World Journal of Advanced Engineering Technology and Sciences

eISSN: 2582-8266

Cross Ref DOI: 10.30574/wjaets

Journal homepage: https://wjaets.com/

(Research Article)

Check for updates

Smart fuzzy logic-based model of traffic light management

Isdore Onyema Akwukwaegbu [1, *], Eleazar Benson Mfonobong [2], Jude-Kennedy Chibuzo Obichere [3] and Chiedozie Francis Paulinus-Nwammuo [4]

[1] *Electrical Engineering Department, Federal University of Technology Owerri, Nigeria.*
[2] *Electronic Engineering Department, Federal University of Technology Owerri, Nigeria.*
[3] *Mechatronics Engineering, Federal University of Technology Owerri, Nigeria.*
[4] *Electrical Engineering, Federal University of Technology Owerri, Nigeria.*

World Journal of Advanced Engineering Technology and Sciences, 2023, 08(02), 344–358

Publication history: Received on 24 February 2023; revised on 15 April 2023; accepted on 17 April 2023

Article DOI: https://doi.org/10.30574/wjaets.2023.8.2.0108

Abstract

Traffic congestion difficulties have resulted in low productivity, significant air pollution, and energy losses in Owerri Metropolis, Nigeria. The paper looked at the design of a smart fuzzy logic traffic light management module, the development of a traffic control program using an Arduino microcontroller system, and the validation of the developed program's functionality using a Proteus circuit model to confirm the efficiency of fuzzy signal control. The traffic light environment of a fuzzy logic controller is simulated using Matlab software, and isolated traffic of multiple junctions is simulated using the Sumo Urban Mobility Simulation (SUMO) environment. The performance of the traditional fixed-time controller vs the fuzzy logic traffic light controller is examined. To allocate pedestrian crossing the right of way, traffic light systems function in tandem with pedestrian displays. The simulation results indicated that the overall durations for the fixed traffic light controller and suggested smart fuzzy logic traffic light controller simulations are 1,426 seconds and 1,328 seconds, respectively. The results also showed that the suggested smart fuzzy logic traffic light controller recorded significant waiting and movement times, indicating its standard appropriateness for Owerri city's various crossings.

Keywords: Smart Fuzzy Logic; Traffic Urgency Decision Module (TUDM); Extended Time Decision Module (ETDM); Sumo Urban Mobility Simulation(SUMO)

1. Introduction

Heavy traffic congestion, which varies according to season, weather, time of day, and building activity, has caused major issues in Owerri. The existing traffic light control system in Owerri city, which is based on a timer, a programmable logic controller (PLC), and a microcontroller, is ineffective. Based on the traditional traffic light control system design, Owerri City has recorded strong traffic lights in the morning before office hours, and evening after office hours. But at midday with no traffic, cars will still wait for a long time before being passed.

The clever fuzzy logic traffic control system, which makes use of an Arduino microcontroller and ultrasonic sensors, is used to govern traffic signal flow at various crossings. The microcontroller maintains the number of cars in its memory and employs ultrasonic sensors to detect the absence and presence of automobiles at various junctions. The fuzzy logic-based microcontroller decision defines distinct ranges for traffic signal delays and updates accordingly. The duration of green time at each junction is determined by the fuzzy logic controller. The two separate modules studied in this study are the traffic urgency decision module (TUDM) and the extended time decision module (ETDM). The TUDM computes urgency for all red light phases, whereas the ETDM computes green light time in relation to phase extension time, with more urgency proportionate to the number of automobiles. The Matlab program is designed to simulate the state of an

* Corresponding author: Akwukwaegbu Isdore Onyema

isolated signalized junction based on fuzzy logic to handle vehicle arrival and queueing in phases. SUMO is used to simulate several traffic junctions and traffic data parameters are gathered and produced in a Python environment. The construction of an intelligent control structure guarantees an ideal solution for all transportation and road traffic system participants [1, 2]. Conventional cyclic light control manages road crossings, while sensor-based traffic control monitors traffic in different directions, and the signals acquired regulate the traffic lights [3, 4]. Interpolation and fuzzy approaches are used at road crossings and rail tracks [5, 6, 7, 8, 9]. Fuzzy logic has been shown to be very beneficial in expert systems and other artificial intelligence applications [9, 10, 11, 12]. The flexibility of human reasoning is implemented by a mathematical model that includes fuzzification, fuzzy logic inference utilizing if-then rules, and defuzzification [11, 12, 13, 14, 15, 16].

The membership function is a graphical depiction of the magnitude of each input's involvement. It assigns a weight to each of the processed inputs, establishes functional overlap between inputs, and finally calculates an output response. The rules employ the input membership values as weighting factors to assess their impact on the final output conclusion's fuzzy output sets. After the functions have been inferred, scaled, and concatenated, they are defuzzified into a crisp out that powers the system [17, 18, 19, 20, 21]. The purpose of transportation research is to optimize the movement of people and products. In many nations, monitoring and controlling city traffic is becoming a big issue. Because the number of road users is continually increasing, and the resources given by present infrastructures are limited, the Federal Road Safety Corps must devise new methods or measures to address this issue [1, 2]. Although it is understood that automatic control systems should relieve people of manual control, such systems do not operate effectively in many situations, particularly when oversaturated or unexpected load conditions, which might be due to limits of the algorithms or sensing devices [2]. This study describes the creation of a smart traffic light control system based on a microcontroller-based fuzzy logic approach capable of emulating human intelligence for traffic light control. Fuzzy logic technology enables the application of real-world rules in a manner akin to how humans think. The fuzzy logic controller can detect not only the presence of cars waiting at the intersection, but also the quantity of cars waiting at the intersection [2]. The program can mimic traffic light management using both traditional fixed-time and fuzzy logic controllers. To simulate and control the functioning of various traffic junctions, software based on Proteus, SUMO, Matlab, and Python is built and deployed. The suggested smart fuzzy logic traffic light controller is capable of resolving Owerri's traffic difficulties.

Multiple traffic crossings are analyzed, a traffic light management system is modeled, and algorithms are evaluated to enhance overall traffic flow in Owerri and its environs. In this experiment, ultrasonic sensors are employed to collect data from automobiles at several junctions. To mimic the efficacy of the fuzzy logic controller in regulating traffic conditions at isolated multiple intersections, software programs are built. The Matlab tool is used to construct the fuzzy logic traffic light controller, and SUMO is used to develop the simulation program. The fuzzy logic controller's inference method is similar to the human thinking process. This is where fuzzy logic technology and artificial intelligence intersect. Humans utilize rules instinctively to carry out their acts. The fuzzy logic controller is in charge of adjusting the length of the green time based on traffic conditions. Each phase of the traffic light has its own condition. When no incoming traffic is identified, the system enters a single default state. This is the default state, and it correlates to the green time for a given technique, generally the main approach. If there are no vehicle queues for the relevant approach, a state in the sequence of states can be skipped.

2. Method

The Sumo Urban Mobility Simulation (SUMO), Python, Arduino microcontroller, ultrasonic sensors, Proteus, and Matlab software were utilized in this work.

On the other hand, and more importantly, creating an intelligent traffic light control system will undoubtedly address Owerri's traffic congestion issues. The modernization of the city's existing traffic light control systems is based on the set time duration of the green phase, which may change traffic lights at a consistent cycle time. Based on the current traffic circumstances, this sort of traffic management system cannot prolong the current green light time length and cannot minimize the vehicle's waiting time at the red light. A fuzzy logic-based traffic light controller may be utilized for optimal regulation of traffic volumes, such as oversaturation or exceptional load circumstances [20, 21, 22, 23, 24, 25, 26, 27, 28]. The paper proposes the deployment of a developed fuzzy logic controller system for regulating traffic signals at numerous intersection models in the city. The suggested fuzzy traffic light controller, which will be implemented at several intersection models, would handle congestion better than the existing fixed time duration traffic management system. This fuzzy traffic control technology helps reduce vehicle waiting time at city red lights. It can extend a current green phase by introducing new time durations using fuzzy rules as illustrated in an algorithm for the smart traffic management system.

The planned smart traffic lights are compared to the present manual traffic signals. The proposed traffic light model will calculate the number of automobiles on the route and optimize the lights based on the data. Roadside units (RSU) comprised of ultrasonic sensors are used to compute the number of automobiles on the road via their connections to RSU, and after the data is gathered, it is transmitted to the algorithm to determine traffic light cycles. This method allows for the length of red and green lights to be changed in response to current traffic congestion on the route. As a result, the duration of red and green lights will vary dynamically throughout the day. This method alleviates traffic-light-caused congestion while also managing traffic flow for high-priority vehicles such as ambulances.

Procedures involved in an algorithm for smart traffic light management development are presented as follows:

- Start.
- Design and simulate a fixed timed traffic light using Proteus.
- Create fuzzy logic rules for intersection traffic light using Matlab.
- Implement fuzzy logic rules into Proteus traffic light with Arduino microcontroller and ultrasonic sensors and simulate.
- Design GUI for traffic light simulation using SUMO.
- Run simulations for both fixed timed traffic light and smart traffic light using PYTHON.
- Display the results.
- End.

2.1. Block Representation of the Proposed Smart Traffic Light System

A smart traffic light controller is created by combining an ATMega AVR-32 with ultrasonic sensors that monitor lane density. The created electrical hardware is utilized to mimic traffic lights in order to provide a comprehensive automation solution for traffic light management. The creation of fuzzy algorithms is based on the information supplied by ultrasonic sensors into a microcontroller, which is then analyzed and provides a choice route for the lane to follow. The fuzzy logic will use high-density or low-density information in the traffic lane to determine which option to apply. Traffic light management on the lane will be handled intelligently with the deployment of the hardware and fuzzy algorithms, as illustrated in figure 1.

The Arduino, which is a fuzzy logic controller, is in charge of altering traffic signal intervals. It can communicate with and interact with sensors. Arduino boards are programmed using the Arduino Integrated Development Environment (IDE), which is freely available from the Arduino community.

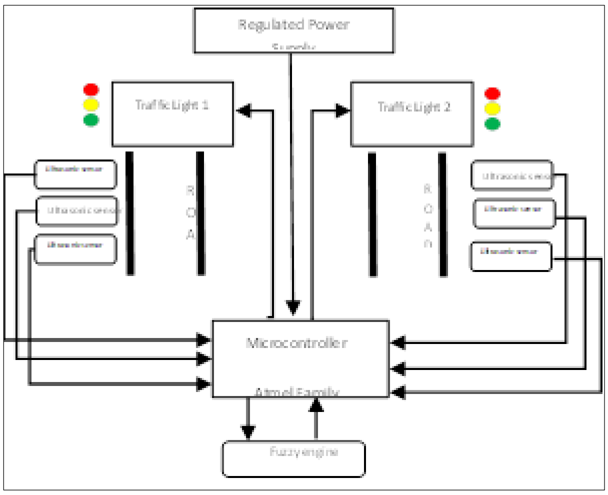

Figure 1 Block representation of the proposed smart traffic light control system

The code is referred to as a sketch, although it is essentially a C/C++ function set. Figure 2 depicts a set of ultrasonic sensors that are used for object detection using two parameters: echo and trigger. These sensors are properly installed to compute the limitations of vehicles in a certain lane. As depicted in Figure 3, the limit would determine the density of traffic, i.e. little traffic, moderate traffic, or heavy traffic. They will send the required information to the microcontroller.

As illustrated in Figure 3, the suggested smart traffic light controller is made up of an Arduino MEGA 2560 and a set of HC-SR04 ultrasonic sensors, which are utilized to construct the prototype in Proteus. The sensors monitor traffic density and provide it to the microcontroller through the wifi module. The microcontroller then assigns a time interval to the traffic signal. The same concept might be expanded up with larger capacity sensors and serve the same function. Previously, Infrared (IR) sensors were used to detect traffic density on the road, but they did not perform correctly owing to interference from the sun and another sensor close. The ultrasonic sensors were chosen due to their efficient functioning on sound reflection with a frequency of 40Khz and their inability to respond to interference induced by other sounds, as illustrated in figure 2 [15, 16, 17, 18, 19].

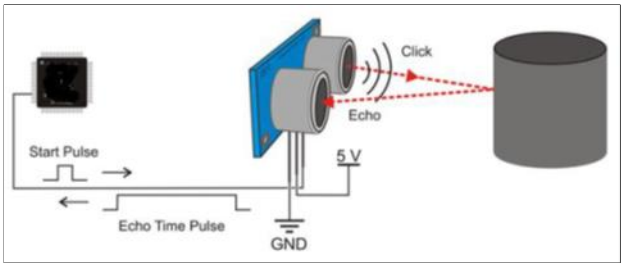

Figure 2 Action of ultrasonic sensor

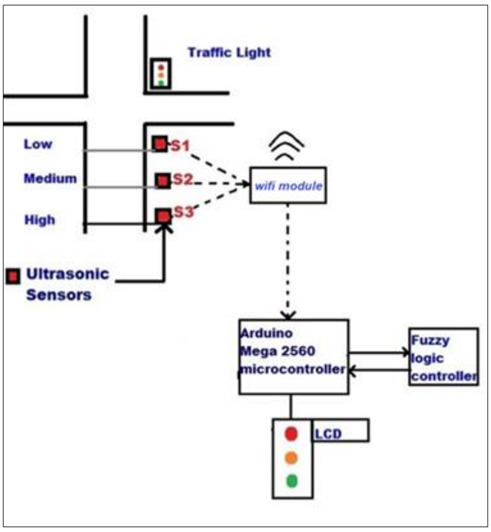

Figure 3 Pictorial model of the proposed smart fuzzy traffic light controller

2.2. Electronic Circuit Model of the Proposed Smart Traffic Light Controller

Figures 4 and 5 demonstrate how the existing traffic system and the created smart algorithm are constructed and implemented in a test rig in Proteus. The test rig is built using light-emitting diodes (LEDs) of various colors as well as traffic lights and ultrasonic sensors as sensors. In figure 4, the written software considers traffic signals at a predetermined time interval, where an available lane remains green till the time cycle elapses. The software put in the microcontroller, as illustrated in figure 5, covers a variety of traffic circumstances.

The suggested traffic system, shown in Figure 3, takes into account the amount of vehicles operating in a typical four-road junction. On each route, an equal number of sensors are connected to control the amount of cars at any one moment. When there is no vehicle in front of the sensors, no barrier is detected. As a result, it sends a high (logic 1) signal to the microcontroller. If any cars are identified by sensors, a low signal (logic 0) is delivered. The microcontroller receives all of the signals sent by the sensors. In the microcontroller, a software is put to detect the road with the most cars. The road with the most vehicles is cleared first by giving the relevant route a green light. The route with the second highest number of vehicles is then cleared in a similar manner, and so on. The interval between giving each route a green light is the same. When a green signal is given to a specific route to allow cars to pass, the other three roads are halted by a red light. The clever algorithm continues in this manner. Proteus 8 professional is used to shape and simulate a comprehensive system model. To imitate the system, a 20-second permitted period for vehicle passing is used, with a 1-second break for the yellow light to turn on at each road. While writing a program on a microcontroller, the time can be changed. Figure 6 depicts the simulation model of the proposed smart traffic signal controller.

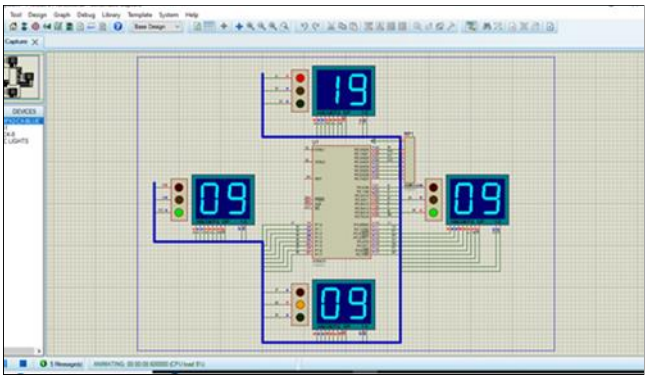

Figure 4 Design of Proteus interface of fixed traffic lights

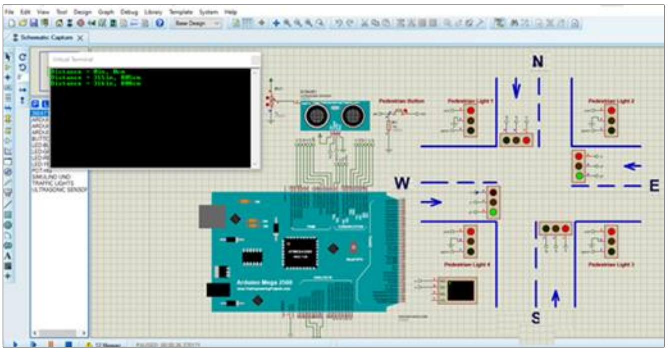

Figure 5 Proteus simulation model of the proposed smart traffic light controller

The car has 20 seconds to travel through the several junctions, while the changeover time is one second. The intended output is captured every 24 seconds by obtaining data from the ultrasonic sensors.

2.3. Development and Formulation of Fuzzy Logic Control System

Due to the instability of a traffic situation on the road, a traffic light management system in a junction is required [15, 16, 17, 18, 19]. It requires intelligent traffic signals. Fuzzy logic control is utilized to create smart traffic signals. The traffic light control system is built in this work by taking into account the traffic flow acquired from the number of incoming cars per minute at the various junctions. Figure 1 shows how the microcontroller manages the traffic lights based on the output of the fuzzy logic. The flow of the fuzzy process is illustrated in figure 6. The traffic junction model was created in MATLAB with the Image Processing Toolbox and the Fuzzy Logic Toolbox [11, 12]. The Fuzzy Logic Toolbox in Matlab includes a graphical user interface (GUI) for designing and implementing fuzzy control systems. The Fuzzy Logic Toolbox includes five core GUI tools for designing, changing, and viewing fuzzy inference systems: the Fuzzy Inference System or FIS Editor, the Membership Function Editor, the Rule Editor, the Rule Viewer, and the surface viewer. They are dynamically connected because any changes made to the FIS are reflected in the system. The FIS editor will be used to design the system at initially.

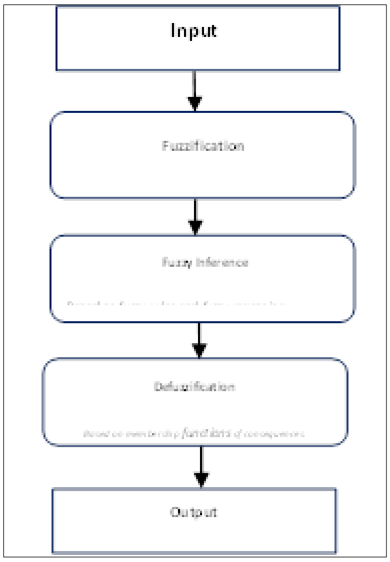

Figure 6 Flow of the Fuzzy Logic Process

The functional designs of the fuzzy logic control system procedures are discussed and presented as follows:

2.3.1. Design of Fuzzy Parameters and their Membership Functions(MF)

The fuzzy inference system Sugeno Method was introduced in 1985. This method is almost the same as Mamdani Method[11, 12]. The fuzzy inference system Sugeno Method is represented in Equation (1).

If Input 1 = x and Input 2 = y, then Output is given as

$$z = ax + by + c \qquad (1)$$

The output of this rule is not in membership functions form, but in a number form that varies linearly to the input variables. If a = b = 0, then the fuzzy inference system is zero order, because the output is a constant number, that is, z = c. The final output of Sugeno Method is calculated using Equation (2).

$$\text{Output} = \frac{\Sigma_{i=1}^{N} w_i z_i}{\Sigma_{i=1}^{N} w_i} \qquad (2)$$

Where: N = Number of rules, W_i = Weight or firing strength and Z_i = Output level.

Deployment rule of Sugeno method in fuzzy logic control system is shown in Figure 7.

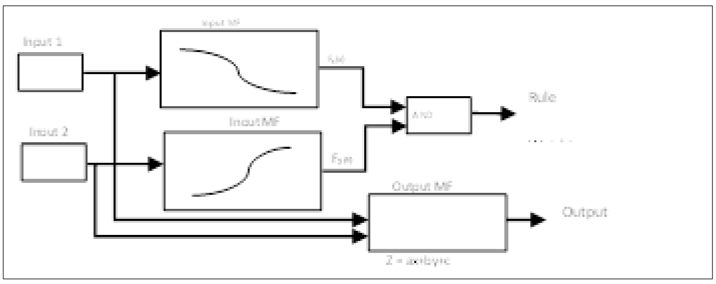

Figure 7 Deployment of Sugeno Method in Fuzzy Logic control system design

For each of the system's input and output fuzzy variables, a two-stage traffic light system has five membership functions: Zero, Small, Medium, Large, and Very-Large. Figures 8 to 10 demonstrate two input variables, Queue and Timer, and one output variable, Urgency, for the traffic urgency judgment module. Figures 11 to 13 depict two input variables, Queue-Lane-1 and Queue-Lane-2, and one Extension-Time output variable for the extension time choice module. The triangle membership function is used to construct each input and output fuzzy variable. The fuzzy logic controller's inference method is similar to the human thinking process. Artificial intelligence is related with fuzzy logic technology.

2.3.2. Design of Traffic Urgency Decision Module (TUDM)

This module computes urgency for all red phases, as seen in figures 14 and 16. The suggested smart system determines the next red-light phase to switch based on the degree of urgency. There are two input variables, Queue and Timer, and one output variable, Urgency. The input variables Queue and Timer count the number of cars in the current red-light phase and the duration of the red light since the last end of the green light. The output variable Urgency represents the traffic urgency during red light phases. In the case of Phase 1, for example, TUDM derives the urgency (U1) by analyzing the queue length and duration of red light since the conclusion of Phase 1's green light. Similarly, for phase 2, TUDM determines the urgency (U2) by processing the queue length and duration of red light since the last end of the green light of phase 2 and so on.

2.3.3. Design of Extension Time Decision Module (ETDM)

The extension time decision module (ETDM) determines the green light time, which is the extension time of the phase with the highest urgency based on the number of cars. There are two input variables, Queue-Lane-1 and Queue-Lane-2, and only one output variable, Extension-Time. The input variable Queue-Lane-1 counts the number of cars in lane 1, whereas the input variable Queue-Lane-2 counts the number of vehicles in the other side, i.e. lane 2 of the red-light phase, which has high traffic urgency. The extension time of the current green light phase is represented by the output variable Extension-Time.

As a result, in Phase 1, ETDM estimates the green light duration, i.e., extension time (E1), by analyzing the queue length of Lane EL1 of Road direction E and Lane WL1 of Road direction W. Similarly, for phase 2, ETDM calculates the extension time (E2) by analyzing the queue lengths of Lane EL2 of Road Direction E and Lane WL2 of Road Direction W, as shown in table 1. In TUDM, 25 fuzzy rules were discovered, while in ETDM, 16 rules were discovered. Some of these fuzzy rules are employed in the design of the TUDM and ETDM seen in figures 14 and 15, respectively.

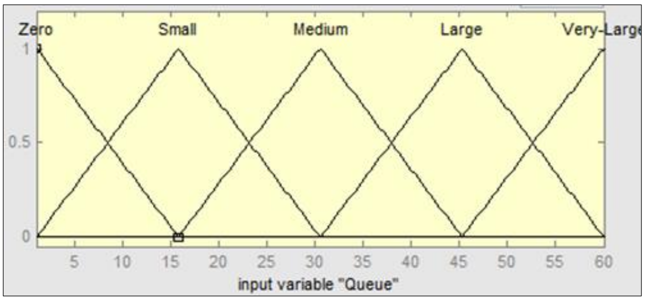

Figure 8 Queue membership

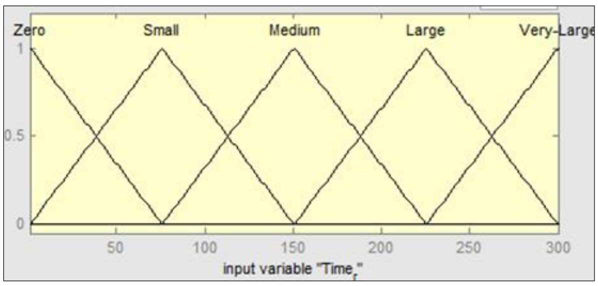

Figure 9 Time membership

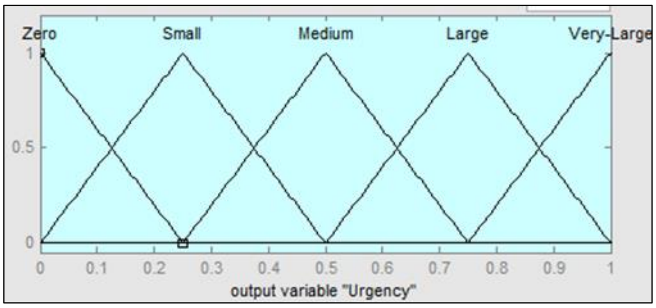

Figure 10 Urgency membership

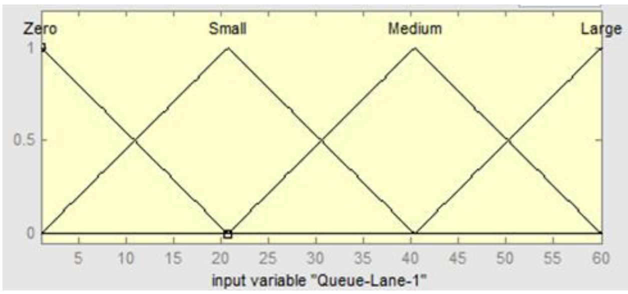

Figure 11 Queue-Lane-1 membership

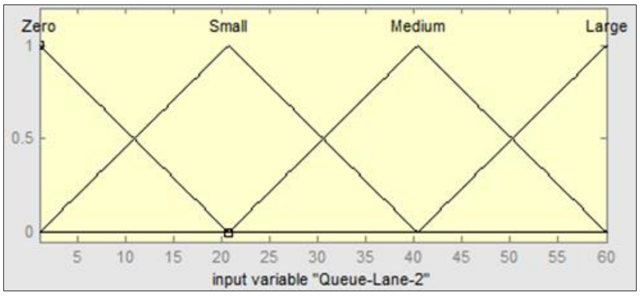

Figure 12 Queue-Lane-2 membership

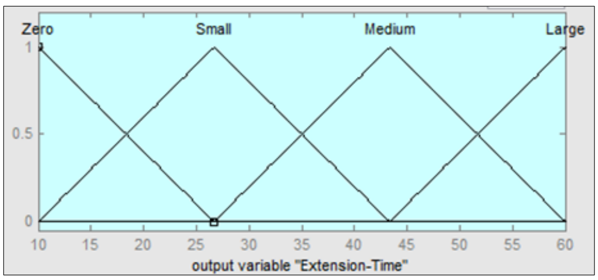

Figure 13 Extension-Time membership

1. If (Queue is Zero) and (Time_r is Zero) then (Urgency is Zero) (1)
2. If (Queue is Zero) and (Time_r is Small) then (Urgency is Small) (1)
3. If (Queue is Zero) and (Time_r is Medium) then (Urgency is Medium) (1)
4. If (Queue is Zero) and (Time_r is Large) then (Urgency is Medium) (1)
5. If (Queue is Zero) and (Time_r is Very-Large) then (Urgency is Large) (1)
6. If (Queue is Small) and (Time_r is Zero) then (Urgency is Zero) (1)
7. If (Queue is Small) and (Time_r is Small) then (Urgency is Small) (1)
8. If (Queue is Small) and (Time_r is Medium) then (Urgency is Medium) (1)
9. If (Queue is Small) and (Time_r is Large) then (Urgency is Large) (1)
10. If (Queue is Small) and (Time_r is Very-Large) then (Urgency is Large) (1)

Figure 14 Fuzzy rules for TUDM develop in MATLAB**.**

1. If (Queue-Lane-1 is Zero) and (Queue-Lane-2 is Zero) then (Extension-Time is Zero) (1)
2. If (Queue-Lane-1 is Zero) and (Queue-Lane-2 is Small) then (Extension-Time is Small) (1)
3. If (Queue-Lane-1 is Zero) and (Queue-Lane-2 is Medium) then (Extension-Time is Medium) (1)
4. If (Queue-Lane-1 is Zero) and (Queue-Lane-2 is Large) then (Extension-Time is Large) (1)
5. If (Queue-Lane-1 is Small) and (Queue-Lane-2 is Zero) then (Extension-Time is Small) (1)
6. If (Queue-Lane-1 is Small) and (Queue-Lane-2 is Small) then (Extension-Time is Small) (1)
7. If (Queue-Lane-1 is Small) and (Queue-Lane-2 is Medium) then (Extension-Time is Medium) (1)
8. If (Queue-Lane-1 is Small) and (Queue-Lane-2 is Large) then (Extension-Time is Large) (1)
9. If (Queue-Lane-1 is Medium) and (Queue-Lane-2 is Zero) then (Extension-Time is Medium) (1)
10. If (Queue-Lane-1 is Medium) and (Queue-Lane-2 is Small) then (Extension-Time is Medium) (1)

Figure 15 Fuzzy rules for ETDM develop in MATLAB

Two separate modules, namely traffic urgency decision module (TUDM) and extended time decision module (ETDM), are implemented in a two-stage traffic light system employing fuzzy logic. Figures 14–16 depict the design of a four-phase signal. Each approach in a cycle travels through green and red time periods.

The maximal implication approach is utilized in a two-stage traffic signal system based on fuzzy logic. Using this method, the final output membership function for each rule is the fuzzy set allocated to that output by clipping the degree of truth values of the linked antecedents' membership functions. After determining the membership degree of each output fuzzy variable, all of the rules that are being fired are merged, and cation is the process of transforming each aggregated fuzzy output set into a single crisp value.

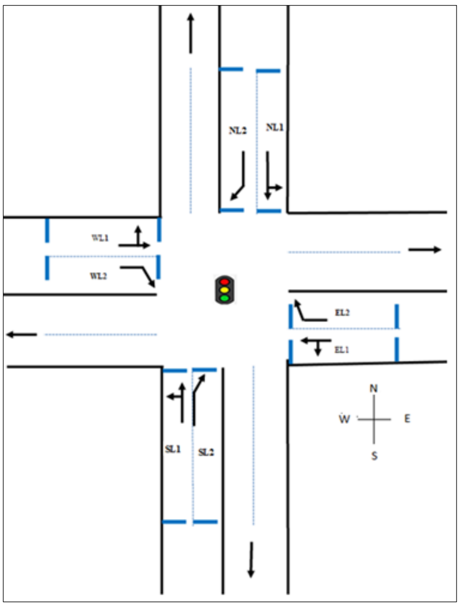

Figure 16 General Structure of intersection

2.4. Deployment of Sumo Urban Mobility Simulation (SUMO)

Sumo Urban Mobility Simulator is used to simulate road vehicles. To determine the best routing in the simulation environment, SUMO employs three built-in algorithms. The Owerri city map's fire service junction was taken from the internet (openstreetmap.org) and renamed map.osm. The entire simulation is run on a real-world map of Owerri. SUMO translates Owerri city map data to XML format in order to provide the best route and traffic system. To convert map.osm files to map.net.xml files, DOS command scripts such as start-command-line.bat and netconvert are utilized. SUMO Neteditor is used to add additional essential details such as road name, traffic signal time, and so on. The route for vehicles is written in map.rou.xml file. The flow chart of the full simulation and the SUMO GUI are presented in figures 17 and 18 respectively.

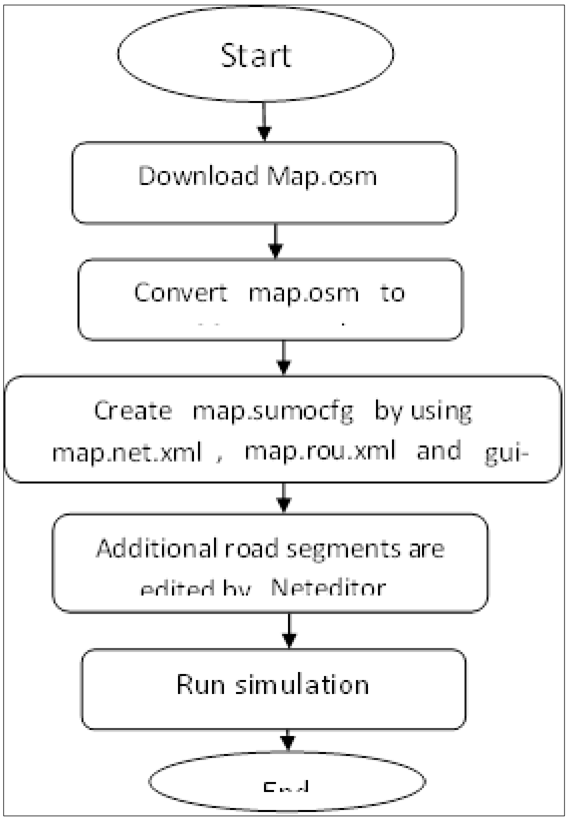

Figure 17 Operational flow chart of the sumo process

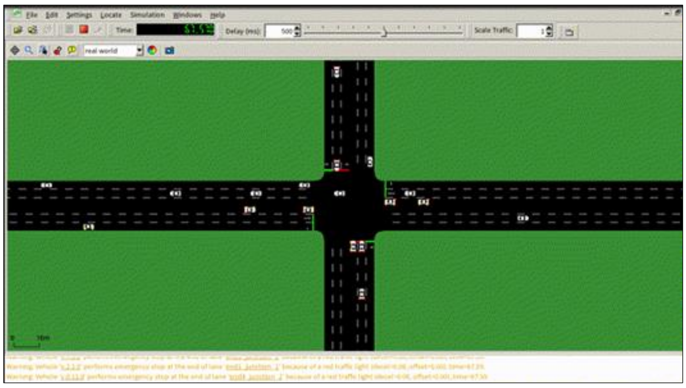

Figure 18 Real-time multiple intersections used to analyze the various algorithms using SUMO's graphical user interface

3. Results and Discussion

Following the completion of the design of a two-stage traffic light controller system, the entire system is tested, and the effects of the input variables on the output variables are noticed. Figure 19 depicts the impacts of multiple inputs, namely Queue, Timer, Queue-Lane-1, and Queue-Lane-2, on the output variables, namely Urgency and Extension-Time, using simulation. When the values of Queue and Time on the x- and y-axes are tiny, the value of Urgency on the z-axis is small. When the queue side is too large and the timer density is low, the value of urgency rises quickly to reach a maximum value. When the queue side is smaller and the timer density is too high, the value of urgency rises faster to reach a maximum value.

A model that mimics a genuine environment may be developed using SUMO by building a road network, adding cars to it, and setting rules for how these vehicles traverse the network. Because the approach is stable for modeling a realistic traffic situation, the article uses the SUMO software to execute its simulations. Because there is no physical implementation, the SUMO software saves time.

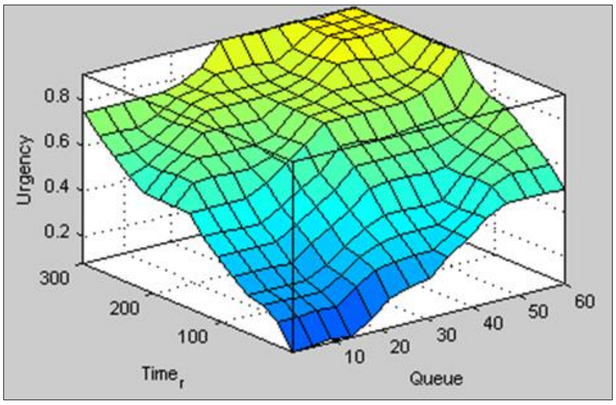

Figure 19 Input variables Queue, Timer Vs output variable Urgency using centroid defuzzification method

A four-way junction with two lanes in each direction is generated by the various intersections. The leftmost lane of each road approaching the intersection is used for traffic going left, while the rightmost lane is utilized for vehicles moving ahead or to the right. A traffic light is used to ensure that traffic from the intersecting roads does not enter the intersection at the same time. The design configuration for traffic lights is divided into four phases, which are as follows:

- NS-G - the north and south roads are green and the others are red.
- WE-G - the west and east roads are green and the others are red.
- WE-Y - the west and east roads are yellow and the others are red.
- NS-Y - the north and south roads are yellow and the others are red.

Figures 20 and 21 show the graphical representations of the crossings. In order to dynamically modify the condition of a traffic signal, an SUMO simulation environment powered by an artificial intelligence written using Python is presented. Python provides access to the "Traci" library, which may be used dynamically. Before the simulation begins, a feasibility test is performed to offer the optimal red light / green light result ratio calculation in order to insert automobiles one by one in each lane. The accepted standard traffic signal timings for default red and green lights range from 2 to 20 minutes. According to the simulation results, a total of 90 and 270 cars are recorded in vertical and horizontal roads for both the fixed traffic light controller and the proposed fuzzy logic traffic light controller, as shown in tables 2 to 4, and the corresponding graphs of simulation time piloted against number of cars are shown in figures 23 to 24. The entire simulation duration for the fixed traffic light controller and the suggested fuzzy logic traffic light controller was 1,426 and 1,328 seconds, respectively. The current approach to traffic signals is based on predetermined cycles or human control.

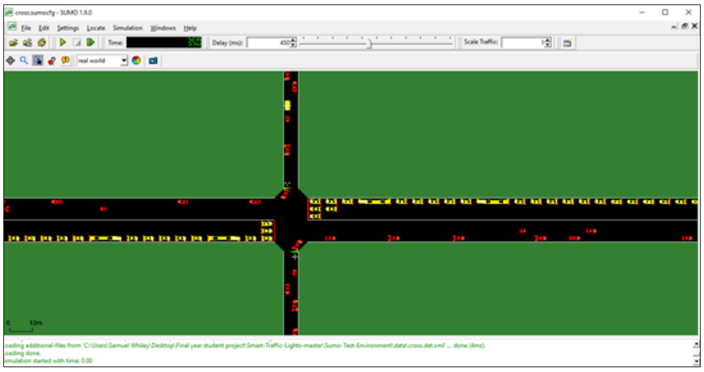

Figure 20 Sumo Urban Mobility Simulation (SUMO) GUI-01

Figure 21 Sumo Urban Mobility Simulation (SUMO) GUI-02

Table 2 Fixed traffic light simulation for cars in vertical and horizontal roads

Number of cars in vertical road	Number of cars in horizontal road	Simulation time(seconds)
30	60	428
30	90	492
30	120	506
Total = 90 Cars	Total = 270 Cars	Total = 1,426 Seconds

Table 3 Fuzzy logic traffic light simulation for cars in vertical and horizontal roads

Number of cars in vertical road	Number of cars in horizontal road	Simulation time(seconds)
30	60	351
30	90	481
30	120	496
Total = 90 Cars	Total = 270 Cars	Total = 1,328 Seconds

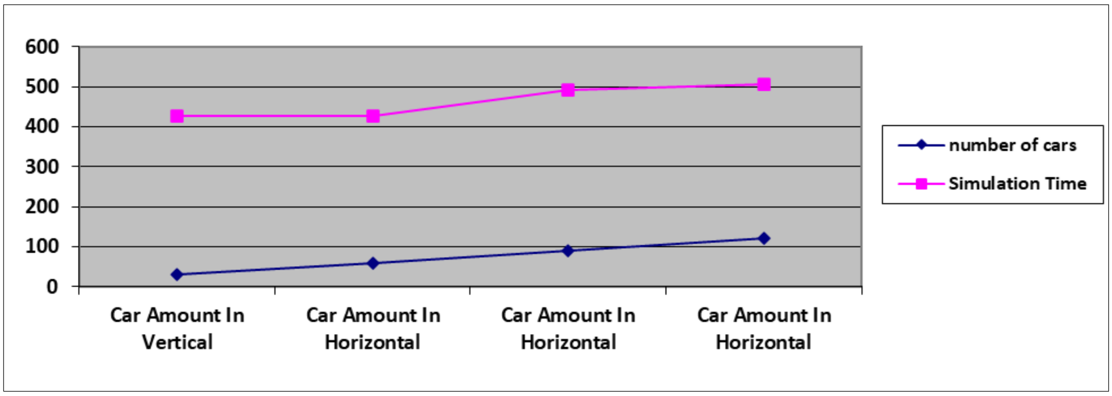

Figure 22 Standard traffic light's chart for a fixed traffic light controller

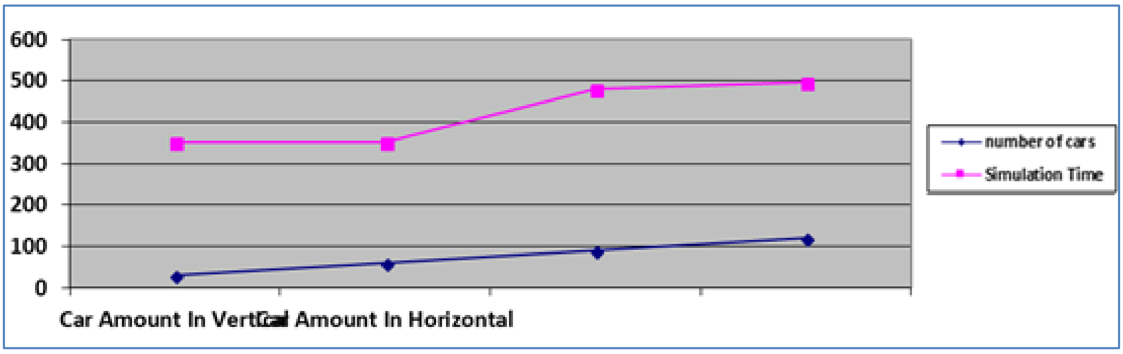

Figure 23 Proposed chart for fuzzy logic traffic light controller

Table 4 Comparison of different simulation time for fixed traffic light controller and fuzzy logic traffic light controller

Simulation time for fixed traffic controller(seconds)	Simulation time for fuzzy logic controller(seconds)
428	351
492	481
506	496
Total = 1,426 Seconds	Total = 1,328 Seconds

Table 5 Simulation results for fixed time controller and fuzzy logic time controller

Simulation parameters	Available controller	Proposed controller
Technique	Fixed time controller	Fuzzy logic time controller
Simulation time(seconds)	578	578
Number of vehicles served	240	240
Average time saved(seconds)	78	155

The suggested fuzzy logic traffic controller has a faster simulation time of 98 seconds than the fixed traffic light controller. As a result, the suggested fuzzy logic traffic light controller delivers improved performance in terms of moving time and reduced waiting time, resulting in significant reductions in fuel consumption, air and noise pollution, and traffic congestion on Owerri city highways. In terms of 578 seconds simulation time and 240 vehicles served, the suggested fuzzy logic traffic light controller recorded a noticeable average saving time of 155 seconds compared to the fixed time controller's unsatisfactory average saving time of 78 seconds, as shown in table 5.

4. Conclusion

The developed fuzzy logic traffic light controller system can estimate traffic density using ultrasonic sensors positioned on each side of the road to increase the time delay for the green signal and eliminate needless waiting time. The simulation results demonstrated that the system with fuzzy logic control Sugeno approach appropriately and efficiently addressed the traffic difficulties at many crossings. In terms of performance, the developed fuzzy logic traffic controller system and fixed time control provide distinct outcomes. The suggested approach outperforms the fixed time-based traffic light control system in terms of the number of lineups and departures. Because of its flexibility, which includes the number of cars sensed at the approaching junction and the extension of the green duration, the fuzzy logic traffic signals controller outperformed the fixed time controller. Because the fixed time controller is an open loop system, the green time is not prolonged when there is a large density of automobiles at numerous intersections. The extension time in the smart fuzzy logic traffic light control system is not a set value; rather, the fuzzy system can adapt to the provided input. To the green time, the fuzzy system supplies output variables such as zero, short, medium, and big. The number of cars detected at the fuzzy controller's input is converted into fuzzy values such as very short, short, medium, large, and very large. Less waiting time reduces not just fuel consumption but also air and noise pollution. It also demonstrates that it may alleviate traffic congestion and save time spent by a green signal on an empty route.

Compliance with ethical standards

Acknowledgments

We will like to acknowledge a senior member of the Department, Prof F. K Opara, who gave us some insight on how to go in bringing this work to life. Also, appreciate all members of staff for their salient contribution.

Disclosure of conflict of interest

We the authors remain sole contributors to this work, without any form of sponsorship from a third party; hereby declare that there is no conflict of interest to the best of our knowledge.

References

[1] Amanjot Kaur, Mohita Garag and Harpreet Kaur, Review of Traffic Management Control Techniques, volume 7, Issue 4, April 2017.

[2] Beattie, A., What is the Difference Between Artificial Intelligence and Neural Networks? Technopedia. Retrieved from http://www-technopedia.com/2/27888/programming/what-isthe-difference-between-artificial-intelligence-and-neural-networks, June 2017.

[3] Chandana K. K., Meenakshi Sundaram S., Cyana D'sa, Meghana N.S., Wamy, Navya K., Smart Traffic Management System for Congestion Control and Warnings using Internet of Things (IoT), Saudi Journal of Engineering Technology, volume 2, Issue 5, pp. 192-196, May, 2018.

[4] Cunha F., Villas L., Boukerche A., et al., Data Communication in Vanets: Protocols, Applications and Challenges, Ad Hoc Network, pp. 44:90-103, 2016.

[5] Monika G., Kalpana N., Gnanasundari P., An Intelligent Automatic Traffic Light Controller Using Embedded System, volume 4, Special Issue 4, April 2017.

[6] Md. Rokebul Islam, Nafis Ibn Shahid, Dewan Tanzim ul Karim, Abdullah A. Mamun, Md. Khalilur Rhaman, An Efficient Algorithm for Detecting Traffic Congestion and a Framework for Smart Traffic Control System, JAM. 31 February 3, 2017.

[7] Ms.Shaili Shinde, Sheetal Jagtap, Intelligent Traffic Management System: A Review, International Journal for Innovative Research in Science & Technology(IJIRST), volume 2, Issue 09, February 2016.

[8] Parno B. and Perrig A., Challenges in Securing Vehicular Networks, Workshop on Hot Topics in Networks (HotNets-IV), College Park, MD, Association for Computing Machinery, Inc., pp. 1-6, November 2005.

[9] Qian Y. and Moayeri N., Design of Secure and Application-Oriented Vanets, VTC Spring, IEEE Vehicular Technology Conference, Singapore, New York: IEEE, pp.794–2799,11–14 May 2008.

[10] Rakha H. and Kamalanathsharma R., Eco-Driving at Signalized Intersections UsingV2i Communication, 14th International IEEE Conference on Intelligent Transportation Systems(ITSC), Washington, DC, pp.341–346, New York: IEEE, 5–7 October 2011.

[11] Keerthi P. and Hariharagopalan, A Survey on Various Traffic Management Schemes for Traffic Clearance, Stolen Vehicle and Emergency Vehicle, International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE), ISSN: 0976-1353, Volume 19, Issue 2, January 2018.

[12] Tousif Osman, Shahreen Shahjahan Psyche, Shafi Ferdous J. M., Hasan Zaman, Intelligent Traffic Management System for Cross Section of Roads using Computer Vision, 2017.

[13] Vishakha Thakare S., Snehal Jadhav R., Sananaj Sayyed G., Poonam Pawar, Design of Smart Traffic Light Controller using Embedded System, Volume 10, Issue 1, Mar. Apr. 2018.

[14] Yan G., Wen D., Olariu S., et al., Security Challenges in Vehicular Cloud Computing, IEEE T. Intell Transp, 14(1): 284–294, 2013.

[15] Abdul Kareem E.I. and Jantan, A., An Intelligent Traffic Light Monitor System using an Adaptive Associative Memory, International Journal of Information Processing and Management, 2(2): 23-39, 2011.

[16] Findler, N. V., Sudeep S., Ziya, M. and Serban, C., Distributed Intelligent Control of Street and Highway Ramp Traffic Signals, Engineering Applications of Artificial Intelligence, 10(3):281- 292, 1997.

[17] GiYoung, L., Kang J. and Hong Y., The Optimization of Traffic Signal light using Artificial Intelligence, Proceedings of the 10th IEEE International Conference on Fuzzy Systems, 2001.

[18] Wen, W., A Dynamic and Automatic Traffic Light Control Expert System for Solving the Road Congestion Problem, Expert Systems with Applications, 34(4):2370-2381, 2008.

[19] Tan K., Khalid M., and Yusof R., Intelligent Traffic Lights Control by Fuzzy Logic, Malaysian Journal of Computer Science, 9(2): 29-35, 1996.

[20] Abdel Nasser H Zaied, Woroud Al Athman "Development of a fuzzy Logic traffic system for isolated signalized intersections in the state of Kuwait", Expert Systems with applications 38(8), 9434-9441, 2011, https://doi.org/10.1016/j.eswa.2011.01.130

[21] Sadia jafari, Zainab Shahbazi and Yung-choel Byun "Traffic Control prediction design based on fuzzy logic and lyapunon approaches to improve the performance of road intersections", Process 2021, 9(12) 2205. Advances in lightweight AI for internet of things devices for smart cities. https://doi.org/10.3390/pr9122205

[22] Fathy, M. and Siyal, M. Y., Real-Time Image Processing Approach to Measure Traffic Queue Parameters, Vision, Image and Signal Processing, IEEE Proceedings, 142(5):297-303, 1995.

[23] T. Ghasempour, G. L. Nicholson, D. Kirkwood, T. Fuiyama and B. Heydecicer, "Distributed Approximate Dynamic Control for Traffic Management of Busy Railway Networks", in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no.9, pp.3788-3798, sept. 2020, https://doi.org/10.1109/TITS.2019.2934083

[24] P. Grandinetti, C. Canudas-De-Wit and F. Garin, "Distributed Optimal Traffic Lights Design for Large-Scale Urban Networks", in IEEE Transactions on Control Systems Technology, vol.27, no 3, pp.950-963, May 2019, https://doi.org/10.1109/TCST.2018.2807792

[25] Huang, Q. and Miller, R. (2004). Reliable Wireless Traffic Signal Protocols for Smart Intersections, Downloaded August 2011 from http://www2.parc.com/spl/members/qhuang/papers/tlights_itsa.pdf ,2004.

[26] Di Febbraro A., Giglio, D. and Sacco, N., Urban Traffic Control Structure based on Hybrid Petri Networks Intelligent Transportation Systems, IEEE Transactions, 5(4):224-237, 2004.

[27] Zamith, M., Leal-Toledo, R.C.P., Clua, E. (2012). "A Novel Cellular Automatic Model for Traffic Freeway Simulation". In Sirakoulis, G. C., Bandini, S.(eds) Cellular Automata. ACRI 2012. Lecture Notes in Computer Science, vol.7495. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33350-7_54 Print ISBN: 978-3-642-33349-1, online ISBN: 978-3-642-33350-7

[28] M. S. Arifin, S. A. Razi, A. Haque, N. Mohammad "A Microcontroller Based Intelligent Traffic Control System", American Journal of Embedded Systems and Applications 7(1), 21-25, 2019, pp.21-25, doi: 10.11648/j.aesa.20190701.13