Term paper based on case study
And
Implementation of a fuzzy application

Fuzzy Traffic Control System

Submitted by:
Md Anwarul Azim (10036952) and Md. Nazmul Huda (10037802)

School of Computing
Queen's University

Content

Abstract

Fuzzy logic can be built on top of the experience of experts and blended with conventional control techniques while it is not the replacement of conventional control methods rather in many cases, fuzzy control system simplify implementation process. Increasing number of vehicles and shortages of roads lead to traffic congestion in many cities and this affects the efficiency, productivity and energy losses. One major factor of this congestion is the traffic signal controller operation methodology at the intersections of roads. In this paper, we proposed a fuzzy traffic control system, which is based on fuzzy logic theory. Our proposed system is able to handle the congestion in the intersection efficiently by controlling the time duration of the green phase interval. It can calculates the green light time duration by using the information about the number of waiting vehicles at red light phase and number of vehicle's frequency during the green phase.

The system will use Min-Max inference procedure and Centroid Defuzzification strategy to get the crisp value for the green light time duration as the output. At the beginning, this paper will provide a brief idea about the fuzzy traffic controller and then discusses four case studies related to the fuzzy logic based traffic control system. Each of these papers represents different idea to implement fuzzy traffic control system. Our proposed fuzzy system is simulated by using Matlab tools, which will presents the figures of all the fuzzy sets and membership functions used in our fuzzy traffic control system.

1. Introduction

Traffic problem is one of the major problems in many metropolitan cities around the world. This traffic problem can affect the economy, slow down the development, reduce the production, increase cost, and hamper people's daily life. There are several causes that can create traffic problem in a big city. Among them increasing number of vehicles, shortage of sufficient roads and highways, and traditional traffic light system. All of these factors can create traffic congestion in the intersection but among them traditional traffic light system is one of the major factors. Traffic signals are common features of urban areas throughout the world, controlling number of vehicles. Their main goals are improving the traffic safety at the intersection, maximizing the capacity at the intersection and minimizing the delays [4]. Fuzzy logic controller allows linguistic and inexact traffic data to be manipulated in controlling the signal timings. A fuzzy control system is a rule-based control system which is characterized by expressing control rules of an expert using a fuzzy theory and determining a control command by a fuzzy inference. A fuzzy logic controller describes a control protocol by means of if-then rules. A typical control system is shown in Figure 1.

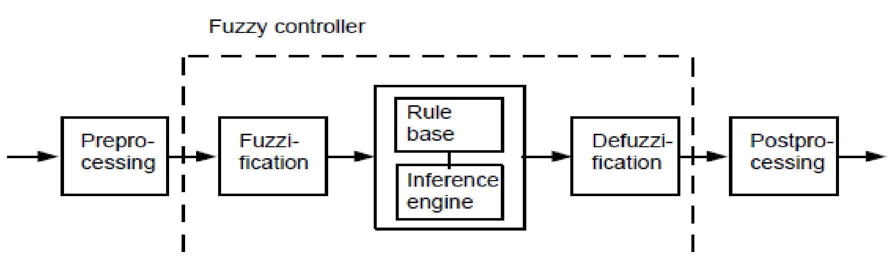


Figure 1: Fuzzy control system

The basic of fuzzy traffic signal controller is to model control strategy based on human expert knowledge [7, 8]. In a conventional traffic light controller, the traffic lights change at constant cycle time which is clearly not the optimal solution. It would be more feasible and sensible to pass more cars at the green interval if there are fewer cars waiting behind the red lights or vice versa [2, 3]. The fuzzy logic theory is introduced in the traffic controller to provide an intelligent green interval response based on dynamic traffic load inputs. A fuzzy logic control scheme is proposed to overcome the inefficiency of conventional traffic controllers that has a preset cycle time regardless of dynamic traffic load.

In this paper we proposed a new fuzzy traffic light control system that can effectively handle the bad traffic situation when there are congestion and long queue of vehicle waiting at red light. It can manage green phase lengths adaptively according to the traffic frequency and waiting queue. By controlling the green phase duration it can effectively reduce the vehicle waiting time at red light. To give a graphical view of the proposed system we developed a simulation by using MATLAB, which explains all the membership functions and fuzzy inference rules used for the system.

The paper is organized as follows: section 2 presents detail discussions of four case studies related to fuzzy logic based traffic control system. Section 3 is a detail discussion of our proposed fuzzy traffic control system. Section 4 deals with simulation, result, and discussion of proposed fuzzy system. Finally the paper's conclusion is presented in section 5. Future work of this fuzzy system is presented in section 6. Section 7 deals with references and the last part is our appendix.

2. Fuzzy Traffic Controller

A complete traffic signal cycle in the signalized intersection usually consists of the red, green and amber (or yellow) phases, which are turned on sequentially in a fixed time interval. Only the cars facing an activated green signal have the right of way to cross the intersection – moving straightforward, turning left, or turning right. The cars facing an activated red signal do not have the right of way and are forbidden to pass through the intersection. However, the cars are allowed to make right turns with the red signal as they do not interfere with those cars having the right of way to cross the intersection. The amber phase is activated in the transition from the green phase to the red phase. Typically, the amber phase is fixed in duration. The durations of the red and green phases are determined by the prevailing traffic conditions over a large coordinated area. Two case study of fuzzy traffic control system are given below:

2.1 Case Study 1

A Design Methodology for the Implementation of Fuzzy Logic Traffic Controller using Field Programmable Gate Arrays [2]:

Design and implementation of a traffic controller for a four-way intersection has been discussed in [2]. It is assumed that the forward going, right-turn, and left-turn are allowed in any approach (northbound, eastbound, southbound and westbound). To achieve a balanced traffic flow, they propose the time duration change in green phase interval. The cars in an approach with heavier traffic flow is given the right of way to cross the intersection for a time interval longer than an approach with lighter traffic flow. The objective is to make the number of cars waiting in queue should be minimized by dynamically increase or decrease green phase duration of the four way intersection.

A basic configuration of the fuzzy logic traffic controller [2] is shown in Figure 2, where the fuzzy traffic controller has four crisp input values (d_i , i = 1, 2, 3, 4) that represents the northbound, eastbound, southbound, and westbound traffic flow conditions respectively. The outputs of the maximum (max) operators are time multiplexed (MUX) before entering into the membership function circuit (MFC), which generates fuzzy sets required for the fuzzy reasoning operations in the fuzzy inference engine. The overall consequent fuzzy set resulted from fuzzy inference are then defuzzified to produce a crisp output.

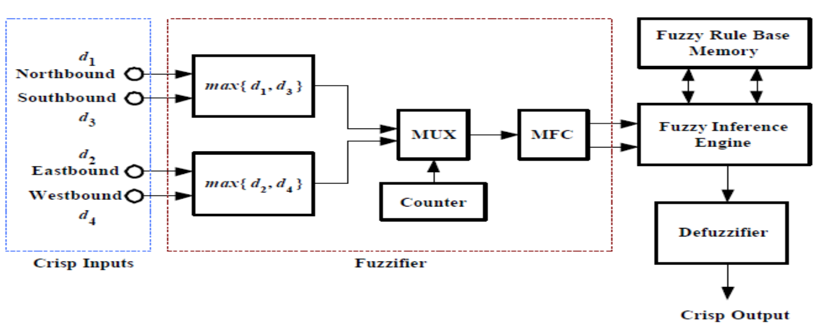


Figure 2: Basic configuration of the fuzzy logic traffic controller [2]

Membership Functions: The fuzzy sets of "traffic flow" are labeled as light, medium and heavy, while the fuzzy set "green phase duration", are short, medium, and long as shown in Figure 3. The two identical sets of membership functions are chosen for of simplicity and efficient implementation.

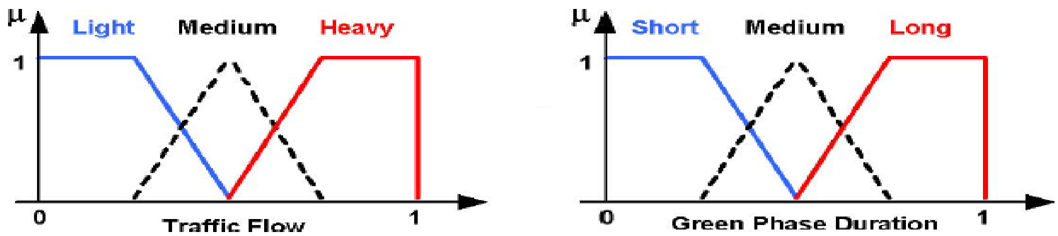


Figure 3: Traffic flow and green phase duration.

Fuzzy Rule Base: The entire IF-THEN rules are shown in Table 1. These rules are used for fuzzy reasoning by the fuzzy traffic controller, as for example, the fuzzy logic control strategy applied to the traffic flow conditions specified in the shaded column has the following form:

IF (NS-bound traffic is Medium) THEN (NS green phase duration is Medium)

IF (EW-bound traffic is Light) THEN (EW green phase duration is Short) and so on.

Table 1: Fuzzy IF-THEN rules for traffic control

	Traffic Flow Condition								
NS-bound traffic	Light	Light	Light	**Med**	Med	Med	Heavy	Heavy	Heavy
EW-bound traffic	Light	Med	Heavy	**Light**	Med	Heavy	Light	Med	Heavy
NS green phase	Short	Short	Short	**Med**	Med	Med	Long	Long	Long
EW green phase	Short	Med	Long	**Short**	Med	Long	Short	Med	Long
	Duration								

A fuzzy controller with large number of fuzzy rules increases the computation complexity. So it is a good idea to use minimal value of membership function. Before the rules can be evaluated the inputs must be fuzzified which determine a specific value from a lookup table indicating low, medium and high.

Fuzzy implication is implemented for each rule in the form of an AND operation that belongs the Mamdani approach, minimum minimum function that truncates the output set is used. Since decisions are based on testing all the rules defined in the fuzzy system, they must be combined to make a decision. On the other hand, aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set and apply here.

The input for the de-fuzzification process is a fuzzy set and the output is a single crisp number that can be accomplished by several methods while the center of area or centroid is a common methods. This method is simple in the case of symmetric fuzzy membership functions as shown above. However since the task requires integration which becomes complex for a series of disparate fuzzy rules. To overcome this, a moment method is proposed as another form of de-fuzzification discussed in [2]. The whole system is simulated using Matlab tools of Fuzzy Inference System (FIS) as shown in Figure 4 and 5.

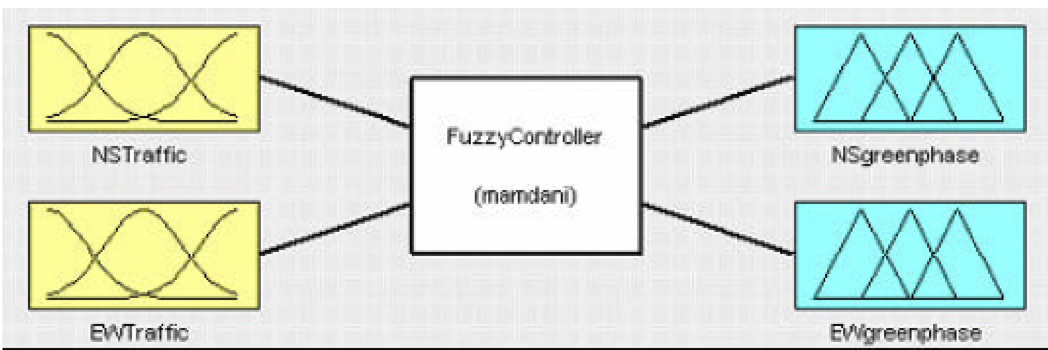


Figure 4: FIS

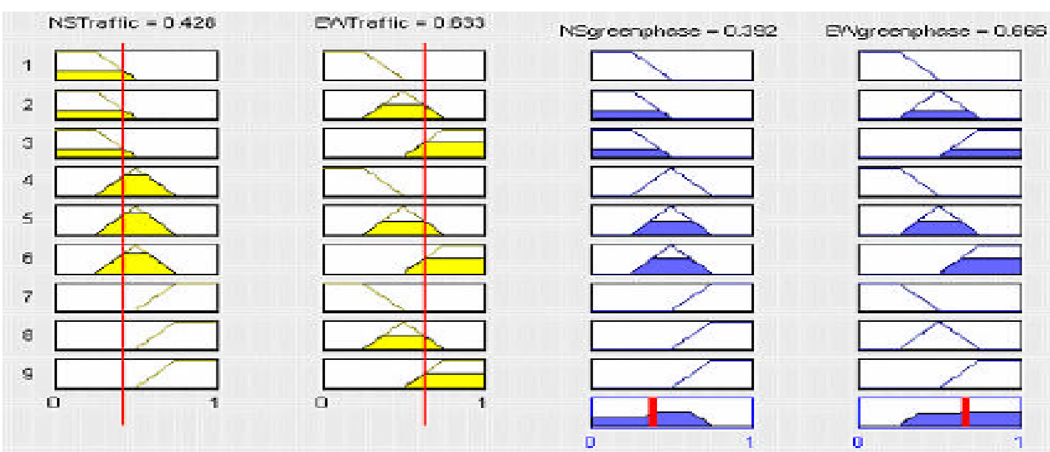


Figure 5: Rule viewer

Hardware design issues: Matlab program is used to implement the fuzzification, mamdani inference engine and defuzzification for two input fuzzy system as well as generating the contents of ROM. Mamdani inference engine makes use of the three fuzzy rules which are defined by Matlab. The output of the Matlab program is the green phase duration for traffic signals. These output values are then stored in a ROM in the form of Memory Initialization File (mif), which can be accessed, by Hardware Description Language (i.e. VHDL). Here ROM is nothing but a truth table for input to the VHDL program where based on these inputs one of the address values of the ROM is selected and the data stored at this address is given at the output port of VHDL code which is nothing but the green phase duration for traffic signals. A dedicated fuzzy chip is implemented to achieve highest execution speed. So once VHDL code is obtained then the next step is to create and design digital logic design for the system. For this purpose some state-of-the-art development tools and programmable logic devices are used. A detail hardware design process can be found in [2].

2.2 Case Study 2

Hardware Implementation of Traffic Controller using Fuzzy Expert System [5]:

This paper describes a functional Fuzzy Traffic Controller (FTC), which utilises fuzzy logic algorithm to achieve a flexible knowledge-based system in hardware design and achieve a better efficiency in the traffic control by minimizing traffic jam occurrences at interchange on roads. It overcome the weakness of conventional traffic controllers with the capability of providing varying green cycle interval based on dynamic traffic load changes at every lane in a 4-way junction control. Fuzzy concept is introduced by the membership function shown in Figure 6 and corresponding linear equations shown in Figure 7 as discussed in [5]. The system detected '–8' for the traffic volume differences between the traffic lanes considering waiting lane and passing lane as the controller input.

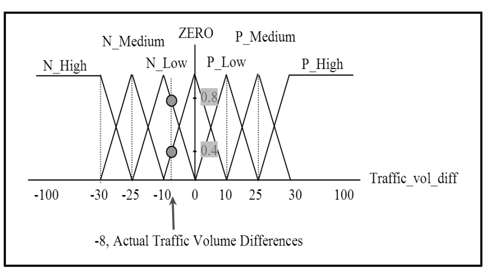


Figure 6: Membership Function of Traffic Volume Differences

Low_Traffic_Vol = min(1, -1/15*X + 25/15)
Medium_Traffic_Vol = min (1/15*X-10/15, -1/5*X+6)
High_Traffic_Vol = min(1, 1/5*X-5)
N_high_vol_diff = min(1, -1/5*X – 5)
N_medium_vol_diff = min(1/5*X + 6, -1/15*X – 2/3)
N_low_vol_diff = min (1/15*X + 5/3, -1/10*X)
Zero_vol_diff = min (1/10*X + 1, -1/10*X + 1)
P_low_vol_diff = min(1/10*X, -1/15*X + 5/3)
P_medium_vol_diff = min(-1/5*X + 6, 1/15*X – 2/3)
P_high_vol_diff = min(1, 1/5*X – 5)

Figure 7: Membership Function Definition

The inference engine determines the matching degree of the current fuzzy input (class) with respect to each rule. Fuzzy rule set for the FTC are shown in Table 2. The concept of Tsukamoto defuzzification model is used to develop the FTC algorithm.

Table 2: Fuzzy Rule

TRAFFIC VOLUME	TRAFFI_VOL _DIFF = ACTIVE LANE TRAFFIC – AVERAGE WAITING TRAFFIC						
	P_HIGH	**P_MEDIUM**	**P_LOW**	**ZERO**	**N_LOW**	**N_MEDIUM**	**N_HIGH**
HIGH	P_HIGH	P_MEDIUM	HIGH	HIGH	HIGH	N_MEDIUM	N_HIGH
MEDIUM	P_HIGH	MEDIUM	MEDIUM	MEDIUM	MEDIUM	MEDIUM	N_HIGH
LOW	P_HIGH	LOW	LOW	LOW	LOW	LOW	N_HIGH

Hardware design issue:

To develop the system, the behavior level of FTC algorithm has been developed using Very High Speed Integrated Circuit (VHSIC) and Hardware Description Language (VHDL) under MAX+PLUS II CAD environment is utilized. FPGA Express (Synthesis tool) also used to get a fully gate level synthesis architecture for the complete fuzzy based VLSI chip. There are several modes of operations in the system such as normal mode, emergency mode, pedestrian mode, alternate (counter clockwise) traffic flow direction, and yellow-light extension mode. Both of the clock wise and counter clockwise operation modes uses fuzzy controller to calculate the green cycle time during the traffic flow on a network. The main modules of the FTC system consists of Traffic Light Controller (TLC), pulse converter, emergency controller and fuzzy controller with operating clock driver using total of 42 inputs and 17 outputs that are shown in Figure 8.

The fuzzy controller is the central of intelligent, knowledge-based processor that read two inputs namely traffic volume and volume difference from traffic light controller. The traffic light controller will be used for calculating green cycle that feedback from fuzzy controller block to determine the traffic light interval.

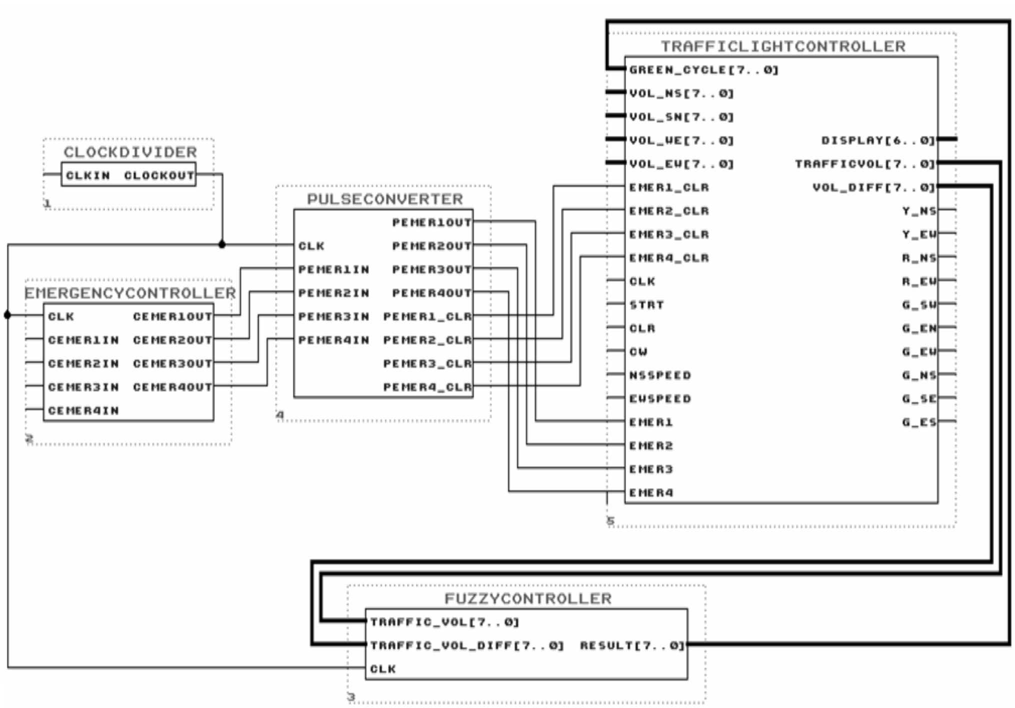


Figure 8: Fuzzy Traffic Controller -interconnection of the entire modules [5].

The emergency controller and pulse converter are activated hand-in-hand time when there is an emergency signal detected. The purpose of emergency controller is to ensure that the system responses to the correct or valid emergency input. The traffic light controller block as the heart of FTC modules receives emergency and clear input from pulse converter module, fuzzy result from fuzzy controller that responsible for the green phase calculation using fuzzy logic. Traffic volume and differences in traffic volume between the passing (active) lane verses waiting lanes is calculated in traffic light controller and fed into fuzzy logic controller sub-module.

2.3 Case Study 3

Control of complex Traffic Junction using Fuzzy Inference [3]:

Many traffic lights are located close to each other due to the closely located number of intersections. So in order to solve the traffic problems we must need to build the system considering the neighbor traffic controller and also the overall traffic situations. The fuzzy traffic light controller system, this paper talked about, can communicate with neighbor junction's traffic controller and manages phase sequences and phase lengths adaptively according to traffic density, waiting time of vehicles and congestion. The authors compared their fuzzy controller with other existing controller, called the present cycle time (PCT) and vehicle-actuated (VA) [6] controllers, which are available in many cities. Some terminologies used in [3] are shown in Figure 9:

- Link: It's a road connecting 2 intersections.
- Cycle: Cycle is a turn of traffic signals
- Phase: Phase is the traffic flow of the green lights.
- Capacity of the link: Capacity of the link is the maximum number of cars exists between the intersections.

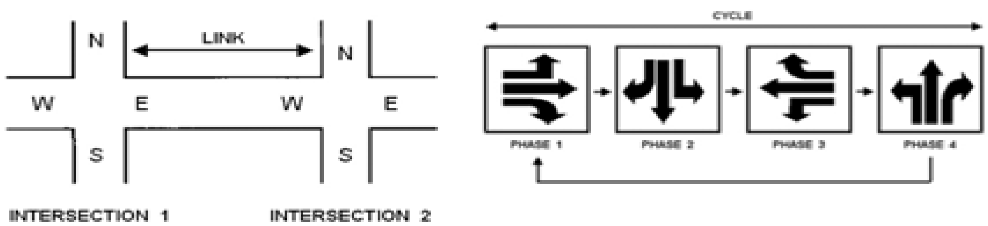


Figure 9: Graphical representations showing link, Cycle and Phase in a typical traffic lights control system [3].

Present Cycle Time (PCT) Controller: This is a simple regular traffic control system. It has a fixed time duration in one cycle for green, amber and red light. This preset time duration does not change or extend according to the conditions of the traffic flow and does not consider the vehicle's density.

Vehicle Actuated (VA) Controller: This type controller has vehicle detector at every road in the junction to detect the vehicles arrived. This method uses three parameters: *Initial Interval*, *Extension Unit* and *Extension Limit*. The time of initial interval begins when the green light phase starts and the green signal is extended by an Extension Unit's time after the initial interval elapses. During this extended period if any vehicle is detected then it extend one more this green signal Extension Unit. Once the Extension Limit is over it will not extend this green signal.

Fuzzy Inference based Controller: This paper considered two main features in the design of the fuzzy traffic lights controller. One is to reduce the heavy traffic congestion and reduce the total delay time of waiting vehicles and other is communicate with neighbor's traffic controller and synchronize the local traffic controller. For example: controlling the outgoing vehicles into neighboring traffic controllers. If intersection overloaded with vehicles, which creates congestion, that congestion would spread to its neighbors and all nearby intersection will be jammed. This proposed fuzzy control system is designed in such a way that, if large number of incoming vehicles creates jammed at the neighbor's intersections, then the number of incoming vehicles coming into the intersection will be reduced. There are three modules in the design of this fuzzy traffic controller is shown in Figure 10.

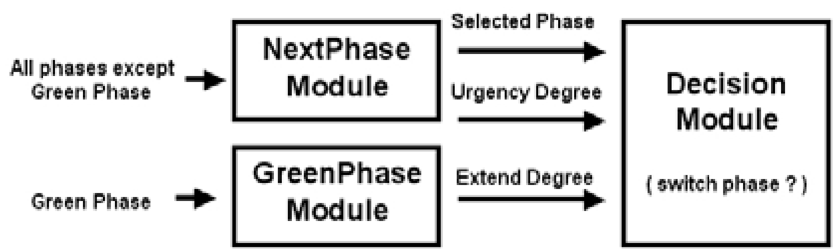


Figure 10: Three Modules of the proposed fuzzy controller [3].

Next Phase Module: This module selects the most urgent phase except the green phase. It has 3 inputs and 2 outputs. 3 inputs: (i) QueueNum,(ii) FrontNum, (ii) RedTime.

QueueNum: During red light phase, after the green light, number of vehicles remains in a line.

FrontNum: FrontNum measure the number of vehicles in the link between the affected intersection and the downstream intersections.

RedTime: It calculates the number of vehicles waiting at a red light just before the green light. This input helps to avoid long waiting time at the red light.

There are 2 outputs considered for the Next Phase Module: (i) Urgency and (ii) Phase.

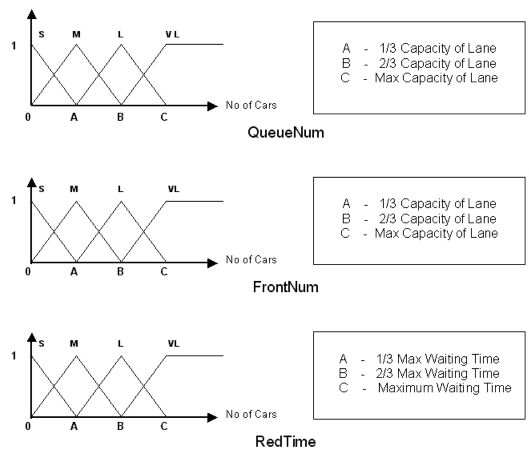
(i) *Urgency:* It is the worsening traffic condition of the selected phase. Its value increases if the traffic of the selected phase is bad.

(ii) *Phase:* This refers to the selected phase for the next phase after the green phase.

The Urgency values of all lanes are combined as the value of that phase. For the next phase, after the green phase, highest Urgency valued phase will be considered.

Fuzzyfication of the NextPhase:

Figure 11 shows the fuzzification component of the input variables of the NextPhase Module. In the 1st graph QueueNum, 4 Sets of QueueNum: S, M, L, VL. Let, consider Max Capacity, C = 12. If number of cars = 2, then we can see from the right side figure these membership of S, μ_s = 0.5. Also the membership of M, μ_M = 0.5



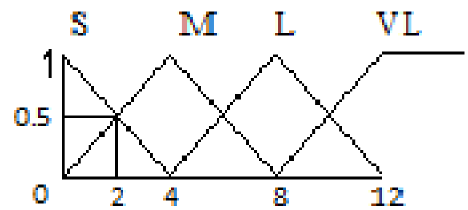


Figure 11: Fuzzification of QueueNum, FrontNum and RedTime in NextPhase Module [3].

The criteria to determine the output, *Urgency*:

1. If the number of vehicle reaches the maximum capacity in the link between the intersection and downstream intersections, no vehicles are allowed to go into that intersection. This prevents traffic congestion getting worse.
2. If *ReadTime* gets very long, the vehicles will be given priority to cross the intersection. This is to make sure that the average waiting time for every vehicle is minimized.

Table 3 shows 28 rules, which have been developed to relate the 3 inputs to the output, *Urgency*. Here we can see *Urgency* increases proportionally as *QueenNum* and *RedTime* increase. But *Urgency* decreases proportionally whenever *FrontNum* increases. Here in rule 22, if the number of cars in QueueNum is Large and number of cars in FrontNum is Small and number of cars waiting at RedTim is large then the Urgency is Very Large. This is because lots of cars are at QueueNum waiting at Red Light and there are few cars in FrontNum, the lane in front of red light, so its need to allow the cars to cross the intersection and let them move forward.

Table 3: Example of Fuzzy Rules of the NextPhase Module for output, *Urgency* [3]

RULE	INPUTS			OUTPUTS
	QueueNum	FrontNum	RedTime	Urgency
1.	Z			Z
2.	S	S	S	S
3.	S	S	M	S
4.	S	S	L	M
5.	S	M	S	S
6.	S	M	M	S
7.	S	M	L	M
8.	S	L	S	S
9.	S	L	M	S
10.	S	L	L	S
11.	M	S	S	S
12.	M	S	M	M
13.	M	S	L	L
14.	M	M	S	S
15.	M	M	M	M
16.	M	M	L	L
17.	M	L	S	S
18.	M	L	M	M
19.	M	L	L	M
20.	L	S	S	M
21.	L	S	M	L
22.	L	S	L	VL
23.	L	M	S	M
24.	L	M	M	L
25.	L	M	L	L
26.	L	L	S	S
27.	L	L	M	M
28.	L	L	L	M

In order to get the two outputs of *NextPhase* Module (*Phase* and *Urgency*), need to compare all the *Urgency* value of every phase except the green phase.

The GreenPhase Module:

This module observes the traffic conditions of the green phase. It has two inputs:

(1) *QueueNum: QueueNum* refers to the remaining vehicles in a lane during the green signal.

(2) *FrontNum*: *FrontNum* is the number of vehicles, which will go to the link during the green phase.

It has one Output:

Extend: This output translates the possibility of extending the green phase. Extend consists 5 membership functions.

There are 10 rules in this module:

According to this rule in Table 4 if the value of *QueueNum* increases, the value of the output *Extend* also increases proportionally. But when *FrontNum* value increases, *Extend* value decreases. Toget the output values of *GreenPhase Module*, need to combine all the values of *Extend* of every lane in that green phase.

Table 4: GreenPhase Module Fuzzy Rules [3]

RULE	INPUTS		OUTPUTS
	QueueNum	FrontNum	Extend
1.	Z		Z
2.	S	S	S
3.	S	M	S
4.	S	L	S
5.	M	S	L
6.	M	M	M
7.	M	L	S
8.	L	S	VL
9.	L	M	VL
10.	L	L	L

The Decision Module:

By comparing the inputs *Urgency* and *Extend* Decision Module determines whether to change the phase or extend the green signal. This Module has 3 inputs: (1) *Selected Phase,* (2), *Urgency* and (3) *Extend.* These inputs are the outputs of *NextPhase* and *GreenPhase* Modules. When the *Urgency* is higher than *Extend,* the next phase has heavier traffic condition then the current green phase and the green signal at the current phase will not be extend. So, the output will change the phase.

Table 5: Rules derived for the Decision Module [3]

RULE	EXTEND	URGENCY	DECISION
1.	Z	Z	NO
2.	Z	S	YES
3.	Z	M	YES
4.	Z	L	YES
5.	Z	VL	YES
6.	S	Z	NO
7.	S	S	NO
8.	S	M	YES
9.	S	L	YES
10.	S	VL	YES
11.	M	Z	NO
12.	M	S	NO
13.	M	M	NO
14.	M	L	YES
15.	M	VL	YES
16.	L	Z	NO
17.	L	S	NO
18.	L	M	NO
19.	L	L	NO
20.	L	VL	YES
21.	VL	Z	NO
22.	VL	S	NO
23.	VL	M	NO

Table 5 shows the 23 rules of Decision Module. Here **Yes** refers to end of green signal phase and change to new phase, and **No** refers to continuing the green signal.

Max-min inference and Centroid Defuzzyfication:

Max-min inference procedure and the Centroid Defuzzyfication strategy have been used in all **3** modules of the fuzzy controller.

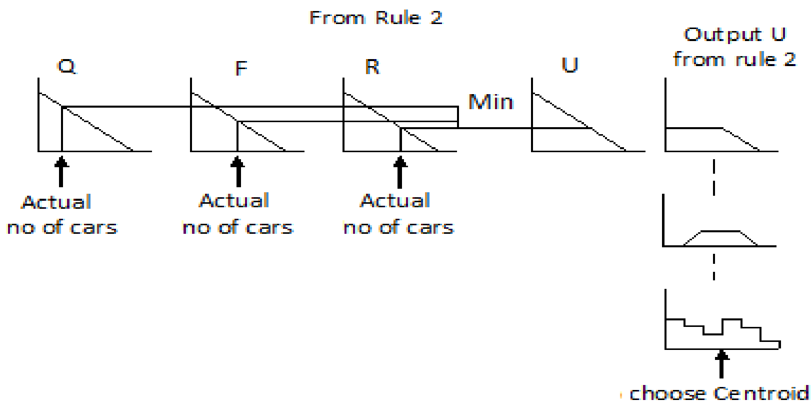


Figure 12: Max-min inference and Centroid Defuzzyfication of NextPhase Module

Table 6: First two rules of Table 3

RULE	INPUTS			OUTPUTS
	QueueNum	FrontNum	RedTime	Urgency
1.	Z			Z
2.	S	S	S	S

This diagram in Figure 12 demonstrates max-min inferencing and centroid defuzzification for NextPhase Module with input QueueNum, FrontNum, and RedTime and an output Urgency. From Table 6, 2nd rule of NextPhase module we can see that if the values of all 3 inputs are small, output Urgency also small. Figure 12 shows the actual numbers of cars of QueueNum(Q), FrontNum(F), and RedTime(R). For all rules, it takes the min value for the output Urgency(U) and clip the graph. Defuzzyfy the output to convert the fuzzy outputs to a single number. In centroid Defuzzyfication it takes the maximum of output and chooses the centroid.

Design of Traffic Simulator:

A simulator was developed showing in Figure 13 using Visual Basic which reflects 4 intersections of a complex junction. In the simulation to reflect the real traffic flow they designed the headways of vehicles, which is the time intervals between successive vehicles passing through a fixed point. It also considered the three types of vehicles (1) *Passenger cars,* (2) *Buses* and (3) *Medium And Heavy Vehicles* such as Lorries and trucks. Real traffic is affected by the ratio of these three types of vehicles due to their different lengths and headways. The simulator can be found at www.cairo.utm.my

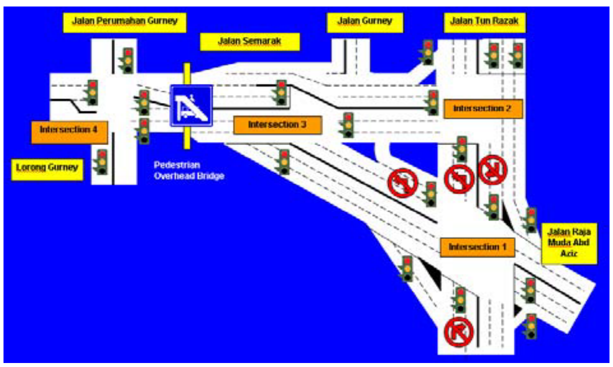


Figure 13: Graphical User Interface of the Traffic Simulator showing the 4 intersections [3]

Simulation Results and Discussions:

In order to compare the results, 3 types of performance indices were used: (i) flow density average, (ii) link overflow and (iii) delay time average.

*(i) Flow density average***:** The average total number of Passenger Car Unit (pcu) in all these 4 intersections at 1 second is called Flow density average.

(ii) Link overflow: It indicates the number times and number of links that are in overflow condition.

(ii) Daily time average: This is the average pcu waiting time at every link before each is allowed to cross the intersection.

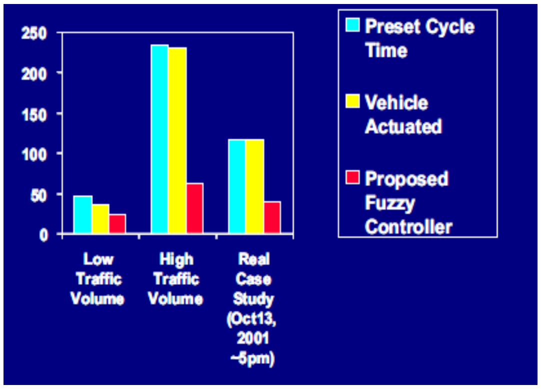


Figure 14: Performance index of flow density average [3]

In order to compare it considered, flow density average as the most important performance indicator, followed by link overflow and delay time average. This Figure 14 shows the results of the 3 traffic controllers based on three traffic volumes: low, high and real.

As mentioned before if the value of the flow density average is lower the performance of the controller will be better. From this graph we can observe in the first simulation, three traffic controllers has been tested under low traffic volume and can not find any significant difference, but the fuzzy controller showed slightly better performance. In the second simulation exercise we can see the improvement of the fuzzy controller in the flow density average compared with other two controllers under high traffic volume. The third simulation was done under actual traffic condition when traffic condition was extremely bad and they found the result showing in following table.

Table 7: Comparison of Simulation Results based on Actual Traffic Conditions of the Junction

	Preset Cycle Time	Vehicle Actuated	Proposed Fuzzy Controller
Link Overflow (times)			
Intersection 1	4013	5239	43
Intersection 2	3232	2786	0
Intersection 3	1794	2201	45
Intersection 4	3548	3660	1719
Flow Density Average (pcu)			
Average	117.02	116.37	39.83
Flow Density Comparison with Fuzzy Controller			
Improvement over Preset Cycle Time			55.91%
Improvement over Vehicle Actuated			54.86%
Delay Time Average (seconds)			
Average	38.67	38.95	39.58
Delay Time Comparison with Fuzzy Controller			
Improvement over Preset Cycle Time			-3.12%
Improvement over Vehicle Actuated			-2.38%

In Table 7 we can observe that in terms of Link Overflow in all four intersections fuzzy traffic controller performed better than other two controllers. It also performed better, which is more than 50% over other two controllers in terms of Flow Density Average. In case of high traffic volume, the fuzzy traffic controller did not performed better than other two controllers, for that reason we can see here the Delay Time Average of the fuzzy controller is not better than other two controllers. But the difference is very small. So, we can conclude that overall fuzzy controller shows better performance over other two controllers.

2.4 Case study 4

Fuzzy Logic Based Traffic Light Controller [4]:

Developing an intelligent traffic light control system can help to solve the traffic congestion in many cities. Most of the regular traffic light control systems are based on the fixed time duration of the green phase, which can change the traffic signals at constant cycle time. This type to traffic control system cannot extend the current green light time duration based on the current traffic situation and cannot reduce the vehicle's waiting time at the red light. A traffic light controller based on fuzzy logic can be used for optimum control of fluctuating traffic volumes such as over saturated or unusual load conditions [4].

This paper takes about the implementation of fuzzy logic control system that can be used for an intersection. The proposed fuzzy traffic light controller, introduced in this paper, is able to manage the congestion better than regular fixed time duration based traffic control system. This fuzzy traffic control system can minimize the vehicle waiting time at red light. It can extend a current green phase by adding different time duration by applying some fuzzy rules.

Fuzzy Traffic Signal Control:

The main goal of this control system is to decide whether to terminate the current phase and change to the next most appropriate phase or continue the current phase [4]. This controller achieve this goal by following two strategies: (1) estimate the traffic intensity on each approach and (2) decide whether to extend or terminate the current phase. In order to estimate the traffic intensity on each approach they used vehicle detector that detects the incoming vehicles. By applying some fuzzy rules this fuzzy system take decision whether to extend or terminate the current phase and go to the next phase.

Following Figure 15 shows the 4 lane intersection where they used proposed fuzzy logic controller. Here we can see that they installed detectors on upstream-line and stop-line on each approach, which will estimate the number of approaching vehicles on each approach during given time interval.

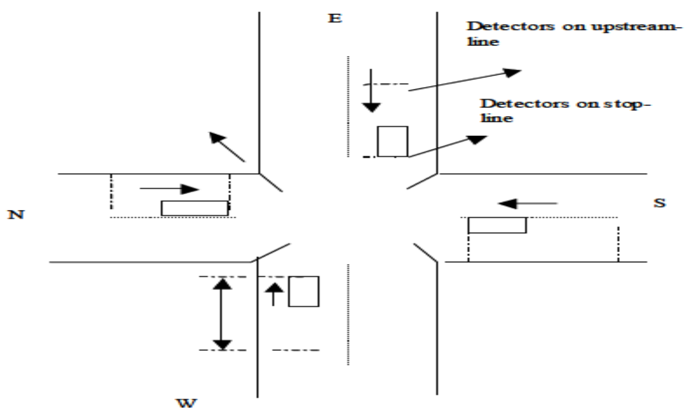


Figure 15: An intersection with detector configuration [4]

The authors designed the traffic intersection simulation in VB6 and fuzzy rule base by using MATLAB tools. There are two inputs and one output fuzzy variables are considered in this paper. They are: arriving vehicles (A), queuing vehicles (Q) and one output variable extension (EXT). Figure 16 shows the membership functions for the arriving vehicles (A). The membership functions for the arriving vehicles (A) at the approach having green phase are few = -4 to 4, small = 0 to 8, medium = 4 to 12 and many = 8 to 16 [4].

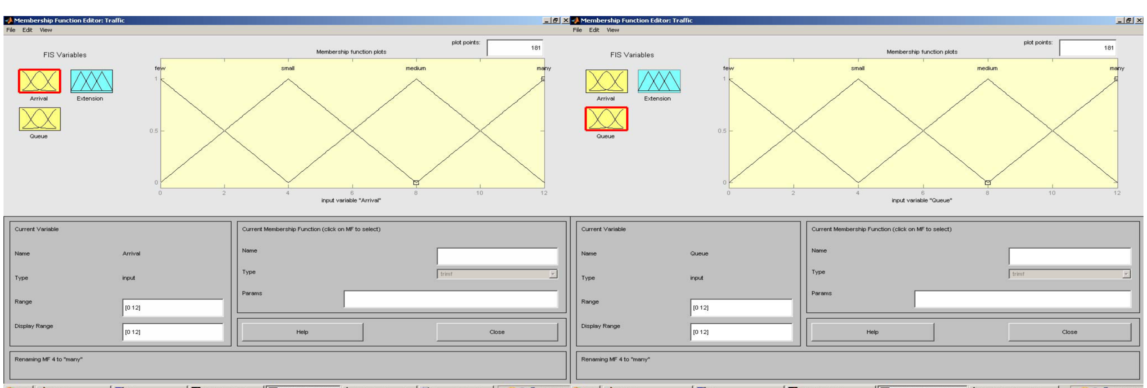


Figure 17: Input Fuzzy Variable 1-Arrival Figure 16: Input Fuzzy Variable 2-Queue

Figure 17 shows the membership functions for the queuing vehicles (Q).The membership functions for the queuing vehicles (Q) at the next approach having red phase are few = -4 to 4, small = 0 to 8, medium = 4 to 12 and many = 8 to 16 [4].

By applying the *IF Then* rules the fuzzy system determines the time duration, which is the output *Extension,* by using tow of its input. The membership functions for the output fuzzy variable extension (EXT) are: zero = 0 to 5 sec, short = 5 to 10 sec, medium = 10 to 15 sec and long = 15 to 20 sec [4]. If the green time is extended after the minimum green time elapsed, the fuzzy controller will determine whether to extend the green light again after the time interval Δt ($\Delta t = 5$ sec.) or terminate the green light.

Fuzzy Rules:

Following are the some example of fuzzy rules they used in this paper to determine whether to extend or terminate the green light after the initial time interval elapsed.

After a minimum green (5 s) [4]

If Arrival is few AND Queue is (few OR small OR medium OR many) then Extension is zero.

Else if Arrival is small AND Queue is (few OR small) then Extension is short.

Else if Arrival is small AND Queue is (medium OR many) then Extension is zero [4].

After the first extension i.e. (ext1 + min. green of 5 s) [4]

If Arrival is few AND Queue is (few OR small OR medium OR many) then Extension is zero.

Else if Arrival is small AND Queue is (few OR small) then Extension is zero [4].

From the number of waiting vehicles at each red light it derived weight W(A) of an approach. The values of W(A) are zero (Z) = -4 to 4, low (L) = 0 to 8, medium (M) = 4 to 12 and high (H) = 8 to 16 [4].

Following are the some example of fuzzy rules used in this paper to determine which approach will get the maximum priority for the green time after the extension. W(E), W(W), W(S) and W(N) denote the weights associated with 'east', 'west', 'south' and 'north' approaches respectively.

If W (W) is high then next green phase is W

Else if W (W) is medium and W (S) and W (E) is low then next phase is W

Else if W (W) and W (S) and W (E) is low then next phase is W [4]

Results:

Figure 18 shows the comparison of the performance, between the fixed time traffic controller and the fuzzy traffic controller, based on number of waiting vehicles at an approach. From this graph we can observe that the fuzzy controller performing better than the fixed time controller.

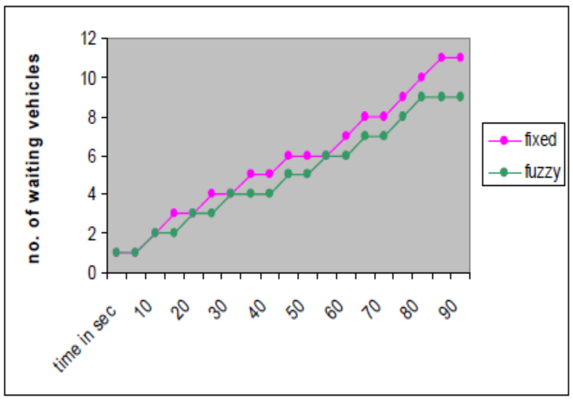


Figure 18: Graph comparing performance of Fixed and Fuzzy Controller [4]

3. Proposed Fuzzy Traffic Light Control system by optimizing green light duration:

Problem Statement:

Increasing number of vehicles and shortages of roads led to traffic congestion in many cities. This congestion affects the efficiency, productivity and energy losses. One major factor of this congestion is the traffic light system that controls the traffic light at intersections. Most of the regular traffic system does not consider variance in traffic light time duration and can not optimize the performance. Our proposed fuzzy traffic light control system can provides better solution of traffic congestion. Instated of fixed time duration it considers different time length for the green light time duration depending on the current traffic density of the intersection. By optimizing the green light time duration our system can handle the congestion better than other regular systems.

Description:

We consider the following intersection of Figure 19 (a), for our proposed system. Here N, S, E, W stands for North, South, East, and West. Figure 19 (b) is our proposed traffic control system.

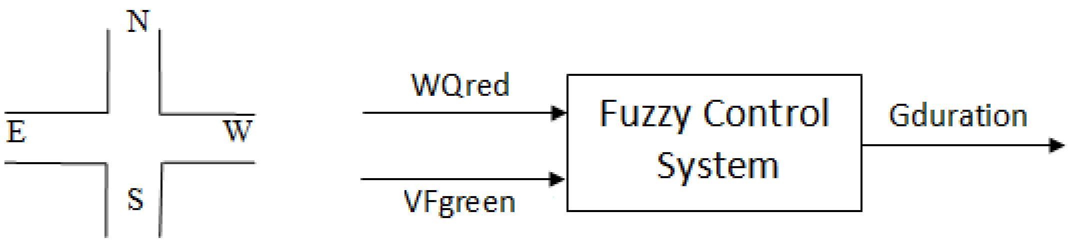


Figure 19: (a) Intersection and (b) Controller

We consider two inputs which is WQred and VFgreen in order to produce output Gduration. The system applies fuzzy rules on its two inputs and produces the output for the Gduration.

WQred: WQred is one input of the control system that refers to the number of vehicle is waiting during the red light, which is detected by the detector installed in the intersection. Fuzzy sets of WQred has three membership function: Low (L) = 0 to 6.4, Medium (M) = 4 to 12 and High (H) = 9 to 16 or more.

VFgreen: This is the 2nd input of the system, which is the number of vehicles are passing through a fix point in 5 seconds. There are detectors at the fix point of each lane of the intersection that detects the vehicles passing through it. It has three membership functions: Low (L) = 0 to 5, Medium (M) = 3 to 9 and High (H) = 7 to 14 or more.

Gduration: This is the only output of our Traffic control system. Gduration stands for Green light time duration that determines the time duration of the green light signal. Gduration has also three membership functions: Short (S), Medium (M) and High (H) that will achieved by manipulating inputs and finally produce a crisp output after defuzzification which has the value 10, 15 and 20 seconds respectively.

The output of the system is the time duration of the green traffic light on the basis of the current congestion situation of the intersection. The detector counts the numbers of approaching vehicles pashing through a fixed point during the green light. It starts counting just after the green signal start until 5 seconds. After 5 seconds it will send the numbers of vehicle information as an input of *VFgreen* to the fuzzy controller. The fuzzy control system also gets the number of waiting vehicles as an input from *WQred.* Finally by applying fuzzy rules the system generates the output of *Gduration* on the basis of inputs *VFgreen and WQred,* and defines the crisp time duration for the green light. Assume that, the system calculates the output before elapsed 10 seconds of the green light duration. Once the green light time duration elapsed, the controller will start the next phase.

Fuzzy Rules:

On the basis of input *WQred and VFgreen* the fuzzy controller applies nine IF THEN rules and gets the output fuzzy value of *Gduration.* Table 8 represents nine fuzzy rules of the fuzzy control system. Here L, M, H and S means Low, Medium, High and Short. For the *Gduration* the time duration we considered 10 seconds for *Short,* 15 seconds for *Medium,* and 20 seconds for *High.*

Table 8: Fuzzy rules of proposed Fuzzy Control System

Rules	WQred	VFgreen	Gduration
1	L	L	S
2	L	M	M
3	L	H	H
4	M	L	S
5	M	M	M
6	M	H	M
7	H	L	S
8	H	M	S
9	H	H	S

For example, in rule 1, if the WQred is Low and VFgreen is also Low, then the output Gduration will be short. Which interprets that time duration of green light will be short. The meaning of this rule is that when there are only few vehicles are waiting at red light and also few vehicles are passing through the green light, the time duration of the green light should be short, so that the vehicles are waiting at red light can cross the intersection without waiting for a long time at red light and before the red light's vehicles queue gets high. This will reduce the vehicle waiting time at red light.

4. Simulation results and discussion

The Fuzzy Logic Toolbox in Matlab which has a graphical user interface (GUI) to efficiently design and implement fuzzy control system. Although it's possible to use the Fuzzy Logic Toolbox by working strictly from the command line, in general it's much easier to build a system graphically reported in [1]. There are five primary GUI tools for building, editing, and observing fuzzy inference systems in the Fuzzy Logic Toolbox are Fuzzy Inference System or FIS Editor, the Membership Function Editor, the Rule Editor, the Rule Viewer, and the surface viewer. They are dynamically linked as any change done to the FIS eventually reflected to the system. At first we use the FIS editor to design the system. By typing *Fuzzy* in the command prompt let us open the FIS editor where two inputs (WQred and VFgreen) and one output (Gduration) are set as discussed above in the section *Description*. Figure 20 shows our proposed fuzzy inference system and membership function of the input and output as discussed in the previous section are shown in Figure 21, 22 and 23 respectively.

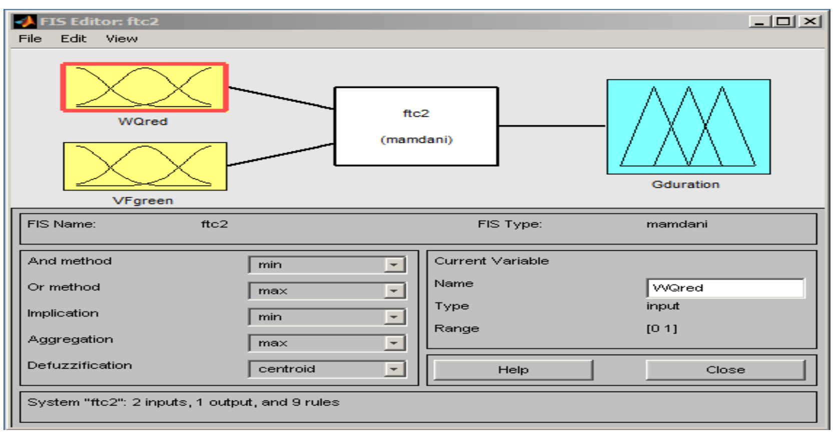


Figure 20: FIS editor

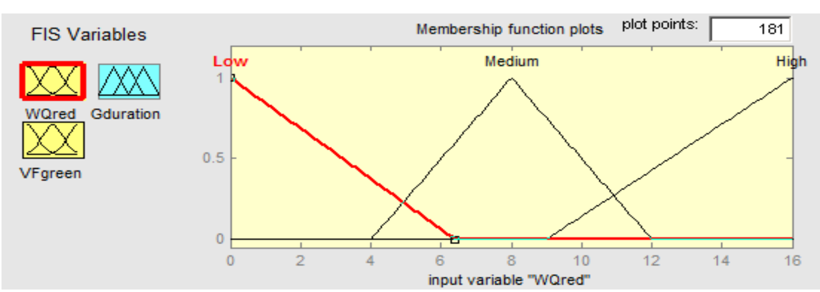


Figure 21: Membership function (WQred)

WQred: low=0 to 6.4, Medium= 4 to 12, High= 9 to 16

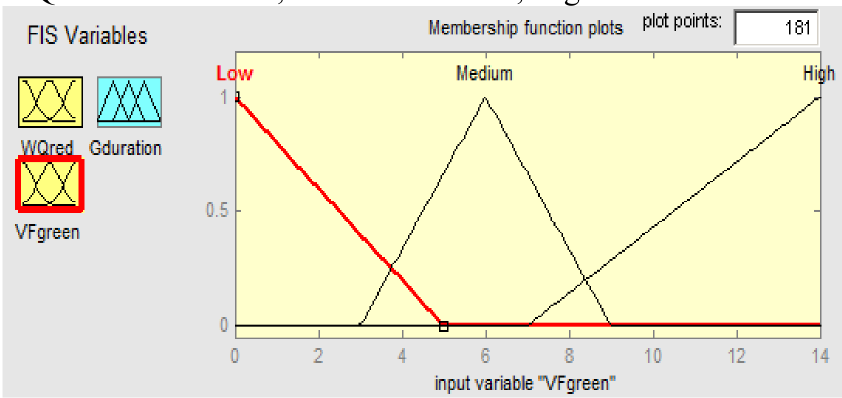


Figure 22: Membership function (VFgreen)

VFgreen: low=0 to 5, Medium= 3 to 9, High= 7 to 14

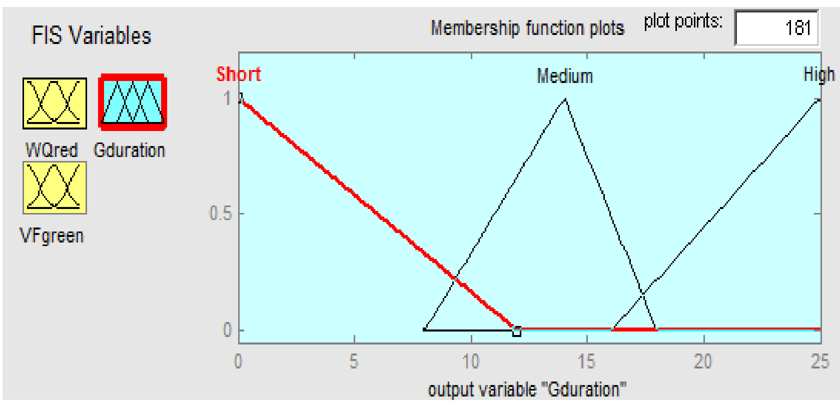


Figure 23: Membership function (Gduration)

Gduration: Short=0 to 12, Medium= 8 to 18, High= 16 to 25

The rules defined in the previous section can be easily derived using rule editor shown in Figure 24 of the FIS and can view graphically using rule viewer shown in Figure 25, where the aggregate output are shown at the bottom and center of the area is considered for defuzzification as shown in red color.

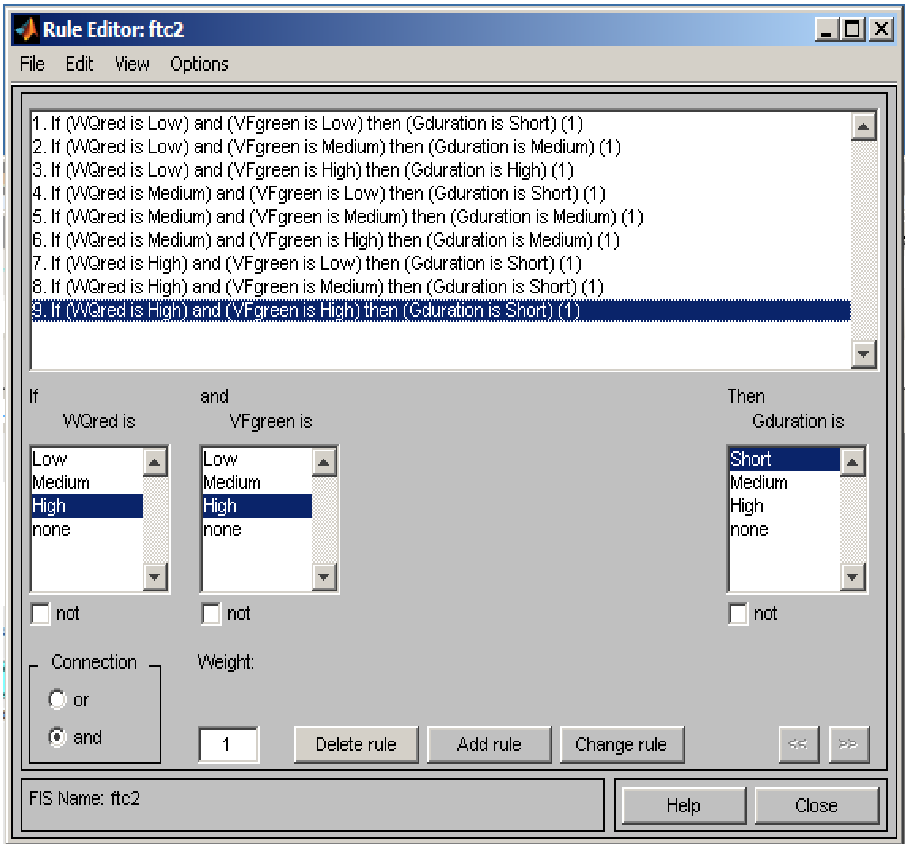


Figure 24: Rule editor

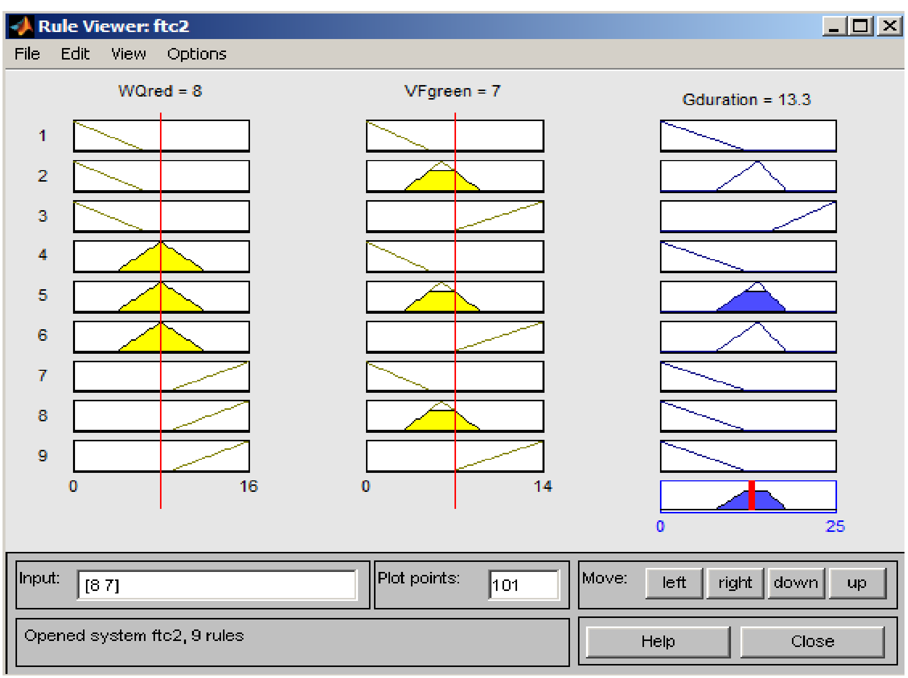


Figure 25: Rule Viewer

FIS output surface: Figure 26 shows the surface view of the system that represents the input and output relationship in terms of data distribution. The Surface Viewer can generate a three-dimensional output surface where two of the inputs vary.

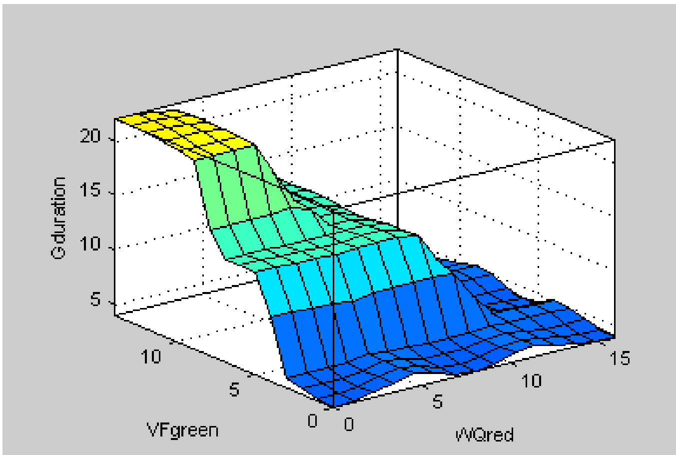


Figure 26: Surface viewer

5. Conclusion

This paper discusses fuzzy traffic control system in general. Some case studies were performed on fuzzy traffic control system while a more complex system can be found in many literatures. From the study, it is observed that regular traffic control system can be improved using fuzzy logic. Most of the cases, fuzzy traffic control system were simulated using Matlab, where fuzzy rules were implemented and fuzzy inference system were realized. After simulation, a single chip fuzzy hardware implementation is possible using state of the art design technology as discussed in some studies. In our proposed system green phase duration is optimized based on current traffic congestion in green signal and waiting vehicle in the red signal lane. The system dynamically change the green light duration based on nine rules to maintain better traffic flow in the multilane intersection. The system is simulated using Matlab and pictorial representation demonstrates the whole fuzzy modeling system. Also the Matlab codes for the FIS are provided in Appendix. It is obvious that by applying fuzzy reasoning in control system especially in this fuzzy traffic control application clearly improve the performance in traffic control system than using a conventional system.

6. Future work

In future, pedestrian control logic can be added with this fuzzy traffic controller. All the traffic controller in the city can be connected by creating network connection and can communicate with each other with sending information about current traffic situation of their intersections, which will be using as inputs for the fuzzy controller. On the basis of these inputs the controller will produce output to better control of the congestion considering the current traffic situations of other neighbour traffic controller's intersection of the network.

7. References

[1] Matlab: Fuzzy logic toolbox.
Available online: http://www.mathworks.com/products/fuzzy-logic/index.html

[2] Ambre, Mandar, "A Design Methodology for the Implementation of Fuzzy Logic Traffic Controller using Field Programmable Gate Arrays" Master of Science Thesis, Department of Electrical and Computer Engineering, FSU, 2004.
Available online: http://etd.lib.fsu.edu/theses/available/etd-04122004-164143/

[3] Khalid, M. et al., "Control of a complex Traffic Junction using Fuzzy Inference" In proceedings of 5th Asian Control Conference, 2004.
Available online: http://ascc2004.ee.mu.oz.au/proceedings/papers/P227.pdf

[4] G.H. Kulkarni, and P.G. Waingankar, "Fuzzy logic based traffic light controller", Second International Conference on Industrial and Information Systems, ICIIS 2007. 8 – 11 August, Sri Lanka,2007, pp107-110.

[5] Md. Shabiul Islam et al.,"Hardware Implementation of Traffic Controller using Fuzzy Expert System", The 2nd International Symposium on Evolving Fuzzy System (EFS'06)", 7-9 September 2006, Lake District, UK.

[6] Malaysia Road Work Department (JKR), "A Guide to the Design of Traffic Signals", Technical Guide 13/87, Kuala Lumpur, 1987.

[7] Jarkko Niittymäki, "Installation and experiences of field testing a fuzzy signal controller", European Journal of Operational Research 131 (2001), pp 273 - 281.

[8] M. B. Khalid, “Module 1: Basics of Fuzzy Sets and Fuzzy Relations”, Unpublished Course Notes, Universiti Teknologi Malaysia.

Appendix

(Matlab Code for the FIS)

```
1.  Name             ftc2
2.  Type             mamdani
3.  Inputs/Outputs   [2 1]
4.  NumInputMFs      [3 3]
5.  NumOutputMFs     3
6.  NumRules         9
7.  AndMethod        min
8.  OrMethod         max
9.  ImpMethod        min
10. AggMethod        max
11. DefuzzMethod     centroid
12. InLabels         WQred
13.                  VFgreen
14. OutLabels        Gduration
15. InRange          [0 1]
16.                  [0 1]
17. OutRange         [0 1]
18. InMFLabels       Low
19.                  Medium
20.                  High
21.                  Low
22.                  Medium
23.                  High
24. OutMFLabels      Short
25.                  Medium
26.                  High
27. InMFTypes        trimf
28.                  trimf
29.                  trimf
30.                  trimf
31.                  trimf
32.                  trimf
33. OutMFTypes       trimf
34.                  trimf
35.                  trimf
36. InMFParams       [-0.4 0 0.4 0]
37.                  [0.1 0.5 0.9 0]
```

38. [0.6 1 1.4 0]
39. [-0.4 0 0.4 0]
40. [0.1 0.5 0.9 0]
41. [0.6 1 1.4 0]
42. OutMFParams [-0.4 0 0.4 0]
43. [0.1 0.5 0.9 0]
44. [0.6 1 1.4 0]
45. Rule Antecedent [1 1]
46. [1 2]
47. [1 3]
48. [2 1]
49. [2 2]
50. [2 3]
51. [3 1]
52. [3 2]
53. [3 3]
45. Rule Consequent 1
46. 2
47. 3
48. 1
49. 2
50. 2
51. 1
52. 1
53. 1
45. Rule Weigth 1
46. 1
47. 1
48. 1
49. 1
50. 1
51. 1
52. 1
53. 1
45. Rule Connection 1
46. 1
47. 1
48. 1
49. 1
50. 1
51. 1
52. 1
53. 1

FIS in a txt file format:
[System]

```
Name='ftc2'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=9
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='WQred'
Range=[0 1]
NumMFs=3
MF1='Low':'trimf',[-0.4 0 0.4]
MF2='Medium':'trimf',[0.1 0.5 0.9]
MF3='High':'trimf',[0.6 1 1.4]

[Input2]
Name='VFgreen'
Range=[0 1]
NumMFs=3
MF1='Low':'trimf',[-0.4 0 0.4]
MF2='Medium':'trimf',[0.1 0.5 0.9]
MF3='High':'trimf',[0.6 1 1.4]

[Output1]
Name='Gduration'
Range=[0 1]
NumMFs=3
MF1='Short':'trimf',[-0.4 0 0.4]
MF2='Medium':'trimf',[0.1 0.5 0.9]
MF3='High':'trimf',[0.6 1 1.4]

[Rules]
1 1, 1 (1) : 1
1 2, 2 (1) : 1
1 3, 3 (1) : 1
2 1, 1 (1) : 1
2 2, 2 (1) : 1
2 3, 2 (1) : 1
3 1, 1 (1) : 1
3 2, 1 (1) : 1
3 3, 1 (1) : 1
```