KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

DEPARTMENT OF ELECTRICALAND ELECTRONIC ENGINEERING

INTELLIGENT TRAFFIC MANAGEMENT FOR THE KUMASI METROPOLIS

By

Frederick Egyin Appiah-Twum

PG5943616

NOVEMBER 2018

INTELLIGENT TRAFFIC MANAGEMENT FOR THE KUMASI METROPOLIS

By

Frederick Egyin Appiah-Twum

(B.Sc. Telecommunications Engineering)

A thesis submitted to the Department of Electrical and Electronic Engineering, Kwame Nkrumah University of Science and Technology, Kumasi in partial fulfilment of the requirements for the award of the degree of

MASTER OF PHILOSOPHY IN TELECOMMUNICATION ENGINEERING

November 2018

Declaration of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief. It contains no materials previously published or written by another person or material which to a substantial extent has been adopted for the award of any other degree or diploma at the Kwame Nkrumah University of Science and Technology, Kumasi or any other educational institution, except where due acknowledgement has been made in the thesis.

………………………… ………………… …………………

Name of Student and ID Signature Date

Certified by:

………………………… …………………… …………………

Name of Supervisor Signature Date

Certified by:

………………………… …………………… …………………

Name of Head of Department Signature Date

Abstract

The problem of vehicular traffic congestion is universal yet there has not been a long-term permanent solution to this problem, and it is increasingly worsening by the day all around the world with severe vehicular traffic taking its toll on all road users. With the upsurge in urban traffic jams, innovative control strategies are therefore essential to allow efficient flow of vehicular movement. It is thus not surprising that a myriad of novel control strategies has been developed over the past years in an attempt to manage the ever-growing urban gridlock. Many of the currently used traffic control strategies are based on the relatively inefficient fixed-time traffic systems, like in the case of Ghana, or on a central traffic-responsive control system, which is challenging to implement and even much more difficult to maintain. As a consequence of inefficiencies in traffic control, road users are saddled with regular and inconveniently long waiting times in queues. To mitigate this problem, a distributed artificial intelligence and multi-agent system is proposed as a viable approach to manage the traffic menace. The proposed system uses historical data for traffic management and was designed and implemented using Simulation of Urban Mobility (SUMO) software. Iterative learning control which is a technique for refining the momentary response performance of a system that functions repetitively over a fixed period of time is used to tune the phase splits of the traffic controller to obtain the optimal controller duty cycles with the least delay, resulting in frequent traffic flows, minimum waiting times and shorter queued vehicles. The result obtained in the comparison of the current fixed time-controlled system and designed system clearly indicated that the proposed system outperformed the fixed-time cycle controllers in every key performance index selected for evaluation.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ALU	Arithmetic Logic Unit
CA	Cellular Automata
CMAS	Cognitive Multi-Agent System
COBOL	Common Business-Oriented Language
CPS	Central Processing Station
CU	Control Unit

CV	Coefficient of Variation
DTP	Daily Traffic Pattern
DTSS	Daily Traffic Signal Schedules
FCTL	Fixed Controlled Traffic Light
FL	Fuzzy Logic
FLC	Fuzzy Logic Controller
FORTRAN	Formula Translation
HCM	Highway Capacity Manual
ITS	Iterative Strategy
MAS	Multi-Agent System
MCA	Multi-Connect Architecture
PELICON	Pedestrian Light Controlled Crossing
RL	Reinforcement learning
SARSA	State-Action, Reward-State Action
SUMO	Simulation of Urban Mobility
TLC	Traffic Light Controller
UI	User Interface

Chapter 1
INTRODUCTION

1.1. General Introduction

The complications associated with the upsurge of inner-city congested roads requires that, imaginative control techniques which allow for vehicles to move all the more openly as possible, for example, in the coordination of traffic signals. Most of the time, these procedures are focussed on a focal traffic responsive control framework, and with the complexities associated in implementing and preserving these frameworks become even more complex taking into account traffic control elements such as traffic lights, detectors, and specific hardware. Another constraint is the nonexistence of interoperability among these components, particularly if obtained from various makers. Therefore, localization is favoured. All things considered, achieving spatial decentralization has always been the first objective of traffic light design. [1].

Decentralized frameworks in traffic ensure that traffic control, is done by means of some specific communication based, centrally controlled organization process. This communication is unacceptable in the field of traffic signal management due to interoperability limitations. Real time restrictions imply that agents cannot manage complex, because of the need to respond instantly to real time traffic situations. Additionally, communication channels might limit productive communication, hence interoperability constraints are faced by the manufacturers of these various hardware. Despite the push to have a standard interoperability and a working convention for the productive management of traffic, in reality the currently accessible equipment does not give such conveniences.

Transportation by road has become most common and widely used form of transport throughout the country, and possibly the world at large. With the constant upsurge in the number of vehicles annually and an equal increase in the number of road users, there is the utmost importance to maintain order on our roads. This is the reason traffic lights have become very crucial for road traffic management and safety for all users.

Pollution, congestion, security, parking, and many significant issues are the factors arising from vehicular traffic congestion and are present every day in all key cities around the globe. Researchers often have the tendency to recommend that the precise staging of traffic lights in urban cities could help to mitigate these problems by enhancing the flow of vehicles through these cities. [2] Traffic management procedures are primarily motivated by enhancing the safety and flow of traffic in our cities.

The way to enhance traffic flow and safety of the current transport system is to introduce automation and intelligence to roadside infrastructure [3]. Intelligent control of traffic has become a very pertinent issue for road traffic management. The issues that make the present framework incapable of proper traffic control essentially are the overwhelming traffic jams with the rising number of vehicles on streets, and substantial traffic congestion has considerably increased in significant urban areas. This happens typically at major intersections in the morning, just before office hours and in the evening, after office hours. The fundamental impact of this issue is the unnecessary wasting of road users' time.

Another issue is the fact that road users still need to wait even where there is no conflicting traffic, because the traffic light remains red for the pre-set periods which turns to restrict the regular movement of cars. Fast and safe transportation systems and rapid regular transit systems are crucial to the economic development of this nation. The transportation of goods, industrial products, and machinery are the prime factors, which will inspire the industrial development of any country. Long waiting times, loss of fuel and money, are results of traffic congestion and the continuous use of improper technology. It is therefore very necessary to have a fast, economical, efficient and effective traffic control system to manage traffic in our cities.

1.2. Benefits of coordinated traffic management

The benefits of having coordinated intelligent distributed traffic management and monitoring systems are as follows

- Enhanced safety for all road users.
- Elimination of congestion that results from the frequent stop and wait style of traffic management. The constant stopping and waiting increases road user trip times.
- Reduction in the emission of vehicular pollutants.
- Decreased costs to vehicle users since the amount of fuel used is vastly reduced.
- Better coordinated traffic signals.

1.3.Problem Statement.

Due to the continually increasing demand for transportation services, vehicles have become an indispensable part of our daily lives, but regrettably the very limited road resources and infrastructure during periods such as morning or evenings where traffic densities are high. Ensures that there is an enormous strain on the road resource available. Traffic management systems give us the prospect to be able to deal with commuters and road users in an orderly manner. It is imperative to understand that traffic management is complex and very difficult to manage. Traffic flows in major cities are constantly changing due to various factors such as the time of the day, the period of the week and year. Roadworks, construction, and accidents further complicate traffic management and control.

The current traffic management system in place is done with fixed time cycle controllers. In practice, fixed cycle controllers are unable to cope with most traffic in Ghana. The inability of the current traffic management system in the Kumasi Metropolis to deal with the large volume of traffic and its irregular nature on the limited road resource effectively and efficiently, results in prolonged queuing and congestion during peak hours. This resultant sustained queuing at traffic intersection points are due to the constraints in the current time-based design, primarily the failure of the current time-based design to become accustomed to changing traffic densities in the Kumasi Metropolitan Area.

1.4. Objectives of the research

1.4.1. General Objectives

The prime objective of the thesis is to design a distributed intelligent traffic management system adept at changing itself to the demands of the current road user demands, while alleviating the effects of the fixed cycled controllers in the metropolis, without necessarily making any major changes to the limited road resource or infrastructure in place.

1.4.2. Specific Objectives

The following are the specific objectives set in order to achieve the general objective:

- To design an intelligent distributed traffic management system that learns based on historical data to optimize the duty cycles of traffic light controllers.

- To simulate the performance of the proposed system.
- To analyse and compare the efficiency of the proposed system to the current system

1.5.Justification of the research

The efficient and effective coordination of systems where there is a deficiency in resources has been a fundamental issue of a number of works. In the traffic domain, the management of traffic systems has been characteristically attempted in various manners. Keeping with the current research trends and from attempted works in literature, there are still significant issues in the area of using historical data in active traffic management in a fully distributed traffic management system, with no lasting solution yet to address this issue.

The main purpose here is to show that historical data can be used to effortlessly manage traffic with very little or no human intervention. Introducing a traffic learning mechanism in the traffic controller to reduce congestion on the limited road resource network is possible, though it may take a longer time to achieve it since agents which are necessary for this intelligent management need time to observe other agents, build models of them, and learn their behaviours.

However, realizing this intelligent management is not trivial. Therefore, the initial inspiration for this work is to adopt mechanisms for the emergence of cooperation among the participants of a system, especially when they have their own goals and are not necessarily cooperative. The research is to design and test efficiency of a distributed traffic management system, that uses historical data to analyse traffic patterns, learn the behaviour of these traffic patterns in different scenarios and adapt their duty cycles accordingly to accommodate all scenarios in traffic management.

1.6.Research questions and hypothesis

<u>Research Questions</u>

The extensive review of literature on the various types of fixed traffic management has identified the following research question:

- Can the system identify and interact with road network elements, to achieve active management and control in enhancing traffic flow?
- Can the system learn various traffic patterns and optimize the duty cycles of traffic controllers accordingly, to accommodate all traffic scenarios?
- Can historical data be used to optimize traffic efficiently in a fully distributed network?

Research Hypothesis

Based on reviewed literature the formulated hypothesis is:

The proposed system is a coordinated distributed intelligent traffic management system that identifies and interacts with road network elements, to learn various traffic patterns and optimize the duty cycles of traffic controllers accordingly to accommodate all traffic scenarios, while significantly enhancing the flow of traffic.

1.7.Scope of the research

The project is limited to how the distributed and coordinated system uses traffic data to learn and optimize the duty cycles of traffic controllers in a traffic simulator.

Chapter 2

Literature review

2.1. Introduction

The issue traffic management systems on various roads in all parts of the world try to efficiently manage the road resource and control traffic in an orderly and civil manner. This ensures that the roads which are a limited resource are used as efficiently as possible. The normal purpose of traffic lights in traffic management entails more than coordination to guarantee that vehicles and pedestrians move as effortlessly, and securely as possible [4]. A range of diverse systems are used to realize effective traffic management, these systems may vary from simple clockwork mechanisms to much more sophisticated computerized control and organizational systems [5].

Based on the density of traffic, systems are used to govern traffic where there are either individual intersections, arterials, and a grid or network. Dependent on the control strategy, fixed time and traffic-responsive signal control strategies can be relied on. This implies different alternatives can be highlighted [2]. In order to manage the traffic on roads efficiently all solutions should take into consideration the modelling, prediction and control of traffic.

2.2. Modelling traffic

The traffic simulator has become a useful apparatus for measuring traffic densities virtually. The two techniques commonly used to model traffic are elaborated. Gas-kinetic representations that require equations concerning traffic concentrations to velocity and these are known as Macroscopic traffic. These equations are augmented with terms for accumulation and reduction of pressure to account for spectacles like stop-and-go traffic and unplanned congestion. Although macroscopic models can be tuned to simulate certain driver behaviours, they do not provide a direct, flexible, way of modelling and optimizing them, making them less suited for the research. Conditional to these guidelines, many behaviours arise with the interaction of vehicles. A unique detailed way of simulating all the basic driving guidelines of cars is by using cellular automata (CA). [7] CA requires distinct cells that are detailed. A roadcell may include a car or may be empty. Local transition instructions govern the dynamics of the system. Vehicles intensify their speeds by certain quantities until they arrive at their maximum velocities at each distinct time interval. To avoid accidents, the incidence of slower moving vehicle ahead, will be decreased to avoid collision. Arbitrariness is obtained by including for particular vehicles, a slight chance of decelerating. The Cognitive Multi-Agent System approach is an innovative method for vehicular simulation and optimization, whereby various agents interact. [8] Additionally, using varied multi-agent systems ensure that diverse elements can have different sensors, goals, behaviours and learning capacities consequently, permitting the investigation of a vast range of microscopic traffic models.

2.3. Predicting Traffic

The capacity to forecast traffic situations is significantly aimed at optimum regulation, where it is known that a particular road will be congested during a period under present circumstances, data can be communicated to pedestrians and vehicle owners to avoid the route, thus, relieving the entire network from congestion.

Additionally, if the significance of different driving policies can be accurately projected, an ideal conclusion can be achieved by relating the projected outcomes. The easiest procedure for vehicular forecasting at an intersection is by gauging traffic over periods of time, and presuming circumstances are constant for that time interval. The real-world situation is designed by means of detection devices preinstalled.

2.4. Traffic control

The resulting data from modelling and predicting traffic becomes very crucial to traffic control since it becomes the input in the design of traffic lights. Traffic lights are control devices at the intersections of routes, zebra crossings, and other settings to regulate the stream of traffic [9]. Traffic lights toggle the right of way rendered to road users by showing lights of a typical colour, which, are red, amber or yellow, and green, that are standardized for universal use. In the usual order of colour phases:

- Green phases permit vehicles to advance on the routes desired when it is safe and advance dependent on the traffic situation on the opposite side of the intersection.

- The amber or yellow light cautions that the traffic controller is about to turn red. A segment during which red and yellow are exhibited concurrently specifies that the signal is about to turn green. [10] Actions mandatory for motorists on a yellow light differ, there are instructions demanding motorists to stop if it is safe to do so, and others allowing motorists to head towards the intersection.

- A flashing amber signal is a warning. A blinking amber is deployed only at PELICON crossings [11] which, is category of pedestrian crossing, which comprises a set of poles each with a regular set of traffic lights fronting approaching traffic, a push button and two illuminated, coloured pictograms facing the pedestrian across the road. In place of the joint red–amber signal, which signifies that drivers may proceed if no people are on the pedestrian crossing
- The red signal prohibits all traffic from going on.

.

- A flashing red phase is regarded as a stop sign.

The evolution of traffic light controllers since its invention has contributed exponentially to the improvement and advancement of traffic over the years. With these improvements come better and even more sophisticated traffic controllers and management policies to cope with the various densities and distribution of vehicles. Computers have also enhanced the management of traffic, as they have the ability to monitor traffic and change lights accordingly, with the aid of traffic prediction software, computers can accurately predict and accordingly control the traffic of a city [12].

2.5. Traffic systems.

There are various theoretical developments and designs in literature that people have proposed or constructed to for the purpose of traffic control and management. These designs range from singular simple clockwork mechanisms to sophisticated computerized control and coordination systems. These systems have been proposed by individuals and organizations in an attempt to mitigate the traffic congestion situation that is worsening by the day.

2.5.1. Fixed cycled traffic light controllers

Fundamentally, these consist of a traffic controller, a traffic light head, and detectors. For fixed time controllers, the traffic display is in the green state for exactly the same period for every cycle [13]. This is irrespective of traffic conditions. In fixed time traffic control, a stage is the series of non-conflicting phases, which work together, this means the stage, begins when the last of the phases end. The phase is then shown to a specific traffic link. At a junction, each phase is an electrical circuit from the controller that feeds one or many signal heads. In Ghana, fixed time cycled controllers control all traffic lights, where a particular cycle is defined where all sides get a green signal at a point [14].

Split phase times evaluate how long a light will be in a particular state. These fixed cycles have to be adjusted to specific times of the day, month or year, and to certain situations such as road works or accidents that may include vehicular collisions. This tends to complicate traffic management using fixed time cycled controllers. The major complication with fixed time controllers is the fact that green cycles are static regardless of the traffic conditions present at traffic intersections. This makes them very ineffective in the proper management of the flow of traffic in heavily congested areas [15]. In lightly trafficked areas such as back roads and side roads, it becomes very wasteful because there are some cycles there are no waiting or oncoming vehicles. These cycles would be better used for busier approaches. Trying to optimize the controllers is complex, and for optimal performance, the controllers need to be configured regularly with the changing traffic conditions.

Modern trends in fixed time traffic management consider the control of networks of signalized intersections. At the commencement of each cycle, a controller chooses the duration of every stage at the beginning of each cycle that is a function of all queues in the network. Nonconflicting phases are set where vehicles may move at known saturation rates. Demand is modelled by vehicles moving towards and leaving the network at constant typical rates, with the assumed burst sizes and movement with average turn ratios. The motion of the vehicles is modelled as store and forward queuing network. The fixed traffic controllers have three main functions. First to stabilize a demand for bounded queues. Maximum queues require only local information; phases and cycles which are selected based on queues adjacent to the intersection. The second function uses maximum queues to stabilize demand where required. Finally, maximum queues need no prerequisite knowledge of demand though it relies heavily on turn ratios. Fixed time controllers guarantee limits on queue size, delays, and queue clearance periods [16].

Allen J Miller in his paper derived a mathematical relation for finding the typical delay of vehicles at a junction, measured by fixed controllers. The relation he found was comparable to the delays observed in practice. This relation was then used by Miller to optimize the fixed controllers based on the average delay formula. Miller defined delay as the transformation between the delayed and the un-delayed trip times, and this includes the deceleration and accelerated distances on each side of a junction. He considered that a vehicle keeps steady speed until it gets to a stop line where, it decelerates instantly, instead of one that slowed down gradually at the far end of the queue. He also assumed that at the green phase, vehicles accelerate instantaneously to its final speed, which he could model mathematically. To make sure the delays were constant the first vehicle would wait at the stop line until a little after the beginning of the green phase to allow for acceleration delays in practice. This incursion in the green phase was the lost time. The remainder of the active green time phase and the amber phase that followed after was the effective green phase [17]. This amber phase was added because vehicles may or may not stop depending on the distance to the signal and the speed of the vehicle. Miller's model essentially had to be as realistic as possible since:

- The distribution of the number of vehicular arrivals was in the period of the order of about half a minute.
- Vehicular distributions were only allowed to pass when the traffic signals were green.

Though the features were general, for the expression of delay to be obtained, the above conditions were satisfied. The significance, however, was to derive a practical relation with a good approximation to reality [18].

Igbinosun and Omosigho [24] highlight, the inadequacies of prior time-dependent models, which comprise the hypothesis that arrivals and departures are of an explicit distribution and have fixed rates. Secondly, the original queue length is presumed zero at the beginning of the assessment period. They propose a model with the arrivals considered Markovian and individualistically distributed with respect to the Poisson arrival rate. Where there are numerous lanes regulated with a specific Traffic Light Controller, they adopt that all the lanes are equal. Hence, a single server case with no overtaking. They base their research solely on the basis that for the duration of the green stage, sk (k = 1, 2, 3…... m) vehicles can be served, but no queue is permitted before the segment. The early queued dimensions may or may not be zero at the launch of the green phase. A maximum quantity of automobiles (Qmax) is permitted at any given green signal phase without hindering other intersections. They additionally accept that the controller cycle ranges from red to green, the yellow phase is presumed to be either allocated to the red or green phase. The cars exiting the green phase is not a constant, but that of the green phase is constant and that the traffic light controller is a hundred percent effective [19].

Fixed time cycled controllers to an extent are able to manage traffic, but during off peak hours, it does a fair job but failed inevitably at managing traffic during peak periods.

2.5.2. Expert systems

Knowledge-based expert systems are computer systems with the ability to emulate the decision making the ability of the human expert, these systems are designed to solve complex problems by reasoning through bodies of knowledge, constructed principally as if-then rules in place of pseudo codes. The purpose of the knowledge-based expert system is the emulation of social problems and then providing human-solving activities in complex real-world tasks. The ability of these systems and their possible applications to transport have generated considerable interest inside the transport-engineering field, due to the aggregated transport growth, that results in traffic congestion, safety concerns, and environmental pollution. The use of the knowledgebased expert system is likely to be important to the transport sector. [23]

In the areas of urban infrastructural design, [23] transport planning, safety and maintenance, vehicle scheduling, traffic monitoring and control especially in urban cities traffic management is crucial. However, with the efficiency and practicality of these systems, there are still some challenges [22]. The advantages Knowledge-based expert systems have over fixed time cycled controllers include the fact that, the order of implementation is not definite prior to resolving the problem, the separation of knowledge and control parameters and the establishment of data provides probable solutions, can function well with incomplete data and the incremental evolution capability. The knowledge-based expert system is distributed into three fundamental modules; the knowledge base that is the state memory, the inference engine that is the control strategies and short-term or dynamic memory called context. The other three supplementary components, which are important in emerging usable and extensively accepted systems, are the knowledge acquisition module, the explanation module and the user interface [25] [26].

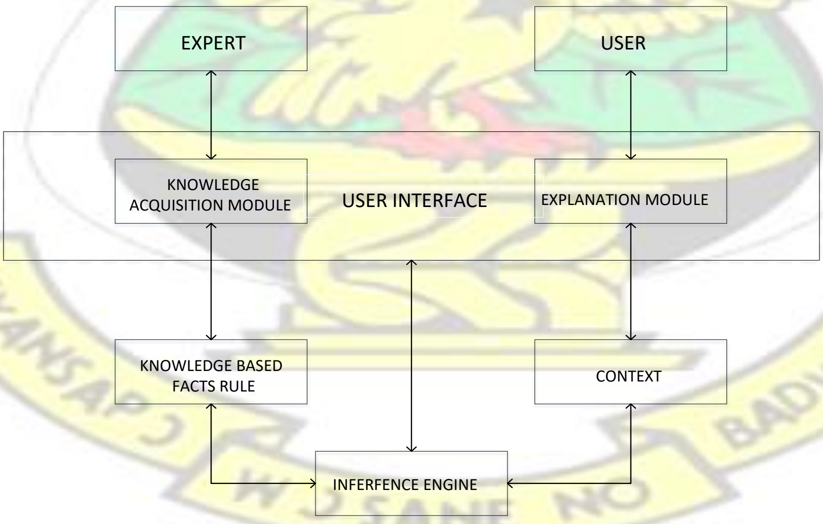

Figure 1.2 Basic structure of knowledge-based expert system

2.5.2.1. Knowledge Base

The knowledge base is the storage place for the system's domain-specific knowledge required for comprehending, formulating and resolving the problem. It encompasses the facts held in a database and rules or additional knowledge representations or heuristics that guide the use of knowledge to resolve a specified issue. Therefore, it characterizes the power of the expert system.

2.5.2.2. Inference Engine

The inference engine achieves the actual processing of the knowledge. It consists of the control strategy or problem-solving mechanism. It combines information provided by the user with the rules or facts in the knowledge base to recommend to the user via the user interface, which is used to solve a problem or to reach a goal. Considering what deductions can be reached or what further information is required, this is a key contrast between knowledge-based expert systems and conventionally fixed time cycled controller programs. Therefore, it is termed the brain of knowledge-based expert systems.

2.5.2.3. Context

This is also referred to as the short-term memory. The context contains the dynamic or problem specific knowledge, the user response to a question asked by the system, and other provisional information produced by the system. It is also recognized as the working memory, the workplace or short-term memory.

2.5.2.4. Knowledge Acquisition Module

The acquisition module is considered a subdivision of knowledge engineering. The main objectives in knowledge acquisition modules are to aid the expert system. That is, curtail or remove the role of the knowledge engineer as the transcriber of the expert's knowledge.

Moreover, it is to enhance the growth and conservation of the knowledge base.

2.5.2.5. Explanation Module

The explanation module gives the system the capacity to explain its reasoning process and provides definitions and other statistics to the user. In addition, this module helps the field expert verify the system's reasoning in the system debugging.

2.5.2.6. User Interface

The user logs into the system through a user interface that is user-friendly so that human and machine can communicate directly and proficiently. Language processors, which are software programs used to perform tasks, such as processing program code to machine code. Language processors are used to implement the GUI of the user interface. Language processors are written in languages such as FORTRAN and COBOL [27].

The foremost importance of designing an expert system is to relate the proficiency and knowledge attained from experts to a computer program. The requirement of expert systems developers and knowledge engineers is to develop efficient methods so that system can do this desirably. [28] [29].

Zozaya-Goristiza and Hendrickson [32] initially tackled an expert system method to the traffic control problem. After their effort, expert systems including the French System [31] were realized. Even though queue times had been reduced, they were still heavily limited, owing to the fact that they are required to manage extremely large volumes of information, they are extremely slow. Additionally, their design catered for global difficulties and were rather incapable of accounting for significant local variations. Distributed and dynamic management theories have not been used even though they offer various benefits. The traffic system defined by Findler and Stapp [31] is grounded on the transmission of data in a network of vehicular signals. The designers presume a grid street configuration, with many simplifications. for each intersection to transmits uninterrupted with the four processors they recommend a processor at adjacent intersections. The operation of the system is founded on a set of co-operating real-time expert systems employed in aggregation with a simulation-based planner [32].

Their decision to alter the cycle start periods relates to the weighted references conveyed from the adjacent and more distant processors. Though their work is responsible for a major degree of system localization, it does not deal with conflict resolution very well. Although expert systems are adept at learning rules and perform better than fixed time cycled controllers, they are flawed because they have to make various assumptions to avoid complex computations.

2.6. Prediction based optimization

This uses predictive analysis, deals with the extraction of information from data and using the extracted data to predict trends and patterns [33]. Frequently an unknown event of interest is in the future, but applying prediction-based optimization to any event or unknown in any particular instance of time be it the past present or future can be determined. This means the key essential in prediction-based optimization, depends on establishing similarities between explanatory variables [34] and are predicted variables acquired from previous occurrences. These variables are then used to decide the unknown outcome. The accuracy and usability of the resulst are very reliant to a large extent on the degree of the data analysis done and the quality of the assumptions made.

Tavladakis et al. [36] characterize a traffic light controller outlined by methods of a simple predictor. Estimations taken in the present cycles are utilized to test a few conceivable settings for the following cycles. The setting resulting in the minimum number of queued vehicles is performed. The framework appeared profoundly self-adaptive, and perhaps possibly too much. It only required data of one cycle which could not deal with robust unstable rush hour gridlock streams well enough. In this case, their design adapted too speedily, causing the system to fluctuate and essentially fail.

Liu et al. [44] design a strategy to resolve issues with fluctuations. Traffic detectors placed on two sides of a junction in conjunction with vehicle detection systems are utilized to evaluate delay times of vehicles at an intersection, unsurprising to an evaluated delay time by means of a filter function to smooth out arbitrary instabilities. The management framework attempts to curtail not only the over-all delay but the added deviations from the normalized delay as well. Since it is not any more profitable to allow vehicles to queue for prolonged periods, regardless of whether giving it a chance to continue would expand the aggregate waiting time, this introduces a kind of fairness. Information of around 15 minutes is utilized to decide the ideal settings for the following cycle, and even using a straightforward optimization algorithm; the system performs fine compared to fixed timed controllers. The system they designed performed fine contrasted with the fixed timed controllers. The only issue with the system was the exponentially long waiting and queuing times

2.7. Fuzzy Logic Traffic Controllers

Fuzzy logic is a branch of computing subject to levels of truth, unlike the typical true or false and ‘0’ and ‘1’ Boolean logic on which contemporary computers principles are constructed on. This approach to traffic characterizes the period in which the light ought to be in a specific state before exchanging to the following, however, the course of action of the states may not be prearranged. The controllers are smart enough to skip idle states. The number of arriving vehicles is quantized into fuzzy variables, for example, numerous medium and none, depending on the number of quantized variables. The actuation of these fuzzy variables is done by a member function [40]. Tan et al. [39] characterize a fuzzy logic controller at a single intersection that ought to imitate human insight.

Fuzzy rationale proposes an official technique for dealing with terms like," all the more"," less"," longer". Subsequently, rules equivalent to “if there is more traffic in the northbound to southbound direction, the lights should remain green longer” can be fabricated with the fuzzy logic controller. It manages the time that the activity light should remain in a particular state, before changing to the following state. The instruction of states is pre-programmed, but the controller can hop a state if there is no traffic in a particular direction. The stimulation of the variables in a precise situation is assumed by a membership function.

Fuzzy principles manage if the length of the present state should be extended. In tests, the fuzzy logic controller showed to be more adaptable than settled controllers were and vehicle impelled controllers, allowing the activity to stream all the more promptly and plunging waiting time. A disadvantage of the controller is that it gives off an impression of being dependent on the pre-set evaluations of the fuzzy variables. They might cause the framework to flop if the whole measurement of traffic fluctuates. Moreover, the framework was only verified for a single traffic intersection.

Lee et al. [40] likewise design a system with fuzzy rationale for checking various intersections. Controllers gained additional data on automobiles at the prior and subsequent junctions and were able to sustain green waves. The framework outperformed the fixed timed cycled controller and was at its best in either light or overwhelming activity. The controller easily took care of varieties in rush hour gridlock streams yet it required diverse imperative settings for every intersection.

Choi et al. [50] pondered the use of fuzzy logic controllers and modified them to manage blocked traffic flows. Assessments with fixed fuzzy-logic traffic light controllers deduced that this modification prompted longer movement stream concerning extremely congested traffic circumstances

2.8. Evolutionary algorithms

Taale et al. [42] relate the utilization of evolutionary algorithms (a (μ, λ)) advancement methodology to develop a traffic light controller for a solitary simulated juncture crossroads utilizing the generally fixed traffic light controller from Holland known as the RWS Ccontroller [33]. They discovered comparative outcomes for the two frameworks. Unfortunately, they did not endeavour to execute their plan on various isolated intersections, since the design of such a complex system of activity components is extremely complicated and developing controllers for these crossing points required complex research questions.

2.9. Reinforcement learning

This is different to machine learning hence, a subdivision of artificial intelligence that facilitates machines and software agents to be able to dynamically control the ideal behaviour within a definite context to increase performance. The reinforcement signal is the simple reward feedback essential for an agent to study its behaviour [34]. As clearly stated, reinforcement learning enables these agents to learn based on feedback from the environment. The learning process can be either static or dynamic dependent on the area of study.

The automation requires little or no need for expert human intervention and very little time would be required in the fabrication of complex sets of rules in the design of expert systems [35]. Thorpe [47] premiered studies on reinforcement learning for TLC, He used a TL-based value function. Thorpe proposed a neural system for the TL based value function, which forecasts the average queuing time for all vehicles waiting at the intersection. This meant that Thorpe's TLC had to handle an enormous number of states, where learning periods and variance was huge. Additionally, Thorpe used to a certain degree other forms of Reinforcement Learning such as the State–action–reward–state–action (SARSA) algorithm for learning a Markov decision process policy with suitability traces [42].

Thorpe trained just single TLC and confirmed it by instantiating it on a system of 4 × 4 traffic controllers. The controller could make plans to let either movement on the north-south hub or on the east-west hub pass. A neural system was utilized to visualize the Q-values for every decision, in light of the quantity waiting cars and the period since the controller last changed. The objective state is the state in which there no cars waiting. The system outperformed both fixed and rule-based controllers in a realistic simulation with varying speed. The performance was said to be near ideal Controllers prepared on a solitary intersection had an indistinguishable execution from controllers prepared in a grid system. Thorpe's approach is not at all like Wiering et al. [38] he utilized a traffic light-based value function, and they used a car-based one. Thorpe utilized somewhat other forms of RL, with eligibility traces by and Wiering et al. Using model-based RL. Correlation of all traffic frameworks described.

Table 2.1 Comparison of all traffic systems described

Table 2.1 is a summary of all the traffic systems described and their respective limitations.

Traffic Systems	Fixed cycled Controllers	Expert systems	Prediction Based Optimization	Fuzzy Logic	Reinforcement learning
Function	Static nonconflicting phases, which work together.	Uses a set of rules to decide the next actions in traffic control	Current cycle is used to predict and test several possible settings resulting in the least number of queued vehicles being used and executed.	defines the period in which the light should be in a particular state before switching to the next, though the order of the states is not predetermined	Agent based traffic control
Limitation	Need to be updated regularly to changes in traffic conditions	Too many assumptions were made to avoid complex computations	Single cycle data was not enough to handle strong fluctuations in traffic flow. System adapted too quickly resulting in poor performance.	Requires different parameter settings for each junction. Complex and requires information from both previous and the next junctions to operate efficiently	All controllers are connected and can cause failure if one controller fails. Traffic light controller have to deal with an enormous number of states.

2.10. Traffic control Strategies

Traffic control strategies refer to the various generational methods that have been used to control traffic policies in literature as shown in Table 2.2. Traffic congestion in urban road and freeway networks leads to a strong degradation of the network infrastructure and accordingly reduced throughput, which can be countered via suitable control measures and strategies.

Table 2.2. Traffic Control Strategies

Control Systems	First generation noncomputerized systems	Second generation centralized computer control	Third generation distributed computer control
Function	Control functions are performed either by specially designed hardwired logic in the form of an electromechanical device or by electronic logic.	Control tasks carried out by single computer program where the number of intersections is relatively small	Hierarchically structured distributed control systems used in large networks, it is necessary to have sub control centres.
			Totally distributed traffic control systems that require no control centres, each local processor solves the control tasks occurring at its own intersection.
Limitations	Cannot cope with unexpected changes in traffic flow	Requires the installation of a computer hierarchy system when coupling several hundreds or thousands of intersections	Self-learning traffic lights from pre-existing data to ensure optimal performance.
	Precomputing signal plans from historical traffic data	Single point of failure for coupled intersections	

2.11. Related Work

Anokye et al [51] considered traffic management in the Kumasi Metropolis using parameters to determine the behaviour of traffic modelled after M/M/1 queuing system. They assumed that time interval between successive arrivals and serving time is independent and identically distributed. The they also found out that given sufficient amount of time, at most only one arrival can occur. Their system also assumed to reach a steady state, was dependent on the rate of arrival and service is constant. The queuing discipline observed was first-come first-served (FCFS). Their attempt at traffic control was aimed ensuring that road users obeyed the queuing disciple to ensure the smooth flow of traffic.

Current research trends have moved towards the third-generation which is the totally distributed traffic control strategies, with related works looking at traffic control strategies on a network wide level. Systems such as the Split Cycle Offset Optimization Technique and the Sydney Co-ordinated Adaptive Traffic and many other similar variations all over the world. Table 2.3 gives the comparison between SCOOT and SCATS

Table 2.3 Comparison between SCOOT and SCATS

Traffic Control Strategies	Split Cycle Offset Optimization Technique (SCOOT)	The Sydney Co-ordinated Adaptive Traffic Control (SCATS)
Description	urban traffic control and management system established by the Transport Research Laboratory (TRL) with the UK traffic systems industry. SCOOT is an adaptive system, which self-adjusting to traffic fluctuations.	System designed by the Roads and Traffic Authority (RTA) of New South Wales, Australia
Method of Operation	It continuously measures traffic volumes approaching all junctions in the network, then varies the traffic signal timings to minimize the Performance Index (PI), SCOOT performs a real-time optimization of signal settings utilizing a traffic simulator	SCATS is a hierarchical, cyclic, traffic responsive signal control strategy that adjusts cycle time, splits and offsets in response to real-time traffic demand to minimize overall stops and delay. It is not a model structure design, but has a library of plans that it chooses from and therefore relies widely on available traffic data.
Limitations	Inability to handle closely spaced signals due to its detection configuration requirements	The error messages ie. flags & alarms, are not easy to decipher and, do not provide the opportunity for corrective actions by system operators.
	Cannot cope with unexpected short-term random fluctuations	Does not have predictive capabilities

Chapter 3

RESEARCH AND METHODOLOGY

3.1. Conceptual framework

The proposed system is a distributed intelligent traffic management system adept at changing itself to meet the demands of the current road user, while alleviating the effects of the fixed cycled controllers in place. It uses a historical data input, which is encoded into a construct, that the Multi Connect Architecture Associative Memory can process. Newly collected traffic flows are gathered over a period. A Pattern Checking Algorithm is used to check whether the newly collected traffic flows are recurrent with any class of historical traffic flow patterns. The pattern Checking Algorithm is then used to compare the current and historical data. A Pearson product-moment correlation coefficient is calculated to measure the linear correlation between the newly collected and historical data traffic patterns, non-recurrent data is made recurrent by a pattern updating algorithm. The updated traffic flows are stored in the database to be historical traffic flow patterns which are also the inputs of Iterative Tuning strategy. After the processing is done, the system response is sent to the Central Processing Station, which interfaces the traffic light controller.

3.2. Modelling in SUMO

SUMO which is the Simulation of urban mobility is an open source, highly portable, microscopic and continuous road traffic simulation package designed to handle large road networks. Primarily SUMO simulations are done in two ways. The initially via the command line SUMO and the second one accomplished in the command SUMO-GUI. In the same way, both methodologies are command line features. The prime difference between them is the fact that sumo does the simulation in the background, minus GUI. There are fundamentally two different types of information that are essential in order to commence a SUMO simulation:

- A traffic pattern: A Network Topology contains an array of roads, railways, pedestrian ways, marine routes or other means of moving cars, buses, trams, trucks, trains boats or people.
- A traffic demand pattern: A Traffic Pattern Demand encompasses the cars, buses, trams, trucks, trains, boats or people moving around, in a particular pattern in the network.

The two elements above are incorporated by means of a configuration, which are the essential parameters to be defined in order to run a simulation.

In addition to a Network File, defining the network topology (*.net.xml) and other Route Files, defining the traffic pattern as vehicles (*.rou.xml), Additional Files (*.add.xml) could be quantified. These additional files may be added, and label add-ons for the simulation, for example, the characterisation of induction loops or additional kinds of sensors fitted in the network, a background picture containing the aerial photograph of the selected region, giving a more comprehensive visualization of the environment, Points of Interests, like building names, shopping stores, or other landmarks.

A SUMO network file (*.net.xml) explains the traffic-related portion of a map, the roads, and intersections where the simulated automobiles ride along and across. At a coarse scale, a SUMO network is a network graph. Nodes, commonly called junctions in SUMO framework, characterize intersections and edges of roads or streets

3.2.1. Modelling the Current System in SUMO

In modelling the current system, which is the current system is modelled after the actual traffic management system used in the Kumasi metropolis which is the fixed timed cycled controllers, we need to consider the following

- The selection of the area to be simulated using the OpenSteetMaps street maps.
- Conversion from the OSM street maps to a SUMO network.
- Simulation of the current system in the SUMO environment.

3.3.2 The Osmwebwizard.Py Script

The selection of the specific area to be modelled in SUMO is done using the osmwebwizard.py. The OSM Web Wizard is essentially a collection of python scripts placed under the directory tools in the SUMO installation root. OSM Web wizard is started by starting the osmwebwizard.py command in the tools directory. The OSM Web Wizard gives one of the simplest solutions, to begin with, SUMO.

The osmwebwizard.py is created on a selection of an OpenStreetMap map that allows the configuration of randomized traffic that enables demand, run, and visualization scenarios in the SUMO-GUI.

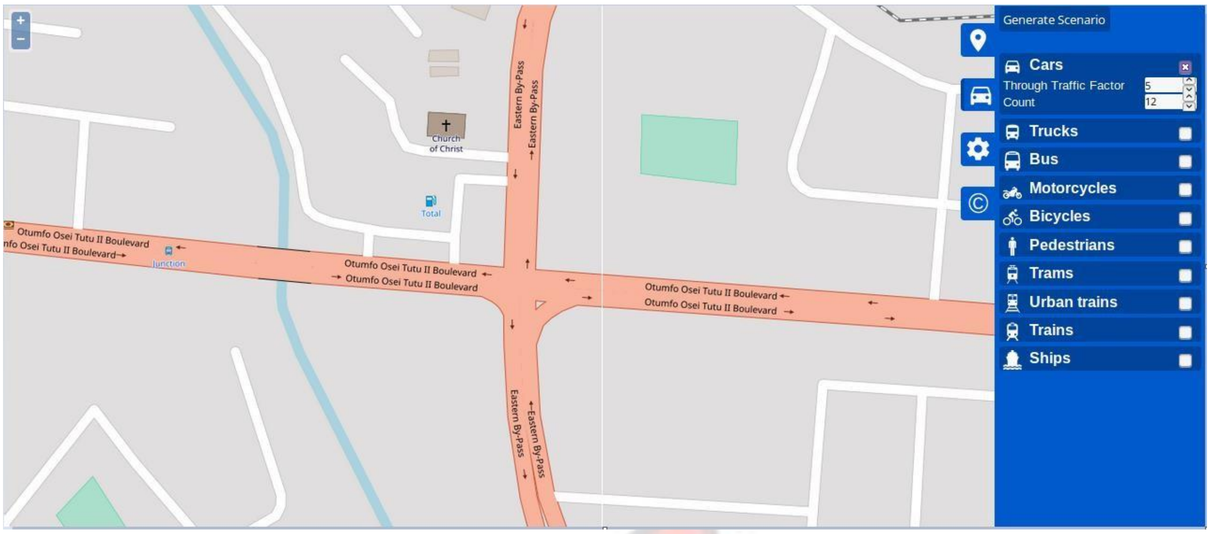

Figure 3.1 OSM Street map view of the Selected area

The script allows the zooming and panning to the selected area of interest. Picking the appropriate simulation area is key for the simulation process. If the area chosen for the simulation is very large, it takes a long a long time to load the simulation. The simulation also becomes very slow taking a very long to execute and run.

The next step is to select the definite area for the generation of the simulation setting. The area selection is triggered by selecting the checkbox Select Area at the blue zone section panel on the right part of the map. For the specifications of the research, Tech junction to Top High, Top High to Oforikrom, Amakom to Oforikrom, Oforikrom to Anloga and Aboabo to Oforikrom all in the Kumasi metropolis are considered. The selection of this particular intersection is due to the difficult management of traffic as a result of the volume of vehicles that use the intersection.

.

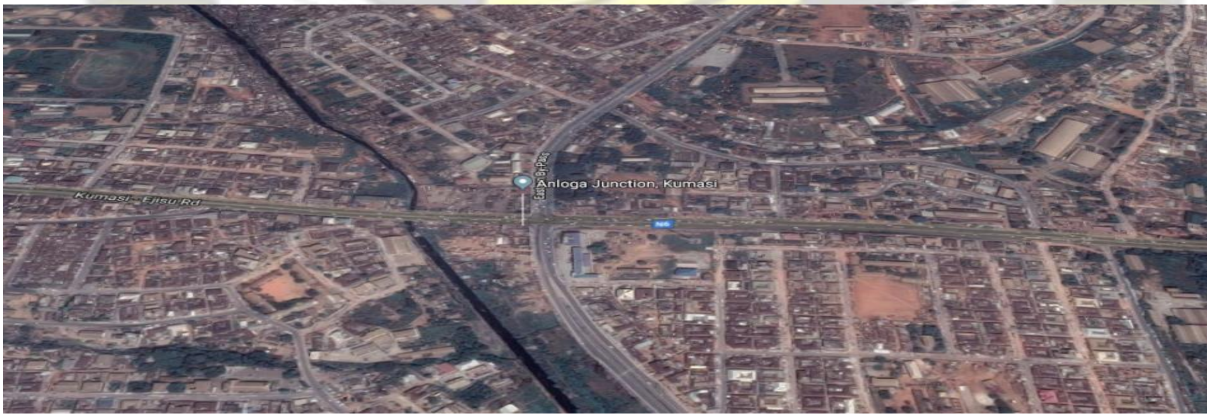

Figure 3.2 Satellite view of the selected area

3.3.3 Demand Generation

The demand generation panel does the demand generation. It can be activated by clicking on the car pictogram.

Simulation of Urban Mobility provides several methods of transport. At the demand generation panel, the individual types of transport are activated or deactivated by clicking the equivalent checkboxes. For each mode of transport, the OSM Web Wizard creates random demand based on a certain probability distribution, which is influenced by two constraints.

- Every single time a vehicle is fashioned the OSM Web Wizard arbitrarily chooses departure and arrival edges for the vehicle. The Through Traffic Factor explains how many times it is more likely for an edge at the boundary of the simulation area being chosen compared to an edge entirely located inside the simulation area. A big value for the Through Traffic Factor implies that many vehicles depart and arrive at the boundary of the simulation area, which corresponds to a scenario with a lot of through traffic. • The Count constraint outlines how many vehicles are produced for the network per hour and lane-kilometre. For instance
- A network containing 3 edges with a total length of 15 km
- That each has 4 lanes which allows the current traffic mode
- and the count value is set to 100
- Then 15 * 4 * 100 = 6000 vehicles per hour will be generated. This translates to a random Trips parameter of p=4 which means a new vehicle is inserted every 4 seconds somewhere in the network.

The next step is generating and running the scenario.

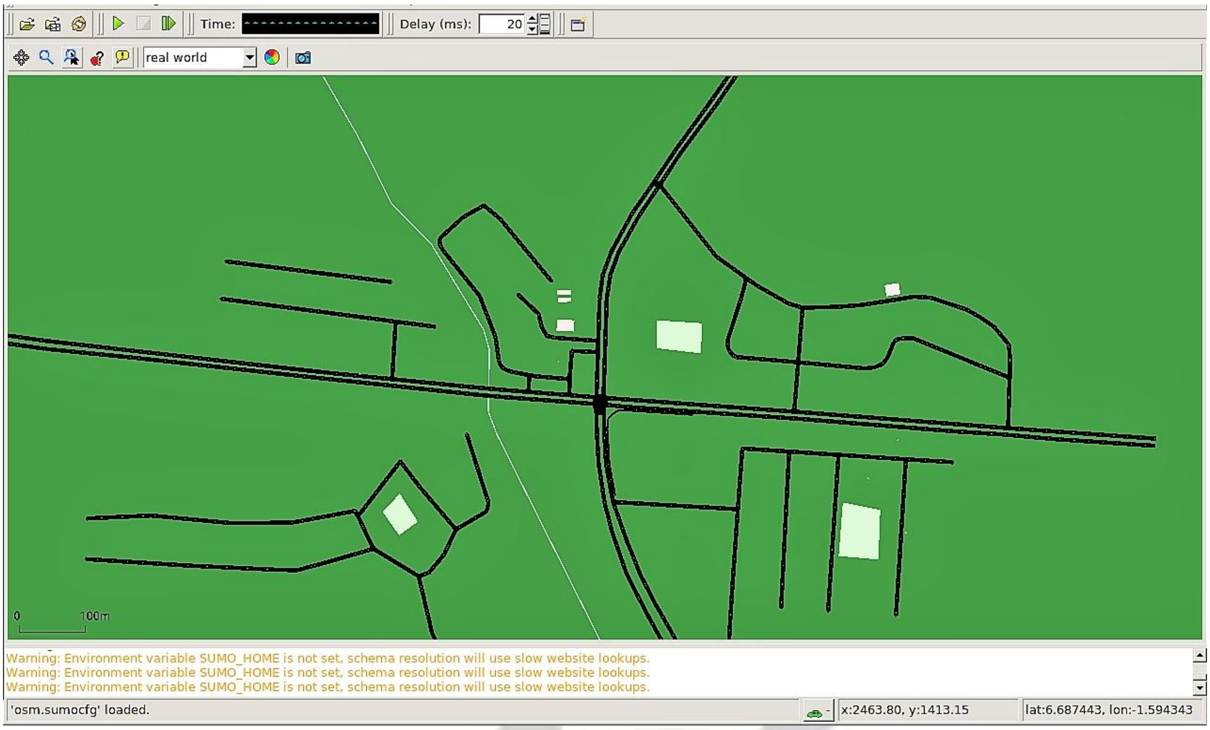

Figure 3.3 SUMO generated network for the selected area

3.3.4 Scenario Generation

The complete scenario is created inevitably once the Generate Scenario button in the control panel has been selected. The scenario generation takes a period of time dependent on others on the size of the scenario. After the scenario generation phase is complete the SUMO Graphic User Interface begins and the simulation is commenced by choosing the Play-button in the SUMO-GUI. This gives the basis for the design of both the current and the proposed system designs. In simulating traffic controllers in SUMO, NETCONVERT and NETGENERATE create TLC and programs for intersections during the calculation of the networks. Nonetheless, these computed programs may be dissimilar to those found in actual networks. To feed the simulation with actual traffic light plans, the software is run with additional program descriptions. In addition, the software allows for loading definitions, which describes when and how a set of traffic lights can switch from one program to another. Another possibility is to manage traffic light plans visually in NETEDIT. It is very important to remember the colour states in traffic simulation.

Table 3.1 Traffic indicator descriptions in SUMO

Table 3.1 gives a description of all the traffic phases in the simulation of urban mobility simulator and their relevance. The characters are used to describe the various phases of traffic control. The traffic indicator descriptions are used to adapt the traffic controllers in the simulator to make them practical.

Character	Description
R	'red light' for a signal - vehicles must stop
Y	'amber (yellow) light' for a signal - vehicles will start to decelerate if far away from the junction, otherwise they pass
G	'green light' for a signal, priority - vehicles may pass the junction
U	'red+yellow light' for a signal, may be used to indicate upcoming green phase but vehicles may not drive yet (shown as orange in the gui)
O	'off - no signal' signal is switched off, vehicles have the right of way

3.3.5 Modelling the Current System Performance

The current system was simulated using fixed time cycle controllers and a default cycle time of 120 seconds. All green phases come after yellow phases. If a green phase permits for somewhat differing flows, that is forward going and left turns from the opposite directions, it is followed by the next green phase with full priority to the partially conflicted streams and this would classically be a left-turning phase. Practically, there are regular phases where all streams have red to allow clearing an intersection, and this is essentially programmed into SUMO. The traffic controllers are added as additional files and further edited in NETEDIT to match it to the current first generation fixed timed cycles in the metropolis.

3.3.6 Modelling the Proposed System Performance

In simulating the proposed system, just as in the current system an osmwebwizard.py script is used to select the simulation area, input demand and vehicular variables and generate a SUMO network of the area under consideration, which are the selected junctions in and around the Anloga junction specifically. Unlike the current system, the proposed is a distributed intelligent traffic management system that uses historical traffic data to learn and optimize the duty cycles of traffic controllers changing traffic intensities and densities. The first thing considered in modelling the system is the capturing process. SUMO enables the use of lane area detectors also known as E2 detectors which has an effective detection zone characterized by the start and end positions in the simulator, vehicles are electronically counted when in the effective detection zone. This is used to confirm traffic on area along lanes. In the real world, this is comparable to vehicle tracking cameras. The outputs of E2 Detectors are designed for computing queued vehicles and it keeps track of all automobiles, which are presently in its area.

3.2 Simulation Attributes

In simulating the proposed system in Sumo, a python file is created. In the script, that integrated with the TraCI, which is the traffic control interface, to generate a client-server design to allow for dynamicity in the simulation. A SUMO generated network of the desired simulation area, which for the purposes of the design is Tech junction to Top High, Top High to Oforikrom, Amakom to Oforikrom, Oforikrom to Anloga and Aboabo to Oforikrom located in and around the Anloga Junction, is required. Implementation of a dynamic traffic control algorithm can be used to alter the phase times and cycles of the traffic controllers. This is done by the concept of calculating the oncoming vehicular densities captured by the E1 detector known as induction loop detectors, which emulates vehicle-tracking cameras and using the count of these vehicles to change the duty cycles of the traffic controllers to prevent congestion.

Table 3.2 Description of attributes

Attribute Name	Value Type	Description
Id	id (string)	A string holding the id of the detector
Lane	referenced lane id	The id of the lane the detector is laid on. The lane must be a part of the network used. This argument excludes the argument lanes.
Lanes	referenced lane id list	A space-separated list of lane-ids, which the detector covers. All lanes must be a part of the network and form a continuous sequence. This argument excludes the arguments lane and length.
Pos	Float	The position on the first lane covered by the detector. See information about the same attribute within the detector loop description for further information. Per default, the start position is placed at the first lane's begin.
endPos	Float	The end position on the last lane covered by the detector. Per default, the end position is placed at the last lane's end.
length	Float	The length of the detector in meters. If the detector reaches over the lane's end, it is extended to preceding consecutive lanes.
Freq	Int	The aggregation period the values the detector collects shall be summed up. Either freq or tl must be specified
Tl	Id	The traffic light that triggers aggregation when switching. Either freq or tl must be specified
To	Id	The id of an outgoing lane that triggers aggregation in conjunction with traffic light switching. This is only used together with tl.
timeThreshold	Int	The time-based threshold that describes how much time has to pass until a vehicle is recognized as halting; in s, default: 1s.
speedThreshold	Float	The speed-based threshold that describes how slow a vehicle has to be to be recognized as halting; in m/s, default: 5/3.6m/s.
jamThreshold	Float	The minimum distance to the next standing vehicle in order to make this vehicle count as a participant to the jam; in m, default: 10m.

3.5. Proposed System

3.5.1. Proposed System Flow Chart

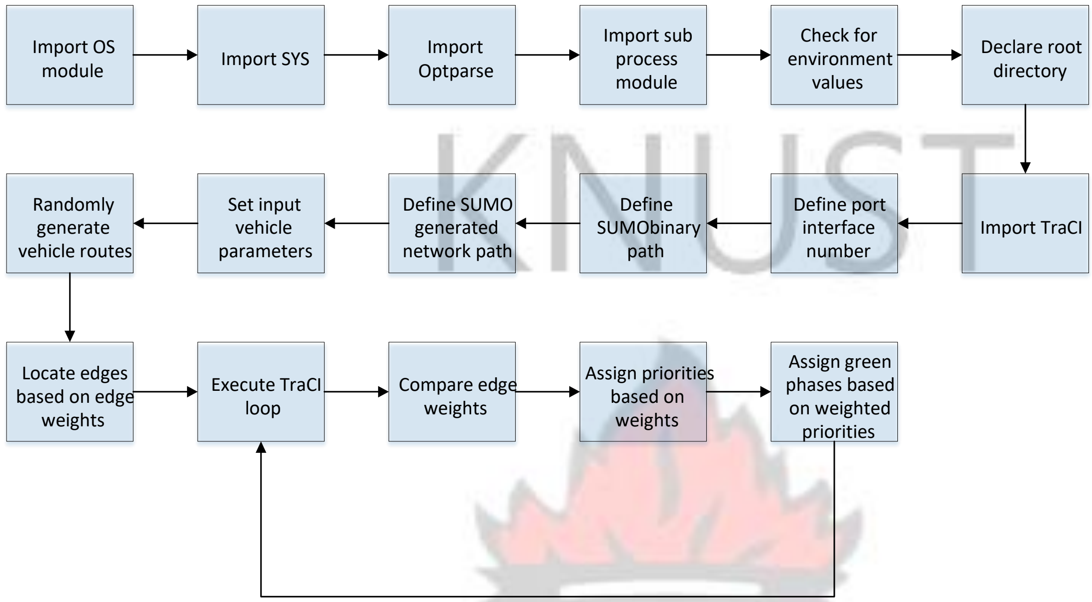

Figure 3.4 Simulation flow chart

The figure 3.4, the flow chart summarizes how the project is carried out by the simulator, from importing all system modules to assigning priorities based on weighted edges.

3.5.2. Proposed System Design

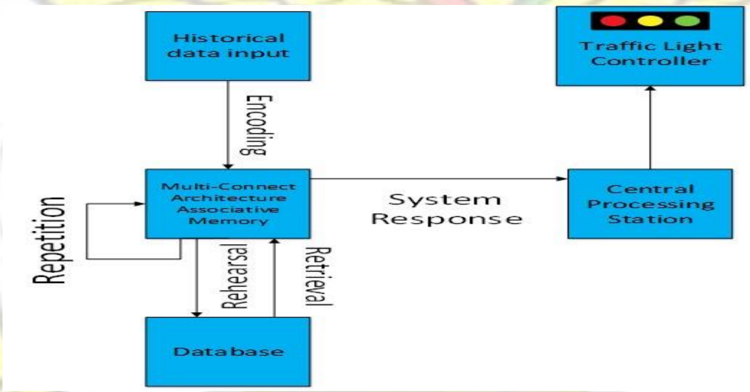

Figure 3.5 Proposed system design

For the proposed system as shown in figure 3.5 to effectively capture and process traffic, a degree of intelligence and sophistication is required hence the system is trained. The proposed system introduces intelligence in managing traffic efficiently by using a training and learning algorithm implemented in the MCA Associative memory.

3.6. SYSTEM DESIGN COMPONENTS

Implementing the system in the Kumasi Metropolis will require the following design components.

3.6.1 Historical Data Input

The historical data input is collected statistically from major intersection points within the metropolis over a period. It includes counting cars at various times of the day and computing this iteratively to give the appropriate historical data and future predictions.

3.6.2 Multi-connect Architecture Associative Memory

MCA is established as an adapted Hopfield neural network. Thus, its architecture, learning process and convergence process are revised founded on two philosophies:

- Use minutest net size.
- The learning procedure will be completed to a limited number of example's vectors to abstain from taking in a similar vector a few times.

Dependent on the first principle. The formation of a neural network capable of carrying out associative recall, the network has only three neurons irrespective of the length of the training vectors. In any case, it simply a traditional Hopfield neural system, where every hub is interconnected however, not to itself by no less than one association and the most extreme is constrained to four associations. Every association quality or weights is symmetric so the weight from hub I to hub j is equivalent to that from hub j to hub I, that is, wij=wji and wii=0. This structure net prompting assumed edges to be zero. Hypothetically, as obsolete Hopfield net, this net has two phases, which is the learning and joining stages. These procedures are changed. It is noticed, the information in this kind of net isn't in a singular course since it is plausible for signs to float from a hub back to itself by means of different hubs. There is criticism in the system, or that it is repeating since hubs could be utilized intermittently to create data. This is to be separated with the feedforward nets. This introduces the altered design of MCA. To attain simplicity, this section presents two algorithms for both phases, which are the learning and convergence. Based on the MCA learning algorithm the training pattern will be separated into n vectors named vk (where $0 < k \leq n$). Each vector v will be in size three. Therefore, there are no more than eight likelihoods for these vectors. This restricted number of these vectors lead to making the mandatory weights also limited. Although these weights have the same structures of that weight resulted from Hopfield neural network learning process as a square matrix, symmetric matrices and the diagonal elements of these matrices are zero, but the size is 3×3 regardless of patterns' vector length and the fact that the number of these matrices is four. Depending on MCA convergence algorithm, just like in the MCA learning algorithm all the four weights and energy function matrices (w0, w1, w2, w3, and e) will be modified and the unknown pattern will be divided into k and v vectors with size three. Using the initial weight and energy matrices. This algorithm computes the summation of energy functions between all vectors in the unidentified pattern with all weights matrices in each stored pattern and stored these matrices in the energy function matrix calculates energy functions between the unknown pattern and all the stored pattern weight matrices w. The MCA is adept at both the learning and convergence and is crucial in the training of the system.

3.6.3 The Central Processing Station

The central processing station is the portion of the system that conveys the system response to the traffic controller. To accomplish this, it uses the elementary numerical, logical, and incoming or outgoing processes of the system. The two typical components of central processing station are the ALU, which implements arithmetic and logical operations, and the CU, which recalls instructions from memory, interprets, and executes them, requesting the ALU when necessary. The role of the central processing unit in the system is to interface the system response, which is the actually optimized duty cycles with the traffic controller for the management of traffic. This is reflected by the traffic controller as a set of lights with variable split and phase times.

3.6.4 The Database

A database is a prearranged collection of data, which is any classification of one or more symbols given meaning by explicit acts of interpretation. Databases characteristically collate the data to reflect aspects of authenticity in a way that provisions processes requiring information. The database stores trained traffic data over a long period and are required due to the storage limitation the MCA creates.

3.7. Recurrent traffic flow patterns

In municipal traffic systems, the utmost commonly used methodology to label signalized intersections in the Highway Capacity Manual (HCM) [68]. Before Iterative Tuning strategy is familiarized, theoretical bases regarding previous traffic information and Iterative Tuning strategy are enumerated. The research recognises that though traffic may appear random, it is actually not with individuals having regular patterns associated with their movements to and from places such as schools, work places and homes. This implies statistical analysis can be used to generate present traffic patterns and predict future patterns and their corresponding signal schedules. The basic idea here is tuning the signal schedules in anticipation of the daily traffic patterns.

- Daily Traffic Pattern (DTP) outlines the details of traffic records for twentyfour hours of a particular day. Based on the analysis of Chrobok et al. [120], DTPs of normal working days are comparable since procedures are almost the same with the exception of days before holidays or weekends, particularly the midday traffic flows vary. Consequently, working days are characterized by two modules ○ Normal Working Days: working days except the days before holidays.
  - ○ Last Working Days: working days before holidays.
- Daily Traffic Signal Schedules (DTSS) are phase periods for a junction from 0: 00 to 24: 00. For an intersection, if DTPs of phase periods are repetitive, traffic weights in all phases are repetitive and repetitive phase lengths are applied. Consequently, DTSS will be indistinguishable for the dates with repetitive DTPs of phases. The foundation for Iterative Tuning strategy is the recurrence of DTP. The supposition made here is that infrastructures comprising road, residences, and workplaces changes gradually

Iterative tuning is a method of modifying the performance response of a system that occurs repeatedly over a period of time. The foundation for IT strategy is the recurrence of the DTP.

To be able to use iterative tuning strategy, it is imperative to consider the Coefficient of Variation, which is the ratio of how much the DTPs vary from the mean DTP, expressed as a percentage. Mathematically, it expressed as the ratio of the standard deviation (S), to the mean (M).

$$M = \frac{1}{\omega} \sum_{\omega \in nW} x_{j}{}^{\omega}{}_{,i}(\tau) \quad (1)$$

$$s = \sqrt{\left(\frac{1}{n^{\omega}-1} \sum_{\omega \in W} \left(x_{j,i}^{\omega}(\tau) - M\right)\right)^2} \quad (2)$$

$$CV = \frac{S}{M} \quad (3)$$

Equation (3) enables the derivation of the *CV* which then becomes the first test for repetitiveness. This implies that if the *CV* is a small percentage, traffic flow patterns are similar and DTPs are almost the same.

A junction that has the same green phase. For every day, there occurs one set of traffic flows $x_{j}{}^{\omega}{}_{,i}(\tau)$ where w is the index of working days; and ω ∈ W where W is the set of working days. Mean M in Eq. (1), Standard Deviation S in Eq. (2) and Coefficient of Variation V in Eq. (3) are utilized to analyse the variations. Where n^{ω} is used to describe the number of working days.

3.8. Iterative tuning strategy for phase splits

Iterative learning control is a technique for refining the momentary response performance of a system that functions repetitively over a fixed period of time. Iterative Tuning of the phase splits happens in the MCA associative memory. Iterative Tuning (IT) is an approach with phase splits that use historically already existing traffic information. The examination of historical raw data confirms that the traffic flow configurations on normal working days and weekends are recurrent weekly with very trivial or no disparities. Through the expectation of traffic information, Iterative Tuning controllers for all intersections fine-tune the days traffic signal schedules iteratively to decrease the delay time of the traffic network. In Iterative Tuning methodology, daily traffic signal schedules are acquired established by daily traffic patterns respectively. Respectively every class of daily traffic configuration has its own daily traffic signal plans.

The Iterative Tuning methodology provides the subsequent structures:

- It entails little system data and the systematic prototype of the system is not obligatory.
- It is an off-line tuning approach hence the online calculation time is negligible.
- Recurrent disturbances can be disallowed and resulting errors can be recompensed.
- It adjusts to the variations in traffic patterns iteratively.

Iterative tuning strategy is summarized in figure 3.6

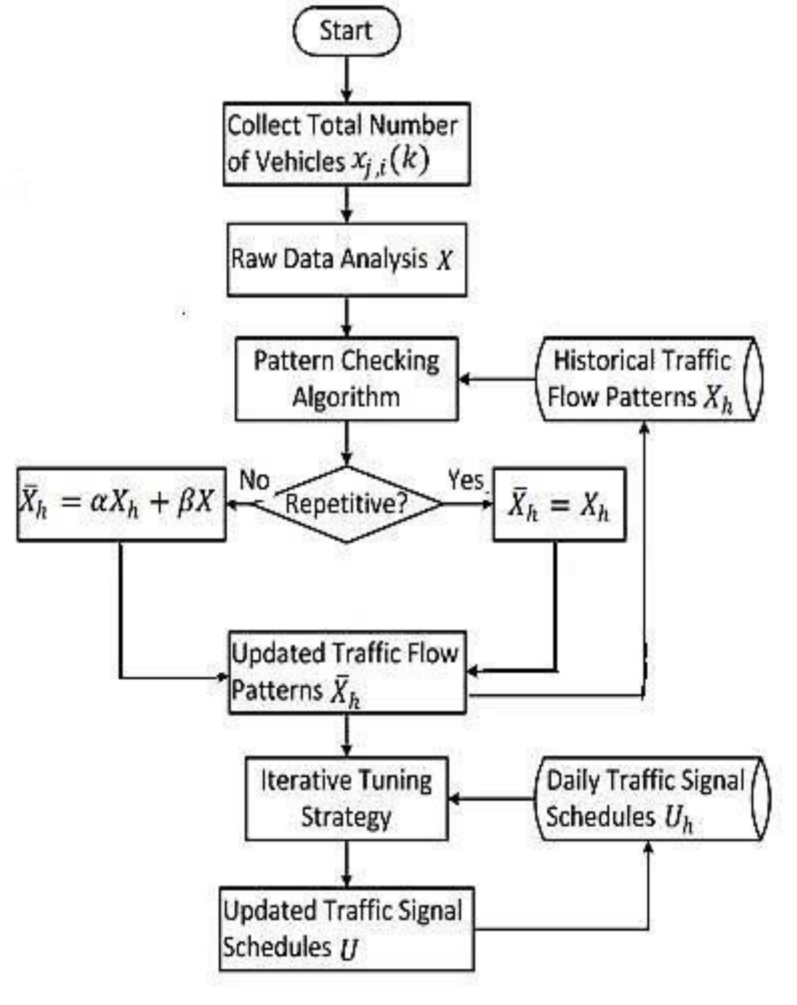

Figure 3.6 Iterative Tuning flow chart

3.8.1. Data and Processing

Before IT strategy is adapted, conceptual frameworks concerning previously existing historic traffic data and Iterative Tuning strategy are summarized for per each intersection, already existing historical traffic data patterns of the full day are stored in memory. For a junction j ∈ J, xj, i(k) ∈ X, ∀i ∈ Fj is compiled over the complete day, with *X* signifying newly compiled traffic information; Fj is represented as the collection of lane groupings of junction j; with k denoting the index of cycles. In anticipation of a special event, traffic flows vary. Hence, a Pattern Checking Algorithm is used to check if the newly collected traffic flows *X* are recurrent with any or all classes of pre-existing historic traffic flow pattern information *Xh* or not.

3.8.2. Pattern Checking Algorithm

This is a Pearson product-moment correlation coefficient γ is then computed to check if the linear correlation between *X* and X_h. *γth* is then set as the threshold Pearson coefficient. If γ ≥

γth, traffic patterns X are recurrent with historical traffic flow patterns Xh. If $\gamma < \gamma th$, traffic patterns X are not recurrent with historical traffic flow patterns.

In both modules of working days, historical traffic data is updated in Equation. (4) to try to adapt to the slowly altering traffic conditions.

$$X_h = (\alpha X \ X_h h + \beta X), \text{ if } \gamma\gamma < \gamma \geq \gamma th; \tag{4}$$

Where α and β characterise the weighting coefficients that have satisfied $\alpha+\beta=1$, this is mandatory in the regulation of the updating speed and not the performance of Iterative Tuning strategy. The modified traffic flows X_h is stored in the database to represent historical traffic flow patterns X_h, which are correspondingly the inputs of the Iterative Tuning strategy.

The updated traffic flows $\overline{X_h}$ are stored in the database to be historical traffic flow patterns X_h, which are also the inputs of IT strategy.

For junction $j \in J$, constructed on the modified traffic flows $\overline{X_h}$, traffic flows $X_j(k)$ for all lane collections are recovered. For a particular phase $p \in P_j$ with phase period $u_{j,p}(k)$. With P_j representing the group of phases of junction j, where there is one or more equivalent lane groups $x_{j,i}(k)$ which have the right of way $X_j(k)$ with $i \in F_{j,p}$, and $F_{j,p}$ becomes the collection of lane groups of junction j that has the right of way during phase p. Maximum phase occupancy $o_{j,p}(k)$ is then computed as the maximum ratio of the traffic flows per lane, $\frac{x_{j,i}(k)}{n_i^{lg}}$ and road capacities $su_{j,p}(k)$, which are conveyed in Equation (5)

$$o_{j.p}(k) = max_{i \in F_{j.p}} \frac{\left\{\frac{x_{j.i}(k)}{n_i^{lg}}\right\}}{su_{j.p}(k)} \tag{5}$$

With n_i^{lg} representing the number of lanes in the lane group i; s signifying the saturation flow per lane which is assumed to be static since every lane has a fixed number of vehicles that will make it gridlocked. Phase occupancy $o_{j,p}(k)$ is converted into phase occupancy errors $e_{j,p}(k)$ in Equation. (6). As in Equation. (7), where $e_{j,p}(k)) \in E_j(k)$, $\forall p \in P_j$ an approach zero, phase occupancies are balanced and the minimum delay times are obtained. If the vector representation $E_{j,p}(k)$ are obtained from the Phase occupancy errors and the maximum phase occupancy with n_p signifying the quantity of the controller phases assigned to the junctions

$$e_{j.p}(k) = O_{j.p} - \frac{\sum_{p\in Pj} O_{j.p}(k)}{p}\ {}_n \tag{6}$$

$$e_{j.p}(k) \rightarrow 0 \tag{7}$$

Phase duration $U_{j.h}$(k) can be retrieved from the database of traffic signal schedules U_h. The tuning technique for the traffic controller phase splits is modelled in Equation. (8)

$$\hat{U}_j(k) = U_{j.h}(k) + LE_j(k+1)$$

$$L = I\lambda \tag{8}$$

$$U_j(k) = C\{\hat{U}_j(k)\}$$

L is defined as the tuning function a that represents the tuning function and is denoted by L = λI; I is the identity matrix with suitable dimensions; λ can be approximated by a trial-and-error method, which does not adversely affect the performance; C{} is the function to consider constraints. For phase $U_{j.p}$(k) = C{$\hat{u}_{j.p}$(k)}, with $U_{j.P}$(k) ∈ U_P(k), ∀p ∈P_j, constraints are considered. As shown in Eq. (9), function C{} takes phase constraints into deliberation. For the apprehension of safety, maximum phase time U_{max} and minimum phase time U_{min} for each phase are predefined.

$$U_{min} < \hat{U}_j(k) < U_{max} \tag{9}$$

Equation (9) is realized by:

$$C\{\hat{U}_{j.p}(k)\} = \begin{cases} \hat{U}_{j.p}(k), & if\, U_{min} < \hat{U}_j(k) < U_{max} \\ U_{max}, & if\, \hat{U}_{j.p}(k), \geq U_{max} \\ U_{min}, & if\, \hat{U}_{j.p}(k), \leq U_{min} \end{cases} \tag{10}$$

$$C = \sum_{p\in Pj} U_{j.p}(k) + t_L$$

Function C{} also ruminates the constraint of cycle length in Eq. (11).

$$C = \sum_{p\in Pj} U_{j.p}(k) + t_L \tag{11}$$

Where C is the cycle length t_L and this is given as the total lost time within a cycle. This constraint in equation. (11) is fulfilled by

$$c\{\hat{U}_{j.} \quad {}_{p \in P j} \quad {}_{j.p}(\ )k \quad {}_{p}(k)\} = \frac{\{\hat{U}_{j.p}(k)\}}{\Sigma \quad U}(C - t_L) \tag{12}$$

Equations Eq. (10) and Equation. (12) are calculated alternately until the constraints Equation. (9) and Equation. (11) are satisfied synchronously. Moreover, phase durations are regulated to be accurate to the resolution of controllers

As phase durations $U_j(k) \in U$, $\forall j \in J$, $\forall k$ is gotten, with U as the vector of phase lengths for the complete day, they will be used for the next time with recurrent traffic flows, deposited in the database as traffic signal schedules U_h.

Where,

- $\boldsymbol{U_{min}}$ minimum phase times
- $\mathbf{U_{max}}$ maximum phase times
- Tuning procedure for phase splits $\hat{U}_j(k)$
- Cycle length constraints C
- Total time lost within the cycle $\mathbf{t_L}$
- cMaximum phase occupancy $o_{j.p}(k)$
- maximum ratio of traffic flows per lane $x_{j.i}(k)$
- Number of lanes in group i n^{lg_i}
- Interarrival rates $\boldsymbol{\lambda}$
- saturation flow per lane $su_{j.p}(k)$
- Phase occupancy of errors $e_{j.p}(k)$
- Vector Rep of phase (k) occupancy of errors $E_{j.p}$
- Tuning procedure for phase splits $\hat{U}_j(k)$
- Tuning function L
- Identity matrix I
- Constraints C{ }
- phase durations $U_j(k)$

Chapter 4
Results and Discussion

4.1. Introduction

The Kumasi Metropolis faces the issue of automobile congestion, which is getting even more severe on daily basis. Few years back, it took about twenty (20) minutes to commute from The KNUST junction to the Kumasi central market but nowadays it takes about one hour with a car. Consequently, this thesis strives to develop a model with vehicular traffic management based on the use of historical data analysis. Thus, Iterative Tuning Strategy for determining the phase splits will regulate the appropriate periods for the red, amber and green lights to be either active or not in order to decrease congestion in the Kumasi metropolis of the Ashanti Region of Ghana.

4.2. Data Sources

This thesis is grounded on data documented within working phases of the morning period with times ranging from 5am to 12pm, the afternoon period from 12pm to 4pm and evening period from 4pm to 9pm. With emphasis placed on morning and evening rush hour periods where roads are typically busy in the Kumasi Metropolis in the Ashanti region. Various intersections were considered at various random intervals and days with the necessary extrapolations made to give a stable traffic pattern in the metropolis.

Table 4.1 data sources

LOCATION	SESSION	ARRIVAL	DEPARTURE
		Average No. cars per hour	Average No. cars per hour
Tech-to-Top High	MORNING	45	60
Top High-to-Oforikrom	MORNING	53	71
Amakom-to-Oforikrom	MORNING	40	50
Aboabo-to-Oforikrom	MORNING	11	12
Oforikrom-to-Anloga	MORNING	75	81
Tech-to-Top High	AFTERNOON	30	50

Top High-to-Oforikrom	AFTERNOON	49	55
Amakom-to-Oforikrom	AFTERNOON	36	50
Aboabo-to-Oforikrom	AFTERNOON	11	10
Oforikrom-to-Anloga	AFTERNOON	58	63
Tech-to-Top High	EVENING	40	55
Top High-to-Oforikrom	EVENING	55	76
Amakom-to-Oforikrom	EVENING	45	60
Aboabo-to-Oforikrom	EVENING	13	15
Oforikrom-to-Anloga	EVENING	88	91

4.3. Daily and Weekly Traffic Patterns

Based on the numerous car counts obtained by the data sources and estimates from the extrapolation, traffic patterns for various days by the hour can be determined. Furthermore, with a regression GP algorithm, future traffic trends can be predicted, so the system is able to predict the expected traffic densities by the hour, enabling the correct phase splits to be set.

Saturday Traffic Data and Prediction.

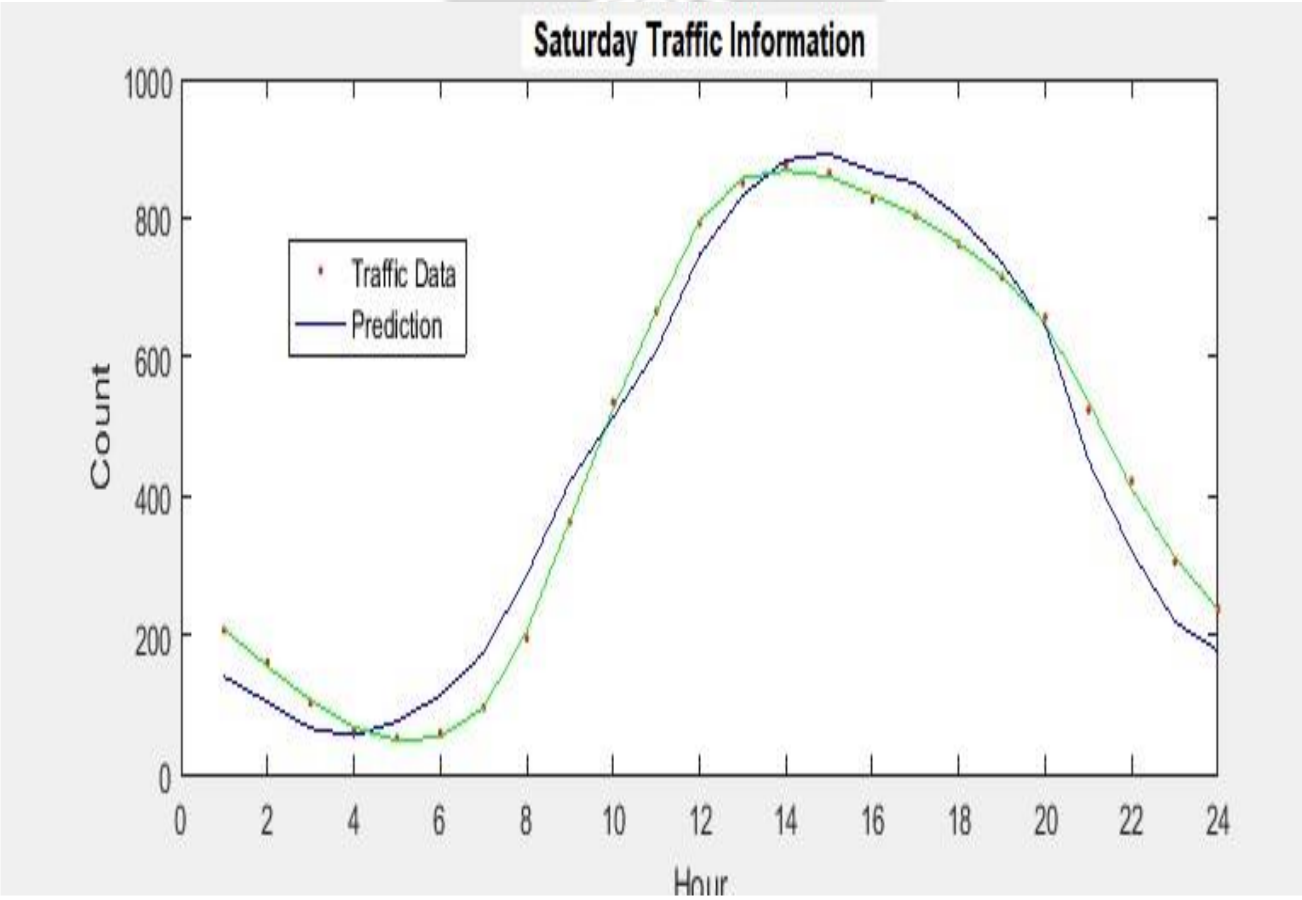

Figure 4.1 Saturday Traffic Data and Prediction

From figure 4.1 which gives an estimate of traffic from 01:00 hours to 00:00 hours lasting the entirety of one day and future traffic trends predicted iteratively. It is clearly shown that

Saturday traffic is at its highest between 12:00 hours and 18:00 hours with a peak value of about 900 vehicles around 15:00 hours. This is due to the fact that Saturdays are usually not too busy with people choosing to stay home for the weekend. People moving to and from market centres and shopping malls around midday constitutes the peak of traffic as shown in figure 4.1.

4.3.1. Monday Traffic Data and Prediction

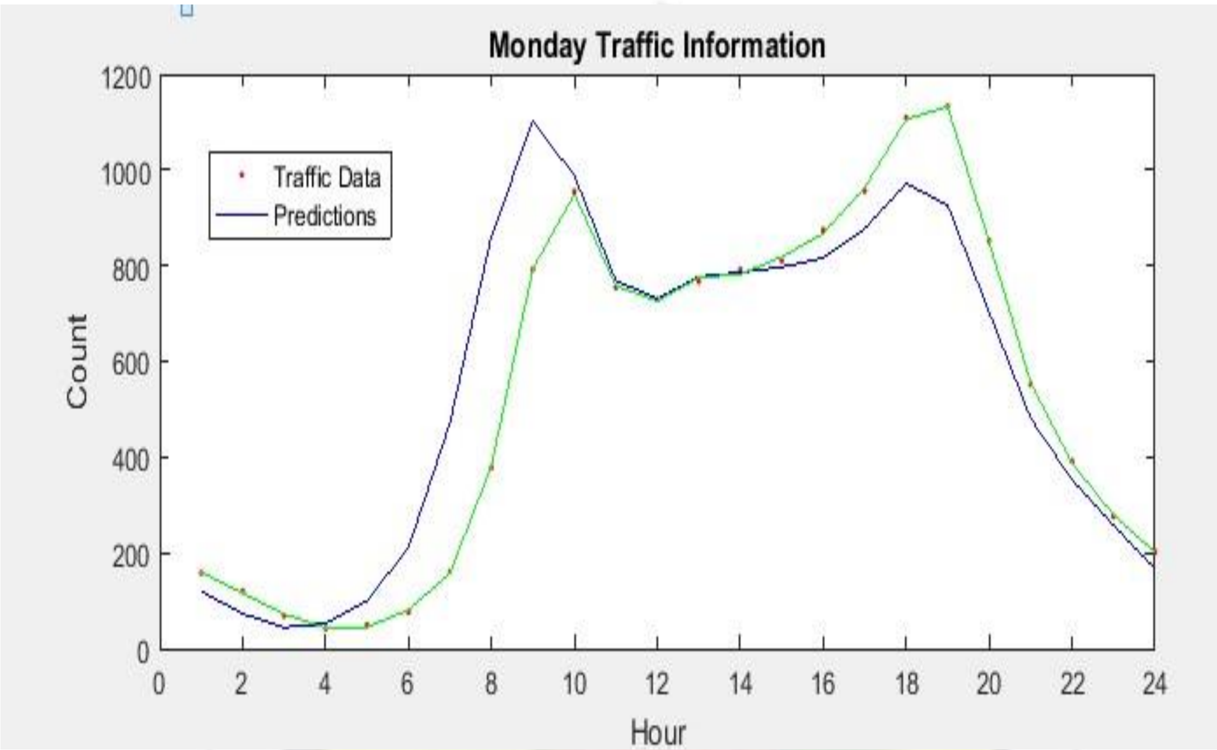

Figure 4.2 Monday Traffic Data and Prediction

Mondays are usually very busy days in the Kumasi Metropolis. From figure 4.2 this is clearly shown. There are two main peaks representing the two main rush hours of the day with the first, which is the morning rush hours occurring between 07:00 hours and 09:00 hours with a peak value of around 1000 cars. This is significant because at this period there are numerous private and public vehicles in the Metropolis with people making their ways from their homes to their various work locations and schools. The second hour period occurs between the hours of 17:00 hours and 19:00 having a peak value of about 1100 cars, with people returning from their work locations home. It is noticeable that between the two peak periods, between the hours of 10:00 to 16:00 hours. The volume of traffic reduces significantly

4.3.2. Wednesday Traffic Data and Prediction

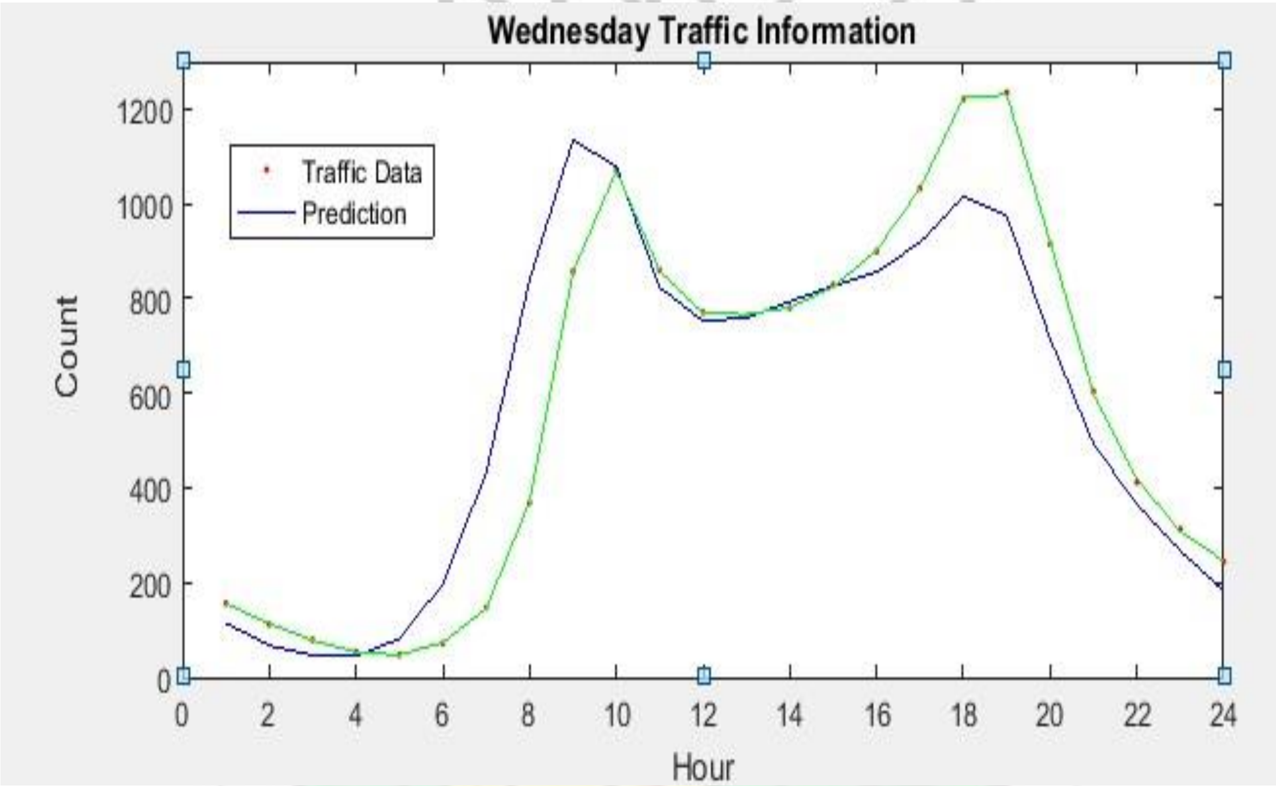

Figure 4.3 Wednesday Traffic Data and Prediction

From figure 4.3, which represents Wednesday traffic data and its corresponding predictions, two peak points are clearly defined in the graph. The first representing the morning rush hours between 07:00 hours and 09:00 hours with a peak car count of about 1000 cars. The second peak representing the evening rush hours with a peak car count of about 1200 hours, between the hours of 17:00 and 19:00. It is seen that the traffic pattern in figure 12 is similar to figure 11 showing that weekday traffic patterns are quite similar.

.

4.3.4. Weekday Traffic Data and Prediction

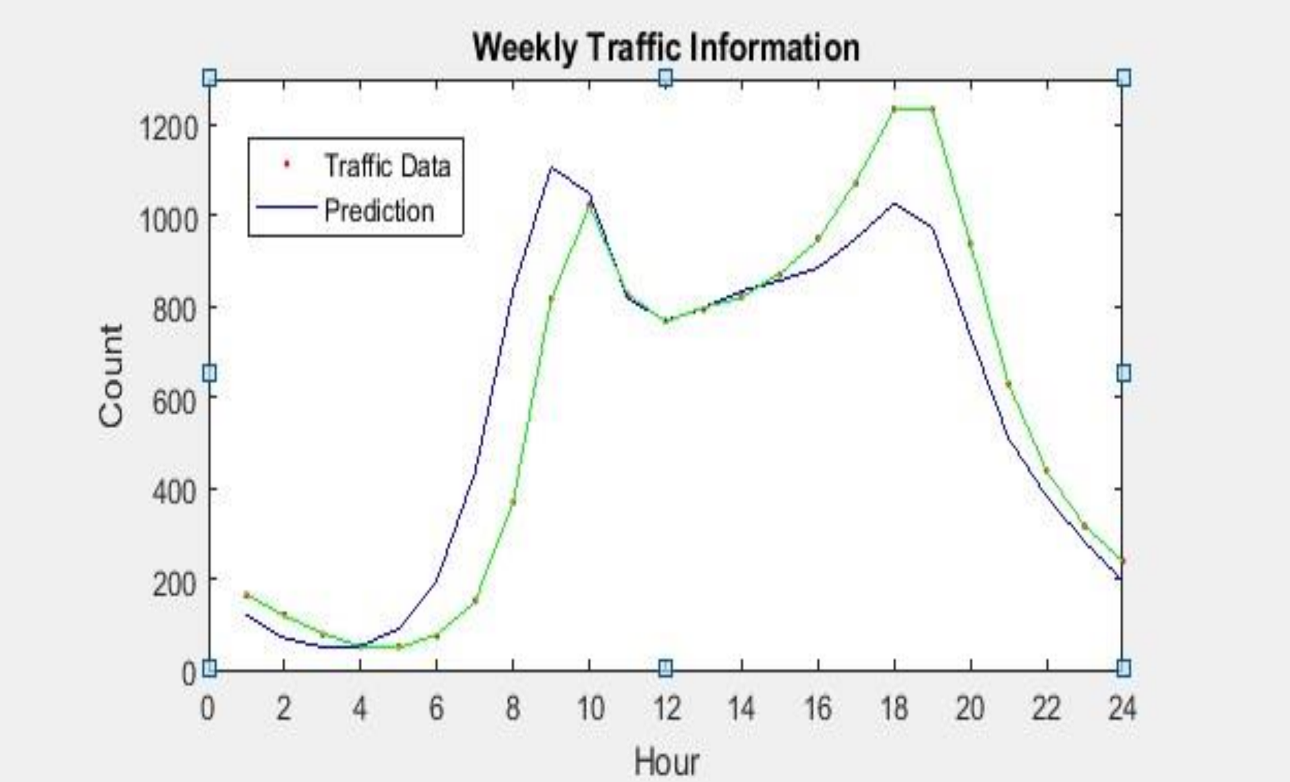

Figure 4.4 Weekday Traffic Data and Prediction

Figure 4.4 represents the weekly traffic profile. It is a representation of all traffic from Monday to Sunday and its corresponding traffic predictions. Ideally, weekly traffic profile is used as the historical data input to the system.

4.4. System Input Variables

To be able to constructively, relate the efficiency of the current and proposed systems, it is important to have a fair measure in place. The two systems in comparison are the same except for the technologies that run them. Table 4.2 shows the inputs to the systems in comparison

Table 4.2 System Input Variables

System	Through traffic factor cars	Count cars	Through traffic factor trucks	Count trucks	Through traffic factor buses	Count bus	loaded persons [#]
Current System	10	15	5	8	6	10	42
Proposed System	10	15	5	8	6	10	42

The Through Traffic Factor is a variable than implies that many vehicles depart and arrive at the boundary of the simulation area, which corresponds to a scenario with a lot of through traffic. The Count constraint outlines how many vehicles are produced for the network per hour and lane-kilometre.

From table 4.2 it is emphatically shown that the input for both systems are the same hence, gives the basis for a fair comparison in analysing the performance of both systems. There are four hundred and thirty-eight vehicles and forty-two vehicles that would be inserted into the simulation from its start to finish.

4.5. System Key Performance Indicators

Another important factor to consider in the comparison of these systems are the type of key performance indicators chosen as shown in table 4.3, which are performance measurements that evaluate the successes or the failures of the systems being measured. Since the aim of the research is to eliminate congestion in the Kumasi Metropolitan, Key Performance Indicators chosen are used to measure or have a direct influence on traffic congestion. The table 4.3 indicates the particular KPIs chosen to evaluate the performance of the systems in comparison.

Table 4.3 System Key Performance Indicators

Key Performance Indicators	Units
Vehicular Speed	[m/s]
Trip Queue Times	[s]
Trip Time Loss	[s]
Trip Duration	[s]

The KPIs chosen are vehicular speed, trip waiting times, trip time loss, and trip duration. All values are achieved from the SUMO simulation are graphically represented for convenience.

4.5.1. Trip Time Loss

The first KPI measure is the trip time loss, which gives an indication of the time lost due to driving below the ideal speeds, ideal speeds includes the individual speed factor which is used to sample a vehicle speed factor form a normal distribution and slowdowns due to intersections will incur time loss. Scheduled stops are not considered in trip time loss measurements. Figure 4.5 is the comparison of the current and proposed systems in terms of their trip time losses. The red signifies the proposed system with the blue representing the current system.

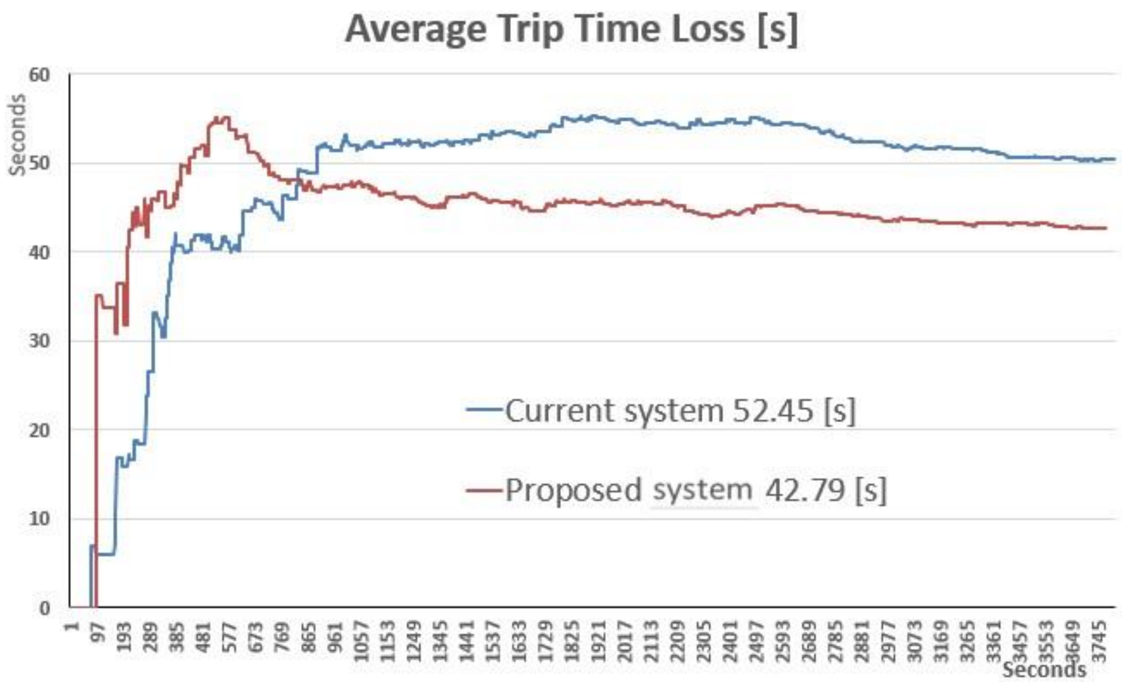

Figure 4.5 Trip Time Loss

From figure 4.5 the vertical axis represents, the trip time loss in seconds and the horizontal axis represents the simulation time in seconds. The proposed system has an average trip time loss value of 42.79 seconds as against the 52.45 seconds with the current system. This indicates that on an average, the proposed system outperforms the current system by 9.66 seconds, which is very significant in traffic management. This means vehicles in the simulation on the average for the proposed system would gain almost ten more seconds on the trip times.

4.5.2. Trip Duration

The trip duration is the time required by an individual to accomplish a particular route. The trip duration is measured in simulation seconds. Figure 4.6 is an assessment of the current and proposed systems in terms of their trip duration. The red signifies the proposed system with the blue expressing the current system.

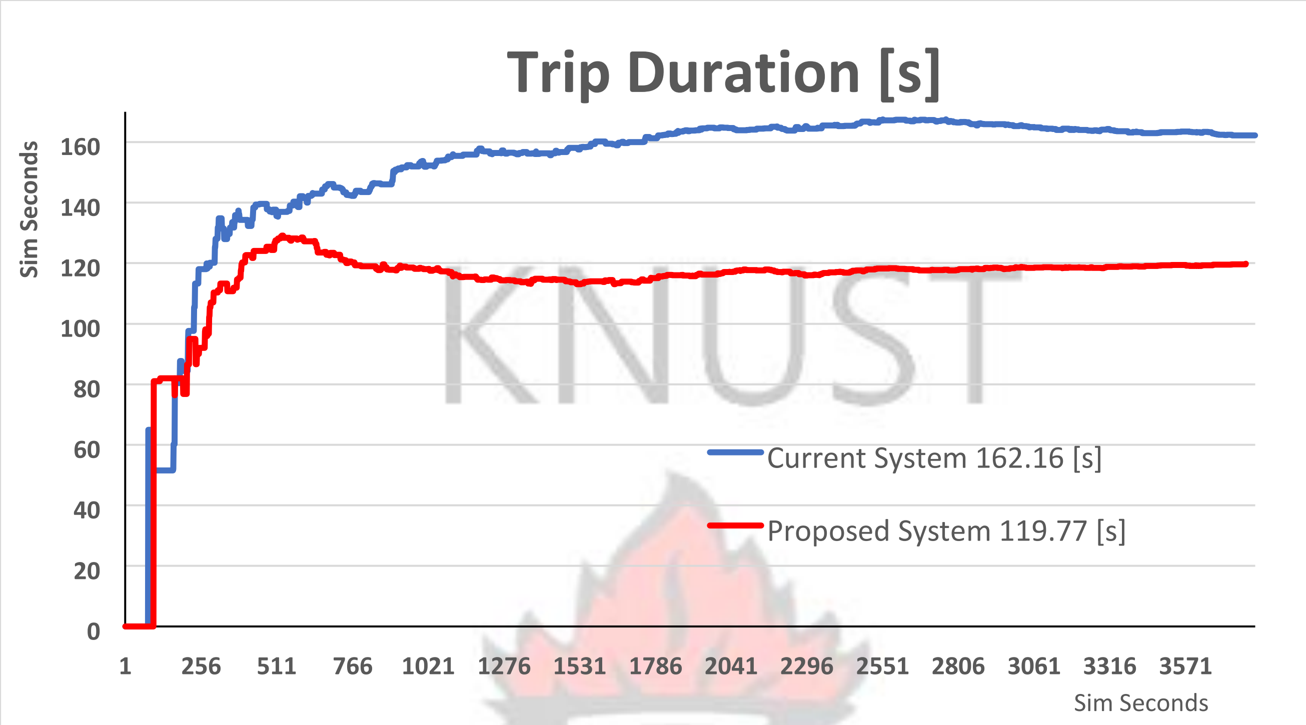

Figure 2.6 Trip Duration

From figure 4.6 the vertical axis represents, the trip duration in seconds and the horizontal axis represents the simulation time in seconds. The proposed system has an average trip duration of 119.77 seconds as against the 162.16 seconds with the current system. This specifies that on an average, the proposed system performed better than the current system by 42.39 seconds, which is very substantial in the management of traffic. This means vehicles in the simulation on the average for the proposed system, would leave the simulation almost 43 seconds faster than in the current system.

4.5.3. Vehicular Speed

This is the measure of the speed with which, a vehicle traverses the network in the simulation. Vehicular speed is computed in meters per simulation seconds. Beneath is an assessment of the current and proposed systems in terms of their vehicular speeds. The red implies the proposed system with the blue shows the current system.

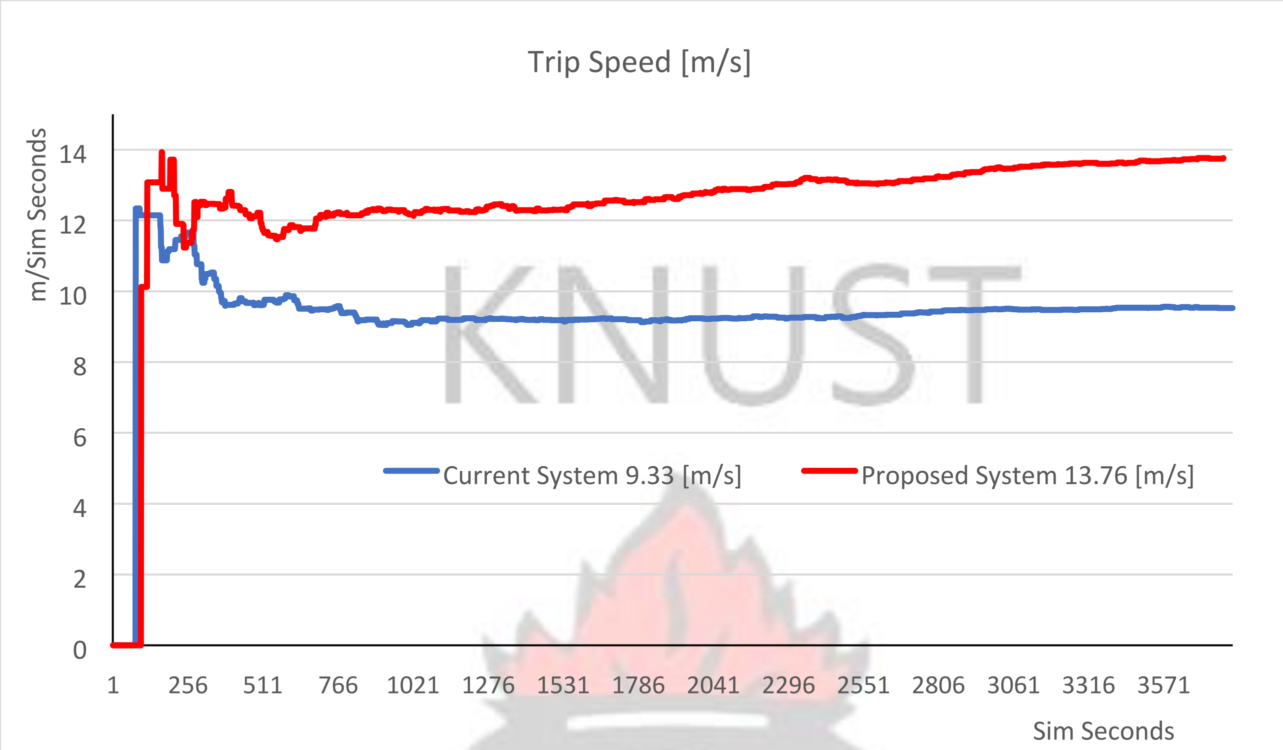

Figure 4.7 Trip Speed

From figure 4.7 the vertical axis represents, the vehicular speeds in meters per simulation seconds and the horizontal axis represents the simulation time in seconds. The proposed system has an average speed of 9.33 meters/second as against the 13.76 meters/seconds with the current system. This specifies that on an average, the proposed system achieved 4.43 m/s better than the current system, which is very important in the management of traffic. This means vehicles in the simulation on the average for the proposed system, would drive at speeds almost 4 m/s seconds quicker than in the current system.

4.5.4. Trip Queue Times

The last KPI evaluated is the trip Queue time, which is the time in which a particular vehicle has to stop and wait, scheduled stops are not considered when measuring the queuing times of vehicles. The trip waiting time, is a direct measure of congestion in the system. Figure 4.8 is the comparison of the current and proposed systems in terms of their trip time losses. The red signifies the proposed system whiles the blue representing the current system.

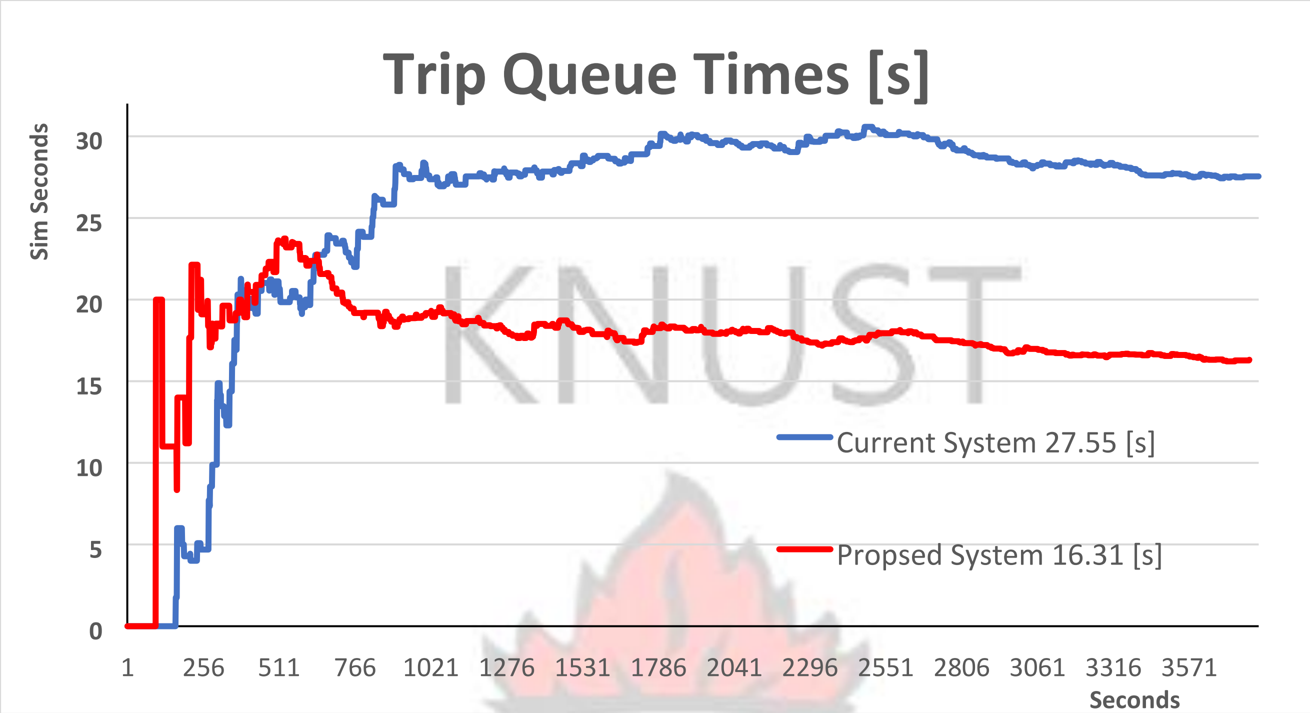

Figure 4.8 Trip Queue Times

From figure 4.8 the vertical axis represents, the trip queue time, in seconds with the horizontal axis represents the simulation time in seconds. The proposed system has an average trip queue time of 16.31 seconds as against the 27.55 seconds with the current system. This specifies that on an average, the proposed system performed better than the current system by 11.24 seconds, which is very substantial in the management of traffic. This means vehicles in the simulation for the proposed system, would spend about 11 seconds less on the average waiting in the current system.

.

Chapter 5
Conclusion

5.1. Introduction

In this research, an intelligent traffic monitoring system that uses historical traffic data to learn and optimize the duty cycles of traffic controllers was designed. The system uses iterative tuning strategy to tune the phase splits iteratively to balance the traffic demands from all directions in a junction. The design when implemented will be adept to manage efficiently the flow of traffic in the Kumasi Metropolis by enhancing the phase and split times of traffic controllers to enable them cope with the erratic behaviour of traffic in the region.

The system was implemented in the Simulation of Urban Mobility environment, and is extensively tested for the training and learning phases of the system design and the overall operation of the traffic management system.

The result obtained in the comparison of the current and proposed systems indicated clearly that the proposed system outperformed the current fixed time cycle controllers in every key performance indicator selected. Moreover, the management of the traffic was drastically improved in the selected region.

5.2. Recommendation

This research is proposing the use of historical traffic data to learn and continuously optimize the duty cycles of traffic controllers. The adoption of the design will enhance the flow of traffic, which is a necessity in the Kumasi metropolis.

It is recommended that future work should establish the relation between pedestrians and traffic control and the effect pedestrians have on the Key Performance Indicators selected.

Future work should also ensure the actual development and testing of the design to prove the systems effectiveness in its performance.

References

[1] H. Strobel, Computer Controlled Urban Transportation. A Survey of Concepts, Methods and International Experiences, vol. 10 of IIASA Int. Series, John Wiley and Sons, 1982

[2] R. Horowitz and P. Varaiya, "Control design of an automated highway system", Proceedings of the IEEE, vol. 88, no. 7, pp. 913-925, 2000.

[3] A. Attar, P. Chaudhary, P. Dhuri and P. Venkatesan, "Intelligent Traffic Management System", Endocrinology, vol. 1, no. 4, pp. 1-3, 2015.

[4] R. Rahmat, "Intelligent Urban Traffic Control System", 2014 Thesis / University of Malaysia.

[5] K. Button and D. Hensher, “Handbook of transport systems and traffic control. Bingley, UK”: Emerald, 2009.

[6] Sensorsportal.com, 2016. [Online]. Available: http://www.sensorsportal.com/HTML/DIGEST/july_08/P_302.pdf. [Accessed: 14-Aug- 2018]

[7] S. Lo and C. Hsu, "Cellular automata simulation for mixed manual and automated control traffic", Mathematical and Computer Modelling Volume 51, no. 78, 2010, p. 100, 2010.

[8] C. Chira, "The Development of a Multi-Agent Design Information Management and Support System", Doctoral dissertation, Galway-Mayo Institute of Technology, Ireland, 2005

[9] J. Vilchez, "Representatively and univocity of traffic signs and their effect on trajectory movement in a driving simulation task: Warning signs", Traffic Injury Prevention, vol. 18, no. 5, pp. 544-549, 2016

[10] Sessions, Gordon M. (1971). Traffic devices: historical aspects thereof. Washington: Institute of Traffic Engineers

[11] C. McShane, "The Origins and Globalization of Traffic Control Signals", Journal of Urban History, vol. 25, no. 3, pp. 379-404, 1999.

[12] "Traffic Lights Sequence – Driving Test Tips", Drivingtesttips.biz, 2018. [Online]. Available: https://www.drivingtesttips.biz/traffic-lights-sequence.html. [Accessed: 19-Mar- 2018].

[13] J. Vilchez, "Representatively and univocity of traffic signs and their effect on trajectory movement in a driving simulation task: Warning signs", Traffic Injury Prevention, vol. 18, no. 5, pp. 544-549, 2016.

[14] J. Cui, A. Allan, M. Taylor and D. Lin, "An Examination of Pedestrian Trip Behaviour in Underground Pedestrian Systems", International Planning Studies, vol. 20, no. 3, pp. 209-227, 2014.

[15] P. Sorensen, Moving Los Angeles. Santa Monica, CA: RAND Corp., 2008.

[16] "The history and evolution of Traffic Lights", Scienceabc.com. [Online]. Available: https://www.scienceabc.com/innovation/ready-steady-go-the-evolution-of-trafficlights.html#The_Computerization_of_Traffic_Lights. [Accessed: 20- Mar- 2018].

[17] NEREM record. New York, N.Y.: Lewis Winner, 1961.

[18] Shafini Halim, Red light running at fixed-time and vehicle actuated traffic signal control system in urban and sub-urban area. 2011 Thesis.

[19] Managing urban traffic congestion. Paris: OECD, 2007.

[20] S. Ossowski, Co-ordination in artificial agent societies, social structures and its implications for autonomous problem-solving agents., No. 1535 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg New York: Springer, 1999.

[21] J. van Leeuwaarden, "Delay Analysis for the Fixed-Cycle Traffic-Light Queue", Transportation Science, vol. 40, no. 2, pp. 189-199, 2006

[22] Assessment of solid-state lighting, phase two. National Academies of Sciences, Engineering, and Medicine.

[23] K. Lim and G. Kim, "Knowledge-Based Expert System in Traffic Signal Control Systems.", S-space.snu.ac.kr, 2018. [Online]. Available: http://svxspace.snu.ac.kr/handle/10371/90495?mode=full. [Accessed: 06- Oct- 2017]

[24] L. Igbinosun and S. Omosigho, "Traffic flow model at fixed control signals with discrete service time distribution", Croatian Operational Research Review, vol. 7, no. 1, pp. 1932, 2016.

[25] Yeh C, Ritchie S.G., Schneider J.B. Potential applications of knowledge-based expert systems in transportation planning and engineering. TRR 1076, 1986. pp.58-65.

[26] Waterman D.A. A guide to expert systems. Addison-Wesley, Reading, Mass. 1986.

[27] P. Athanas and H. Silverman, "Processor reconfiguration through instruction-set metamorphosis", Computer, vol. 26, no. 3, pp. 11-18, 1993.

[28] Ambrosino G., Boero M. Artificial intelligence techniques for urban traffic control. Transport research 24A, London. 1990.

[29] Michael A.P. “Knowledge-based systems for transport network analysis: A fifth generation perspective on transport network problems”. Transport research vol. 24A 1990. pp.3-14

[30] Kirby H.R. and F.O Montgomery. “Toward a rule-based approach for traffic signal design”. Civil Engineering systems Vol. 4, 1987. pp.20-26. U.K.

[31] Findler and Stapp, "Distributed Approach to Optimized Control of Street Traffic Signals", Journal of Transportation Engineering, vol. 118, no. 1, pp. 99-110, 1992.

[32] Zozaya G.C. and Hendrickson C. T. Expert system for traffic signal setting assistance. Journal of transport engineering, Vol. 113(2) 1987. pp.l08-126.

[33] B. Foraste, and G. Scemana, ‘‘Surveillance and congested traffic control in Paris by expert system,’’ in IEEE Int. Conf. on Road Traffic Control, London, pp. 91–94, 1986

[34] C. Nyce, "Predictive Analytics White Paper", American Institute for Chartered Property Casualty Underwriters/Insurance Institute of America, vol. 1, 2007.

[35] T. Mitchell, Machine learning. Boston, Mass.: McGraw-Hill WCB, 2008.

[36] K.Tavladakis, K. Kostalias, A. Dollas, K. Kalaitzakis and N. Voulgaris, "Development of a Microcontroller-Based, Distributed, Adaptive Traffic Control System", IFAC Proceedings Volumes, vol. 30, no. 8, pp. 937-942, 1997.

[37] I. Rechenberg, "Evolution strategy: Nature’s way of optimization. In Bergmann, editor, Methods and Applications, Possibilities and Limitations.” Lecture notes in Engineering, pp. 106– 126, 1989

[38] L. Zadeh, G. Klir and B. Yuan, Fuzzy sets, fuzzy logic, and fuzzy systems. Singapur: World Scientific, 2011.

[39] K. Tan, M. Khalid and R. Yusof, "Intelligent traffic lights control by fuzzy logic.", Malaysian Journal of Computer Science, vol. 9, no. 2, 1996.

[40] J. Lee, K. Lee, K. Seong, C. Kim, and H. Lee-Kwang, "Traffic control of intersection group based on fuzzy logic.", Proceedings of the 6th International Fuzzy Systems Association World Congress, pp. 465–468., 2018.

[41] M. Wiering, J. van Veenen, J. Vreeken and A. Koopman, "Intelligent Traffic Light Control", Intelligent Systems Group, Institute of Information and Computer Sciences, Utrecht University, 2004.

[42] H. Taale, M. Preuß, T. Back, A. Eiben, J. de Graaf and C. Schippers, "Optimizing traffic light controllers by means of evolutionary algorithms.", EUFIT, 1999.

[43] T. Thorpe, "Vehicle traffic light control using sarsa", Master's thesis Department of Computer Science, Colorado State University, vol. 1, no. 1, 1997.

[44] H. Liu, J. Oh and W. Recker, "Adaptive signal control system with on-line performance measure", In 81st Annual Meeting of the Transportation Research Board, 2002.

[45] Dimitri P. Bertsekas. "Dynamic Programming and Optimal Control: Approximate Dynamic Programming, Vol.II", Athena Scientific, 2012.

[46] van Otterlo, M. Wiering, M. (2012). "Reinforcement learning and markov decision processes". Reinforcement Learning. Springer Berlin Heidelberg: 3–42

[47] T. Thorpe and C. Anderson, "Traffic light control using sarsa with three state representations", Technical report, IBM corporation, 1998.

[48] H. C. Manual, "*Highway Capacity Manual*", 4th ed. Washington, DC, 2000.

[49] R.Chrobok, O. Kaumann, J. Wahle and M. Schreckenberg, "Different methods of traffic forecast based on real data", European Journal of Operational Research, vol. 155, no. 3, pp. 558-568, 2004

[50] W. Choi, H. Yoon, K. Kim, I. Chung and S. Lee, "A traffic light controlling FLC considering the traffic congestion.", AFSS 2002, International Conference on Fuzzy Systems, pp. 69–75., 2002.

[51] N. Darroch, G. F. Newell, and R. W. J. Morris. "Queues for a vehicle-actuated traffic light. Operations Research", 12(6):882–895, 1964.