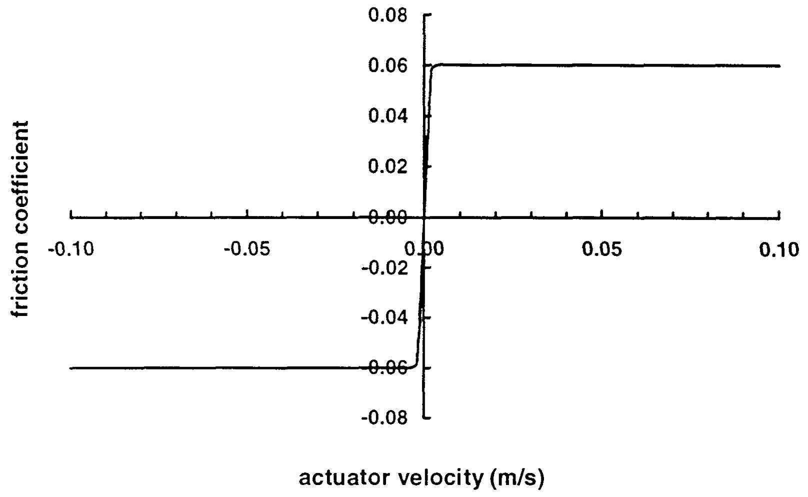


Fig 9.4 Friction function

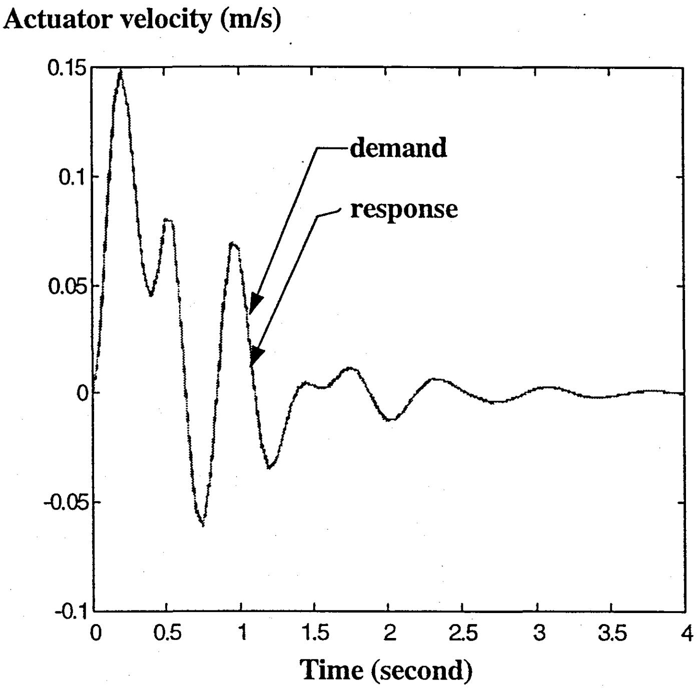


Fig. 9.5 Comparison between the demand velocity and actual velocity response of the starboard actuator with the neuro-control system under ramp lateral force inputs, which involve maximum lateral acceleration of 7.848 m/s^2 (0.8 G); the half-car is loaded with the vehicle body mass of 665 kg

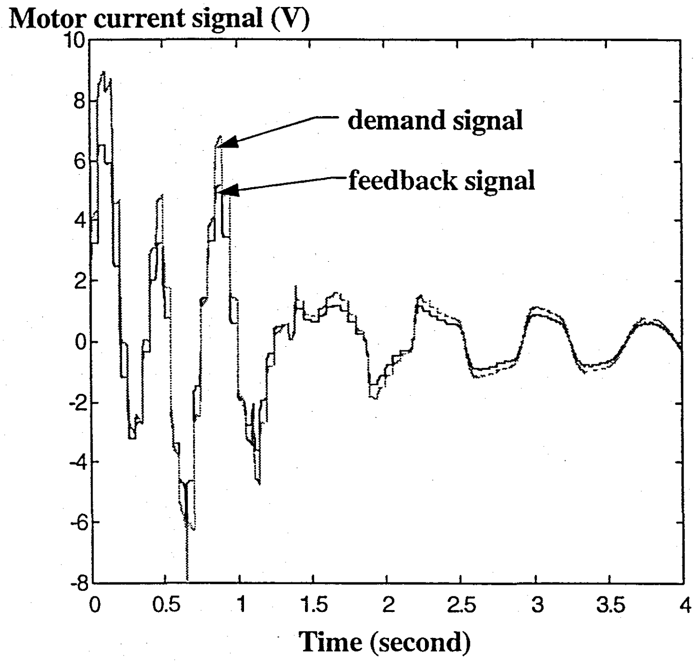
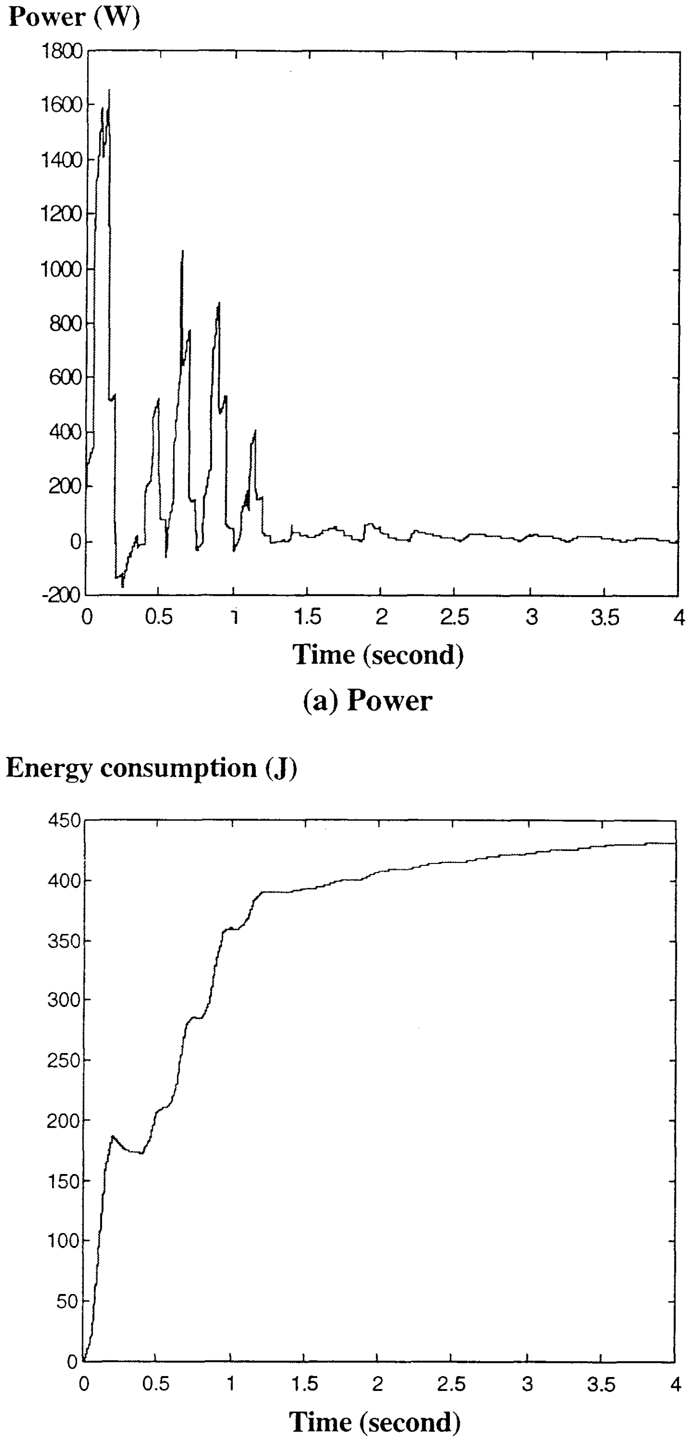


Fig. 9.6 Comparison between the motor current demand and current feedback signals of the starboard actuator with the neuro-control system under ramp lateral force inputs, which involve maximum lateral acceleration of 7.848 m/s^2 (0.8 G); the half-car is loaded with the vehicle body mass of 665 kg



(a) Power

(b) Energy consumption

Fig. 9.7 Power and energy consumption of the starboard actuator with the neuro-control system under ramp lateral force inputs, which involve maximum lateral acceleration of 7.848 m/s^2 (0.8 G); the half-car is loaded with the vehicle body mass of 665 kg

Chapter 10
Conclusions

Energy consumption problems in automotive active suspension systems have resulted in a general lack of commercial enthusiasm for them. One way to tackle the high energy consumption is to employ low energy system types, which involve variable geometry. However, the control system design problems are novel, since variable geometry implies non-linearity.

Multi-layer neural networks can learn to control non-linear systems to reduce cost functions of general form under external inputs, e.g. road disturbance or lateral forces. One contribution of this work is to demonstrate how a neural network can be integrated with a dynamic system physical model to minimise a defined cost function relating to the controlled system.

The spring/damper unit length change to the wheel displacement ratio can be varied by actuation, which is substantially perpendicular to the suspension force. In the variable geometry design, one end of each spring/damper unit moves on an inclined track of each lower swing arm of the suspension. The inclined track was designed to ensure that both preload of the spring/damper unit and leverage ratio change work together in the same direction resulting from actuator movement. The developed half-car model with the variable geometry demonstrated that the nominal vehicle body height was maintained by symmetrical actuation on the starboard and port sides for the various vehicle body weight conditions. The neuro-controller successfully reduced the cost function (consisting of the weighted sum of the squares of the body height change, body roll and actuator velocities) and it improved the responses of the body height change and body roll under all conditions tried. These involved various levels of ramp lateral force inputs and vehicle body weight changes.

It turned out via the simulations using the developed actuator model that a variable geometry active suspension system is economical in energy consumption as compared with other actively controlled suspension systems taken from the literature. Since the variable geometry system without actuation is a conventional passive suspension, there appear to be possibilities to work as a passive vibration isolator on a straight road and to obtain the benefits of active suspensions - levelling function, body-attitude-control and preview control for special occasions, i.e. bump and pot-hole. The neuro-controlled system demonstrated better performance but consumed higher energy than the P + D controlled system. Therefore, a practical controller design will need to involve compromise between performance and energy cost.

As a continuation of the work, the following activities are suggested:

a. A full-car model, having four-wheel stations with variable geometry, could be developed in the computer;

b. The full-car model with a neuro-controller could be tested under various levels of lateral force inputs for roll control performance or longitudinal force inputs for pitch control performance;

c. The system could be tested as a vibration isolator with cylindrical or cross level road disturbance input;

d. Preview techniques could be applied to rear suspension control using front suspension sensors in order to improve on vibration isolation and to cope with bumps and pot-holes;

e. To improve on learning efficiency, there remains a further investigation of the neural network learning algorithm, involving the following factors: network structure, the number of layers, the number of processing elements, transfer function and update rule of weighting parameters;

f. Detailed design for particular applications from economical cars to special purpose cars, e.g. military vehicles and off-road vehicles, could be considered. Designs could be derived from full studies of mechatronic systems and learning control.

REFERENCES

Darling, J. and Ross-Martin, T. J., 1997, A theoretical investigation of a prototype active roll control system. *Proceedings of the Institution of Mechanical Engineers, Journal of Automobile Engineering (Part D),* **211,** pp. 3-12.

Demuth, H. and Beale, M., 1994, *Neural Network Toolbox User's Guide* (The Math Works Inc., South Natick, MA).

El-Gindy, M. and Palkovics, L., 1993, Possible application of artificial neural networks to vehicle dynamics and control: a literature review. *International Journal of Vehicle Design,* **14**(5-6), pp. 592-614.

Elbeheiry, E. M., Karnopp, D. C., Elaraby, H. E and Abdelraaouf, A. M, 1995, Advanced ground vehicle suspension systems- a classified bibliography. *Vehicle System Dynamics,* **24,** pp. 231-258.

Feldkamp, L. A., Puskorius, G. V., Davis, Jr. L. I. and Yuan, F., 1992, Neural control systems trained by dynamic gradient methods for automotive applications. *IJCNN International Joint Conference on Neural Networks*, **2,** 7 th - 11 th June 1992, Baltimore, MD, pp. 798-804.

Franklin, G. F. and Powell J. D., 1980, *Digital Control of Dynamic Systems* (Addison-Wesley, Reading, MA).

Gordon, T. J., Marsh, C. and Milsted, M. G., 1990, Control law design for active and semi-active automobile suspension systems. *VDI International Congress: Numerical Analysis in Automotive Engineering* (VDI Berichte) October 1990, Wurzburg, **816,** pp. 537-546.

Gordon, T. J., Marsh, C. and Milsted, M. G., 1991, A comparison of adaptive LQG and non-linear controllers for vehicle suspension systems. *Vehicle System Dynamics,* **20,** pp. 321-340.

Gordon, T. J., Marsh, C. and Wu, Q. H., 1993, Stochastic optimal control of active vehicle suspensions using learning automata. *Proceedings of the Institution of Mechanical Engineers, Journal of Systems and Control Engineering (Part I),* **207,** pp. 143-152.

Gordon, T. J., Palkovics, L., Pilbeam, C. and Sharp, R. S., 1994, Second generation approaches to active and semi-active suspension control system design. *Vehicle System Dynamics,* **23,** pp. 158-171.

Hagan, M. T., Demuth, H. B. and Beale, M., 1996, *Neural Network Design* (PWS Publishing, Boston, MA).

Harris, C. J. (Eds), 1994, *Advances in Intelligent Control* (Taylor & Francis, London).

Harris, C. J., Moore, C. G. and Brown, M., 1992, *Intelligent Control: Some Aspects of Fuzzy Logic and Neural Networks* (World Scientific Press, London and Singapore).

Hicks, R. W., 1992, *Classic Motorbikes* (Tiger Books International, London).

Hillebrecht, P., Konik, D., Pfeil, D. Wallentowitz, H. and Zieglmeier, F., 1992, The active suspension between customer benefit and technological competition. *Proceedings of FISITA'92, Total Vehicle Dynamics,* **2,** March 1992, London (Mechanical Engineering Publications, London) pp. 221-230.

Howell, M. N., Frost, G. P. and Gordon, T. J., 1996, The application of interconnected stochastic learning automata to controller design for semi-active automobile suspensions. *Proceedings of the International Symposium on Advanced Vehicle Control, AVEC '96* (JSAE) pp.105-117.

Kiguchi, K. and Fukuda, T., 1996, Hybrid control of robot manipulators for an unknown environment using fuzzy neural networks. *Proceedings of the CESA'96, IMACS Multiconference-Symposium on Robotics and Cybernetics,* July 1996, Lille (IEEE Systems, Man and Cybernetics) pp. 777-782.

Lang, R. and Walz, U., 1991, Active roll reduction, *EAEC 3 rd International Conference, Vehicle Dynamics and Powertrain Engineering,* 1991, Strasbourg, pp. 88-92.

Leighton, N. J. and Pullen, J., 1994, A novel active suspension system for automotive application. *Proceedings of the Institution of Mechanical Engineers, Journal of Automobile Engineering (Part D),* **208**(D4), pp. 243-250.

Leighton, N. J., 1995, Simulation and implementation of an energy efficient active suspension system. *Institution of Mechanical Engineers, AUTOTECH 95,* C498/25/009.

Lin, Y. - J., Lu, Y. - Q. and Padovan, J., 1993, Fuzzy logic control of vehicle suspension systems. *International Journal of Vehicle Design,* **14**(5-6), pp. 457-470.

Moran, A. and Nagai, M., 1992, Optimal preview control of rear suspension using nonlinear neural networks. *Proceedings of the International Symposium on Advanced Vehicle Control, AVEC '92* (JSAE) pp. 117-122.

Moscinski, J. and Ogonowski, Z., 1995, *Advanced control with MATLAB and SIMULINK* (The University Press, Cambridge).

Narendra, K. S. and Parthasarathy, K., 1991, Gradient method for the optimization of dynamical systems containing neural networks. *IEEE Transactions on Neural Networks,* **2**(2), pp. 252-262.

Nelder, J. A. and Mead, R., 1964, A simplex method for function minimisation. *Computer Journal,* **7,** pp. 308-313.

Newland, D. E., 1989, *Mechanical Vibration Analysis and Computation* (Longman, Harlow).

Nguyen, H. D. and Widrow, B., 1991, Neural networks for self-learning control systems. *International Journal of CONTROL,* **54**(6), pp. 1439-1451.

Segel, L., 1993, An overview of developments in road-vehicle dynamics: past, present and future. *Proceedings of the Institution of Mechanical Engineers Conference, Vehicle Ride and Handling,* 15-17 November, 1993, Birmingham (Mechanical Engineering Publications, London) pp. 1-12.

Sharp, R. S. and Crolla, D. A., 1987, Road vehicle suspension system design- a review. *Vehicle System Dynamics,* **16**, pp. 167-192.

Sharp, R. S. and Hassan, S. A., 1987, On the performance capabilities of active automobile suspension system of limited bandwidth. *Vehicle System Dynamics,* **16,** pp.213-225.

Sharp, R. S. and Hassan, J. H., 1988, Performance predictions for a pneumatic active car suspension system. *Proceedings of the Institution of Mechanical Engineers, Journal of Automobile Engineering (Part D),* **202,** pp. 243-250.

Sharp, R. S. and Pan, D., 1993, On the design of an active roll control system for a luxury car. *Proceedings of the Institution of Mechanical Engineers, Journal of Automobile Engineering (Part D),* **207,** pp. 275-284.

Sharp, R. S., 1995, Preview control of active suspensions, *Smart Vehicles,* Edited by J.P. Pauwelussen and H.B. Pacejka (Swets and Zeitlinger, Hisse) pp.166-182.

The Math Works Inc., 1992, *SIMULINK: Dynamic System Simulation Software User's Guide* (The Math Works Inc., Natick, MA).

van der Knaap, A. C. M., Venhovens, P. J. Th. and Pacejka, H. B., 1994, Evaluation and practical implementaion of a low power attitude and vibration control systems. *Proceedings of the International Symposium on Advanced Vehicle Control, AVEC '94* (JSAE) pp. 318-324.

Venhovens, P. J. Th., van der Knaap, A. C. M. and Pacejka, H. B., 1992, Semi-active vibration and attitude control. *Proceedings of the International Symposium on Advanced Vehicle Control, AVEC '92* (JSAE) pp. 170-175.

Wallentowitz, H. and Konik, D., 1991, Actively influenced suspension systems - survey of actual patent literature. *EAEC 3 rd International Conference, Vehicle Dynamics and Powertrain Engineering,* 1991, Strasbourg.

Williams, R. A., Best, A. and Crawford, I. L., 1993, Refined low frequency active suspension. *Proceedings of the Institution of Mechanical Engineers Conference, Vehicle Ride and Handling,* 15-17 November, 1993, Birmingham (Mechanical Engineering Publications, London) pp. 285-300.

Williams, R. A. and Miller, S. A., 1994, Power consumption in automotive active suspensions. *10 th International Conference on System Engineering,* September 1994, Coventry, **2,** pp.1405-1414.

Wilson, D. A., Sharp, R. S. and Hassan, S. A., 1986, The application of linear optimal control theory to the design of active automotive suspensions. *Vehicle System Dynamics,* **15,** pp.105-118.

Wong, J. Y.,1993, *Theory of Ground Vehicles - SECOND EDITION.* (John Wiley & Sons Inc., Third Avenue, New York, NY)

Wright, P. G. and Williams, D. A., 1984, The application of active suspension to high performance road vehicles. *Proceedings of the Institution of Mechanical Engineers,* Paper C239/84.

Wright, P. G. and Williams, D. A., 1989, The case for an irreversible active suspension system. *Proceedings of Autotechnologies '89 Conference*, Monaco, SAE_C/P-221.

Yeh, E. C., Lu, S. H. and Chen, C. C., 1994, A genetic algorithm based fuzzy system for semi-active suspension system design. *Proceedings of the International Symposium on Advanced Vehicle Control, AVEC '94* (JSAE) pp. 189-194.

Zadeh, A. G., Fahim, A. and El-Gindy, M., 1997, Neural network and fuzzy logic applications to vehicle systems: literature survey. *International Journal of Vehicle Design,* **18**(2), pp. 132-193.

Appendix A

Back-propagation for the Multi-Layer Neural Networks in Chapters 5 and 6

The architecture of the multi-layer neural network is shown in Fig. 5.2. There are S_p inputs, S_h hidden elements and $S_r = 1$ output from the network. The input-output relationship is:

$$u = L(w^R S(\mathrm{w}^{\mathrm{H}} L(x) + b^H) + b^R). \tag{A.1}$$

Output from each layer follows as:

Output layer: $$u = L(w^R a^H + b^R) = w^R a^H + b^R, \tag{A.2}$$

where w^R is a $1 \times S_h$ vector, a^H is a $S_h \times 1$ vector and b^R has one element.

Hidden layer: $$a^H = S(a^{H*}), \tag{A.3}$$

where $$a^{H*} = \mathrm{w}^{\mathrm{H}} x + b^H,$$

w^{H} is a $S_h \times S_p$ matrix, x is a $S_p \times 1$ vector and b^H is a $S_h \times 1$ vector.

Solving $\partial F_{NN}(x,w)/\partial w$, partial derivatives of u with respect to the weight parameters and biases are given for each layer as follows:

Output layer: Partial derivatives of u with respect to h th weighting parameter and bias are:

$$\frac{\partial u}{\partial w^R{}_{1,h}} = a^H{}_h, \qquad \frac{\partial u}{\partial b^R{}_1} = 1, \text{ respectively}. \tag{A.4}$$

Hidden layer: Partial derivatives of u with respect to the weighting parameter (between p th element at the input layer and h th element of the hidden layer) and h th bias can be:

$$\frac{\partial\, u}{\partial\, w^{H}{}_{h,p}} = \frac{\partial\, u}{\partial\, a^{H}{}_{h}} \cdot \frac{\partial\, a^{H}{}_{h}}{\partial\, a^{*H}{}_{h}} \cdot \frac{\partial\, a^{*H}{}_{h}}{\partial\, w^{H}{}_{h,p}} \qquad \text{(A.5)}$$

$$= w^{R}{}_{1,h}\ S'(a^{*H}{}_{h})\, x_{p},$$

$$\frac{\partial\, u}{\partial\, b^{H}{}_{h,p}} = \frac{\partial\, u}{\partial\, a^{H}{}_{h}} \cdot \frac{\partial\, a^{H}{}_{h}}{\partial\, a^{*H}{}_{h}} \cdot \frac{\partial\, a^{*H}{}_{h}}{\partial\, b^{H}{}_{h,p}}$$

$$= w^{R}{}_{1,h}\ S'(a^{*H}{}_{h})\, 1,$$

respectively. Solving $\partial\, F_{NN}(\boldsymbol{x},\boldsymbol{w})/\partial\, \boldsymbol{x}$, partial derivatives of u with respect to $\boldsymbol{x}$ are given as:

$$\frac{\partial\, u}{\partial\, \boldsymbol{x}} = \frac{\partial\, u}{\partial\, \boldsymbol{a}^{H}} \cdot \frac{\partial\, \boldsymbol{a}^{H}}{\partial\, \boldsymbol{a}^{*H}} \cdot \frac{\partial\, \boldsymbol{a}^{*H}}{\partial\, \boldsymbol{x}} \qquad \text{(A.6)}$$

$$= \boldsymbol{w}^{R}\ \mathbf{S}'(\boldsymbol{a}^{*H})\ \mathbf{w}^{\mathrm{H}},$$

where $\mathbf{S}'(\boldsymbol{a}^{*H})$ is a $S_h \times S_h$ diagonal matrix.

Appendix B

The Error due to the Mistake of the Parameter Setting

In chapter 7, the value of the wheel camber inertia was set to 40 kg m^2, incorrectly. The parameter value should be 1 kg m^2. However, the mistake was found too late to correct it. Therefore, simulation results with the wheel camber inertia of 40 kg m^2 and 1 kg m^2 are compared in this appendix to establish that the results are not critically affected.

Fig. B.1 shows a comparison between the vehicle body height responses of passive half-car models with the wheel camber inertia of 40 kg m^2 and of 1 kg m^2 under ramp lateral force inputs, which involve maximum lateral acceleration of 0.8 G. Fig. B.2 shows a comparison between the vehicle body roll responses in a similar manner.

From these figures, it can be concluded that the mistake does not affect these calculation results seriously.

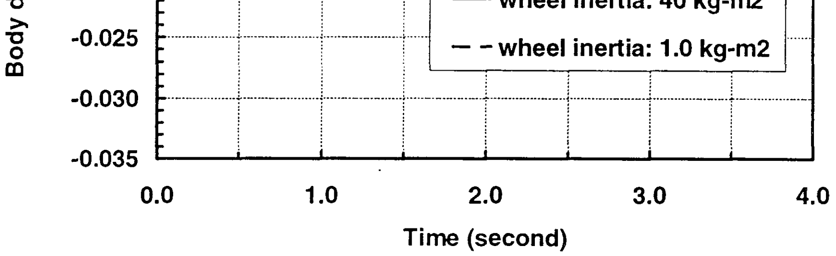


Fig. B.1 Comparison between the vehicle body height responses of passive half-car models with the wheel camber inertia of 40 kg m^2 and of 1 kg m^2 under ramp lateral force inputs, which involve maximum lateral acceleration of 0.8 G.

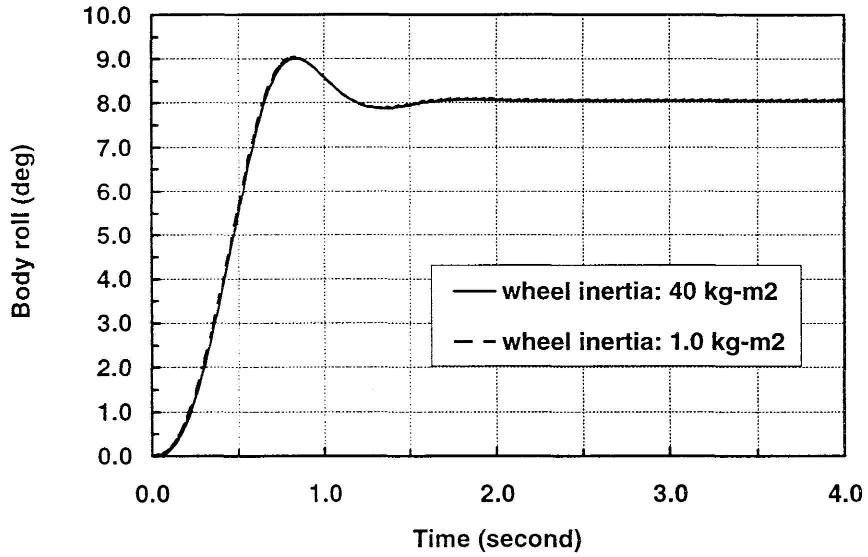


Fig. B.2 Comparison between the vehicle body roll responses of passive half-car models with the wheel camber inertia of 40 kg m^2 and of 1 kg m^2 under ramp lateral force inputs, which involve maximum lateral acceleration of 0.8 G.

Appendix C

Model Building of Variable Geometry Active Suspension System using AUTOSIM

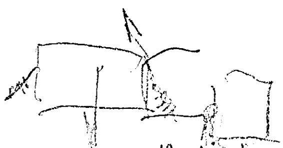
Firstly, there is the AUTOSIM programme of the variable geometry active suspension system described in Chapter 7. Secondly, there is the Rich Text Format file, written by AUTOSIM. The file was read by MICROSOFT WORD and printed from it.

AUTOSIM programme for variable geometry active suspension system (Programme name: Hal21r)

```
; AUTOSIM PROGRAM FOR VARIABLE LEVERAGE RATIO
;ACTIVE SUSPENSION SYSTEM
;HALF CAR MODEL
;WRITTEN 24/9/97 BY YUKIO;

(reset)
(setsym *multibody-system-name* "HAL21r")
(si)
(add-gravity :direction [nz])
(setsym *no-zees* t)

;ADD INITIAL POINTS ON BODY N
(add-point p1   :name "starboard lower arm inner pivot"   :body n
    :coordinates (0 y1 z1))
(add-point p2   :name "starboard upper arm inner pivot"   :body n
    :coordinates (0 y2 z2))
(add-point p3   :name "starboard spring damper top mounting point"   :body n
    :coordinates (0 y3 z3))
(add-point p5   :name "starboard upper arm outer pivot"   :body n
    :coordinates (0 y5 z5))
(add-point p6   :name "starboard lower arm outer pivot"   :body n
    :coordinates (0 y6 z6))
(add-point p7   :name "starboard tyre contact point"   :body n
    :coordinates (0 y7 z7))
(add-point p8   :name "starboard hub carrier centre mass point"   :body n
    :coordinates (0 y8 z8))
(add-point p9  :name "port lower arm inner pivot"  :body n
    :coordinates (0 y9 z9))
(add-point p10  :name "port upper arm inner pivot"  :body n
    :coordinates (0 y10 z10))
(add-point p11  :name "port spring damper top mounting point"  :body n
    :coordinates (0 y11 z11))
(add-point p13  :name "port upper arm outer pivot"  :body n
    :coordinates (0 y13 z13))
(add-point p14  :name "port lower arm outer pivot"  :body n
    :coordinates (0 y14 z14))
(add-point p15  :name "port tyre contact point"  :body n
    :coordinates (0 y15 z15))
(add-point p16  :name "port hub carrier centre mass point"  :body n
    :coordinates (0 y16 z16))
(add-point p17  :name "vehicle body centre mass point"  :body n
    :coordinates (0 y17 z17))

;VEHICLE MAIN BODY
(add-body vehicle  :parent n
    :name "vehicle body"
    :mass Mbod
    :inertia-matrix (ixx 0 0)
    :cm-coordinates p17
    :translate z
    :body-rotation-axes x  :small-angles t
    :parent-rotation-axis x
    :reference-axis y)

(add-point vehiclep3  :body vehicle  :coordinates p3)
(add-point vehiclep11  :body vehicle  :coordinates p11)

;STARBOARD SIDE OF VEHICLE
(add-body s_la  :parent vehicle  :name "starboard lower arm"  :mass 0
    :inertia-matrix 0  :joint-coordinates p1  :body-rotation-axes x
    :small-angles t  :parent-rotation-axis x  :reference-axis y)

(add-body s_ua  :parent vehicle  :name "starboard upper arm"  :mass 0
    :inertia-matrix 0  :joint-coordinates p2  :body-rotation-axes x
    :small-angles t   :parent-rotation-axis x   :reference-axis y)

(add-point uap5   :body s_ua   :coordinates p5)

(add-body s_hc   :parent s_la   :name "starboard hub carrier"   :mass Mwhl
    :inertia-matrix (ixxwhl 0 0)   :joint-coordinates p6   :cm-coordinates p8
    :body-rotation-axes x   :small-angles t   :parent-rotation-axis x   :reference-axis y)

(add-point hcp5   :body s_hc   :coordinates p5)
(add-point hcp6   :body s_hc   :coordinates p6)
(add-point hcp7   :body s_hc   :coordinates p7)

(add-speed-constraint "dot ((vel(uap5,hcp5)),[vehicley])"  :u"ru(s_hc)")
(add-speed-constraint "dot ((vel(uap5,hcp5)),[vehiclez])"  :u"ru(s_ua)")

;PORT SIDE OF VEHICLE
(add-body p_la   :parent vehicle   :name "port lower arm"   :mass 0
    :inertia-matrix 0   :joint-coordinates p9   :body-rotation-axes x
    :small-angles t   :parent-rotation-axis x   :reference-axis y)

(add-body p_ua   :parent vehicle   :name "port upper arm"   :mass 0
    :inertia-matrix 0   :joint-coordinates p10   :body-rotation-axes x
    :small-angles t   :parent-rotation-axis x   :reference-axis y)

(add-point uap13   :body p_ua   :coordinates p13)

(add-body p_hc   :parent p_la   :name "port hub carrier"   :mass Mwhl
    :inertia-matrix (ixxwhl 0 0)   :joint-coordinates p14   :cm-coordinates p16
    :body-rotation-axes x   :small-angles t   :parent-rotation-axis x   :reference-axis y)

(add-point hcp13   :body p_hc   :coordinates p13)
(add-point hcp14   :body p_hc   :coordinates p14)
(add-point hcp15   :body p_hc   :coordinates p15)

(add-speed-constraint "dot ((vel(uap13,hcp13)),[vehicley])"  :u"ru(p_hc)")
(add-speed-constraint "dot ((vel(uap13,hcp13)),[vehiclez])"  :u"ru(p_ua)")

;PD CONTROLLED ACTUATIONS AND MOVING POINTS
(setsym s_zdis "mag(pos(p3,p6))-mag(pos(vehiclep3,hcp6))")
```

```
(setsym p_zdis "mag(pos(p11,p14))-mag(pos(vehiclep11,hcp14))")
(setsym ds_zdis "dxdt(@s_zdis)")
(setsym dp_zdis "dxdt(@p_zdis)")

;(add-variables dyvars real smov dsmov)
(add-state-variable smov dsmov l)
                             G13                              G14
(add-equation difeqn dsmov "Grp*(@s_zdis-@p_zdis)+Grd*(@ds_zdis-@dp_zdis) +
                                  G12
G11 Ghp*(@s_zdis+@p_zdis)+Ghd*(@ds_zdis+@dp_zdis)"
    :comment "PD controlled differential equations of actuator dynamics (starboard)")

;(add-variables dyvars real pmov dpmov)
(add-state-variable pmov dpmov l)
(add-equation difeqn dpmov  "Grp*(@s_zdis-@p_zdis)+Grd*(@ds_zdis-@dp_zdis)-
Ghp*(@s_zdis+@p_zdis)-Ghd*(@ds_zdis+@dp_zdis)"
    :comment "PD controlled differential equations of actuator dynamics (port)")

(set-units bank "a")

(add-point lap4   :body s_la
    :name "s. actuation point"
    :coordinates (0 "y4+smov" "z4-smov*tan(bank)")
    :moving t)

(add-point lap12    :body p_la
    :name "p. actuation point"
    :coordinates (0 "y12+pmov" "z12+pmov*tan(bank)")
    :moving t)

;ADD LATERAL FORCE TO VEHICLE CENTRE MASS
(add-table LAcc "lateral acceleration to vehicle body (m/s/s) vs. time"   :npts 30
    :table-function tabf   :yunits "l/t/t"
    :values ((0 0) (0.5 7.848)))

(add-line-force centrF  :name "lateral force to vehicle body"
    :direction [ny]   :magnitude "Mbod*LAcc(t)"   :point1 vehiclecm)

(setsym Preload "(ini_Mbod*G/2)/ini_dldz)")
(setsym Tpreload "(Mbod/2+Mwhl)*G")
(set-units ini_Mbod "f*t**2/l")
```

???



Mbod/2 (Mbod/2+Mwhl)

[illegible] M = -ks(x-x0)-cs

```
;ADD VERTICAL FORCES
(add-strut s_spdp   :name "starboard spring/damper force"
     :magnitude "-ks*(x-x0)-cs*v+@Preload"
     :point1 vehiclep3   :point2 lap4)

(add-line-force s_verF   :name "starboard tyre vertical force"
    :direction [nz]   :magnitude  "-kt*(x-x0)-ct*v-@Tpreload"
    :point1  hcp7)

(add-strut p_spdp   :name "port spring/damper force"
     :magnitude "-ks*(x-x0)-cs*v+@Preload"
     :point1 vehiclep11   :point2 lap12)

(add-line-force p_verF   :name "port tyre vertical force"
    :direction [nz]   :magnitude  "-kt*(x-x0)-ct*v-@Tpreload"
    :point1  hcp15)

(setsym load_trans "fm(s_verF)/(fm(s_verF)+fm(p_verF))")

;ADD LATERAL FORCES
(add-line-force s_whlF   :name "starboard tyre lateral external force"
    :direction [ny]   :magnitude "Mwhl*LAcc(t)"
    :point1 s_hccm)

(add-line-force s_sideF   :name "starboard tyre side force"
    :direction [ny]   :magnitude "-1*@load_trans*(Mbod+Mwhl*2)*LAcc(t)"
    :point1 hcp7)

(add-line-force p_whlF   :name "port tyre lateral external force"
    :direction [ny]   :magnitude "Mwhl*LAcc(t)"
    :point1 p_hccm)

(add-line-force p_sideF   :name "port tyre side force"
    :direction [ny]   :magnitude "-1*(1-@load_trans)*(Mbod+Mwhl*2)*LAcc(t)"
    :point1 hcp15)

;OUTPUT
(add-standard-output)
```

```
(add-out @s_zdis "s_zdis"  :long-name "starboard potentiometer displacement"
    :body s_hc  :units l)
(add-out @p_zdis "p_zdis"  :long-name "port potentiometer displacement"
    :body p_hc  :units l)
(add-out @ds_zdis "ds_zdis"  :long-name "starboard potentiometer velocity"
    :body s_hc  :units l/t)
(add-out @dp_zdis "dp_zdis"  :long-name "port potentiometer velocity"
    :body p_hc  :units l/t)

;DIAGNOSTICS
(print-points)  (print-coordinates)  (print-bodies)

;;;;This command is used for the FORTRAN simulation
;;;;;;(finish)

;;;;This command is used to create the RTF file
(dynamics  :formalism numerical-lud)

;UNITS AND DEFAULT PARAMETER VALUES
(set-defaults
    y1 0.3 z1 -0.15 y2 0.4 z2 -0.35 y3 0.485 z3 -0.55 y4 0.185 z4 -0.015
    y5 0.67 z5 -0.5 y6 0.67 z6 -0.18 y7 0.75 z7 0 y8 0.75 z8 -0.34
    y9 -0.3 z9 -0.15 y10 -0.4 z10 -0.35 y11 -0.485 z11 -0.55 y12 -0.185 z12 -0.015
    y13 -0.67 z13 -0.5 y14 -0.67 z14 -0.18 y15 -0.75 z15 0 y16 -0.75 z16 -0.34
    y17 0 z17 -0.5
    ks 60000 cs 6000 kt 2000000 ct 50
    Mbod 665 Mwhl 25 ixx 100 ixxwhl 40 ini_dldz 0.5 ini_Mbod 590
    bank 0.349 Ghd 0.02 Ghp 0.22 Grd 0.7835 Grp 2.8866
    iprint 50 step 0.001 stopt 10)

(set-defaults format "'TEXT")

(set-names
    ks "suspension spring stiffness"
    cs "suspension damping coefficient"
    kt "tyre spring stiffness"
    ct "tyre damping coefficient"
    Mbod "mass of vehicle body"
    Mwhl "mass of wheel"
ixx "inertia of vehicle body"
ixxwhl "camber inertia of hub carrier"
ini_dldz "initial value of the leverage ratio"
ini_Mbod "mass of vehicle body (nominal condition)"
bank "angle of actuation on coordinate sys. of lower arm"
Grp "gain of roll controller regarding body roll" G13
Grd "gain of roll controller regarding roll rate" G14
Ghp "gain of height controller regarding body height" G11
Ghd "gein of height controller regarding body height velocity" G12
smov "starboard actuator displacement"
pmov "port actuator displacement")
```

;;;;;This command is used to create the RTF file

```
(write-to-file write-eqs "half21R.rtf") ???
```

The equations of motion of the system has been written by AUTOSIM in Rich Text Format (RTF file). The RTF file created has been read into WORD and edited below (except pre-computed constants):

****

Equations For The Hal21r

Dynamic simulation of hal21r. Version created by AUTOSIM 2.0h on September 24, 1997. Copyright (c) The Regents of The University of Michigan, 1989 - 1995. All rights reserved. The hal21r is represented mathematically by 14 ordinary differential equations that describe its kinematical and dynamical behavior. It is composed of 7 bodies, has 4 DOF, and includes 9 forces and 0 moments.

Bodies

Vehicle body (VEHICLE); parent=N; trans coord = Q_1; rot coord = Q_2
Starboard lower arm (S_LA); parent=VEHICLE; rot coord = Q_3
Starboard hub carrier (S_HC); parent=S_LA; rot coord = Q_4
Starboard upper arm (S_UA); parent=VEHICLE; rot coord = Q_5
Port lower arm (P_LA); parent=VEHICLE; rot coord = Q_6
Port hub carrier (P_HC); parent=P_LA; rot coord = Q_7
Port upper arm (P_UA); parent=VEHICLE; rot coord = Q_8

Multibody Coordinates

Q_1	Abs. Z trans. of VEHICLE0 (m)
Q_2	Abs. X rot. of VEHICLE (rad)

Q_3	X rot. of S_LA rel. to VEHICLE (rad)
Q_4	X rot. of S_HC rel. to S_LA (rad)
Q_5	X rot. of S_UA rel. to VEHICLE (rad)
Q_6	X rot. of P_LA rel. to VEHICLE (rad)
Q_7	X rot. of P_HC rel. to P_LA (rad)
Q_8	X rot. of P_UA rel. to VEHICLE (rad)
S_{MOV}	starboard actuator displacement (m)
P_{MOV}	port actuator displacement (m)

Independent Speeds

U_1	Abs. Z trans. speed of VEHICLE0 (m/s)
U_2	Abs. X rot. speed of VEHICLE (rad/s)
U_3	X rot. speed of S_LA rel. to VEHICLE (rad/s)
U_4	X rot. speed of P_LA rel. to VEHICLE (rad/s)

Forces

CENTRF Lateral force to vehicle body; Acts on the vehicle body from the inertial reference through mass center of the vehicle body. Magnitude = F_{M1}; Direction = $\mathbf{n}_y$.

S_SPDP Starboard spring/damper force; Acts on the vehicle body from the starboard lower arm through VEHICLEP3 and s. actuation point. Magnitude = $-F_{M2}$; Direction = $(1/(-pc102 -pc105 -(pc100 -pc103)\, s_{mov} -(pc104 -pc106 + (pc98 + pc107)\, s_{mov})\, q_3 + (y_4 + s_{mov})\, (pc99 + s_{mov} -pc98\, q_3) + (z_4 -pc97\, s_{mov})\, (pc101 -pc97\, s_{mov} + pc100\, q_3))^{0.5}\, pc100\, \mathbf{vehicle}_y + 1/(-pc102 -pc105 -(pc100 -pc103)\, s_{mov} -(pc104 -pc106 + (pc98 + pc107)\, s_{mov})\, q_3 + (y_4 + s_{mov})\, (pc99 + s_{mov} -pc98\, q_3) + (z_4 -pc97\, s_{mov})\, (pc101 -pc97\, s_{mov} + pc100\, q_3))^{0.5}\, pc98\, \mathbf{vehicle}_z -1/(-pc102 -pc105 -(pc100 -pc103)\, s_{mov} -(pc104 -pc106 + (pc98 + pc107)\, s_{mov})\, q_3 + (y_4 + s_{mov})\, (pc99 + s_{mov} -pc98\, q_3) + (z_4 -pc97\, s_{mov})\, (pc101 - pc97\, s_{mov} + pc100\, q_3))^{0.5}\, (y_4 + s_{mov})\, \mathbf{s\_la}_y -1/(-pc102 -pc105 -(pc100 -pc103)\, s_{mov} -(pc104 -pc106 + (pc98 + pc107)\, s_{mov})\, q_3 + (y_4 + s_{mov})\, (pc99 + s_{mov} -pc98\, q_3) + (z_4 -pc97\, s_{mov})\, (pc101 -pc97\, s_{mov} + pc100\, q_3))^{0.5}\, (z_4 -pc97\, s_{mov})\, \mathbf{s\_la}_z)$.

S_VERF Starboard tyre vertical force; Acts on the starboard hub carrier from the inertial reference through HCP7. Magnitude = $-F_{M3}$; Direction = $\mathbf{n}_z$.

P_SPDP Port spring/damper force; Acts on the vehicle body from the port lower arm through VEHICLEP11 and p. actuation point. Magnitude = $-F_{M4}$; Direction = $-(1/(pc112 + pc115 + (pc110 + pc113)\, p_{mov} + (pc114 -pc116 + (pc108 -pc117)\, p_{mov})\, q_6 + (y_{12} + p_{mov})\, (pc109 + p_{mov} + pc108\, q_6) + (z_{12} + pc97\, p_{mov})\, (pc111 + pc97\, p_{mov} -pc110\, q_6))^{0.5}\, pc110\, \mathbf{vehicle}_y + 1/(pc112 + pc115 + (pc110 + pc113)\, p_{mov} + (pc114 -pc116 + (pc108 -pc117)\, p_{mov})\, q_6 + (y_{12} + p_{mov})\, (pc109 + p_{mov} + pc108\, q_6) + (z_{12} + pc97\, p_{mov})\, (pc111 + pc97\, p_{mov} -pc110\, q_6))^{0.5}\, pc108\, \mathbf{vehicle}_z + 1/(pc112 + pc115 + (pc110 + pc113)\, p_{mov} + (pc114 -pc116 + (pc108 -pc117)\, p_{mov})\, q_6 + (y_{12} + p_{mov})\, (pc109 + p_{mov} + pc108\, q_6) + (z_{12} + pc97\, p_{mov})\, (pc111 + pc97\, p_{mov} -pc110\, q_6))^{0.5}\, (y_{12} + p_{mov})\, \mathbf{p\_la}_y + 1/(pc112 + pc115 + (pc110 + pc113)\, p_{mov} + (pc114 -pc116 + (pc108 -pc117)\, p_{mov})\, q_6 + (y_{12} + p_{mov})$

$(p_{c109} + p_{\mathrm{mov}} + p_{c108}\, q_6) + (z_{12} + p_{c97}\, p_{\mathrm{mov}})\, (p_{c111} + p_{c97}\, p_{\mathrm{mov}} - p_{c110}\, q_6))^{0.5}\, (z_{12} + p_{c97}\, p_{\mathrm{mov}})\, \mathbf{p\_la}_{\mathrm{Z}})$.

P_VERF Port tyre vertical force; Acts on the port hub carrier from the inertial reference through HCP15. Magnitude = $-F_{M5}$; Direction = $\mathbf{n}_{\mathrm{Z}}$.

S_WHLF Starboard tyre lateral external force; Acts on the starboard hub carrier from the inertial reference through mass center of the starboard hub carrier. Magnitude = F_{M6}; Direction = $\mathbf{n}_{\mathrm{Y}}$.

S_SIDEF Starboard tyre side force; Acts on the starboard hub carrier from the inertial reference through HCP7. Magnitude = $-F_{M7}$; Direction = $\mathbf{n}_{\mathrm{Y}}$.

P_WHLF Port tyre lateral external force; Acts on the port hub carrier from the inertial reference through mass center of the port hub carrier. Magnitude = F_{M8}; Direction = $\mathbf{n}_{\mathrm{Y}}$.

P_SIDEF Port tyre side force; Acts on the port hub carrier from the inertial reference through HCP15. Magnitude = $-F_{M9}$; Direction = $\mathbf{n}_{\mathrm{Y}}$.

Parameters

B_{ANK}	angle of actuation on coordinate sys. of lower arm (0.349 -)
C_{S}	suspension damping coefficient (6000 -)
C_{T}	tyre damping coefficient (50 -)
G_{HD}	gein of height controller regarding body height velocity (0.02 -)
G_{HP}	gain of height controller regarding body height (0.22 -)
G_{RP}	gain of roll controller regarding body roll (2.8866 -)
$I_{\mathrm{NI\_DLDZ}}$	initial value of the leverage ratio (0.5 -)
$I_{\mathrm{NI\_MBOD}}$	mass of vehicle body (nominal condition) (590 -)
I_{XXWHL}	camber inertia of hub carrier (40 -)
I_{XX}	inertia of vehicle body (100 -)
K_{S}	suspension spring stiffness (60000 -)
K_{T}	tyre spring stiffness (2000000 -)
M_{BOD}	mass of vehicle body (665 -)
M_{WHL}	mass of wheel (25 -)
Y_{10}	Y coordinate of attachment point for the port upper arm (-0.4 -)
Y_{11}	Y coordinate of VEHICLEP11 (-0.485 -)
Y_{12}	term in Y coordinate of p. actuation point (-0.185 -)
Y_{13}	Y coordinate of port upper arm outer pivot (-0.67 -)
Y_{14}	Y coordinate of port lower arm outer pivot (-0.67 -)
Y_{15}	Y coordinate of port tyre contact point (-0.75 -)
Y_{16}	Y coordinate of port hub carrier centre mass point (-0.75 -)
Y_{17}	Y coordinate of mass center of composite body VEHICLE (0 -)
Y_{1}	Y coordinate of attachment point for the starboard lower arm (0.3 -)
Y_{2}	Y coordinate of attachment point for the starboard upper arm (0.4 -)
Y_{3}	Y coordinate of VEHICLEP3 (0.485 -)
Y_{4}	term in Y coordinate of s. actuation point (0.185 -)
Y_{5}	Y coordinate of starboard upper arm outer pivot (0.67 -)

Y_6	Y coordinate of starboard lower arm outer pivot (0.67 -)
Y_7	Y coordinate of starboard tyre contact point (0.75 -)
Y_8	Y coordinate of starboard hub carrier centre mass point (0.75 -)
Y_9	Y coordinate of attachment point for the port lower arm (-0.3 -)
Z_{10}	Z coordinate of attachment point for the port upper arm (-0.35 -)
Z_{11}	Z coordinate of VEHICLEP11 (-0.55 -)
Z_{12}	term in Z coordinate of p. actuation point (-0.015 -)
Z_{13}	Z coordinate of port upper arm outer pivot (-0.5 -)
Z_{14}	Z coordinate of port lower arm outer pivot (-0.18 -)
Z_{15}	Z coordinate of port tyre contact point (0 -)
Z_{16}	Z coordinate of port hub carrier centre mass point (-0.34 -)
Z_{17}	Z coordinate of mass center of composite body VEHICLE (-0.5 -)
Z_1	Z coordinate of attachment point for the starboard lower arm (-0.15 -)
Z_2	Z coordinate of attachment point for the starboard upper arm (-0.35 -)
Z_3	Z coordinate of VEHICLEP3 (-0.55 -)
Z_4	term in Z coordinate of s. actuation point (-0.015 -)
Z_5	Z coordinate of starboard upper arm outer pivot (-0.5 -)
Z_6	Z coordinate of starboard lower arm outer pivot (-0.18 -)
Z_7	Z coordinate of starboard tyre contact point (0 -)
Z_8	Z coordinate of starboard hub carrier centre mass point (-0.34 -)
Z_9	Z coordinate of attachment point for the port lower arm (-0.15 -)

Precomputed Constants

(There are 378 pre-computed constants: from P_{C1} to P_{C378})

Equations Of Motion

Each derivative evaluation requires 456 multiply/divides, 567 add/subtracts, and 25 function/subroutine calls.

define extra variables

$S_{\mathrm{MOV}} = Q_9$
$P_{\mathrm{MOV}} = Q_{10}$

Kinematical equations

$Q'_1 = U_1$
$Q'_2 = U_2$
$Q'_3 = U_3$
$Q'_4 = -((P_{C23} + P_{C47})\, U_3 + U_2\, (P_{C44} + P_{C46} -(P_{C20} + P_{C45})\, (Q_3 + Q_4)))$
$Q'_5 = P_{C42}\, U_3 + U_2\, (P_{C37} - P_{C43}\, (Q_3 + Q_4))$
$Q'_6 = U_4$

$$Q'_7 = -((P_{C71} + P_{C95})\, U_4 + U_2\, (-P_{C92} + P_{C94} + (P_{C68} + P_{C93})\, (Q_6 + Q_7)))$$
$$Q'_8 = -(P_{C90}\, U_4 - U_2\, (P_{C85} - P_{C91}\, (Q_6 + Q_7)))$$

External subroutines and equations for forces and moments

define expression for lateral force to vehicle body

$$F_{M1} = M_{\mathrm{BOD}}\, \mathrm{lacc}(T)$$

define expression for starboard spring/damper force

$$\begin{aligned} F_{M2} = &-(P_{C156} + P_{C158} + 1/(P_{C165} + S_{\mathrm{MOV}}\, (Y_4 + P_{C99} - P_{C100} + P_{C103} - P_{C161} - P_{C162} \\ &+ (1.0 + P_{C163})\, S_{\mathrm{MOV}}))^{0.5}\, (P_{C172} - P_{C173} + (P_{C170} + P_{C171})\, S_{\mathrm{MOV}})\, U_3 - (P_{C165} + \\ &S_{\mathrm{MOV}}\, (Y_4 + P_{C99} - P_{C100} + P_{C103} - P_{C161} - P_{C162} + (1.0 + P_{C163})\, S_{\mathrm{MOV}}) - (2 \\ &P_{C104} - 2\, P_{C106} + (2\, P_{C107} + P_{C164})\, S_{\mathrm{MOV}})\, Q_3)^{0.5}\, K_{\mathrm{S}}) \end{aligned}$$

define expression for starboard tyre vertical force

$$\begin{aligned} F_{M3} = &P_{C176} + C_{\mathrm{T}}\, U_1 + K_{\mathrm{T}}\, Q_1 - P_{C178}\, Q_4 + (P_{C184} + P_{C185} + P_{C186} + P_{C187})\, U_2 + \\ &(P_{C187} - P_{C188} + P_{C189})\, U_3 + (P_{C177} - P_{C178} + P_{C179})\, Q_2 - (P_{C178} - P_{C179})\, Q_3 \end{aligned}$$

define expression for port spring/damper force

$$\begin{aligned} F_{M4} = &-(P_{C156} + P_{C191} - 1/(P_{C197} + P_{\mathrm{MOV}}\, (Y_{12} + P_{C109} + P_{C110} + P_{C113} + P_{C194} + \\ &P_{C195} + (1.0 + P_{C163})\, P_{\mathrm{MOV}}))^{0.5}\, (P_{C204} - P_{C205} - (P_{C202} - P_{C203})\, P_{\mathrm{MOV}})\, U_4 - \\ &(P_{C197} + P_{\mathrm{MOV}}\, (Y_{12} + P_{C109} + P_{C110} + P_{C113} + P_{C194} + P_{C195} + (1.0 + P_{C163}) \\ &P_{\mathrm{MOV}}) - (-2\, P_{C114} + 2\, P_{C116} + (2\, P_{C117} - P_{C196})\, P_{\mathrm{MOV}})\, Q_6)^{0.5}\, K_{\mathrm{S}}) \end{aligned}$$

define expression for port tyre vertical force

$$\begin{aligned} F_{M5} = &P_{C176} + C_{\mathrm{T}}\, U_1 + K_{\mathrm{T}}\, Q_1 + P_{C207}\, Q_7 + (P_{C209} + P_{C210} + P_{C211} - P_{C212})\, U_2 - \\ &(P_{C212} - P_{C213} + P_{C214})\, U_4 + (P_{C206} + P_{C207} - P_{C208})\, Q_2 + (P_{C207} - P_{C208})\, Q_6 \end{aligned}$$

define expression for starboard tyre lateral external force

$$F_{M6} = M_{\mathrm{WHL}}\, \mathrm{lacc}(T)$$

define expression for starboard tyre side force

$$F_{M7} = P_{C216}\, \mathrm{lacc}(T)\, F_{M3}/(F_{M3} + F_{M5})$$

define expression for port tyre lateral external force

$$F_{M8} = M_{\mathrm{WHL}}\, \mathrm{lacc}(T)$$

define expression for port tyre side force

$$F_{M9} = P_{C216}\,\mathrm{lacc}(T)\,(1.0 - F_{M3}/(F_{M3} + F_{M5}))$$

terms for force array eq. 4

$$G_{FA1} = -(-P_{C312} + F_{M3} + F_{M5} + Q_2 (1/(-P_{C264} - P_{C261} S_{\mathrm{MOV}} - (P_{C263} + P_{C262} S_{\mathrm{MOV}}) Q_3 + (Y_4 + S_{\mathrm{MOV}}) (P_{C99} + S_{\mathrm{MOV}} - P_{C98} Q_3) + (Z_4 - P_{C97} S_{\mathrm{MOV}}) (P_{C101} - P_{C97} S_{\mathrm{MOV}} + P_{C100} Q_3))^{0.5} (Z_4 - P_{C97} S_{\mathrm{MOV}}) Q_3 F_{M2} + 1/(P_{C260} + P_{C257} P_{\mathrm{MOV}} + (P_{C259} + P_{C258} P_{\mathrm{MOV}}) Q_6 + (Y_{12} + P_{\mathrm{MOV}}) (P_{C109} + P_{\mathrm{MOV}} + P_{C108} Q_6) + (Z_{12} + P_{C97} P_{\mathrm{MOV}}) (P_{C111} + P_{C97} P_{\mathrm{MOV}} - P_{C110} Q_6))^{0.5} (Z_{12} + P_{C97} P_{\mathrm{MOV}}) Q_6 F_{M4}))$$

$$G_{FA2} = P_{C313} - Q_2 (P_{C314} + Y_{17} F_{M1}) - Z_{17} (F_{M1} + 1/(-P_{C264} - P_{C261} S_{\mathrm{MOV}} - (P_{C263} + P_{C262} S_{\mathrm{MOV}}) Q_3 + (Y_4 + S_{\mathrm{MOV}}) (P_{C99} + S_{\mathrm{MOV}} - P_{C98} Q_3) + (Z_4 - P_{C97} S_{\mathrm{MOV}}) (P_{C101} - P_{C97} S_{\mathrm{MOV}} + P_{C100} Q_3))^{0.5} (P_{C308} + S_{\mathrm{MOV}} - (Z_4 - P_{C97} S_{\mathrm{MOV}}) Q_3) F_{M2} + 1/(P_{C260} + P_{C257} P_{\mathrm{MOV}} + (P_{C259} + P_{C258} P_{\mathrm{MOV}}) Q_6 + (Y_{12} + P_{\mathrm{MOV}}) (P_{C109} + P_{\mathrm{MOV}} + P_{C108} Q_6) + (Z_{12} + P_{C97} P_{\mathrm{MOV}}) (P_{C111} + P_{C97} P_{\mathrm{MOV}} - P_{C110} Q_6))^{0.5} (P_{C309} + P_{\mathrm{MOV}} - (Z_{12} + P_{C97} P_{\mathrm{MOV}}) Q_6) F_{M4}) - (P_{C288} - P_{C3} Q_4 - P_{C287} (Q_3 + Q_4)) (F_{M6} - F_{M7} - F_{M3} (Q_2 + Q_3 + Q_4)) + (P_{C301} + P_{C51} Q_7 - P_{C300} (Q_6 + Q_7)) (-F_{M8} + F_{M9} + F_{M5} (Q_2 + Q_6 + Q_7)) + F_{M2} (1/(P_{C333} + S_{\mathrm{MOV}} (P_{C351} + P_{C350} S_{\mathrm{MOV}}) - (P_{C353} + P_{C352} S_{\mathrm{MOV}}) Q_3)^{0.5} (P_{C334} - P_{C335} S_{\mathrm{MOV}} + (P_{C336} + Y_{17} S_{\mathrm{MOV}}) Q_3) + 1/(-P_{C264} - P_{C261} S_{\mathrm{MOV}} - (P_{C263} + P_{C262} S_{\mathrm{MOV}}) Q_3 + (Y_4 + S_{\mathrm{MOV}}) (P_{C99} + S_{\mathrm{MOV}} - P_{C98} Q_3) + (Z_4 - P_{C97} S_{\mathrm{MOV}}) (P_{C101} - P_{C97} S_{\mathrm{MOV}} + P_{C100} Q_3))^{0.5} (-P_{C357} + P_{C354} S_{\mathrm{MOV}} - (P_{C356} + P_{C355} S_{\mathrm{MOV}}) Q_3 - (Y_4 + S_{\mathrm{MOV}}) (P_{C267} - P_{C265} Q_3) + (Z_4 - P_{C97} S_{\mathrm{MOV}}) (P_{C265} + P_{C267} Q_3))) + F_{M4} (1/(P_{C337} + P_{\mathrm{MOV}} (P_{C358} + P_{C350} P_{\mathrm{MOV}}) + (P_{C360} + P_{C359} P_{\mathrm{MOV}}) Q_6)^{0.5} (P_{C338} + P_{C335} P_{\mathrm{MOV}} + (P_{C339} + Y_{17} P_{\mathrm{MOV}}) Q_6) + 1/(P_{C260} + P_{C257} P_{\mathrm{MOV}} + (P_{C259} + P_{C258} P_{\mathrm{MOV}}) Q_6 + (Y_{12} + P_{\mathrm{MOV}}) (P_{C109} + P_{\mathrm{MOV}} + P_{C108} Q_6) + (Z_{12} + P_{C97} P_{\mathrm{MOV}}) (P_{C111} + P_{C97} P_{\mathrm{MOV}} - P_{C110} Q_6))^{0.5} (-P_{C364} + P_{C361} P_{\mathrm{MOV}} - (P_{C363} + P_{C362} P_{\mathrm{MOV}}) Q_6 + (Y_{12} + P_{\mathrm{MOV}}) (P_{C272} - P_{C270} Q_6) - (Z_{12} + P_{C97} P_{\mathrm{MOV}}) (P_{C270} + P_{C272} Q_6))) + (P_{C284} + P_{C285} Q_3 + P_{C340} Q_4) (G - P_{C6} (P_{C221} U_3 + U_2 (P_{C222} + P_{C217} (Q_3 + Q_4)))^2) + (P_{C297} + P_{C298} Q_6 + P_{C341} Q_7) (G + P_{C54} (P_{C223} U_4 + U_2 (P_{C224} - P_{C219} (Q_6 + Q_7)))^2) - (P_{C283} + P_{C4} Q_4 + P_{C282} (Q_3 + Q_4)) (F_{M3} + (Q_2 + Q_3 + Q_4) (F_{M6} - F_{M7})) - (P_{C296} - P_{C52} Q_7 + P_{C295} (Q_6 + Q_7)) (F_{M5} - (Q_2 + Q_6 + Q_7) (-F_{M8} + F_{M9})) - (P_{C289} - P_{C290} Q_3 - P_{C342} Q_4) (G (Q_2 + Q_3 + Q_4) - P_{C5} (P_{C221} U_3 + U_2 (P_{C222} + P_{C217} (Q_3 + Q_4)))^2) - (P_{C302} - P_{C303} Q_6 - P_{C343} Q_7) (G (Q_2 + Q_6 + Q_7) + P_{C53} (P_{C223} U_4 + U_2 (P_{C224} - P_{C219} (Q_6 + Q_7)))^2) - (P_{C344} + P_{C217} (Q_3 + Q_4)) (F_{M3} (P_{C275} - P_{C276} (Q_2 + Q_3 + Q_4)) - F_{M7} (P_{C276} + P_{C275} (Q_2 + Q_3 + Q_4))) + (P_{C345} - P_{C219} (Q_6 + Q_7)) (F_{M5} (P_{C277} - P_{C278} (Q_2 + Q_6 + Q_7)) - F_{M9} (P_{C278} + P_{C277} (Q_2 + Q_6 + Q_7)))$$

$$G_{FA3}= -(1/(-P_{C264} -P_{C261} S_{\mathrm{MOV}} -(P_{C263} + P_{C262} S_{\mathrm{MOV}}) Q_3 + (Y_4 + S_{\mathrm{MOV}}) (P_{C99} + S_{\mathrm{MOV}} -P_{C98} Q_3) + (Z_4 -P_{C97} S_{\mathrm{MOV}}) (P_{C101} -P_{C97} S_{\mathrm{MOV}} + P_{C100} Q_3))^{0.5} (-P_{C263} -P_{C262} S_{\mathrm{MOV}} + (P_{C315} + P_{C261} S_{\mathrm{MOV}}) Q_3) F_{M2} + F_{M3} (P_{C347} -P_{C346} (Q_2 + Q_3 + Q_4)) -F_{M7} (P_{C346} + P_{C347} (Q_2 + Q_3 + Q_4)) + (P_{C248} -P_{C3} Q_4) (F_{M6} -F_{M7} -F_{M3} (Q_2 + Q_3 + Q_4)) + (P_{C246} + P_{C4} Q_4) (F_{M3} + (Q_2 + Q_3 + Q_4) (F_{M6} -F_{M7})) -(P_{C293} + P_{C286} Q_4) (G -P_{C6} (P_{C221} U_3 + U_2 (P_{C222} + P_{C217} (Q_3 + Q_4)))^2) + (P_{C292} -P_{C291} Q_4) (G (Q_2 + Q_3 + Q_4) -P_{C5} (P_{C221} U_3 + U_2 (P_{C222} + P_{C217} (Q_3 + Q_4)))^2))$$

$$G_{FA4}= 1/(P_{C260} + P_{C257} P_{\mathrm{MOV}} + (P_{C259} + P_{C258} P_{\mathrm{MOV}}) Q_6 + (Y_{12} + P_{\mathrm{MOV}}) (P_{C109} + P_{\mathrm{MOV}} + P_{C108} Q_6) + (Z_{12} + P_{C97} P_{\mathrm{MOV}}) (P_{C111} + P_{C97} P_{\mathrm{MOV}} -P_{C110} Q_6))^{0.5} (-P_{C259} -P_{C258} P_{\mathrm{MOV}} + (P_{C324} + P_{C257} P_{\mathrm{MOV}}) Q_6) F_{M4} + F_{M5} (P_{C349} -P_{C348} (Q_2 + Q_6 + Q_7)) -F_{M9} (P_{C348} + P_{C349} (Q_2 + Q_6 + Q_7)) -(P_{C256} -P_{C51} Q_7) (-F_{M8} + F_{M9} + F_{M5} (Q_2 + Q_6 + Q_7)) + (P_{C254} + P_{C52} Q_7) (F_{M5} -(Q_2 + Q_6 + Q_7) (-F_{M8} + F_{M9})) -(P_{C306} + P_{C299} Q_7) (G + P_{C54} (P_{C223} U_4 + U_2 (P_{C224} -P_{C219} (Q_6 + Q_7)))^2) + (P_{C305} -P_{C304} Q_7) (G (Q_2 + Q_6 + Q_7) + P_{C53} (P_{C223} U_4 + U_2 (P_{C224} -P_{C219} (Q_6 + Q_7)))^2)$$

terms for mass matrix, element 4, 4

$$M_{1,1} = P_{C216}$$

$$M_{2,1} = P_{C369} -P_{C280} Q_2 + P_{C285} Q_3 + P_{C298} Q_6 + P_{C340} Q_4 + P_{C341} Q_7 -P_{C289} (Q_2 + Q_3 + Q_4) -P_{C302} (Q_2 + Q_6 + Q_7)$$

$$M_{1,2} = M_{2,1}$$

$$M_{2,2} = P_{C372} + (P_{C284} + P_{C285} Q_3 + P_{C340} Q_4) (P_{C283} + P_{C4} Q_4 + P_{C282} (Q_3 + Q_4)) + (P_{C289} -P_{C290} Q_3 -P_{C342} Q_4) (P_{C288} -P_{C3} Q_4 -P_{C287} (Q_3 + Q_4)) + (P_{C297} + P_{C298} Q_6 + P_{C341} Q_7) (P_{C296} -P_{C52} Q_7 + P_{C295} (Q_6 + Q_7)) + (P_{C302} -P_{C303} Q_6 -P_{C343} Q_7) (P_{C301} + P_{C51} Q_7 -P_{C300} (Q_6 + Q_7)) + (P_{C344} + P_{C217} (Q_3 + Q_4)) (P_{C365} + P_{C281} (Q_3 + Q_4)) -(P_{C345} -P_{C219} (Q_6 + Q_7)) (-P_{C367} + P_{C294} (Q_6 + Q_7))$$

$$M_{3,1} = P_{C293} + P_{C286} Q_4 -P_{C292} (Q_2 + Q_3 + Q_4)$$

$$M_{1,3} = M_{3,1}$$

$$M_{3,2} = P_{C373} + P_{C374} (Q_3 + Q_4) + (P_{C293} + P_{C286} Q_4) (P_{C283} + P_{C4} Q_4 + P_{C282} (Q_3 + Q_4)) + (P_{C292} -P_{C291} Q_4) (P_{C288} -P_{C3} Q_4 -P_{C287} (Q_3 + Q_4))$$

$$M_{2,3} = M_{3,2}$$

$$M_{3,3} = P_{C375} + (P_{C246} + P_{C4} Q_4) (P_{C293} + P_{C286} Q_4) + (P_{C248} -P_{C3} Q_4) (P_{C292} -P_{C291} Q_4)$$

$$M_{4,1} = -(P_{C306} + P_{C299} Q_7 -P_{C305} (Q_2 + Q_6 + Q_7))$$

$$M_{1,4} = M_{4,1}$$

$$M_{4,2} = -(-P_{C376} + P_{C377} (Q_6 + Q_7) + (P_{C306} + P_{C299} Q_7) (P_{C296} -P_{C52} Q_7 + P_{C295} (Q_6 + Q_7)) + (P_{C305} -P_{C304} Q_7) (P_{C301} + P_{C51} Q_7 -P_{C300} (Q_6 + Q_7)))$$

$$M_{2,4} = M_{4,2}$$

$$M_{4,3} = 0$$

$M_{3,4} = 0$
$M_{4,4} = P_{C378} + (P_{C254} + P_{C52}\ Q_7)\ (P_{C306} + P_{C299}\ Q_7) + (P_{C256} - P_{C51}\ Q_7)\ (P_{C305} - P_{C304}\ Q_7)$

Set derivatives of extra state variables

$Q'_9 = D_{\mathrm{SMOV}}$
$Q'_{10} = D_{\mathrm{PMOV}}$

****

Above equations and corresponding information between the lines of *****'s have been written by AUTOSIM. The equations are in the form:

$$\begin{aligned}
&M_{1,1}\frac{dU_1}{dt}+M_{1,2}\frac{dU_2}{dt}+M_{1,3}\frac{dU_3}{dt}+M_{1,4}\frac{dU_4}{dt}=G_{FA1},\\
&M_{2,1}\frac{dU_1}{dt}+M_{2,2}\frac{dU_2}{dt}+M_{2,3}\frac{dU_3}{dt}+M_{2,4}\frac{dU_4}{dt}=G_{FA2},\\
&M_{3,1}\frac{dU_1}{dt}+M_{3,2}\frac{dU_2}{dt}+M_{3,3}\frac{dU_3}{dt}+M_{3,4}\frac{dU_4}{dt}=G_{FA3},\\
&M_{4,1}\frac{dU_1}{dt}+M_{4,2}\frac{dU_2}{dt}+M_{4,3}\frac{dU_3}{dt}+M_{4,4}\frac{dU_4}{dt}=G_{FA4},
\end{aligned}\tag{C.1}$$

which involve non-linear components.

Appendix D

Back-propagation for the Multi-Layer Neural Networks in Chapters 8

The architecture of the neuro-controller involving two multi-layer networks for body height control and roll control is shown in Fig. 8.2. The actuator velocities on the starboard and port sides are:

$$\dot{\boldsymbol{u}} = [\dot{u}_1 \; \dot{u}_2]^{\mathrm{T}} = \begin{bmatrix} a^{R1} + a^{R2} \\ -a^{R1} + a^{R2} \end{bmatrix}, \tag{D.1}$$

where a^{R1} and a^{R2} are the outputs of the height control and roll control networks, respectively (superscripts 1 and 2 denote the height control and roll control networks, respectively). The input-output relationship of each network is the same as the one which was described in Appendix A. Therefore, the partial derivative of each network output with respect to a weighting parameter also follows the description in Appendix A. For example, the partial derivatives of $\dot{\boldsymbol{u}}$ with respect to h th weight at the output layer of the height control network is:

$$\frac{\partial \dot{\boldsymbol{u}}}{\partial w^{R1}{}_{1,h}} = \begin{bmatrix} a^{H1}{}_h \\ -a^{H1}{}_h \end{bmatrix}, \tag{D.2}$$

where $a^{H1}{}_h$ is the output of h th element at the hidden layer of the height control network.

Solving $\partial F_{NN}(\boldsymbol{y},\boldsymbol{w})/\partial \boldsymbol{y}$, partial derivatives of $\dot{\boldsymbol{u}}$ with respect to $\boldsymbol{y}$ are given as:

$$\frac{\partial \dot{\boldsymbol{u}}}{\partial \boldsymbol{y}} = \begin{bmatrix} \dfrac{\partial a^{R1}}{\partial y_1} & \dfrac{\partial a^{R1}}{\partial y_2} & \dfrac{\partial a^{R2}}{\partial y_3} & \dfrac{\partial a^{R2}}{\partial y_4} \\ -\dfrac{\partial a^{R1}}{\partial y_1} & -\dfrac{\partial a^{R1}}{\partial y_2} & \dfrac{\partial a^{R2}}{\partial y_3} & \dfrac{\partial a^{R2}}{\partial y_4} \end{bmatrix}, \tag{D.3}$$

where each element in the above matrix follows the description in Appendix A.

Appendix E

Neural Network Off-Line Training to Mimic the Relevant Input-Output Relationships of the P + D Controllers

This appendix describes how to mimic the relevant input-output relationships of the P + D controllers using neural networks.

Firstly, the P + D controllers (i.e. the height and roll controllers in chapter 7) run in the system (see Fig. 7.11) and an input-output relationship of each controller was sampled in a look-up data table. The form of the look-up data table involving proportional and differential inputs and a control output is illustrated in Fig. E.1.

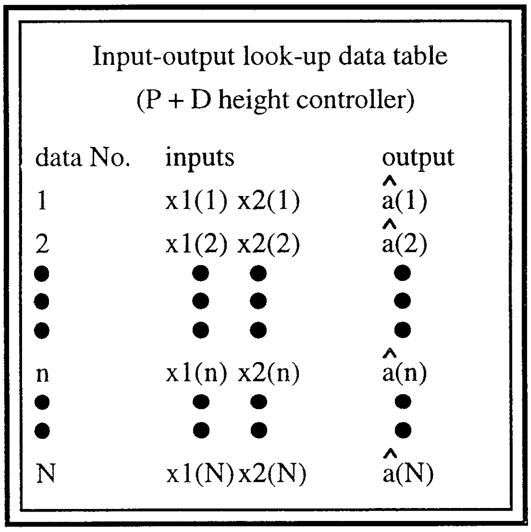
Secondly, each of two neural networks was trained, off-line, to mimic each of the two look-up data tables in a learning structure depicted in Fig. E.2. Each training proceeded in order to minimise the error function, J_e:

$$J_e = \frac{1}{N}\sum_{n=1}^{N} e^R(n), \tag{E.1}$$

$$e^R(n) = \frac{1}{2}(\hat{a}^R(n) - a^R(n))^2, \tag{E.2}$$

where a^R and $\hat{a}^R$ are the outputs of network and look-up data table, respectively. The reduction of the error function is shown in Fig. E.3, demonstrating how the neural network mimics the data of the height controller.

Finally, the trained two neural networks were integrated with the variable geometry suspension system and the system run under external lateral force inputs involving the maximum lateral acceleration of 0.8 G. The vehicle responses of the neural-control are compared with those of the P + D control system in Fig. E.4 showing how accurately the neural networks mimic the system.



Input-output look-up data table
(P + D height controller)

data No.	inputs		output
1	x1(1)	x2(1)	$\hat{a}(1)$
2	x1(2)	x2(2)	$\hat{a}(2)$
●	●	●	●
●	●	●	●
●	●	●	●
n	x1(n)	x2(n)	$\hat{a}(n)$
●	●	●	●
●	●	●	●
N	x1(N)	x2(N)	$\hat{a}(N)$

Fig. E.1 The input-output look-up data table

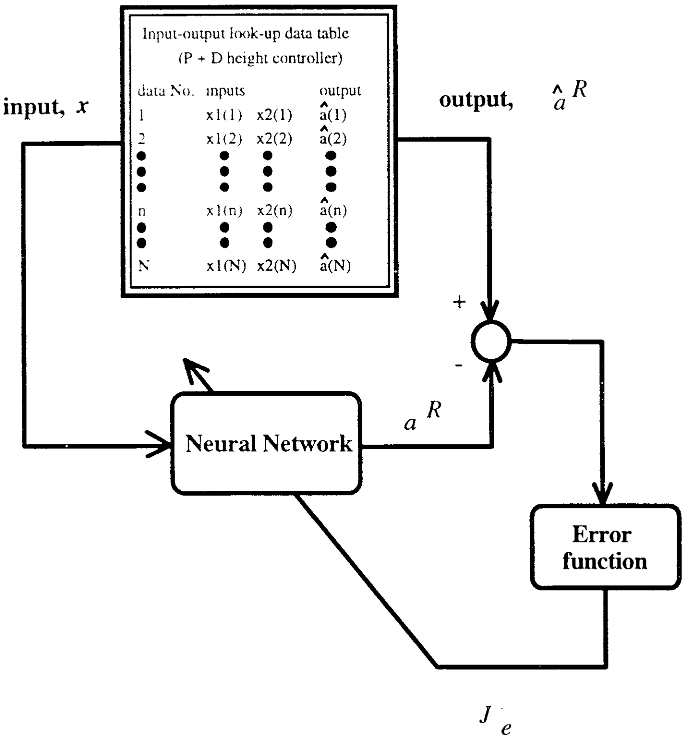

Input-output look-up data table
(P + D height controller)
data No. inputs output
1 x1(1) x2(1) $\hat{a}(1)$
2 x1(2) x2(2) $\hat{a}(2)$
n x1(n) x2(n) $\hat{a}(n)$
N x1(N) x2(N) $\hat{a}(N)$
input, x
output, $\hat{a}^R$
+
-
a^R
Neural Network
Error function
J_e

Fig. E.2 The off-line training structure using the look-up data table

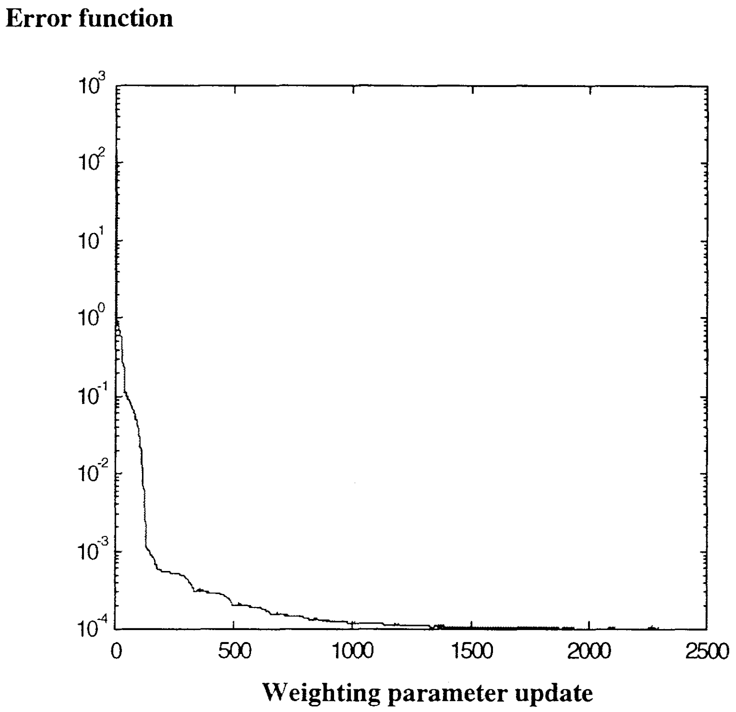
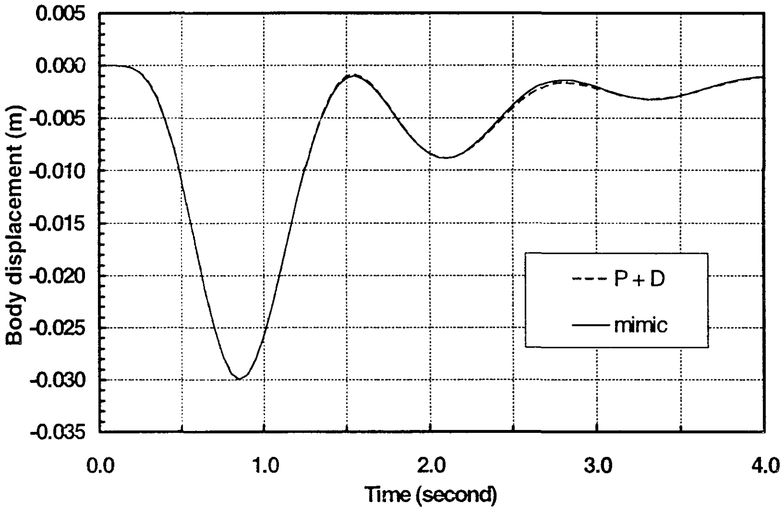
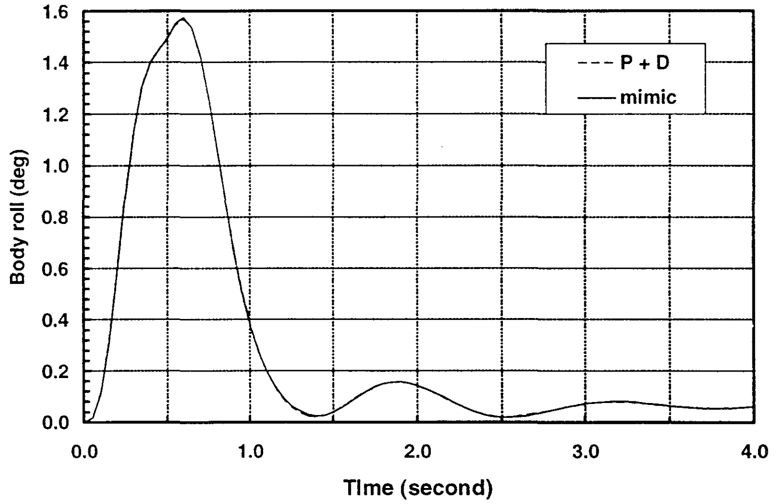


Fig. E.3 The descending error function, as learning proceeds for the data of the height controller (Training information: learning algorithm is back-propagation, 2422 weight update, learning rate is 0.39957 and final error is 1.00 × 10^{-4})



(a) Vehicle body height change response



(b) Vehicle body roll response

Fig. E.4 Comparison between the P + D control and neural-control (off-line mimic) in vehicle body responses: (a) height change and (b) body roll; during 0.8 G cornering (vehicle body mass : 665 kg)