JOURNAL OF ADVANCED TRANSPORTATION
J. Adv. Transp. 2016; **50**:1735–1753
Published online 21 November 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/atr.1426

A genetic algorithm-optimized fuzzy logic controller to avoid rear-end collisions

Chen Chen[1,2]*, Meilian Li[1], Jisheng Sui[3], Kangwen Wei[1] and Qingqi Pei[1]

[1] *State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China*
[2] *Xidian-Ningbo Information Technology Institute, Ningbo, China*
[3] *Information and Communication Company, Jilin Electric Power Company Limited, Changchun, China*

SUMMARY

In this paper, a rear-end collision control model is proposed using the fuzzy logic control scheme. Through detailed analysis of car-following cases, our fuzzy control system is established with reasonable control rules. Furthermore, a genetic algorithm is introduced into the fuzzy rules refining process to reduce the computational complexity while maintaining accuracy. Numerical results indicate that our genetic algorithm-optimized fuzzy logic controller outperforms the traditional fuzzy logic controller in terms of better safety guarantee and higher traffic efficiency. Copyright © 2016 John Wiley & Sons, Ltd.

KEY WORDS: rear-end collision; fuzzy logic; genetic algorithm

1. INTRODUCTION

A statistical report from the General Estimates System and the Fatality Analysis Reporting System points out that rear-end collisions are most common in traffic accidents and are the cause of most traffic injuries and property damages [1]. To alleviate the growing risks of rear-end collisions, the auxiliary or automatic control methods look promising. Actually, there have been many classic automatic control schemes proposed, such as proportional–integral–derivative control [2], sliding mode control [3], and the linear quadratic optimal control [4]. Although these methods enable accurate control to be exercised, they are highly dependent on precise mathematical models. In practice, however, an accurate control model for vehicle collision avoidance is almost unavailable.

Instead of precise control, the fuzzy logic-based controllers (FLC) [5–9] are also well studied to reduce the possible vehicle collisions in the last few years. Because of its superiority in solving problems in multiparameter, nonlinear systems as well as capturing the driving characteristics in a vehicular environment, it is feasible and preferable to apply FLC-related models to vehicular active control issues. However, because the effect of an FLC relies on the number of fuzzy rules, an excessive number of such will directly impair its effectiveness. Moreover, the number of fuzzy rules increases exponentially with the number of fuzzy subsets, resulting in greater computational complexity of the controller. Among various schemes that have been proposed to address the aforementioned problems and has become an active research topic recently is combining neural network Back-Propagation (BP) with fuzzy control [7, 10]. However, the neural network converges slowly and is prone to be stuck in local optimum. Moreover, its training result is highly dependent on the original values. Unlike the neural network, the genetic algorithm (GA) is capable of searching for the global optimal solution and is almost application independent. To efficiently reduce the risk in a car-following scenario, this paper proposes a rear-end collision avoidance system that uses the fuzzy logic control model with its fuzzy rules optimized by the GA. This GA-based FLC (GFLC) should be able to allow drivers to promptly

*Correspondence to: Chen Chen, State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China. E-mail: cc2000@mail.xidian.edu.cn avoid approaching risks and accurately make a brake, as well as accommodating driving behaviors of drivers. Before giving out the principle of our proposed GFLC, a brief background of FLC and GA is first given with summarized advantages.

1.1. Fuzzy logic controller

Fuzzy logic controller [11] is one of the most common intelligent control models composed of the fuzzy linguistic variable, fuzzy set, and fuzzy reasoning module. It does not require a precise measurement or estimation to the controlled variables, thus providing an effective approach to challenges especially in the industrial control domains. A common structure of the FLC that consists of the fuzzification, fuzzy reasoning, and defuzzification process is shown in Figure 1.

According to previous studies, an FLC has the following advantages:

(1) Using linguistic descriptions, there is no need to mathematically model the controlled parameters.
(2) System robustness can be achieved by addressing the lagged time-varying, nonlinear, and other complex problems.
(3) It enables mathematical variables to be represented with linguistic variables and can describe expertise with fuzzy conditional statements.
(4) By using linguistic rules and heuristic knowledge, an FLC has the ability to simulate the people's way of thinking [12], which benefits its coping mechanism within the complex system.

1.2. Genetic algorithm

Genetic algorithm was introduced by Holland [13]. It illustrates the biological evolutionary mechanism and has been widely recognized in a broad variety of domains, like functional value optimization, pattern recognition, and control [14]. GA is a global search algorithm, and its main idea is as follows: an initial population is generated and then gradually yields optimal approximate solutions via evolution based on the natural law. Individuals with high adaptive values in each generation are retained based on the individual fitness through crossover and mutation, creating a new generation of individuals with higher fitness level. The individual with the highest fitness level in the last generation is regarded as the approximate solution to the problem. GAs are often viewed as function optimizers, although the range of problems to which GAs have been applied is quite broad. The advantages of the GAs can also be summarized as follows [15]:

(1) It can solve every optimization problem, which can be described with the chromosome encoding.
(2) It creates multiple solutions.
(3) Because the GA execution technique is not dependent on the error surface, it can solve multidimensional, non-differential, non-continuous, and even non-parametrical problems.
(4) Structural GA enables the possibility to solve the solution structure and solution parameter problems at the same time.
(5) The method is very easy to understand, and it practically does not demand the knowledge of mathematics.
(6) GAs are easily transferred to existing simulations and models.

From the aforementioned descriptions, it can be concluded that a wise combination of the FLC and GA model will provide a promising solution to our rear-end collision control problem and reach a good

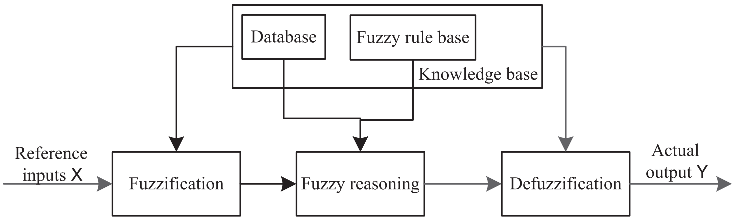


Figure 1. Structure of fuzzy logic controller.

trade-off between complexity and accuracy. Actually, many previous works have delved into solving vehicle collision problems using artificial intelligence algorithms. Milanés *et al.* [5] proposed an approach to the avoidance of rear-end collisions in congested traffic situations. In their work, two fuzzy controllers, a collision warning system and a collision avoidance system, have been developed to warn the driver or automatically avoid the possible collision during an emergency. Their controllers took the time to collision into consideration while ignoring the transient acceleration/deceleration of the trailing/leading cars. In this paper, we not only take into consideration the transient acceleration/deceleration of the trailing/leading cars, but also optimize the introduced fuzzy rules by turning to GA to yield fewer but the most effective rules. Mon and Lin [7] developed a supervisory recurrent fuzzy neural network control system that is composed of a recurrent fuzzy neural network controller and a supervisory controller. The former is investigated to mimic an ideal controller, while the latter is designed to compensate for the approximation error between the recurrent fuzzy neural network controller and the ideal controller. Although different from our system framework, their idea to refine the outputs of the fuzzy processing unit using a supervisory controller is similar to ours where the initialized fuzzy rules are further processed by the GA to improve the control efficiency and precision. The latest work of Yin and Wang [16] also investigated the vehicle rear-end collision avoidance issue by studying the safety distance mathematical model of proactive head restraint. The safety distance calculation is carried out by analyzing the fuzzy relationship between dynamical and kinematical parameters using a fuzzy logic model. Although their work seems to fully consider the internal and external factors resulting in a collision involving the driver's characteristics, weather conditions, road situations, and vehicle's characteristics, the thresholds they used for linguistic variable definition were a little arbitrary and were not given detailed explanations, which make their work applicable only under a specific condition. In [17], a fuzzy-based active control strategy to keep the vehicle away from possible collisions has been proposed by evaluating the crisis index of urban driving conditions. Because the crisis index or risk level is hard to calculate using mathematical formulas, the fuzzy logic controller is introduced here to generate the final risk evaluation index according to the input relative distance between the trailing and leading vehicles, the trailing vehicle speed, and the time to collision. Even though their work has taken the different crisis states into account, the fuzzy rules setting without further refinement process and real-test data verification seems arbitrary and imprecise.

In our work, the relative distance error and the relative speed error between the leading and trailing vehicles are introduced as the input variables of our fuzzy logic controllers, while the acceleration speed is determined as an output by which the leading/trailing vehicles could avoid collision through acceleration/deceleration. In addition, because of the dependence of the proposed FLC's performance on several factors (e.g., control rules and input/output membership functions), GA is used to further optimize the control rules by finding the optimal combination of input/output variables.

The remainder of this paper is organized as follows: Section 2 presents our proposed GFLC scheme in detail. The numerical results and performance evaluation are given in Section 3. Section 4 concludes this paper.

2. GENETIC ALGORITHM-BASED FUZZY LOGIC CONTROLLER

In practical cases, the factors that influence driving safety are very complicated and usually include weather conditions, road surface situations, response time of drivers, vehicle dynamics, and so on. Because most of these factors are nonlinear and time varying, obtaining an accurate mathematical model of the vehicle control system is difficult. In this context, some of the traditional methods (e.g., proportional–integral–derivative control and linear quadric-form optimal control) are unable to exercise accurate control. On the other hand, fuzzy control has advanced considerably in control and reason applications in recent years mainly because of its independence from accurate mathematical models. Its performance relies on the number of fuzzy rule bases, the number of linguistic variables as well as the types and ranges of membership functions. An excessive number of fuzzy control rule bases and linguistic variables results in longer FLC search time and higher computational complexity. Therefore, the key to FLC design is to determine the fuzzy control rule base and input/output membership function, which are usually acquired by expert summarization. However, in practical

DOI: 10.1002/atr

applications, an FLC generally cannot always guarantee its effectiveness, highlighting the need for optimization. By reviewing previous studies, there are many optimization schemes that optimize fuzzy control rules [18, 19] or the input/output quantified scale factors [20]. In this paper, GA is used to optimize the control rules by finding the optimal input/output variables combination. The flowchart of our proposed GFLC to avoid the rear-end collisions in car-following scenarios is shown in Figure 2.

To exercise effective control over vehicles, a dual-input single-output fuzzy controller is designed as Figure 2 illustrates. The relative distance error *ds* and the relative speed error *dv* between the leading and trailing vehicles are defined as the input variables. The output acceleration FA_d is defined as the output variable. The input variables are given as

$$ds = D - S, \tag{1}$$

$$dv = LV - FV \tag{2}$$

where *D* denotes the distance between the leading and trailing vehicles, *S* denotes the expected inter-vehicle space, *LV* denotes the speed of the leading vehicle, and *FV* denotes the speed of the following vehicle.

2.1. Fuzzification of input/output variables

In our work, the ranges of the input/output variables are properly set based on practical considerations. According to the braking distance algorithm from Mazda [21], we have

$$S = \frac{1}{2}\left(\frac{FV^2}{a_{-\max}} - \frac{LV^2}{a_{-\max}}\right) + FV\cdot\tau_1 + dv\cdot\tau_2 + d_0 \tag{3}$$

where τ_1 and τ_2 denote the delay. d_0 denotes the distance between the two vehicles after they stop. Generally, $\tau_1 = 0.1$ seconds, $\tau_2 = 0.6$ seconds, and $d_0 = 1.5$ m. In the case of the leading vehicle braking immediately at a speed of 120 km/hour (i.e., 33.33 meters/second), the expected inter-vehicle space is 67.5 m. So the relative distance error is set to the range from −67.5 to 67.5 m. According to the Chinese laws on the speedway [22], the vehicles drive at a speed varying from 60 to 120 km/hour, so the range of *dv* in our work is [−60 km/hour, 60 km/hour]. Taking into account the comfort of drivers and passengers, the acceleration of the trailing vehicle is set in the range [−8 meters/second^2, 8 meters/second^2].

The relative distance error *ds*, relative speed error *dv*, and acceleration FA_d are first fuzzified into seven fuzzy subsets: negative large (NL), negative medium (NM), negative small (NS), zero (Z), positive small (PS), positive medium (PM), and positive large (PL). Before the fuzzification, scaling transformations need to be made to the actual input variables such that they can be transformed into the specified domains. In this paper, this is carried out in a linear manner:

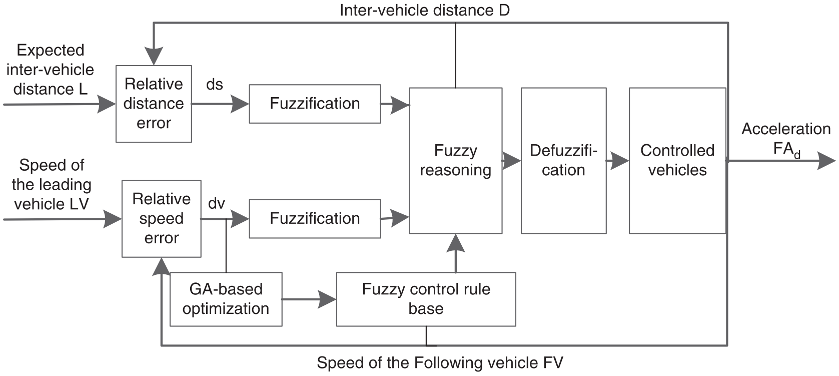


Figure 2. Genetic algorithm (GA)-based fuzzy logic controller.

$$x_0 = \frac{x_{\min} + x_{\max}}{2} + k\left(x_0^* - \frac{x_{\min}^* + x_{\max}^*}{2}\right), \tag{4}$$

$$k = \frac{x_{\max} - x_{\min}}{x_{\max}^* - x_{\min}^*} \tag{5}$$

where x_0^* denotes the actual input variable, $\left[x_{\min}^*, x_{\max}^*\right]$ denotes the range of the variable, $[x_{\min}, x_{\max}]$ denotes the specified domain of the variable, and k denotes the scaling factor. For the relative distance error *ds*, the scaling factor $k_1 = 0.05$. Using Equation (4), it can be determined that the fuzzy domain of the relative distance error is $[-6, 6]$. Because the range of the relative speed error *dv* is $[-16.67$ meters/second, 16.67 meters/second$]$ and the scale factor $k_1 = 0.36$, the fuzzy domain of the relative speed error is $[-6, 6]$ by Equation ((4)). Similarly, because acceleration FA_d is limited to $[-8$ meters/second^2, 8 meters/second$^2]$ and the scale factor $k_1 = 0.75$, its fuzzy domain is $[-6, 6]$.

2.2. Determination of membership functions

Next, the membership functions have to be determined to represent the degree of truth as an extension of valuation. In our work, the triangle function is used as the membership functions for the relative distance error *ds*, the relative speed error *dv*, and the acceleration FA_d. The triangle function used is

$$\eta(l, \delta_1, \delta_2, \delta_3) = \left\{ \begin{array}{cc} 0 & l \le \delta_1 \\ \frac{1 - \delta_1}{\delta_2 - \delta_1} & \delta_1 < l \le \delta_2 \\ \frac{\delta_3 - l}{\delta_3 - \delta_2} & \delta_2 < l \le \delta_3 \\ 0 & l > \delta_3 \end{array} \right\} \tag{6}$$

where δ_1 and δ_3 denote the "feet" of the triangle, δ_2 denotes the "peak" of the triangle, and $\eta(l)$ denotes the membership of the variable l. The domain of each linguistic variable is partitioned into seven parts, as shown in Figures 3–5.

2.3. Establishment of the fuzzy rules

To construct the fuzzy rules used to control the risky vehicles when a specific condition is met, the experience of drivers should be fully considered. According to the common response for a normal person, the following typical driving scenarios are taken into account:

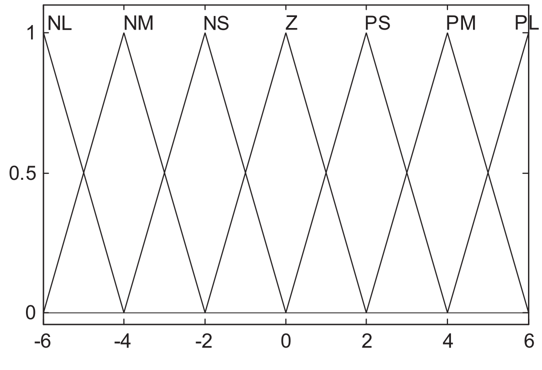


Figure 3. Membership function of *ds*. NL, negative large; NM, negative medium; NS, negative small; Z, zero; PS, positive small; PM, positive medium; PL, positive large.

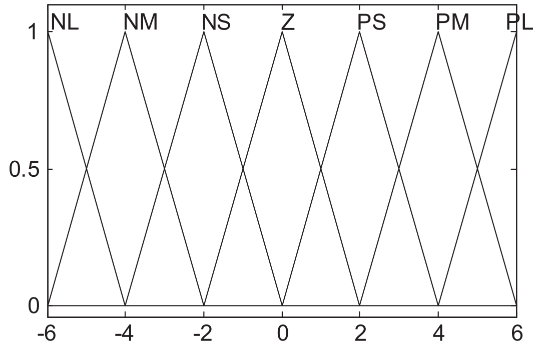


Figure 4. Membership function of *dv*. NL, negative large; NM, negative medium; NS, negative small; Z, zero; PS, positive small; PM, positive medium; PL, positive large.

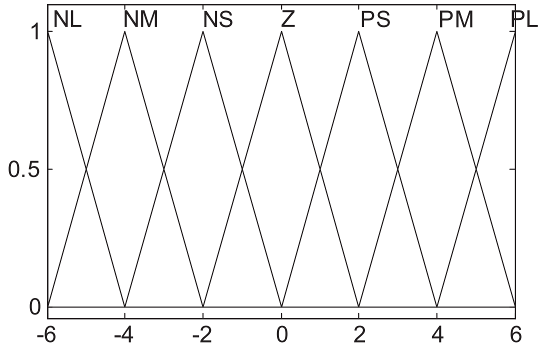


Figure 5. Membership function of FA_d. NL, negative large; NM, negative medium; NS, negative small; Z, zero; PS, positive small; PM, positive medium; PL, positive large.

1 If the inter-vehicle distance is small and the speed of the trailing vehicle is substantially larger than that of the leading vehicle, then rear-end collision has a high probability of occurrence. In this case, the trailing vehicle should brake immediately to decelerate for collision avoidance.
2 If the inter-vehicle distance is still large and the speed of the trailing vehicle is close to that of the leading vehicle, then rear-end collision has a small probability of occurrence. In this case, the speed of the trailing vehicle can remain or increase properly.
3 If the inter-vehicle distance is small and the speed of the trailing vehicle is close to that of the leading vehicle, then the driver of the trailing vehicle should brake slightly in order to effectively prevent rear-end collision and ensure driving safety.

Variants of the aforementioned three scenarios should be processed, taking into account both driving safety and driver's comfort. In some cases, comfort can be traded off for higher driving safety. Based on the aforementioned driving experiences as well as trial and error, a set of 49 rear-end avoidance fuzzy control rules representing the actual driving experience is devised and shown in Table I. The *g*th rule R^g is

$$\text{if } x_{ds} \text{ is } A^g \text{ and } x_{dv} \text{ is } B^g, \text{ then } y_{FA_d} \text{ is } C^g \tag{7}$$

where x_{ds} is the input value of *ds*, x_{dv} is the input value of *dv*, y_{FA_d} is the output value, A^g and B^g denote the fuzzy subsets of *ds* and *dv*, respectively. C^g denotes the fuzzy subset of FA_d. The fuzzy rule R^g can be regarded as the fuzzy implication $A^g \times B^g \rightarrow C^g$ over the product space $U_{fz} \times V_{fz}$, where $U_{fz} = A^g \times B^g$.

From Table I, it can be observed that the relation of *ds* and *dv* with FA_d is nonlinear. Because of its consistency with people's way of thinking, the fuzzy relation is eminently acceptable to drivers. Figure 6 shows the 3D curved surface corresponding to the fuzzy rules in Table I.

From the *dv* axis in the 3D curved surface, it can be seen that as the trailing vehicle's speed gets faster than the leading vehicle, the output deceleration increases until it reaches the mechanical

DOI: 10.1002/atr

Table I. Fuzzy control rules for the real-end avoidance system.

		Relative distance error						
Control variable		NL	NM	NS	Z	PS	PM	PL
Relative speed error	NL	NL	NL	NL	NM	NM	NS	NS
	NM	NL	NL	NM	NM	NS	NS	Z
	NS	NL	NM	NS	NS	NS	Z	Z
	Z	NM	NS	Z	NS	Z	PS	PS
	PS	NM	Z	Z	Z	Z	PM	PL
	PM	NS	Z	Z	PS	PM	PL	PL
	PL	NS	Z	Z	PS	PL	PL	PL

NL, negative large; NM, negative medium; NS, negative small; Z, zero; PS, positive small; PM, positive medium; PL, positive large.

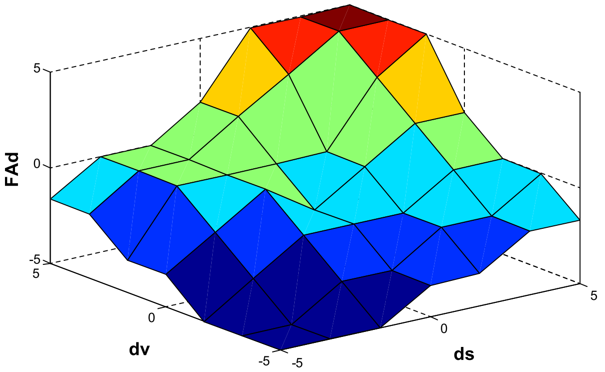


Figure 6. 3D curved surface of the fuzzy control rules.

maximum deceleration of the vehicle. From the *ds* axis, it can be seen that the output deceleration increases with a decrease in the inter-vehicle distance. Therefore, this 3D curved surface plot meets the control requirements for collision avoidance.

2.4. Optimizing to the fuzzy rules using genetic algorithm

The fuzzy rules determined in the previous subsection will be optimized with the GA to reduce the problem scale of our collision control strategy. Figure 7 is the flowchart of the GA used. The detailed steps are given as follows.

2.4.1. Coding of the fuzzy control rule

To guarantee consistency of the fuzzy rules, this paper sequentially fixes the two input variables and only encodes the output linguistic variables (i.e., the control variables). The linguistic values of the control variables encompass seven fuzzy sets "{NL, NM, NS, Z, PS, PM, PL}," which are denoted by 1, 2, 3, 4, 5, 6, 7 in the fuzzy control simulations, respectively. Next, the fuzzy control table in Table I is coded into the following matrix according to the aforementioned fuzzy sets in a left-to-right, top-to-bottom manner.

To optimize the fuzzy rules, we first encode it using the binary coding scheme. Each rule is represented by four binary numbers, the first of which is the control bit and the other three are the rule representation bits. The tandem coding method is employed by connecting these 49 rules to form a chromosome. According to our predefined fuzzy rules, these 49 rules will be the parameters to be optimized. The antecedents of the rules are *ds* and *dv*, and the consequent is FA_d. Because the control variable has seven fuzzy sets, the consequent of the rules can be represented via three-digit binary coding, that is, 000, 001, 010, 011, 100, 101, and 110. If a certain fuzzy rule does not exist, then let X denote the consequent of the rule and its code is 111. Thus, the 49 control rules in Table I can be numbered as rule1 to rule49, sequentially. Next, the aforementioned 49 rules can be connected in a

DOI: 10.1002/atr

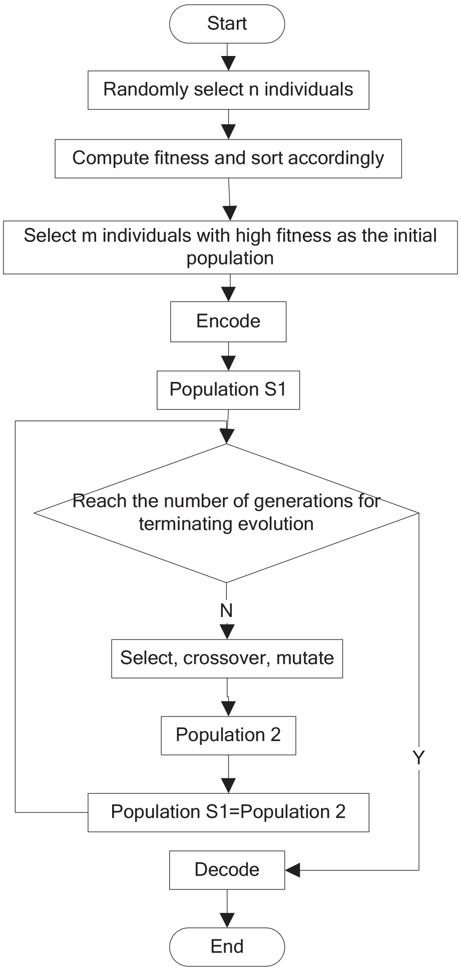

Start
Randomly select n individuals
Compute fitness and sort accordingly
Select m individuals with high fitness as the initial population
Encode
Population S1
Reach the number of generations for terminating evolution
N
Select, crossover, mutate
Population 2
Y
Population S1=Population 2
Decode
End

Figure 7. Genetic algorithm flowchart.

tandem manner to provide the chromosome of the rules as shown in Figure 8. The chromosome of the rules is the unit of the population, that is, the operand of GA. Each bit of the control gene controls the corresponding rule. As a result, the GA-based optimization yields the fuzzy rules consisting of fewer but more effective rules.

2.4.2. Determination of the fitness function

Because our FLC is designed to avoid rear-end collisions, the key to achieving this objective is to ensure that both the relative distance error *ds* and the relative speed error *dv* are small. Therefore, the objective function *J* in this paper is defined as

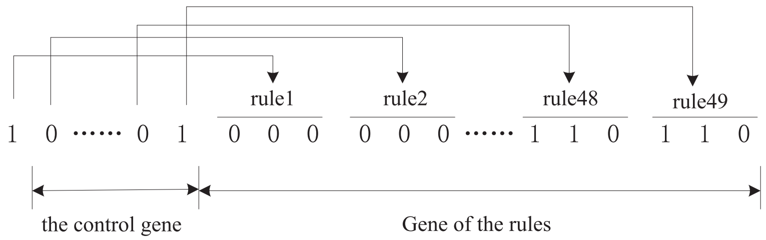


Figure 8. Chromosome of the rules.

DOI: 10.1002/atr

$$J = |ds| + |dv| \tag{8}$$

It is worth noting from Equation (8) that the smaller the value of J, the better the performance of our FLC. However, generally, the GA prefers individuals with high fitness. So our objective function needs to be properly transformed into a fitness function, which is given by

$$f = \frac{1}{1 + J} \tag{9}$$

2.4.3. Determination of the genetic parameters

The effectiveness of the GA relies on the related genetic parameters, including the population size n, the crossover probability P_c, and the mutation probability P_m. Usually, the large size of the initial population means a high space and time complexity. The size of the initial population is usually within the range [20, 100] [23]. In our work, the initial population is obtained in the following way:

(1) Randomly extract 80 matrixes with seven rows and seven columns as the matrixes of the control variables. Elements of all matrixes are integers ranging from 1 to 7.
(2) Compute the fitness f of each individual, and sort these n individuals in the descending order of f.
(3) Eliminate $m = 30$ individuals with small fitness, and define the remaining individuals as the initial population S1, thereby obtaining an initial population whose size $n_{S1} = 50$.

Note that each individual represents a chromosome of the rules. In this paper, the crossover probability P_c is 0.7. Because a small mutation probability P_m can effectively prevent damage to the good individuals, P_m is set to 0.001. The number of iterations is $G = 30$.

2.4.4. Determination of selection, crossover, and mutation operations

Biological evolution is achieved via genetic operations. Therefore, genetic operators are essential for optimization of fuzzy rules.

1 Selection operator. For this paper, the fitness proportional selection operator is used. The fitness is first calculated using the fitness function, and then, the replication probability of each individual is obtained. The number of replications of this individual in the next generation is equal to the product of the replication probability and the population size. Therefore, the higher the replication probability, the more replications in the next generation; otherwise, the individual has few replications in the next generation or is even eliminated. The probability P_{si} that each individual is selected is denoted by

$$P_{si} = \frac{f_i}{\sum_{j=1}^{n_{s1}} f_j} \tag{10}$$

where n_{S1} is the population size and f_i is the fitness of the ith individual in the population.

2 Crossover operator. During the crossover operation, the chromosomes before the cross point undergo the mutation operation, while those after the cross point undergo crossover and mutation operations. The crossover probability $P_c = 0.7$ and the crossover operation are illustrated in Figure 9.
3 Mutation operator. The basic-bit mutation method is used in our work. Because binary coding is carried out on the fuzzy control rules, the mutation operator involves reversing genes of some gene bits (e.g., 0 → 1 or 1 → 0), as shown in Figure 10.

The mutation probability $P_m = 0.001$. Note that the mutation operations improve the local search ability of GA and avoid premature convergence.

Finally, through the GA optimization, the number of the fuzzy rules has been reduced from 49 to 28, as shown in Table II.

DOI: 10.1002/atr

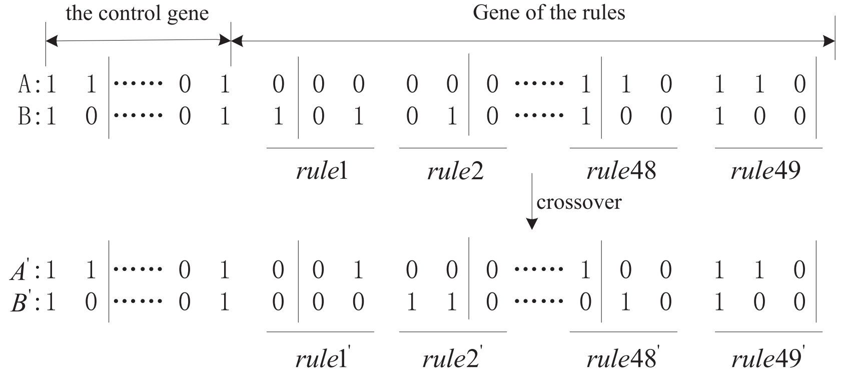

the control gene
Gene of the rules
A:1 1 ······ 0 1 0 0 0 0 0 0 ······ 1 1 0 1 1 0
B:1 0 ······ 0 1 1 0 1 0 1 0 ······ 1 0 0 1 0 0
rule1
rule2
rule48
rule49
crossover
A':1 1 ······ 0 1 0 0 1 0 0 0 ······ 1 0 0 1 1 0
B':1 0 ······ 0 1 0 0 0 1 1 0 ······ 0 1 0 1 0 0
rule1'
rule2'
rule48'
rule49'

Figure 9. Illustration of the crossover operation.

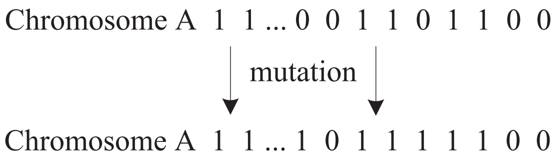

Chromosome A 1 1 ... 0 0 1 1 0 1 1 0 0
mutation
Chromosome A 1 1 ... 1 0 1 1 1 1 1 0 0

Figure 10. Illustration of the mutation operator.

Table II. Optimized fuzzy rule base.

		Relative distance error						
Control variable		NL	NM	NS	Z	PS	PM	PL
Relative distance error	NL	X	X	X	X	X	X	X
	NM	X	X	NS	NS	NM	NS	NS
	NS	X	NM	NS	NS	NS	Z	X
	Z	NM	NS	X	NS	PS	PS	PS
	PS	NM	PS	PS	Z	PS	PM	X
	PM	X	PS	PM	PS	Z	PL	X
	PL	X	PM	X	X	X	X	X

NL, negative large; NM, negative medium; NS, negative small; Z, zero; PS, positive small; PM, positive medium; PL, positive large.

2.5. *Defuzzification of the output variables*

After refining the fuzzy rules, the vehicle's acceleration can be computed based on the rules in Table II, and the accurate values of the acceleration are available via defuzzification. The defuzzification process takes the fuzzy sets and produces a single-value output. In this paper, the control variable is defuzzified via the gravity method:

$$Y_{\text{out}} = \frac{\sum_{g=1}^{7} l \cdot \eta_{C^g}(l)}{\sum_{g=1}^{7} \eta_{C^g}(l)} \tag{11}$$

where l, as we have stated before, is the value of the output in the domain, $\eta_{C^g}(l)$ is the membership of l with respect to the fuzzy set C^g.

Consider that the values of the relative distance error and the relative speed error are -3.5 and -1.6, respectively. The membership can be computed via the membership function η. Therefore, the

DOI: 10.1002/atr

membership of *ds* is $\eta_{NM}(-3.5)=0.75$ with respect to NM and $\eta_{NS}(-3.5)=0.25$ with respect to NS. The membership of *dv* is $\eta_{NS}(-1.6)=0.8$ with respect to NS and $\eta_{Z}(-1.6)=0.2$ with respect to Z.

According to the memberships of the input variables with respect to the fuzzy subset and the fuzzy rule table, three effective fuzzy rules can be concluded:

(1) If *ds* belongs to NM and *dv* belongs to NS, then the control variable FA_d belongs to NM;
(2) If *ds* belongs to NM and *dv* belongs to Z, then the control variable FA_d belongs to NS; and
(3) If *ds* belongs to NS and *dv* belongs to NS, then the control variable FA_d belongs to NS.

According to the Mamdani's inference rule [24], the membership of the control variable FA_d is

$$\eta_{NM}(0.75,,0.8)=\min(0.75,,0.8)=0.75, \tag{12}$$

$$\eta_{NM}(0.75,,0.2)=\min(0.75,,0.2)=0.2, \tag{13}$$

$$\eta_{NM}(0.25,,0.8)=\min(0.25,,0.8)=0.25 \tag{14}$$

By defuzzifying using the gravity method, the control variables can be accurately computed as

$$\begin{aligned} Y_{\text{out}} &= \frac{1}{0.75+0.75+0.2+0.2+0.25+0.25}\cdot \\ &\quad(-3.5{\cdot}0.75-4.5{\cdot}0.75-3.6{\cdot}0.2- \\ &\quad 0.4{\cdot}0.2-3.5{\cdot}0.25-0.5{\cdot}0.25) \\ &= -3.25. \end{aligned} \tag{15}$$

Instead of being used directly as the control quantity, Y_{out} needs to be converted into the actual range via scaling transform. When Y_{out} is -3.25, the actual control quantity is -4.33. That is, the trailing vehicle needs to decelerate at 4.33 meters/second^2 to avoid a possible rear-end collision.

3. SIMULATIONS

To validate the accuracy and effectiveness of our controller, GFLC is developed via MATLAB Simulink as shown in Figure 11. Note that this implementation is basically consistent with our system

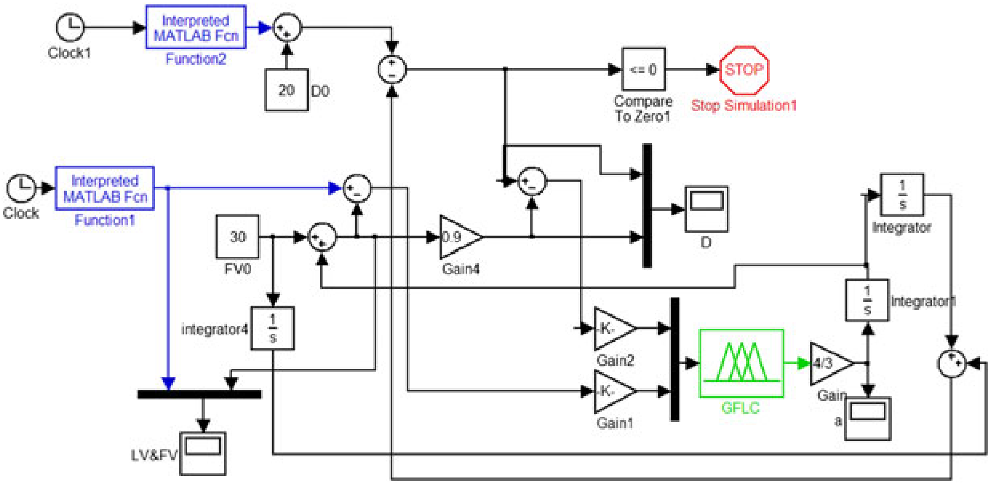


Figure 11. Simulation model of genetic algorithm-based fuzzy logic controller (GFLC).

DOI: 10.1002/atr

design structure shown in Figure 2, where function 2 calculates the relative distance error and function 1 generates the relative speed error. Afterwards, the errors between relative distance and speed will be fuzzified into different linguistic variables. Then, the "GFLC" module will be invoked to choose the corresponding fuzzy rule and execute the fuzzy reasoning, defuzzification, and GA-based optimization process one by one. Finally, the acceleration variable FA_d will be given as the FLC's output. The simulation will be halted if the two vehicles have collided; this will be decided by the comparator module on the top right corner.

To demonstrate the superiority of our GFLC controller, both FLC and GFLC are simulated under typical car-following scenarios. The simulation setup is as follows: the sampling time is 0.1 seconds; total simulation time is 80 seconds; initial distance between two vehicles is 20 m; the initial speeds of the leading and trailing vehicles are 20 and 30 meters/second, respectively. Note that in our simulation, dv is initially set to -10 meters/second to increase the possibility of rear-end collision. The leading vehicle keeps a constant speed before 20 seconds and then accelerates at 1 meters/second^2 during 20–30 seconds. During 31–33 seconds, the leading vehicle begins to brake at 4 meters/second^2. Note that the risk of rear-end collision with this configuration is very high between the leading and trailing vehicles, so that the trailing vehicle ought to brake immediately to avoid the collision. Numerical results are plotted in Figures 12–15 with the acceleration, speed, and inter-vehicle distance compared between FLC and GFLC. To obtain the statistical average, each test is repeated 50 times.

The comparison for accelerations of the trailing vehicle is shown in Figure 12 between FLC and the proposed GFLC. To further explore the difference between two models, a statistical analysis has been given using the Statistical Product and Service Solutions toolset to the two sampling datasets. At first, the normal distribution hypothesis is tested for the two datasets using the Kolmogorov–Smirnov method, and the results indicate that two datasets both follow nonparametric distributions. Note that the significance level is set to 0.05 for all tests. Next, from Figure 12, it can be observed that both controllers instruct the trailing vehicle to brake during the early period of the simulation for rear-end collision avoidance. Because we make the leading vehicle begin to brake at 4 meters/second^2 during 31–33 seconds, we will first check the response of two controllers during this period. By checking the statistics of FLC and GFLC during 31–33 seconds with a Mann–Whitney test, it is noted that there was significantly (M-W(U) = 38, p = 0.041) lower acceleration (with rank 9.45) for GFLC than for FLC (with rank 9.45). These statistics indicate that our GFLC could quickly respond to the detected dangers with sufficient speed reduction. In addition, it is noticeable that there is a sharp peak of accelerations around 16 seconds for the FLC model as shown in Figure 12. To explore the statistical difference of two controllers in the vicinity of this peak, we further check the dataset from the beginning to the end of this peak, that is, 13.8–20 seconds. During 13.8–20 seconds, the FLC outputs a sharp peak but the GFLC still recommends a slow deceleration. By checking the corresponding statistics with a Mann–Whitney test, it can be found that there was significant difference (M-W(U) = 49, p = 5.0749E-10) between two algorithms during 13.8–20 seconds. Note that the drastic speed increasing and decreasing of FLC will make the driver and passenger uncomfortable even injured in

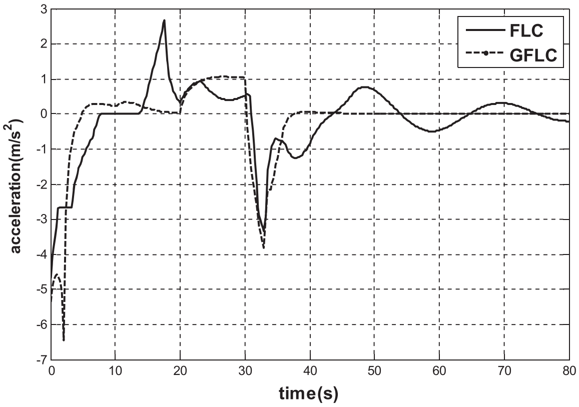


Figure 12. Comparison for accelerations of the trailing vehicle. FLC, fuzzy logic controller; GFLC, genetic algorithm-based fuzzy logic controller.

DOI: 10.1002/atr

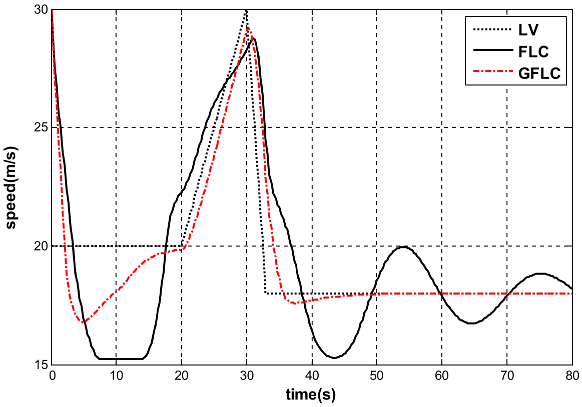


Figure 13. Comparison for transient speed of the trailing vehicle. LV, leading vehicle; FLC, fuzzy logic controller; GFLC, genetic algorithm-based fuzzy logic controller.

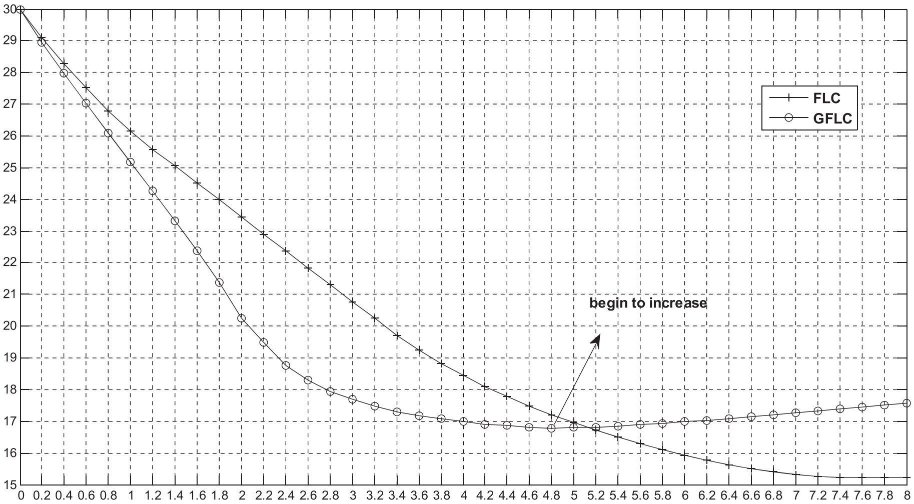


Figure 14. Comparison for transient speed of the trailing vehicle before 8 seconds. FLC, fuzzy logic controller; GFLC, genetic algorithm-based fuzzy logic controller.

practice. On the contrary, our GFLC could instruct the vehicle to drive smoothly and safely during the car-following process compared with the FLC. Through checking the dataset of GFLC, it can be noted that the trailing vehicle continues to brake till 37 seconds; after which, a positive accelerating occurrence implies that our GFLC considers the collision risk is now low. As a result, we further checked the period from 37 seconds to the end of simulation, that is, 80 seconds, and found that our GFLC almost keeps a zero acceleration for the trailing vehicle (with the mean and standard deviation only 0.009005 and 0.01716, respectively). However, during this period, FLC still yields nonzero accelerations that vibrate around 0 meters/second^2. The standard deviation of FLC during this period even exceeds 0.55 meters/second^2. This result also suggests that our optimized fuzzy rules are more rational and more precise than those of FLC.

The transient speed comparisons between two models are depicted in Figure 13. For clarity, the transient speed of the leading vehicle is also plotted. Note that in this scenario, both datasets also follow the nonparametric distributions verified by the Kolmogorov–Smirnov test. It can be observed that the leading vehicle keeps a steady speed before 20 seconds (with a 20 m/second mean and zero standard deviation), while the trailing vehicle adapts itself according to the instructions of the corresponding controllers to avoid collision. It is also worth noting that before 7.8 seconds, shown in Figure 14, FLC outputs continuous deceleration for the trailing vehicle, while GFLC outputs a

DOI: 10.1002/atr

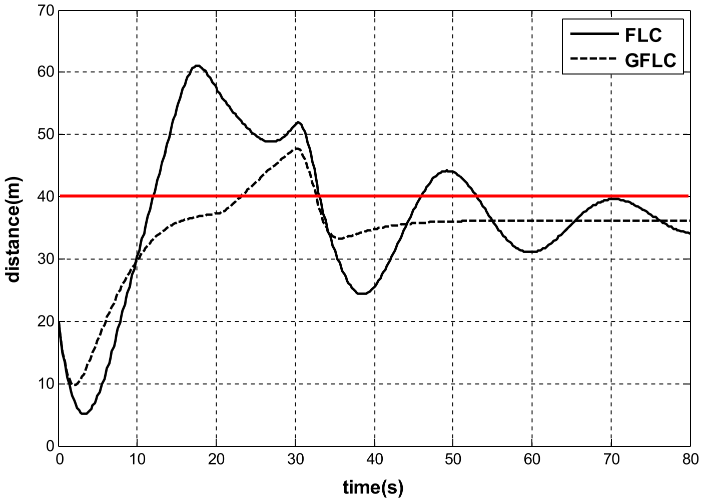


Figure 15. Comparison of inter-vehicle distance. FLC, fuzzy logic controller; GFLC, genetic algorithm-based fuzzy logic controller.

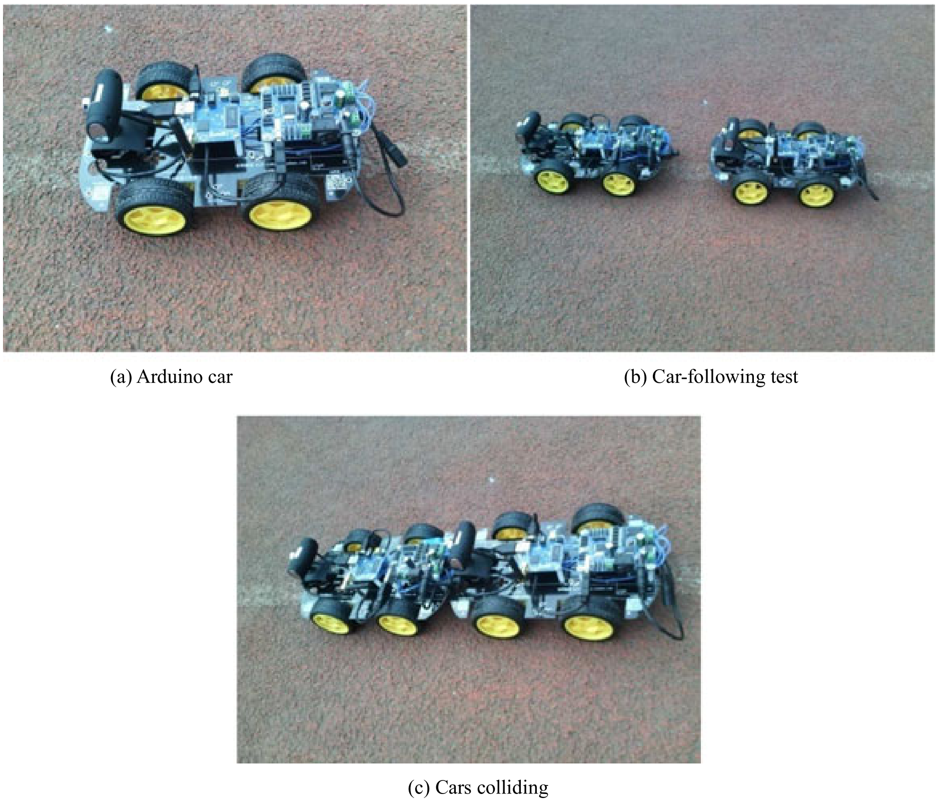
decrease before 4.8 seconds and then an increase till approximately 20 seconds. Actually, because the *dv* is initially set to −10 meters/second, the trailing vehicle will reduce the gap to zero between two vehicles after 2 seconds. Under such circumstance, our GFLC first outputs a deceleration to avoid the possible collision in advance. After a continuous deceleration before 4.8 seconds, the inter-vehicle distance returns to a safe level and then an acceleration has been given out by our GFLC till about 20 seconds. In comparison, FLC instructs the trailing vehicle to drive at a very low speed. The mean value during 8–13.8 seconds is only 15.2167 meters/second with a very small standard deviation, that is, 0.0013. This statistics indicates that FLC has overestimated the risk and result in a lower driving efficiency. A Mann–Whitney test to the two datasets showed that there is significant difference between two models (M-W(U) = 3392, $p = 0.00001$) and the speed recommended by GFLC (with rank 118.42) is more than FLC (with rank 84.58) during 0–20 seconds, implying a better driving efficiency of GFLC. During 20–30 seconds, because of the acceleration of the leading vehicle, both controllers suggest a fast speedup for the trailing vehicles. However, by checking the points carefully, it has been found that our GFLC outputs more precise controlling instructions that are almost strictly consistent with our simulation configurations. On the contrary, FLC instructs the trailing vehicle to accelerate too fast for too long (with a range of 13.536 and mean of 23.3017), say about 14–31 seconds, which may not conform to the real driving experiences of most drivers [25]. During 31–33 seconds of simulation, GFLC instructs the trailing vehicle to decelerate as soon as the leading vehicle decelerates, effectively avoiding the possible rear-end collision. With FLC, the trailing vehicle's speed changes relatively slowly, thus increasing the rear-end collision risk. During 34–80 seconds, GFLC enables the trailing vehicle to drive at an almost constant (with the standard deviation coefficient only 0.01569) rate of 17.9777 meters/second for mean, reducing the risk and at the same time improving the driving efficiency. With FLC, it shows great fluctuation (with the standard deviation 1.5645 and standard deviation coefficient 0.08673), indicating the instability of its adopted fuzzy rules.

To show the superiority of our controller from another side, we also compare the inter-vehicle distance of two models. It is well known that the distance between two vehicles directly influences the collision risk and the driving efficiency [26]. From Figures 12 and 15, it can be observed that the inter-vehicle distance actually changes with the variation of acceleration. With GFLC, the inter-vehicle distance changes more slightly (with the standard deviation and standard deviation coefficient equaling to 7.2458 and 0.2092, respectively) than FLC (with the standard deviation and standard deviation coefficient equaling to 12.6482 and 0.3417, respectively) during the entire simulation procedure and almost remains the same during 40–80 seconds (with the standard deviation only 0.3079 compared with 4.2854 for FLC). In contrast, FLC causes continuous fluctuations and outputs a higher standard deviation coefficient, that is, 0.1175 compared with only 0.008547 for GFLC. At about 4 seconds, FLC even outputs an inter-vehicle distance of no more than 6 m, which

DOI: 10.1002/atr

greatly increases the rear-end collision risk. Similarly, FLC also gives out a very large value, that is, over 60 m around 18 seconds, which is really unnecessary considering the continuous low speed of the trailing vehicle before. It is also worth noting that the lasting duration of inter-vehicle distance over 40 m (above the red line in Figure 15) exceeds approximately 27 seconds for FLC while it is only about 9 seconds for our GFLC. This large difference confirms once again that GFLC is more effective than FLC in terms of driving efficiency. A Mann–Whitney test to both datasets (following nonparametric distributions) indicates that there is significant difference between two models ($p = 0.001$).

To verify our numerical results in a real environment, a practical validation test is also implemented on two programmable autopilot Arduino cars as shown in Figure 16. The cars are driven and steered by two front wheels and are equipped with an Arduino Mega board, which is based on the ATmega2560 processor, a single-axis gyroscope and a two-axis accelerometer for attitude determination. Arduino [27] is an open-source electronics prototyping platform based on a flexible, easy-to-use hardware and software. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on processing). The spacing between a car and its preceding vehicle is measured by two infrared sensors (i.e., the actual spacing measurement is the averaged value of the two sensors). The longitudinal speed and acceleration are detected by an incremental encoder sensor that was installed on the shaft of the rear wheels and an acceleration sensor that was installed on top of the vehicle, respectively. The ATmega2560 processor is used as the real-time computing and control unit. It is worth noting that even robot–vehicle tests still have their limitations with regard to the credibility of the results. For instance, the vehicular dynamics on road and practical impact on drivers cannot be completely simulated through demonstrator robots. Therefore, the objective of our simulation is just verifying the correctness of the proposed model and algorithm; the application of GFLC to the real road environment still needs comprehensive tests and verifications before practical usage.



(a) Arduino car

(b) Car-following test

(c) Cars colliding

Figure 16. Collision avoidance test bed based on Arduino cars.

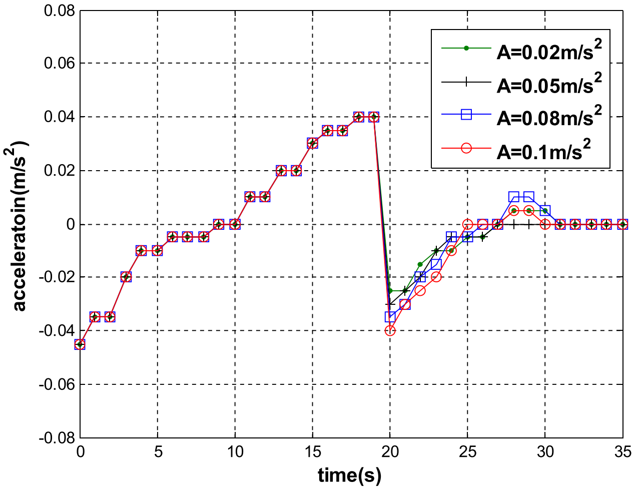


Figure 17. The accelerations of the trailing car under different cases

Our test configurations are as follows:

(1) The initial distance between two cars is 5 m. The initial speeds of the leading and trailing cars are 0.3 and 0.5 meters/second, respectively.
(2) The leading car keeps a constant speed before 20 seconds and then brakes during 20–23 seconds with a deceleration of 0.02, 0.05, 0.08, and 0.1 meters/second^2, respectively.
(3) Results of each test are traced into a log file every second. To obtain the statistical average, each test is repeated 15 times.

Figure 17 shows the output accelerations of the trailing car under different cases, where A denotes the adopted accelerations during 20–23 seconds. Before 20 seconds, it can be noted that there is no apparent difference among the four groups of tests with A varying. Because dv in these cases is also negative, that is, -0.2 meters/second, our GFLC controller suggested a continuous deceleration before 10 seconds to reduce the possible rear-end collision risk. After 10 seconds, because two cars have kept a safe distance, a gradual acceleration is instructed to increase the driving efficiency on the condition that safety is guaranteed. After 20 seconds, because different decelerations for the leading car are executed, our GFLC controller also outputs different accelerations for the trailing car. It is worth noting that our controller outputs different decelerating strength at 20 seconds according to the values

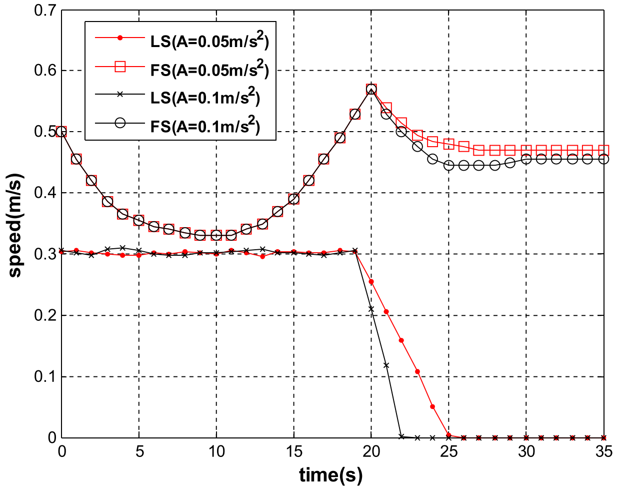


Figure 18. Speeds of the leading and trailing cars under different cases.

DOI: 10.1002/atr

of A. These results are reasonable considering the risk levels caused by the sudden braking of the leading car. After the immediate deceleration, a continuous decelerating is suggested by our GFLC controller but with the deceleration value slowly increasing. After approximately 31 seconds, all cases output a zero acceleration because the leading car has actually stopped, except for the case where $A = 0.02$ meters/second^2.

The sampled speed per second for both leading and trailing cars is plotted in Figure 18 with different settings of A. Because the speed of cars are captured by the incremental encoder sensor and logged into a file every second, the speeds before 20 seconds have a little deviation from the theoretical values. Note that the speed of the trailing car after 20 seconds is in direct proportion to the value of A, where a larger A will result in a lower suggested speed. Actually, by comparing the results of Figures 17 and 18, it can be concluded that both figures are consistent to show the same actions of the car-following process. For example, after approximately 31 seconds, the trailing car almost keeps a constant speed in Figure 18, which is verified by the accelerations in Figure 17 during the same period. After 15 simulations, no collision occurred so that safety of the two cars has been guaranteed using our GFLC model.

4. CONCLUSIONS

In this paper, a novel fuzzy controller is proposed, which leverages GA to optimize the fuzzy control rules, to improve the performance on rear-end collision avoidance of the vehicular control system, and enhances the emergency response ability. Simulation results successfully indicate that the proposed controller outperforms the traditional one and is capable of avoiding the possible rear-end collision while increasing the traffic efficiency. Our future work will attempt to incorporate the advanced sensor system into our vehicular dynamics control framework and propose an effective active-control system. This could further reduce collision risks by introducing precise prediction algorithm depending on the online data collection module.

5. LIST OF SYMBOLS AND ABBREVIATIONS

ds	relative distance error
dv	relative speed error
FA_d	output acceleration
D	the distance between the leading and trailing vehicles
S	the expected inter-vehicle space
LV	the speed of the leading vehicle
FV	the speed of the following vehicle
τ_1,τ_2	delay
d_0	the distance between the two vehicles after they stop
x_0^*	the actual input variable
$[x_{\min}^*, x_{\max}^*]$	the range of the variable
$[x_{\min}, x_{\max}]$	the specified domain of the variable
k	the scaling factor
δ_1,δ_3	the "feet" of the triangle
δ_2	the "peak" of the triangle
$\eta(l)$	the membership of the variable l.
x_{dx}	the input value of ds
x_{dv}	the input value of dv
Y_{FA_d}	output value
A^g,B^g	the fuzzy subsets of ds and dv
C^g	the fuzzy subset of FA_d
R^g	fuzzy rule
J	objective function
n_{s1}	population size
f_i	the fitness of the ith individual in the population

DOI: 10.1002/atr

P_{si}	the probability of each individual is selected
P_c	crossover probability
P_m	mutation probability
l	the value of the output in the domain
$\eta c^g(l)$	the membership of l with respect to the fuzzy set C^g
FLC	fuzzy logic-based controllers
BP	back-propagation
GA	genetic algorithm
GFLC	GA-based FLC
NL	negative large
NM	negative medium
NS	negative small
Z	zero
PS	positive small
PM	positive medium
PL	positive large

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (61201133, 61571338), the National Science and Technology Major Project of the Ministry of Science and Technology of China (2015zx03002006-003), the Natural Science Foundation of Shaanxi Province (2014JM2-6089), the National High-tech R&D Program of China (863 Program-2015AA015701), the Ningbo Huimin projects of science and technology (2015C50047), the Research collaboration innovation program of Xi'an (CXY1522-3), the state grid cooperation project (HX0115014015), and the "111 Project" of China (B08038).

REFERENCES

1. 2011 *Traffic Safety Facts FARS/GES Annual Report (Final Edition)*. U.S. DOT: Washington, DC, USA, DOT HS 811 754, 2013.
2. Moon S, Moon I, Yi K. Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance. *Control Engineering Practice* 2009; **17**(4): 442–455.
3. Gracia L, Garelli F, Sala A. Reactive sliding-mode algorithm for collision avoidance in robotic systems. *Control Systems Technology, IEEE Transactions on* 2013; **21**(6): 2391–2399.
4. Van den Berg J *et al.* LQG-obstacles: feedback control with collision avoidance for mobile robots with motion and sensing uncertainty. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* IEEE, 2012; **44**(8): 346–353.
5. Milanés V *et al.* A fuzzy aid rear-end collision warning/avoidance system. *Expert Systems with Applications* 2012; **39**(10): 9097–9107.
6. David KKA *et al.* Unmanned underwater vehicle navigation and collision avoidance using fuzzy logic. In *System Integration (SII), 2013 IEEE/SICE International Symposium on* IEEE: USA, New York, 2013.
7. Mon Y-J, Lin C-M. Supervisory recurrent fuzzy neural network control for vehicle collision avoidance system design. *Neural Computing and Applications* 2012; **21**(8): 2163–2169.
8. Pinto, AC *et al.* Anti-collision fuzzy logic system for automobile vehicles. 2013, SAE Technical Paper.
9. Ibrahim MI *et al.* Mobile robot obstacle avoidance in various type of static environments using fuzzy logic approach. In *Electrical, Electronics and System Engineering (ICEESE), 2014 International Conference on* IEEE, 2014: 83–88.
10. Hui NB, Mahendar V Pratihar DK. Time-optimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approaches. *Fuzzy Sets and Systems* 2006; **157**(16): 2171–2204.
11. Berenji HR. Fuzzy logic controllers. In *An Introduction to Fuzzy Logic Applications in Intelligent Systems*. Springer: Germany, Heidelberg, 1992: 69–96.
12. Rivero LC *et al.* Fuzzy logic and RULA method for assessing the risk of working. *Procedia Manufacturing* 2015; **3**: 4816–4822.
13. Holland JH. Genetic algorithms. *Scientific American* 1992; **267**(1): 66–72.
14. Grefenstette JJ. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*. Psychology Press: The UK, Abingdon, 2013.
15. Man K-F, Tang KS, Kwong S. *Genetic Algorithms: Concepts and Designs*. Springer Science & Business Media: Germany, Heidelberg, 2012.

16. Yin X, Wang M. Research on safety distance mathematical model of pro-active head restraint in rear-end collision avoidance system. *International Journal of Security and its Applications* 2015; **9**(1): 347–356.
17. Li X, Shaobin W, Li F. Fuzzy based collision avoidance control strategy considering crisis index in low speed urban area. In *Transportation Electrification Asia-Pacific (ITEC Asia-Pacific), 2014 IEEE Conference and Expo* IEEE, 2014: 1–6.
18. Hwang, H-S *et al.* Identification of fuzzy control rules utilizing genetic algorithms and its application to mobile robots. Algorithms and Architectures for Real-Time Control (Korea, 1992), 2014: 249–254.
19. Jalali A *et al.* Model-free adaptive fuzzy sliding mode controller optimized by particle swarm for robot manipulator. *International Journal of Information Engineering and Electronic Business (IJIEEB)* 2013; **5**(1): 68.
20. Betin F *et al.* Determination of scaling factors for fuzzy logic control using the sliding-mode approach: application to control of a DC machine drive. *IEEE Transactions on Industrial Electronics* 2007; **54**(1): 296–309.
21. Doi A *et al.* Development of a rear-end collision avoidance system with automatic brake control. *JSAE Review* 1994; **15**(4): 335–340.
22. Zhang W *et al.* Road safety in China: analysis of current challenges. *Journal of Safety Research* 2010; **41**(1): 25–30.
23. Fan L, Joo EM. Design for auto-tuning PID controller based on genetic algorithms. In *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on* IEEE, 2009: 1924–1928.
24. Mamdani EH. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers* 1977; **100**(12): 1182–1191.
25. Schmidt RA. Unintended acceleration: a review of human factors contributions. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 1989; **31**(3): 345–364.
26. Chen C *et al.* Improving driving safety based on safe distance design in vehicular sensor networks. *International Journal of Distributed Sensor Networks* 2012; **2012**(4): 3485–3490.
27. Banzi M, Shiloh M. *Make: Getting Started with Arduino: The Open Source Electronics Prototyping Platform*. Maker Media, Inc: San Francisco, CA, 2014.

DOI: 10.1002/atr