

GENETIC LEARNING AUTOMATA AND FUZZY CLASSIFIER SYSTEM APPLIED TO ACTIVE CHASSIS SYSTEM

Felipe V. Brandão
Department of Automotive Engineering - Loughborough University
Ashby Road, Loughborough, Leicestershire, LE11 – 3TU, England, UK
F.Brandao@lboro.ac.uk

Mark N. Howell
Department of Automotive Engineering - Loughborough University
Ashby Road, Loughborough, Leicestershire, LE11 – 3TU, England, UK

Abstract. *An application in the automotive filed for the Genetic Learning Automata with Fuzzy Classifier System is presented in this work. As a non-linear model free based strategy, the major advantages of this approach are its modularity and its extensibility. A controller designed using this method for a simple longitudinal vehicle dynamic model is applied to a non-linear model with 107 degrees of freedom giving a reasonable performance. Comparisons with a conventional controller are also carried out.*

Keywords. *Fuzzy Classifier, Genetic Learning, Learning Automata, ABS, Active Chassis System*

1. Introduction

Active chassis systems have attracted the attention of many researches over the past twenty years with over 200 paper published in this field. Some researches have examined different vehicle models, with the optimisation and tuning of the control parameters for a specific case (Marsh, 1995; Brandao, 1999 and Brennan, 2001). Evolutionary algorithms have also been applied for this purpose as Li *et al*. (2000) and Maclay *et al.* (1993).

One of the most important problems that the researchers nowadays have to face is that the controller is usually designed for a specific vehicle, moreover for a specific model of a certain car. If any Original Equipment Manufacture decides to apply the controller on a different vehicle, it may have to be redesigned for the specific application. Therefore, what this work focuses on is a model free approach to design a controller for the Anti-lock Brake System (ABS), which is robust enough to support parameter changes. In this way, the controller can be applied to a wide range of vehicle with a reasonable performance and few tuning time.

In this paper, a new method to approach this problem is presented. Basically, it uses an evolutionary computing approach, the Genetic Learning Automata (GLA), to determine a Fuzzy Logic Controller (FLC) for an ABS. The FLC is a model free control approach and the GLA is used as a search engine to find the best set of solutions that matches the design objectives. Furthermore, FLC has been applied with considerable success in consumer product and industrial systems as is described in the literature (Cordon, 2001; Austin, 2000; Layne, 1993). It is suitable for a non-linear control system design and it is easy to incorporate the engineering knowledge (Jantzen, 1998). Moreover, the practical motivations for its use are that it is extremely easy to understand since it emulates a human control strategy. The hardware implementation is quick and easy and finally, the development is cheap.

2. Dynamic Model

There were mainly two vehicle models which were used in this work. The first one was a simple 2 degrees of freedom (d.o.f.) which just simulates the longitudinal dynamics, and the other one was an Adams model which represents the real vehicle.

Initially, Fig. (1) shows the diagram of the simple model having as d.o.f., the forward velocity and wheel spin velocity. The bellow equation represents its dynamics, where M is body mass, J is the wheel rotational inertia moment, T_b and T_d are the brake and drive torque at the wheel, F_x^{tyre} is the longitudinal tyre force, R_{eff} is the effective rolling radius, U is the forward vehicle velocity and finally, ω is the wheel spin velocity. The vehicle parameters, which are shown in Tab. (1), represent the simplification of the real data. Moreover, the R_{eff} is assumed constant in order to speed up the simulations.

$$\begin{cases} M\ \dot{U} = F_x^{tyre} \\ J\ \dot{\omega} = T_b - T_d - F_x^{tyre}\ R_{eff} \end{cases} \tag{1}$$

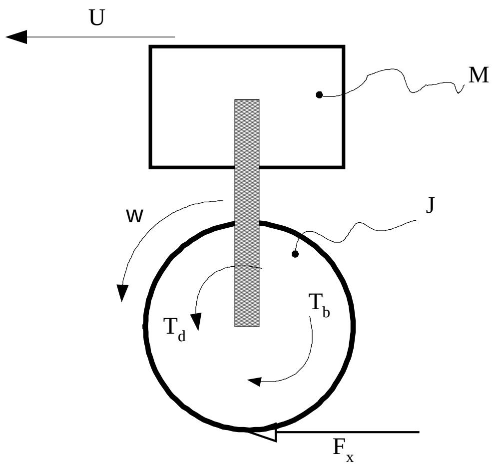

Figure 1. Vehicle mode – 2 d.o.f. .

Table 1 – Vehicle Parameters.

Body mass	M	1500 kg
Wheel rotational inertia moment	J	0.5 kgm^2
Effective rolling radius	R_{eff}	0.3 m

For its simplicity and less computational expense, the tyre model presented by Athan *et al.* (1997) was adopted in this work. Only the longitudinal dynamics was contemplated by this model which direct relates the vertical force of the tyre with its longitudinal force, Eq. (2).

$$F_x^{tyre} = \mu_s(\sigma) \times F_z^{tyre} \tag{2}$$

Then the slip ratio of the wheel for the longitudinal dynamic, σ, is given by Eq. (3), where u_x is the forward speed of the wheel.

$$\sigma = \frac{R_{eff}\ \omega}{u_x} - 1 \tag{3}$$

The following equations show coefficient of friction between the road and tyre as a non-linear function of the longitudinal slip ratio. For the dry surface, Eq. (4), the peak adhesion is assumed 0.9 at 22% of slip, with slide adhesion of 0.729. Then, the wet pavement, Eq. (5), has the peak friction is assumed 0.47 at 8% slip, with slide coefficient of 0.21. Finally, in the iced ground, Eq. (6), the maximum friction is 0.17 at 15% slip and slide of 0.13.

$$\mu = 0.9 \times (1.07 \times (1 - \exp(-17.73\sigma)) - 0.26\sigma) \tag{4}$$

$$\mu = 0.47 \times (1.07 \times (1 - \exp(-77.3\sigma)) - 0.6\sigma) \tag{5}$$

$$\mu = 0.17 \times (1.07 \times (1 - \exp(-38\sigma)) - 0.3\sigma) \tag{6}$$

Finally, a medium sport utility vehicle was modelled in Adams having overall weight around 1500 kg, distance between axles of 2.7 m, average semi-track of 0.70 m, front engine, and independent four wheel brake. This model had 107 d.o.f. and its diagram is shown on Fig.(2).

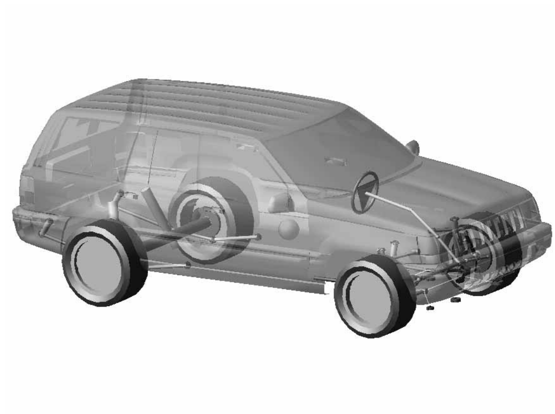

Figure 2. Adams' model.

3. Learning System

3.1. Genetic Learning Automata

The GLA is a synthesis of the Genetic Algorithm (GA) (Goldberg, 1989) and the learning automata (Najin, 1994). Howell *et al.* (2002) presents a detailed discussion of the algorithm. Brandão *et al.* (2001) have applied the GLA to solve the identification problem for vehicle suspension system.

The fitness function or cost function adopted here has the general form

$$f(x) = \sum_i \frac{C_i}{1 + 2\ e_i^{rms}} \tag{7}$$

Where e_i^{rms} is the least squared error of the variable e_i, which is one of the design variables. C_i is a weight constant used in order to pondered the importance of each design variable during the learning process.

By contrast to the GA, which does not have an inherent stop rule and so it is difficult to decide when to stop it, the GLA has an inherent stopping criteria. If all bit positions of the chromosome string for the whole population have converged, the system has found its maximum. Besides the inherent criteria, two other stopping criteria were introduced to speed up the process as the optimal solution was not a requirement for this problem.

Firstly, the process stopped when the ratio between the population average fitness and the fittest individual reached value greater than (1-ε), where ε was the desired precision.

$$\frac{\text{population average fitness}}{\text{fittest individual}} > 1 - \varepsilon \tag{8}$$

Secondly, if for ten consecutive generations the difference between the actual mean fitness and the previous mean fitness kept less than a desired limit ξ, the algorithm assumed that the solution for that specific learning process could not be improved anymore or the improvement was too small and halt.

$$\text{mean fitness(i) - mean fitness(i-1)} < \xi \tag{9}$$

3.2. Fuzzy Logic System

The Fuzzy Logic System (FLS) adopted in this work to design the Fuzzy Logic Controller (FLC) for the ABS was based on the Mandani Fuzzy Rule-Based System. The basic structure of this system is shown on Fig. (3). The system implemented had four main parts, the Fuzzification Interface, the Inference System, the Knowledge Base (KB) and the Deffuzzification Interface. The initial phase was the Fuzzification Interface which was responsible for the mapping between the real input and the fuzzy sets defined on the standard fuzzy universe and also contained the input scaling factor. The final stage was the Deffuzzification Interface, which had a scaling factor as well, was in charge for the mapping between the fuzzy universe and the real output domain. The KB stored the available knowledge about the problem in a very intuitive way, as IF-THEN rules, and was divided into two parts, the Data Base (BD) and the Rule Base (RB). Finally, the Inference System was responsible for averaging the latter rules put into effect by the KB. Plenty of different deffuzification methods are available. The process of transforming a fuzzy output of a fuzzy inference system into a crisp output applied in this work was the Bisector of area (BOA). This method picks the abscissa of the vertical line that divides the area under the curve in two equal halves.

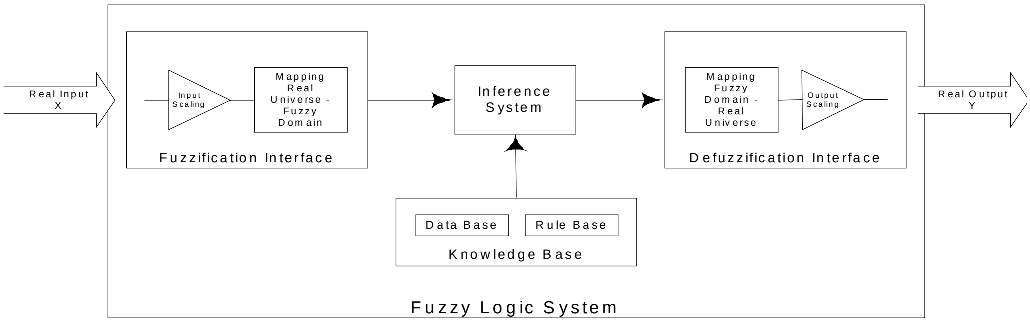

Figure 3. Scheme of the Fuzzy Logic System.

3.3. Classifier System

A Classifier System (CS) is defined by Goldberg (1989) as a machine learn system which learns rules in order to guide its performance in an arbitrary environment. Any CS is basically compound by three main components - rule and message system, reward system and rule generation system.

The characteristic that distinguishes the CS from other learning mechanisms is the capability to adapt their heuristic to changes in demand. A CS can develop trending strategies in a stock management system, in a chemical plant it can perform process control (Geyer-Schulz, 1995), and in a vehicle could be used to integrate several intelligent systems, such as anti-lock braking system and traction control system.

A learning CS is a form of machine learning which dispenses with a human expert and attempts to evolve a meaningful rule base via environmental feedback and the recombination of existing rules to form new improved rules. Like an expert system, all knowledge in a learning CS is coded as production IF-THEN rules. Thus, it is an expert system without an expert.

There are mainly two approaches for this genetics-based machine learning (Goldberg, 1989). The first one, namely the Pittsburgh approach, is concentrated on rule bases and needs only a genetic algorithm as learning component. Each individual represents a rule base which is evaluated in a simulated environment. The genetic algorithm solves the rule base discovery problem by generating a new set of rules bases for the next generation. The second one, namely the Michigan approach, is the rule learning family and the Holland classifier system is one example of this family. This group has to solve two problems, the first is the apportionment of credits which is the problem of reinforcement, and the other one is to discover new useful rules when the existing rules prove inadequate. The Pittsburgh approach was taken on to be used in this work.

3.4. Design Methodology

The Genetic Learning Automata Fuzzy Classifier System (GLAFCS) employed the global search capability of the GLA to determine FCS rule and data bases that can be intuitively understood by vehicle chassis engineers. Considering the scheme presented on Fig. (3), the learning procedure was divided into four parts. Initially the scale factors were set based on the vehicle model and the expected values for the input variables. Secondly, making use of standard membership functions, a set of rules for the controller was learnt using GLA. Next, the scaling factors (linear and non-linear) which had been set previously were fine-tuned. Finally, a training section took place in order to verify if the final controller had the desired behaviour.

4. Controller Design

In this project, the improvement of the vehicle's stopping distance for different kinds of surfaces was the requirement for the design controller. In order to archive this objective, the controller should avoid the wheel lock-up. The available variables were the wheel speed, ω, and an estimate forward vehicle speed, *U*.

The whole learning process took place for the vehicle model with 2 d.o.f. when was learnt the KB for the ABS. Based on the input variables, the wheel slip was estimated using Eq. (3). Figure (4) shows the diagram of the controller structure where the wheel slip estimator block is placed just before the fuzzy set. This block has as outputs the wheel slip and its discrete-time change of error.

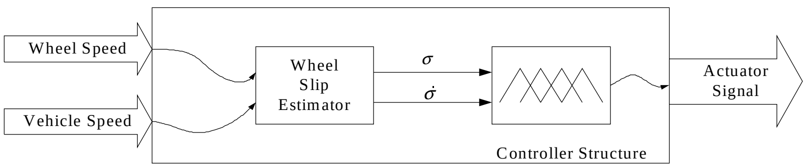

Figure 4. Standard modules for ABS fuzzy controller.

The input signals for the fuzzy controller are defined in Eq. (10) and Eq. (11), where $G_{in,i}$ and $P_{in,i}$ are defined in the next sub-section. Equation (12) shows the output signals, S_{out}, as a function of controller output, S. The gains G_{out} and P_{out} are also defined on the next sub-section.

$$\sigma_{in} = G_{in,1} \times sign(\sigma) \times \sigma^{P_{in,1}} \tag{10}$$

$$\dot{\sigma}_{in} = G_{in,2} \times sign(\dot{\sigma}) \times \dot{\sigma}^{P_{in,2}} \tag{11}$$

$$S_{out} = G_{out} \times sign(S) \times S^{P_{out}} \tag{12}$$

During the learning process, the GLA was applied to find the set of rules used on FCS and tune all scaling factors for the inputs and outputs of the controllers. The following sub-sections detail each learning step.

4.1. Initial Scalling Factor

Defining the scaling factor was the first step in the learning process. Initially those factors were defined based on engineering knowledge about the problem. All the input gains were set to one and the output signal from the controller, which was normalised, had to be multiplied by a gain in order to get the correct desired force. Therefore, the gains used are

$G_{in,1}$ = 1
$P_{in,1}$ = 1
$G_{in,2}$ = 1
$P_{in,2}$ = 1
G_{out} = 1/0.8
P_{out} = 1

Two saturation functions were applied due to limit the controllers inputs within the fuzzy controller universe. Another one with the limits between –1 and +1 was placed just after the controller output owing to the actuator input signal limitations.

4.2. Learning Rules

The learning rule process was divided into four steps. Firstly, a set of standard Membership Functions (MF) for each controller input and output were defined, Fig. (5), and the genetic learning automata parameters were set to

Max of Generations = 2000
Population Size = 20
Crossover Rate = 0.9
Mutation Rate = 0.001
Learning Rate = 0.05
Keep Population = 0.20

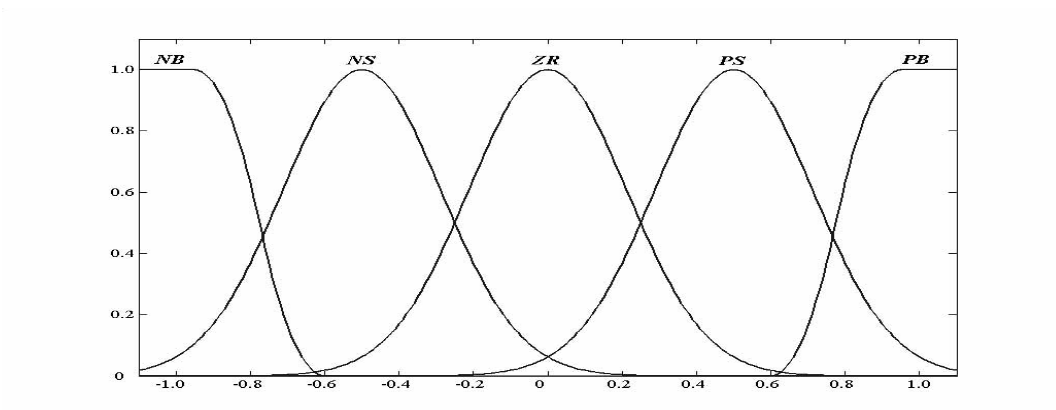

Figure 5. Standard Membership Function. (NB – negative big, NS –negative small, ZR – zero, PS –positive small, PB – positive big).

Secondly, based on engineering knowledge the first and last five rules were set to, respectively, PB and NB, as those rules were less likely to be activated. Then, the training set took place. After 2000 generations or the convergence is reached, the rules which remained activated for more than 10% of the total time were fixed and a new learning run was reinitiated. This procedure went on three times consecutively. After the third run, no additional improvement was verified. Therefore, this process was halted and the final set of rules which represents the KB of this problem is shown in Tab. (2). The bolded rules were adjusted based on engineering expertise, in order to have a smother response. They were changed from ZR to NB and from PS to PB. Both rules were activated less than 1% of the total simulation time. So, no significant learn was attributed to those rules.

Table 2 – Body Acceleration Rules.

		Change of error				
		NB	NS	ZR	PS	PB
Error	NB	PB	PB	PB	PB	PB
	NS	PB	PB	PS	PB	**PB**
	ZR	NB	NS	ZR	ZR	NB
	PS	**NB**	NS	NS	NB	NB
	PB	NB	NB	NB	NB	NB

4.3. Learning Scaling Factors

As far as the KB was learnt, the controller would be applied to any vehicle if the scaling factors (SF) were correctly adjusted. The SF was one of the most sensible parts of the Fuzzy System, thus it had to be carefully adjusted. In fact, it was the best way to fine-adjust the controller to a certain vehicle. The GLA with the same parameters from the previous sub-section was used to learn the gains and they are shown in the Tab. (3), where $\bar{x}$ is the average value and s is the standard variance. The output gains were not learnt owing to actuator characteristics dependence, $G_{out} = 0.8^{-1}$ and P_{out} = 1.

Table 3 – Scaling Factors.

Gain	$\bar{x} \pm s$
$G_{in,1}$	1.9870 ± 0.03230
$P_{in,1}$	1.0338 ± 0.09664
$G_{in,2}$	0.9095 ± 0.00749
$P_{in,2}$	0.1709 ± 0.00748

4.4. Expand to Adams' Model

The main advantage of the proposed structure was its modularity. As far as the controller for the simple vehicle model was designed, it could be applied to each wheel of the vehicle. An ideal scheme is shown on Fig. (6). As the GLAFCS is a model free approach, once the KB base was learnt, the modulus can be applied to any similar problem without significant performance degradation. Some adaptation is recommended, such as a fine tuning of scaling factor. Nevertheless, in this work, the controller learnt for the simplified model was applied straightforward to Adams' model following the scheme shown on Fig. (6). A comparison with another ABS controller will be carried out in Section 6, in order to verify whether this approach would lead to reasonable results. Actually, this approach was not searching for the optimal controller, but just for a very good one that can be implemented to a huge number of different vehicles.

5. Slide Mode Controller

A slide mode controller (Slotine, 1991) based on the error between the reference and actual wheel slip was designed. The reference slip was set to be the optimal slip for each simulation road. The slip error was defined by subtracting the actual slip from the desired slip and feed it into Eq. (13). So, the slide mode control would assume values equal to +1 or –1 depending on the signal of the error. A first order filter was placed just after the signal output so as to prevent excessive chattering. This controller was applied for both models, the 2 d.o.f. and Adams.

$$S_{out} = sign(\sigma_{error}) \tag{13}$$

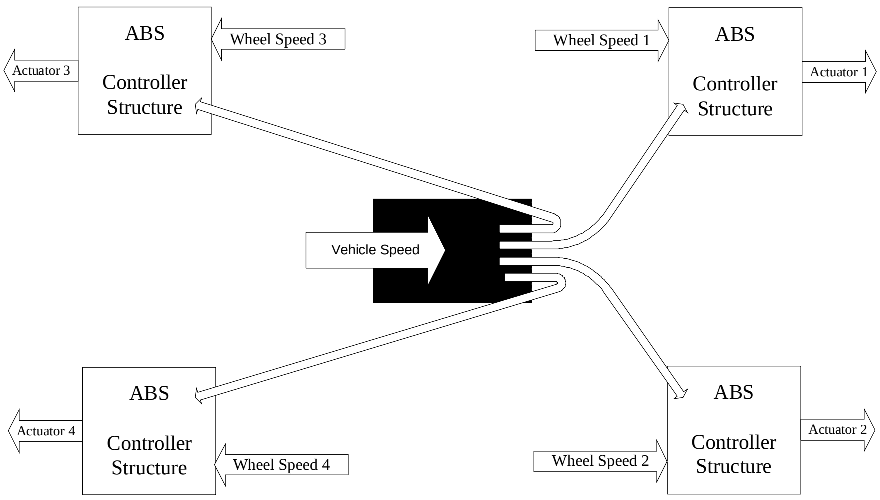

Figure 6. Ideal scheme for expanding.

6. Comparison

The comparison took place on two levels. The first one, the performance of the 2 d.o.f. vehicle model using the slide mode control was compared with the same model equipped with the fuzzy logic controller. The other analysis carried out was the robustness of the learnt controller. The controller learnt for the simple model was applied to the Adams' model.

As the main objective of the ABS was to reduce the stopping distance, and the actuator chattering was a non-desirable behaviour, the performance index was defined as function of the distance travelled during the braking and the power spectrum of the controller output, Eq. (14), where c_1 and c_2 are weighting constants for those effects. The manoeuvre adopted in this study was a full braking from 40 m/s to 10 m/s. The set of simulation was carried out on three different surfaces – dry, wet and ice.

$$P.I. = c_1 \times \int PSD(S_{out})\, dt \; + c_2 \times \text{brakingdistance} \qquad (14)$$

As can be seen on Tab. (4), the stopping brake distance for the slide mode controller is always smaller than the other one. However, the undesirable chattering behaviour is strongly presented with the slide controller, as shown on Fig. (7). Thus, the slide controller has a better performance concerning the braking distance, while generating a lot of chattering. The *P.I.* weights those effects and produces an overall value for each controller. The smaller the *P.I.*, better the controller. Indeed, the Fuzzy Logic controller has a slight better over all performance, as presented on Tab. (5).

Table 4 – Braking distance for the 2 d.o.f. model.

Road Surface	Slide Mode	Fuzzy Logic
Dry	87.56 m	87.60 m
Wet	176.62 m	183.13 m
Ice	452.45 m	467.45 m
Average	238.88 m	246.06 m

Table 5 – Performance Index for the 2 d.o.f. model.

Road Surface	Slide Mode	Fuzzy Logic
Dry	1.50	1.39
Wet	3.11	2.85
Ice	7.86	6.71
Average	4.16	3.65

Figure (8) shows the performance of the SUV modelled in Adams with and without ABS, on dry and wet roads. Figures (8.a) and (8.b) show the forward acceleration for the dry and wet surfaces, where is possible to verify in both cases that for the same brake pedal demand, the acceleration obtained using the ABS controller is higher than the one without the controller, as the controller tries to keep the vehicle within the optimal slip condition. The wheel lock-up avoidance is also obtained while using the ABS controller – Fig. (8.b) and (8.c), and the longitudinal wheel slip was kept within the target range while the ABS controller was running.

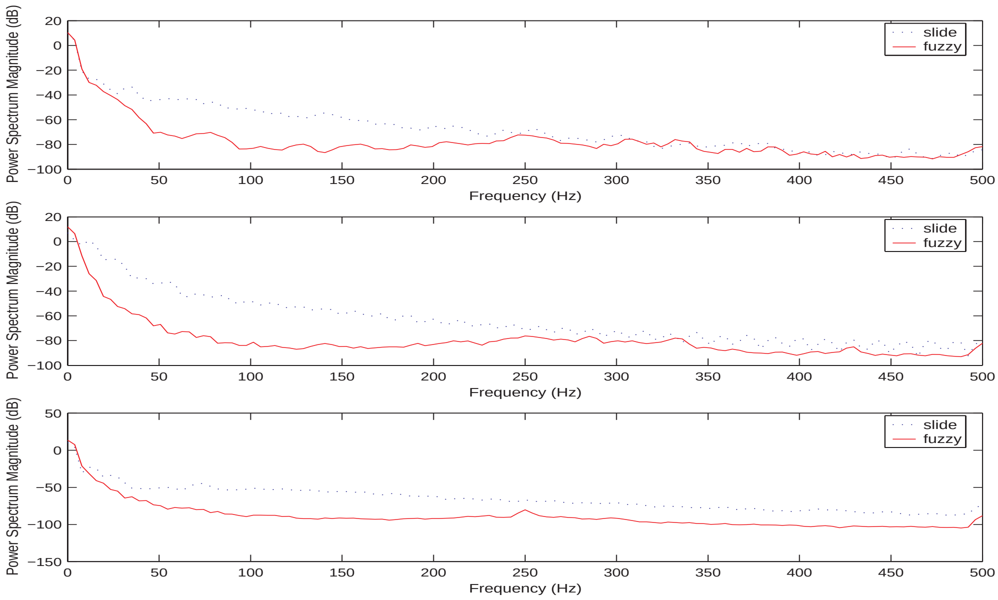

Figure 7. Power Spectrum for three different surfaces (dry – top, wet – middle, ice – bottom) and for two controllers (slide mode – dotted line, fuzzy logic – solid line).

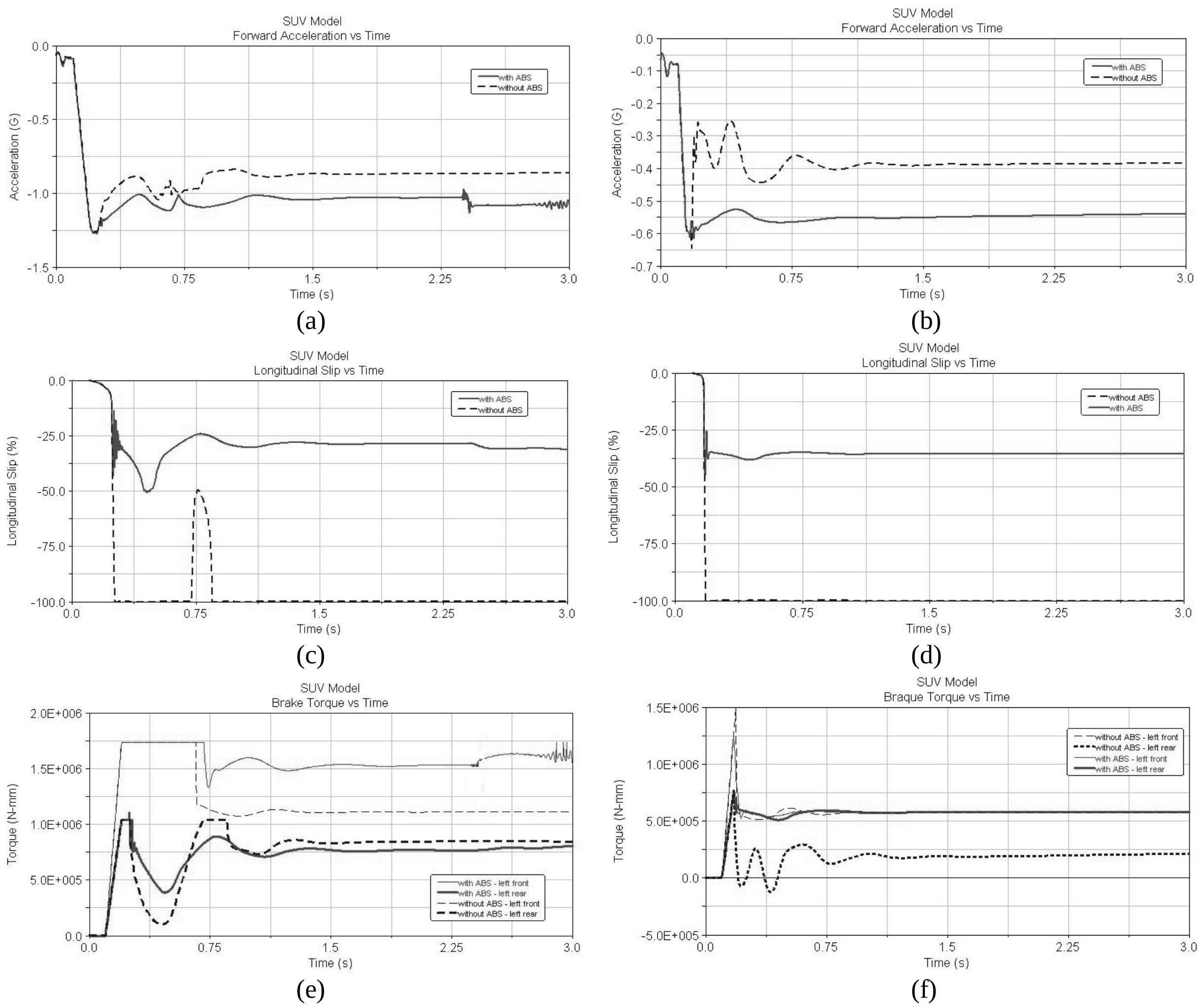

Figure 8. SUV with and without ABS on dry and wet roads. Where a, b and c are the forward acceleration, longitudinal slip and brake torque for dry surface. The other three graphs are for the wet surface. (solid – with ABS, dotted – without ABS).

The ABS controller acts directly on each servo-valves responsible for the modulation of the brake pressure for each individual wheel. As an illustration, the brake torques for the front and rear left brakes are shown in Fig. (8.d) and (8.e). On dry surface, the front brake torque with ABS is higher than the other configuration and the rear brake torque are almost the same. So, the over all brake torque is higher with ABS. The same behaviour occurs with the wet road simulation.

7. Conclusion

The aim of this work was to prove that a model free based strategy as FCS can be applied to automotive systems and has major advantage of modularity and extensibility. It was shown that a controller designed for a 2 d.o.f. model can work when applied to a 107 d.o.f. model with a reasonable performance. Nevertheless, fine tuning of the input gains is recommended in order to optimise the vehicle performance.

Some comparisons with the slide mode controller was carried out and the FLC had a very good performance, better than the other one, even though an optimal controller was not looked for. The main reason for this is that the fuzzy controller is a non-linear control with a smoother output, instead of the bang-bang controller which has a crisp output.

The same approach has been extended to others systems such as roll and yaw controllers, as well as to integration controller strategies.

8. References

Athan, T. W. and P. Y. Papalambros (1997). Multicriteria Optimization of Anti-lock Braking System Control Algorithms. http://helix.gatech.edu/Classes/ME2202/2000S3/Projects/Windley/abseqns.htm, Gordon and Breach Science: 1-25.

Austin, L. and D. Morreu (2000). “Recent Advances in Antilock Braking Systems and Traction Control Systems.” Journal of Automotive Engineering - IMechE Proceeding Part D Vol 214 No D6: 625-639.

Brandao, F. V. (1999). Influência da Suspensão na Dirigibilidade e Estabilidade de Veículos em Curva. Departamento de Engenharia Mecânica. Rio de Janeiro, Pontifícia Universidade Católica do Rio de Janeiro: 165.

Brandao, F. V. and M. Howell (2001). The Application of the Genetic Learning Automata to Parameter Identification of a Dynamic System. 5th IEEE International Conference of Intelligent Engineering Systems, Helsinki, Finland.

Brennan, S. and A. Alleyne (2001). Integrated vehicle control via coordinated steering and wheel torque inputs. Proceedings of the American Control Conference. 1: 7-12.

Cordon, O., F. Herrera, et al. (2001). Genetic Fuzzy Systems: Evolutionary Tunning and Learning of Fuzzy Knowledge Bases, World Scientific Printers (S) Pte Ltd.

Geyer-Schulz, A. (1995). “Holland Classifeir System.” ACM - Association of Computing Machinery(0-89791-722-7/95/0006): 43-55.

Goldberg, D. E. (1989). Genetic algorithms in search, optimisation and machine learning, Addison-Wesley Publishing Company, Inc.

Howell, M., T. J. Gordon, et al. (2002). “Genetic Learning Automata for Function Optimization.” IEEE Transactions on Control Systems, Man, and Cybernetics - Part B: Cybernetics 32, No. 6: 804-815.

Jantzen, J. (1998). Tutorial On Fuzzy Logic. Lyngby, Technical University Of Denmark, Department of Automation.

Layne, J. R. and K. M. Passino (1993). “Fuzzy Model Reference Learning Control for Cargo Ship Steering.” IEEE Control Systems December(0272-1708/93): 23-34.

Li, T.-H. S. and Y.-P. Kuo (2000). “Evolutionary Algorithm for Passive Suspension Systems.” JSME International Journal Series C, Vol. 43, No.3: 537-544.

Maclay, D. and R. Dorey (1993). “Applying Genetic Search Techniques to Drivetrain Modeling.” IEEE Control Systems(0272-1708/93): 50-55.

Marsh, C., T. J. Gordon, et al. (1995). “Application of Learning Automata to Controller Design in Slow-Active Automobile Suspensions.” Vehicle System Dynamics 24: 597-616.

Najin, K. and A. S. Pznyak (1994). Learning Automata - Theory and Applications, Pergamon.

Slotine, J.-J. E. and W. Li (1991). Applied nonlinear control. London, Prentice-Hall.