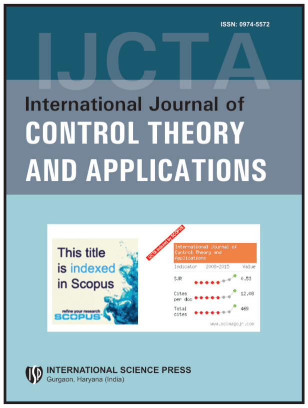


International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 25 • 2017

Ensemble FLC with Optimized Rule Base for Full Range Vehicle Longitudinal Control

Wisemin Lins.A.[a]

[a]*Department of Electrical and Electronics Engineering, Vels University, Chennai. Email: wisemineee@gmail.com*

Abstract: Utmost responsibility of Vehicle Longitudinal Controller (VLC) designed for autonomous driving is to provide the vehicle occupant both safety and comfort in urban, rural and highway operating zones. In this research work, heuristic based fuzzy controller with rule base formulated using soft computing technique (Genetic Algorithm (FRB-GA), Particle Swarm Optimization (FRB-PSO) and Differential Evolution (FRB-DE)) is proposed to achieve a balance between comfort and safety for all operating zones. New performance metric, Total Cost Function Index (TCFI) which comprises of occupant safety factor and comfort factor along with minimizing maximum penetrating distance (Max PD) is proposed. Standard driving patterns and simple manoeuvres for each operating zones are used as test scenarios. On observing the result, FRB-PSO performed well in highway and FRB-DE performed well in urban and rural operating zones. Hence this research paper proposes an Ensemble Fuzzy Logic Controller architecture for Full Range Vehicle Longitudinal Control (EFLC-FRVLC). It uses two parallel connected fuzzy controllers. Rule base of the first and the second controller is loaded with FRB-PSO and FRB-DE respectively. Proposed controller shows an improved performance in all operating zones. Results obtained from Model in Loop testing using CARMAKER - MATLAB/Simulink vindicates the performance of the proposed controller.

Keywords: Vehicle longitudinal controller, passenger safety and comfort, fuzzy logic controller rule base formulation, genetic algorithm, particle swarm optimization, differential evolution.

1. INTRODUCTION

Accidents on the roads can be reduced to a larger extend when drivers are provided assistance with intelligent systems like Vehicle Longitudinal Controller (VLC) while driving through severe traffic condition and during long tiring driving. VLC operates in Distance Control Mode (DCM) to avoid rear end collision whenever a lead vehicle is present in the same lane of travel and operates in Velocity Control Mode (VCM) when there is no lead vehicle. Conventional Adaptive Cruise Controller (ACC) can operate above 8.33 m/sec (30 km/hr) which limits its usage to highway operating zone [9]. For velocities between 8.33 m/sec and 1.38 m/sec (5 km/hr), Low Speed ACC (LS-ACC) was developed which overcomes the drawback of ACC and supports the driver during low speed driving (rural operating zone). LS-ACC system needs the intervention of the driver to reinitiate the movement of the vehicle, whenever the vehicle stops [11]. Repeated application of throttle, brake and clutch are required in urban operating zone, which leads to severe straining of driver's leg. Stop and Go Low Speed ACC (S&G-ACC) was developed where the system automatically reinitiate the movement of the vehicle after a complete halt/stop depending on the movement of the vehicle travelling in front of the host vehicle [1]. Utmost responsibility of the VLC is to provide safety and comfort to the occupants of the vehicle in all operating zones. Acceleration and Jerk of the vehicle defines the comfort level experienced by the vehicle occupants. According to [7], bounded longitudinal acceleration and jerk can assure certain degree of comfort in longitudinal control, especially in Stop and Go scenarios. Depending on the velocity of travel (i) Constant Time Gap (CTG) spacing policy and (ii) Variable Time Gap (VTG) spacing policy describes about the proper Inter Vehicle Distance (IVD) that needs to be maintained between the vehicles traveling in the same lane. CTG assumes that both lead and the host vehicle have identical and constant decelerations during braking manoeuver. This assumption produces solutions with high jerks and consequently low comfort [7]. In addition, this model could produce undesirable large inter-distances reducing the lane capacity. Among the various controllers present, heuristic approach based Fuzzy Logic Controller (FLC) permits designing of controller for the system without extensive knowledge of the equations of the process and it represents in a very effective way the human reasoning methods. Hence this research work focuses on using FLC for VLC. FLC for VLC with two inputs (Distance Error, D_e and Velocity Error, V_e), two outputs (Brake command and Velocity set point ratio), and seven triangular membership functions for both the inputs and seven singleton membership functions for the output and 42 rules are proposed in [11]. Tsai et. al., in [3] had used the same set of input variables with five triangular membership functions for the input and output. The author had used brake and throttle (Gas) as output variables. The rule base was populated with 25 rules. Several research articles for VLC application had shown the usage of FLC with triangular membership function with Distance Error and Velocity Error as input and throttle/brake command as output. According to [9], if these errors are maintained minimum during DCM, and if the IVD is less during low speed operation [8] then number of vehicles which could be accommodated in the lane during heavy traffic condition can be increased meanwhile avoiding rear end collision. Rest of this article is organized as follows: Research method that this research work addresses is explained in section 2 followed by the design of FLC in section 3. Section 4 describes about the test scenarios and the rule base formulation using soft computing techniques along with the comparison and discussion of the test result. The proposed Ensemble Fuzzy Logic Controller for Full Range Vehicle Longitudinal Control (EFLC-FRVLC) along with Model in Loop (MIL) testing for validating the proposed controller is discussed in Section 5. The overall performance and the outcome of the research work are summarized in the conclusion.

The Research Method

Controllers designed so far for the VLC, was not optimized to offer both safety and comfort. [14] and [5] considered the velocity and distance error components for validating the performance of the vehicle. These parameters deal with safety factor and not the comfort of the occupant. Hence, this research work proposes a Total Cost Function Index (TCFI) as mentioned in equation (1). *TCFI* has, (a) Safety component and (b) Comfort component. Safety component comprises of velocity and distance errors. Maintaining a low value of this component assures safety to vehicle occupant by avoiding rear end collision. Comfort component comprises of vehicle's acceleration and jerk. Proper controlling of this component provides comfort to the occupant.

$$\mathrm{TCFI} = \mathrm{V}_{mse} + \mathrm{D}_{mse} + \mathrm{A}_{ms} + \mathrm{J}_{ms} \tag{1}$$

$$\mathrm{V}_{mse} = \frac{1}{\mathrm{T}}\int_0^t \mathrm{V}_e^2 \, dt \tag{2}$$

$$\mathrm{V}_e = \mathrm{T}_{vel} - \mathrm{H\_V}_{vel}$$

$$\mathrm{D}_{mse} = \frac{1}{\mathrm{T}}\int_0^t \mathrm{D}_e^2 \, dt \tag{3}$$

$$D_e = D_{dist} - M_{dist}$$

$$A_{ms} = \frac{1}{T}\int_0^t Ac_h^2 \, dt \quad (4)$$

$$J_{ms} = \frac{1}{T}\int_0^t J_h^2 \, dt \quad (5)$$

Objective function: minimize (TCFI, Max PD) (8)

where, V_{mse} is the mean square velocity error, D_{mse} is the mean square distance error, A_{ms} is the mean square acceleration, J_{ms} is the mean square jerk, V_e is the velocity error, T_{vel} and H_V_{vel} are the target and host velocities in m/sec, D_e is the distance error, D_{dist} and M_{dist} are the expected and inter-vehicle distance maintained in m, Ac_h is the host vehicle acceleration in m/sec^2, J_h is the host vehicle jerk in m/sec^3 and *Max PD* is the instantaneous maximum value of D_e in *m* when $D_{dist} > M_{dist}$. In order to achieve a balance between comfort and safety *TCFI* and *Max PD* needs to be minimum. Safety and comfort being contrary parameters in vehicle operation, formulating the rule base using expert knowledge is tough [10] and of less prominence [2]. Rule base formulated with an expert application knowledge of the system may not provide an optimal balance of safety and comfort. Hence this research work, proposes the use of soft computing techniques to formulate the rule base for obtaining the optimal balance.

Table 1
Velocity and typical stopping distances

Velocity (m/s)	*Perceiving Distance (Meter)*	*Braking Distance (Meter)*	*Overall Stopping Distance (Meter)*
8.94	6	6	12
13.4	9	14	23
17.9	12	24	36
22.4	15	38	53
26.8	18	55	73
31.3	21	75	96

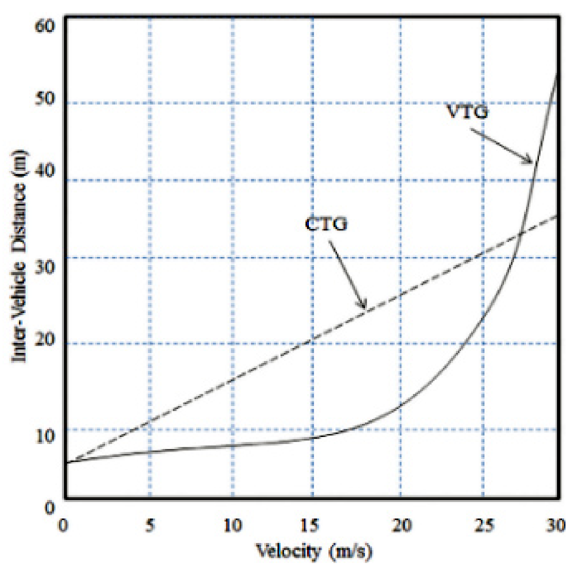


Figure 1: Velocity vs IVD policy

New spacing policy (Table 1) is framed considering the total time required to perceive the obstacle and apply the brake for bringing the vehicle to control. This spacing policy maintains a low but safer distance at low speed operations (Figure 1) when compared to CTG spacing policy.

Design of Fuzzy Logic Controller

Fuzzy control system consists of fuzzification, inference engine and defuzzification to obtain the crisp output (control signal). Set of Linguistic Statements (LS) characterize the fuzzy system. These statements are represented in the form of "*IF–THEN*" conditional statements known as fuzzy rules. *IF* part of the fuzzy rule, handles the input and *THEN* part handles the output. Figure 2 shows a fuzzy structure for VLC with two input, single output is developed for this research work.

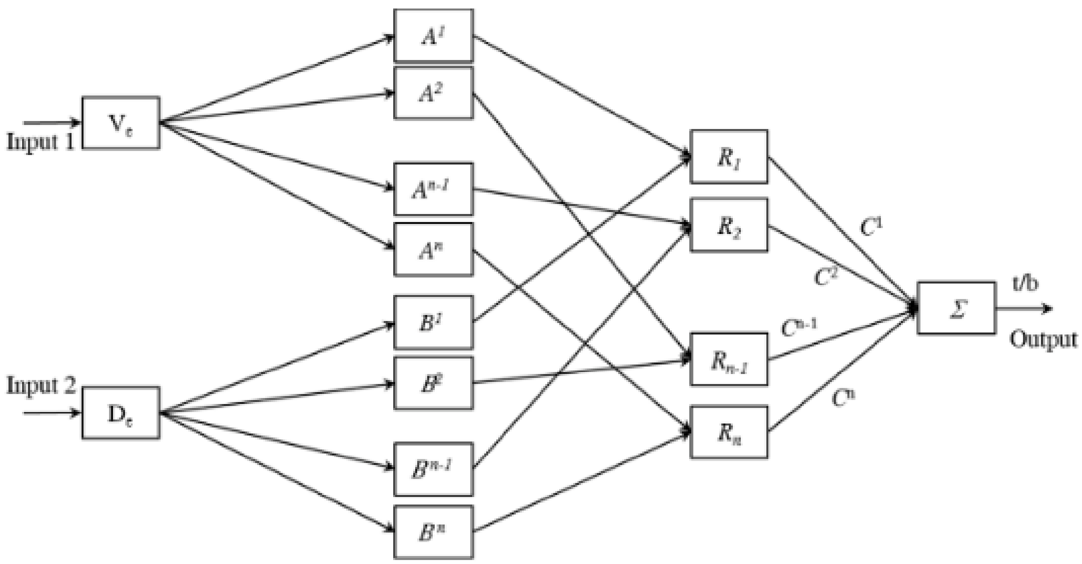


Figure 2: Fuzzy structure for VLC

A^n and B^n are the antecedents of V_e and D_e respectively. n denotes the number of linguistic variables used. C^n and R_n represents the consequent and the rules of the fuzzy controller respectively. The input V_e and D_e ranges between +38 m/sec and +96 m respectively. Output variable t/b varies between +1. Nine Linguistic values are taken for each input and output variables. Acceleration, Jerk of the vehicle and the error components mentioned in equation (1) can be controlled by the crisp value (9) obtained as an output from FLC. The crisp output of the controller is given by

$$t/b = \frac{\Sigma_i \omega_i \hat{C}_i}{\Sigma_i \omega_i} \tag{9}$$

where, ω_i and ω represent the weight and the crisp value respectively. Set of rules in the rule base of the fuzzy controller, predominantly influence the crisp output (t/b).

Fuzzy Rule Base Formulation and Testing

Based on the number of input variables, output variables and the number of membership functions, 729 rules could be framed. But majority of them would be trivial. Hence soft computing techniques are used to formulate the rule base which are intended to provide the optimum balance between safety and comfort to the occupant of the vehicle. MATLAB optimization framework for the proposed FRVLC is shown in Figure 3. The vehicle longitudinal model available in MATLAB/Simulink is used as the plant. The script for designing, developing and formulating the fuzzy controller is written in MATLAB.

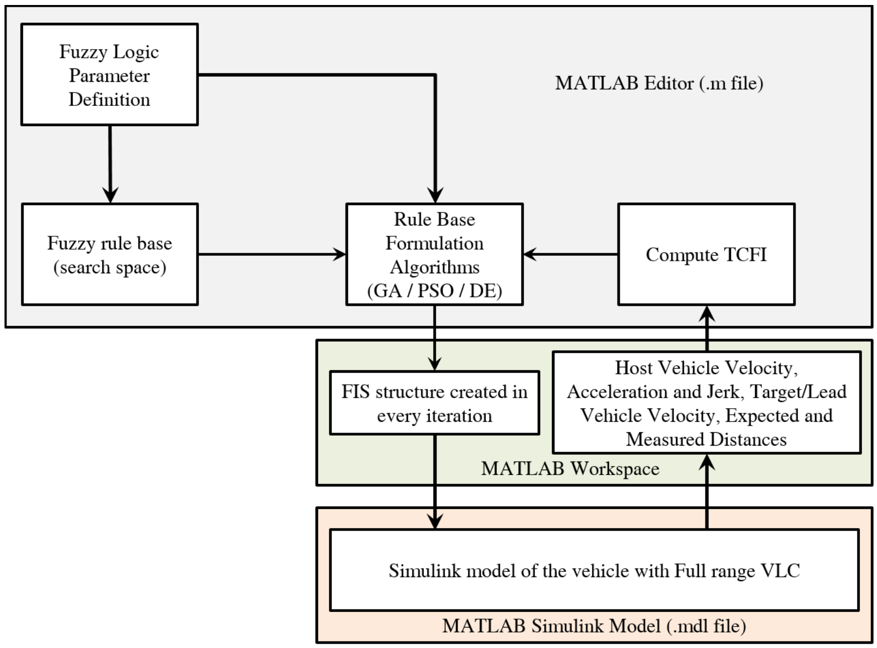
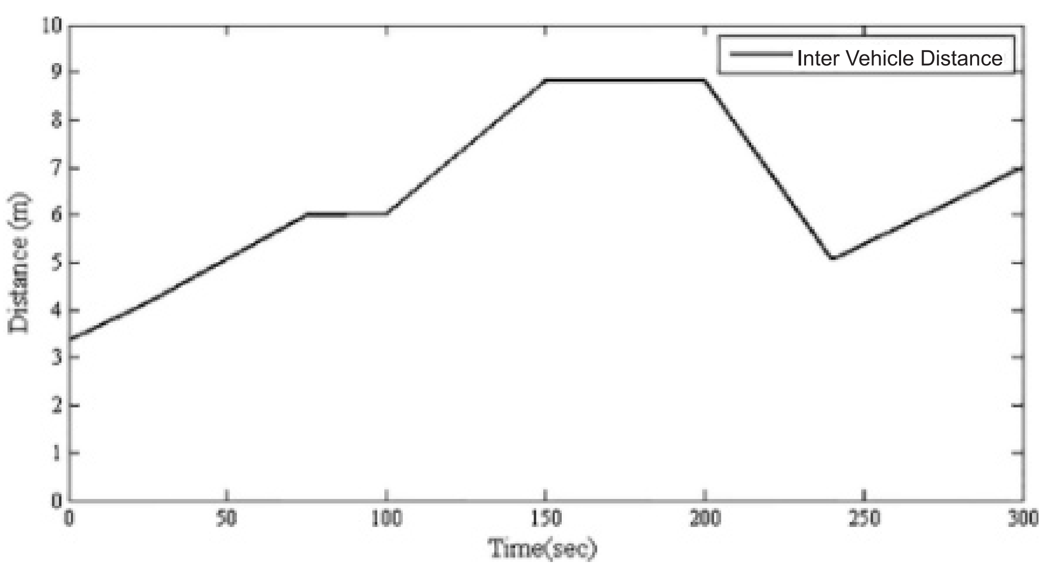


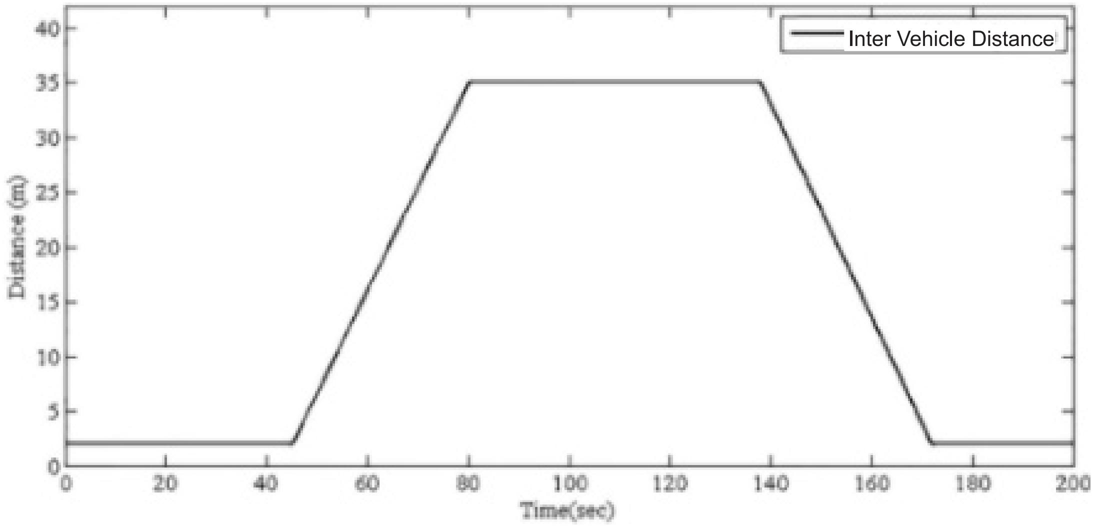
Figure 3: Optimization framework for FRVLC

2. TEST SCENARIOS

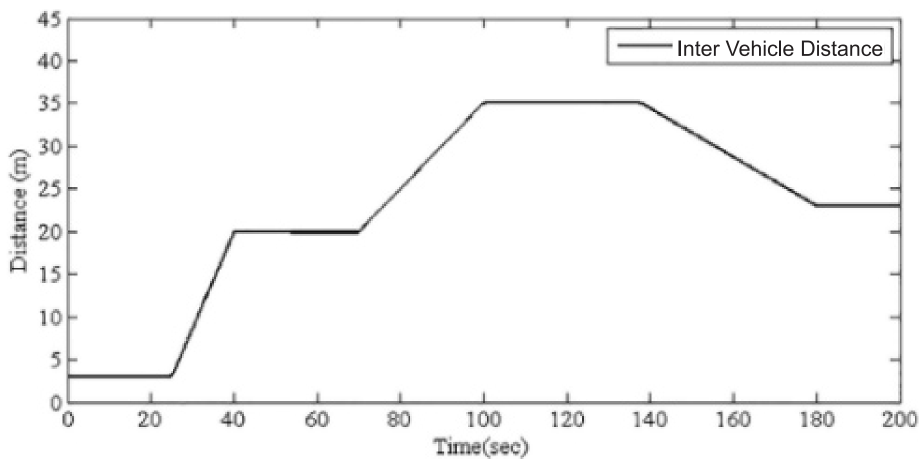
As the driving behaviour (acceleration rate, max. velocity, breaking severity, cruising period, etc.,) changes within and for different operating zones, vehicle manoeuvring (IVD pattern) as shown in Figure 4 (a-d) and driving pattern (standard driving pattern of a region) as shown in Figure 5 (a-c) are used as test scenarios. These test scenarios are provided as an input for formulating a rule base for achieving optimal balancing of comfort and safety in all operating zones.



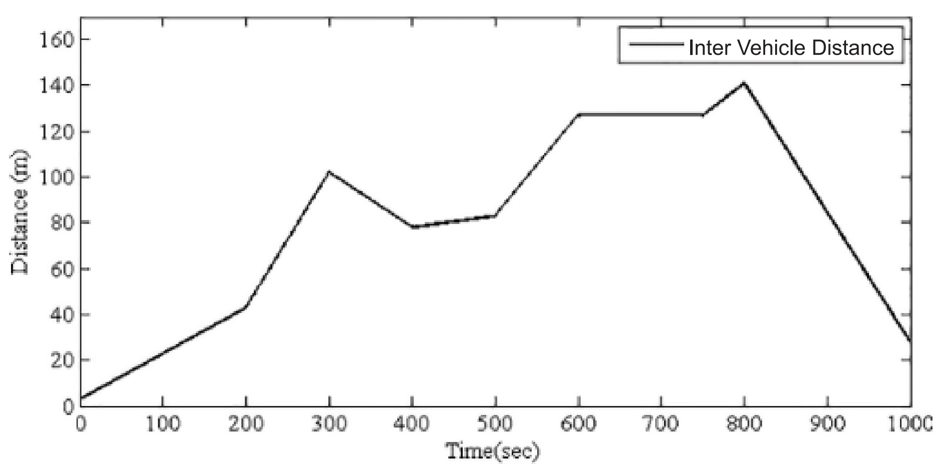
(a) Low speed manoeuvring



(b) Stop and go manoeuvring

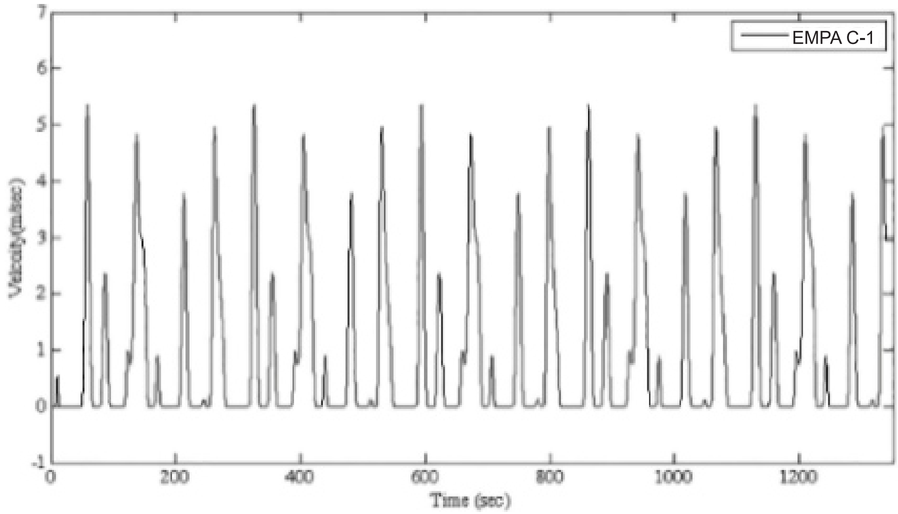


(c) Mid speed manoeuvring

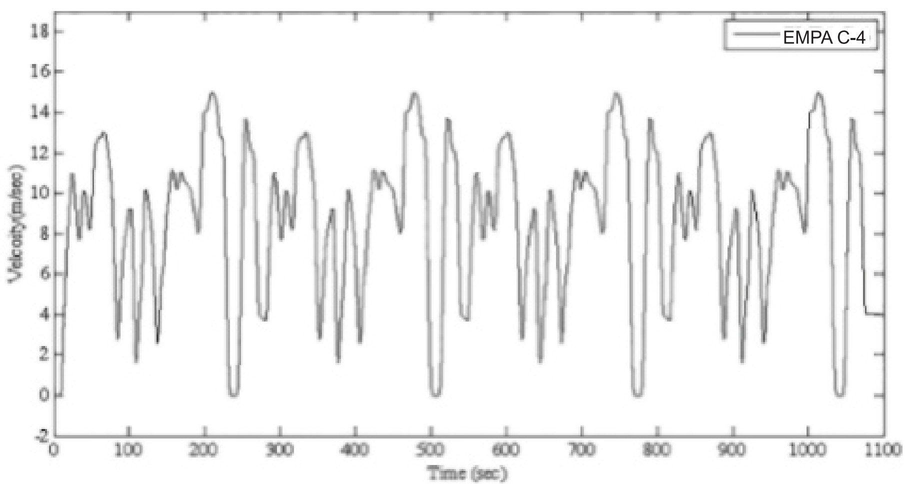


(d) High speed manoeuvring

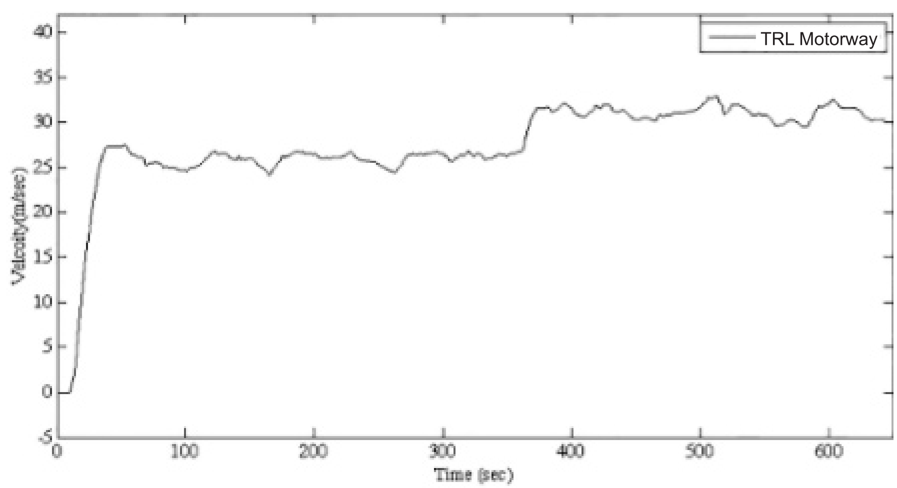
Figure 4: Time vs IVD Profile



(a) Urban driving pattern (EMPA –C1).



(b) Rural driving pattern (EMPA – C4)

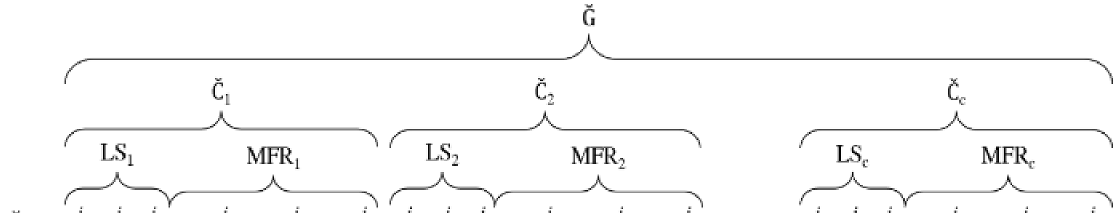


(c) **Highway driving pattern (TRL Motorway)**

Figure 5: Time vs velocity profile

Fuzzy Rule Base Formulation using GA (FRB-GA)

Inspired by the natural evolution, John Holland developed GA, which is a class of Evolutionary Algorithms (EA). The solution for the optimization problem is generated by natural processes such as selection, mutation, crossover and inheritance [4]. Linguistic Statement of a j^{th} rule (LS_j) holds the antecedents & consequent. Similarly Membership Function values of a j^{th} rule (MFR_j) holds the middle point and span values.



$$\breve{G}_1^c = A^j_{1,1}B^j_{1,1}C^j_{1,1}\,MFA^j_{1,1}MFB^j_{1,1}MFC^j_{1,1},\ A^j_{1,2}B^j_{1,2}C^j_{1,2}MFA^j_{1,2}MFB^j_{1,2}MFC^j_{1,2},\ldots\ldots A^j_{1,c}B^j_{1,c}C^j_{1,c}MFA^j_{1,c}MFB^j_{1,c}MFC^j_{1,c}$$

$$\breve{G}_2^c = A^j_{2,1}B^j_{2,1}C^j_{2,1}MFA^j_{2,1}MFB^j_{2,1}MFC^j_{2,1},\ A^j_{2,2}B^j_{2,2}C^j_{2,2}MFA^j_{2,2}MFB^j_{2,2}MFC^j_{2,2},\ldots\ldots A^j_{2,c}B^j_{2,c}C^j_{2,c}MFA^j_{2,c}MFB^j_{2,c}MFC^j_{2,c}$$

$$\vdots$$

$$\breve{G}_g^c = A^j_{g,1}B^j_{g,1}C^j_{g,1}MFA^j_{g,1}MFB^j_{g,1}MFC^j_{g,1},\ A^j_{g,2}B^j_{g,2}C^j_{g,2}MFA^j_{g,2}MFB^j_{g,2}MFC^j_{g,2},\ldots\ldots A^j_{g,c}B^j_{g,c}C^j_{g,c}MFA^j_{g,c}MFB^j_{g,c}MFC^j_{g,c}$$

Figure 6: Genome structure with '*c*' number of chromosome (Č) and '*g*' number of genomes (Ğ)

A Chromosome structure (Č) is formed by combining LS with MFR of the same rule. Each element in this structure Č (antecedents, consequent, middle point and span value) is a gene. Genome (Ğ) is formed by combining c number of chromosomes. In Figure 6, $j = 1, 2, \ldots, n$; $c = 5, 6, \ldots, 729$; $p = 1, 2, \ldots, 15$, where n is the number of linguistic variables used, r is the number of rules, p is the number of Genomes considered. Each genome is a possible solution in the search space. Since binary coding possess better searching ability, reduced coding/decoding complexity and easy implementation of genetic operations [10], each gene in the chromosome structure (Č) is represented in a 4 bit binary form (NVL – 0001, NL – 0010, NM – 0011, …, PL – 1000 and PVL – 1001). Binary coding of a k^{th} chromosome is presented as an example in Figure 7.

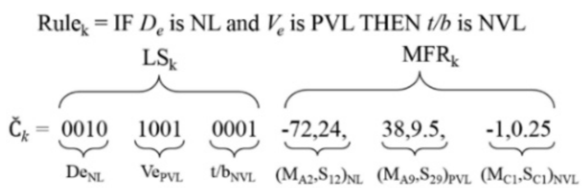


Figure 7: Binary coding of k^{th} chromosome

Algorithm

(i) Set the search space size

(ii) Set the number of iterations and genomes as 100 and 15 respectively

(iii) Set the upper and lower limits for c

(iv) Randomly select 15 genomes having c number of chromosomes as initial population

(v) Compute the cost function as per equation (8)

(vi) Select the best two of the genomes as "Parents" for the next generation

(vii) Perform Crossover and Mutation operations to obtain the offspring.

(viii) Check for maximum iteration, if reached, go to x else continue

(ix) Increment the iteration count and go to *v*

(x) Check for maximum value for *c,* if reached, Stop the iteration, else continue

(xi) Increment *c*, Reset Iteration count to 1 and go to *iv*

In order to improve the reliability of the obtained rule base, following factors are considered:

- Different values of Crossover (C_{pr}) and Mutation (M_{pr}) probability rates are considered (Table 2).
- To ensure higher selection probability for each rule, 15 genomes are considered concurrently.

Fuzzy Rule Base Formulation using PSO (FRB-PSO)

Bird flock movement inspired Eberhart and Kennedy thus resulting in the development of PSO algorithm which is kind of social optimization: meaning the position and velocity of a bird in the flock is influenced by the neighbours [2].

$$\acute{S}_1^p = \overbrace{\overbrace{\overbrace{A_{1,1}^j B_{1,1}^j C_{1,1}^j}^{LS_1}\ \overbrace{MFA_{1,1}^j MFB_{1,1}^j MFC_{1,1}^j}^{MFR_1}}^{\acute{P}_1}.\ \overbrace{\overbrace{A_{1,2}^j B_{1,2}^j C_{1,2}^j}^{LS_2}\overbrace{MFA_{1,2}^j MFB_{1,2}^j MFC_{1,2}^j}^{MFR_2}}^{\acute{P}_2}.\ldots\ldots\overbrace{\overbrace{A_{1,p}^j B_{1,p}^j C_{1,p}^j}^{LS_p}\overbrace{MFA_{1,p}^j MFB_{1,p}^j MFC_{1,p}^j}^{MFR_p}}^{\acute{P}_p}}^{\acute{S}}$$

$$\acute{S}_2^p = A_{2,1}^j B_{2,1}^j C_{2,1}^j MFA_{2,1}^j MFB_{2,1}^j MFC_{2,1}^j.\ A_{2,2}^j B_{2,2}^j C_{2,2}^j MFA_{2,2}^j MFB_{2,2}^j MFC_{2,2}^j.\ldots\ldots A_{2,p}^j B_{2,p}^j C_{2,p}^j MFA_{2,p}^j MFB_{2,p}^j MFC_{2,p}^j$$

$$\vdots$$

$$\acute{S}_s^p = A_{s,1}^j B_{s,1}^j C_{s,1}^j MFA_{s,1}^j MFB_{s,1}^j MFC_{s,1}^j.\ A_{s,2}^j B_{s,2}^j C_{s,2}^j MFA_{s,2}^j MFB_{s,2}^j MFC_{s,2}^j.\ldots\ldots A_{s,p}^j B_{s,p}^j C_{s,p}^j MFA_{s,p}^j MFB_{s,p}^j MFC_{s,p}^j$$

Figure 8: Swarm Structure with number of '*p*' Particles (Ṕ) and *'s'* number of Swarm (Ś)

Particles, swarm, velocity and position are the important terminologies used in PSO [12]. Particles (Ṕ) in PSO are framed by combining LS and MFR of the rules. *p* number of particles form a swarm (Ś) (Figure 8). Each swarm is a candidate solution in the search space and represents the rule base of the FLC. In Figure 8, $j = 1, 2, \ldots, n$; $p = 5, 6, \ldots, 729$; and $s = 1, 2, \ldots, 15$; where *n* is the number of linguistic variables used and *s* is the number of swarm. For every value of *p*, PSO undergoes 100 iterations. Each particle (ṕ) adjusts its velocity and position in light of its own experience (P_{best}) and the global experiences (G_{best}) according to equations (10) and (11) respectively.

$$V_{p,s}^{p'}(t+1) = V_{p,s}^{p'}(t) + \lambda_{1,p,s}(t+1)(P_{best}X_{p,s}^{p'}(t+1) - X_{p,s}^{p'}(t) + \lambda_{2,p,s}(t+1)(G_{best}X_{p,s}^{p'}(t+1) - X_{p,s}^{p'}(t) \quad (10)$$

The position update is obtained using

$$X_{p,s}^{p'}(t+1) = X_{p,s}^{p'}(t) + V_{p,s}^{p'}(t+1) \quad (11)$$

$\lambda_{1,p,s}$ and $\lambda_{2,p,s}$ are the factors that influence the velocity update of each particle and the convergence time to reach the best rule base. Different values of $\lambda_{1,p,s}$ and $\lambda_{2,p,s}$ are considered (Table 2) to obtain a better result

Algorithm

(i) Set the search space size

(ii) Set the number of iterations and swarm as 100 and 15 respectively

(iii) Set the upper and lower limits for *p*

(iv) Randomly select 15 swarm each having *p* number of particles

(v) Compute the cost function as per equation (8)

(vi) Update personal and global best position for every particles in the swarm

(vii) Update velocity and position

(viii) Check for maximum iteration, if reached, go to *x* else continue

(ix) Increment the iteration count and go to v

(x) Check for maximum value for *p*, if reached, Stop the iteration, else continue

(xi) Increment *p*, Reset Iteration count to 1 and go to iv

Fuzzy Rule Base Formulation using DE (FRB–DE)

Price and Storn developed Differential Evolution (DE). It is a simple, faster and population based stochastic function mimicking method. DE searches the optimal combination of the rules in the entire rules available as the search space. Solution for the optimization problem is obtained in DE through initialization, mutation, recombination and selection. If the selected population has satisfied the convergence criteria then the process stops else DE repeats from mutation [13]. Base vector (X) of DE is formed by selecting *n* rules from the search space. 15 such set of base vectors are selected. Donor vector (V) is formed from the base vector as per equation (12) in the mutation process. In recombination, the trial vector (T) is formed according to equation (13) which could be a combination of base and donor vector or any one depending on the Recombination factor R_{cr}.

$$V_{ri} = \begin{cases} X_{r(i)} + \sigma(X_{r(i+1)} - X_{r(i)}), & \text{if } i \le (n-2) \\ X_{r(i)} + \sigma(X_{r(1)} - X_{r(n)}), & \text{if } i = (n-1) \\ X_{r(n)} + \sigma(X_{r(2)} - X_{r(1)}), & \text{if } i \le (n) \end{cases} \tag{12}$$

$$T_{ri} = \begin{cases} V_{ri}, & \text{if } rand_{ij} \le R_{cr} \\ X_i, & else \end{cases} \tag{13}$$

where, Σ is the DE mutation factor. Each rule in the vector is identified by the suffix *i* and the set of vectors are identified by the suffix *r*. *n* represent the maximum number of rules considered. Value of *r* varies from 1 to 15 and *i* varies between 5 and 729. Different combination of Σ and R_{cr} as given in Table 2 were used for obtaining the best fuzzy rule. The fitness function for the trail $F(T_{ri})$ and the base vectors $F(X_{ri})$ are computed according to equation (8). The vector which has the least of the fitness function in every set is selected for the next iteration.

Algorithm

(i) Set the search space size

(ii) Set the number of iterations and set *r* as 100 and 15 respectively

(iii) Set the limits of *i* ($5 < i < 729$)

(iv) Randomly select *r* number of combinations of base vectors, each having *i* number of rules as initial population

(v) Compute trial vector according to equation (12) and (13)

(vi) Compute the cost function using base and trial vector as per equation (8)

(vii) Select the vectors having the least TCFI in all *r*

(viii) Check for maximum iteration, if reached, go to *x* else continue

(ix) Increment the iteration count and go to *v*

(x) Check for maximum value for *i,* if reached, Stop the iteration, else continue

(xi) Increment *i*, Reset Iteration count to 1 and go to *iv*

Table 2
Parameters of FRB – GA, FRB – PSO & FRB – D

	Combination		*C1*	*C2*	*C3*	*C4*	*C5*
GA	Genetic operators probability rates	C_{pr}	0.8	0.4	0.5	0.7	0.6
		M_{pr}	0.02	0.08	0.07	0.01	0.04
	No. of rules obtained		54	28	44	36	25
PSO	Influencing factors	$\lambda_{1,p,s}$	0.8	0.7	0.3	0.6	0.2
		$\lambda_{1,p,s}$	0.2	0.3	0.7	0.3	0.8
	No. of rules obtained		48	16	24	10	36
DE	DE Operators	Σ	0.6	0.9	0.8	0.7	0.4
		R_{cr}	0.3	0.3	0.4	0.6	0.6
	No. of rules obtained		15	30	28	39	16

Table 3
Performance comparison (Max PD & TCFI) of the proposed FRB – GA, FRB – PSO & FRB – DE

Controllers	*Parameters*	*Urban operating zone*			*Rural operating zone*		*Highway operating zone*	
		Driving Pattern	*Manoeuvring*		*Driving Pattern*	*Manoeuvring*	*Driving Pattern*	*Manoeuvring*
		EMPA. C-1	*Low Speed*	*Stop and Go*	*EMPA. C-4*	*Mid Speed*	*TRL. Motorway*	*High Speed*
Benalie (2009)	Max PD	8.562	0	0	111.963	0	7.491	11.537
	TCFI	109.338	22.154	95.801	15691.3	133.934	1973.09	1298.19
Tsai (2010)	Max PD	0	0	0	0	0	7.093	11.537
	TCFI	14.498	22.11	34.974	60.292	28.647	501.979	991.334
FRB - GA (C5)	Max PD	0.151	0.676	1.288	0.3	0.648	0.483	0.62
	TCFI	0.164	0.041	0.148	0.428	0.095	500.7	5858.7
FRB – PSO (C4)	Max PD	0.969	0.784	1.288	5.552	0.781	0.450	0.601
	TCFI	0.164	0.045	0.151	1.739	~~0.102~~0.095	0.078	0.017
FRB – DE (C1)	Max PD	0.147	0.609	0.591	0.349	0.661	0.541	0.674
	TCFI	0.107	0.036	0.114	0.429	0.088	2.513	~~0.125~~5858.763

Table 4
Performance comparison (Max PD & TCFI) of the proposed EFLC-FRVLC

Controller	*Parameters*	*urban operating zone*			*Rural operating zone*		*Highway operating zone*	
		Driving Pattern	*Manoeuvring*		*Driving Pattern*	*Manoeuvring*	*Driving Pattern*	*Manoeuvring*
		EMPA. C-1	*Low Speed*	*Stop and Go*	*EMPA. C-4*	*Mid Speed*	*TRL. Motorway*	*High Speed*
Proposed - EFLC-FRVLC	Max PD	0.147	0.621	0.490	0.349	0.583	0.461	0.601
	TCFI	0.107	0.036	0.114	0.43	0.088	0.08	0.017

Performance Comparison

In order to maintain symmetry of formulation among the three soft computing techniques, the rule base size is varied from 5 to 100 and for each rule size 100 iterations were performed. For the urban and rural operating zones, FRB-DE with 15 rules (C1 combination, $\Sigma = 0.6$, $R_{cr} = 0.3$) gave a minimum value of TCFI. For the highway operating zone, FRB-PSO with 10 rules (C4 combination, $\lambda_1 = 0.6$, $\lambda_1 = 0.3$) gave the least TCFI value.

In all the operating zones the performance of FRB-GA controller gave the poor TCFI value. Among the five combinations used for FRB-GA, C5 combination ($C_{pr} = 0.6$, $M_{pr} = 0.04$) gave the least of TCFI. Though in some scenarios, all the three proposed technique for rule base formulation gave same TCFI value, the technique with least number of rules among the qualified was selected. The performance of these controllers had shown a larger improvement in terms of TCFI and Max PD over the controllers proposed in [11] and [3]. Table 3 provides the performance comparison for various test scenarios used for this research work. On observing the test results (Table 3), FRB-PSO shown a better performance in highway and FRB-DE shown a better performance in rural and urban operation zones.

Ensemble Fuzzy Logic Controller

In order to have a better performance in all the operating zones, the EFLC - FRVLC is proposed whose architecture is shown in Figure 9. Two FLC controllers are connected in parallel. Both the controllers FLC1 and FLC2 receive the inputs (V_e and D_e) simultaneously through a mux and provide simultaneous control output. Only one control output will be selected and used for controlling the vehicle. When the host vehicle velocity is below a threshold value (10m/sec) FLC with FRB-DE rule base will be selected and if the host vehicle velocity is above the threshold value FLC with FRB-PSO rule base will be selected. The proposed controller is also tested for the scenarios mentioned section 4.1. Result shows (Table 4) that the proposed controller has an improved performance in all the test scenarios.

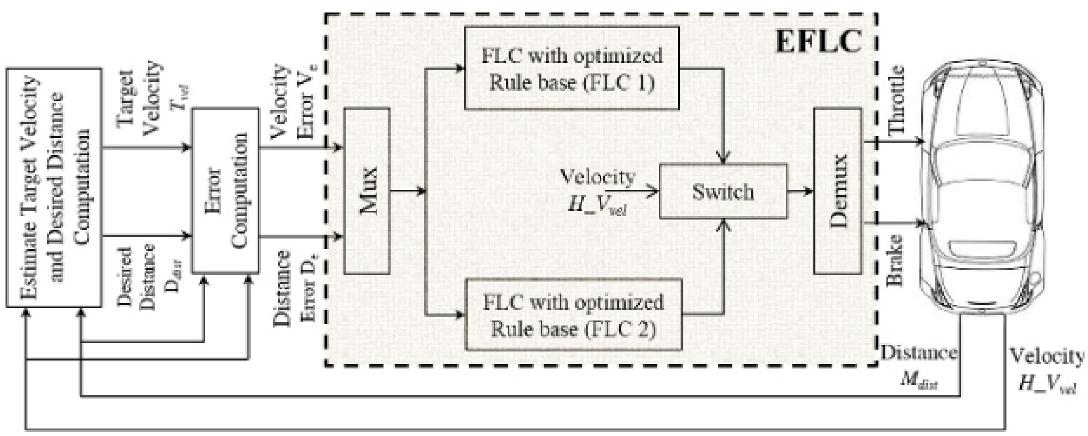


Figure 9: Proposed EFLC – FRVLC architecture

Model in Loop (MIL) Testing

MIL testing is performed with controller designed in MATLAB/Simulink and the virtual vehicle model available in CARMAKER. This Vehicle model produces a similar behavior of a real car (Seniz and Akca 2014) where it consists of road sensors, a provision to customize the following parameters which enables the testing more closer to the real world testing. Vehicle structure

- Road parameters like width, roughness, slope grade, humps etc.,
- Traffic conditions

- Road signs
- Generation of road pattern by generating .kml file from Google Earth/map.

A test run is created in CARMAKER along with the road and traffic scenarios. For testing BMW 5 model demo car is selected as the host vehicle.

Generating Test Track

Track of 7m wide with 0.5m marginal width is selected. Using Google Earth the .kml file of the road from Alandurai, Coimbatore India to Karunya University, Coimbatore India is generated. The length of the road is around 5.2 km. Figure 10 shows the bird eye view of the road used for testing.

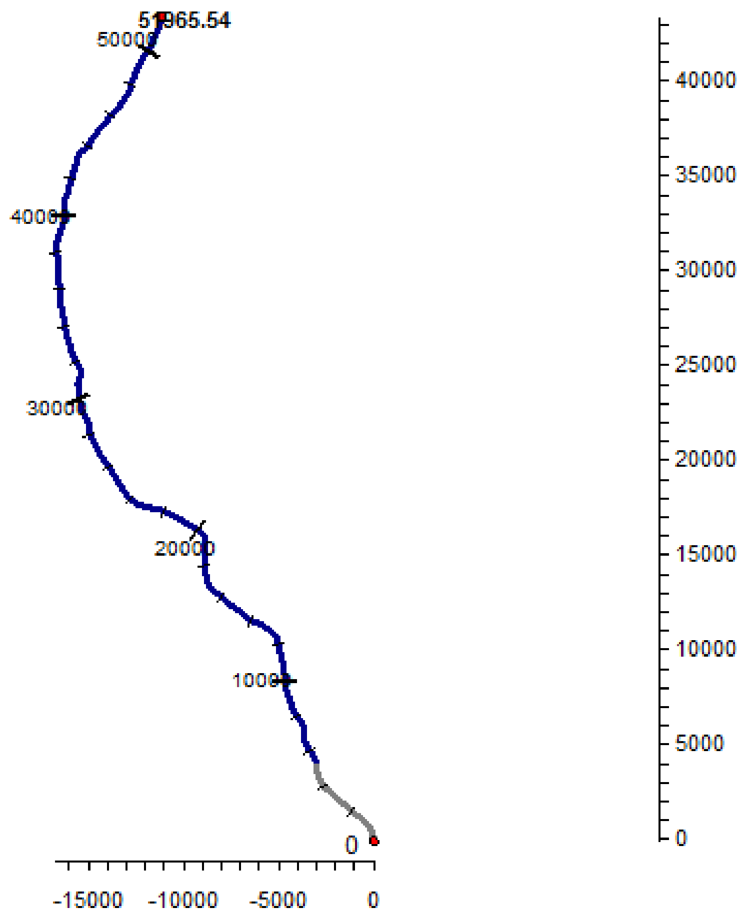


Figure 10: Bird's eye view of the test road

In addition, the actual road grading data of the test track is also added to make the track similar to that of the real road.

Generating Road Traffic

Two vehicles (VW Beetle 2012 and MB_Actros_1996 + Trailer) are used to create the traffic situation which makes the vehicle under test to undergo various maneuvers like Cruising (VCM), Stop & Go, Cut in and Cut out; Hard braking and Vehicle following (DCM). Initial displacement of these vehicles are 500 and 2350m from the origin.

Test Results

Time vs velocity curve obtained during MIL testing is shown in Figure 11. Conventional controllers proposed by Benalie et. al., (2009), Tsai et. al., (2010) along with the four proposed controllers (FRB-GA, FRB-PSO, FRB-DE and EFLC-FRVLC) are considered for testing.

Cruising: Host vehicle (BMW5 Series) is driven to reach the set cruising velocity of 22.22m/s (80km/hr). The host vehicle reaches the cruising velocity at 21st sec. FRB-GA and FRB-DE controller were not able to attain the set cruising velocity

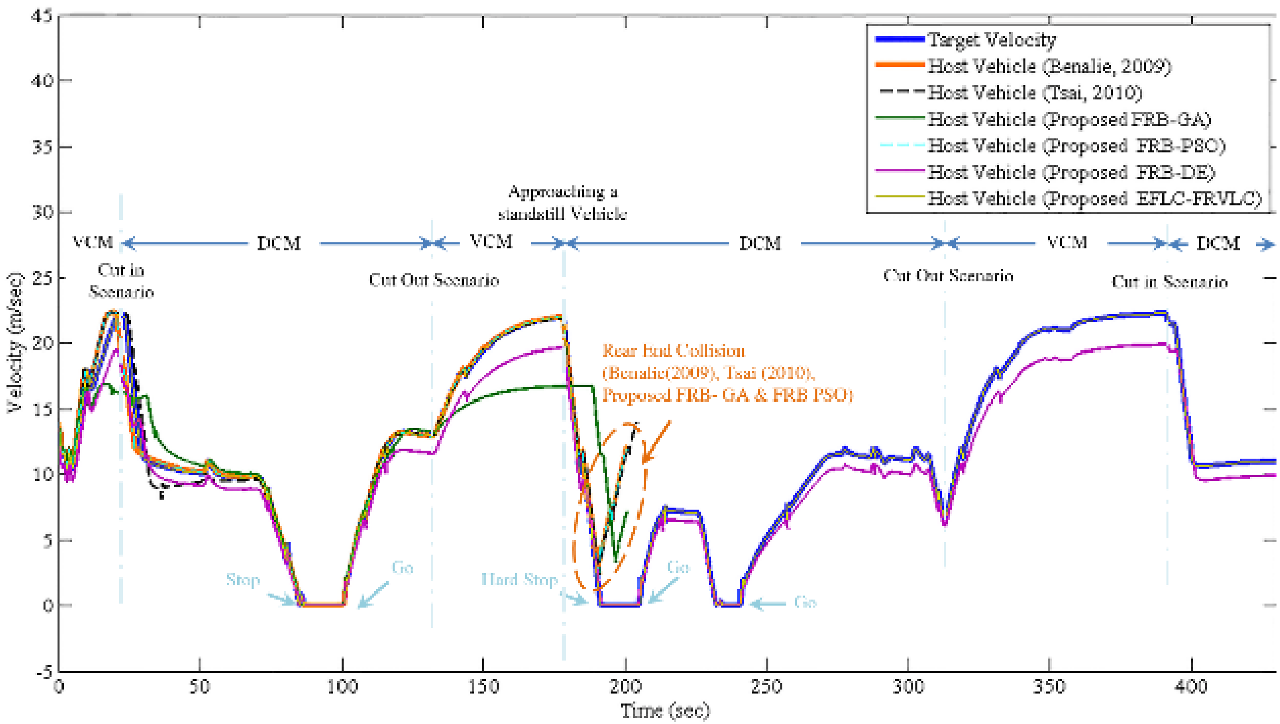


Figure 11: MIL testing response (Time vs velocity curve)

Cut in scenario and vehicle following: At 23rd sec a lead vehicle (VW Beetle 2012) is detected by the host vehicle and the controller switches from VCM to DCM. Host vehicle gradually adopts the changes made by the lead vehicle.

Stop and Go with braking: Braking scenario was tested by allowing the vehicle to decelerate at 0.61 m/s^2 from 9.7 m/s to 0 between 70th and 85th sec. All the controllers were able to stop the vehicle during this test scenario. After 100th sec the lead vehicle starts moving.

Cut out scenario and Cruising: Host vehicle follows the lead vehicle from 100th to 135th sec and there after lead vehicle changes its lane making the controller switch back to VCM. Host vehicle stats to cruise at 22.22 m/s till 178th sec. EFLC-FRVLC shows a closer tracking of target velocity than FRB- DE controller.

Hard stop: At 178th sec, the host vehicle detects a lead vehicle (MB_Actros_1996 + Trailer) in the same lane. As the lead vehicle velocity is zero, controller of the host vehicle needs to bring down the vehicle velocity to zero from 22.22m/sec. From Figure 11, it is observed that the deceleration was 1.69 m/s^2 between 178th and 191st sec. Controllers proposed by Benalie et. al., (2009), Tsai et. al., (2010), FLC with FRB-GA and FLC with FRB-PSO could not produce sufficient braking authority. Hence causing rear end collision.

Stop and Go scenario: From 225th sec to 230th sec the vehicle undergoes a braking with a deceleration of about 1.38 m/s^2. During this braking scenario, rear end collision was avoided by both EFLC-FRVLC and FRB-DE controllers. After 10 sec halt, the host vehicle starts moving and reaches a velocity of 11.1 m/s in 25 sec.

Cut out scenario and Cruising: Lead vehicle changes the lane at 310th sec after its velocity drops to 6.9 m/s. From 310th sec, host vehicle operates in VCM to drive at the cruising velocity. EFLC-FRVLC shows a closer tracking than FRB-DE controller

Cut in scenario and vehicle following: At 390th sec, VW Beetle 2012 moves into the lane of the host vehicle and slows down to run at 11.1 m/s. The host vehicle was able to adopt to this velocity.

Performance in terms of TCFI and Max PD is provided in Table 5. Proposed EFLC-FRVLC had shown 89.77 % improvement in terms of TCFI over FRB-DE and 38.1 % improvement in terms of Max PD over FRB-DE.

Table 5
Performance comparison for MIL testing

Controller	*Max PD*	*TCFI*
Benalie (2009)	Rear end collision Occurred. Test Aborted during run	
Tsai (2010)		
FRB – GA		
FRB – PSO		
FRB-DE	27.055	108.839
PROPOSED EFLC-FRVLC	16.745	11.129

3. CONCLUSION

GA, PSO and DE algorithms were used to formulate the rule base of the heuristic base fuzzy controller to achieve a balance between vehicle occupant safety and comfort. New performance metrics TCFI along with Max PD was proposed which incorporates safety and comfort in all operating zones like urban, rural and highway was achieved. Vehicle with FRB-DE shown a better performance in urban and rural operating zones. Vehicle with FRB-PSO had shown a better tracking at higher velocities but suffers from poor braking authority. Considering this EFLC-FRVLC was proposed. Rule base obtained using PSO and DE are loaded in the two parallel connected FLCs. The result shows that the proposed EFLC-FRVLC was able to perform well in all the above mentioned operating zones. MIL testing result shows that FRB-DE and EFLC-FRVLC have completed the total test run. Proposed EFLC-FRVLC had shown 38% and 87.8 % improvement in terms of Max PD and TCFI respectively over FRB-DE.

Acknowledgement

The authors would like to acknowledge IPG, Germany for providing the trial license for Carmaker and their technical support to carry out this research work.

REFERENCES

[1] B. Venhovens P. Naab K., “Stop and Go Cruise Control,” in FISITA World Automotive Congress, 2000, Vol. 1(2), pp. 1–8.

[2] Atoufi and H. Shah-Hosseini, “Advanced Knowledge Based Systems:; Model; Applications & Research,” in TMRF India, Vol. 1, Akerkar and Sajja, Eds. 2010, pp. 126–159.

[3] C. Tsai, S.M. Hsieh, and C.T. Chen, "Fuzzy longitudinal controller design and experimentation for adaptive cruise control and stop & Go," J. Intell. Robot. Syst. Theory Appl., Vol. 59, No. 2, pp. 167–189, 2010.

[4] Goldberg and J. Holland, "Genetic Algorithms and Machine Learning," Mach. Learn., Vol. 3, pp. 95–99, 1988.

[5] Seniz and S. Akca, "ETS Fuzzy Driver Model for Simultaneous Longitudinal and Lateral Vehicle Control," Int. J. Automot. Technol., Vol. 15, No. 5, pp. 781–794, 2014.

[6] J.E. Naranjo, C. González, R. García, and T. de Pedro, "Cooperative throttle and brake fuzzy control for ACC+Stop & Go maneuvers," IEEE Trans. Veh. Technol., Vol. 56, No. 4 I, pp. 1623–1630, 2007.

[7] J.J. Martinez and C. Canudas-de-Wit, "A Safe Longitudinal Control for Adaptive Cruise Control and Stop-and-Go Scenarios," IEEE Trans. Control Syst. Technol., Vol. 15, No. 2, pp. 246–258, 2007.

[8] J. Ploeg, E. Semsar-kazerooni, G. Lijster, N. Van De Wouw, and H. Nijmeijer, "Graceful Degradation of Cooperative Adaptive Cruise Control," IEEE Trans. Intell. Transp. Syst., Vol. 16, No. 1, pp. 488–497, 2015.

[9] J. Wang and R. Rajamani, "Should adaptive cruise-control systems be designed to maintain a constant time gap between vehicles?," IEEE Trans. Veh. Technol., Vol. 53, No. 5, pp. 1480–1490, 2004.

[10] J. Zhang and Y. Li, "Application of Genetic Algorithm in Optimization of Fuzzy Control Rules," Sixth Int. Conf. Intell. Syst. Des. Appl., Vol. 1, pp. 6–9, 2006.

[11] N. Benalie, W. Pananurak, S. Thanok, and M. Parnichkun, "Improvement of adaptive cruise control system based on speed characteristics and time headway," 2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS 2009, pp. 2403–2408, 2009.

[12] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," MHS'95. Proc. Sixth Int. Symp. Micro Mach. Hum. Sci., pp. 39–43, 1995.

[13] R. Storn and K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," J. Glob. Optim., Vol. 11, No. 4, pp. 341–359, 1997.

[14] S. Li, K. Li, R. Rajamani, and J. Wang, "Model predictive multi-objective vehicular adaptive cruise control," IEEE Trans. Control Syst. Technol., Vol. 19, No. 3, pp. 556–566, 2011.