Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Fuzzy Sets and Systems 147 (2004) 57–74
FUZZY sets and systems
www.elsevier.com/locate/fss

Elicitation and fine-tuning of fuzzy control rules using symbiotic evolution

M. Jamei, M. Mahfouf*, D.A. Linkens

Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield S1 3JD, UK

Abstract

This paper exploits the ability of Symbiotic Evolution (SE), as a generic methodology, to elicit a fuzzy rule-base of the Mamdani-type. Almost all fuzzy rule-base generation algorithms produce rule-bases with redundant and overlapped membership functions that limit their interpretability elegance in their application. We address this problem by applying an algorithm to merge any similar membership functions. It is shown that our proposed algorithm leads generally to a more transparent and more interpretable rule-base with a minimum number of membership functions and a reduced number of rules. In addition, a new post-processing approach is proposed for recovering any probable performance lost after membership functions merging. The proposed methodology has been applied successfully for the design of an active control suspension system using a non-linear Bond Graphs (BG) based half-car model with parameters that relate to a Ford Fiesta MK2.

Keywords: Fuzzy control; Genetic algorithms; Fuzzy Inference Systems; Symbiotic rule generation; Similarity measures; Fine-tuning; Bond graphs; Transportation; Active suspension systems

1. Introduction

Fuzzy systems design can be divided into three main stages; namely, the selection of a proper set of input and output variables and their related universes of discourse, the identification of a suitable structure for the fuzzy system, and the extraction of an optimal fuzzy rule-base. One may argue that the most crucial of the aforementioned stages is the formulation of the fuzzy rule-base; since it governs the whole fuzzy system behaviour. An expert's knowledge is used generally to construct a set of *If–Then* fuzzy statements to implement approximate reasoning. However, in many cases, there is not enough knowledge to elicit an optimised rule-base. As a

* Corresponding author. Tel.: +44-114-222-5607; fax: +44-114-222-5624.
E-mail addresses: m.jamei@shef.ac.uk (M. Jamei), m.mahfouf@sheffield.ac.uk (M. Mahfouf), d.linkens@shef.ac.uk (D.A. Linkens).

0165-0114/$ - see front matter
doi:10.1016/j.fss.2003.11.011

result, many data-driven rule generation methods have been suggested. Generally, the construction of a fuzzy system based on input–output data is not an easy task, particularly when a priori knowledge about the process is vague or not available. Many approaches were proposed for tuning a fuzzy system [6,13–15,18]. Recently, evolutionary and genetic algorithms, which are global, parallel, and not gradient-base methods, were proposed as a means to optimise fuzzy systems. For instance, Karr [12] used a Genetic Algorithm (GA) to produce the membership functions (MFs) for an adaptive fuzzy logic controller. In a fuzzy system, the MFs and the fuzzy rule-base are closely correlated and designing each part independently will not provide optimal results, especially when no or little prior knowledge is available. Consequently, and in order to obtain an optimal fuzzy controller the MFs and rule-base should be designed and optimised simultaneously. Homaifar [10] used a GA to generate MFs and the rule-base simultaneously. Nawa and his co-workers [19] proposed a method for the discovery of relevant fuzzy rules using the pseudo-bacterial GA. However, in many of these methods the MFs partitions and the fuzzy labels are usually pre-defined and the rule base to be optimised becomes inevitably sizeable. In addition, the MFs and the rules are encoded using a very long chromosome. The major disadvantage of partitioning the universe of discourse into grid-like sections is that the number of fuzzy rules, and hence the length of all chromosomes within the GA increases exponentially as the dimension of the fuzzy variables increases [11]. Hence, most of the previously proposed approaches suffered from the curse of dimensionality. Juang and co-workers [11] proposed an efficient genetic reinforcement-learning algorithm for designing fuzzy systems based on Symbiotic Evolution (SE) and applied it to Takagi–Sugeno–Kang (TSK) type fuzzy systems. The proposed method consists of partitioning the universe of discourse relating to the fuzzy variables in a more flexible way, leading to a relatively smaller number of rules compared to that obtained using standard partitioning.

In this paper, a methodology for eliciting the fuzzy rule-base of the Mamdani-type fuzzy system using SE algorithm is proposed. In addition, and in order to accelerate the optimization process, the Selective Breeding [16] algorithm is included in the evolution process of the GA structure. The proposed algorithm is applied to design an active suspension system successfully using a non-linear Bond Graphs (BG) representation of a half-car, in this case a Ford Fiesta MK2. Because the initially generated fuzzy rules are generally less transparent with redundant membership functions, a new algorithm for approximating Gaussian membership functions with Trapezoidal membership functions is applied in order to merge similar membership functions and hence to obtain a more transparent rule-base with a minimum number of membership functions. Section 2 of this paper discusses the SE method and the proposed fuzzy rule-base generation. Similarity measures and membership functions merging are explained in Section 3. Section 4 describes the post-processing algorithm for fine-tuning of the generated fuzzy rule-base, and Section 5 presents the model behind the half-car suspension system as well as the results of experiments. Finally, in Section 6, concluding remarks relating the overall study will be drawn.

2. Symbiotic evolution-based fuzzy rules generation

A fuzzy rule-base consists of a set of *If–Then* fuzzy rules, which for a multi-input single output system, can be defined as follows:

$$
\begin{aligned}
&IF\ x_1\ is\ A_{11}\ and \ldots x_n\ is\ A_{n1}\ THEN\ y\ is\ B_1\\
&IF\ x_1\ is\ A_{12}\ and \ldots x_n\ is\ A_{n2}\ THEN\ y\ is\ B_2\\
&\ldots \qquad\qquad \ldots \qquad\qquad \ldots\\
&IF\ x_1\ is\ A_{1n}\ and \ldots x_n\ is\ A_{nn}\ THEN\ y\ is\ B_n
\end{aligned} \tag{1}
$$

where x_i are input variables, y is the output variable, and A_{ij} and B_i are the input and output fuzzy labels, respectively. The fuzzy rule-base generation is the most crucial phase in the fuzzy system design stage. However, many researchers have suggested different methods for generating and tuning the fuzzy rule-base; this area being still under extensive investigation. Most of the proposed rule-base generation methods are founded on identifying an optimal rule-base as a whole. Consequently, those methods focus on the performance of the entire fuzzy rule-base instead of considering the performance of individual rules in the rule-base. These approaches, generally, suffer from the curse of dimensionality since any increase in the dimension of the system can impact significantly on the dimension of the rule-base and as a result can compromise performance.

The proposed approach in this paper, alternatively, employs SE to select and combine individual fuzzy rules, based on their expertise, to work synergetically in order to achieve an optimal performance.

SE is inspired by nature; in an ecological unit, species work together towards some common aims. However, in order to survive and produce their offspring, they have to adapt themselves to the ever-changing environment. In this process, there are symbiotic relations between individuals such as competition, exploitation, and benefit. The relations are commonly called symbiosis [9]. In order to apply SE within a rule-base generation context, each chromosome in the population represents only one fuzzy rule instead of the whole fuzzy rule-base.

Each rule, in a fuzzy rule-base, is responsible for a particular part of the control surface, in other words, each fuzzy rule is specialized or expert in a specific area. As a result, if a fuzzy rule performs well in one rule-base, it will presumably succeed in another rule-base, however, this is not true for a node in a neural network. In light of these aforementioned facts, each rule contributes only as a partial solution and this can be interpreted as a specialization. Hence, none of the rules in the population can perform well alone or take over the population since this is just a part of the whole optimal solution [17]. Therefore, the SE method maintains the diversity of the population and prevents the population from converging prematurely and hence providing sub-optimal solutions.

In this current research work, Gaussian membership functions as $\mu(c,\sigma)$ are used; where c is the centre and σ is the standard deviation (or the width) of the membership function. The parameters of each membership function are encoded into either Binary or Gray bits and concatenated as a chromosome as shown in Fig. 1; where c_{i} and σ_{i} are the parameters of the input fuzzy variables and c_{o} and σ_{o} are the parameters of the output fuzzy variable. Therefore, each chromosome represents only an individual fuzzy rule in the population which is a partial solution to the problem.

However, in a conventional method of GA-based coding each chromosome consists of a complete set of fuzzy rules as shown in Fig. 1b. In the SE algorithm, the population sets up a pool of promising rules that are randomly initialised for the first generation; randomly selecting and combining Nr chromosomes from the population constructs an Nr-rule Fuzzy Inference System (FIS). The objective function value which is equal to the performance of the generated FIS is then calculated, and based

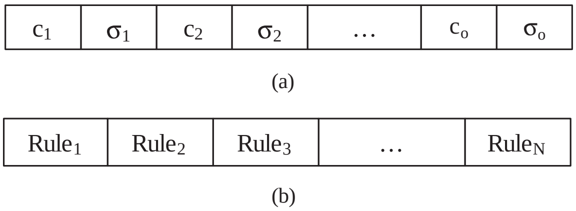

Fig. 1. Representation of (a) a fuzzy rule as a chromosome and (b) a fuzzy rule-base as a chromosome.

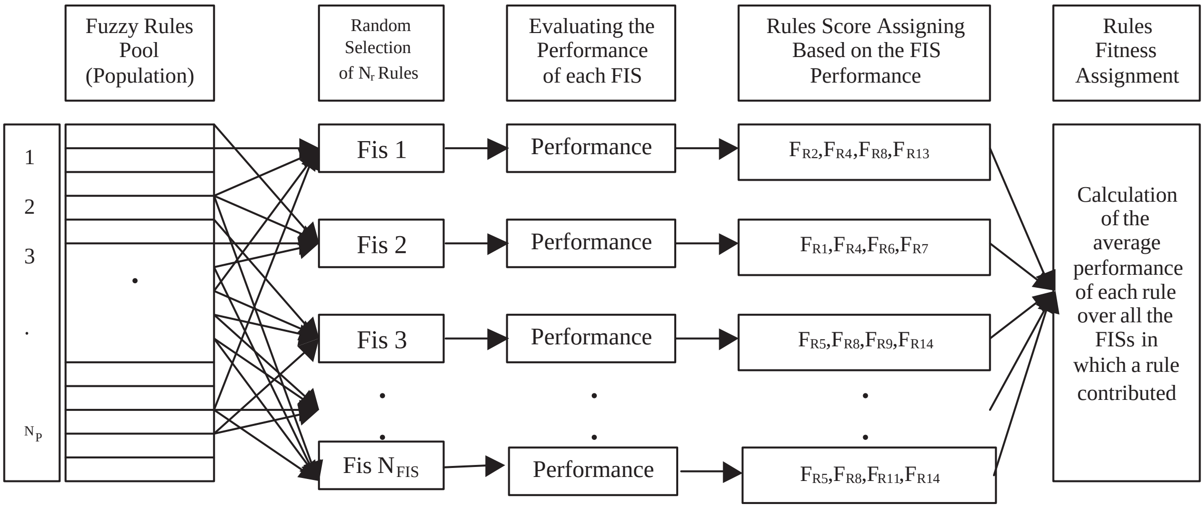

Fig. 2. Rules fitness assignment.

on its performance an equal fitness score is assigned to all contributing rules as shown in Fig. 2; this process continues until all N_{FIS} rule-bases are set up and evaluated. The rules fitness assignment is a crucial and challenging step in the algorithm and currently, the overall fitness of an individual is calculated by averaging its fitnesses over all the FISs where it has contributed. At this stage, the GA is ready to select the fittest rules and to reproduce the next generation of rules. When the potential parents are chosen, multi-point crossover and mutation operations are applied but the resulting offspring have not directly set up the new generation. Instead, individuals from the old generation and offspring sets are combined and the best individuals are selected to represent the new generation.

In the proposed algorithm, the Binary coding method is used to define the parameters of the Gaussian MFs. We believe that Binary coding is necessary to keep a minimum distance between centres of adjoining MFs by selecting the desired number of bits to represent a particular centre of the Gaussian function. The universe of discourse of each fuzzy variable is normalized between $[-1, 1]$ and all centres of the fuzzy sets are coded as a 4-bit strand leading to a minimum distance of 0.13 between MFs. In addition, since there is no advantage in allowing the standard deviation of the fuzzy sets to change over a continues range, it is chosen from a pre-defined set of values i.e. $\{0.15, 0.25, 0.3, 0.4\}$. The restrictions on the parameters of the MFs prevent a large overlap, a very narrow, a singleton set, or even a very wide and universal set to be produced. Hence, at the optimisation phase, the generated MFs will be more interpretable than other methods and merging similar MFs will be easier compared to other data-driven rule generation methods. In addition, we used additional measures to prevent redundant or inconsistent rules to be included in the rule-base relating to a specific FIS. In order to implement these measures, each randomly selected rule is checked against the rules which exist in the current FIS and if a similar or an inconsistent rule is already included in the FIS this rule is rejected and a new one is chosen instead. Moreover, some precautions were implemented to ensure that all MFs are selected at least a certain number of times. With the intention of implementing this algorithm, the number of times that a rule is selected is recorded and then if it is less than a pre-defined value, the rule is added deterministically to the current rule-base.

The SE algorithm represents the optimal rules using their MFs, hence, the first rule is described as follows:

$$\textit{If } x_1 \textit{ is } MF_{1i} \textit{ and } x_2 \textit{ is } MF_{2i} \textit{ and} \cdots \textit{and } x_n \textit{ is } MF_{ni} \textit{ Then } y \textit{ is } MF_i. \tag{2}$$

Therefore, each fuzzy input or output variable consists of the same number of MFs as the rules, however, some of the MFs are likely to be identical.

3. Similarity measures and MF merging

Similarity of MFs is a measure that defines to what extent the fuzzy sets are identical. Similar fuzzy sets express almost the same region in the universe of discourse of a fuzzy variable; in other words, they describe the same concept. Therefore, the fuzzy system is made more complex than necessary by those fuzzy sets and they should be merged. There are two methods for analysing the similarity of fuzzy sets, namely set-theoretic and geometric similarity measures. The geometric similarity measures point up closeness of fuzzy sets, however, set-theoretic measures achieve the degree of equality of fuzzy sets. Hence, the latter measures were selected for inclusion within our algorithm.

Let A and B be two fuzzy sets with the membership functions μ_A and μ_B, respectively. The similarity of those fuzzy sets may vary from 0, which means "completely distinct", to 1, which means that the fuzzy sets are equal. The most common similarity measure of fuzzy sets in the literature is based on the intersection and union operations among fuzzy sets and defined as follows:

$$S(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}, \tag{3}$$

where S is the similarity measure and $|\cdot|$ designates the cardinality or the size of a set, and intersection and union operators are showed by $\cap$ *and* $\cup$, respectively [5]. However, implementation of this measure in a discrete universe is an easy task, but in a continues universe of discourse will prove computationally intensive, particularly for Gaussian or Bell-shaped membership functions. Therefore, some simplification methods for calculating the similarity of fuzzy sets have been previously suggested. For instance, Chao et al. [3] used a triangular function with centre c and width $\sigma\sqrt{\pi}$ instead of a Gaussian function $G(c,\sigma)$ for such a similarity measure evaluation. Chen [4] recently proposed a more accurate approximation of any symmetrical membership functions by using a trapezoidal

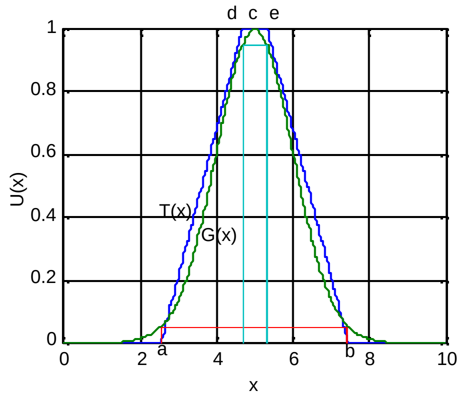

Fig. 3. Membership function approximation.

membership function,

$$T(x) = ma\left\{min\left\{\frac{x-a}{d-a}, 1, \frac{b-x}{b-e}\right\}, 0\right\},$$

where the parameters of the trapezoidal membership function, a, b, e and d are calculated using the Gaussian membership function. Also, two α-cut sets of a fuzzy are used; the α-cut of a fuzzy set A of a fuzzy variable x, designated by A_α, is the crisp subset of the universe of discourse of X whose membership function is greater than or equal to α, $A_\alpha = \{x \in X | \mu(x) \geqslant \alpha\}$, where α is a parameter in the range $0 < \alpha \leqslant 1$. The two α-cut sets are the top α-cut, $A_{\alpha T}$, and the bottom α-cut, $A_{\alpha B}$, where $\alpha_T \in [0.95, 1]$ and $\alpha_B \in [0, 0.05]$ are as shown in Fig. 3. At each iteration of similarity analysis, the similarity of all pairs of fuzzy sets of a fuzzy variable are calculated then, if the maximum similarity value is greater than a pre-defined threshold the most similar fuzzy sets will be replaced by a new fuzzy set. The lower the threshold, the fewer number of fuzzy sets are used in the fuzzy rule-base, however it is worth noting that the threshold selection is application-dependent.

It is worth noting that the performance of the controller will degrade as the threshold decreases, although, this is not always the case. The similarity analysis of each fuzzy variable is independent of the other ones. If the similarity of two fuzzy sets is greater than the threshold, both sets are replaced by a new fuzzy set having parameters defined according to the following equation:

$$c_{\text{new}} = \frac{c_1\sigma_1 + c_2\sigma_2}{\sigma_1 + \sigma_2}\,\sigma_{\text{new}} = \frac{\sigma_1 + \sigma_2}{2}. \tag{4}$$

After fuzzy sets simplification rule-base merging is likely, particularly if the rules are highly redundant.

4. Membership functions post-processing or fine-tuning

As already mentioned, merging redundant membership functions, generally, deteriorates the performance of the fuzzy controller due to the fact that the sensitivity of the controller to the inputs is

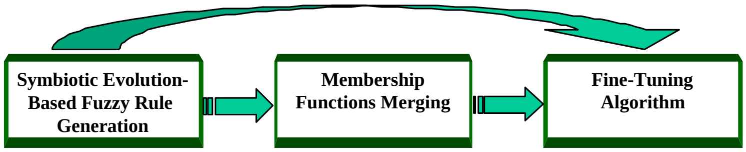

Fig. 4. A schematic of the proposed symbiotic evolution-based fuzzy rule generation.

usually reduced. In order to recover the declined performance of the fuzzy controller, we propose to apply a post-processing or fine-tuning procedure on the merged membership functions. Therefore, the controller enjoys both facets of an optimal fuzzy controller in terms of interpretability and accuracy.

Having considered the facts that the symbiotic algorithm generates an optimal controller and the merging process does not significantly change the controller's structure, one may conclude that is not necessary to spend a substantial effort to recover any lost performance. This being true, the post-processing is a fine-tuning and not a coarse one. In light of the aforementioned points, the proposed fine-tuning algorithm is not allowed to change the structure of the fuzzy controller. In this process, a GA is employed to distinguish the optimal mean of the Gaussian membership functions in a restricted small interval, which in this case is $[-0.1, 0.1]$, around the current means. However, the width of the membership functions is kept constant; in this manner, the possibility of generating overlapped membership functions is reduced.

The fine-tuning procedure can also improve the performance of the obtained optimal fuzzy controller even before merging and Section 5 below strongly reinforces this assertion. The main reason for this performance improvement can be explained through the nature of fuzzy rule generation algorithms themselves. These algorithms are deliberately made to explore a very wide area for potential optimal fuzzy rules and therefore cannot precisely explore the area around the optimal solutions, whereas, a fine-tuning method can easily investigate this area and improve the performance.

Fig. 4 represents the overall symbiotic evolution-based rule generation sequence which allows one to obtain an optimal and transparent fuzzy controller; in addition, it shows that the fine-tuning process can also benefit the original obtained controller without even going through the membership functions merging process.

5. Simulation results

The proposed algorithm was applied to design a half-car fuzzy control-based active suspension system as a test bed. In an active suspension system design exercise, the main objective is to improve the ride performance while keeping a good tyre to road contact, which is essential for car handling and safety. The vertical body acceleration produced on the seats is the most common and meaningful measure of ride performance [7]. The vehicles have a longitudinal distance between the front and rear axles and they are multi-input systems that respond to the vertical bounce and the pitch motions. Hence, in addition to the vertical body acceleration which is included in a quarter-car model, a half-car model includes also an angular movement, called pitch, around its lateral axis. The pitch motions are the primary source of longitudinal vibrations at locations above the centre of gravity of vehicles. Therefore, the symbiotic evolution is used to identify a fuzzy controller that

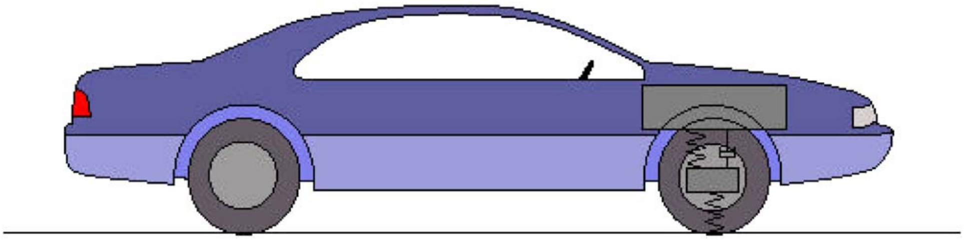

Fig. 5. The quarter-car model of a vehicle.

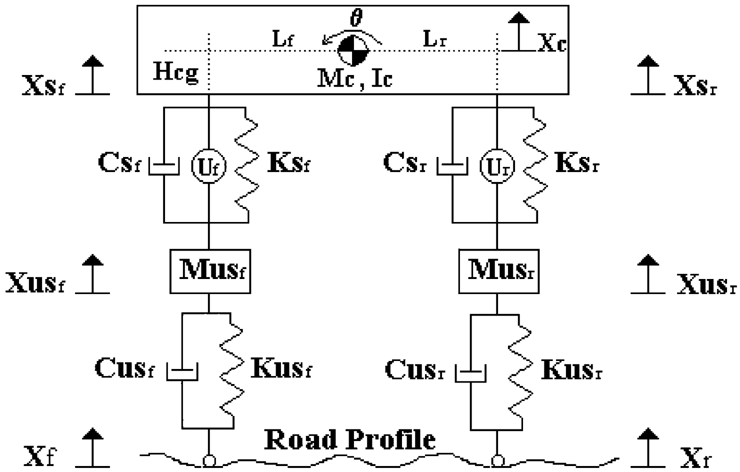

Fig. 6. A schematic diagram representing the half-car model.

minimizes the level of Root Mean Square (RMS) of the body acceleration, heave, and the RMS of the pitch acceleration around the lateral axis of the car body whilst maintaining contact between the wheels and the road surface.

The developed system is a non-linear Bond Graph (BG) model of a half-car suspension system. Fig. 5 shows how a quarter-car model is related to the car's physical components and Fig. 6 shows a schematic diagram for a half-car model.

Since in a fuzzy logic controller the fuzzy rule-base is built using linguistic variables, opting for the BG modelling method with physical variables gives a better understanding of the model and the controller. Generally, a half-car model is composed of the front and rear quarter-car models that interact through the car body or the sprung mass (M_c) and the body inertia (I_c). The f and r subscripts, in Fig. 6, refer to the front and rear sections, respectively. Each quarter-car model consists of the stiffness (K_s) and damping (C_s) characteristics and each part is connected to the related unsprung mass (M_{us}) of the axle. Also, each tyre is modelled as a pair of spring (K_{us}) and damper (C_{us}). In this configuration (see Fig. 6), the actuators (U) are situated in parallel with the main springs and dampers. The dynamics of the actuators are reduced to a time delay equal to 0.001 s and they can apply their force in both upward and downward positions.

Table 1
The parameters related to the half-car model of a Ford Fiesta MK2

Description	Values
Sprung mass	433.50 (kg)
Sprung mass pitch moment of inertia (about CG)	1026 (kg m^2)
Wheel base	2.288 (m)
Horizontal distance from front axle to body CG	0.846 (m)

Table 2
The parameters related to the front and rear sections of a Ford Fiesta MK2

Description	Front	Rear
Unsprung mass	28.85 (kg)	36.20 (kg)
Tyre linear spring rate	184.00×10^3 (N/m)	184.00×10^3 (N/m)
Tyre damping coefficient	5000.00 (N s/m)	5000.00 (N s/m)
Damper compression rate	538.00 (N s/m)	385.00 (N s/m)
Damper rebound coefficient	1828.00 (N s/m)	1444.00 (N s/m)
Suspension spring stiffness	217.0×10^2 (N/m)	217.0×10^2 (N/m)
Tyre freed radius	0.270 (m)	0.270 (m)

X_s, X_{us}, and X_r are the car body, the unsprung mass and the road displacements respectively in both front and rear quarter-cars. When the velocity of a shock absorber is positive (rebound) the value of the damping coefficient is different with its coefficient while the velocity is negative (jounce); therefore, the damper is modelled as a non-linear element. Tables 1 and 2 present the parameters related to a Ford Fiesta MK2 car which were used to simulate the BG model [8], and Fig. 7 represents the developed BG model. The vertical acceleration is obtained by considering the forces acting on the unsprung and sprung masses and the pitch acceleration is calculated by considering the inertia of the sprung mass.

The half-car model differential equations can be obtained by applying the principles of BGs, the front and rear compressive forces and front and rear tyre forces equations are:

$$F_f = Ks_f(Xs_f - Xus_f) + Cs_f(\dot{X}s_f - \dot{X}us_f) - U_f, \tag{5}$$

$$F_r = Ks_r(Xs_r - Xus_r) + Cs_r(\dot{X}s_r - \dot{X}us_r) - U_r, \tag{6}$$

$$Fus_f = Kus_f(Xus_f - X_f) + Cus_f(\dot{X}us_f - \dot{X}_f), \tag{7}$$

$$Fus_r = Kus_r(Xus_r - X_r) + Cus_r(\dot{X}us_r - \dot{X}_r). \tag{8}$$

The F_f *and* F_r forces impose an acceleration to the car body, which in turn causes a pitch and a bounce as follows:

$$F_fR_f - F_rR_r = Ic\ddot{\theta}, \tag{9}$$

$$-F_f - F_r = Mc\ddot{X}c, \tag{10}$$

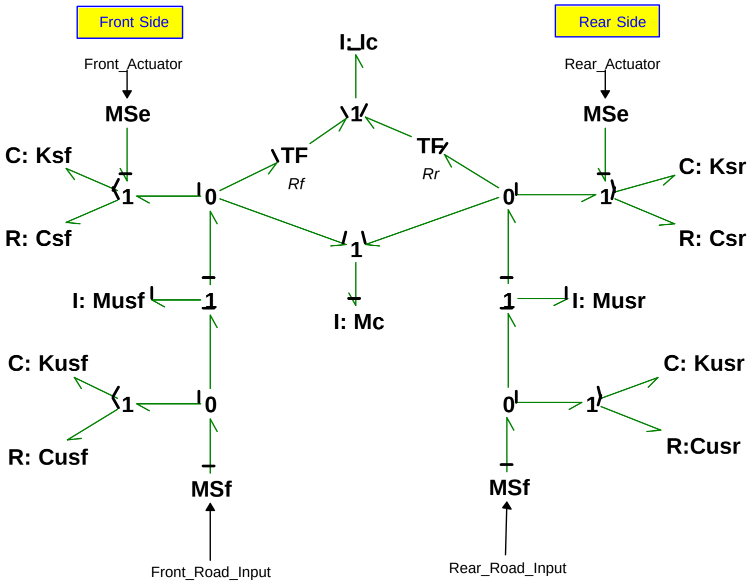

Fig. 7. The BGs representation of the half-car model.

where R_f *and* R_r are:

$$R_f = L_f \operatorname{Cos}\theta + \mathrm{HcgSin}\,\theta, \tag{11}$$

$$R_r = -(L_r \operatorname{Cos}\theta - \mathrm{HcgSin}\,\theta). \tag{12}$$

By considering small values for θ, (11) and (12) will become:

$$R_f = L_f, \tag{13}$$

$$R_r = -L_r. \tag{14}$$

In addition, by considering the tyre forces, the unsprung masses accelerations can be calculated as follows:

$$-F_f + Fus_f = Mus_f \ddot{X}us_f, \tag{15}$$

$$-F_r + Fus_r = Mus_r \ddot{X}us_r. \tag{16}$$

In order to model the road surface roughness (w), the following first-order differential equation is used [20]:

$$\dot{w} + avw = av\xi, cov[\xi(t)] = 2\sigma^2, \tag{17}$$

where a is a positive constant, v is the vehicle speed, and ξ is a zero-mean Gaussian random process with the standard deviation σ. The road model parameters were set to generate a road profile within the new pavements category with an International Roughness Index (IRI) of the order of 2.92 (see Table 3).

Table 3
The road model parameters

Parameter	Value
a	0.15
v	20 (m/s)
σ	50×10^{-5}

It is assumed that, for a rear wheel, the road disturbance is an identical but time-delayed version to that for the front wheel on the same side of the vehicle [2]. Hence, for a half-car model, a road profile is generated and applied to the front wheel and the same profile with a time-delay is applied to the rear wheel; this time-delay is obtained by diving the vehicle wheelbase by the vehicle speed.

In order to design the fuzzy controller, it was decided to use the suspension travel, rattlespace, and the suspension velocity as the controller inputs. However, since there are two sets of these variables, the combination of the front and rear suspension travels, $(Xs_f - Xus_f) + (Xs_r - Xus_r)$, and the combination of the front and rear suspension velocities, $(\dot{X}s_f - \dot{X}us_f) + (\dot{X}s_r - \dot{X}us_r)$ were used. Moreover, the controller output is the required forces that should be applied to the system by both actuators.

The developed model was simulated using a 0.001 s step size for a 3-s period and in each generation 30 fuzzy controllers were randomly set up by selecting 9 fuzzy rules from the pool of 80 rules. In each generation, the performance of all fuzzy controllers were assessed, and based on their performance a fitness score was assigned. After evaluating all fuzzy controllers, the fitness of the fuzzy rules were calculated, and based on these values GA produced the next generation. Obviously, simulating and evaluating N_{FIS} active suspension systems at each generation requires considerable computing power. Hence, with the intention of speeding up the optimization process, all the related computing procedures, including the simulation, the symbiotic evolution algorithm, and the fuzzy controller were entirely implement using C++ language-based code rather than a Matlab-based code.

Fig. 8a presents the heave and pitch accelerations for the passive suspension system and Fig. 8b shows their power spectrum density (PSD). The most important range of the PSD for the ride and handling purposes is located roughly in the range of 1–15 Hz.

With the intention of computing a weighted combination of RMSs of the heave and pitch accelerations, the following cost function was defined:

$$\Psi(\ddot{X}c_{\text{RMS}}, \ddot{\theta}_{\text{RMS}}) = \lambda \ddot{X}c^2_{\text{RMS}} + (1-\lambda)\ddot{\theta}^2_{\text{RMS}} \quad 0 \leqslant \lambda \leqslant 1, \tag{18}$$

where λ defines the relative priority of the heave and pitch during the optimisation process. Using $\lambda = 0.5$ in the objective function, the performance of the passive suspension system for the RMSs of heave and pitch accelerations were 0.4580, 0.1693, respectively, and the overall cost function obtained was 0.2384. Subsequently, the symbiotic evolution algorithm was applied in which the proportional selection method was the Function Normalization [16]. Moreover, the sampling method was the stochastic universal sampling instead of the roulette wheel, since the former exhibited less variance than repeated calls to the roulette wheel algorithm and eliminated sampling errors [1]. The generated membership functions before and after fine-tuning are shown in the upper and lower sections of

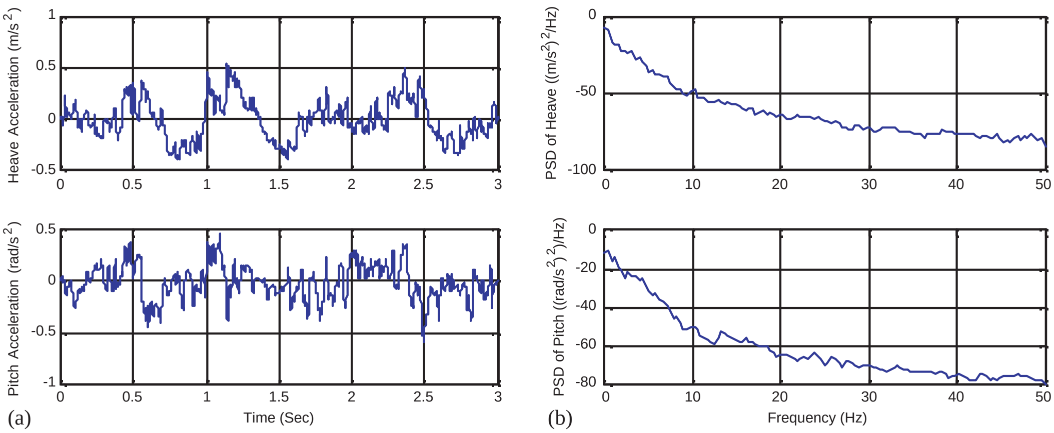

Fig. 8. The heave and pitch accelerations: (a) PSD of the heave and pitch accelerations and (b) for the passive suspension system.

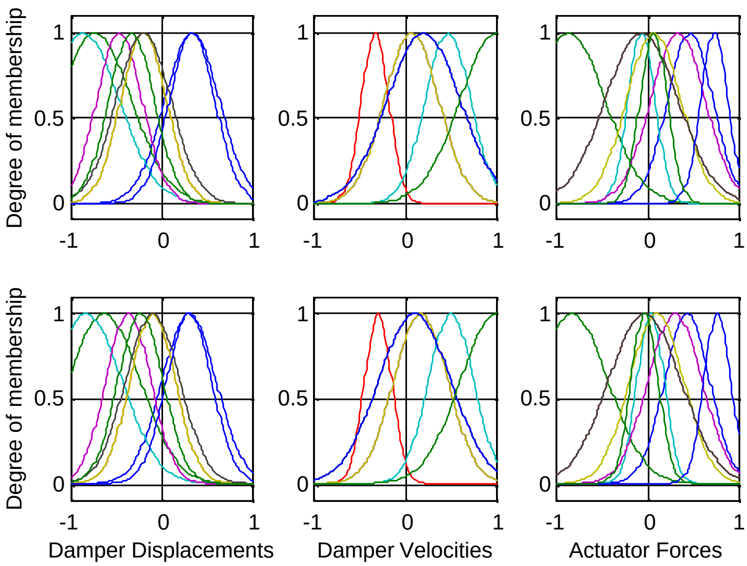

Fig. 9. The generated MFs using symbiotic evolution, before fine-tuning (upper MFs), and after fine-tuning (lower MFs).

Fig. 9 respectively and the control surface of the fine-tuned controller is shown in Fig. 10. As Fig. 9 shows, after fine-tuning, the membership functions are slightly rearranged; however, these changes may seem too insignificant but they brought about more than 13% improvement. The overall cost function and the RMSs for the heave and pitch accelerations in the passive and active suspension systems are given in Table 4. Generally, the performance of the heave and the overall cost function were significantly improved, however, the pitch acceleration was slightly worse. Obviously, the concentration of the symbiotic evolution method, or their relative priority in the overall cost function, can be changed towards the pitch or heave by varying λ. Fig. 11 shows the pitch and

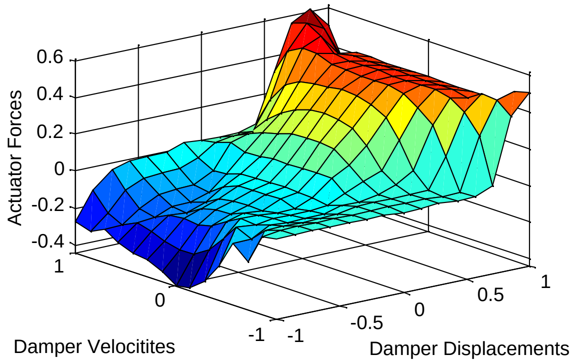

Fig. 10. The control surface for the generated rule-base after fine-tuning.

Table 4
Comparison of RMSs for the passive and active suspension systems

RMS	Passive suspension system	Active suspension system	Active suspension system after fine-tuning
Pitch acceleration	0.1693	0.1920	0.1886
Heave acceleration	0.4580	0.2514	0.2212
Overall performance	0.2384	0.1001	0.0845

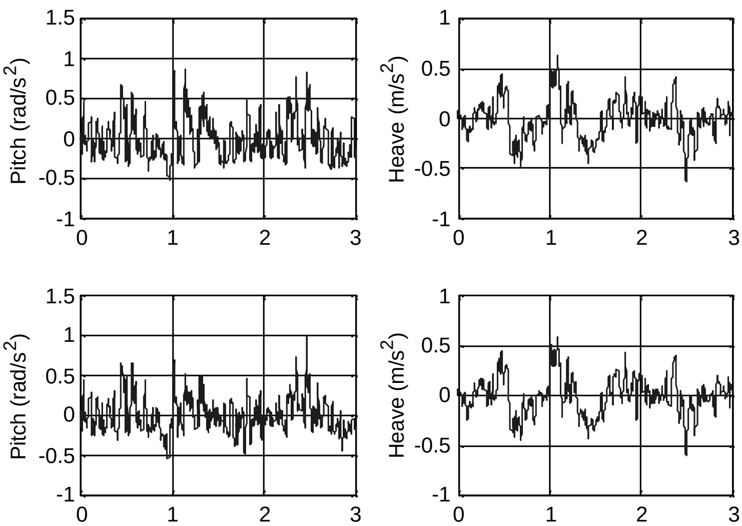

Fig. 11. The pitch and heave accelerations from the symbiotic evolution, before fine-tuning (upper), and after fine-tuning (lower).

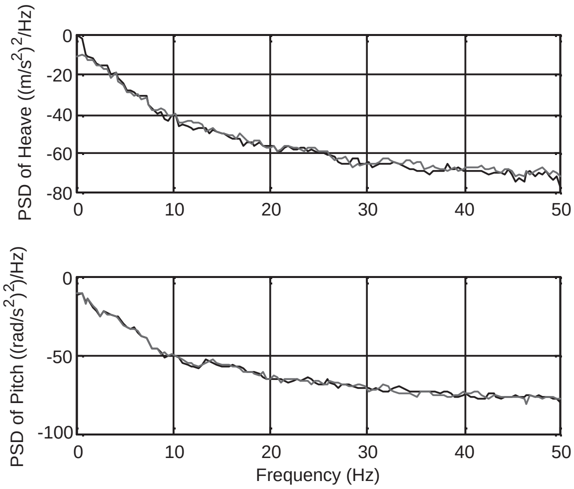

Fig. 12. Comparison of the PSD of the heave and pitch accelerations for the passive (black) and the active (grey) suspension systems.

heave accelerations before and after fine-tuning, in the upper and lower sections, respectively. In order to compare the performance of the active suspension system in the frequency domain the PSD of both heave and pitch accelerations are shown against their passive suspension system counterparts in Fig. 12. The heave has a lower amplitude in the lower frequency range in comparison with its passive counterpart; however, the pitch acceleration has a slightly higher amplitude in that area in comparison with its passive counterpart. Moreover, this figure indicates that the amplitude of PSD in high frequencies (higher than 15 Hz) for both passive and active suspension systems are very similar; hence, the controller did not impose any high frequency vibrations.

As shown in Fig. 9, some of the generated MFs have a high degree of similarity; hence, it is necessary to merge them together to achieve a fuzzy rule-base with a small degree of redundancy. Therefore, the merging algorithm of Section 3, which uses different thresholds, was implemented to reduce the number of membership functions. Figs. 12 and 14 show the merged MFs using thresholds of 0.7 and 0.4, respectively. In addition, the heave and pitch accelerations for each case are represented in Figs. 13, 15 and 16. These figures reveal that in both cases there are degradations in the performance of the active suspension system after merging, however, a significant level of this lost performance was recovered after the fine-tuning process was applied. Hence, in all these examined cases, the usefulness of the fine-tuning process was proved even when no merging was applied to the generated MFs.

Table 5 indicates the number of MFs after applying the similarity analysis and the merging procedure for three thresholds. As this table shows, the level of transparency and interpretability of the fuzzy system and the number of MFs depends on the pre-defined threshold. However, a small threshold brings about a more transparent fuzzy rule-base but simultaneously causes performance deterioration which can be offset via the fine-tuning process. Table 6 shows that in all cases the symbiotic evolution algorithm succeeded in generating a good fuzzy controller, in addition, the fine-tuning process for any pre-defined threshold improved the overall performance.

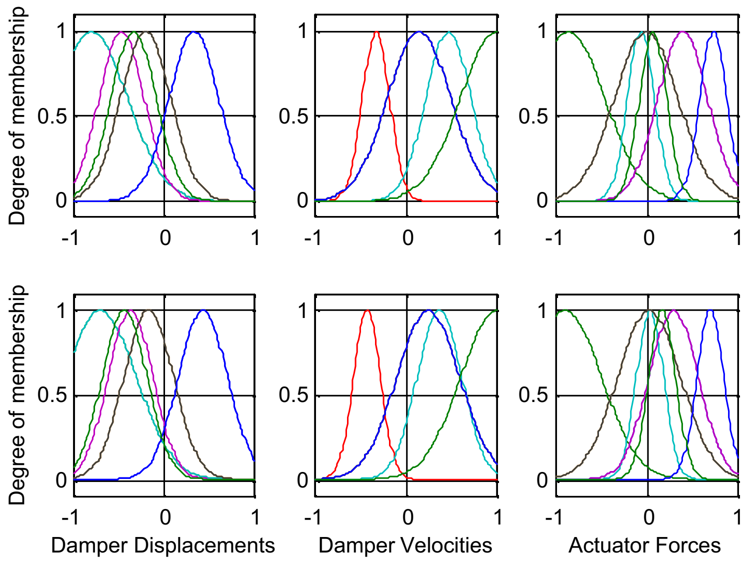

Fig. 13. The merged MFs with a threshold of 0.7, before fine-tuning (upper MFs), and after fine-tuning (lower MFs).

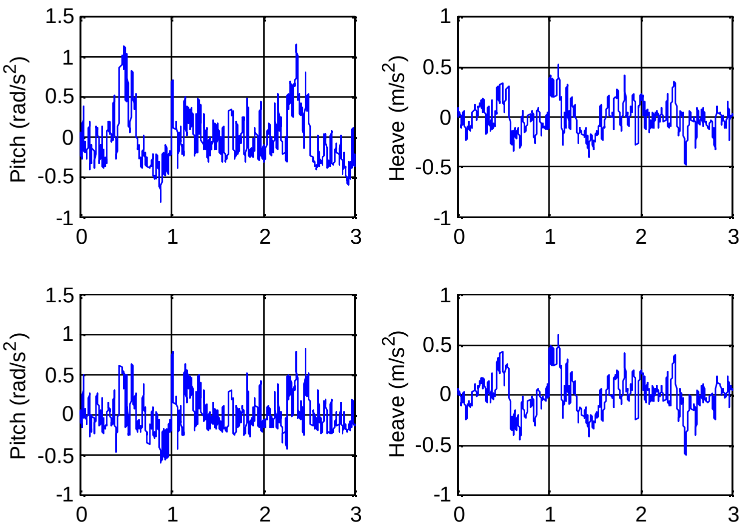

Fig. 14. The pitch and heave accelerations after merging similar MFs with a threshold of 0.7, before fine-tuning (upper), and after fine-tuning (lower).

6. Conclusions

In this paper, the symbiotic evolution methodology for designing a Mamdani-type fuzzy system was proposed. Because the generated membership functions are redundant or overlapped, a similarity analysis was applied to merge any similar MFs in the fuzzy system based on a pre-defined

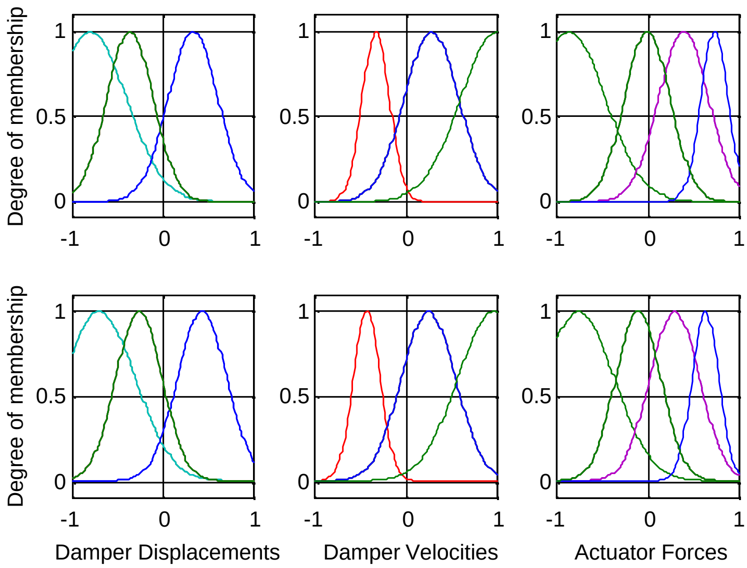

Fig. 15. The merged MFs with a threshold of 0.4, before fine-tuning (upper MFs), and after fine-tuning (lower MFs).

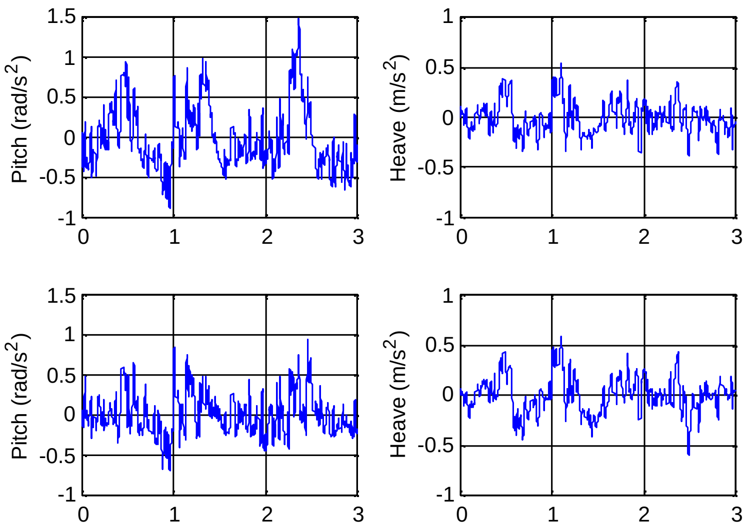

Fig. 16. The pitch and heave accelerations after merging similar MFs with a threshold of 0.4, before fine-tuning (upper), and after fine-tuning (lower).

threshold. As a result, the final elicited rule-base proved more transparent and more interpretable with a minimum number of membership functions. In addition, with the intention of improving the performance of the fuzzy controller after merging, a fine-tuning approach was proposed that significantly improved the efficiency of the controller even before merging. The proposed methodology was applied to the design of an active control suspension system successfully using a non-linear half-car

Table 5
The number of MFs for the inputs and output obtained after merging with the given thresholds

Fuzzy variables	Threshold		
	1	0.7	0.4
MFs of Input 1	8	5	3
MFs of Input 2	5	4	3
MFs of Output	8	6	4

Table 6
Comparison of performances for the passive and active suspension systems

Description	Passive suspension system	Active Suspension System					
		Threshold = 1		Threshold = 0.7		Threshold = 0.4	
		Fine-tuning		Fine-tuning		Fine-tuning	
		Before	After	Before	After	Before	After
Pitch RMS	0.1693	0.1920	0.1886	0.1567	0.1852	0.1551	0.1840
Heave RMS	0.4580	0.2514	0.2212	0.3275	0.2335	0.3989	0.2594
Overall performance	0.2384	0.1001	0.0845	0.1318	0.0888	0.1832	0.1011

model. In this application a weighted combination of the RMSs of the pitch and heave accelerations were minimized to improve the ride performance of the suspension system. It is hoped to extend the proposed methodology for MIMO systems and also for prediction.

Acknowledgements

The authors wish to thank the reviewers for the useful comments they made to improve the quality of this paper and the first author would like to express his gratitude to the Ministry of Science, Research, and Technology of the Government of Iran for their financial support.

References

[1] T. Bäck, D.B. Fogel, Z. Michalewics, Handbook of Evolutionary Computation, Institute of Physics Pub., Oxford University Press, Bristol, 1997.
[2] A.W. Burton, Active vibration control on automotive chassis systems, Comput. Control Eng. J. 4 (5) (1993) 225–232.
[3] C.T. Chao, Y.J. Chen, C.C. Teng, Simplification of fuzzy-neural systems using similarity analysis, IEEE Trans. Systems Man Cybernet. Part B: Cybernetics 26 (2) (1996) 344–354.
[4] M.-Y. Chen, D.A. Linkens, Rule-based self-generation and simplification for data-driven fuzzy models, Fuzz-IEEE, Melbourne, Australia, 2001, pp. 400–403.
[5] D.J. Dubois, H.M. Prade, Fuzzy Sets and Systems: Theory and Applications, Academic Press, New York, 1980.
[6] C.-S. Fahn, K.-T. Lan, Z.-B. Chern, Fuzzy rules generation using new evolutionary algorithms combined with multilayer perceptrons, IEEE Trans. Ind. Electron. 46 (6) (1999) 1103–1113.

[7] T.D. Gillespie, Fundamentals of Vehicle Dynamics, Society of Automotive Engineers, Inc., Warrendale, 1992.
[8] L.R. Hickson, Design and development of an active roll control suspension, Ph.D., Department of Mechanical Engineering, The University of Bath, Bath, 1996.
[9] K. Hirasawa, Y. Ishikawa, J. Hu, J. Murata, J. Mao, Genetic symbiosis algorithm, 2000 Congress on Evolutionary Computation, La Jolla, CA, USA, 2000, pp. 1377–1384.
[10] A. Homaifar, E. McCormick, Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms, IEEE Trans. Fuzzy Systems 3 (4) (1995) 129–139.
[11] C.-F. Juang, J.-Y. Lin, C.-T. Lin, Genetic reinforcement learning through symbiotic evolution for fuzzy controller design, IEEE Trans. Systems Man Cybernet.—Part B: Cybernetics 30 (2) (2000) 290–301.
[12] C.L. Karr, Design of an adaptive fuzzy logic controller using a genetic algorithm, The Fourth International Conference on Genetic Algorithms, 1992, pp. 450–457.
[13] A. Krone, H. Kiendl, Rule-based decision analysis with fuzzy-rosa method, EFDAN, Dortmund, Germany, 1996, pp. 109–114.
[14] C.T. Lin, C.S.G. Lee, Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems, PTR Prentice-Hall, Upper Saddle River, NJ, 1996.
[15] D.A. Linkens, M. Chen, Input selection and partition validation for fuzzy modelling using neural network, Fuzzy Sets and Systems 107 (3) (1999) 299–308.
[16] D.A. Linkens, H.O. Nyongesa, Genetic algorithms for fuzzy control, Part 1: Offline system development and application, IEE Proc. Control Theory Appl. 142 (3) (1995) 161–176.
[17] D.E. Moriarty, R. Miikkulainen, Efficient reinforcement learning through symbiotic evolution, Mach. Learning 22 (1996) 11–32.
[18] O. Nelles, Ga-based generation of fuzzy rules, in: W. Pedrycz (Ed.), Fuzzy Evolutionary Computation, Kluwer Academic Publishers, Boston, 1997, pp. 269–295.
[19] N.E. Nawa, T. Furuhashi, T. Hashiyama, Y. Uchikawa, A study on the discovery of relevant fuzzy rules using pseudo-bacterial genetic algorithm, IEEE Trans. Ind. Electron. 46 (6) (1999) 1080–1089.
[20] H.-S. Roh, Y. Park, Preview control of active vehicle suspensions based on a state and input estimator, Developments in Tire, Wheel, Steering, and Suspension Technology, Symposium sessions of February 1998 SAE Congress and Exposition, Detroit, MI, USA, 1998, pp. 201–208.