Dynamic Simulation of The Car Platoon System

Farnoush Yousefi Kejani

***Abstract*— This paper illustrates a dynamic simulation of a car-following system, which contains three cars in one single line. This simulation is implemented by the multilayer perceptron (MLP) based on the fuzzy logic. This simulation is designed to prevent a car following collision system by adjusting the velocity and distance among the three cars. The input and the target value in the MLP are generated by 25 fuzzy rules in the fuzzy inference system (FIS). In order to represent a perfect generalization of the FIS system, the number of hidden nodes in the MLP networks should be set precisely. Setting the unappropriated number of hidden nodes in the MLP networks causes an overfitting of trained data, and creates noise in the regression model. The data in the MLP networks are trained by the Levenberg-Marquart backpropagation algorithm. If I reduce the velocity and distance among the cars, it causes an accident. So, the problem is tackled with changing the type of my membership function, the geometric type of membership function, and the defuzzification method. Also, the implantation of the car-following in the Simulink in the MATLAB software is not convenient for the user to put the initial values or the desired values in. In order to overcome this limitation, I design the car-platoon system in GUIDE in the MATLAB software as well. I then develop the simulation of the car-platoon, which contains four cars**

I. Introduction

[22] Nowadays, car-following models (CF) play a significant role in traffic engineering and safety search. [35] Car following behavior was implemented through several methods such as stimulus-response concept which was represented by Chandler et al. in 1958, and Gazis et al. in 1959, safety distance models, which was illustrated by Gipps in 1981, Krauss in 1997, and Treiber et al. in 2000, Psychophysical models which were represented by Leutzbach and Wiedemann in 1986, and optimal velocity models were illustrated by Bando et al. in 1995. In the (CF) model structure, the current vehicle has longitudinal space based on the lead vehicle driving in one single line. The CF modeling methods are divided into two parts: the analytical model and artificial intelligence (AI) models. [7] AI technology represented by negevitsky in 2011; many researchers have used this method to implement an automated CF modeling. In the process of developing the CF model, the California vehicle code is executed by pipe in [7] 1953 and Reuschel in 1950 with an analytical CF model. The fuzzy inference system firstly was represented by Chakroborty and Kikuchi in 1992; in order to fuzzify the driver behavior within the fuzzy sets. [21]Several papers have shown that the traffic and transportation problems have been solved by using the fuzzy sets in fuzzy logic such as Nakatsuyama et al. (1983), Sugeno and Murakami (1985), Sugeno and Nishida (1985), Yamazaki and Sugeno (1985), Sasaki and Akiyama (1986, 1988), Kagaya et al. (1992), Sugeno et al. (1989) Teodorovic and Kikuchi (1990), Chakroborthy (1990), Perincherry (1990), Perincherry and Kikuchi (1990), Chen et al. (1990), Teodorovi6 and Babic (1992) and Vukadinovic and Teodorovi6 (1992). Also, Khodayari et al. in (2012) represented a neural-network based on the CF model. The implementation of the car-following with neural-network based mainly has two advantages. [23] First, implementation the CF through the neural-networks has the ability to manage noisy data. This theory is represented by Adeli in 2001, Kalogirou in 2000, and Rafiq et al. in 2001. Second, the neural-networks doesn't base on the pervious knowledge or simplify assumptions.

Also, there are some challenges to simulate the car-following system which affects rules in fuzzy inferences system and weights in neural networks. Detecting the driver behavior is the most critical role in car-following design. [9] Driver facial features like the eye movement which determine the driver behavior such as fatigue, drunk, drowsy, and distract driving. In order to enhance the active automotive safety, a significant amount of studies have been done by ford, Nissan Institute, Columbia University, Vienna University of Technology, Nagoya University, Tsinghua University, Jilin University and the Chinese University of Hong Cong. For example, [7] in the real world, the actual driver could estimate the appropriate velocity and speed based on the lead vehicle in the short period in three categories. First, in slow-moving traffic, the current vehicle has a steady velocity because the speed of the lead vehicle will not change. Second, if the traffic becomes congested, the lead vehicle is going to be decreasing its speed, based on this fact that the current vehicle should be able to predict the appropriate its speed and decrease its acceleration. In the third categoric, when the traffic is finished the acceleration of the current vehicle should be changed based on the head distance and the driver characteristic. For example, an aggressive driver has a risky acceleration with the small head a way distance. On the other hand, the conservative follower has a safe acceleration, and a fearful driver decelerates their velocity. The acceleration of the follower vehicle should be defined with a variety of fuzzy sets. The variable is more realizable for driver with more fuzzy sets. In order to determine the right velocity and safe distance between the current vehicle and the lead vehicle fuzzy variables which are defined in fuzzy sets should be transformed into fuzzy membership function.

In order to overcome this critical issue, the authors in [9] represent a verity of methods to classify a driving behavior to achieve our desire and precision outputs. [9] The data which are collected from the sensors should be clustered through these algorithms neural networks (NN), Hidden Markov model(HMM), fuzzy control theory and Gaussian membership function (GMM). K-means algorithm and fuzzy control method are widely clustered the multi-dimensional unsupervised data into different groups. Also, Neural networks are utilized for driver classification. There are two distinct kinds of artificial neural networks to implement this classification. Back Propagation(BP), and Learning Vector Quantization (LVQ). To analyze these two methods, the BP has advantages to implement the driver classifications. The BP algorithm could train the feed forward neural network (FFNN) algorithms which are a potent system in order to design intelligent diagnostic system. Also, the cerebellar model articulation controller (CMAC) has a high convergence rate to cluster each driver's characteristics.

[8] Car-following and the lane-changing maneuver have been improved in order to prevent an accident. The acceleration of the following automated vehicle is adjusted by designing the throttle and brake controller. General Motors (GM) research group develop a car-following system in order to achieve its steady distance and velocity among the cars. Neural networks use brake/throttle fuzzy controller in a view to controlling the velocity in single-lane platoons. [8] In Spread Spectrum radar receives the speed and the distance between the host vehicle and the lead vehicle which are the input of the Cascaded Fuzzy Inference System (CFIS). The CFIS system is implemented in order to control the acceleration or declaration rate of the host vehicle. The implementation of the collision prevention system for car-following and the lane-changing maneuver includes four sections: four radar sensors, lane-changing control device, a CFIS based car-following control device, an emergency braking device and a vehicle inference. Radar sensors measure the distance between the host and the lead vehicle. Also, the CFIS car-following control device measures the acceleration or declaration of the host vehicle. The Lane-changing control device determines the appropriate time for changing the line based on the acceleration or declaration of the lead vehicle. Finally, vehicle inference uses some sensors like speedometer, brake, steering sensor in order to prevent an accident in car-following and lane-change modeling.

[16] To improve the safety of the Driver Agent, autonomous navigation should be implemented. Some results have been investigated to obtain the pose of the Driver Agent(AD) based on the prior maps. The initial pose of the DA should be measured through theses sensors: odometer, inertial measurement unit (IMU), inertial navigation system (INS), sonar, laser range finders, and cameras. Since these sensors couldn't estimate the correct pose of the DA, so it is not a suitable way to solve the Global Urban Localization (GUL) problem. To implement an appropriate map for outdoor mobile navigation, we could use the combination of the Fuzzy C-means (FCM) and Adaptive Network-Fuzzy Inference System (ANFIS). This combination could split the pictures which are available on the Internet (Google Earth) such as building, forest, fields, and roads; Next, the Monte Carlo Location (MCL) is used to find the pose of the automated vehicle. The (FCM) could cluster areas which are available in Google Earth; then the ANFIS system creates a boundary between different fuzzy group areas.

The rest of this paper is organized as follows. This article is divided into five sections. In the next section, the literature review is described. In the section III methodology is explained. Section IV describes my experimental results and analyze four cars in my simulation. section V gives concluding remarks. At the end I illustrate some Figures, which the details in my simulation.

II. LITERATURE REVIEW

[4] The largest simulation of the car-following(CF) in order to prevent an accident was made by General Motor (GM), the authors in this paper introduce two types of car-following models: one is General Motor (GM), and a proposed a fuzzy inference logic. Also, they want to compare the structure of these two models. The implementation of GM has some drawbacks which a proposed fuzzy inference can solve these limitations. The car-following behavior tries to control the distance between the current vehicle at a safe distance and keep it as a constant safe distance. To achieve the steady, safe distance, the current vehicle should accelerate or decelerate based on the speed of lead vehicle. Now, we want to consider the reaction of these two types of CF in some situation. First, we should explain our conditions, after that we could compare these two models. In order to control 'close-in' situations, we should set some rules in fuzzy inferences system. For example, if the distance between the current vehicle and the lead vehicle is large, then the current automated vehicle could accelerate, even though the lead vehicle is decelerating. On the other hand, when the current vehicle comes in congested traffic with small distance among the cars, the current vehicle should decelerate, even though the lead vehicle accelerates. The GM model couldn't express these properties, on the other hand, the proposed model could explain these features. Another situation which CF should control it is the 'Drift 'situation. In this case, the distance between the current vehicle and the lead vehicle couldn't remain in the stable distance. Because the current vehicle couldn't predict the speed of lead vehicle, also it couldn't stay in the steady speed. The GM model couldn't express these properties, on the other hand, the proposed model could explain these properties. There is two main major area to control the stability in car-following: one is the local stability and the asymptotic stability. Local stability controls the distance headway and the velocity among a pair of vehicles. On the other hand, asymptotic stability implies the stability in a platoon of cars. In the both cases of the stability in car-following both the GM model and the proposed model achieve the local and asymptotic stability.

[7] The Car-Following(CF) model simulates human driver characteristics in three sub classes: the fuzzy logic model, the cellular automata model, and the multi agent model. This part which describes the architecture of the multi-agent CF model and illustrates the model of its frame work. This model frame works include five layers structure: perception, anticipation, inference, strategy, which is used by the DA. First, The CF

driving receives all the necessary information, like the velocity and the distance between the current vehicle and the lead vehicle, from the outside of the world to adjust an optimal acceleration or deceleration of the vehicle. Fuzzy inferences system should be determined an optimal acceleration or declaration. So, we consider the corresponding action and the criteria points in each phase: In the perception stage, by gathering human driver's characteristics from the real world to simulate a driver agent (DA) is a major step to implement the CF modeling. The DA achieve the human driver's characteristic either accurate or fuzzy information through the observation in the real world. Fuzzy data estimates and measures by feeling parameters such as space, velocity, and acceleration. On the other hand, the accurate value which is crisp is measured by the instrument panel. Specific values are used to determine the velocity of vehicle, the number of lanes, the lane position, traffic signs, and signals. To simulate the DA in CF modeling the necessary information is measured either by fuzzy inference system or by the instrument panels.

[17] the acceleration or declaration of the following vehicle should be determined by the adaptive network fuzzy inference system (ANFIS) which is a multilayer feed-forward network. The initial input and output membership function of ANFIS is determined by a fuzzy inference system(FIS). A fuzzy controller is improved by using the ANFIS in order to achieve the optimal acceleration or declaration.

[7] In order to process the Car-Following(CF) model, which it simulates human driver characteristics, in the anticipation stage only the fuzzy information is used in order to estimate its velocity, and decision-making process. Herman and Rothery in 1965 implements multi-vehicle interaction. This part analyzes the work mechanism approach of the sub models. In the inference, stage simulates the DA like the human driver based on the information which achieves from the perception and anticipation phase. Also in this step generate suitable action for the strategy sub-model. Also, the strategy phase produces the optimal degree of safety in different criteria like safety, degree of comfort, and the degree of satisfaction which are generated for the instruction of inference sub-model. Also in this step, based on these criteria, the optimal acceleration, which is provided by defuzzied methods, should be determined. In anticipation phase, the DA should predict the short-time future driving like human driving. So, in this part, the driver agent predicts the acceleration and velocity in order to create fuzzy rules in fuzzy inferences system. Also, the characteristic of the driver should affect the fuzzy rules. As a result, a defuzzification of fuzzy rules generates accurate results. The two essential defuzzied functions are the Mamdani and the Sugeno defuzzification methods. The process of the anticipation step should be contained in three phases. First, generate fuzzy rules based on fuzzy sets. The second phase is reasoning based on the fuzzy rules. The main two algorithms which are used in antecedent step are the min and the prod methods. The third phase is aggregating the result of fuzzy rules. They could successfully simulate the DA with setting the 152 fuzzy rules.

[5] The car following simulation are developed in traffic simulation. In this situation, automated vehicles drive and change their line and steering by the AUTOPIA model. The essential factors to implement the automated vehicle include in three phases: Control System, Strategy, and sensors. In progress of this implementation, fuzzy logic embedded is used to automate their speed and steering. These vehicles are controlled by steering, throttle, and brake pedals. Also, infrared cameras in the night to improve night vision which is very important. To simulate an automated vehicle, use two fuzzy_logic_based controllers. The steering of the vehicle is controlled by DC servo motor. To automate the vehicle should be automated through these implementations. The brake pedal is automated by DC servo motor, and I/O card should transfer the correct gear into vehicle. Also, initial velocity of the vehicle and maintain in a safe velocity and over taking capabilities based on the distance between a current car and a lead car is another essential factor to simulate the human driving. The speed and steering of the automated car should be with two fuzzy-logic-based controllers: one is the steering (lateral) control and speed (longitudinal) control.

In order to improve traffic mobility, the Reinforcement Learning (RL) algorithm has been implemented as a precise structure for traffic signal control. [18] Thorpe and Anderson in 1996 firstly applied an RL algorithm to manage a separated signalized intersection, so he figures out that the RL-based signal is fixed time by decreasing the overall waiting time of vehicles. Furthermore, wiering in 2000 represented the implementation of reinforcement learning (RL) to manage traffic light agent in the traffic simulation. [19] Also, ramp metering is an efficient method to control motorway throughput. In order to decrease the travel time on the freeway, the authors represented some control systems such as capacity-density, ALINEA, and the model-based system. Among these methods, the model-based algorithm has been selected as a modern method, while it needs to calculate the high computational tasks. The Reinforcement Learning (RL) algorithm recently represented by Jacob and Abdulhai based on the Markov decision process (MDP); in order to overcome the model-based limitations. Also, coordinated ramp control, continuous state space, and indirect RL have been implemented by the RL algorithm.

[20] The adaptive cruise control(ACC) represented into the market first in Japan (1995), then in Europe (1998) and North America (2000) on 1995 in order to simulate the traffic situation. The ACC system is equipped with three components: radar, laser, vision, and their combinations. The ACC system combines with a (stop – Go) controller to control the speed of the automated vehicle by accelerating, decelerate and brake actions in a variety of situations. In order to develop the traffic simulation, the guidance system should be implemented by fuzzy logic and fuzzy variables. Since the implementation of the automated vehicle has a complicated mathematical formula and isn't linearized structure, the best way to implement this model like human driving is fuzzy logic. [5] Steering control is performed with two fuzzy variables: lateral-error and angular-error. The angular-error referrers the angle of the actual velocity and its desired velocity, and the lateral-error implies a distance between the actual vehicle and the desired vehicle. The turning angle of the automated vehicle is limited, in order to prevent an accident. The output of the singleton membership function determines this limitation which is similar to human driving and the correct route like human driving. The speed of the automated is controlled by setting fuzzy rules. These fuzzy rules have two inputs: the speed-error variable, which implies the difference

between the actual vehicle speed and our desired speed, and the acceleration of the automated vehicle. Also, these fuzzy rules have four outputs: throttle up, throttle down, break down, break up.

Also, the paper in [6] describes a microscopic traffic simulation is difficult because it has inherent complexity. The more recently ways to simulate traffic flow are multi-agent technology, a hierarchical modular modeling, and distributed simulation. To design this simulation, there are some traffic simulation tools like AIMSUN, VISSIM, and MITSIM. Nowadays, traffic flow simulates with AIMSUN online (real-time).

In this part, I would like to describe multilayer perceptron(MLP), which I use this type of neural networks in my simulation. [26] There are some supervised neural network training algorithms which these algorithms were utilized separately, but they are connected. These neural network algorithms such as perceptron, the LMS algorithm, three Madaline rules, and the backpropagation techniques. [26] In 1960 both algorithms like perceptron rules and the LMS algorithm were first introduced. Also in the same year, several neural networks were developed rapidly. [27] There are different kinds of feed-forward neural networks with hidden layers. These algorithms include Multilayer Perceptron (MLP), Radial Basis Function Neural Network (RBFNN) and General Regression Neural Network (GRNN). An MLP algorithm is usually trained by the backpropagation algorithm. The MLP algorithm is very robust regarding the regression for any mapping. The MLP algorithm could provide nonlinear map an N-dimensional input signal to M-dimensional output signal. This is the main advantages of the MLP algorithm in comparison with another type of neural networks. I implement my simulation through the fuzzy inferences system through the fuzzy rules and fuzzy sets. Each fuzzy set in every fuzzy rule describes the different condition of the cars. The output of this fuzzy inference system is the acceleration, which is defined based on the center of gravity. I extracted 1000 points within the FIS system, and I generalized my simulation through the MLP algorithm. In order to achieve the optimal generalization, I should set the number of hidden layers correctly.

III. METHODOLOGY

The primary goal of this project is implementing the car-following (CF) model for three cars with an FIS system and generalize with the MPL algorithm. I analyze the implementation of the CF in these two ways. First, implement the car-following system with fuzzy logic. Second, I generalized my simulation through the MLP networks.

Every input and fuzzy output variable should be defined in fuzzy membership function. The fuzzy system contains four parts: fuzzifier, inference engine, rules and defuzzification methods.

In the fuzzier part takes a crisp value from the universe of discourse of input and convert it into fuzzy sets. Also, inference engine which is based on rules. The rules are controlled by IF and THEN statement. As I consider in the literature review, characteristics of the object should affect in setting the rules. At the end of the defuzzied step should convert the fuzzy variable into the crisp value as an output. There are several defuzzifier functions: the center of gravity method which computes the weighted average of data points in fuzzy sets. Another defuzzifier function is the bisector method which can divide the area of under the shape into two equal parts. The mean of maxima is another type of defuzzifier method which calculates the mean of maxima in the fuzzy set. Fuzzy rules are created based on their input and output in ‘fuzzy logic' Toolbox in MATLAB. The FIS system has two inputs and one output.

[7] To analyze the working mechanism of the car-following system, we should set fuzzy variables like the velocity and acceleration of the current vehicle and the lead vehicle. Membership functions have three major components. First, every driver has the corresponding membership function based on the driver characteristics. Also, membership functions implement based on different criteria. For example, in the same range of acceleration, an aggressive driver would rather have higher accelerations, whereas a careful driver would rather have lower acceleration. Second, the main major of the fuzzy inferences system is to define the uncertain driver feeling, which should be used by intersection of two fuzzy sets. The intersection of two fuzzy sets which name is t-norm. Third, in order to define the steady velocity, the corresponding acceleration should be set to zero. For example, an aggressive driver prefers to have a small distance between the current vehicle and the lead vehicle, while the conservative drivers prefer to have long head away distance. In each step of fuzzy inference system have two inputs which are the velocity and distance and a result of acceleration. This actual result should compare to the desired result, and generate mean square errors in MLP algorithm. Iterate this step until this system achieves their desire output at a safe speed and velocity, in order to achieve a safe acceleration. The authors describe that they could successfully simulate the Driver Agent (DA) which is very similar to human driving behavior.

Since the CF modeling will be generalized by the MLP algorithm, the output of the membership functions should be defuzzied, in order to generate an acceleration as a crisp value. The ‘fuzzy logic toolbox' of MATLAB provides membership functions with a variety of geometric shapes, like triangular, trapezium, sigmoid, Gaussian, bell shaped, etc. Also, a good fit inputs and outputs should be found by the calibration of the membership functions. For example, one of the advantages of the sigmoid membership function is the shape of sigmoid function do not change too much during to find the suitable input and output. The range parameters in the fuzzy inferences system are defined to achieve the expected minimum and maximum. Also, a various type of sigmoid function is illustrated by adjusting the parameters values. After that, we should set the fuzzy rules to adjust the velocity and the distance among three cars. Every fuzzy rule adjusts with changing the values of the parameters in order to prevent an accident among three cars. I explain three different kinds of neural networks which are suitable to implement the car-following system. First, I explained the Reinforcement Learning(RL) algorithm, second the Adaptive Neuro-Fuzzy Inference System(ANFIS), and the last one is multilayer perceptron (MLP). I implement my simulation within the multi-layer perceptron (MLP), in which the input and the target should be defined in fuzzy inference system.

A. Reinforcement Learning(RL)

[14] Nowadays, the RL algorithms have been used in a variety of areas in modernized industrial world such as games, traffic light control, scheduling, portfolio optimization, inventory control, maintenance management, supply chain management, and supplier selection. Generally, the fuzzy RL algorithm is promoted harmony in paraments of fuzzy function by using the gradient descent. We can use a fuzzy function in value and policy function based on the architecture of RL algorithm. [14] For example, approximate reasoning-based intelligent control (ARIC), Generalized ARIC (GARIC), reinforcement neural network-based fuzzy logic, and reinforcement learning strategy based on fuzzy adaptive learning control.

[13] To implement the car-following system the reinforcement learning algorithm, uses fuzzy driving rules which are embedded in a neural network structure. The reinforcement learning algorithm is suitable to simulate the human driving and predict the behavior of the driver in an extended period. This proposed neural network has four layers. [12] The first layer indicates the input layer which represents continuous state variables. The second layer represents the fuzzy membership function. Also, the third layer represents fuzzy rules which are related to the second layer. The fourth layer indicates the action nodes. The most critical issue is the weights which are located between the fuzzy rules and the action layer. Reinforcement learning algorithm updates the weights between these two layers in order to decrease the errors. The feedback of the reinforcement learning algorithm is essential in order to choose the suitable action. For example, the reward feedback of this algorithm indicates the action of this neural network is similar to the driver behavior, while penalize action indicated the action of this neural network is far away from the driver behavior.

[11] I implement the car-following system with the Simulink in MATLAB. [15] Ljung in 2004 illustrates Identification Toolbox and Granier in 2006 represents continuous Identification Toolbox in Simulink. [15] Also, Crika & Fikar in 1998 constitute a variety of libraries for MATLAB and Simulink. The Simulink mostly use the Proposed Recursive Identification Algorithm Library (RIA). [15] Since the Recursive Identification Algorithm Library have a variety of recursive blocks, so there are some recursive algorithms which are implemented by these recursive blocks such as Least Square Method (RLS), Recursive Leaky Incremental Estimation (RILE), Damped Least Square(DLS), Adaptive Control with Selective Memory (ASCM), Variable method (RIV), Extended Least Square Method (RELS), Prediction Error Method (RPEM) and Extended Instrumental Variable method (ERIV). Also, some linear dynamic model like ARX, ARAMAX, OE are implemented by the recursive Identification Algorithm.

B. The Adaptive Neuro-Fuzzy Inference System

[36]The ANFIS models have been used in a wide variety of fields such as CO prediction, which is represented by Jain and Khare in 2010, SO2 prediction, which is illustrated by Yildirim and Bayramoglu in 2006, hydrodynamic modeling, which is represented by Razzak et al. in 2012, air quality modeling, which is represented by Rahman et al. in 2013, rapid five-day biochemical oxygen demand (BOD5) estimation, which is represented by Noori et al.in 2013, and traffic flow prediction ,which is described by Rahman in 2010. [10] when the fuzzy inference system (FIS) is generated defuzzied values, the output of this scheme should be used as an antecedent value in the artificial neural network. MATLAB Toolbox has the application which name is a neuro-fuzzy designer(ANFIS). [13] The adaptive neuro-fuzzy inference system (ANFIS) is another way to combine the fuzzy inferences system and artificial neural networks(NN) together. Typical artificial neural network model has lots of combination between layers and nodes in every layer. The main goal of the ANFIS model is to find the best combination of its input and output with the minimum error. The total number of layers is entirely depended on the characteristics of the system. The ANFIS Model structure is divided into five layers in the MATLAB software: input, “inputmf”, rules, “outputmf”, and output. In the input layer, the ANFIS structure just accepts the crisp value. The “inputmf” layer represents fuzzified values, and each node in “inputmf” is connected to every node in the input layer. Generally, the Gaussian membership function generates the inputs for ANFIS architecture. [13] Zahedi et al. in 2010 represent the Principal Component Analysis (PCA) which is one of the most efficient ways to prearrangements of input data for ANFIS architecture.

C. A multilayer perceptron (MLP)

[24] The perception neural networks were introduced by Rosenblatt in 1958 based on the work of [25] Hebb in 1949. The first perceptron neural network was represented by Widrow in 1959. Since a two-layer perceptron couldn't solve the logical formulae, so Minsky and Papert in 1969 have shown hidden layers in perceptron. In the prosses of the perceptron algorithm, feed-forward neural networks are a multilayer perceptron as is shown in Figure 6.

I implement the MLP neural network in the car-following system through the Neural Network Toolbox (nnstart) in MATLAB. In order to open Neural Network GUI, type the nnstart in the command windows in MATLAB. Select the getting started wizards, and choose the fitting tools in the nnstart wizard. The next wizard which is the fitting tools, we can load the input value and the target value. In the same wizard, next step we can choose the number of hidden neurons to train the dataset. After we trained the data set, the regression plot represents whether the network outputs are equal to the targets or not. If we want to have a more accurate result, we can retrain the network again. So, this change could change the initial weights and the bias of the network. At the end when I receive my trained data set, I can use the Simulink Diagram to illustrate my dynamic simulation.

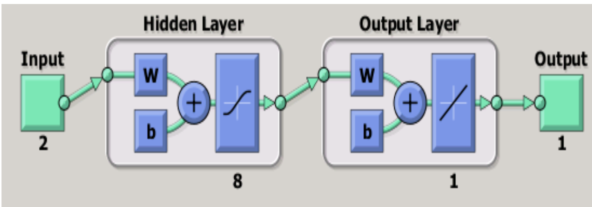

Fig. 6. *The structure of the MLP in my simulation of the car-platoon*

[31] My car-following system is implemented by the MLP. I choose this algorithm because it is fast and reliable network to solve the problems. Also, this algorithm is very robust to generalize my FIS system. The Simulink in MATLAB software is not a convenient system to change the initial and desired values because I should change my values in the FIS system. To solve this problem, [30] I implement the car-following system by using GUIDE in MATLAB. I use the GUI development environment (GUIDE) system to make my system more convenient; in order to create boxes to set the initial and desired values quickly by users. [30] I illustrate my Graphic User Inference(GUI) in the GUIDE layout editor. The GUI model typically is controlled by menus, toolbars, buttons, and sliders. I create my custom GUI; in order to show three cars at the same time. There are a variety of tools to modify my figure. Since I create new GUI, it generates MATLAB codes automatically. Every bottom in my own GUI has a call back function. I could transfer my FIS system into my GUI model through the call back functions.

IV. EXPERIMENTAL RESULTS AND ANALYSES

The primary goal of this paper is to simulate the car-platoons, which contains three cars, in a constant velocity and speed in order to prevent an accident. The desired and initial velocity and distance should be set by the user.

First, the FIS system is designed by fuzzy rules, defuzzification methods, membership function, the range of membership function, and the shape of the membership function. I analyze my simulation in a verity of a situation in which have the different velocity and distance. I set my rules, select centroid method as my defuzzification method, select Mamdani-type FIS with a trapezoidal geometric shape in the particular range. If I set the desired distance at 100 metrics, and the velocity at the 25 m/s, three cars come in the steady distance and velocity. However, decreasing the desired velocity and distance among the cars causes an accident. This problem is tackled with these changes. Since Simulink in MATLAB is not convenient to receive the expected values and initial values by users, so I implement the CF by GUIDE in MATLAB software to receive the input easily.

Second, I extracted 1000 points to generalize the FIS system through the MLP networks, and trained my data through the Levenberg-Marquart backpropagation algorithm, and show the results in the regression diagram [29] The first regression model, which was the least squares, was introduced by Legendre in 1805 and by Gauss in 1809. [27] There are several regression analyses such as linear regression, multiple linear regression, and nonlinear regression. Regression analysis indicates how the dependent variable reacts to a change in the independent variables. The trained data which are adjusted to the desired target versus the underfitting data which have lots of noise and errors in the regression diagram can be found in Figure 9 and Figure 10 in appendix. I explain more details in the rest of this section about my implantation

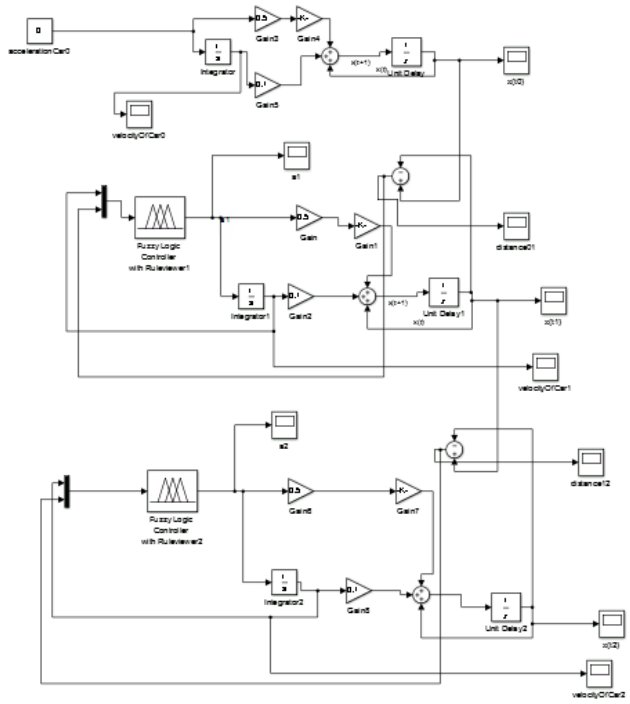

Fig. 1. The implementation of three cars in car platoon through the Fuzzy Inferences System.

Figure 1 shows the simulation of three cars which is implemented with fuzzy inferences system. In order to process this simulation, first of all, we should set the fuzzy inferences system based on their rules, and change the parameters in membership function editor wizard in MATLAB software to obtain the suitable trapezoidal shape. There are two main critical issues which affect the whole system of car-following. First set the correct fuzzy rules based on the logic of the system and choose the right parameters in membership function editor wizard. These fundamental elements could control the entire system in order to achieve these gaols. First, three cars should remain in steady velocity as the v_{desire}, and the distance between the three cars should be the same value of as the d_{desire}. If the fuzzy rules are not set correctly, the three cars couldn't remain in the steady speed and distance. The diagram, which shows the distance between the first and second car do not reach the desire output because the fuzzy rules are not set correctly, can be found in appendix in Figure 7.

For setting rules in fuzzy inferences system, the velocity, distance, and acceleration have these fuzzy variables. The fuzzy sets of the velocity for (v_n) cars should have these possible values: {Very slow, slow, just right, fast, very fast}. Also "Just Right" represents (v_n) is the same as (v_{desire}). The fuzzy sets of distance among the cars should have these possible values: {Very far, far, just right, close, very close}. Also, "just right" shows the distance between cars n and (n-1) as the desired distance (d_d).

The fuzzy sets of acceleration among the cars should have these Possible values: {Brake hard, brake, none, accelerate, accelerate hard}. Since the velocity has five fuzzy sets and the distance have five fuzzy sets, so we can set 25 rules in fuzzy inference system. The most critical points to implement this car-following system are how to define parameters values for triangular geometric shape, and set the fuzzy rules in fuzzy inference system in order to control three cars. These parts take plenty of time; because I should change these range in parameters to create the appropriate boundary and set a variety of rules in order to achieve my desired goal. The definition of variables which are used in my simulation can be found in table I.

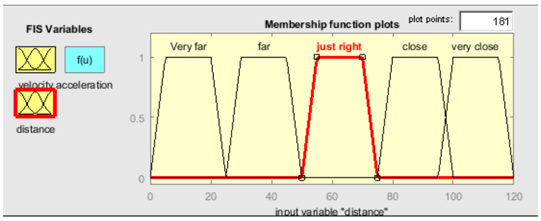

Fig. 3. This figure shows the Sugeno-type FIS with trapezoidal geometric shape

To simulate the car following model, we should calculate some mathematical formula. For example, integral of acceleration over time generates speed. The velocity of the second car is implemented by Integrator block, which shows in Figure 1. Also, integration of speed over time makes the position of the car. Figure 8 in the appendix shows that the integral of acceleration of the second and third car over time generates the velocity of the first car. Next, by simulating this formula, we can achieve the position of the cars in short period of times.

$$x(t+\Delta t) = x(t) + v(t) * \Delta t + 0.5\, a(\Delta t)(\Delta t)$$

I suppose that the first car has the constant velocity, and all the three cars have the same initial velocity which is 15 m/s.Also, the three cars have the zero acceleration at the first of movement. To simulate the second car the fuzzy inferences system (FIS) generates the a_1, and by doing the integral of acceleration of the second car, overtime produces the $velocityOfCar_1$.Also, Subtraction of $x(t_1)$ from the $x(t_0)$ produces the $distance_{01}$. The first Fuzzy inference system in Figure 1 has two input variables $distance_{01}$, and $velocityOfCar_1$, and one output which name is a_1. Also, the second FIS in Figure 1 has two input variables $distance_{12}$, and $velocityOfCar_2$, and one output which name is a_2.

In order to process this simulation, we should expand this scenario for the third car. To simulate the third car the fuzzy inferences system (FIS) generates the a_2, and the integral of the acceleration over time produce the $velocityOfCar_2$. Also, subtraction of $x(t_2)$ from the $x(t_1)$ produce the $distance_{01}$.

The major goal of this simulation is to control three cars in the steady velocity and steady distance among three cars to prevent an accident. I suppose that the velocity of the car_0 has the constant value which is 25m/s. Also, I set the d_{desire} and v_{desire} in the fuzzy set in fuzzy inference system. When I set d_{desire} in 100 metrics and v_{desire} in 25m/s in the FIS system, so I receive my goal successfully. The Figure 2 shows that these three cars remain in steady speed and constant velocity. The $distance_{01}$ and $distance_{12}$ have the steady distance (100 metrics), and $velocityOfCar_1$ and $velocityOfCar_2$ have the constant speed (25m/s). I supposed that the first car has the constant velocity which is 25 m/s. The initial velocity, the initial distance, the d_{desire}, the v_{desire} could be changed as a user input. In order to analyze this simulation, I discover that the MLP results are based on the different v_{desire} and d_{desire} which I set these values in fuzzy inference system. When I set my v_{desire} at 22.5 m/s and set the d_{desire} at 90 metrics, the three cars come in the constant speed and velocity. The MLP regression diagram shows that I achieve my goal successfully. On the other hand, if I change the v_{desire} at 10 m/s and the d_{desire} at 25 metrics, this simulation causes an accident among the cars, so the MLP neural networks show these errors. These noises seem to indicate that I didn't reach the target.

If I set the d_{desire} lower than 90 metrics and set the v_{desire} lower than 22.5 m/s, it causes an accident. So, this limitation is tackled by doing these changes. I convert Mamdani Fuzzy inference system into Sugeno-type FIS. The main difference between these two methods is how they generate crisp output from the fuzzy inputs. Sugeno Fuzzy inference system is produced the crisp output by weighted average. While Mamdani Fuzzy inference system is provided the crisp output by using the methods of defuzzification of a fuzzy production, the Sugeno-type FIS for a dynamic nonlinear system has better processing time; because it uses weighted average to generate crisp output. Also, I change the centroid as defuzzification method into the wtaver, and improve the triangular membership function into trapezoid geometric shapes, as is shown in Figure 3. I try to create correct boundary for the trapezoid geometric shapes, in order to define the range of each fuzzy sets through these boundaries. As a result, if I reduce my desire velocity and distance my FIS could control these three cars, in order to prevent an accident. I will try the FIS system in different desired, and initial values. If I set my d_{desire} in 50 metrics and set my v_{desire} in 20 m/s, $d_{initial1}$ in 80 m/s, $d_{initial2}$ in 70 metrics, the FIS system could control three cars on the same line. Also, If I set my d_{desire} in 25 metrics v_{desire} in 15 m/s with the pervious initial values, the three cars in the FIS system try to achieve their d_{desire}, and remain in their constant velocity. As a result, If I set d_{desire} in 10 metrics, and the v_{desire} in 10 m/s it causes an accident.

In order to change the initial and desired value, they should be changed in the FIS system. This issue which is the main limitation in this simulation, So I implement my simulation through the GUIDE (GUI development environment) model

TABLE I

THE DEFINITION OF VARIABLES WHICH ARE USED IN MY SIMULATION

Paraments	Description
$velocityOfCar_0$	the velocity of the first car
$velocityOfCar_1$	the velocity of the second car
$velocityOfCar_2$	the velocity of the third car
$distance_{01}$	the distance between the first and second car
$distance_{12}$	the distance between the second and third car
$d_{initial1}$	the initial distance between the first two cars
$d_{initial2}$	the initial distance between the second and third car
$v_{desired}$	the desired velocity of the cars
$v_{initial}$	the initial velocity of all cars
$x(t_0)$	the position of the first car
$x(t_1)$	the position of the second car
$x(t_2)$	the position of the third car
a_1	the acceleration of the second car
a_2	the acceleration of the third car

in MATLAB software which can be found in appendix in Figure 14. I create four boxes, d_{desire}, v_{desire}, $d_{initial1}$,$d_{initial2}$ for the user to put their desired value in this dynamic system. The input of the FIS system is within the input boundary (from 0 to 120). The FIS system has one or more inputs [27] membership function will be interpolated. The output value of the membership function is determined based on the wtaver of the output surface. For example, when the inputs are 24.4 and 97.4 as the velocity and distance, the output as acceleration is -0.0388 based on the fuzzy logic rules. The rule viewer picture can be found in the appendix in Figure 12.

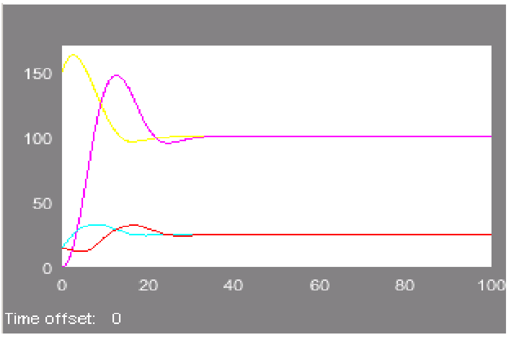

Fig. 2. This figure shows that three cars come in the constant velocity (25 m/s) and a constant distance (100 metrics).

In the end, we can extract 1000 points to build my data set and send these points into workspace in order to train the dataset in MLP neural networks. In order to prepare my dataset these variables such as a_1, a_2, $distance_{01}$, and $distance_{12}$ should be saved in workspace. I create this variable *'feature'=* *[$velocityOfCar_1$ (:,2), $distance_{01}$(:,2)] as* an input for the MLP neural network, and create the *label= [a_1 (:,2)]* as the target in MLP neural network; in order to receive a regression plot of the first and second car. Next, I create this variable *'feature_1'= [$velocityOfCar_2$ (:,2), $distance_{12}$(:,2)] as* an input for the MLP neural network, and create the *label_1= [a_2 (:,2)]* as the target in MLP neural network; in order to receive a regression plot of the second and the third car. Since the input data are fitted to the target, the regression diagram in the MLP neural networks show the trained data with the minimum errors. The input data and the output data in the MLP neural networks are extracted from the fuzzy inferences system, so the data depend On the rules in the fuzzy logic system. In order to generate correct data from the fuzzy system, I should set the rules correctly which it takes lots of time to fix it.

The main inputs of the MLP model are the velocity and distance. Also, the output is the acceleration of the following vehicle based on the lead vehicle in the short period. The MLP algorithm has trained the data through the backpropagation algorithm until the number of epochs reaches 41 or the mean square function achieves the minimum errors.

[27] The main critical point in the MLP is the number of hidden layers. The underfitting situation occurs if the number of hidden nodes is too few which causes training errors and high generalization errors. On the other hand, the overfitting situation happens the number of hidden nodes is too high, which causes low training error, but still have high generalization errors.

TABLE II

THE MLP TRAINING PARAMETERS FOR THREE CARS

Paraments	value	Description
Epochs	41	Maximum number of epochs to train
Time	23	Maximum time to train in seconds
min_grad	1*e-07*	Minimum performance gradient
mu	0.001	Initial learning rate (mu)
mu_max	1*e*+10	Maximum mu
Validation Checks	6	Measure the generalization of the network
Goal	0	Performance goal based on MSE

[27] Training parameters in the MLP algorithm are listed in Table II. When the number of epochs reaches 41, or the Mean Squared Error (MSE) reach zero, or the mu variable exceeds 1*e*+10, the network stop training. The MLP algorithm is trained by Levenberg-Marquart backpropagation. The network generalization is determined by the validation of the data points; in order to prevent overfitting. If the MLP model gives a good result on training, while it has a bad result in validation process, overfitting happens.

I tried to extend my simulation to implement four cars. The simulation of the four cars in the FIS system can be found in appendix in Figure 11. These four cars remain in steady velocity 25(m/s) and in 100 metrics the diagram which shows that these four cars drive in steady velocity and distance can be found in appendix in Figure 13. The four car-platoon is more sensible to change the desired velocity and distance in comparison to three car-platoons.

[27] Figure 4 shows the progress of the training data. While the number of epochs increases, the blue training curve indicates the training error based on the MSE is decreased. The green curve shows the validation error starts to decrease after epoch zero. The best performance achieves at epoch 28. Also, the red curve in Figure 4 indicates the testing of this performance. The red curve determines the error decrease after epoch 0, so stopping the training at epoch 28 is the best approach.

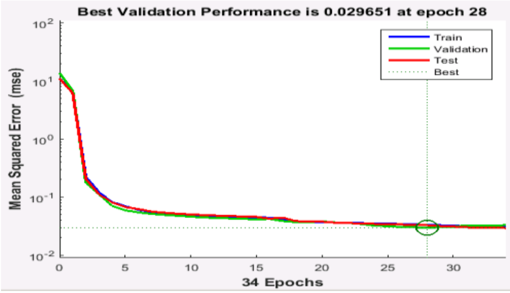

Fig. 4. The progress of the training data which is related to the four cars in car-platoon

Figure 5 shows the implementation of three cars through the MLP neural network. The input and the target value for the simulation of the MLP neural network are achieved through the defuzzied value in fuzzy inferences system. If these three cars, in the car-platoon system, drive in the steady distance and velocity as the desire, the MLP neural networks show the input data fit the target value. However, if three cars do not drive at a steady speed and velocity, these MLP neural networks show lots of errors and noise. The most critical issues in the simulation of the three car-platoon the MLP neural networks is the number of hidden layers. [28] Linoff, Blum, and Berry represented different rules to determine the suitable hidden layers. They explained that the number of nodes in the hidden layer shouldn't exceed from the double nodes in the input layer. In my simulation, the number of hidden nodes is between the 8 and 10 nodes. I illustrate the structure of my simulation in MLP model. This structure can be found in Figure 6.

In this section, I would like to compare the structure of my car-following system to the recent implementation. Since the structure of the car-following changes over the time, so the authors in the [34] implement the car-following system with an Evolving Local Linear Neuro-Fuzzy (ELLNF), which could adjust its structure over time. Evolving model is the best way to implement the car-following processing system. The car-following system has a lot of changes, so the evolving model has the ability to change its structure based on the variations. Also, the adaptive fuzzy systems and adaptive neural network systems are the other solutions to control time variant system. However, the adaptive model couldn't cover all of the operation condition system in many real-world industrial systems. The main advantages of the evolving system are to adopt new data. Also, this system could change its structure based on the new input data. For example, this system could add more rules in the FIS system or add neurons in the neural networks in order to adjust its structure based on the new input data. The evolving system has been developed into the evolving fuzzy system(EFS), and evolving fuzzy neural networks. In the process of this scheme, Local Linear Neuro-Fuzzy (LINF) has the ability to represent the best performance in a variety of identifications, prediction, and the control applications. The dynamic of the car-following has developed over the time, but the uncertain behavior of the driver still is the challenging issue in the CF model. The LINF model is implemented by a Takagi-Sugeno fuzzy system is powerful to control complex non-linearities model. Also, this system measures the distance between the cars through the Hierarchical Binary Tree Algorithm(HBT), which is an offline learning technique. The combustion process is implanted by the Box-Jenkins Time Series. The input of the Box-Jenkins Time Series is the methane air mixture with gas flow rate and the co_2 as the output of this system. This modernized system could generate more precise results in comparison to my simulation. Since the changing the CF system over the time, and the uncertain behavior of the driver are the most challenging issues in the CF system, the evolving system could cover most of this limitation. However, my car-following system doesn't have these facilities. Also, the distance among the tree cars is measured by the HBT algorithm, which generated more accurate results. The velocity, distance regression diagrams which are shown in this paper has the lower variations in comparison to my velocity and distance diagrams.

Also, the authors in [35] implemented the CF, in which it includes a variety of types of vehicles; not just car-car situations. For example, it includes car, two wheelers, bus, truck, light commercial vehicles (LCV), and auto-rickshaw. This paper represents how the distance could be measured precisely in the mixed traffic. In the follower-lead vehicles situations, the author's analysis the behavior of the following vehicle in different type of vehicles. For example, a heavy vehicle has the highest mean value of overlap, which is 26.6 percent. Most of the vehicles should follow a heavy vehicle as a follower vehicle because it has a lesser visibility on the sides. The two-wheeler vehicle has the high mean value of overlap, which is 25.6 percent because it has a small size among the vehicles. Also, the auto-rickshaw overtakes at the lower seed, so vehicles move away from the path of the auto-rickshaws. Vehicles don’t drive behind the auto-rickshaw. If we want to simulate the CF system which is similar to a real world, we have to analyze an actual situation in an intersection which has different types of vehicles. Every type of vehicles as a follower vehicle has a different estimation to predict the suitable distance among the vehicles. In my simulation, I implement my simulation just in car-car situation. This is a good suggestion to improve my tragedy in a mixed traffic which is more similar to real intersection.

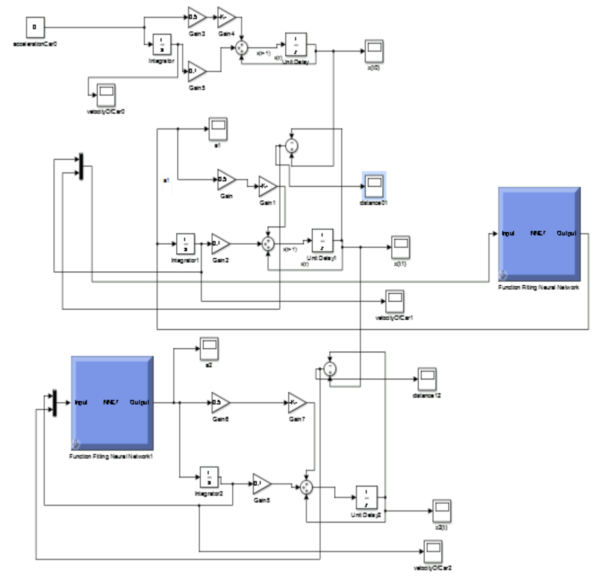

Fig. 5. The simulation of three cars through the MLP neural network.

V. Conclusion

The car-following system is implemented by the MLP neural networks based on the fuzzy inference system. The simulation of the CF through these methods is similar to the real world. The challenging issues in this simulation are the fuzzy rules and the number of hidden nodes in the MLP networks. If I don’t set the fuzzy rules correctly, the cars don't come with a steady velocity and distance. I create 25 fuzzy rules in the FIS system, and the crisp output is defined based on the centroid of the output surface. When I reduced my desired values, it causes an accident. To overcome this problem, I changed the FIS system elements such as the type of membership function, the type of defuzzied method, the geometric of the membership function, and the fuzzy rules. I extracted 1000 points from the FIS system, in order to create my dataset. Next, the MLP networks generalize the FIS system. When the fuzzy rules don't set correctly, or the hidden nodes don’t choose correctly, the MLP regression diagram shows the underfitted trained data. To generalize the FIS system, there are three main neural networks such as the ANFIS, the BP, the reinforcement, and the MLP algorithm. I chose an MLP algorithm which is trained by backpropagation algorithm. The MLP is very robust because of the regression for any mapping.

In the literature review, I discover some solutions to the challenge issues in the car following system. For example, the characteristics of the Driver Agent (DA) could affect the acceleration or declaration of the following vehicle. To overcome this problem the driver behaviors should be classified by one of these methods [9] such as neural networks (NN), Hidden Markov Model (HMM), fuzzy control theory, Gaussian membership function (GMM), Back Propagation (BP), or Learning Vector Quantization (LVQ). [17] Also, the acceleration or declaration of the following vehicle could be estimated more accurately by the adaptive network fuzzy inference system (ANFIS), which is a multilayer feed-forward network. [8] In order to improve the car-following in one single line into the lane-changing maneuver, the collision prevention system should be implemented with four radar sensors, lane-changing control devices, a Cascaded Fuzzy Inference System (CFIS), an emergency braking device and a vehicle inference. [16] The safety of the Driver Agent should be improved by implementing an autonomous outdoor mobile navigation through the combination of the Fuzzy C-means (FCM) and the (ANFIS) system. Also, the paper [37] illustrates the advanced traffic management is controlled with an adaptive neuro-fuzzy inference system(ANFIS), based the calibrations for PARAMICS model. The PARAMICS is a microscope road simulation software, in order to generate the model and behavior of each vehicle in a road. The authors represent the ANFIS modeling as a one of the most popular algorithms for the neural networks.

I implement the automated car-platoon which contain three cars. However, the recent implementations show that they are not considered just car-car situations. [35] Another type of vehicles such as two wheelers, bus, truck, light commercial vehicles (LCV), and auto-rickshaw, and its corresponding size should affect the other automated vehicles in the CF. This is an excellent suggestion to imagine the real situation in an intersection. To simulate the CF similar to real world, every feature of the driver behavior, and every condition of a road should be considered. I believe that the best implementation of the CF should cover all of these limitations.

References

[1] L. Chong, M. M. Abbas, A. M. Flintsch, and B. Higgs, “A rule-based neural network approach to model driver naturalistic behavior in traffic,” *Transportation Research Part C: Emerging Technologies*, vol. 32, pp. 207–223, 2013.

[2] Baciu, I., L. Ghiormez, and C. Vasar. "Study for the electric arc of alternative current at the single phase welding machine usin the Matlab/Simulink environment." *IOP Conference Series: Materials Science and Engineering* 163 (2017): 012027. Web.

[3] Zheng, Zuduo, and Majid Sarvi. "Modeling, calibrating, and validating car following and lane changing behavior." *Transportation Research Part C: Emerging Technologies* 71 (2016): 182-83. Web.

[4] Chakroborty, Partha, and Shinya Kikuchi. "Evaluation of the General Motors based car-following models and a proposed fuzzy inference model." *Transportation Research Part C: Emerging Technologies* 7.4 (1999): 209-35. Web.

[5] Naranjo, Jose, Miguel Sotelo, Carlos Gonzalez, Ricardo Garcia, and Teresa Pedro. "Using Fuzzy Logic in Automated Vehicle Control." *IEEE Intelligent Systems* 22.1 (2007): 36-45. Web.

[6] Chouder, Aissa, Santiago Silvestre, Nawel Sadaoui, and Lazhar Rahmani. "Modeling and simulation of a grid connected PV system based on the evaluation of main PV module parameters." *Simulation Modelling Practice and Theory* 20.1 (2012): 46-58. Web.

[7] Hao, Haiming, Wanjing Ma, and Hongfeng Xu. "A fuzzy logic-based multi-agent car-following model." *Transportation Research Part C: Emerging Technologies* 69 (2016): 477-96. Web.

[8] J. Mar and H.-T. Lin, “The car-following and lane-changing collision prevention system based on the cascaded fuzzy inference system.,” *IEEE Trans. Vehicular Technology*, vol. 54, no. 3, pp. 910–924, 2005 [Online]. Available: http://dblp.uni-trier.de/db/journals/tvt/tvt54.html#MarL05

[9] N. Lin, C. Zong, M. Tomizuka, P. Song, Z. Zhang, and G. Li, “An Overview on Study of Identification of Driver Behavior Characteristics for Automotive Control,” *Mathematical Problems in Engineering*, vol. 2014, pp. 1–15, 2014.

[10] W. Phootrakornchai and S. Jiriwibhakorn, “Online critical clearing time estimation using an adaptive neuro-fuzzy inference system (ANFIS),” *International Journal of Electrical Power & Energy Systems*, vol. 73, pp. 170–181, 2015.

[11] I. Baciu, L. Ghiormez, and C. Vasar, “Study for the electric arc of alternative current at the single phase welding machine using the Matlab/Simulink environment,” *IOP Conference Series: Materials Science and Engineering*, vol. 163, p. 012027, 2017.

[12] Chong, L., Abbas, M. M., Flintsch, A. M., & Higgs, B. (2013). A rule-based neural network approach to model driver naturalistic behavior in traffic. *Transportation Research Part C: Emerging Technologies, 32,* 207-223.

[13] Ghaedi, M., A.m. Ghaedi, F. Abdi, M. Roosta, A. Vafaei, and A. Asghari. "Principal component analysis- adaptive neuro-fuzzy inference system modeling and genetic algorithm optimization of adsorption of methylene blue by activated carbon derived from Pistacia khinjuk." *Ecotoxicology and Environmental Safety* 96 (2013): 110-17. Web.

[14] M. H. F. Zarandi, S. V. Moosavi, and M. Zarinbal, “A fuzzy reinforcement learning algorithm for inventory control in supply chains,” *The International Journal of Advanced Manufacturing Technology*, vol. 65, no. 1-4, pp. 557–569, Aug. 2012.

[15] Navratil, P., and J. Ivanka. "Recursive identification algorithms library in MATLAB & Simulink." In *Vibration Problems ICOVP 2011: the 10th International Conference on Vibration Problems*, p. 296. ICOVP 2011 Supplement, 2011.

[16] Dogruer, C. U., A. B. Koku, and M. Dolen. "Outdoor mapping and localization using satellite images." *Robotica* 28.07 (2010): 1001-012. Web.

[17] J. Mar and F.-J. Lin, “An ANFIS controller for the car-following collision prevention system,” *IEEE Transactions on Vehicular Technology*, vol. 50, no. 4, pp. 1106–1113, 2001.

[18] J. Jin and X. Ma, “Adaptive Group-based Signal Control by Reinforcement Learning,” *Transportation Research Procedia*, vol. 10, pp. 207–216, 2015.

[19] C. Lu, J. Huang, and J. Gong, “Reinforcement Learning for Ramp Control: An Analysis of Learning Parameters,” *PROMET - Traffic&Transportation*, vol. 28, no. 4, 2016.

[20] M. Pereira, M. Beggiato, and T. Petzoldt, “Use of adaptive cruise control functions on motorways and urban roads: Changes over time in an on-road study,” *Applied Ergonomics*, vol. 50, pp. 105–112, 2015.

[21] D. C. A. Teodorović, “Fuzzy sets theory applications in traffic and transportation,” *European Journal of Operational Research*, vol. 74, no. 3, pp. 379–390, 1994.

[22] M. Brackstone and M. Mcdonald, “Car-following: a historical review,” *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 2, no. 4, pp. 181–196, 1999.

[23] Zheng, Jian, Koji Suzuki, and Motohiro Fujita. "Car-following behavior with instantaneous driver–vehicle reaction delay: A neural-network-based methodology." *Transportation Research Part C: Emerging Technologies* 36 (2013): 339-51. Web.

[24] Behler, Jörg. "Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations." *Physical Chemistry Chemical Physics* 13.40 (2011): 17930. Web.

[25] Santín, Daniel, Francisco J. Delgado, and Aurelia Valiño. "The measurement of technical efficiency: a neural network approach." *Applied Economics* 36.6 (2004): 627-35. Web.

[26] Widrow, B., and M.a. Lehr. "30 years of adaptive neural networks: perceptron, Madaline, and backpropagation." *Proceedings of the IEEE* 78.9 (1990): 1415-442. Web.

[27] Nassif, Ali Bou, Danny Ho, and Luiz Fernando Capretz. "Towards an early software estimation using log-linear regression and a multilayer perceptron model." *Journal of Systems and Software* 86.1 (2013): 144-60. Web.

[28] A. Azar, "Fast neural network learning algorithms for medical applications", *Neural Computing and Applications*, vol. 23, no. 3-4, pp. 1019-1034, 2012.

[29] A. Azadeh, R. Arab, and S. Behfard, “An adaptive intelligent algorithm for forecasting long term gasoline demand estimation: The cases of USA, Canada, Japan, Kuwait and Iran,” *Expert Systems with Applications*, vol. 37, no. 12, pp. 7427–7437, 2010.

[30] J. Huertas, S. Sánchez-Solano, A. Barriga, and I. Baturone, “A hardware implementation of fuzzy controllers using analog-digital VLSI techniques,” *Computers & Electrical Engineering*, vol. 20, no. 5, pp. 409–419, 1994.

[31] "MATLAB GUI." MATLAB GUI - MATLAB & Simulink. Accessed August 03, 2017. https://www.mathworks.com/discovery/matlab-gui.html.

[32] "Documentation." Perceptron Neural Networks - MATLAB & Simulink. Accessed August 03, 2017. https://www.mathworks.com/help/nnet/ug/perceptron-neural-networks.html.

[33] "Documentation." Getting Started with Neural Network Toolbox. Accessed August 03, 2017. https://www.mathworks.com/help/nnet/getting-started-with-neural-network-toolbox.html.

[34] Kazemi, Reza, and Majid Abdollahzade. "Introducing an Evolving Local Neuro-Fuzzy Model – Application to modeling of car-following behavior." *ISA Transactions* 59 (2015): 375-84. doi:10.1016/j.isatra.2015.09.002.

[35] Asaithambi, Gowri, and Sehle Basheer. "Analysis and Modeling of Vehicle Following Behavior in Mixed Traffic Conditions." *Transportation Research Procedia* 25 (2017): 5094-103. doi:10.1016/j.trpro.2017.07.001.

[36] Reza, Imran, Nedal T. Ratrout, and Syed Masiur Rahman. "Calibration protocol for PARAMICS microscopic traffic simulation model: application of neuro-fuzzy approach." *Canadian Journal of Civil Engineering* 43, no. 4 (2016): 361-68. doi:10.1139/cjce-2015-0435.

Appendix

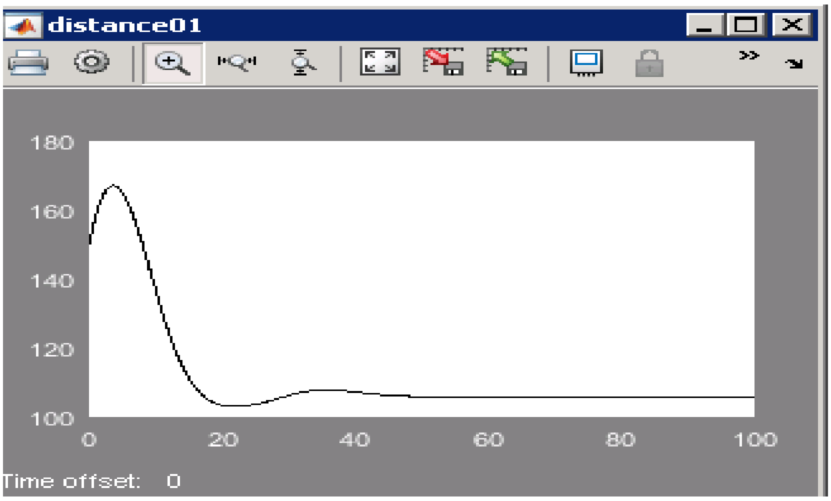

Fig. 7. the distance between the first and the second car do not reach the desired distance (100 m) because the fuzzy rules are not set correctly.

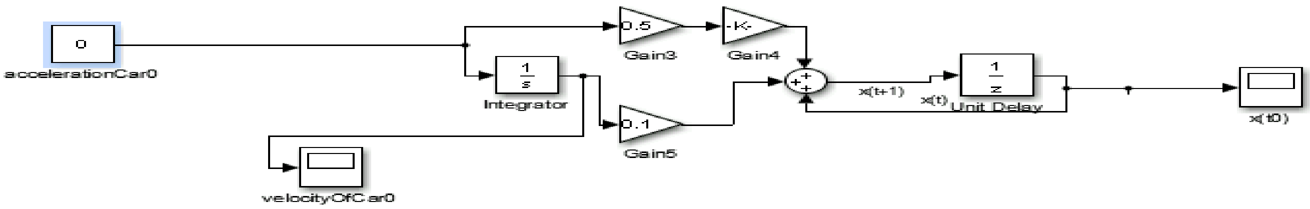

Fig. 8. This picture shows the simulation of this formulae $x(t + \Delta t) = x(t) + v(t) * \Delta t + 0.5\ a(\Delta t)(\Delta t)$

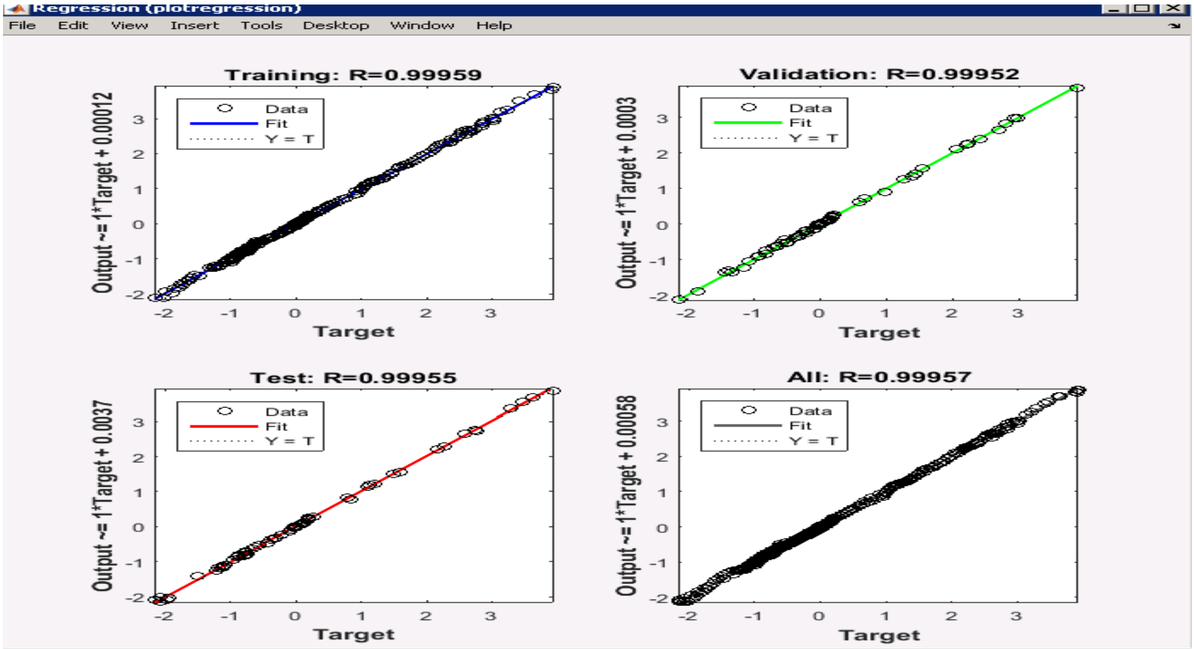

Fig. 9. The fitting trained data in the MLP neural networks, which the v_{desire} set at 22.5 m/s and the d_{desire} set at 90 metrics

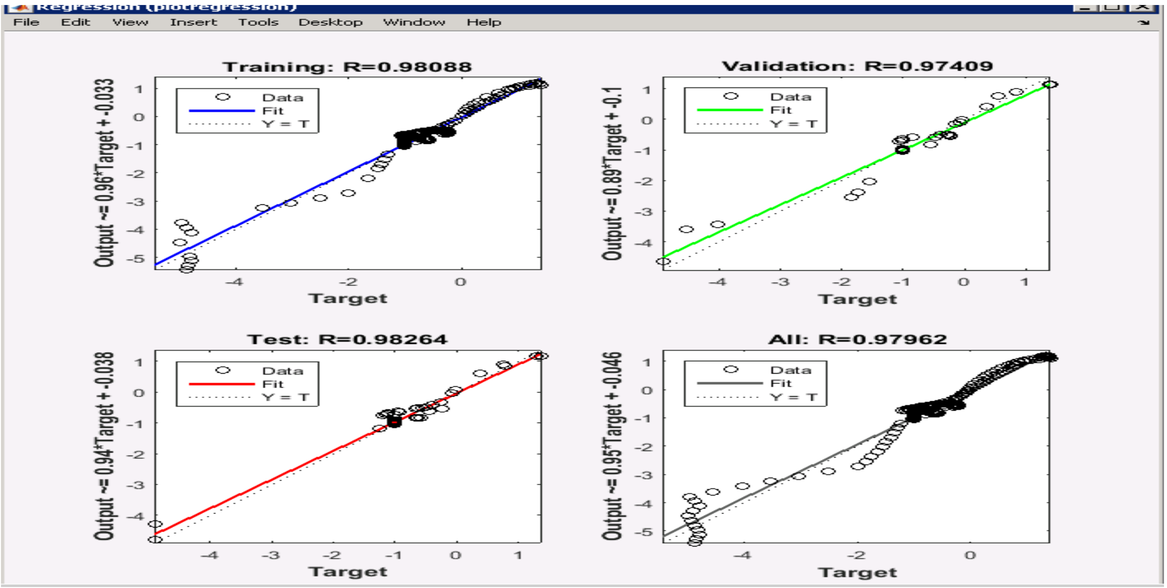

Fig. 10. The underfitting of the trained data in an MLP neural networks, which the v_{desire} set at 10 m/s and the d_{desire} set at 10 metrics in the FIS system.

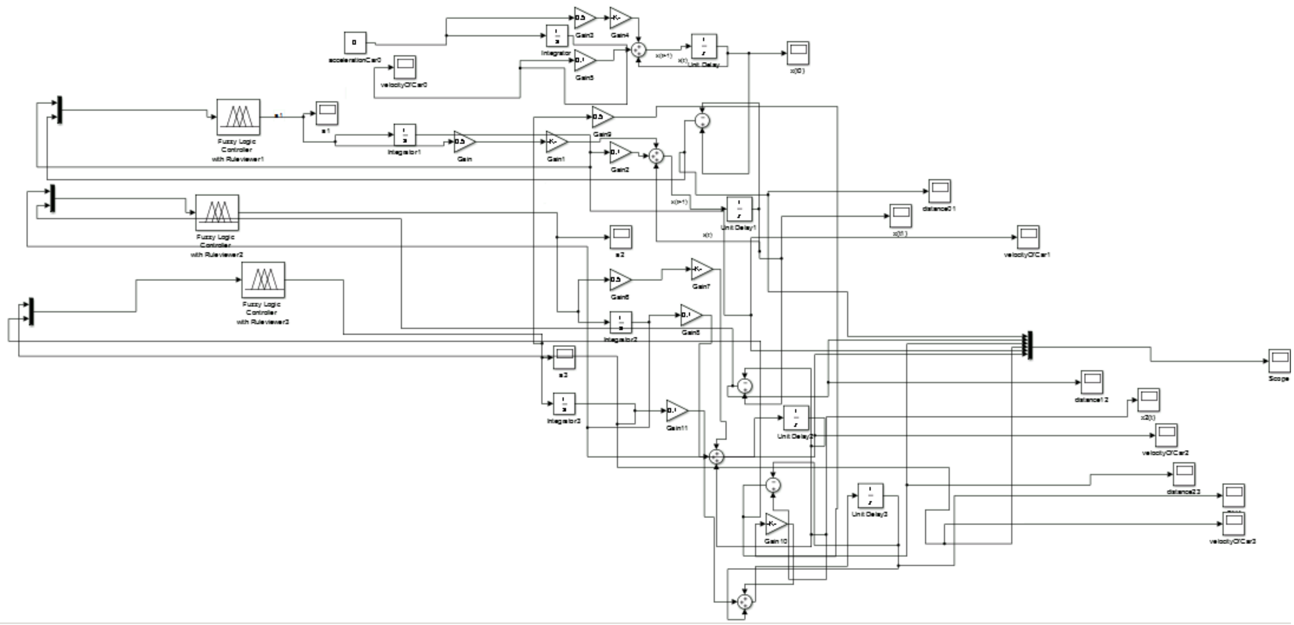

Fig. 11. The simulation of four cars in car-platoon in the FIS system

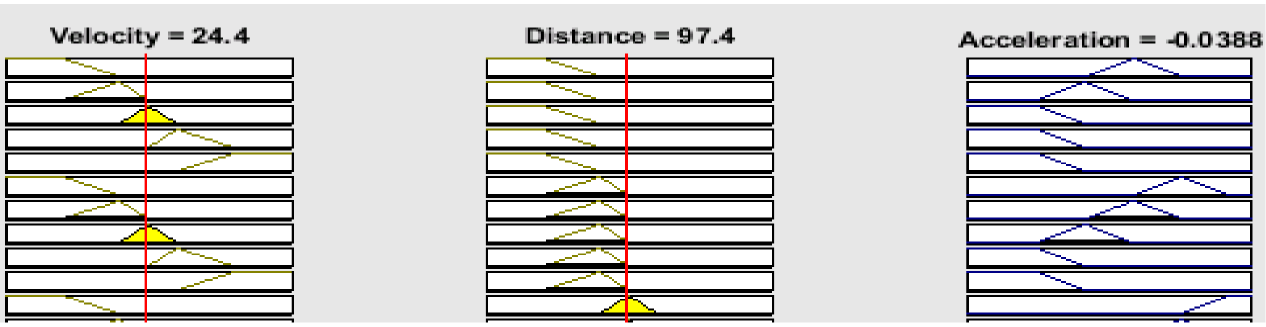

Fig. 12. Fuzzy rule viewer which shows that when the inputs are 24.4 and 97.4 as a velocity and distance the output that would be -0.03888 as an acceleration

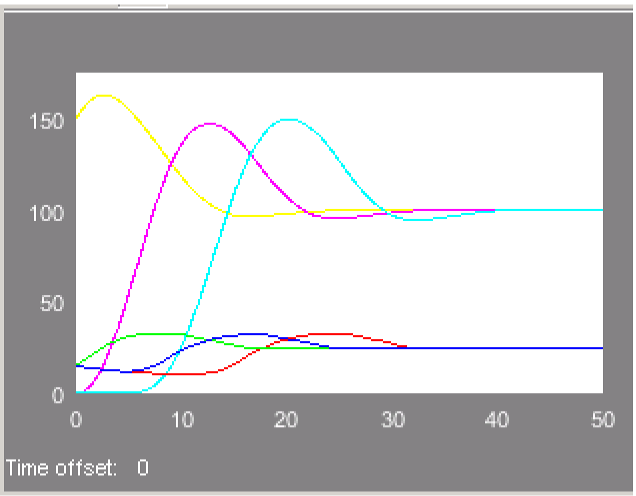

Fig. 13. Four cars come in a constant velocity (25 m/s), and distance (100 metrics)

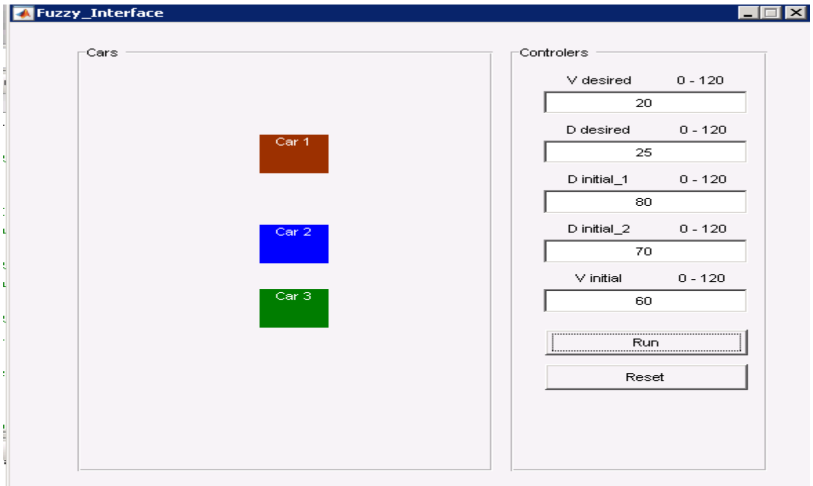

Fig. 14. Dynamic simulation of three cars in the GUIDE in MATLAB software