A REAL TIME TRAFFIC SIGNAL CONTROL SYSTEM

A dissertation submitted to the
Department of Electrical Engineering, University of Moratuwa
in partial fulfillment of the requirements for the
degree of Master of Science

by

JAYAKODY ARACHCHIGE NIMANTHI NISHANI KUMUDU JAYAKODY

Supervised by: Dr. Lanka Udawatta and Dr. Sisil Kumarawadu,

**Department of Electrical Engineering
University of Moratuwa, Sri Lanka**

January 2008

DECLARATION

The work submitted in this dissertation is the result of my own investigation, except where otherwise stated.

It has not already been accepted for any degree, and is also not being concurrently submitted for any other degree.

UOM Verified Signature

[illegible]
25.01.2008

We endorse the declaration by the candidate.

UOM Verified Signature

Dr. Lanka Udawatta

UOM Verified Signature

Dr. Sisil Kumarawadu

CONTENTS

Abstract

Traffic congestion problem in Colombo city is getting worse since traditional traffic control system could not fulfill the need. Since the existing system is a fixed time fixed cycle control system, it cannot fit with dynamic traffic environment.

In this research, a decentralized control strategy to control a traffic network grid is presented. Single controller is to control traffic signals of all approaches at one intersection and each approach green time is given by its separate Fuzzy Inference System. Vehicle arrival data are to be collected by lane detectors. Inductive Loop Detectors are proposed for this purpose. Herein, a methodology is developed to decide green time of each approach based on the arrival data by the Fuzzy Inference System and the Cycle time. Influence to the particular intersection is identified and is factorized as an input to the Fuzzy Inference System. Later, the green time is decided by the FIS. Results for this mechanism are shown for one intersection on a simulated environment modeled by Matlab.

Calculations have been done based on the real data obtained for fifteen occasions. Results for three sets of data from both existing fixed time system and the intelligent model have been compared based on the calculations done for the total vehicle delay time, expected at the passing the particular intersection. It shows 51.6% of minimized total vehicle seconds delay by the intelligent traffic control model over the fixed time control system.

To my parents

Acknowledgement

This is to pay my warm gratitude to the Department of Electrical Engineering, University of Moratuwa, SriLanka for giving me the opportunity to read for the M.Sc in Industrial Automation and to do this research during the period of October 2005 to January 2008.

I would like to extend my warm thanks with respect to my research supervisors Dr. Lanka Udawatta and Dr. Sisil Kumarawadu, who have given endless support and guidance while I was doing the research and following the program. Their breadth of knowledge and the advices given me at the presentations made me highly impressed to do the research in depth and to develop myself by enriching academic, research oriented and professional maturity. Their countless advices and the guidance were invaluable.

I would also like to thank Professor Ranjith Perera, the Head of the Department. He has given me support and encouragement, and his advice and feedback about my research at presentations have greatly enhanced and strengthened the study. I thank him for all the time and energy he has paid for my work.

I also wish to extend thanks to my friends and colleagues who were invaluable in completing this study, and to the management of Electro-Serv (pvt) ltd who has given me the support by releasing me from the duty to follow this program.

I am indebted to my parents and to my brother Suranga Jayakody for the guidance they have given to me. I recall their constant support, encouragement, love and guidance given me every time.

Finally, I would like to share my research experience with all of you.

Nimanthi Jayakody
University of Moratuwa, Sri Lanka
January 2008

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

The problem of controlling traffic signals, which is the problem of deciding when to allow vehicles to cross an intersection, has been dealt in many ways. Dramatic progress has been achieved from the traffic guide, who would direct drivers at intersections with a flag to the implementation of fully actuated signal control systems to the present day systems of interconnected signals in corridors. The availability of detectors of many kinds; e.g. loop detection, video detection, and infrared detection, that are capable of transmitting information on the real traffic situation, and the development of microprocessors which can process this data, motivates the development of real-time traffic control systems.

Urban vehicular traffic is an expression of human activity and is highly variable in time and space. Traffic load is highly dependent on parameters such as time, day, season, weather and unpredictable situations. If these parameters are not taken in to account, the traffic control system will create bottlenecks and delays. Therefore, the control of such traffic requires a high degree of adaptability to produce suitable responses to this variability. Existing system (fixed time control system) however, impose certain rigidities that very much restrict the opportunity to get adapted with the real time traffic load.

A traffic intersection is defined by a location in which the sharing of right of way by two or more traffic streams is required. In order to accomplish this sharing, intersection control is used. The traffic signal operates by allotting green time to the intersection approaches by a predefined way. The manner in which green time is allotted to these approaches is categorized as follows [3], [9].

(a) Pretimed or Fixed: The traffic signal provides a fixed amount of green time to each approach during a cycle. This green duration is fixed for each interval for some period of time.

(b) Actuated: The traffic signal provides a minimum length of green time to each approach during a cycle. This length may be incremented based on vehicle arrivals to the approach displaying green as observed by a detection device. The length of every green interval is also constrained by a maximum green time specification.

(c) Adaptive: The traffic signal provides green time to each intersection approach based on anticipated arrivals for a cycle. Generally, as arrivals change from cycle to cycle, the length of green time provided to each approach also changes.

In intelligent traffic control systems reviews, the main terminologies found are split, cycle and offset [29]. The most existing adaptive methods are to optimize these three factors. The cycle time means the total time allowed for all approaches to control the traffic flow at the particular intersection. The split is meant as the green phase or the fractions of the cycle time given to each approach or the green time at the approach. Offset is the delay between two neighboring intersections to initiate each cycle.

Fixed time signal means that controlling the traffic light by predefined timetables [7]. The traffic signal provides a fixed amount of green time to each approach during a cycle. This green duration is fixed for each interval for some period of time. In traffic actuated signal, cycles, signal phases and signal intervals of the traffic light are defined in controllers. Conventional traffic-control strategies have limitations in handling unanticipated traffic demands. In coordinated systems, whether they are pre-timed or coordinated actuated, engineers use historical traffic data to develop a timing plan that would optimize operation of the intersection [9].

An adaptive traffic-signal control on the other hand collects traffic data in real-time, and therefore is expected to mitigate the problem of using outdated data and to improve overall system performance. While an improvement in average delay throughout the system and lower travel time through the corridor are expected to be the main outcomes of an adaptive system, this improvement is achieved at the expense of extra delays for side street users. In network based control systems, the sensors will send monitored traffic data to master controller and according to those values, the master controller calculate and periodically distribute appropriate traffic signal control strategy to the controllers at intersections [19]. In decentralized control system, individual controllers implement its own control strategy.

1.2 Motivation

Today in Srilanka, the traffic congestion in urban area especially in Colombo is an important issue to be solved. One of the reasons of this congestion is imperfect operation of traffic signal control.

In Colombo, it is used fixed cycle, fixed time based control system. Being not get adapted to the actual traffic load, it results for heavy traffic congestion on roads at peak hours, most of the times in morning and in evening when people get rush for work and school. In some junctions, traffic control by a police officer still exists. Therefore, a proper coordination among junctions cannot be expected and it results for increasing the delay at intersections.

The objective of the implementation of an Intelligent Control System is to minimize the average total delay for all vehicles entering the intersection. It provides many advantages over existing fixed-time plans, including minimizing traffic delays by providing effective signal operation, improving travel time, reducing fuel consumption and improving air quality and reducing maintenance and operations cost. The goal of this research is to study and apply traffic control system that operates in real time, adjusting signal timing to accommodate changing traffic patterns. It is not based on a fixed cycle length. By an intelligent traffic control system, the green time is provided to each intersection approach based on anticipated arrivals for a cycle. Generally, as arrivals change from cycle to cycle, the length of green time provided to each approach also changes. In this study, it is proposed a simulation model for a fuzzy controlled decentralized traffic control system to improve existing system and to reduce total delay time.

1.3 Literature review

Over the course of performing a literature review, it was found that over the past 20 to 30 years, research in the area of adaptive signal control has gradually been increasing and practical applications are beginning to be seen around the world. Multi – agent system to urban traffic signal control by Szu-Yin Lin., et al[1] has shown an agent model for a decentralized traffic signal control. Development of an autonomous adaptive traffic control system by K. Tavladaskis[2] shows a network approach.

Researchers have offered the definitions for the parameters and measures of effectiveness in various ways. In this research, the definitions and the parameters given by Szu-Yin Lin, et al have been used.

The world popular traffic control methods are Split Cycle Offset Optimization Technique SCOOT, Sydney Coordinated Adaptive Traffic Control System SCATS, Real time Hierarchical Optimized Distributed and Effective System RHODES and the Optimized Policy for Adaptive Control OPAC. Here SCOOT and SCATS use a cycle based approach on a network while RHODES and OPAC work on the concept of rolling horizon approach. The TRYS approach by Josefa Hernandez[4], and RHODES by Pitu Mirchandani., et al[5] have shown more hierarchical manner to the adaptive control system. . The web page http://www.scoot-utc.com/[14] gives a complete picture of SCOOTS and the places it has been installed. RHODES by Pitu Mirchandani, et al[5] have discussed the control system RHODES. The architecture, which decomposes the traffic control problem to several sub problems, predicts traffic flows at appropriate resolution levels and the optimization modules to solve the hierarchical sub problems while utilizing a data structure for fast solution of the sub problems and the analysis are presented. The web page http://ops.fhwa.dot.gov/publications/adaptivecontrol/ [25] discusses about OPAC and http://www.traffic-tech.com/pdf/scatsbrochure.pdf [30] discusses SCATS control system in detail.

A micro simulation model, SIGSIM for simulating movement of individual vehicles in a signal controlled road network is presented in Adaptive traffic control using evolutionary algorithms by Jenan Sha Aban, et al [6]. Results obtained for mean rates of delay over simulation run is also given.

Josefa Hernandez, et al have discussed the TRYS model, a good way of presenting the traffic control model in Real time traffic Management through knowledge based models: TRYS approach [4]. A set of data bases, each one describing the physical structure of a problematic area, a collection of frames representing demand models of different problem areas and the knowledge base to support control decisions organized as a collection of control frames are included in the TRYS approach.

Michael and Randy Machemehl[3] have shown more experiment oriented delay optimized approach and the delays have been calculated.

1.4 Contributions of the research

This project is investigating a method of adaptive traffic signal control over the real time approach, which will meet the changing traffic needs. It has a decentralized control strategy. It was designed to suit for an intersection in Colombo city. The controller is to be installed at the junction and control strategy is to control the traffic flow at that junction only. Since each junction has its own controller, the need for a central controller is not needed.

Single input single output Fuzzy Inference System is to decide the green time of that particular approach and all four splits are totalized to calculate the cycle time. The opening of each approach at the given cycle is taken place starting from the north approach and is ended at West. The outputs of the controller are energized following that mentioned sequence. A simulation model was developed to meet the above design. Simulation results for a set of data and the comparison with the existing fixed time control system are also provided.

1.5 Organization of the report

In this chapter, a focus of the concept development is introduced. In chapter 2, four most popular adaptive traffic control systems are described from reviewing the previous work carried out in this topic. In chapter 2, the way of detection, the way of getting influence and the effect of detection values to the influence are discussed. Chapter 3 presents the Fuzzy Control Strategy development in the research. The way of control mechanism developed and the way of controller perform are discussed in this chapter. Chapter 4 provides the simulation results for a set of cases and a comparison of the intelligent control model with the existing fixed time control system is also provided.

Chapter 2: Available Adaptive traffic control systems

2.1 Introduction:

Over the course of performing a literature review, it was found that there are four methodologies that stand out from other attempts at adaptive signal control. They are significant due to their relative acceptance in the field, as well as the relative extent of their real- world implementation. In this chapter, the most popular adaptive traffic control systems are discussed.

2.2 Major methodolies:

There are many adaptive systems available but many have been developed specifically for a particular city and whilst they may be available for use by others, they do not have the support and ongoing development opportunities afforded by more commercially available systems. The most widely used among these, and available in a relatively off-the-shelf deployment solution are the British SCOOT (Split Cycle Offset Optimization Technique) [14] and the Australian SCATS (Sydney Coordinated Adaptive Traffic System) systems [28]. In addition to them, RHODES (Real time Hierarchical Optimized Distributed and Effective system) [5] and the OPAC (Optimized Policies for Adaptive Control) [30] are used.

2.3 SCOOT (Split Cycle Offset Optimization Technique):

SCOOT was developed in the early 1980's by the transport research laboratory in the United Kingdom. For the detection method, inductive loops are most common, though other types of detectors can be used. For best results, detectors are required on each link. Installing inductive loops, and maintaining them subsequently, is a significant element in the cost of SCOOT, although less than would be required if all the junctions were operated by isolated Vehicle Actuation. Overhead detectors have been used successfully in some situations [14].

SCOOT differs from SCATS in that it uses a second set of advance vehicle detectors typically 50-300 meters upstream of the stop line. The advance detectors provide a count of the vehicles approaching at each junction. This gives the system a higher resolution picture of traffic flows and a count of the number of vehicles in each queue, several seconds before they touch the stop line. It also provides exceptional queue length detection information to the system, which is triggered when the traffic queue backs up to the upstream detector. Under the SCOOT system green waves can be dynamically delayed on a just in time basis based on the arrival of vehicles at the upstream detector, allowing extra time to the previous green phase where warranted in heavy traffic conditions.

A SCOOT network is divided into "regions", each containing a number of "nodes" (signalled junctions and pedestrian crossings) that all run at the same cycle time to allow co-ordination [14, 25, 27].

Nodes may be "double cycled" (i.e. operate at half of the regional cycle time) at pedestrian crossings or under saturated junctions. Region boundaries are located across links where co-ordination is least critical, e.g. long links. Data on the regions, nodes, stages, links and detectors will need to be stored in the SCOOT database.

When all the equipment has been installed and the network data input into the database, the system will need to be validated. Validation of SCOOT is the process of calibrating the SCOOT traffic model so that it reflects as accurately as possible the actual events on the street network. This is critical, to ensure effective performance of the system. Those parts of the system that have been validated can be operated under SCOOT control whilst further nodes are being validated. Once the system has been validated, the traffic management parameters can be set to manage traffic in line with the authority's strategy [14].

When vehicles pass the detector, SCOOT receives the information and converts the data into its internal units and uses them to construct "Cyclic flow profiles" for each link. Vehicles are modeled down the link at cruise speed and join the back of the queue (if present). During the green, vehicles discharge from the stop line at the validated saturation flow rate.

The data from the model is then used by SCOOT in three optimizers which are continuously adapting three key traffic control parameters - the amount of green for each approach (Split), the time between adjacent signals (Offset) and the time allowed for all approaches to a signaled intersection (Cycle time)[14].

At every junction and for every phase, the split optimizer, a few seconds before the phase change is due to take place, will make a decision as to whether to make the change earlier, later or as due. The optimizer implements the decision, which only affects the phase change time by a few seconds that minimizes the maximum degree of saturation for the approaches to the intersection.

During a predetermined phase in each cycle, and for every junction in the system, the offset optimizer makes a decision to alter, by a fixed amount, all the scheduled change times. The optimizer uses information stored in the cyclic flow profiles and by comparing the sum of performance indices on all the adjacent LINKs for the scheduled offset and the possible changed offsets.

A SCOOT system is split into cycle time regions which have pre-determined minimum and maximum boundaries. The Cycle time Optimizer can vary the cycle time of each region in small intervals at attempt to ensure that the most heavily loaded node in the system is operating at a 90% saturation. If all stop bars are operating at less than 90% then the optimizer will make incremental reductions in cycle time.

These three optimizers are used to continuously adapt these parameters for all intersections in the SCOOT controlled area, minimizing wasted green time at intersections and reducing stops and delays by synchronizing adjacent sets of signals. This means that signal timings evolve as the traffic situation changes without any of the harmful disruption caused by changing fixed time plans on more traditional urban traffic control systems [27].

If one takes the example of a busy arterial road with side roads intersecting it, a SCOOT controlled traffic signal system will seek to maximize the traffic carrying capacity of the arterial road by slotting traffic flows from the side road, far side turn filters, and pedestrian crossing phases into predicted gaps in vehicle flows along the arterial route. The SCOOT system uses upstream vehicle detectors to predict vehicle arrivals at the stop line, in advance of real time. Upstream detectors on the side roads give the system information on queue lengths, which in turn allows it to maximize the green wave along the arterial route, by recovering un-needed green time allocated to non-priority phases. As AM and PM traffic peaks, SCOOT increases the cycle time dramatically to assist traffic flow fluidity along main routes by reducing the incidence of start/stop events, which might otherwise bring traffic to a standstill.

The SCOOT system is used in many other urban areas, including Toronto, San Diego, Anaheim, London and Bangkok.

2.4 SCATS – Sydney Coordinated Adaptive Traffic System:

SCATS is somewhat newer, having being created in the early 1990's by the Roads and Traffic Authority of New South Wales, Austrailia. SCATS is a proven adaptive system that is currently controlling more than 20,000 intersections on six continents [28]. SCATS's ability to improve traffic signal operations and provide benefits to the traveling public has been proven in deployments across the United States [30].

SCATS gathers data on traffic flows in real time at each intersection. This data is fed via the traffic control signal box to a central computer. The computer makes incremental adjustments to traffic light timings based on minute by minute changes in traffic flow at each intersection. SCATS performs a vehicle count at each stop line, and also measures the gap between vehicles as they pass through each junction. As the gap between vehicles increases, the lights are wasting green time and SCATS seeks to reallocate green time to where demand is greatest. The SCATS system is also used in many other urban areas including Hong Kong, Sydney, Melborne and Oakland Country.

Utilizing the Microsoft Windows Operating System, SCATS offers intuitive and efficient system operations and data entry [28]. Using contemporary off-the-shelf servers and workstations, SCATS's scalable architecture is suitable for systems ranging from 10 intersections to over 2,000 intersections. A single server can control up to 250 intersections and additional servers can be added to provide control for more intersections. Its client-server architecture is suitable for conventional office networks, and its low processing requirements allows SCATS to be installed on typical office computers. SCATS provides its users contemporary, non-proprietary controller options including support for Type 170-E, Model 2070, and Eagle M50 series controllers.

SCATS also provide a flexible communications architecture, supporting:

_ Point-to-point communications

_ Point-to-multipoint communications

_ IP-based communications

Its robust communications protocol operates over fiberoptic networks, wireless networks, and twisted pair networks found in many municipalities.

The proposed system has two control levels; strategic, and tactical. The Strategic Level establishes limits on the operation of control at each intersection. The Tactical Level Control resides at the controller, and provides for cycle-by-cycle adjustments within limits set by the Strategic Level [7, 24].

Combined, both control levels will provide three main elements:

- Cycle length calculations for each critical intersection;
- Phase split calculation for each intersection; and
- Determination of offsets between adjacent intersections.

By design, the Cupertino city arterials network to be coordinated is subdivided into small groups, and each group is having a maximum of one critical intersection. A group may be any number of intersections, but is always be composed of at least one intersection. One key function of the *Strategic Control* is to decide when to join two or more groups into one coordinated systems. The desired cycle lengths for individual groups are used for this purpose. When two groups require similar cycle lengths, they could be joined to form one coordinated system for the next control period. The desirable cycle length is calculated for the critical intersection in a group. This occurs through a look-up table based on the minimum delay cycle length calculations by the local controllers. The phase split is then calculated for the critical intersection. This is done by the local controllers to achieve equal degrees of saturation (where detection is available) or look-up tables in the case of minor intersections with limited detection. For non-critical intersections within the group, the phase split is generally determined by the minimum phase length requirements of the non-coordinated phases.

The other main function of the *Strategic Level* control is to select a common cycle length, matching phase sequence, and offsets for various groups and to pass these control constraints to individual controllers.

The controller is having the ability to fine tune splits and offsets on a cycle-by-cycle basis, and to determine desired cycle lengths. Controller software is to provide this functionality.

In order to measure degree of saturation, loops at the stop line will need to be sized to provide a monotonic relationship between time space and speed. This will require a loop array 16 feet long [24]. In simple terms, the time space between vehicles is relatively independent of the size of following vehicles, but directly related to the speed of a vehicle. Within the range of non-congested speeds encountered at traffic signals, a loop array of 16 feet length will give a reasonable estimate of the speed of traffic, and an accurate count. It therefore provides the raw data for the degree of saturation and volume calculations.

2.5 RHODES – Real time Hierarchical Optimized Distributed and Effective System:

RHODES is the newest of these four systems having being produced in the mid 1990's at the University of Arizona at Tucson [5]. RHODES: Architecture, algorithms and analysis by Pitu Mirchandani..,et al [5] gives a detailed description about the system.

The system takes input from the surface street detectors, predicts the future traffic streams at various hierarchical levels of aggregation, both specially and temporally and outputs the signal control settings that respond to these predictions. The optimization criterion can be any that is provided by the jurisdiction using the system but must be based on traffic measures of effectiveness such as average delays, stops, throughput, etc.

At the highest level of RHODES is a "dynamic network loading" model that captures the slow-varying characteristics of traffic. These characteristics pertain to the network geometry (available routes including road closures, construction, etc.) and the typical route selection of travelers. Based on the slow-varying characteristics of the network traffic loads, estimates of the load on each particular link, in terms of vehicles per hour, can be calculated. The load estimates then allow RHODES to allocate "green time" for each different demand pattern and each phase (North-South through movement, North-South left turn, East-West left turn, and so on). These decisions are made at the middle level of the hierarchy, referred to as "network flow control". Traffic flow characteristics at this level are measured in terms of platoons of vehicles and their speeds. Given the approximate green times, the "intersection control" at the third level selects the appropriate phase change epochs based on observed and predicted arrivals of individual vehicles at each intersection.

Essentially, at each level of the hierarchy there is an estimation/prediction component and a control component.

There are three aspects of the RHODES philosophy that make it a viable and effective system to adaptively control traffic signals. First, it recognizes that recent technological advances in communication, control, and computation make it possible to move data quickly from the street to the computing processors, make processing of this data to algorithmically select optimal signal timings fast and allow the flexibility to implement through modern controllers a wide variety of control strategies. Second, RHODES recognizes that there are natural stochastic variations in the traffic flow and therefore one must expect the data to stochastically vary. And third, RHODES proactively responds to these variations by explicitly predicting individual vehicle arrivals, platoon arrivals and traffic flow rates for the three corresponding levels of hierarchies.

The two important issues to predict the traffic flow are the length of the time horizon and the number of prediction points per time horizon, which is called the prediction frequency.

The prediction time horizon provides the real time traffic adaptive signal timing control logic with the ability to plan future signal timing decisions. If the prediction time horizon is short, perhaps several seconds, then the signal timing decisions are restricted. If the predictions are more over a 10 seconds time horizon, the signal timing logic can only make timing decision that extend or shorten the current phase. On the other hand, if the predictions are made over a long horizon, the signal timing decisions can include decisions on phase sequencing and phase durations. The prediction frequency provides information about the distribution of vehicle arrivals over time. If the predictions are made at a frequency of only one prediction for the decision time horizon, then the signal timing logic must assume that the vehicles arrive uniformly during that time period. If the predictions are made more frequently, like every seconds over the prediction horizon, then the signal timing logic will have a more accurate representation of the distribution of vehicle arrivals over time.

RHODES is a hierarchical control system [5] that uses predictive optimization, allowing intersection network levels of control. RHODES includes a main controller, a platoon simulator (APRES-NET), a section optimizer (REALBAND), an individual vehicle simulator (PREDICT) and a local optimizer (COP). The detector requirements for RHODES are fairly flexible. At minimum, RHODES requires upstream detectors for each approach to the intersections in the network. RHODES also can use stop-bar detectors to calibrate saturation flow rates and improve traffic queue estimates.

The PREDICT algorithm in RHODES uses the output of the detectors on the approach of each upstream intersection, together with information on the traffic state and planned phase timings for the upstream signals, to predict future arrivals at the intersection. This approach allows a longer prediction time horizon since the travel distance to the intersection is longer and the delays at the upstream signal are considered. A benefit of this approach is that it includes the effects of the upstream traffic signals in the intersection control optimization problem.

The several parameters are needed to be provided to the PREDICT model. They are the travel times on links (detector to detector) which depends on the link free flow speed and current traffic volumes, the queue discharge rates which also depends on volumes such as queue spillbacks and opposing and cross traffic volumes and the turning probabilities.

Through-traffic queue discharge rates are effected by downstream through-traffic volumes, which can be easily measured. Likewise, left-turn queue discharge rates depend on opposing traffic volumes, and right-turn queue discharge rates depend on cross-traffic in that direction. These three discharge rates are initially given from calculated default functions - functions of traffic volumes, but are then adjusted based on how well they predict remaining queues at the stop-bar presence detectors.

The resolution of traffic at the network flow control level (i.e. level 2 of the RHODES hierarchy) is in platoons. Typically, RHODES will use a 20 - 40 second rolling horizon to predict arrivals and queues at each intersection, based on upstream detector data; at the network flow control level, RHODES will use a 200 - 300 second rolling horizon.

Fixed control strategies are based on a signal-timing plan defined in terms of operating parameters for traditional signal control, namely cycle time, splits, and offsets. These parameters are generally developed based on traffic studies and standard procedures, such as the Highway Capacity Manual, or signal timing software such as TRANSYT and PASSER. The traffic studies result in estimates of traffic conditions, link volumes and turning percentages, for specified time periods. Signal timing parameters are developed for each of these time periods and, typically, implemented on a time-of-day basis with no consideration of current actual traffic conditions. In many cases, even the use of standard procedures for the development of signal timing plans is abandoned and traffic engineers operate in a judgment-based fashion with moderate levels of success. None of these approaches is truly traffic-adaptive or even attempt to actually minimize some measure of traffic performance such as average vehicle delay.

The RHODES approach is to predict both the short-term and the medium term fluctuations of the traffic (in terms of individual vehicle arrivals and platoon movements respectively), and explicitly set phases that maximize a given traffic performance measure. RHODES does not necessarily require a prespecified phase sequence, but since many traffic engineers prefer a pre-specified sequence, RHODES has been developed to allow the traffic engineer to specify a desired sequence. In other word, in the RHODES control strategy, the emphasis shifts from changing timing parameters in reacting to traffic conditions just observed to proactively setting phase durations for predicted traffic conditions.

Each decision has an associated value based on a performance measure such as stops or delay. This value is determined by using the predicted vehicle arrivals, the current and prior decisions, and an imbedded traffic flow model that accounts for estimated queues, startup lost time, queue discharge and arrivals, as well as other traffic dynamics that relate the decision to the performance measure.

The Dynamic Programming is completed when each possible decision for each stage has been evaluated in a forward recursion. Then a backward recursion is used to determine the sequence of phases and phase durations that will result in the lowest value of the performance measure over the optimization horizon.

The decision for the first stage of the optimization is implemented as the desired signal control. Just prior to the end of this first phase, the optimization problem is solved again in a rolling horizon approach. The sequence of phases in the second optimization begins with the current phase which allow for the phase to be terminated early or extended based on the re-evaluation with more recent observations and predictions.

Platoons are defined from observed detector data as a flow density above a pre-specified level for some length of time [21]. Each platoon is characterized in terms of size (number of vehicles) and speed. When two (or more) platoons are predicted to arrive at an intersection and they request opposing signal phases, a conflict is said to occur. A decision tree is built where each branch of the tree represents one possible resolution of a conflict. The decision tree developed is based on the predicted platoon movement over some predefined horizon, such as 200-300 seconds, with node and two out-links for each conflict resolution. REALBAND evaluates, using APRES-NET, the performance for each branch of the decision tree. When all branches have been explored, a path on the tree (corresponding to a set of conflict resolutions) is chosen with best estimated performance.

The article, Manage traffic via innovative signal control in the web page of http://ops.fhwa.dot.gov/publications/adaptivecontrol/ [25] gives results oriented analysis of RHODES and OPAC.

2.6 OPAC – Optimized Poicies for Adaptive Control:

The central goal of signal control is to optimize vehicle flow through a given road network. In most cases, the most powerful and cost-effective approach is to improve the timing of the traffic signals at all intersections in the network. Adaptive signal control uses very recent data to adjust signal timing “on the fly” based on close to real-time traffic conditions. Software systems like MIST (Management Information System for Transportation) and OPAC (Optimized Policies for Adaptive Control) have been used successfully by municipalities, state DOTs, and others for this purpose. OPAC is a system first proposed by Nathan Gartner at the University of Massachusets at Lowell in the early 1980’s [25, 30].

OPAC is a traffic adaptive signal control algorithm that provides for continuous online optimization of phase duration, cycle length, and offsets in immediate response to real-time traffic flows. OPAC eliminates the need to develop and store timing plans for different traffic conditions while controlling both isolated intersections and networks of intersections in coordinated operation.

OPAC uses a predictive type of optimization with a rolling horizon [25]. This congestion control strategy—which attempts to maximize throughput—adjusts splits, offsets, and cycle length but maintains the specified phase order. For uncongested networks, OPAC uses a local level of control at the intersection to determine the phase online and a network level of control for synchronization, which is provided either by fixed-time plans (obtained offline), or a virtual cycle (determined online).

The types of control and levels of local and global influence are flexible. Predictions are based on detectors located approximately 10 to 15 seconds upstream. After the initial 10 to 15 seconds, a model predicts traffic patterns. System monitoring and coordinated control features are provided through MIST, which communicates with individual intersection controllers and monitoring devices, gathering data to feed the OPAC algorithm and transmitting signal timing instructions rendered by OPAC.

The implementation of an adaptive control system such as OPAC provides many advantages over existing fixed-timed plans by minimizing traffic delays through effective signal operation, improving travel time, reducing fuel consumption, and improving air quality while simultaneously reducing maintenance and operation costs and facilitating automated traffic data collection.

Several U.S. cities, including Chicago and Seattle, already have such sophisticated computer-guided traffic management systems in place; Cary, N.C., and Sarasota, Fla., are among those currently implementing such systems; and other cities around the nation are following suit [25, 18].

SCOOT and SCATS generally use a cycle based approach on a network, adjusting the cycle time, split and offsets among cycles in the network to optimize a measure of effectiveness [3, 18]. OPAC and RHODES vary somewhat from this, with OPAC being cycle based and RHODES optimizes a measure of effectiveness over a fixed prediction horizon and then extends the horizon by a fixed time step and reiterates the optimization until an optimal split of the given cycle is found. Advancements in approaches to OPAC have allowed for some variability in network wide cycle lengths [18].

The main objective of the research is to make a simulation model of an adaptive traffic control system, which is better to implement in Colombo city to minimize delay caused by existing fixed time traffic control system. The detection method for the proposed system and the design methodologies are described by the next chapter.

Chapter 3: Decentralized Intelligent Control Model

3.1 Introduction:

The essential feature of the real time traffic control system is its capability to identify the real time traffic load on the road. And the key factor to be sensed in the control strategy is the influence to the approach. After getting the vehicle counts from each detector, the influence is to be known to decide the respective split or green phase. In this chapter, function of the detection, the surrounding traffic data collection methods and the inductive loop detectors, which are selected for this research, are discussed. Later, the way of getting influence and the effect of detection values to the influence are discussed.

3.2 Detection in adaptive Control:

3.2.1Function of detection:

In the adaptive traffic control process in its current state, field detection serves to obtain the arrival times for vehicles approaching an intersection. In the field, this arrival time is difficult to measure. In the simplest case one detector is placed and the placement it installed gives its own issues {9].

When the detector is installed at the stop line, the measurement of the arrival time at the intersection of vehicles moving through a green phase, but becomes problematic for vehicles arriving during a stop phase, since only a small number of vehicles can sit on a detector and have their arrival times accurately recorded. Vehicles that proceed after the red signal has turned green will not have their desired arrival times recorded. When the detector is installed at a distance upstream of the stop line, allows determination of the arrival time of a vehicle continuously moving at its desired speed at the detector and through the intersection, but may problematic for vehicles in other situations, generally those that are required to change the speed between the detector and the intersection, such as to stop when encountering a signal state change, to slow to make a turn, to avoid an incident or to perform a passing maneuver.

The preferable approach of the two is to place the detector upstream of the stop line as this will allow at least some determination of the arrival time of a vehicle in any situation.

3.2.2 Detection methods:

Detection technologies should involve the following factors [3, 4, 14].

- Cost: Utilization of as much of the existing detector equipment as possible should be considered. If the replacement of that equipment required, it should be low cost in terms of both installation and maintenance.
- Practicality: Consideration should include the ability to place the desired detectors where they are needed, or whether issues such as proximity of intersections in a network, weather conditions or physical obstacles may hinder placement. Also to be considered is the conformity of the detector system in a network that is detector types and configuration should be as uniform as possible throughout a network.

- Accuracy: The ability of the detector technology to provide data that reflect actual field conditions is vital. Sensitivity, ability to deal with various conditions and deterioration of data accuracy over time are issues to reflect.

The most common detector technology in world is the inductive loop detector, in which generates a current as a vehicle passes over it due to the vehicle's magnetic influence on the inductive loop [29].

Advanced detection technology may allow. easier implementation of the proposed detection strategies. This is especially true in the case of multi-lane urban network applications where incidents erratic driver behaviour and transit use make the ability to identify the specific vehicles important to gathering data. Video detection is such a technology. But video detection has some drawbacks. That is physical placement may be difficult because of availability of mounting structures, blocking by obstacles and angles required for proper detection, and they may also not operate as effectively at night or during obscuring weather conditions.

3.2.3 Inductive loop detectors:

Considering the main advantages of cost effectiveness and the reliability, Inductive loop detectors have been selected as the detection method.

The inductive loop consists of a multi-winding wire coil installed in the floor and makes up the actual sensor element. The system for the contact-free detection of metallic objects is formed by the inductive loop and the switching unit, referred to as loop [2].

The inductive loop and a capacitor which is integrated in the loop detector form an LC oscillator. The frequency of resonance of this resonant circuit is determined by the capacitance of the capacitor and the magnitude of the loop inductance. The capacitance of the capacitor and thus the resonance frequency can be modified using an internal switch. This prevents interference between two adjacent inductive loops or detectors (Figure. 3.1).

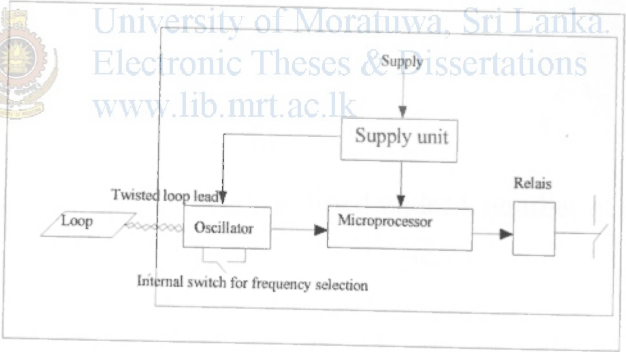

Figure 3.1-Block diagram of the Inductive Loop

The general rule is that the higher the inductance, the lower the frequency of resonance.

If a current is present in the loop, an alternating magnetic field is generated around it. The magnetic field lines get closed to form loops. When a vehicle moves onto the inductive loop, currents are induced by the alternating field of the loop in the metallic parts of the vehicle (chassis, axles). A magnetic field is generated by these currents,

which opposes the loop field. This results in a change in the loop inductance and thus the frequency. This change in frequency is detected by the loop detector.
Depending on the detector type, the output relay is either energized or released when a predefined switching threshold is exceeded. This switching threshold can be adjusted using the sensitivity switch (High – Medium – Low).

«High» is meant when the switching threshold is low and «Low» is meant when the threshold is high. The sensitivity should always be adapted to the respective area of application. The output relay is not switched on by slow frequency changes (e.g. due to temperature variations).

In general, implementation of the inductive loop detectors is to enhance detection process, which is an essential thing in influence calculation to determine the optimum split.

3.3 Influence Detection:

3.3.1 Structure of the intelligent control:

The intelligent control mechanism can be defined under a number of models. They are simply the detection model as discussed earlier, the prediction model and the Decision model as discussed below [3, 11, 12].

•**Detection**: This is the process by which vehicles that enter a given approach to an intersection are recognized for processing by the adaptive control scheme. In this research, inductive loop detectors are recommended to use considering the cost effectiveness and the reliability in use.

•**Prediction**: This is the process by which the data from the detection process are used to determine the influence to the respective approach at the intersection. This has to be done considering the vehicle counts over the cycle time to predict the influence to the given approach in next cycle.

•**Decision:** This is the process by which the predicted influences are used to decide green times to the various approaches of the intersection. In this research, this green time is decided by a Fuzzy inference system.

3.3.2 Formulating the Influence function for traffic signal control:

The prediction model is for the purpose of forecasting possible traffic flow in a particular intersection. As shown in figure 3.2, it is noticed that the traffic flow of intersection A is influenced by vehicles passing through nearby intersections. The influence of nearby intersections to intersection A is inversely proportional to the distance between these intersections [1]. In other words, the far the two intersections are, the less the influence is.

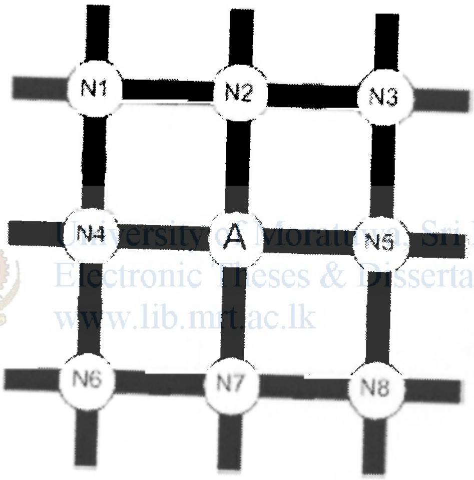

Figure 3.2- part of the traffic network

As same as the distance, the number of vehicles enter to the particular approach also add a considerable effect to the influence. It is noticed that the vehicle count is direct proportional to the influence. Hence the influence can be expressed in the following way [1].

The parameter INFLUENCE α No. of vehicles entering the phase

α 1/distance

INFLUENCE = 1/ distance * car queue. (1)

According to the equation (1), the influence is calculated and be input to the fuzzy inference system to decide the green phase or split. The factor influence for fuzzy inference system stands in between the range of zero to the maximum of one. Hence for each intersection, it has to decide separately for all approaches and taken as a factor to the maximum influence of the same intersection.

The considered intersection A as shown in figure 3.2 is detailed and shown in figure 3.3. As shown there, the considered intersection is 300 m from West, 1 Km from North, 500 m from East and 1 Km from South neighboring intersections. Inductive Loop detectors are to be installed as shown. The relay outputs of each detector are the inputs to the intersection controller.

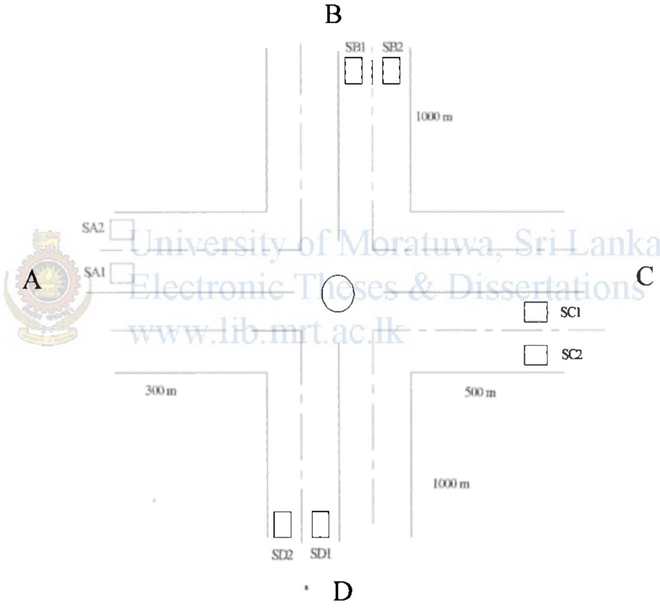

Figure 3.3-Intersection A

S_{A1} and S_{A2} detectors are to detect the vehicles entering to the West Approach A. Likewise S_{B1}, S_{B2} are to detect the vehicles entering North Approach B, S_{C1}, S_{C2} are to detect East Approach C count and S_{D1}, S_{D2} are to detect the South Approach D vehicle counts. The total count will reach the intersection at the average speed of 50 Km/h. If the average vehicle length is 3 m, the maximum number of vehicles which can pass across the intersection is experimented as 275 vehicles during 60 secs time duration when one approach is given its green phase. The maximum influence is given when the vehicle count exceeds 275 in West approach A, which is having the lowest distance to the mentioned intersection. Hence the calculated influence is taken as a factor, divided by this maximum influence.

The following assumptions were made for the formulation of each approach influence.

- 85% of phase vehicle count at the intersection is to enter to the front phase
- 15% of Left track vehicle count is to left turn
- 15% of Right track vehicle count is to Right turn

Basic formula for the influence for the particular approach is given by the equation (2)

$$\textbf{INFLUENCE} = \textbf{(0.85 x direct phase Influence) + (0.15 x Left phase Influence)} \\ \textbf{+ (0.15 x Right phase Influence)} \qquad \textbf{(2)}$$

Hence the Influence at each approach is as follows where d_A, d_B, d_C and d_D stand for the distance from neighboring intersections starting from west to south to the particular intersection.

- INFLUENCE A = $1/d_C$ x0.85 $(S_{C1} + S_{C2}) + 1/d_B$ x0.15 S_{B1} + $1/d_D$ x 0.15 S_{D2} (3)

- INFLUENCE B = $1/d_D$ x0.85 $(S_{D1} + S_{D2}) + 1/d_A$ x0.15 S_{A2} + $1/d_C$ x 0.15 S_{C1} (4)

- INFLUENCE C = $1/d_A$ x0.85 $(S_{A1} + S_{A2}) + 1/d_D$ x0.15 S_{D1} + $1/d_B$ x 0.15 S_{B2} (5)

- INFLUENCE D = $1/d_B$ x0.85 $(S_{B1} + S_{B2}) + 1/d_A$ x0.15 S_{A1} + $1/d_C$ x 0.15 S_{C2} (6)

Getting those influences into the account, the green phase for each approach is decided by the Fuzzy Inference System as discussed by the next chapter.

Chapter 4: Fuzzy Control Strategy Development

4.1 Introduction:

Getting the influence described in last chapter for the input of the Fuzzy Inference System, the Split is decided. Fuzzy sets of the input and the output, Fuzzy rule development and the way of deciding the split according to the changing traffic load are described in this chapter. Detectors give their values periodically. Getting those values in to account, the influence is calculated by the intersection controller and those influence values are taken for the input to the fuzzy inference system. FIS determines the green phase for each approach. Starting from detector value count to the output phase light energizing, the controller has to act as the brain of the intersection control system. The way of control mechanism developed and the way of controller behave accordingly are discussed in this chapter.

4.2 Fuzzy Inference System

4.2.1 Fuzzy basics:

The concept of Fuzzy Logic (FL) was conceived by Lotfi Zadeh, a professor at the University of California at Berkley, and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership. This approach to set theory was not applied to control systems until the 70's due to insufficient small-computer capability prior to that time [8].

Fuzzy Logic incorporates a simple, rule-based IF X AND Y THEN Z approach to a solving control problem rather than attempting to model a system mathematically. The Fuzzy Logic model is empirically-based, relying on an operator's experience rather than their technical understanding of the system.

Fuzzy Logic requires some numerical parameters in order to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless very responsive performance is required in which case empirical tuning would determine them.

Fuzzy Logic was conceived as a better method for sorting and handling data but has proven to be an excellent choice for many control system applications since it mimics human control logic. It can be built into anything from small, hand-held products to large computerized process control systems. It uses an imprecise but very descriptive language to deal with input data more like a human operator. It is very robust and forgiving of operator and data input and often works when first implemented with little or no tuning. Fuzzy inference systems have been successfully applied in fields such as automatic control, data classification, decision analysis, expert systems, and computer vision. Because of its multidisciplinary nature, fuzzy inference systems are associated with a number of names, such as fuzzy-rule-based systems, fuzzy expert systems, fuzzy modeling, fuzzy associative memory, fuzzy logic controllers, and simply (and ambiguously) fuzzy systems. The design of a fuzzy inference system includes fuzzification of inputs and outputs with a knowledge base that contains the membership functions and the rule sets and the defuzification method to get the reasoned output values. In this research, a single input single output fuzzy inference system is used to decide the split for each approach. Four fuzzy inference systems are to be implemented in one intersection controller to decide each green phase for four approaches at the intersection.

4.2.2 Fuzzification:

Mamdani's fuzzy inference method [22] is the most commonly seen fuzzy methodology. Mamdani's method was among the first control systems built using fuzzy set theory. The fuzzification comprises the process of transforming crisp values into grades of membership for linguistic terms of fuzzy sets. The membership function is used to associate a grade to each linguistic term.

Influence is taken as the fuzzy input. Since it has been set as a factor compared to the maximum possible influence, it gets a value in between zero and one. Fuzzification is the process by which system input is evaluated to determine the degree at which it belong to a particular fuzzy set, on a scale from 0 to 1. The system input has been categorized into five fuzzy sets as mentioned in Table 4.1.

A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse, a fancy name for a simple concept.

If X is the universe of discourse which is the influence which stands in between 0 to 1 and its elements are denoted by x, then a fuzzy set A in X is defined as a set of ordered pairs [22].

$A = \{x, \mu_A(x) \mid x \in X\}$

The membership function $\mu_A(x)$ of the input influence categorization is taken as trapezoidal membership function, which has a flat top and really is just a truncated triangular curve. These straight line membership functions have the advantage of simplicity [8].

A	$\mu_A(x)$
Very Low	0 – 0.2
Low	0 – 0.6
Average	0.2 – 0.75
High	0.5 – 1
Very High	0.8 - 1

Table 4.1- Membership values of the Input Influence

In Figure 4.1, the membership functions with their values are graphically represented. The y-axis in Figure 4.1 represents the degree of truth from completely false (0) to completely true (1). The x-axis represents the range of input values for the particular system input.

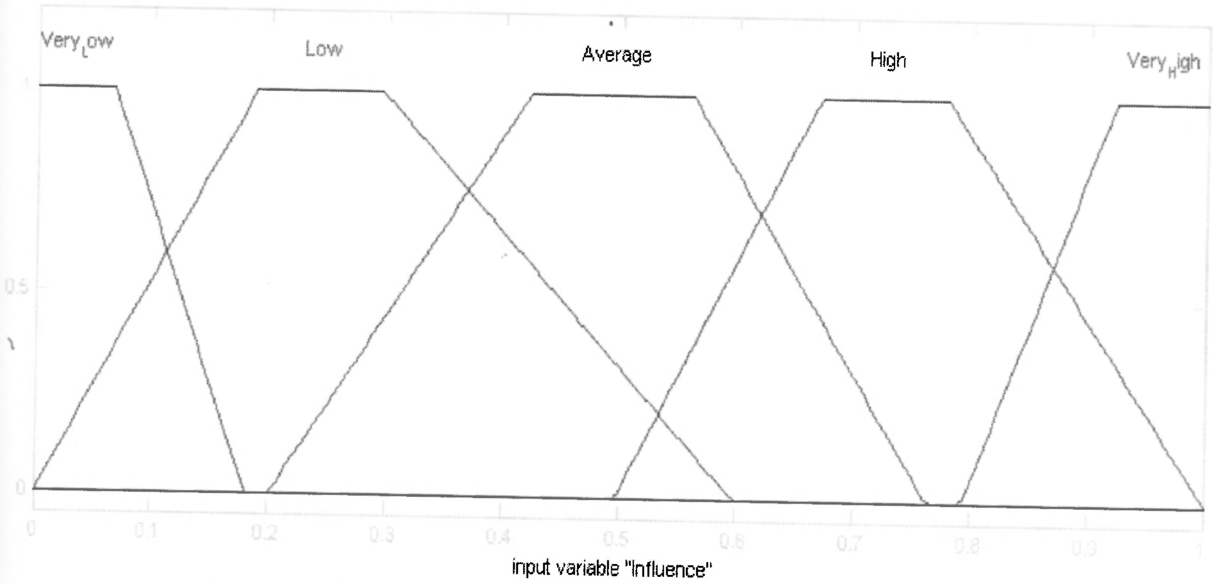

Figure 4.1- Trapezoidal Membership Function for the Input Influence

The membership function $\mu_A(x)$ of the output Split categorization is taken as triangular membership function, which has a sharp top.

A	$\mu_A(x)$ (Sec.s)
Very short	0 – 9
Short	10 – 20
Average	25 – 35
Long	45 – 55
Very long	55 - 60

Table 4.2- Membership values of the Output Split

In Figure 4.2, the membership functions with their values for the output Split are graphically represented. The y-axis in Figure 4.2 represents the degree of truth from completely false (0) to completely true (1). The x-axis represents the range of output values for the particular system output 0 to the maximum of 60 secs..

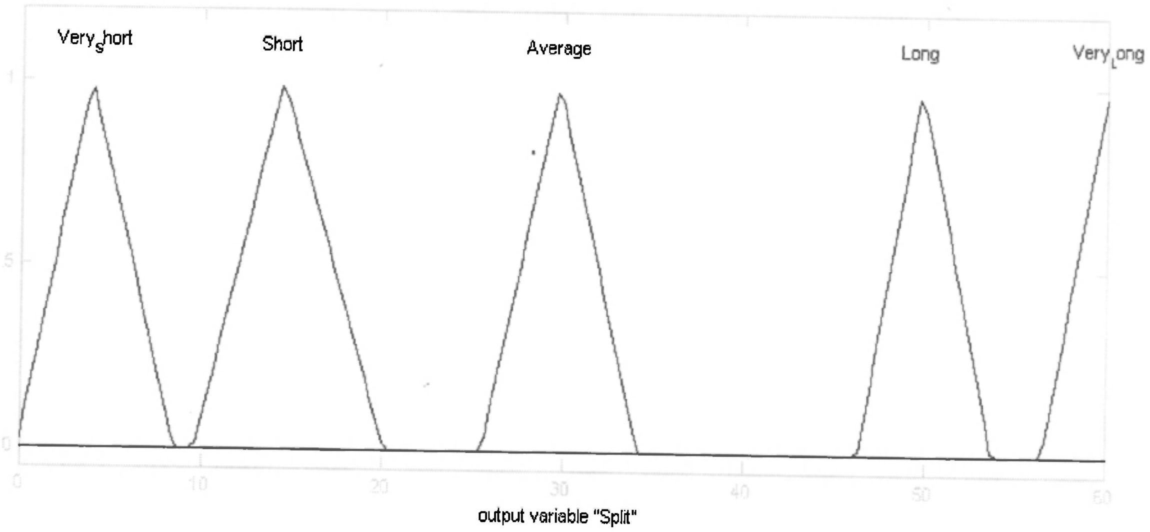

Figure 4.2 -Triangular Membership Function for the Output Split

The membership function is a graphical representation of the magnitude of participation of each input [8]. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp output which drives the system. There are different membership functions associated with each input and output response. The membership functions associate a weighting factor with values of each input and the effective rules. These weighting factors determine the degree of influence or degree of membership (DOM) each active rule has. By computing the logical product of the membership weights for each active rule, a set of fuzzy output response magnitudes are produced. All that remains is to combine and defuzzify these output responses.

Figure 4.3 represents the variation of Split according to the given influence. It can be noticed that with the increase of influence to the particular approach, the Split also increases accordingly.

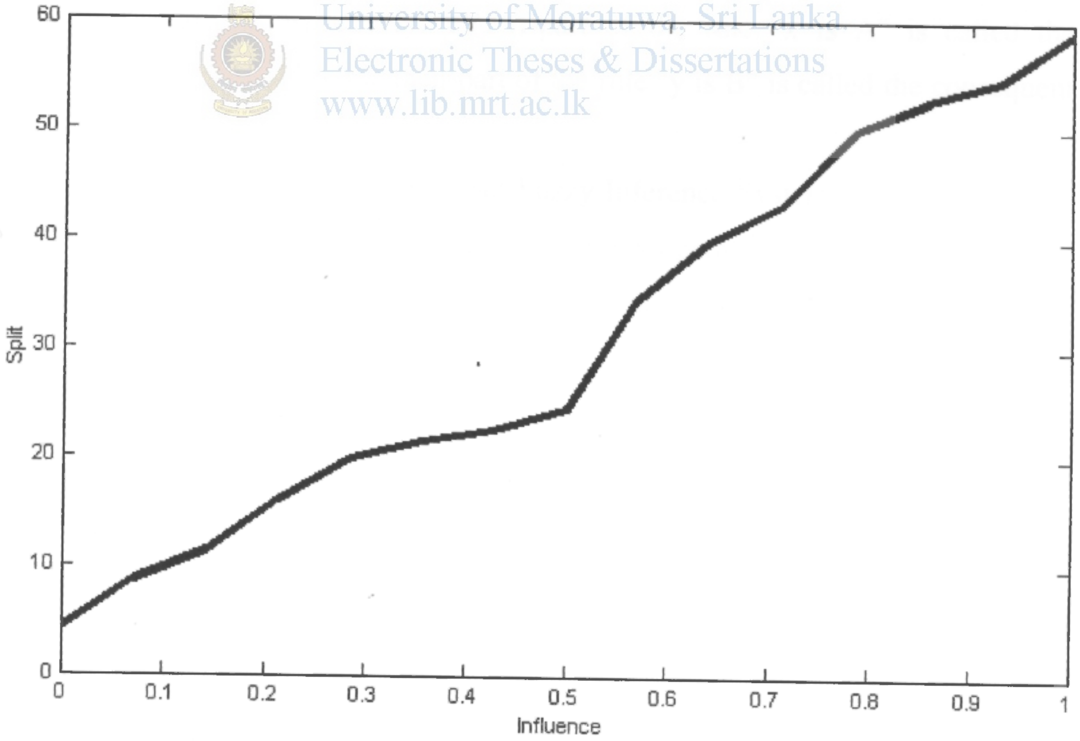

Figure 4.3 -Variation of Split Vs. Influence

4.2.3 Rule base:

As inputs are received by the system, the rulebase is evaluated. The antecedent (IF X AND Y) blocks test the inputs and produce conclusions. The consequent (THEN Z) blocks of some rules are satisfied while others are not. The conclusions are combined to form logical sums. These conclusions feed into the inference process where each response output member function's firing strength (0 to 1) is determined. A firing strength for each output membership function is computed. All that remains is to combine these logical sums in a defuzzification process to produce the crisp output.

Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic. These if-then rule statements are used to formulate the conditional statements that comprise fuzzy logic. A single fuzzy if-then rule assumes the form [22]

if x is A then y is B

where A and B are linguistic values defined by fuzzy sets on the ranges (universes of discourse) X and Y, respectively. The if-part of the rule "x is A" is called the antecedent or premise, while the then-part of the rule "y is B" is called the consequent or conclusion.

In this research single input single output Fuzzy Inference System is used and hence simple five rules were developed. Therefore complex operators are not required to get the output value determined. Those rules are mentioned below.

If influence is VERY LOW, then split is VERY SHORT
If influence is LOW, then split is SHORT
If influence is AVERAGE, then split is AVERAGE
If influence is HIGH, then split is LONG
If influence is VERY HIGH, then split is VERY LONG

Rule evaluation is how Fuzzy Logic performs calculations. The fuzzy values produced membership functions are passed through the rule list to find the fuzzy output. The consequent specifies a fuzzy set be assigned to the output. The implication function then modifies that fuzzy set to the degree specified by the antecedent. The most common ways to modify the output fuzzy set are truncation using the min function (where the fuzzy set is "chopped off") or scaling using the prod function (where the output fuzzy set is "squashed"). Both are supported by the Fuzzy Logic Toolbox. Figure 4.4 represents the graphical representation of rule evaluation to a particular input value.

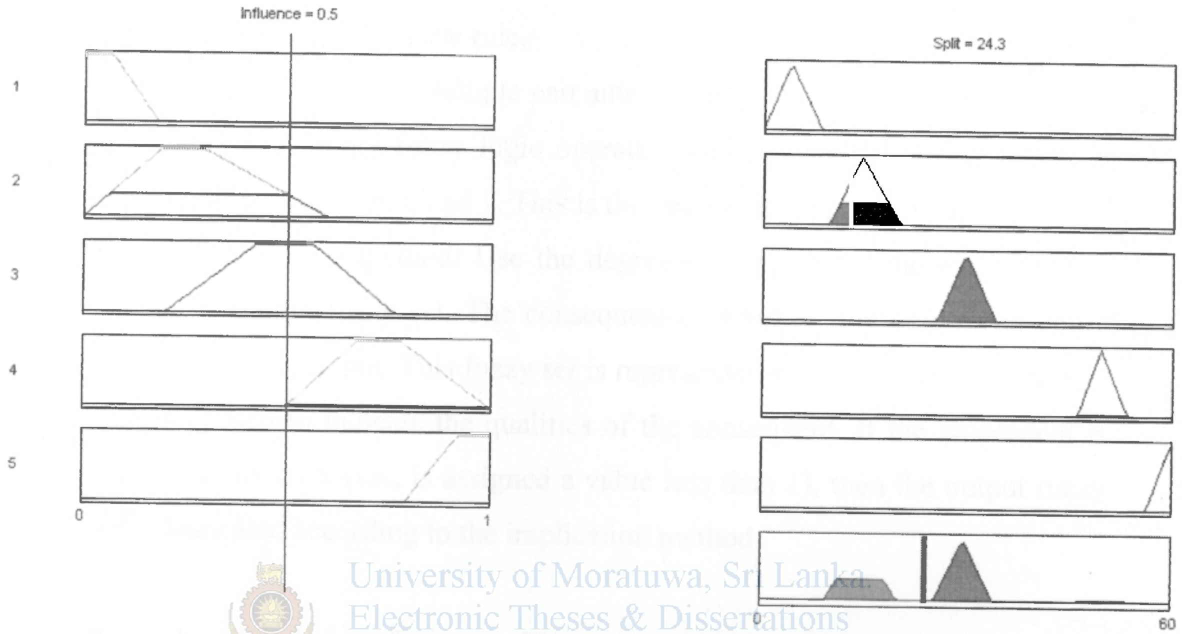

Figure 4.4- Graphical representation of rule evaluation.

In Figure 4.4, the way of split determination to the input value of the influence at 0.5 is shown. When the influence is at 0.5, it gives a membership value of zero to the membership function VERY LOW. In the same way, it gets 0.45 of membership value from the membership function LOW, the maximum of "completely true" membership value 1 from the membership function AVERAGE and the membership value 0 from both membership functions HIGH and VERY HIGH. That grading is shown at the left hand side of the diagram. At the right hand side of the diagram, the split determination is shown according to each rule. At the bottom of the right hand side, it is shown the final determination of the split according to the averaging technique. After getting each split with reference to all of the rules mentioned above, they are averaged to determine the final best fit value for the output split. In this example it was determined as 24.3 seconds to be the split for 0.5 of the influence.

Interpreting if-then rules is a three-part process. This process is explained in detail below [8, 22].

1. Fuzzify inputs: Resolve all fuzzy statements in the antecedent to a degree of membership between 0 and 1. If there is only one part to the antecedent, this is the degree of support for the rule.
2. Apply fuzzy operator to multiple part antecedents: If there are multiple parts to the antecedent, apply fuzzy logic operators and resolve the antecedent to a single number between 0 and 1. This is the degree of support for the rule.
3. Apply \implication method: Use the degree of support for the entire rule to shape the output fuzzy set. The consequent of a fuzzy rule assigns an entire fuzzy set to the output. This fuzzy set is represented by a membership function that is chosen to indicate the qualities of the consequent. If the antecedent is only partially true, (i.e., is assigned a value less than 1), then the output fuzzy set is truncated according to the implication method.

The output of each rule is a fuzzy set. The output fuzzy sets for each rule are then aggregated into a single output fuzzy set. The logical product of each rule is inferred to arrive at a combined magnitude for each output membership function. This can be done by max-min, max-dot, averaging, RSS, or other methods. Once inferred, the magnitudes are mapped into their respective output membership functions, delineating all or part of them. The "fuzzy centroid" of the composite area of the member functions is computed and the final result taken as the crisp output. Tuning the system amounts to "tweaking" the rules and membership function definition parameters to achieve acceptable system response. Finally the resulting set is defuzzified, or resolved to a single number.

Since decisions are based on the testing of all of the rules in an FIS, the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable, just prior to the final step, defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable.

4.2.4 Determination of the single output:

The final step in the Fuzzy Logic calculation is defuzzification, when the raw fuzzy outputs are evaluated to create a composite system output.

Mamdani-type inference, which has been defined in the Fuzzy Logic Toolbox, expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification. It's possible, and in many cases much more efficient, to use a single spike as the output membership function rather than a distributed fuzzy set. Hence a triangular shaped spike has been used as the membership function for the split as shown in Figure 4.2. This is sometimes known as a singleton output membership function, and it can be thought of as a pre-defuzzified fuzzy set. It enhances the efficiency of the defuzzification process because it greatly simplifies the computation required by the more general Mamdani method, which finds the centroid of a two-dimensional function. Rather than integrating across the two-dimensional function to find the centroid, we use the weighted average of a few data points.

In the Fuzzy Logic Toolbox, there are five parts of the fuzzy inference process: fuzzification of the input variables, application of the fuzzy operator (AND or OR) in the antecedent, implication from the antecedent to the consequent, aggregation of the consequents across the rules, and defuzzification.

Once proper weighting has been assigned to each rule, the implication method is implemented. A consequent is a fuzzy set represented by a membership function, which weights appropriately the linguistic characteristics that are attributed to it. The consequent is reshaped using a function associated with the antecedent (a single number). The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set. Implication is implemented for each rule.

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set. Perhaps the most popular defuzzification method is the centroid calculation, which returns the center of area under the curve.

As shown in Figure 4.2, the split for the influence at 0.5 is determined in this way and the single defuzzified output split has been determined as 24.3 seconds.

Four Fuzzy Inference Systems are implemented to decide each four splits at the intersection and the control strategy to control the changing traffic load at real time is discussed below.

4.3 Control Strategy

4.3.1 System Architecture:

The basic structure of the proposed system includes one independent controller for each intersection. The functions that a system intersection controller performs are the following.

- It reads the detector's state and hence it computes traffic flow information.
- It computes the timing schedule for the intersection based on traffic flow.
- It switches traffic lights on and off

The whole system can be integrated in a programmable logic controller or in a micro controller. The number of inputs should be equal to the number of detectors and the number of outputs should be the number of light signals connected to that particular approaches. For the intersection shown in Figure 3.3, the system should include eight digital inputs module completes with twenty four outputs module integrated to the controller.

Figure 4.5 denotes the block diagram of the control mechanism. Getting influence and the delayed cycle time as system inputs, the rule base determines the values for the approach split for each rule. They are aggregated and defuzzified to get a single value for the split. By totalizing four splits, the cycle time is determined.

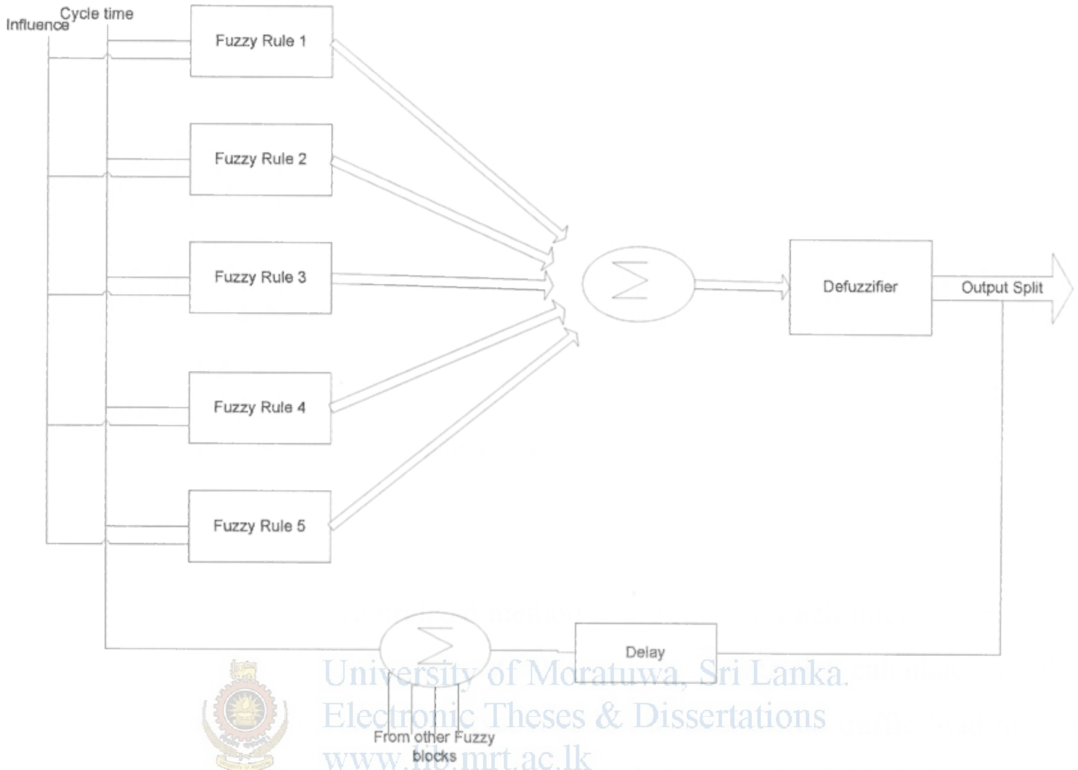

Figure 4.5- Block diagram of the Control System

4.3.2 Decentralized approach:

Many adaptive traffic control systems developed in the world are centralized, network control systems. There one intersection controller is placed at the intersection and performs all functions and computations required at the intersection. The intersection controllers are connected to a central computer named master controller. Intersection controllers act according to the instructions given by the master controller and the master controller calculates the split, offset and cycle time required to the control of traffic load. Hence the total traffic grid is performed for a common cycle. Figure 4.6 shows the way of the communication network built in many other traffic control structures, which are having a network control approach [2].

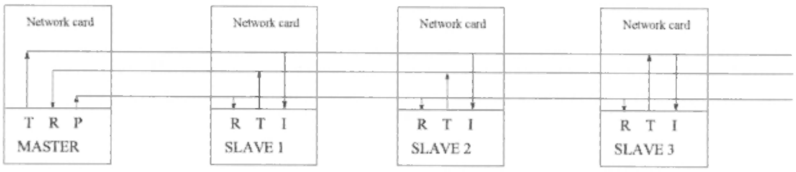

Figure 4.6-Master Slave Control Model

In the proposed system, a decentralized method is introduced. Each intersection at the grid is having its own intersection controller and this controller calculates all the parameters such as split, cycle and offset required to control the traffic load at that intersection. Hence a master controller is not needed. As a result of this independency, intersections at the grid do not perform according to a common optimizing method and hence do not having common parameters at all. With decentralized control strategy, controller may be able to react to incident with short response time and in a proactive way. Since no master controller is required and not functioning according to a common algorithm for the whole grid, adding new intersections to the network increases only the computation loading on neighborhood intersections while rest of the network remains same.

4.3.3 Prediction method:

For proactive traffic control, it is important to predict vehicle arrivals, turning probabilities and queues at intersections, in order to compute phase timings that optimize a given measure of effectiveness [6, 23]. Considering the intersection shown in Figure 3.3, it has four approaches. Associated with each approach are several possible traffic movements of left turn, right turn and a through movement. Any non conflicting combination of movements that can share the intersection at any one time can be assigned a single phase that allows those movements protected use of the intersection.

In the proposed system, detectors are to be installed just after the neighboring intersections hence vehicles are get counted at the beginning of the path to the particular intersection. Thus, the vehicles will get a chance to be on travel after got counted by the relevant detector and the chance has been maximized to have the green phase get activated for the approach to enter, when they reach the intersection. As a result of that, the delay has been minimized.

The vehicle flow pattern is identical to each intersection. If both cross roads have equal priorities as shown in Figure 3.3, High priority is given to the through movement and 15% of vehicle count is assumed to get left or right turns. But this should be differing from intersection to intersection. If one road gets more priority than the other crossed road as if it is a crossed lane, high priority is given to the main road and these turning vehicle percentages are to be got after doing an analysis of these vehicles movement on this intersection along peak hours.

Two issues are important to predicting traffic flow. They are the length of the time horizon and the number of prediction points per time horizon, which is the prediction frequency [3]. The prediction time horizon provides the real time traffic adaptive signal timing control logic with the ability to plan the future signal timing decisions.

It is assumed as the traffic load at the current cycle time is similar to the load got in the previous cycle. If differences are there, it gives negligible effect since the longest cycle time allowed for all intersections is 240 seconds. Hence considerable change cannot be predicted in adjacent time periods. Hence it uses the output of the detectors on the approach as inputs of the controller and inbuilt counters get the counted numerical values of the moved vehicles across the detectors. These counted values are taken for the next phase and cycle time prediction. The signal timing logic assumes that the vehicles arrive uniformly during the particular cycle time and were taken for next cycle calculations.

4.3.4 Control mechanism:

Figure 4.7 shows the control flow diagram of the control strategy. The system adapts its timing to traffic demands by continuously changing the cycle time. Each cycle may have a different duration compared to the previous cycles. The cycle time is computed by the intersection controllers in a distributed fashion at the end of each cycle. The computation of that cycle time is based on traffic measurements taken during the previous cycle. The cycle time can change according to the traffic demand.

Before the beginning of a traffic light cycle, the controller computes the influence according to the detector measurements during the previous cycle. From these influences, the optimal green phases for each approach are computed as described in chapter 3.

At the initial start up, all the splits of all approaches are taken as 15 seconds and the cycle time is to be 60 seconds. The output timers are set in to these values and the light signals are energized accordingly. Starting from approach A, green phase is given 15 seconds each and rotates clockwise till approach D gets its green phase on. While the cycle is running, detectors energize their relay outputs for each vehicle passing across. These output values are the inputs to the controller and the counters in control program count these input pulses. Influence is calculated by the controller and the phase values are determined. By totalizing those four splits, cycle time is computed. After the determination of those splits, the output timers are given those split values and the counters are reset. Then the next cycle run is started with energizing the traffic signals for the determined time period at its green phase. After that cycle run finished, the timers are reset and the next cycle run is started. The same methodology continues and repeats on all subsequent cycles till the controller gets power off or any kind of fault detection.

In order for a controller to compute the optimal timing for the next cycle, it must predict traffic conditions. Thus, the controller assumes that during the next cycle the distribution of arrivals will be exactly the same with that of previous cycle. For that reason, the controller keeps detailed information for the arrivals during the complete cycle time.

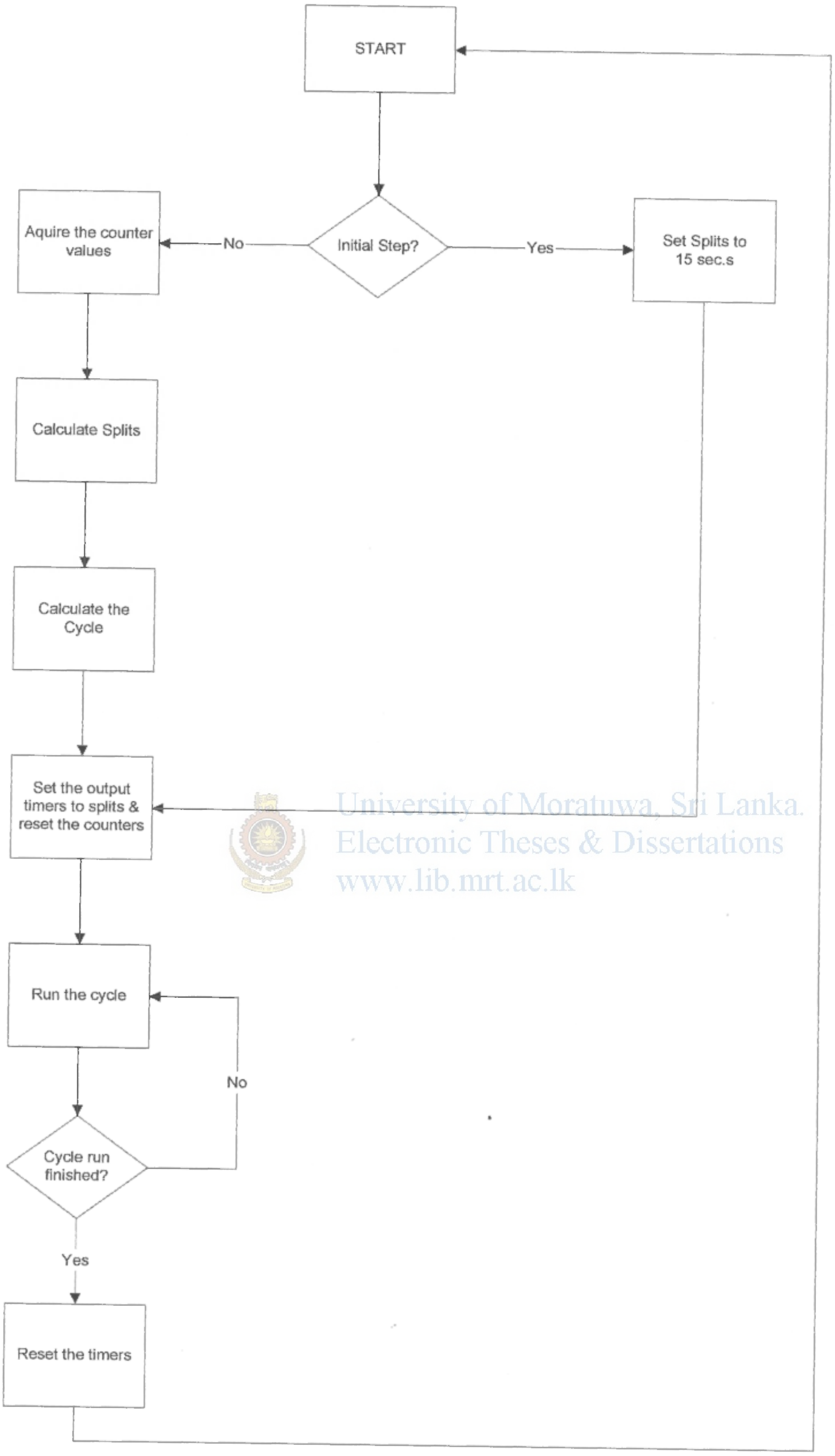

Figure 4.7- Flow Chart for Proposed Adaptive Control Mechanism

The strategy for determining green to each approach and repeating the same sequence of control program has been developed as shown in Figure 4.7 and the simulation model with simulation results for a set of data and comparison results against the existing fixed time traffic control system are described in the next chapter.

Chapter 5: Simulation and Results

5.1 Introduction:

The performance of the mechanism is tested by simulation. Simulation model was developed by Matlab Simulink and the study area was the Narahenpita junction at morning peak hours. The details of the simulaton model and the simulation results are described in this chapter. Later the use of traffic data derived from the previous steps in the process to implement a traffic signal control scheme is examined. Three sets of data from Table 5.1 were taken to do the delay calculation. Both fixed time control system and the intelligent model are compared.

5.2 Simulation Model Development

5.2.1 Simulink basics:

Simulink is a software package that enables to system model, simulate, and analyze systems whose outputs change over time. Such systems are often referred to as dynamic systems. Simulink can be used to explore the behavior of a wide range of real-world dynamic systems. Simulating a dynamic system refers to the process of computing a system's states and outputs over a span of time, using information provided by the system's model [22].

At the start of the simulation, the model specifies the initial states and outputs of the system to be simulated. At each step, Simulink computes new values for the system's inputs, states, and outputs and updates the model to reflect the computed values. At the end of the simulation, the model reflects the final values of the system's inputs, states, and outputs.

A Simulink Block Diagram is a pictorial model of a dynamic system [22]. It consists of a set of symbols, called blocks, interconnected by lines. Each block represents an elementary dynamic system that produces an output either continuously (a continuous block) or at specific points in time (a discrete block). The lines represent connections of block inputs to block outputs. Every block in a block diagram is an instance of a specific type of block. The type of the block determines the relationship between a block's outputs and its inputs, states, and time. A block diagram can contain any number of instances of any type of block needed to model a system.

5.2.2 Traffic simulation model of an intersection:

It is clear that any type of traffic control algorithm needs to be tested in the laboratory before it is implemented and evaluated in the field. The most appropriate method to do this testing is to have a realistic simulation model of traffic flow at an intersection, emulate the traffic detections and observe the resulting changes that would come about if the algorithm was implemented in place of the current control system. Hence the simulation was done with the Matlab developed simulation model.

Figure 5.1 shows the simulation model developed for the intersection shown in Figure 3.3. In the left hand side, the eight detectors are shown. According to the equations (2) – (4), certain percentages of the detector values are taken to calculate the influence. Hence the summation blocks add the detectors and from the gain block, influence is calculated and given as a factorized output, which stands in between 0 and 1 to the Fuzzy Inference System with rule viewer. Four Fuzzy Inference Systems are shown for each separate approach. The influence parameters for each are separately developed prior to insert as a gain at the gain block. The right hand side display blocks provide the output of the Fuzzy Inference Systems and they are the green phases for each approach as shown. At the bottom display, it is shown the cycle time for those particular approach splits.

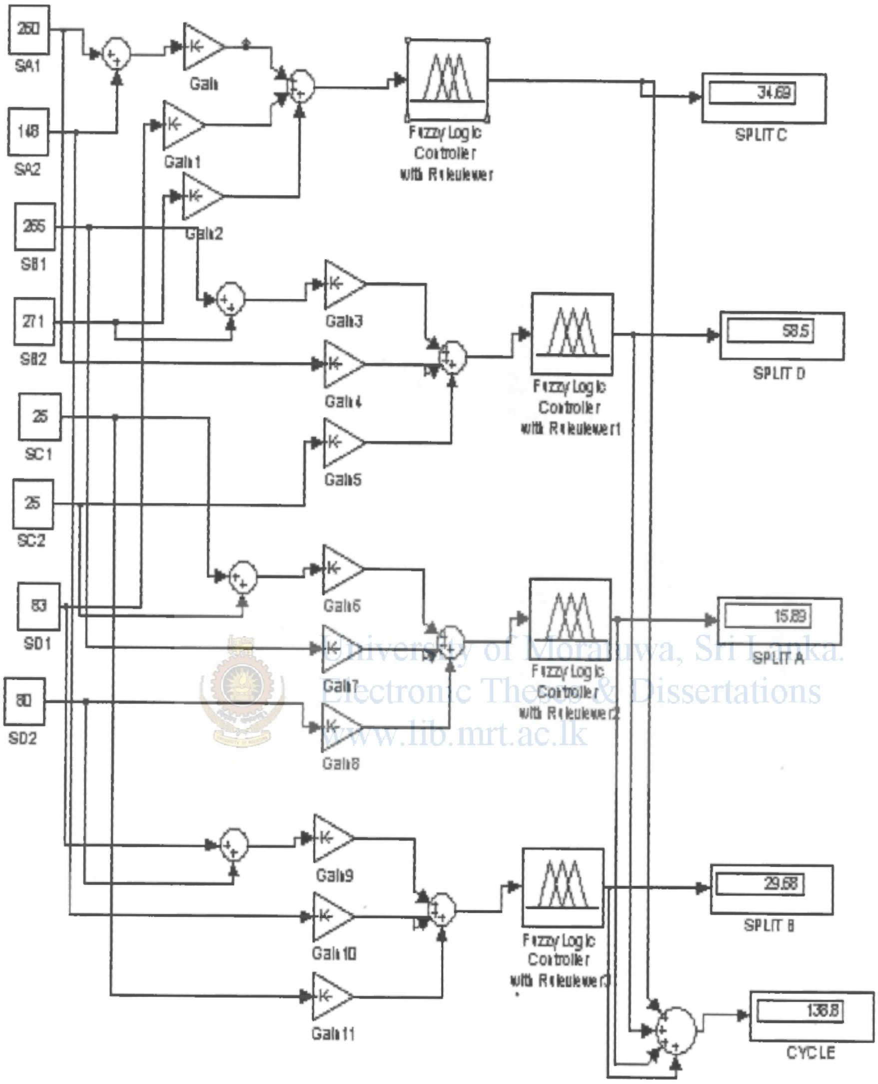

Figure 5.1 Traffic Simulation Model developed by Matlab

5.2.3 Simulation results:

Contained in Table 5.1 are the associated statistics for set counts of generated vehicle arrivals to the intersection shown in Figure 3.3. Detector count values are shown there at five cycle times.

Contained in Table 5.2 are the influences calculated and Table 5.3 gives the splits determined for the data, given to the fuzzy inference systems.

A		B		C		D		CYCLE
S_{A1}	S_{A2}	S_{B1}	S_{B2}	S_{C1}	S_{C2}	S_{D1}	S_{D2}	
60	65	25	20	100	100	5	15	60
103	95	34	30	160	135	16	20	62.41
162	175	93	110	225	210	53	50	94.94
190	200	118	123	230	243	75	72	151.1
221	218	194	190	302	320	201	198	155.7
90	92	35	40	85	81	44	52	190.3
81	78	67	70	53	56	33	38	80.32
52	55	75	78	34	38	22	25	67.32
63	68	51	55	108	115	89	94	56.64
20	26	18	22	15	12	8	5	76.22
16	12	8	5	2	0	4	6	34.49
112	118	102	98	89	94	98	92	25.83
202	198	181	190	178	185	78	83	106.7
120	115	32	38	4	6	51	58	149.7
5	12	115	112	3	6	41	38	76.6

Table 5.1- Detector counts at Approach Counters

A	B	C	D	CYCLE
0.198	0.06	0.077	0.105	62.41
0.29	0.102	0.125	0.15	94.94
0.448	0.244	0.24	0.425	151.1
0.499	0.329	0.281	0.504	155.7
0.702	0.808	0.371	0.783	190.3
0.285	0.167	0.576	0.145	80.32
0.227	0.125	0.508	0.189	67.32
0.127	0.085	0.347	0.183	56.64
0.177	0.242	0.428	0.17	76.22
0.046	0.031	0.147	0.052	34.49
0.046	0.016	0.088	0.021	25.83
0.304	0.269	0.743	0.277	106.7
0.567	0.316	0.99	0.515	149.7
0.372	0.165	0.741	0.132	76.6
0.014	0.081	0.077	0.215	41.37

Table 5.2- Influences Calculated for the Detector Data

A	B	C	D	CYCLE
21.69	9.16	21.87	9.75	62.41
32.54	10.98	39.05	12.37	94.94
51.81	19.07	59.01	21.18	151.1
53.99	20.71	58.93	22.09	155.7
58.95	36.73	58.8	35.86	190.3
20.74	13.02	35.18	11.38	80.32
16.45	10.43	25.74	14.7	67.32
11.68	9.05	21.21	14.71	56.64
22.5	17.99	22.37	13.36	76.22
8.06	6.93	11.5	7.99	34.49
4.71	5.8	9.14	6.17	25.83
21.59	19.27	46.26	19.55	106.7
43.39	20.6	58.91	26.76	149.7
7.06	12.84	45.98	10.73	76.6
7.52	8.9	8.83	16.1	41.37

Table 5.3- Splits determined by the Fuzzy Inference Systems

Considering the above results, it can be noticed that when detector values getting high, the influence also getting increased and hence the split determined by the FIS is also getting long. Thus, it can be noticed that the system is behaving according to the real time traffic load. A comparison done with existing fixed time control system and the improvements obtained by the implementation of the real time traffic control system are discussed below.

5.3 Comparison with existing Fixed time Traffic Control System

5.3.1 Delay calculation:

The delay calculation was done to compare the intelligent model with the existing fixed time traffic control system.

In considering stopped delay, there are three possible conditions for which a vehicle can arrive at any given approach [3, 6]. These are:

- A vehicle can arrive while the approach does not have a green phase, but before the time the green phase begins.
- A vehicle can arrive while the approach has a green phase.
- A vehicle can arrive while the approach does not have a green phase, and after the time the green phase has ended.

If a vehicle arrives while the approach does not have a green phase in either condition, it must experience stopped delay, while it will experience no stopped delay if it arrives during the approach's green phase.

Figure 5.2 shows the way of assigning splits of four approaches of the intersection shown in Figure 3.3 over the cycle time. Starting from approach A, splits are assigned for all four approaches one after the other. If a vehicle reaches the stop line at t1 and is to enter to the approach A, it can pass the intersection without delay. But the vehicle came at the same time and is expecting to enter to the approach B, it has to wait till the split B of the same cycle starts after the split A ended. Likewise all the vehicles reach there at t1 have to wait till their assigned split starts except if they expect to enter to the approach A.

But when considering the vehicles reached the stop line at t_2 and expecting to enter the approach A, the delay is little bit long compared to the rest because they have entered there just after the assigned split for them has ended. Hence they have to wait at the stop line till the next cycle starts. As same as the vehicles reach the stop line at t_3 with the purpose to enter the approach B, have to wait till the split B starts in next cycle run.

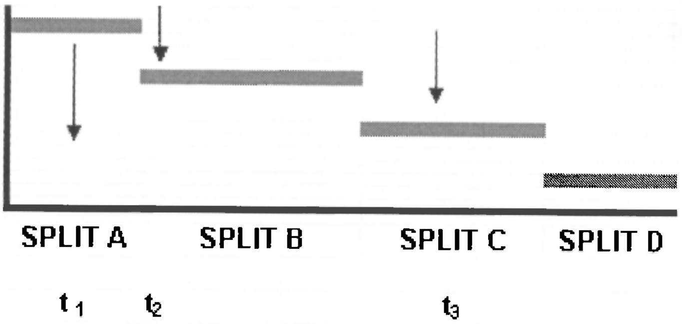

Figure 5.2- Arrivals at an Intersection over the Cycle Time.

The following conditions were considered while calculating the delay in fixed time control system.

- For the intersection shown in Figure 3.3, all four approaches have given similar priorities. Hence all approach green phases are set in to 30 ms each.
- Vehicles are moving at 50 kmh average speed.

5.3.2 Delay calculation for the approach A:

Considering the first three sets of detector values of Table 5.1, the calculation can be generalized as follows [3].

C_{j-1} : Previous cycle time

C: Current cycle time

n_A : Detector count at approach A

S_A : Split at approach A

Since the counters are installed just after the intersection at East, and the majority of the vehicles coming from approach C would expected to enter to the approach A at the determined green phase A, each vehicle after get counted by the detector is to be on travel for 36.23 seconds assuming that the average speed of the vehicle is 50kmh and it has to travel 500 meters before reach the stop line of the approach A.

Further, it has been assumed that the vehicles entering the approach C, which have been counted by the detector SC1 are uniformly counted over the considered cycle C_{j-1}.

Thus, $(C_{j-1} - 36.23)n_A$ number of vehicles have to wait at the stop line $(C_{j-1} -$ 36.23)/2 seconds average. The vehicle counted at the beginning of the cycle C_{j-1}, has to wait $(C_{j-1} - 36.23)$ seconds and the vehicle just counted at 36.23 seconds before cycle C starts will never face any delay. Hence the average delay has been considered.

$s_A\, n_A$ number of vehicles move without delay since when they reach the stop line at approach A, its split has started.

$\{C_{j-1} - (C_{j-1} - 36.23) - s_A\}\, n_A$ vehicles that is $(36.23 - s_A)n_A$ have to wait at stop line (C-s_A) seconds till the next split starts in next cycle run.

Table 5.4 and Table 5.5 are the tabulations calculated for the delay of intelligent model and the fixed time control model. Here the delay 1 and the delay 2 are meant for the before green delay and the after green delay.

Cycle	n_A	**$(C_{j-1}$ - 36.23)**	**$(C_{j-1}$ - 36.23)/2**	**Delay1**
1	1.666667	23.77	11.885	470.8440833
2	2.563692	26.18	13.09	878.5674091
3	2.369918	58.71	29.355	4084.392366

$s_A\, n_A$	**36.23-s_A**	**(C-s_A)**	**Delay2**	**Delay total**	**Delay %**
36.15	14.54	40.72	**986.7813**	1457.62542	**2335.56388**
83.42253	3.69	62.4	**0**	878.567409	**925.392257**
122.7854	0	99.29	**0**	4084.39237	**2703.10547**

Table 5.4- Delay Calculation for the Intelligent Model at Split A

Delay percentages were taken for the comparison with the fixed time control model. For the fixed time control model, the total cycle time is never changing and it has been set to 120 seconds. But for the intelligent model, it is changing from time to time. Hence for the comparison, it has to average it to a common time horizon. Hence the percentage delay was taken by calculating the total delay accumulated for a 100 seconds cycle time for both fixed time control and the intelligent model.

Cycle	n_A	$(C_{j-1}$ - 36.23)	$(C_{j-1}$ - 36.23)/2	Delay1
1	1.666667	83.77	41.885	5847.844083
2	2.563692	83.77	41.885	8995.24166
3	2.369918	83.77	41.885	8315.346021

s_A n_A	36.23- s_A	(C- s_A)	Delay2	Delay total	Delay %
50	6.23	90	934.5	6782.34408	**5651.9534**
76.91075	6.23	90	1437.462	10432.7036	**8693.91967**
71.09754	6.23	90	1328.813	9644.15896	**8036.79913**

Table 5.5- Delay Calculation for the Fixed time Control Model at Split A

5.3.3 Delay calculation for the approach B:

Considering the first three sets of detector values of Table 5.1, the calculation can be generalized as follows.

C_{j-1} : Previous cycle time

C: Current cycle time

n_B : Detector count at approach B

s_B : Split at approach B

$s_{A(j+1)}$: Split at approach A at the next cycle C_{j+1}

For the approach B, the majority from the vehicles are counted by the detector S_{D1}. For this calculation, each vehicle has to move 1000 meters after got counted by the detector and hence has to be on travel for 72.5 seconds to reach the stop line at approach B.

Considering the average delay as usual, s_A n_B number of vehicles will experience the before green delay at the stop line and have to wait (C_{j-1} + s_A /2 – 72.5) seconds.

s_B n_B number of vehicles will enter the approach without delay.

Similar to the before calculation, $(s_C + s_D)n_B$ number of vehicles have to experience after green delay and have to wait $((s_C + s_D)/2 + s_{A(j+1)})$ seconds till the next green phase begins.

Table 5.6 and Table 5.7 are the tabulations calculated for the delay of intelligent model and the fixed time control model.

Cycle	n_B	**$(C_{j-1} + s_A$ /2-72.5)**	s_A	**Delay1**	$s_B\ n_B$	$s_C + s_D$	**$(s_C + s_D)/2$ $+ s_{A(j+1)}$**	**Delay2**	**Delay total**	**Delay %**
1	0.083333	-1.655	21.69	0	0.763333	31.62	48.35	127.4023	127.40225	**204.137558**
2	0.256369	6.18	32.54	51.55512258	2.814934	51.42	77.52	1021.908	1073.46274	**1130.67489**
3	0.558247	48.345	51.81	1398.272444	10.64578	80.19	91.905	4114.206	5512.47808	**3648.23169**

Table 5.6-Delay Calculation for the Intelligent Model at Split B

When the waiting time getting negative, it meant as a zero delay since it appears when the previous cycle time is shorter comparing the travel time after the vehicle got counted.

Cycle	n_B	**$(C_{j-1} + s_A$ /2-72.5)**	s_A	**Delay1**	$s_B\ n_B$	$s_C + s_D$	$(s_C + s_D)/2$ $+ s_{A(j+1)}$	**Delay2**	**Delay total**	**Delay %**
1	0.083333	62.5	30	156.25	2.5	60	60	300	456.25	**380.208333**
2	0.256369	62.5	30	480.6921968	7.691075	60	60	922.929	1403.62121	**1169.68435**
3	0.558247	62.5	30	1046.713714	16.74742	60	60	2009.69	3056.40404	**2547.00337**

Table 5.7- Delay Calculation for the Fixed time Control Model at Split B

5.3.4 Delay calculation for the approach C:

Considering the first three sets of detector values of Table 5.1, the calculation can be generalized as follows.

C_{j-1} : Previous cycle time

C: Current cycle time

n_C : Detector count at approach C

s_C : Split at approach C

$s_{A(j+1)}$: Split at approach A at the next cycle C_{j+1}

$s_{B(j+1)}$: Split at approach B at the next cycle C_{j+1}

For this calculation, the before green delay has been calculated as delay 1 and the delay 2 for two sets of vehicles.

Split C is on for the approach C, in which the majority from the vehicle count from the intersection at West is expected to be entered. Each vehicle has to travel 300 meters over 21.74 seconds after getting counted by the detector S_{A1}.

s_A n_C number of vehicles have experience a before green delay of $(C_{j-1} + s_A /2 - 21.74) + s_B$ averaged seconds at the stop line.

s_B n_C number of vehicles have experience a before green delay of $(C_{j-1} - s_A + s_B/2 - 21.74)$ averaged seconds at the stop line.

s_C n_C number of vehicles will never experience a delay.

$(C_{j-1} - s_A - s_B - s_C) n_C$ number of vehicles will have after green delay and will wait $(s_D/2 + s_{A(j+1)} + s_{B(j+1)})$ averaged seconds before pass the intersection.

Table 5.8 and Table 5.9 are the tabulations calculated for the delay of intelligent model and the fixed time control model.

Cycle	n_C	$(C_{j-1} + s_A /2 - 21.74) + s_B$	s_A	Delay1
1	1	58.265	21.69	1263.76785
2	1.650377	67.92	32.54	3647.524922
3	1.706341	118.175	51.81	10447.32224
$s_C\ n_C$	**$s_B\ n_C$**	**$(C_{j-1} - s_A + s_B/2 - 21.74)$**	**Delay2**	
21.87	9.16	21.15	193.734	
64.4472	18.12113	13.62	246.8099	
100.6912	32.53992	30.925	1006.297	
$(C_{j-1} - s_A - s_B - s_C) n_C$	**$(s_D/2 + s_{A(j+1)} + s_{B(j+1)})$**	**Delay3**	**Delay total**	**Delay %**
7.28	48.395	352.3156	1809.81745	**2899.884**
-33.271591	77.065	0	3894.33477	**4101.89**
-59.636613	85.29	0	11453.6193	**7580.158**

Table 5.8- Delay Calculation for the Intelligent Model at Split C

Cycle	n_C	$(C_{j-1} + s_A /2 - 21.74) + s_B$	s_A	Delay1
1	1	143.26	30	4297.8
2	1.650377	143.26	30	7092.988303
3	1.706341	143.26	30	7333.511692
$s_C\ n_C$	**$s_B\ n_C$**	**$(C_{j-1} - s_A + s_B/2 - 21.74)$**	**Delay2**	
30	30	83.26	2497.8	
49.5113	49.5113	83.26	4122.311	
51.19023	51.19023	83.26	4262.098	
$(C_{j-1} - s_A - s_B - s_C) n_C$	**$(s_D/2 + s_{A(j+1)} + s_{B(j+1)})$**	**Delay3**	**Delay total**	**Delay %**
30	75	2250	9045.6	**7538**
49.5112963	75	3713.347	14928.6461	**12440.54**
51.1902254	75	3839.267	15434.8768	**12862.4**

Table 5.9- Delay Calculation for the Fixed time Control Model at Split C

Here also zero delay will receive when negative delay time received.

5.3.5 Delay calculation for the approach D:

Considering the first three sets of detector values of Table 5.1, the calculation can be generalized as follows.

C_{j-1} : Previous cycle time

C: Current cycle time

n_C : Detector count at approach C

s_C : Split at approach C

$s_{A(j+1)}$: Split at approach A at the next cycle C_{j+1}

$s_{B(j+1)}$: Split at approach B at the next cycle C_{j+1}

For this calculation, the before green delay has been calculated as delay 1, delay 2 and the delay 3.

Split D is on for the approach D, in which the majority from the vehicle count from the intersection at North is expected to be entered. Each vehicle has to travel 1000 meters over 72.5 seconds after getting counted by the detector S_{B1}.

$s_A\ n_D$ number of vehicles have experience a before green delay of $(C_{j-1} + s_A /2 - 72.5) + s_B + s_C$ averaged seconds at the stop line.

$s_B\ n_D$ number of vehicles have experience a before green delay of $(C_{j-1} - s_A + s_B/2 - 72.5) + s_C$ averaged seconds at the stop line.

$s_C\ n_D$ number of vehicles have experience a before green delay of $(C_{j-1} - s_A + s_B/2 - 72.5)$ averaged seconds at the stop line.

$(C_{j-1} - s_A - s_B - s_C)\, n_D$ number of vehicles will move without delay occurrence from the intersection.

Table 5.10 and Table 5.11 are the tabulations calculated for the delay of intelligent model and the fixed time control model.

Cycle	n_D	$(C_{j-1} + s_A /2 - 72.5) + s_B + s_C$	s_A	Delay1	$(C_{j-1} - s_A - s_B - s_C)$	$s_B\ n_D$	$(C_{j-1} - s_A + s_B/2 - 72.5) + s_C$	Delay2	$s_C\ n_D$	$(C_{j-1} - s_A - s_B + s_C/2 - 72.5)$	Delay3	Delay total	Delay %
1	0.416667	31.03	21.69	280.433625	7.28	3.816667	-7.74	0	9.1125	-32.415	**0**	280.433625	**449.3409**
2	0.544784	133.8	32.54	996.4508188	-20.16	5.981734	1.91	11.42511	21.2738343	-34.085	**0**	1007.87593	**1061.593**
3	0.979566	161.85	51.81	8214.100595	-34.95	18.68032	39.175	731.8017	57.8041921	-18.935	**0**	8945.9023	**5920.518**

Table 5.10-Delay Calculation for the Intelligent Model at Split D

Cycle	n_D	$(C_{j-1} + s_A /2 - 72.5) + s_B + s_C$	s_A	Delay1	$(C_{j-1} - s_A - s_B - s_C)$	$s_B\ n_D$	$(C_{j-1} - s_A + s_B/2 - 72.5) + s_C$	Delay2	$s_C\ n_D$	$(C_{j-1} - s_A - s_B + s_C/2 - 72.5)$	Delay3	Delay total	Delay %
1	0.416667	60	30	750	30	12.5	62.5	781.25	12.5	2.5	31.25	1562.5	**2503.605**
2	0.544784	58.345	30	2002.083	30	16.34353	62.5	1021.471	16.3435347	2.5	40.85884	3064.41275	**3227.736**
3	0.979566	58.345	30	1714.583421	30	29.38698	62.5	1836.686	29.3869813	2.5	73.46745	3624.7372	**2398.9**

Table 5.11-Delay Calculation for the Fixed time Control Model at Split D

With comparing the results obtained, the summary of the all tabulations are given in Table 5.12.

Cycle time	Total Delay % IC	Total Delay % FC	Improvement %
62.4	5865.0	17325.6	66.1
94.9	7219.6	25531.9	71.7
151.1	18662.2	27092.8	31.1

Table 5.12- Improvement obtained by the Intelligent Control Model

It is clear that for low, medium and averaged high traffic flow data, it has been improved 51.6% minimization of the total vehicle seconds delay by the intelligent traffic control model over the fixed time control system.

Chapter 6: Conclusions and Future work

6.1 Conclusions

In this research, the main focus is to develop an intelligent traffic signal control system based on Fuzzy logics. The mechanism for the development of an intersection wise traffic control system has been described.

First, one traffic four way intersection has been selected for experimenting the traffic load and simulation. Since the methodology described in the thesis is decentralized approach, one intersection results were sufficient. According to the traffic data at various times, the influence functions were built and the methodology was developed to sense the influence by the controller. The detector values are taken periodically and the influences are calculated accordingly.

The Fuzzy Inference System in Matlab tool box was used to develop the fuzzy rules. Five fuzzy rules were developed for deciding traffic conditions and the way of deciding output green time was developed.

Matlab Simulink was used to develop the simulation model of the intelligent traffic signal control system. Finally, the results taken from simulating the traffic load data at various times in both control methods, the intelligent model and the existing fixed time control method have been shown and the comparison of both methods with improvements are shown.

At each intersection, there should be a controller. And all the rules mentioned are to be implemented individually, considering the traffic levels at each intersection.

By the decentralized method developed, dependency of a central controller is avoided and hence it is easy for further expansions coming through construction, and the new intersections which would come to the grid later. Thus, adding new intersections to the network increases only the computation loading on neighborhood intersections while rest of the network remains the same.

6.2 Future work

The main purpose of the Intelligent Traffic Control Model is to optimize the delay occurring on roads and hence to minimize the heavy traffic jam at city roads. The method developed in this research is decentralized approach and hence to optimize the delay at each intersection, traffic data are to be studied in each intersection in the grid. The influence is to be taken accordingly.

Make of the program to implement the method described herein by a programmable logic controller is needed.

Developing a mathematical model to optimize the delay and to find the equation of influence by mathematical modeling is suggested. Rather been depending upon the assumptions, influence calculation from a proper experiment through results is suggested.

For finding machine learning techniques such as Neural networks, to implement adaptive methods to suggest alternative ways to move for the vehicles by sensing the huge disturbances on roads if accident or construction work occurred by the controller is suggested.

References

[1] Szu-yin, Duen-Kai Chen, and Ruey-Shun Chen "Apply adaptive and cooperative multi agent system to urban traffic signal control", Proc. Of the 1996 IEEE International Conference, pp 582-589, 1996

[2] Tavaladakis. K, Voulgaris N.C. "Development of an autonomous adaptive traffic control system", Paper presented at the 78th Transportation Research Board Meeting, Washington D.C,2000

[3] Michael Shenoda, Randy Machemehl. "Development of a Phase by phase, arrival based, delay optimized Adaptive traffic Signal Control methodology with Metaheuristic Search", Master thesis, University of Texas at Austin, October 2006

[4] Josefa Hernandez,Jose Cuena, Martin Molina, "real time traffic management through knowledge based models: The TRYS approach" , Paper presented at 12th European Conference on Artificial Intelligence, 2000

[5] Pitu Mirchandani, Larry Head, "RHODES: A real time traffic signal control system: Architecture, Algorithms and analysis", Transportation Research Record, Part B, 2001

[6] Jeanan Sha Aban, Andy Tomlinson, Benjamin Heydecker, larry Bull, "Adaptive Traffic control using evolutionary algorithms", Working paper from the Transport Operations Research Group, Newcastle University, 2000

[7] Jim McGuire, David O' Keeffe, "The Limerick Adaptive Urban Traffic Control System Project", NASA Joint program on Remote Sensing, 2002.

[8] David Olsen, "Fuzzy Logic Control and Autonomous Robotics:Investigating the Motorola MC68HC12 on a line following Robot", Paper presented at Wolfram Research Inc, 2003

[9] Jung Hoon Lee, Beom Hee Lee," A real- time traffic control scheme of multiple AGV systems, a routing table approach", Proceedings of the third Annual World Congress on Intelligent Transport Systems USA, 2000

[10] Tsu-Tian Lee,"Intelligent Transportation Systems", Paper presented at American Control Conference, 2002

[11] Chang E.C.P, Lei J.C, Messer C. J," Arterial Signal Timming Optimization Using PASSER II-87 – Micro computer User's Giude", Texas Transportation Institute, Texas, pp.73-85, 1998

[12] Waalace C. E., et al," TRANSYT - 7F User's Manual", Transportation Research Center, University of Florida, Gainesville, Florida, pp 65-71, 1998

[13] Chang, E. C. P., Lei, J. C., and Messer, C. J., "Arterial Signal Timing Optimization Using PASSER II-87 ," Microcomputer User's Guide, pp. 73-85,1989

[14] http://www.scoot-utc.com (Seimens web page)

[15] Wallace, C. E., et al., "TRANSYT-7F User's Manual", Transportation Research Center, University of Florida, Gainesville, Florida, pp.91-115, 1988.

[16] Wann-Ming Wey,"A Study on an Urban Network Traffic Signal Control,"Transportation Planning Journal,Vol.29 No.4 ,pp.693~708, December 2000.

[17] Danko A. Roozemond, Jan L.H. Rogier, "Agent control traffic lights", ESIT2000, pp. 14-15, September 2000, Aachen, Germany.

[18] http://www.pwmag.com/industry (web)

[19] Enrique D. Ferreira, Eswaran Subrahmanian, Dietrich Manstetten "Intelligent agents in decentralized traffic control," IEEE Intelligent Transportation System Conference Proceedings,Vol 8, pp.12-35, August, 2001

[20] Stuart J. Russell, Peter Norvig, "Artificial intelligence/a modern approach", Englewood Cliffs, NJ. Prentice Hall International, pp. 81-125, 1995.

[21] David E Lucas, Pitu B. Mirchandani, K Larry Head,"Using remote simulation to evaluate Real time traffic control strategies",pp.73-90, 2001.

[22] http://www.mathwork.com (Software Manual Matlab)

[23]Chandru Mirchandani,''Automatic Traffic control system using existing Infrastructure", Paper presented at the IEE International Conference on road traffic signaling at London, U.K.

[24] http://www.signalsystems.org.vt.edu (web)

[25] http://www.opsfhwa.dot.gov/publications/adaptivecontrol (web)

[26] Hunt, P.B., D. I. Robertson, R. D. Bretherton, and R. I. Winton. "SCOOT - a traffic responsive method of coordinating signals. Laboratory Report 1014", Transport and Road Research Lab, Crowthorne, Berkshire, U.K. 1981

[27] Hansen, B.G., P. T. Martin, and H. J. Perrin. "SCOOT real-time adaptive control in a CORSIM simulation environment", Transportation Research Record 1727. National Research Council, Washington, D.C, 2000.

[28] Lowrie, P.R. " SCATS: The Sydney Co-ordinated Adaptive Traffic System", Paper presented at the IEE International conference on road traffic signaling at London, U.K, 1982

[29]Brain Dong, Graham Roff,"Adaptive Self-Confiuring Traffic control systems", CSE237A Final report June 11, 2004.

[30] http://www.traffic-tech.com/pdf/ (web)