DNA coded GA for the Rule Base Optimization of a Fuzzy Logic Controller

Xiao Peng, Prahlad Vadakkepat and Tong Heng Lee
Department of Electrical and Computer Engineering
The National University of Singapore
10 Kent Ridge Crescent, Singapore 119260
{engp0525, prahlad, eleleeth}@nus.edu.sg

Abstract- A DNA coded genetic-algorithm (GA) is proposed to optimize the rule-base of a fuzzy logic controller (FLC). The controller is designed for a vehicle-active suspension system to improve the driving comfort. The DNA coded GA constructed optimal decision-making rules for the fuzzy logic controller. Simulation results demonstrated the effectiveness of the algorithm.

Keywords : DNA, Genetic Algorithm, DNA Coded GA, Fuzzy Logic Controller and Active Suspension System.

1 Introduction

In 1994, Leonard Adleman [1] proposed the idea of DNA Computing and that it could be very helpful in completing previously unsolvable problems. Theoretical work has shown that DNA computing is capable of universal computation. Although simple problems take longer to solve with DNA computing than with supercomputers, more complex problems can be solved faster [2, 3]. DNA computing is very quick, as it can perform many calculations *simultaneously* or *in parallel* [4, 5]. Since DNA computing is such a new idea, the applications for which it may be used in the future remain unknown. It is speculated that DNA computing will be useful in the fields of computer science, biology, chemistry and medicine [6, 7, 8]. The possibilities are high, as DNA computing may allow mathematicians to solve problems that have been unsolvable for years.

A DNA (deoxyribonucleic acid) coded genetic algorithm (GA) is proposed in the work, to optimize the rule table of a fuzzy logic controller (FLC), which is designed for an active suspension system. In the proposed algorithm, every individual is coded as a DNA chromosome.

Active suspension control can improve the ride comfort and holding ability of automobiles under varying road conditions. A good suspension system has to reduce the sprung mass acceleration and should provide adequate suspension deflection to maintain tire-ground contact at the same time [9]. Design methods used in active suspension systems (ASS) are normally based on optimal control strategies [10, 11, 12, 13]. In those methods, suspension systems are optimized with respect to sprung mass acceleration, suspension deflection, and tire deflection. However, when perturbed conditions occur, the fixed optimal state feedback gains may not work well. The fuzzy logic controller has been introduced into the design of active suspension systems, as another method, to tackle the situation.

The kernel of the FLC is a set of linguistic control rules, which can capture human thinking. The rule base determines the control decisions for the process and plays a key role in the FLC. The conventional fuzzy control rules are established by the knowledge and experience of expertise and/or skilled operators. In other words, the rule table is human dependent and may not be an optimal one.

Genetic Algorithms (GAs) are optimum searching algorithms that are based on concepts of natural selections and natural genetics. Studies [14, 15, 16, 17] have showed that genetic algorithms have the ability to optimize the rule table of conventional FLC. With the powerful search capacity of GAs, the FLC design for an active suspension system could provide a set of suboptimal control rules.

The performance of GA is closely associated with the population size. The parallel computation potential and high information density of DNA computing may enable to handle a population size that is too big for the conventional computers. It may then greatly increase the performance of GAs. As a result, genetic algorithms are likely to provide a promising line of research in DNA computing.

The basics of DNA computing is discussed in Section 2. Description of the active suspension system is provided in Section 3. The basic building blocks of a fuzzy logic controller (FLC) and the block diagram of the designed FLC for the active suspension system are described in Section 4. Section 5 deals with the DNA coded GA algorithm. The coding method and the correspondence between the amino acids and the linguistic variable of the FLC are discussed. Simulation results are included in Section 6, followed by discussions in Section 7.

2 DNA Computing

The idea of DNA computing is to use strands of DNA to encode a problem, and to manipulate them using techniques commonly available in any molecular laboratory, in order to simulate operations which can generate solution.

The name of DNA computing should not be confused with biocomputing. Usually, biocomputing refers to everything that the computer scientists can do, to help the biologists in the study of genes. For instance, algorithms and data structures have been developed to investigate the properties of the sequences of nucleotides in DNA or RNA (Ribonucleic Acid). In DNA computing, instead, molecular biology is suggested to solve the problems that computer scientists face. On the other hand, DNA computing is not similar to Genetic Al-

0-7803-6657-3/01/$10.00 © 2001 IEEE

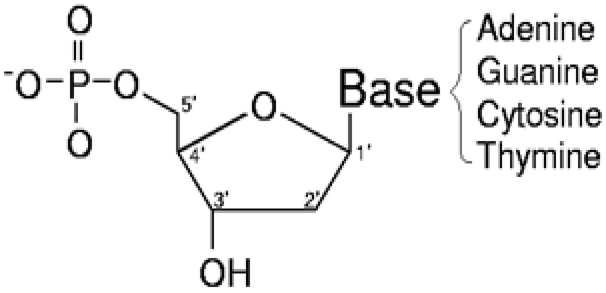

Figure 1: Chemical structure of a nucleotide.

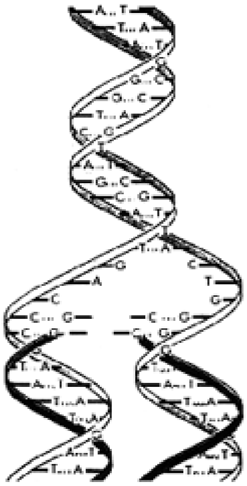

Figure 2: Autoreplication of DNA.

gorithm. GA simulates the rules of the nature of evolution in computation. In this way, GA searches for the optimal solution of a problem. As to DNA computing, it dose not simulate anything in molecular biology, but actually uses DNA strands to perform the computation.

There are four nucleotides that compose a strand of DNA: adenine (A), guanine (G), cytosine (C) and thymine (T). They are also called as bases. The chemical structure of DNA, the famous double helix, consists of a particular bond of two linear sequences of bases. This bond follows a property of complementarity: adenine bonds with thymine and vice versa; cytosine bonds with guanine and vice versa. This is known as the *Watson-Crick* complementarity, denoted as: $\overline{A} = T$; $\overline{T} = A$; $\overline{C} = G$; $\overline{G} = C$. This complementarity enables us to store information in DNA and thus perform the computations.

The chemical structure of a single nucleotide is shown in Figure 1. Single nucleotides are linked together, end to end, to form the DNA strands in a process called polymerization. This linking occurs via the reaction between the $5'$ phosphate of one nucleotide and the $3'$ hydroxyl of another. Every DNA strand will have two distinct ends — one with a free $5'$ PO$_4$ group and the other with a free $3'$ OH group, referred to as the $5'$ and $3'$ ends, respectively. The $3'$ and the $5'$ ends determine the strand's polarity. Only two complementary strands of opposite polarity (also known as, in *antiparallel* fashion) can bond together to form the final double helix (Figure 2). Figure 2 also shows how DNA autoreplicates itself.

3 Active suspension system

A typical active suspension system (ASS) is shown in Figure 3. The system is presented by a two-degree-of-freedom model. In this model, the system is characterized by the spring coefficient k_p (N/m), damping rate b_p (N/m/s) and the sprung load mass m_p(kg). u represents the applied control force. z_p (m) and w (m) are the sprung mass and wheel displacement respectively. The dynamic equation of the active suspension system is approximated by Equation 1:

$$m_p \ddot{z}_p = -k_p(z_p - w) - b_p(\dot{z}_p - \dot{w}) + u \tag{1}$$

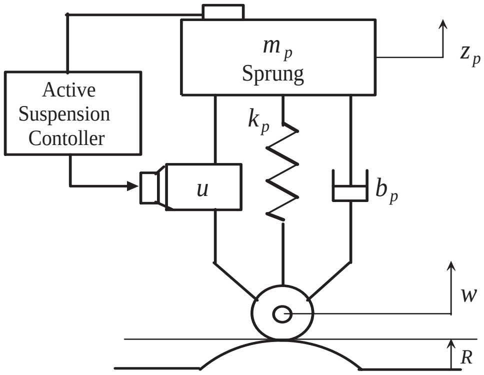

Figure 3: Model of an active suspension system

The deflection in the system is denoted by Equation 2.

$$x_p = z_p - w \tag{2}$$

Based on Equations 1 and 2, the transfer functions between the wheel displacement and, the sprung mass displacement and suspension deflection of the system are obtained.

With the applied force u set to zero:

$$z_{pw}(s) = \frac{b_p s + k_p}{m_p s^2 + b_p s + k_p} w(s) \tag{3}$$

$$x_{pw}(s) = \frac{-m_p s^2}{m_p s^2 + b_p s + k_p} w(s) \tag{4}$$

With zero wheel displacement (w set to zero):

$$z_{pu}(s) = \frac{1}{m_p s^2 + b_p s + k_p} u(s) \tag{5}$$

$$x_{pu}(s) = \frac{1}{m_p s^2 + b_p s + k_p} u(s) \tag{6}$$

From Equations 3 – 6, the equations for sprung mass displacement z_p and suspension deflection x_p when both u and w are present, can be derived as:

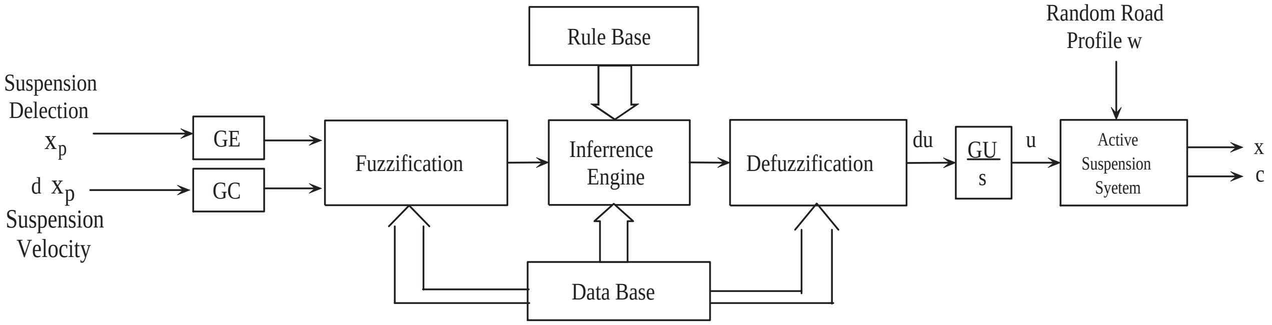

Figure 4: Structure of the fuzzy controlled active suspension system.

$$z_p(s) = \frac{b_p s + k_p}{m_p s^2 + b_p s + k_p} w(s) + \frac{1}{m_p s^2 + b_p s + k_p} u(s) \tag{7}$$

$$x_p(s) = \frac{-m_p s^2}{m_p s^2 + b_p s + k_p} w(s) + \frac{1}{m_p s^2 + b_p s + k_p} u(s) \tag{8}$$

4 Fuzzy logic controller

Unlike boolean logic, fuzzy logic can deal with uncertain and imprecise situations. Linguistic variables (SMALL, MEDIUM, LARGE, etc.) are used to represent the domain knowledge, with their membership values lying between 0 and 1. Basically, a fuzzy logic controller consists of the following components [18].

- A fuzzification interface, to scale and map the measured variables to suitable linguistic variables.
- A knowledge base, comprising linguistic control rule base.
- A decision making logic, to infer the fuzzy logic control action(s) based on the measured variables, which is much akin to the human decision making.
- A defuzzification interface, to scale and map the linguistic control actions inferred, to yield a nonfuzzy control input to the plant/process being controlled.

The structure of the fuzzy logic control system for the vehicle suspension system is shown in Figure 4. The suspension deflection x_p and the suspension velocity dx_p are the inputs variables to the fuzzy logic controller while the change in control du is its output. Bell-shaped membership functions are used to represent the different linguistic variables for x_p, dx_p and du.

Scaling factors GE and GC are used for the suspension deflection and velocity to appropriately map them to the respective universes of discourses. Meanwhile, the final control signal acting on the suspension is,

$$u(k) = \mathrm{GU} du(k) \tag{9}$$

with GU as the scaling factor for du. Obviously, this kind of a fuzzy logic controller implementation falls into the family of proportional-derivative type.

5 DNA coded Genetic Algorithm

DNA computing is a newly developed computation method that has generated great interests among researchers in computer science and biology. A typical test tube can contain a large number of DNA molecules, and biological operations can be performed with them simultaneously. Due to this, any method of computation based on DNA would seem to have potential massive parallelism and power.

GAs are powerful search algorithms based on the mechanism of natural selection and natural genetics, that operate without any knowledge of the task domain and utilize only the fitness of evaluated individuals. The combination of GA and fuzzy logic has already proved as a promising tool for knowledge discovery.

The population size is an important factor which affects the performance of GAs. Bigger population sizes always lead to better results. An increase in the population size means an increase in the diversity of population, hence, the lower possibility of being trapped into premature convergence. However, on conventional computers, with the increase of population size, the computing time will increase greatly too. A tradeoff between the size and computing time is the way out. On the other hand, with the application of DNA computing to GAs, the computation time will not explode with the increase in population size. As a consequence, it is feasible to work with GAs with a huge population size that has never been considered before.

In this paper, a DNA coding method to encode the rule base of FLC is proposed and then the DNA coded GA is used to optimize the controller. The DNA coding is the first and important step to apply the DNA computing in GAs.

As we know, the messenger Rribonucleic acid (mRNA) can be synthesized from the DNA. In the mRNA, three successive bases called codons are allocated sequentially. These

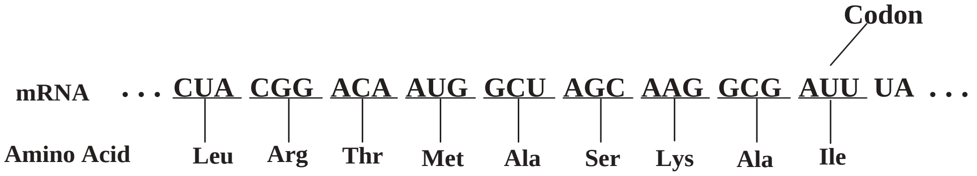

Figure 5: Structure of the fuzzy controlled active suspension system.

Phe	*Leu*	*Ile*	*Val*	*Ser*	*Pro*	*Thr*	*Ala*
NL		NM		NS		Z	
Tyr	*His*	*Cln*	*Asn*	*Lys*	*Asp*		
PS		PM		PL			

Table 1: Correspondence between amino acids and linguistic variables

codons are the codes for amino acids. Sixty four (64) kinds of codons correspond to 20 kinds of amino acids, as shown in Figure 5. The Leu, Arg, Thr, etc., are the abbreviations for the amino acids. For example, Leu stands for leucine, Arg for arginine, Thr for threonine, respectively. Amino acids can be synthesized artificially. The meaning of each amino acid can be defined the way it is desired; such as a variable or a form of a function. In this way, the problem under consideration can be coded as a DNA chromosome.

As to the rule base of FLC, the coding is relatively simple. Table 2 shows the rule base of a FLC for the active suspension system [19]. There are seven linguistic variables to represent the control force: NL, NM, NS, Z, PS, PM and PL. Some of the amino acids are mapped to these variables. The mapping relationship is displayed in Table 1.

There are 49 rules in Table 2. The table has some kind of a symmetry, resulting in part of the rules to be complementary to others. The complementary pairs are:

$$NL \longleftrightarrow PL, NM \longleftrightarrow PM, NS \longleftrightarrow PS, Z \longleftrightarrow Z$$

As a result, it is required to code only 25 rules representing the set of 49 rules. For the rule base in Table 2, the corresponding DNA chromosome may be denoted as:

$$PheLeuIleValValSerThrPheTleSerProSerAla...$$
$$...HisValSerThrThrThrHisHisSerThrThrThr$$

As there are more than one amino acid corresponding to one linguistic variable, the DNA chromosome representing the above rule base may not be unique.

The GA operations on the DNA chromosome can be similar to those of the standard GA. In the crossover operation, the chromosomes are cut first and parts of the parents are exchanged. The offspring are formed as in Figure 6. With the help of enzymes (such as restriction enzyme and ligase), these operations can be carried out in a biological lab.

		Suspension deflection						
		NL	NM	NS	Z	PS	PM	PL
Sprung mass velocity	NL	NL	NL	NM	NS	NS	NS	Z
	NM	NL	NM	NS	Z	NS	Z	PS
	NS	NM	NS	Z	Z	Z	PS	PM
	Z	NM	NS	Z	Z	Z	PS	PM
	PS	NM	NS	Z	Z	Z	PS	PM
	PM	NS	Z	PS	Z	PS	PM	PL
	PL	Z	PS	PS	PS	PM	PL	PL

Table 2: Rule base for fuzzy logic controller

As for mutation, if mutation happens in one position, that position of RNA is replaced with another RNA with a different meaning. If there is a Phe to mutate, another RNA from Table 1 is used to replace the same besides the Leu.

6 Simulation Results

For the active suspension system discussed in Section 3, the parameters considered are [20]: $m_p = 355$ kg, $k_p = 14383$ N/m and $b_p = 1860$ N/m/s, typical of a commercial vehicle. A sampling time of 0.01 is selected. The time domain response of the vehicle is simulated for a time period of 5s over a pseudo-random road profile, with natural frequencies 1 and 2 Hz, which is a worst case scenario. The corresponding pseudo-random road profile is given in Figure 7(a).

To compare the response of the controller with the (original) FLC in [19], the associated universes of discourses and the scaling factors of the FLC are left unchanged. The universes of discourses for the suspension deflection, suspension velocity and change in control is considered as [-8,8], [-8,8] and [-6000, 6000], respectively, resulting in respective scaling factors GE, GC and GU, as 10, 1 and 300.

A typical GA is considered, with the crossover and mutation probabilities 0.85 and 0.01 respectively. Simulation is carried out with population sizes set to 60, 100 and 140. The searching is terminated when no improvement is reported in 5 successive generations. The fitness function is selected as follows:

$$F = \frac{1}{\frac{1}{k}\sum_{n=0}^{n=k} |\ddot{z}_p, step(n)|} \tag{10}$$

Where, k is the sampling number. The original rule base is encoded and included into the initial population. The fitness

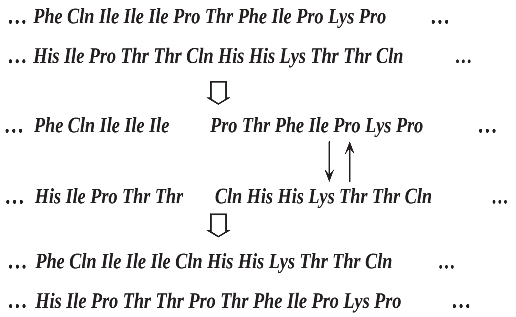
… Phe Cln Ile Ile Ile Pro Thr Phe Ile Pro Lys Pro …
… His Ile Pro Thr Thr Cln His His Lys Thr Thr Cln …
… Phe Cln Ile Ile Ile Pro Thr Phe Ile Pro Lys Pro …
… His Ile Pro Thr Thr Cln His His Lys Thr Thr Cln …
… Phe Cln Ile Ile Ile Cln His His Lys Thr Thr Cln …
… His Ile Pro Thr Thr Pro Thr Phe Ile Pro Lys Pro …

Figure 6: Crossover Operation

value associated with the original rule base was 0.6404.

The best individual with a fitness value of 0.8729 was found when the population size was set to 140. Figure 7(b) shows the time response of suspension deflection of the passive system, the original FLC, and the FLC optimized with the DNA coded GA. In Figure 7(c), the time response of the sprung mass acceleration is shown. Although the original FLC was carefully designed and has a pretty good performance, the GA optimized FLC still makes a rather good improvement over it. Both the suspension acceleration and deflection are brought to a lesser level, without any physical changes on the system and the controller.

7 Discussion

The proposed DNA coded GA has improved the performance of the FLC. The suspension acceleration and deflection was brought down by the optimized FLC, resulting in better ride comfort. With the DNA coding method, the membership functions, scaling factors and other parameters can be coded into DNA chromosome. The discussed coding scheme is capable to utilize the potential of DNA computation.

With a population size of 60, the fitness of the best individual was 0.7296. For a population size of 100, the fitness was 0.8687. The mean fitness value of the population was also noticed to grow with the population size. This is an indication that a larger population size will lead to better results. If the GA can be performed by DNA computing, we are promised to get a much better result.

Bibliography

[1] Leonard Adleman. (1994) ”Molecular Computation of Solutions to Combinatorial Problems,” Science, 266, November 11, 1994, 1021-1024.

[2] R.J. Lipton. (1995) ”Using DNA to solve NP-complete problems,” Science, 268:542–545, Apr.28,1995.

[3] D. Boneh, C. Dunworth, and R. J. Lipton. (1995) ”Breaking DES using a molecular computer,” Tech nical Report CS-TR-489-95, Princeton University, 1995.

[4] D. Boneh, C. Dunworth, R. J. Lipton, and J. Sgall. (1995) ”On the computational power of DNA,” Technical Report CS-TR-499-95, Princeton University, 1995.

[5] E. Winfree. (1995) ”On The Computaional Power of DNA Annealing and Ligation,” Technical report, California Institute of Technology, USA, May 1995.

[6] L. Adleman. (1996) ”On constructing a molecular computer,” In proceedings of the first DIMACS workshop on DNA computing. 1996

[7] D. Beaver. (1995) ”A universal molecular computer,” Technical Report CSE-95-001, Penn State University, 1995.

[8] Eric Baum. (1995) ”Building an Assoicative Menory Vastlay Larger Than the Brain,” Science, 268:583–585, April 1995.

[9] M. Appleyard and P.E. Wellstead. (1995) ”Active suspensions: Some background,” Proc. Inst. Elect. Eng.—Contr. Theory Applicat. , vol. 142, no. 2, pp. 123–128, Mar. 1995.

[10] A.G. Thompson. (1984) ”Optimal and suboptimal linear suspensions for road vehicle,” Vehicle Syst. Dynam., vol. 13, no.1-2, pp. 61C72, 1984. JHLi, MAS, Dhro

[11] J.–H. Li and T.–H.S. Li.(1995) ”Optimal output feedback stabilization of active suspension control using acceleration measurement,” in Proc. IEEE IECON95, Orlando, FL, Nov. 1995, pp. 1100–1111.

[12] M.A. Salman, A.Y. Lee, and N.M. Boustany. (1988) ”Reduced order design of active suspension control,” in Proc. 27th IEEE CDC, 1988, pp. 1038–1043.

[13] D. Hrovat. (1990) ”Optimal active suspension structures for quarter-car vehicle models,” Automatica, vol.26, no. 5, pp. 845–860, 1990.

[14] D. Park, A. Kandel, and G. Langholz.(1994) ”Genetic-based new fuzzy reasoning models with application to fuzzy control,” IEEE Trans. Syst., Man, Cybern., vol. 24, pp. 39–47, Jan./Feb. 1994.

[15] C.C. Wong and S.M. Feng. (1995) ”Switching type fuzzy controller design by genetic algorithm,” Fuzzy Sets Syst., vol.74, no. 2, pp. 175C185, 1995.

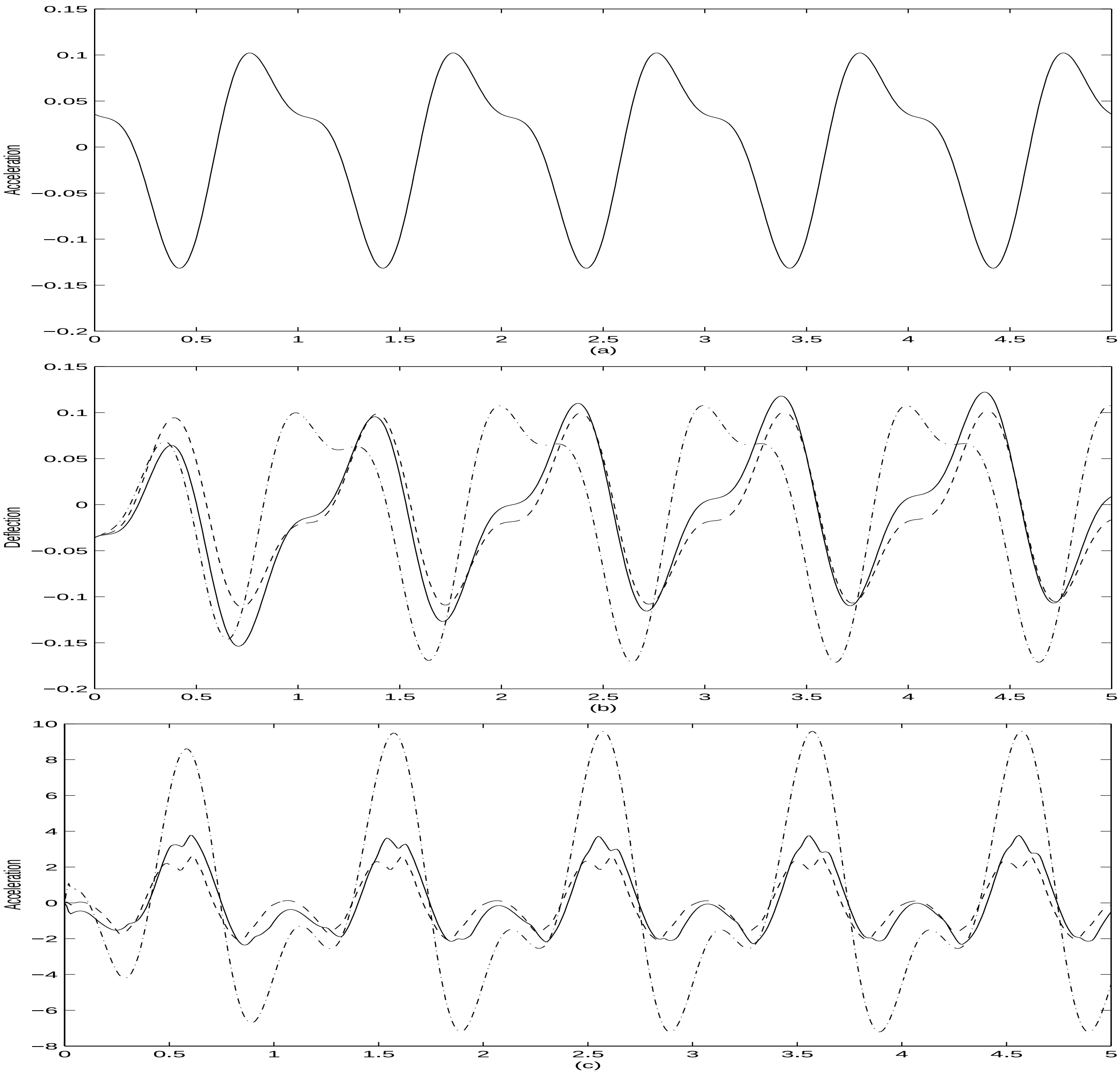

Figure 7: Passive: dot-dash; Original FLC: solid; Optimized FLC: dash

[16] A. Homaifar and E. McCormick. (1995) ”Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms,” IEEE Trans. Fuzzy Syst., vol.3, pp.129–139, May 1995.

[17] H. Surmann. (1996) ”Genetic optimization of a fuzzy system for charging batteries,” IEEE Trans. Ind. Electron., vol. 43, pp. 541C548, Oct. 1996.

[18] C.C. Lee. (1990) ”Fuzzy logic control systems:fuzzy logic controller –Part 1,” IEEE Trans. System Man Cybernet. SMC-20 404C418, 1990.

[19] M.V.C. Rao and V. Prahlad. (1997) ”A tunable fuzzy logic controller for vehicle–active syspension systems,” Fuzzy Sets and Systems 85, 11-21, 1997.

[20] M.H. Sunwoo, K.C. Cheok and H.J. Huang. ”Model reference adaptive control for vehicle for vehicle active suspension systems,” IEEE Log. No. 9144904.