1996

An adaptive fuzzy logic controller for intelligent networking and control

Irshad Nainar
Edith Cowan University

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

An Adaptive Fuzzy Logic Controller for Intelligent Networking and Control

By

Irshad Nainar
Bachelor of Computer Science and Engineering
Madras University, India

A Thesis Submitted in Partial Fulfilment of the Requirements for the

Award of

Master of Science (Computer Science)

Department of Computer Science

Faculty of Science, Technology and Engineering

School of Mathematics, Information Technology and Engineering (SMITE)

Edith Cowan University

Perth, Western Australia

April 1996

ABSTRACT

In this thesis, we present a fuzzy logic control scheme to regulate the flow of traffic approaching a set of intersections. An adaptive Fuzzy Logic Traffic Controller (FLTC) is used to adjust the green phase split of the north-south and east-west approaches of a set of traffic signals based on the actual traffic approaching the intersection. Each intersection is coordinated with its neighbouring intersections by adjusting the offset of the local intersection. The offset is adjusted by a local fuzzy logic controller loacted at each intersection. A new fuzzy control scheme, using a supervisory Fuzzy Logic Controller, is also proposed for adjusting the offset. The fuzzy knowledge base of the supervisory Fuzzy Logic Controller is automatically generated by Genetic Algorithms (GAs). The fuzzy rules generated by the integrated Fuzzy Logic and Genetic Algorithm architecture is found to be effective in optimising the traffic flow.

The effectiveness of the above fuzzy control scheme is established through simulations of the traffic flow approaching an isolated intersection, two adjacent intersections, and a set of three intersections. The superiority of adjusting offset using a supervisory fuzzy logic controller is established through simulations.

Acknowledgement

I would like to extend my sincere gratitude to my supervisors: Mr. Masoud Mohammadian and Dr. Jim Millar for their support, guidance and encouragement throughout the preparation of this thesis.

I would also like to thank the staff members of the Department of Computer Science and the library staff at Edith Cowan University for their co-operation and support during my tenure as a Master's candidate at Edith Cowan University.

Finally, I would like to thank my family and friends for their love and moral support during this period.

Declaration

I certify that this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education; and that to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where due reference is made in the text.

Irshad Nainar

Date : 09 / 04 / 1996

TABLE OF CONTENTS

List of Figures

List of Tables

Abbreviations

<u>**mnemonic**</u>	<u>**meaning**</u>
AI	Artificial Intelligence
FL	Fuzzy Logic
FLC	Fuzzy Logic Control
GA	Genetic Algorithm
SOFLC	Self Organising Fuzzy Logic Controller
KBS	Knowledge Based System
ATC	Area Traffic Control
UTC	Urban Traffic Control
AUTCS	Automated Urban Traffic Control System
DICS	Distributed Intelligent Control System
FDAI	Fuzzy Distributed Arificial Intelligence
FTC	Fuzzy Traffic Controller
FLTC	Fuzzy Logic Traffic Controller

Chapter 1 Introduction

1.1 Introduction

The design of controllers for regulating a process is dependent on the availability of a model for the process. However, it is not always easy to derive a mathematical model for processes which are non-linear, ill-defined and dynamic in nature. Conventional control algorithms like PID (Proportional - Integral - Derivative) and MRAC (Model Reference Adaptive Controller) techniques attempt to cope with these system nonlinearities, but these techniques are too complex and time consuming for most real world applications. (Li Y.F., et al, 1989).

The problems associated with non-linear systems led researchers to incorporate human intelligence into automatic control systems. The rationale for developing control systems based on human intelligence is the ease with which certain industrial processes are controlled by human operators than by automatic control systems. The operators are aware of how the system will respond to their control actions; knowledge which they have acquired through years of experience. This resulted in the design of intelligent control systems and since then many efforts have been made to find methods for designing control systems that incorporate knowledge based on human experience.

These systems, based on expert's knowledge and human operator's experience, are called knowledge based systems. In the control of certain real world applications, sufficiently precise information is often not available and certain decisions have to be taken in an environment which is imprecise and vague. In such situations, the decisions are made on the basis of the decision maker's experience, intuition and evaluation of the parameters. The knowledge gained by experts through years of experience, thus becomes a useful tool in making judicious judgements and decisions in uncertain circumstances.

Fuzzy logic control is a special form of knowledge based control. Fuzzy logic control systems are designed based on the heuristic of the process to form a set of fuzzy rules which basically sum up people's common sense and experience. Specifically, the use of fuzzy logic has proved to enhance the ability of intelligent control systems.

Fuzzy logic provides a gamut of concepts and techniques for representing and inferring from knowledge that is imprecise, uncertain, or unreliable. It is concerned with the formal principles of approximate reasoning. It is much closer in spirit to human thinking and natural language than traditional logical systems (Zadeh L. A., 1988). Fuzzy logic is an attractive proposition when the process is either difficult to control or difficult to model by conventional methods.

A fuzzy logic control system comprises fuzzy control rules that are based on an operator's knowledge. He/She makes a decision based solely on intuition and experience without any knowledge of the underlying dynamics of the system. However, in certain cases, an operator finds it difficult to express the kind of action he/she takes in a particular situation thereby making it difficult to transfer the expert's knowledge into a knowledge base. Moreover, obtaining information by interviewing operators and experts can become a lengthy, costly and a time consuming process. In such cases, an automatic strategy for developing fuzzy control rules is highly desirable.

One technique that is becoming very popular is the design of fuzzy logic controllers which have the capability of learning from evolution. The integration of fuzzy logic and genetic algorithms provides a powerful tool which emulates the decision making ability of a human operator, and the capability to learn an optimal action. Genetic algorithms have been used widely in many areas such as image processing (Fitzpatrick J.M., et al, 1984), travelling salesman problem (Goldberg D.E., et al, 1985) and control applications (Kumar K.K., et al, 1990).

Genetic algorithms (GAs) are randomised and global search techniques that are based on the mechanics of natural selection and natural genetics. They are different from other traditional search techniques, in that, they manipulate codings of candidate solutions to find near optimal solutions based on a system specific performance criterion. GAs exploit historical data to locate new points in the search space with an expected improvement in the performance of the system (Goldberg D., 1989). These properties enable Genetic Algorithms to generate high performance fuzzy rules for a Fuzzy Logic Controller.

In this research, we attempt to develop a Fuzzy Logic Control scheme to integrate and control a network of systems in a common workplace. In such a situation, a given system should be able to communicate with other systems and should also be able to adapt to the changes in the environment while at the same time fulfilling its desired objectives. The effectiveness of this Fuzzy Logic Control scheme is illustrated by applying it to a set of urban traffic signals.

The traffic flow approaching an intersection is regulated by a set of fuzzy control rules which adjusts the green phase splits of the north-south and east-west approaches of the signal, based on the traffic volume at these approaches. Each intersection is coordinated with its neighbouring intersections using another fuzzy logic controller whose rules adjust the offset at each intersection based on the traffic at the neighbouring intersections. Offset is the time difference between the start of each phase among adjacent intersections. Thus, the traffic signal at each intersection is controlled by two fuzzy logic controllers - one, based on the local traffic and the other, based on the vehicular traffic at the neighbouring intersections.

1.2 An overview of the problem

Traffic signals in use today typically operate based on a preset timing schedule. An Area Traffic Control (ATC) system consists of a number of traffic signals which are linked in such a way that any signal timing change is dependent upon conditions at any of the other intersections. The methods for controlling the traffic signals can be classified into two kinds - Fixed-time control and Traffic-Responsive control (Luk J.Y.K., 1984).

In Fixed-time control, timing plans for different times of the day are made off-line and switched into operation according to the time of day. The preparation of these plans and their fine tuning is often a time consuming and labour-intensive task. Vehicle detectors are not required and the coordination of intersections is achieved by linking local controllers to a master controller by means of a system of cables (Luk J.Y.K., 1984).

In Traffic-responsive control, the timing parameters are calculated according to the prevailing traffic conditions. These systems respond to changes in the traffic by performing incremental optimisations at the local level. The two most notable Traffic-responsive methods are the - Sydney Co-ordinated Adaptive Traffic (SCAT) method developed in Australia (Sims A. G., 1979) and Split, Cycle and Offset Optimisation Technique (SCOOT) developed in the U.K (Robertson D., 1969).

Both SCAT and SCOOT incrementally optimise the signal's cycle time, phase split, and offset. The cycle time is the duration for completing all phases of a signal; phase split is the division of the cycle time into periods of green signal for competing approaches; offset is the time relationship between the start of each phase among adjacent intersections.

The Automated Urban Traffic Control Systems (AUTCS) that are in prevalent use today have either a centralised or distributed architecture. In a centralised AUTCS, the information gathering and processing, and the control computations are carried out in a centralised manner by the central computer. In the case of a distributed AUTCS, the central computer plays the role of a supervisory controller accounting for the information between subsystems.

These systems, centralised and distributed, are not without their limitations. Congestion is one of the most relevant factors that limits the performance of conventional traffic control systems. Also, the existing control strategy is unable to respond adequately to unforseen changes in the traffic conditions caused by accidents, road blockages, failure of traffic signals, road maintenance, etc. This is because it is designed to react only to small changes in traffic flows and not to deal with unexpected changes in the traffic environment.

These limitations found in AUTCS can be attributed to the following circumstances:

1. When a large quantity of information has to be processed, the efficiency of the centralised AUTCS is reduced (Scemama G., 1990).

2. AUTCS having a distributed structure also have their drawbacks. The accounting of information between subsystems is not very efficient and the communication structure between modules is very complex (Barriere J., et al., 1986).

3. Moreover, most of the AUTCS operate by means of a quantitative algorithm without taking into consideration the qualitative aspects of the transport process (Wu J., et al., 1991).

In order to resolve the above issues, some kind of strategic control is necessary for treating different problems simultaneously and making appropriate evaluations and decisions. It is thus desirable to use techniques which are based on artificial intelligence principles to solve transport problems in large cities. These systems are called 'Distributed Intelligence Control Systems' (DICS) (Gegov A., 1994).

DICS are a new class of systems based on control theory, artificial intelligence and computer technology. They are characterised by distributed information processing and intelligent operational capabilities (Decker K., 1987). DICS comprises distributed intelligent control units which operate together to achieve a common goal (Yang D., et al., 1985). These systems are characterised by both quantitative and qualitative features which make them far superior to the current AUTCS (Siljak D., 1983).

Fuzzy logic control (FLC) is an alternative to conventional control when the process to be controlled is too complex to be analysed by conventional techniques or when the nature of the information obtained about the system is inexact, imprecise or uncertain. FLC is not incompatible with conventional control techniques but in contrast to them, incorporates the expert's knowledge of the application domain and arrives at a decision along lines that simulate human thinking, rather than being based purely on numerical calculations.

Fuzzy logic is a powerful tool for the design of intelligent systems. It has been successfully applied to many control problems and is now finding its use in solving complex traffic problems. Fuzzy logic provides opportunities for formalising the human way of thinking and perception of the environment (Gegov A., 1994).

In this thesis, a fuzzy control scheme is proposed for regulating the traffic flow approaching a single traffic intersection, two adjacent intersections, and a set of three intersections in a two-way street.

Chiu and Chand (Chiu S., et al., 1993) present a distributed approach to traffic signal control where an adaptive fuzzy logic controller is used for controlling multiple intersections in a network of two-way streets. A set of fuzzy rules is used at each intersection to adjust the cycle time, phase split and offset based on the local traffic and the traffic at the upstream intersection. Thus, the signal timing parameters at each intersection are adjusted based on the local information and coordinated only with adjacent intersections.

A set of forty six control rules is used for adjusting the signal timing parameters. The rules are divided into three fuzzy knowledge bases: a knowledge base consisting of twenty five rules for adjusting cycle time and green phase of east-west approach, a knowledge base consisting of eighteen rules for adjusting offset, and a knowledge base consisting of three rules for determining appropriate constraints on the cycle time value.

The cycle time and the green phase of the east-west approach of a traffic signal are adjusted by a fuzzy logic controller, based on the degree of saturation in the north-south and east-west approaches. The degree of saturation is determined to be the ratio of the number of vehicles that passed through the intersection during the previous green phase to the maximum number of vehicles that can pass through during that period. It determines the effectiveness of the green period. Offset is adjusted to coordinate each intersection with its upstream intersection. It is adjusted by using another local fuzzy logic controller located at each intersection.

The fuzzy rules proposed by Chiu and Chand for adjusting the cycle time and green phase of the east-west approach are based on the assumption that north and south directions are the dominant directions of traffic flow, and they optimise the traffic flow only in those directions. Also, their fuzzy control scheme coordinates each intersection with only its upstream intersection and there is no interaction with any of the other neighbouring intersections.

In this research, the traffic flow approaching an isolated intersection is regulated using a fuzzy logic traffic controller which adjusts the green phase of the north-south and the east-west approaches. The adjustments are made based on the ratio of the number of vehicles waiting at the respective approaches (queue length) to the number of vehicles that passed through the intersection during the previous green phase. Two fuzzy control schemes are investigated for adjusting the offset at each intersection:

(i) The offset is adjusted by a local fuzzy logic controller located at each intersection which coordinates each intersection with only its upstream intersection.

(ii) The offset is adjusted by a supervisory fuzzy logic which coordinates each intersection with its neighbouring intersections rather than just its upstream intersection.

The control algorithm developed using fuzzy logic controllers aims to overcome the limitations of the existing conventional control strategies, which are not adaptive to the changes in the traffic environment. The fuzzy logic control scheme optimises the traffic flow by reducing the waiting time of vehicles and reducing the number of vehicles waiting at the traffic junctions. It adapts to the variations in the traffic conditions and attempts to improve the overall performance of the traffic signals.

1.3 Why Fuzzy Logic ?

The problem of controlling uncertain dynamic systems has intrigued control engineers for several years, especially those systems which are subject to external disturbances and systems that are complex, ill-structured or model-free in nature. For these systems, setting up a model can be very difficult and they are best described qualitatively, and handled by human operators.

In classical control system design, the initial step is to obtain a mathematical model for the process to be controlled. This model represents a priori information about the system. In recent years, a great deal of attention has been paid to model-based control such as linear control, non-linear control, and adaptive control. But, in certain cases, it is difficult to obtain a precise mathematical model for many real world systems which are highly complex and have nonlinear characteristics. In order to overcome this difficulty in control systems, fuzzy logic control can be applied (Mamdani E.H., et al, 1981, Tanscheit R., et al, 1988). Fuzzy logic control is the application of fuzzy logic theory to a control problem.It has proved to be an useful alternative when the system to be controlled is non-linear and uncertain.

Fuzzy Logic Control is a design methodology that simulates the human description of the physical system and the required control strategy in a reasonably natural way. It provides a means of converting this linguistic control strategy into an automatic control strategy. Fuzzy logic control attempts to solve complex control problems by using a set of *If-then* rules such as '*If* x = LARGE and y = ZERO *then* output = LARGE'. These rules are expressed not in the form of equations but in linguistic terms or in a manner expressed by humans.

Traffic flow is usually characterised by ambiguity, uncertainty, subjectivity, and imprecision and some sort of a model has to be developed to satisfactorily deal with these factors and evolve an optimal solution for complex traffic conditions (Teodorovic D., 1994). Regulating vehicle movements at intersections, using traffic lights, has emerged as one of the most effective and flexible means of controlling urban road traffic. However, obtaining a valid model of the traffic flow theory is still difficult.

Fuzzy logic control can be an appropriate tool for controlling traffic lights at an intersection because of its capacity to deal with a wide range of traffic patterns and the uncertainties that exist in the traffic systems. Fuzzy logic is a theory about vagueness and uncertainty and it enables ill-defined concepts to be used for ill-defined situations. A fuzzy controlled traffic signal uses sensors that gives a count of the number of vehicles waiting at the intersection. This information provides the fuzzy logic controller with traffic densities and allows a good assessment of changing traffic patterns. As a result, the fuzzy logic controller can adapt to the uncertainity in the system and change the traffic light accordingly.

1.4 Outline of the thesis

In chapter 2, the basic concepts of Fuzzy Logic and Genetic Algorithms are introduced and the prevailing urban traffic control methods are discussed. The current methodologies for controlling traffic signals are presented in detail and the use of fuzzy logic as a tool for enhancing the current technology is considered. A brief survey of the research on urban traffic control using fuzzy logic is given and some relevant concepts for fuzzy set theory and fuzzy logic based control systems for constructing a FLC are presented. An overview of the three basic operators of a Genetic Algorithm is also presented. Genetic Algorithms are proposed to learn the fuzzy knowledge base for controlling a traffic signal.

In chapter 3, a fuzzy logic controller is developed for controlling the traffic flow approaching an isolated intersection. The traffic flow approaching the intersection is regulated by a set of fuzzy decision rules which adjusts the green phase splits of the north-south and east-west approaches of the signal based on their respective traffic volumes. This control scheme is expected to minimise congestion at the intersection.

In chapter 4, a study of two adjacent intersections is done and the two intersections are coordinated using local fuzzy logic controllers which adjusts the offset at each intersection based on the traffic volume at the adjacent intersection. A new fuzzy control scheme is proposed for coordinating two intersections. In this control scheme, a supervisory fuzzy logic controller is used to adjust the offset at both the intersections. A comparison of the two traffic coordination schemes is made and simulation results are presented.

In chapter 5, the behaviour of a set of three traffic signals is studied. The three intersections are coordinated using local fuzzy logic controllers which adjust the offset at each intersection based on the traffic volume at its upstream intersection. The supervisory fuzzy logic controller introduced in chapter 4 is used to adjust the offset of the three intersections based on the traffic volume at all three intersections rather than just the upstream intersection. The supervisory fuzzy logic controller is expected to perform better than the local fuzzy logic controllers.

In chapter 6, fuzzy logic is integrated with Genetic Algorithms (GAs) to learn the fuzzy knowledge base. The fuzzy control rules generated via genetic evolution are used to regulate the traffic flow approaching two adjacent intersections and a set of three intersections. The fuzzy rules generated by GAs are expected to yield better results than the fuzzy rules generated by hand.

In chapter 7, some conclusions from this research are drawn and discussed and future directions are proposed.

Chapter 2

Urban Traffic Control, Fuzzy Logic and Genetic Algorithms

2.1 Introduction

The demand for transportation has increased over the last few decades. A majority of this increase is due to the spurt in personal transport which is as a result of urbanisation. The increase in the vehicular traffic has brought many problems like pollution, increased accident rates and congestion thereby reducing the efficiency of the transportation system (Patriksson M., 1994).

The increase in the number of vehicles on the road has brought into light the problem of controlling the traffic flow and optimising a strategy of control. Many difficulties arise when attempts are made to model the behaviour of road traffic and evolving an effective means of control. Some of the theoretical problems are the inherent randomness of traffic movement which in itself depends on how the drivers adapt to various conditions and the control variables that affect the modelling of the control strategy. One of the practical problems is the difficulty and cost of collecting data and analysing it to prove that a particular theoretical model achieves the desired results (Robertson D.I., 1979).

Traffic planning, management and control are processes that are linked to certain decisions which have to be made based upon some basic input data which might include travel time, travel costs, queue length of vehicles, etc. In some cases, the input data is precise and available and, assuming that an adequate model exists, satisfactory solutions can be expected from the resulting decisions.

Most of the traffic and transport parameters are characterised by ambiguity, uncertainty, subjectivity, and imprecision and some sort of a model has to be developed to satisfactorily deal with these factors and evolve an optimal solution for complex traffic and transportation processes (Teodorovic D., 1994).

Fuzzy set theory is a convenient mathematical device for treating uncertainty, subjectivity, indetermination, and ambiguity. It is a theory about vagueness and uncertainty. Fuzzy logic enables ill-defined concepts to be used for ill-defined situations. Since its introduction, fuzzy logic has been successfully applied to the control of a wide variety of ill-defined complex industrial processes which require complicated mathematical models (Kickert, W.J.M., et al, 1976, King P.J., et al, 1977, Yasanobu S., et al, 1985, Fujitec F., 1988, Bernard J.A., 1988).

However, fuzzy logic control is not without its limitations. One problem is obtaining an adequate rulebase for the fuzzy logic controller. Rule-elicitation can be performed by interviewing operators, on-line verifications of control actions, etc, but this can be an expensive and a lengthy process and is specific to each application. To overcome this problem, Genetic Algorithms can be used to generate the fuzzy rules for the application (Mohammadian M., et al, 1994, Karr C.L., 1991).

Genetic Algorithms (GAs) are search algorithms that are based on the principles of biological evolution. They simulate the natural search and selection process associated with natural genetics. They are a class of optimisation procedures whose mechanics are based on those of genetics.

GAs have been widely used in many different applications (Caldwell C., et al, 1991, Karr C.L., 1991, Koza J., 1992) and have successfully been applied in the tuning of membership functions of a fuzzy logic controller (Mohammadian M., et al, 1993) and in the generation of fuzzy decision rules (Mohammadian M., et al, 1994). In this research, we propose to use Genetic Algorithms to generate the fuzzy control rules for adjusting the offset of a traffic signal.

The number of fuzzy rules depend upon the number of input variables to the fuzzy logic controller. An increase in the number of input variables results in an exponential rise in the number of fuzzy rules. A ruleset having high dimensionality is difficult to construct by hand. Moreover, the randomness and unevenness of certain non-linear systems makes it difficult to choose an appropriate control action for a possible set of input values.

Hence, to facilitate the construction of knowledge bases, Genetic Algorithms is employed to learn the fuzzy rules. GAs performs a random search in the output fuzzy regions to evolve a knowledge base for the fuzzy logic controller. Each set of fuzzy rules generated by the Genetic Algorithms is evaluated by the fuzzy logic controller based on a system specific performance criterion. In this thesis, GAs is employed to elicit the fuzzy rules for the supervisory fuzzy logic controller adjusting the offset.

2.2 Urban Traffic Control

2.2.1 *Traffic control*

The transportation system is very complex, and its performance depends on many facets of the day-to-day life. The process of evaluating, designing and managing such a system cannot be carried out without the aid of well formulated models. The transportation system as a whole is modelled based on a set of assumptions, the most important ones being that the travel patterns are tangible, stable, and predictable (Patriksson M., 1994).

Traffic control is an intensive technique to promote safe, efficient and convenient movement of people and goods, making a better use of the existing roads (Gartner N.H., et al., 1983). Traffic control can be categorised into three sub-areas: congestion control, incident detection, and traffic light control.

Congestion in road networks has been one of the barriers faced in the improvement of road traffic control. This is due to a lack of understanding of the dynamic behaviour of the traffic system as a whole. The problem with congestion is that it can occur unexpectedly, requiring a change in the traffic control strategy to cope with it. But the advent of fast methods of communication and calculation has created many new opportunities for controlling traffic on congested networks (Smith M.J., et al, 1992).

Incident detection is the capability of the system to classify some congestion phenomena. Congestion could be due to the occurrence of road accidents, or some other incidents (Bielli M., 1991).

Traffic light control is widely used to resolve conflicts among vehicle movements at intersections. The main objective is to reduce the confusion generated by different drivers and improve the safety and comfort of the road users. Traffic signals were first used simply as a means of avoiding collisions and reducing traffic delays at junctions, but, over the years, they have become one of the most effective, flexible and readily available means of controlling road traffic in an urban road network (Smith M.J., et al, 1992).

2.2.2 *Road Traffic Signals*

An Area Traffic Control (ATC) system consists of a number of traffic signals linked in such a way that any signal timing change is in some way dependent upon conditions prevailing at any of the other intersections. The system of signals may be a single linked pair, a linear group or a complete network. The control system at each signalised intersection consists of the following three control elements: cycle time, phase splits and offset (Luk J.Y.K., 1984).

Cycle time is the duration of completing all phases of a signal; phase split is the division of the cycle time into periods of green phase for competing approaches; and offset is the time difference in the starting times of the green phases of adjacent intersections.

Traffic control systems can be grouped into two principal classes: *fixed-time* and *vehicle-actuated* systems.

Fixed time control

A fixed time control system relies on historical data to prepare timing plans for a signalised area. Three to four plans, representing the a.m peak, p.m peak and off-peak conditions are commonly used and a particular plan is switched into operation depending on the time of the day. Vehicle detectors are not required with this method and the coordination of intersections is achieved by linking local controllers to a master controller by means of cables. The master controller adjusts the offset of the local traffic signals to minimise the number of vehicles waiting at the local intersections. A fixed time system can also be implemented in the form of a cableless linked system with the use of crystal clocks in the local controllers.

A fixed time system is simple in structure. It is, however, inflexible in its operation because it cannot respond adequately to unpredictable changes in the traffic demand and is only suited to networks with predictable flow patterns (Luk J.Y.K., 1984).

Vehicle actuated control

A vehicle actuated system, also called traffic responsive control system calculates the control parameters according to the prevailing traffic condition. The change in the signal is influenced by the traffic flow. In this control strategy, one or more vehicle detectors are installed on the approaches to the junction and the green split is adjusted accordingly based on traffic flowing over the detectors. The logic of control is based upon the detection of time gaps in the stream of traffic that is receiving the green. When a gap of several seconds is detected between vehicles, the green phase for that approach is terminated and displayed for another approach (Robertson D.I., 1979).

The traffic-actuated signals are widely used and can provide considerable advantages over fixed-time control. But, most of the fixed-time and traffic-responsive control systems are aimed at short-term effects or to a certain degree, medium-term. These strategies strive to minimise the delay at a single junction and do not optimise the traffic flow of the entire network. There are many forms of implementation of traffic responsive methods having various levels of traffic adaptability. The most notable of these are SCATS (Sydney Coordinated Adaptive Traffic System) developed in Australia and SCOOT (Split, Cycle and Offset Optimisation Technique) developed in the U.K. These methods will be discussed in detail later in this chapter.

Control strategies like SCOOT and TRANSYT (A Traffic Network Study Tool) are concerned with the short-term and medium-term effects and seek to minimise the delay for the network as a whole (Smith M.J., et al, 1992). There is currently no way of dealing with long-term effects since it is very hard to specify with any precision the longer term network wide effects of any control change. Thus, there has been more emphasis on short-term and medium-term optimisation of signal controlled junctions and networks.

2.2.3 *Traffic control systems classification based on architecture philosophy*

Traffic control systems can be classified into the following categories based on their hardware characteristics (Bruno G., et al, 1994):

Non-computerised systems - The early traffic control strategies were operated by electromechanical devices which allowed only fixed-time signal changes to the control of a single junction or an arterial system.

Centralised computerised systems - With the advent of computer systems, the collection and processing of large amount of data was conceivable so that traffic control plans for different areas could be designed. A central computer system gathers traffic data coming from detectors and local controllers and adopts a control strategy for signal plan selection or modification.

Distributed computerised systems - The inability of the centralised computer systems to perform fully traffic responsive control has resulted in the design of distributed computer systems. In these systems, the central computer plays the role of a supervisory controller. The advantages of a distributed computerised system are that the cost of data transmission is reduced and the system as such is more flexible.

2.2.4 *Signal timing parameters*

The optimisation of traffic signal systems timing involves the coordination of the network as a whole. This optimisation is carried out using a three step sequential decision process (Gartner, N.H., et al, 1976).

In the first step, the *cycle time* is calculated based on the requirements of most loaded junctions. In the second step, the *green splits* for the junctions are calculated based on the master cycle which is fixed. Finally, in the third step, a set of optimal *offsets* among signals is determined.

2.2.5 *Coordination of a Network of Intersections*

Traffic signal coordination is one of the most widely used and cost effective means of improving the traffic flow in a network of intersections. The signals at two or more intersections are coordinated on a common cycle time and the offsets are adjusted in such a way that the vehicles passing one intersection arrive at the downstream intersection when the light is green. As a result, the vehicles arriving at the downstream intersection pass through unstopped.

2.2.5.1 *Traffic signal coordination of fixed time plans*

Fixed time plans use preset values to calculate the signal timing, based on previous observations, on the average traffic behaviour over the period of control. As a result, separate fixed time plans are derived for different hours of the day. Different methods use different techniques to optimise the signal timing. Whiting and Hillier (Hiller J.A., 1965) developed a systematic procedure, called the *combination method*, for minimising the total delay in the network of signals. The traffic flow from all sources entering the street is taken into account and the timing signal at the downstream intersection is calculated as a function of the offset along the street. Optimum offsets between the signals are determined by a dynamic programming procedure that finds the best offsets in the network.

Some of the other methods of signal optimisation are *SIGOP* (SIGOP 1966), *MITROP* (Gartner et.al, 1974), and *TRANSYT* (Robertson D.I., 1969). In SIGOP, an ideal offset is calculated for each street that depends only on the conditions along that street. The signal offsets are adjusted by a search procedure that minimises the sum of the squares of the differences between the ideal offsets and the actual offsets. This method is quite efficient but does not guarantee a reduction in the delay time.

In MITROP, the traffic flow along a street is assumed to occur as a single platoon with constant density and the offset is determined by the signal timing at the upstream intersection. A mixed-integer linear programming technique is used to optimise the signal offsets, green time and cycle time simultaneously.

All the three methods (combination method, SIGOP, and MITROP) sacrifice realism in their model to achieve efficiency in their optimisation procedures. TRANSYT, on the other hand, takes into account the different speeds of individual vehicles. The delay at a signal is a non linear function of the signal timings of all upstream intersections. However, the signal optimisation procedure employed is a simple form of rectangular search which might not find the optimal offset. But, the TRANSYT traffic model achieves an improved accuracy when compared to the other optimisation techniques, even though its search procedure is not very efficient. (Robertson D.I., 1979).

2.2.5.2 *Coordination of Traffic Responsive Methods*

Two dynamic traffic responsive methods which are currently in operation are reviewed here.

Sydney Coordinated Adaptive Traffic System (SCATS)

In a SCAT system (Luk J.Y.K., 1984), the signals in an area are coordinated by dividing the area into smaller subareas of about one to ten signalised intersections. Each of these intersections share a common cycle time which is updated every cycle in steps of up to six seconds depending on the Degree of Saturation (DS) of that area. Degree of saturation gives an indication of the effectiveness of the green phase. It is measured using detectors at the stop lines.

Each intersection within a subarea has four phase split plans and five offset plans. The phase split plans express the green times and intergreens which are percentages of the current cycle time. These plans also include other vehicle actuated control tactics for phase skipping, transfer of spare time, and defining phases that benefit from additional time gained by increase in cycle time.

The five offset plans comprise internal and external offsets. The internal offsets between adjacent intersections vary according to the current cycle time and an input parameter known as progressive speed factor which governs the percentage change in the offset. The external offsets are used for merging two subareas. When two adjacent subareas are merged, the common cycle time for the combined area is the larger cycle time of the two separate subareas before merging (Sims A.G., 1979).

In SCAT, the determination of the cycle time, phase split and offset are independent of each other although they are all affected by the degree of saturation. However, in order to minimise the delay and the number of stops, all these three signal timing parameters should be optimised simultaneously. Hence, this approach is not ideally suited for large networks where the traffic flow is high and unpredictable (Luk J.Y.K., 1984).

Split, Cycle and Offset Optimisation Technique (SCOOT)

The technique of optimisation in SCOOT is similar to that of TRANSYT. TRANSYT, being a fixed time system, has a cyclic traffic flow, where the signal settings are the same for every cycle and the flow at each intersection is assumed to remain unchanged for a given time period. SCOOT, on the other hand, measures the traffic flow in real time with the help of vehicle detectors. These detectors are located at a small distance away from the upstream intersection to obtain a count of the number of vehicles waiting at the intersection. The degree of saturation of an approach is estimated from measured flow upstream of the stop line, and a predetermined value of saturation flow. The queue length at the downstream intersection is also predicted from the preset saturation flows.

The queue length in the network is minimised by increasing or decreasing the phase splits and the offset. If the area is heavily congested, the cycle time is also adjusted to minimise congestion. When the traffic flow at a particular intersection is low, SCOOT operates at half its common cycle time thus reducing the delay time of vehicles (Luk J.Y.K., 1984).

Hunt (Hunt P.B., et al, 1981) suggested that SCOOT is most effective when the traffic flow at a junction approaches the capacity, where the demand is unpredictable and the distance between the junctions is short.

2.2.6 *Applications of AI and KB systems to Traffic Control*

In this chapter, some current control strategies were introduced and the systems that are in prominence for controlling traffic in urban areas were described. In this section, some of the limitations of the current traffic control methods will be discussed and the issues addressed by Artificial Intelligence techniques to circumvent these problems, will be presented. Furthermore, the benefits of using Artificial Intelligence techniques to the control of traffic signals will be discussed.

2.2.6.1 *Limitations of the current Urban Traffic Control (UTC) systems*

Fixed time and traffic actuated systems have their own capabilities and are suited to different traffic conditions. Depending upon the traffic conditions in a particular area, either one of these two methods will be appropriate to control the traffic flow in that area. However, there are various factors that affect the performance of the conventional traffic control systems (Ambrosino G., et al, 1994).

1. Congestion is one of the most relevant factors that limits the performance of conventional traffic control systems. When overload of junctions occur, short links get blocked and queue lengths increase. In such cases, the conventional systems are unable to cope with these events and some kind of strategic control is necessary for treating different problems simultaneously and making appropriate evaluations and decisions for different areas.

2. Conventional traffic control systems are unable to respond adequately to unforseen changes in the traffic conditions caused by accidents, road blockages, etc. This is because these systems are designed to deal with predefined average traffic flows or react to small changes in traffic flows.

3. The extent of knowledge available to the UTC systems about the actual traffic behaviour is very limited. The current traffic control systems are able to respond to the traffic flows measured by detectors. However, these detectors do not provide any information about the actual traffic behaviour which is essential to optimise the performance of the system.

4. A major drawback of the current traffic control systems is that, most of the control strategies are applied in an isolated manner with no interaction with other traffic management measures.

The purpose of this research is to address the first two limitations and develop a suitable technique using Fuzzy Logic to improve the efficiency and the design transparency of the urban traffic control system.

2.2.6.2 *Issues addressed by Artificial Intelligence*

In order to realise a traffic control system, the following operations are essential (Bielli M., et al, 1991).

- Traffic data collection
- Data Analysis/Interpretation
- Decision and Control

The management and operations implied by these three levels can be recognised as a knowledge intensive task, given the complexity of the traffic phenomenon and decision making that involves expertise and application of rules.

In order to regulate the traffic flow in a network, an analysis of the current traffic situation and an understanding of the network as a whole is essential. The next step is to improve the quality of decision making, that is, to estimate the consequences of possible actions and to adapt the decisions to the traffic situation.

Artificial Intelligence (AI) techniques along with logic programming contribute to fulfil all the requirements that the quantitative methods of operations research have not been able to meet. AI also called knowledge processing uses the models of human reasoning and problem solving and applies this knowledge to construct a solution for the problem. AI techniques are used to manage complex mathematical models, set up the solving algorithm, and solve unstructured problems with the aid of heuristics thus controlling the whole decision process (Bielli M., et al., 1991).

The limitations of the current traffic control systems can be resolved by using new techniques that are based on artificial intelligence principles and sophisticated computing devices (Gegov A., 1994). These systems are called 'Distributed intelligence control systems' (DICS). DICS are composed of distributed intelligent control units, processing information in a distributed manner and possessing both quantitative and qualitative characteristics.

The Automated Urban Traffic Control Systems (AUTCS) operate by means of only quantitative algorithms that do not reflect the qualitative aspects of the transport process. Since qualitative information is usually expressed in uncertainty, it is difficult to formulate a mathematical model for the system considered. For this reason, certain approximate approaches should be used for solving this problem.

Fuzzy logic has been found to be an effective tool in the control of processes that are ill-defined and complex to model (Mamdani E.H., et al, 1981, Sugeno M., 1985). Fuzzy logic is concerned with the formal principles of approximate reasoning. It allows qualitative information to be represented in a quantitative way. Distributed intelligent control systems incorporating fuzzy logic are the appropriate tools for improving the design transparency of the prevailing traffic control systems.

2.2.7 *Fuzzy Logic as a means for Traffic Control*

Zadeh's (Zadeh L.A., 1965) pioneering work on fuzzy sets has provided a conceptual framework for dealing with problems that are vague and imprecise in nature. The theory of fuzzy sets is capable of providing a basis for the modelling and analysis of complex processes, which in many ways is similar to the approach taken by humans, that is, rough approximation.

The problem of controlling traffic junctions is considered as a classical example of nonprogrammed decision making where the decisions are to be made in an environment that lacks well specified means of coping with the problem (Pappis C.P., et al, 1977). A linguistic control algorithm consisting of fuzzy decision rules can be implemented to deal with this problem. The linguistic control algorithm aims to enhance the appropriateness of the control actions, increase control flexibility and produce performance measures which closely match human's perception of 'good' traffic control (Chiu S., et al., 1993).

A considerable amount of work has been done on the problem of regulating traffic at intersections. Automatic control of traffic signals has been dealt by many authors worldwide and many theoretical papers about the optimal control of signals have been published (Pappis C.P., et al, 1977, Favilla J., et al, 1993).

The first paper that described an attempt to solve this problem using fuzzy logic was written by Pappis and Mamdani (Pappis C.P., et al., 1977). They considered an isolated signalised intersection of two one-way streets and developed a model based on linguistic control algorithm. Pappis and Mamdani assumed an uniform distribution of vehicles arriving at the intersection. They also assumed that the cycle is divided into two periods of 'actual green' and 'actual red', that vehicles leave the queue at the same intensity at which they join it and there is no turning traffic. The arrival of a vehicle is decided by generating a random number and comparing it with the mean vehicle arrival rate.

The control algorithm developed consists of three input variables and a single output variable. The input variables are:

T - The time that has lapsed since the last light change at the intersection.

A - The number of vehicles that passed through the intersection during the previous green phase.

Q - The number of vehicles waiting in line on the one-way street waiting for the light to change to green.

The output variable is:

E - The extension which has values identical to variable ***T***, representing the extension given to the present state of the system.

The variables ***A*** and ***Q*** are assigned linguistic values like '*many*' vehicles, '*more than*' vehicles, '*few*' vehicles. The variables ***T*** and ***E*** are assigned values like '*very short*', '*short*', *medium*'. The control algorithm developed by Pappis and Mamdani consists of rules of the following type:

If ***T*** is *very short* and ***A*** is *many* and ***Q*** is *medium* then ***E*** is *medium.*

A total of 25 rules is used and each rule is a fuzzy relation between ***T***, ***A***, ***Q*** and ***E***. Every ten seconds, a set of five control rules are evaluated ten times for each of the ten seconds in order to determine the extension to the present state of the system. It is assumed that the detecting pads are sufficiently far away from the junction, so that data is available for each of the next ten seconds.

A comparison of the results from the model based on fuzzy logic with results from the classical approach based on stochastic models to controlling a signalised traffic intersection indicated that better results are achieved by the model based on fuzzy logic from the viewpoint of average time loss per vehicle.

Nakatsuyama, Nagahashi and Nishizuka (Nakatsuyama M., et al., 1984) developed a fuzzy logic controller based on the model proposed by Pappis and Mamdani for regulating traffic flow approaching two adjacent intersections in a one-way street. They developed a fuzzy logic phase controller to coordinate two consecutive east-west intersections along a north-south arterial road. The fuzzy logic phase controller determines the offset, which is the time difference between the start of the green phases of the two intersections, using a different set of fuzzy rules.

This model developed by Nakatsuyama, Nagahashi and Nishizuka is based on the traffic conditions. The traffic signal at the first traffic junction is controlled by a fuzzy logic controller and the traffic signal at the second junction is controlled by either the fuzzy logic controller or the fuzzy logic phase controller depending on the traffic density. The fuzzy logic phase controller is effective in coordinating the traffic flow between the two successive traffic junctions only when the traffic flow is large. It is not very effective when the flow is very small or very large.

Nakatsuyama et al. compared their model with a standard vehicle actuated controller for different values of traffic flow rates and showed that the combination of fuzzy logic controller and fuzzy logic phase controller achieves considerably shorter average delay times than a vehicle actuated controller.

Favilla, Machion and Gomide (Favilla J., et al., 1993) developed a Fuzzy Traffic Controller (FTC) which includes a fuzzy logic controller, a state machine and an adaptive module. The adaptive module comprises two adaptive strategies: a statistical adaptive and a fuzzy adaptive strategy.

The fuzzy logic controller compares the incoming traffic at the approach that has the green phase with the vehicle queue at the other approaches. On the basis of this information, it decides whether or not to extend the current green phase. The input variables to the fuzzy logic controller are the ***Arrival*** of vehicles in the approach that has the green light and the ***Queue*** of vehicles in the approach that has the red light. The output variable of the fuzzy logic controller is the ***Extension*** to the current green phase. The fuzzy logic controller has a total of eleven rules.

The main objective of the adaptive module is to optimise the fuzzy traffic controller's performance in a broader ranges of traffic situations. In the statistical adaptive strategy, the membership functions of the input variables are adjusted. During 18 consecutive intervals of 10 seconds each, the vehicle arrival is added up for each lane of each approach and the lane having the maximum value for each approach is stored. Then the average is calculated along with standard deviation and the membership functions of ***Arrival*** and ***Queue*** are updated.

In the fuzzy adaptive strategy, the membership functions of ***Extension*** is adjusted by employing another fuzzy logic controller which has as its inputs the ***residual queue*** at the end of the green phase and the ***queue variation*** during the green phase. The output is the adjustment to the upper limits of the membership functions of ***Extension.***

The results obtained by simulating the traffic flow with and without adaptive strategies showed that the fuzzy adaptive schemes perform better than statistical adaptation. By using the adaptive schemes, the average delay is reduced and the FTC becomes more responsive to traffic flow characteristics. A comparison of the proposed decision making logic with the one proposed by Pappis and Mamdani (Pappis C.P., et al., 1977) indicated a reduction in the queue length with the employment of the fuzzy traffic controller.

Another fuzzy logic controller for regulating traffic approaching a single intersection in a two-way street was designed by Kelsey and Bisset (Kelsey R., et al., 1993). The fuzzy logic controller has three inputs - ***the average density of traffic behind the green light, the average density of traffic behind the red light, and the length of the current green time***. The traffic densities are obtained from two sensors placed on the road. One is placed at the intersection and the other placed 150 feet from the traffic light. This gives a count of the number of vehicles waiting. The output of the controller decides whether to change the phase from green to red or remain the same. The fuzzy logic controller uses 26 fuzzy control rules that are invoked every second.

In order to model the traffic flow to mimic reality, Kelsey and Bisset used physical equations to describe the motion of a car based on the car in front of it. They used a time-delay differential equation derived from traffic flow theory and standard classical physics equations for determining the velocity and position of an object based on the object's acceleration.

The simulation results showed that the fuzzy logic controller allows more cars to pass through than the conventional fixed time and proximity controllers. It improved the throughput and reduced the average waiting time of vehicles indicating that cars spend less time waiting and more time moving. The overall cost of the traffic control system was also reduced.

Hoyer and Jumar (Hoyer R., et al., 1994) developed a fuzzy logic controller for a two-way arterial road with multiple state control instead of two state control, that is, instead of being restricted to two main directions of traffic flow, they also considered turning traffic.

The fuzzy logic controller operates with ten input variables and two output variables. The inputs to the controller are the '***traffic densities***' of different lanes and the '***time elapsed***' since the last state change. The outputs are the '***extension***' to the current green time and the '***selection of the next state***'.

Hoyer et al. considered an intersection with four approaches having 12 main directions. They used six states to control the 12 directions of traffic flows. Depending on the traffic volume, the states are switched into operation. If there is little turning traffic, the states controlling the turning traffic are not switched on and the turning vehicles have to give way to the oncoming vehicles. The fuzzy logic controller is constructed using 72 rules. The effectiveness of this scheme is yet to be investigated.

Nakamiti, Freitas, Prado and Gomide (Nakamiti G., et al., 1994) introduced a Fuzzy Distributed Artificial Intelligence (FDAI) approach to control a network of urban traffic lights. The fuzzy distributed traffic light control system consists of a processor (fuzzy logic controller) situated at each intersection deciding on the setting of its local traffic light and communicating with its neighbour processor only. The readings from the sensors are sent to the local processor whose aim is to optimise the traffic flow and reduce the average queue length and the delay time of vehicles.

Each processor comprises of a local problem solver and a case-based mechanism. The local problem solver reasons upon the processor's knowledge and decides whether to alter its current state. It also decides whether to send any message to its neighbour processors. The case based mechanism helps the local problem solver to analyse the current situation and verify similar past cases. A decision is reached by the problem solver which is fed back to the case based mechanism and this information is stored for future reference.

The exchange of information between the processors is in natural language. A decision is reached based on the processor's local setting, past information and the received messages. This system is still under study and one of the critical points, according to the authors, is to achieve consensus and coordination among the processors.

Chiu and Chand (Chiu S., et al., 1993) proposed a distributed approach to traffic signal control where the signal timing parameters at each intersection are adjusted based on the local traffic condition and on the signal timing parameters at adjacent intersections. They present a distributed system of cooperative fuzzy logic controllers where each local fuzzy logic controller uses a set of fuzzy decision rules to adjust the cycle time, phase splits and offset.

A set of 46 control rules is used to adjust the signal timing parameters. The rules are divided into three decoupled groups: 25 rules for adjusting the cycle time and phase splits, 18 rules for adjusting offset, and three rules for determining constraints on the cycle time value so that coordination is possible.

The inputs to the fuzzy logic controller for adjusting the cycle time and phase split are the '***highest degree of saturation among the east-west approach***' and the '***highest degree of***

saturation among the north-south approach'. The outputs are the '***adjustment to the current cycle time***' and '***adjustment to the east-west green phase***'. The degree of saturation is given by the ratio of the actual number of vehicles that passed through the intersection during the green period to the maximum number of vehicles that could have passed through the intersection during the same period. The rules are evaluated at every phase change and the maximum adjustment to the cycle time and green phase is 20% of the current value.

The offset is adjusted by another fuzzy logic controller which determines the dominant direction of traffic from the vehicle count for each approach and then determines the upstream intersection it wishes to be coordinated. The inputs to the fuzzy logic controller are the '***difference between the traffic volume in the dominant direction and the remaining directions***' and '***the required adjustment***', which is the amount by which the current green phase is to be shortened/extended divided by the current green period. The output is the '***allowable adjustment to the current green phase***'.

The set of three rules determines the allowable difference between the local cycle time and that of the upstream intersection based on the vehicle volume at the local intersection. By enforcing a common cycle time when the traffic flowing from the upstream intersection to the local intersection is high, it is possible to minimise the number of stops at the local intersection.

A network of intersections is used in the simulation and the results indicate a significant reduction in the average waiting time per vehicle and in the number of stops per minute.

In this research, a fuzzy logic control scheme to regulate the traffic flow approaching a set of intersections, is presented. Instead of using local fuzzy logic controllers, as proposed by Chiu and Chand, to coordinate each intersection with only the upstream intersection, a single supervisory fuzzy logic controller is proposed, which coordinates the intersections based on the traffic flow from all directions. As a result, the offset at each intersection is adjusted based on the traffic volume at all its neighbouring intersections rather than just the upstream intersection. Also, instead of using proximity sensors, sensors that give an estimate of the queue length are used.

The input variables to the fuzzy logic traffic controller are ***ratio of queue length to number of vehicles that passed through the intersection during the previous green phase, in the north-south approach*** and ***ratio of queue length to number of vehicles that passed through the intersection during the previous green phase, in the east-west approach.*** The output variables are ***green phase adjustments to the north-south approach*** and ***green phase adjustments to the east-west approach.*** This fuzzy control scheme is aimed at improving the overall performance of the system.

The supervisory fuzzy logic controller adjusting the offset of the three intersections, consists of three input and three output parameters, thereby making it difficult to determine an appropriate control action for a given set of input conditions. To overcome this problem, Genetic Algorithms are used to learn the fuzzy rules for adjusting the offset.

2.3 Fuzzy logic and Fuzzy Logic Controller

The concept of fuzzy sets was introduced in 1965 by Lotfi Zadeh (Zadeh L.A., 1965) as a means of representing vagueness in applications. He suggested a modified set theory in which an individual can have a value that ranged over a continuum of values instead of being either 0 or 1. Fuzzy set theory is an extension to traditional set theory and fuzzy logic is the corresponding logic to manipulate the fuzzy sets.

Fuzzy logic attempts to model computer reasoning on the kind of imprecision found in human reasoning. Through fuzzy logic, a system not only can represent imprecise concepts such as *Fast*, *Tall*, etc, but through a set of sound mathematical principles, it can also use these concepts to make deductions about the system. Fuzzy logic aims to model imprecise reasoning or common sense reasoning for uncertain, ill-defined, and complex processes which do not require a high level of precision.

A Fuzzy Logic Controller (FLC) uses fuzzy logic to determine the course of action. It provides an algorithm which converts the linguistic control strategy based on expert knowledge into an automatic control strategy. The process is controlled by linguistic variables rather than crisp numerical variables. FLCs are an attractive option when the process to be controlled is ill-defined and normally requires a skilled human operator. Over the past few years, fuzzy logic control has been widely applied to a variety of control problems and has been found to be a good alternative to conventional control methods. Some of the applications include cement-kiln process control (Mamdani E.H., et al., 1981), robot control (Tanscheit R., et al., 1988), image processing (Kandel A., 1982), and automatic train operation (Sugeno M., 1985).

2.3.1 *Fuzzy set theory*

2.3.1.1 *Fuzzy sets*

Fuzzy set theory is an extension to the classical set theory. As with classical sets, fuzzy sets are defined over an universe of discourse. For a given universe of discourse U, a fuzzy set is determined by a membership function which maps elements of U on to a membership range which is usually in the range [0,1].

Let U be a collection of objects denoted by {u} where 'u' represents the generic element of U. A fuzzy set A in the universe of discourse U is characterised by a membership function $\mu_A(u)$ which maps each element of U to a real number in the interval [0,1], namely (μ_A: U -> [0,1]). The membership function represents the grade of membership of u in A.

The fuzzy set A can thus be represented as :

$$A = \{ (u, \mu_A(u)) / u \in U \}$$

When U is continous, the fuzzy set A is represented as :

$$A = \int_u \mu_A(u) / u$$

When U is discrete, A is represented as :

$$A = \sum \mu_A(u_i) / u_i$$

A fuzzy set can be considered to be a generalisation of an ordinary set, such that in an ordinary set, an element will have a membership function $\mu_A = 0$ or 1. In the classical set theory, an element either belongs to or does not belong to a set but, the elements belonging to a fuzzy set show a gradual transition from membership to non-membership. Thus, fuzzy sets allow an element in the set to have a degree of membership of any real value between zero and one which is called the membership value. This value determines to what degree an element belongs to a set.

2.3.1.2 *Support set, Crossover point, Fuzzy singleton*

The support set of a fuzzy set A is the crisp set of all points 'u' in U such that $\mu_A(u) > 0$.

An element 'u' in U at which $\mu_A(u) = 0.5$ is called a crossover point.

A fuzzy set whose support is a single point in U is called a fuzzy singleton.

2.3.1.3 *Fuzzy set operations*

All the normal set operations can be defined on fuzzy sets. Let A and B be two fuzzy sets in X with membership functions μ_A and μ_B. The traditional set theory operations of union, intersection and complement of classical subsets of X can be extended for fuzzy sets via their membership functions (Yan J., et al, 1994).

Union :

$$\mu_{A \cup B}(u) = \max \{ \mu_A(u) , \mu_B(u) \} \text{ for all } u \in U$$

Intersection :

$$\mu_{A \cap B}(u) = \min \{ \mu_A(u) , \mu_B(u) \} \text{ for all } u \in U$$

Complement :

$$\mu_{\bar{A}} (u) = 1 - \mu_A (u) \text{ for all } u \in U$$

2.3.2 *Linguistic Variables and values*

The term linguistic variable (Zadeh L. A., 1965) denotes a variable defined in a universe of discourse and taking on some value such as *small, large,* etc. These linguistic values which are represented in natural language are called as linguistic values or primary terms. These linguistic values are modelled by fuzzy sets. Each linguistic variable involves a finite collection of primary terms. Another important aspect of fuzzy sets is the concept of linguistic hedges such as *slightly, very, more or less.*

For example, a linguistic variable 'Pressure' can have values such as *very high, high, slightly high,* and *not very high.* Thus, a linguistic variable is defined by both primary terms and linguistic hedges. The linguistic hedges resemble the concentration operation (*very high*), dilation operation (*slightly high*) and complement operation (*not very high*). It introduces into the system a shade of vagueness which makes it possible to model human decision making process.

2.3.3 *Fuzzy Logic*

Fuzzy logic is the logic corresponding to fuzzy sets. In classical two-valued logic, or boolean logic or binary logic, a proposition is either true or false. The only permitted membership values are 0 or 1. Every item in the universe of discourse is either a full member of the set or not a member at all. Two valued logic works well for problems which are linear and systems that can be modelled precisely and it has proved to be effective in solving such problems. In mutivalued logic, a proposition may be true, false or have an intermediate truth value. The set of truth values is assumed to be evenly divided over the interval [0,1]. In fuzzy logic, the membership function can have values ranging from 0 to 1 (Yan J., et al., 1994).

2.3.4 *Fuzzy Inference Rules*

A fuzzy inference rule or a fuzzy relation is often expressed through the conditional logic structure '*If-then*'. They are of the form "*If* A *then* B", where A and B are fuzzy sets characterised by appropriate membership functions. These rules tend to capture the impreciseness shown in human reasoning when having to make a decision in an environment of uncertainty and imprecision. For example, a fuzzy rule for controlling current in the compressor inside an airconditioning unit could be written as '*If* temperature is *high* and humidity is *low*, *then* supply *moderate* amount of current'. The *If* clause is called the antecedent and the *then* clause is called the consequent. Such rules are generally obtained from, and reflect the experience and the knowledge of human experts.

Another form of fuzzy *If-then* rules, proposed by Takagi and Sugeno (Takagi T., et al., 1983), shows the involvement of fuzzy sets only in the premise part. An example of a fuzzy rule using Takagi and Sugeno's fuzzy inference rule can be given by:

If velocity is *high then* force = k * (velocity)

where *high* in the antecedent part is a linguistic value represented by a membership function. The consequent is a non-fuzzy equation, that is, the output variables are crisp.
For the computer implementation of a fuzzy rule, the linguistic values *high, low* and *moderate* must be mapped to numerical values. Fuzzy set theory allows these terms to be defined through membership functions and assigns these qualitative values to fuzzy sets.

2.3.5 *Membership functions*

Each linguistic value is characterised by a membership function. There are two ways to define the membership function of fuzzy sets: numerical and functional. A numerical definition expresses the degree of membership function of a fuzzy set as a vector of numbers whose dimension depends on the number of discrete elements in the universe of discourse. A functional definition defines the membership function of a fuzzy set in an analytical expression where the degree of membership for each element is calculated (Lee C.C., 1990).

Certain standard shapes of membership functions are commonly used for representing the fuzzy sets based on the universe of discourse. The membership functions commonly used are: (a) S-function, (b) π-function, (c) triangular form, (d) trapezoid form and (e) exponential form (Yan J., et al., 1994). The triangular form and the trapezoid form are most widely used for determining the degree of membership.

Triangular membership function

The membership functions have a triangular shape whose precise appearance is determined by the values a,b,c as shown in Figure 2.1. 'a' and 'c' are the lower and upper limits of the fuzzy sets and 'b' is the average of 'a' and 'c'. For an element x, the membership function is defined as follows:

$$\mu_A(x) = 0 \ , \ x < a$$

$$\mu_A(x) = (x - a) / (b - a) \ , \ a \leq x \leq b$$

$\mu_A(x) = (c - x) / (c - b) \quad , \; b \leq x \leq c$

$\mu_A(x) = 0 \quad , \; x > a$

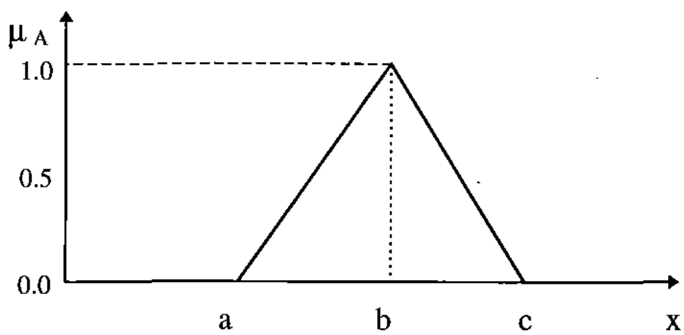


Figure 2.1 Triangular fuzzy membership function

Trapezoidal membership function

A Trapezoidal fuzzy membership function is shown in the Figure 2.2. 'a' and 'd' are the lower and upper limits of the fuzzy sets and the region between 'b' and 'c' always has a membership value equal to one. The membership function for a trapezoidal fuzzy number is given as follows:

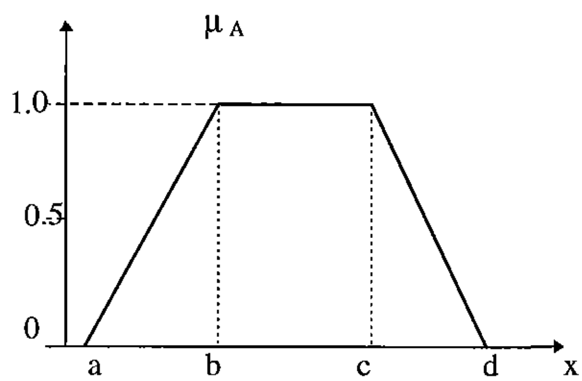


Figure 2.2 Trapezoidal fuzzy membership function

$\mu_A(x) = 0, \quad x < a$

$\mu_A(x) = (x - a) / (b - a), \quad a \leq x \leq b$

$\mu_A(x) = 1, \quad b \leq x \leq c$

$\mu_A(x) = (d - x) / (d - c), \quad c \leq x \leq d$

$\mu_A(x) = 0, \quad x > d$

2.3.6 *Fuzzy Logic Controller(FLC)*

The applications incorporating fuzzy logic have their inputs, outputs and control response specified in terms similar to those that might be used by human operators. Complex mathematical models of the system are not required. The knowledge base constructed, based on the experience of the human expert, is in the form of rules which are easily understandable. Such systems are called as fuzzy inference systems and they are also known as fuzzy models, Fuzzy Associative Memories (FAM), or fuzzy logic controllers when they are used in control problems (Jang J. S. R., 1991).

2.3.6.1 *Basic structure of a Fuzzy Logic Controller*

The main elements of a fuzzy logic controller are the fuzzification unit, the fuzzy knowledge base, the fuzzy logic inference unit, and the defuzzification unit. The basic structure is shown in Figure 2.3 (Yan J., et al., 1994).

1. The fuzzification unit maps the measured inputs, which are in the form of crisp inputs, into fuzzy linguistic values used by the fuzzy reasoning mechanism.

2. The fuzzy knowledge base contains two main types of information: (a) a database defining the membership function of fuzzy sets used as values for each system variable and (b) a rule base which maps fuzzy values of the input to fuzzy values of the output.

3. The fuzzy logic inference unit performs various fuzzy logic operations to infer the control action for the given fuzzy inputs.

4. The defuzzification unit converts the inferred fuzzy control action to the required crisp control value.

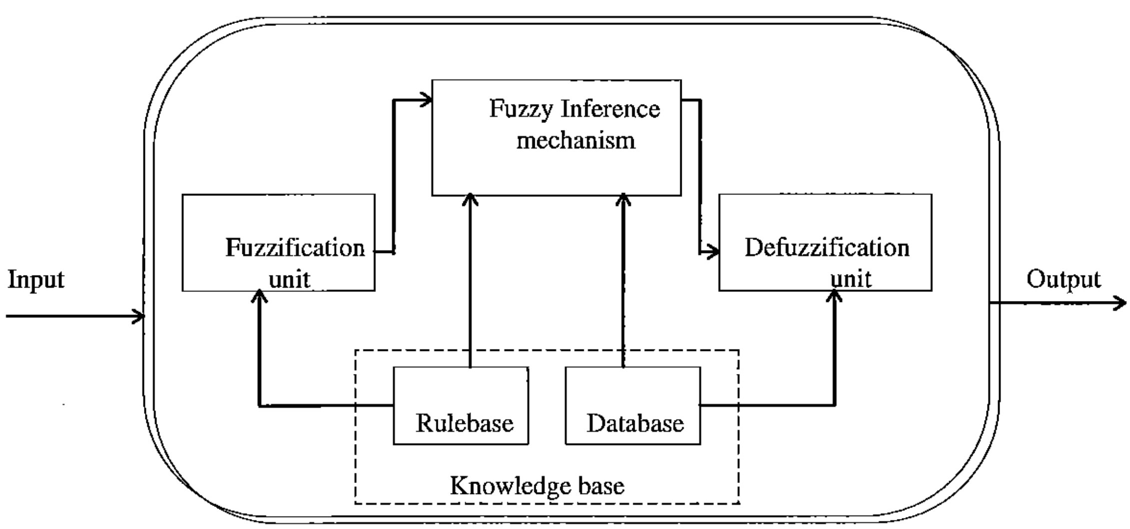


Figure 2.3 Basic structure of a Fuzzy Logic Controller

The system variables are of two types, input variables measured from the control process and output variables used by the FLC to control the process. Depending upon the design objectives, different types of FLC can be constructed. For instance, the FLC may have a fixed number of fuzzy control rules (a static fuzzy knowledge base) or it may have learning capability through modification of the knowledge base (a dynamic fuzzy knowledge base) (Yan J., et al, 1994).

2.3.6.2 *Design and Implementation of a fuzzy logic controller*

In designing a fuzzy logic controller, the following factors should be kept in mind.

1. Identify the input and output variables and their universe of discourse. These variables determine the state of the process and the control actions to be considered.

2. Determine the scale factors of the input and output variables.

3. Define the fuzzy membership functions. These functions are used in setting up the fuzzy sets for the input and output variables.

4. Construct the fuzzy rule base. The rule base gives the relationship between the input and output fuzzy sets.

5. Design an inference mechanism that uses the rule base to logically obtain the control statements for an input value and thereby determine the fuzzy outputs.

6. A defuzzification strategy, for translating the fuzzy output sets to crisp outputs which are applied to the process to be controlled.

2.3.6.3 *System variables and fuzzy parameters*

The system variables or fuzzy parameters, which include the input and output variables, are usually linguistic ie, expressed in natural language, and they take values corresponding to their fuzzy sets. These system variables are different from the input and output values which are crisp in nature with many values in a permitted range.

The design of the fuzzy sets is the critical part of the design. The number of input and output variables varies depending on the complexity of the system. A system with n input and m output variables is called a n-input m-output system. The fuzzy sets for each system variable are defined in linguistic terms such as **PB** (*Positive Big*), **PS** (*Positive Small*), **ZE** (*Zero*), **NS** (*Negative Small*) and **NB** (*Negative Big*). The number of fuzzy sets for each variable determines the number of fuzzy membership function for each variable. The membership function for each fuzzy set is then defined on the universe of discourse of the fuzzy variable. Usually, triangular or trapezoidal membership functions are used as these require less computation time than the other membership functions (Yan J., et al., 1994).

2.3.6.4 *Fuzzification*

Fuzzification is the process of mapping from observed inputs to fuzzy sets in the universe of discourse. In fuzzy control applications, the observed data is usually crisp and hence fuzzification is necessary to map the crisp inputs to the corresponding fuzzy values for the input variables. The mapped data are further converted into linguistic terms as labels for the fuzzy sets defined for the system input variables.

2.3.6.5 *Fuzzy Knowledge base*

The knowledge base of a fuzzy logic controller comprises two components, namely, a database and a fuzzy control rule base. The database defines the fuzzy sets for the system variables with the membership functions defined over the universe of discourse for each variable. The rule base contains the fuzzy control rules intended to achieve the control objectives.

2.3.6.6 *Specification of the rulebase*

The rulebase comprises the fuzzy decision rules for controlling a process. The formulation of the ruleset is comparable to that of an expert system except that the fuzzy rules incorporate linguistic variables which are similar to how a human operates the system. The fuzzy rules are derived from the expert's knowledge and intuition. The rules are in the form: *If* [x_1 is A_1 and x_2 is A_2 and ....] *Then* [y_1 is B_1 and y_2 is B_2 and ....] where A_i and B_i are the input and output fuzzy sets respectively. The incorporation of fuzzy terms gives fuzzy logic its strength.

The number of fuzzy sets of an input variable defines the number of rules required. Usually, five to seven fuzzy sets are chosen for an input or output variable. A fuzzy rule is written for every possible combination that could exist in the system to be controlled. An increase in the number of input variables results in an exponential increase in the number of fuzzy rules. There is no strict formal standard for the structure of fuzzy rules (Yan J., et al, 1994).

2.3.6.7 *Fuzzy Reasoning techniques*

Several types of fuzzy reasoning have been proposed in the literature. Depending on the fuzzy *If-then* rules employed and the types of fuzzy reasoning, an appropriate fuzzy control action is taken. Among the various fuzzy inference methods, the most commonly used are the following (Yan J., et al., 1994):

1. MAX-MIN fuzzy inference method.
2. MAX-DOT fuzzy inference method.

Let us assume a fuzzy control rule base with two rules:

Rule 1 : *IF* x is A_1 and y is B_1 *THEN* z is C_1

Rule 2 : *IF* x is A_2 and y is B_2 *THEN* z is C_2

Let the firing strength of the *i*th rule be denoted by α_i. For inputs x_0 and y_0, the firing strengths α_1 and α_2 of the rule base can be denoted by

$$\alpha_1 = \mu_{A1}(x_0) \wedge \mu_{B1}(y_0)$$

$$\alpha_2 = \mu_{A2}(x_0) \wedge \mu_{B2}(y_0)$$

where $\wedge$ stands for the Intersection or minimum operator.

1. MAX-MIN fuzzy reasoning

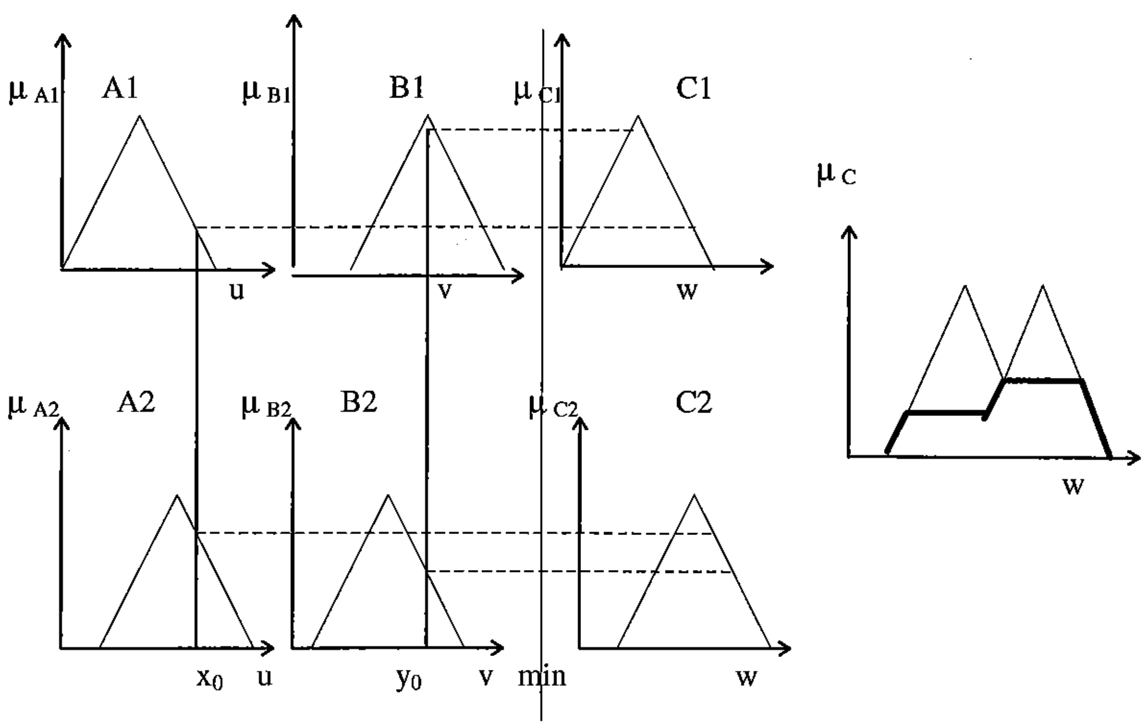


Figure 2.4 MAX-MIN fuzzy inference method

In MAX-MIN fuzzy reasoning, the minimum operation rule proposed by Mamdani (Mamdani E.H., et al, 1981) is used for fuzzy implication. The control decision led by the

ith rule is given by $\alpha_i \wedge \mu_{ci}(w)$ where $\mu_{ci}(w)$ is the membership value of the output fuzzy sets. Thus, the membership grade of the consequent is given by

$$\mu_c(w) = (\alpha_1 \wedge \mu_{c1}(w)) \vee (\alpha_2 \wedge \mu_{c2}(w))$$

where $\vee$ stands for maximum or union operator.

Figure 2.4 shows the MAX-MIN inference process for the crisp inputs x_0 and y_0.

2. MAX-DOT fuzzy reasoning

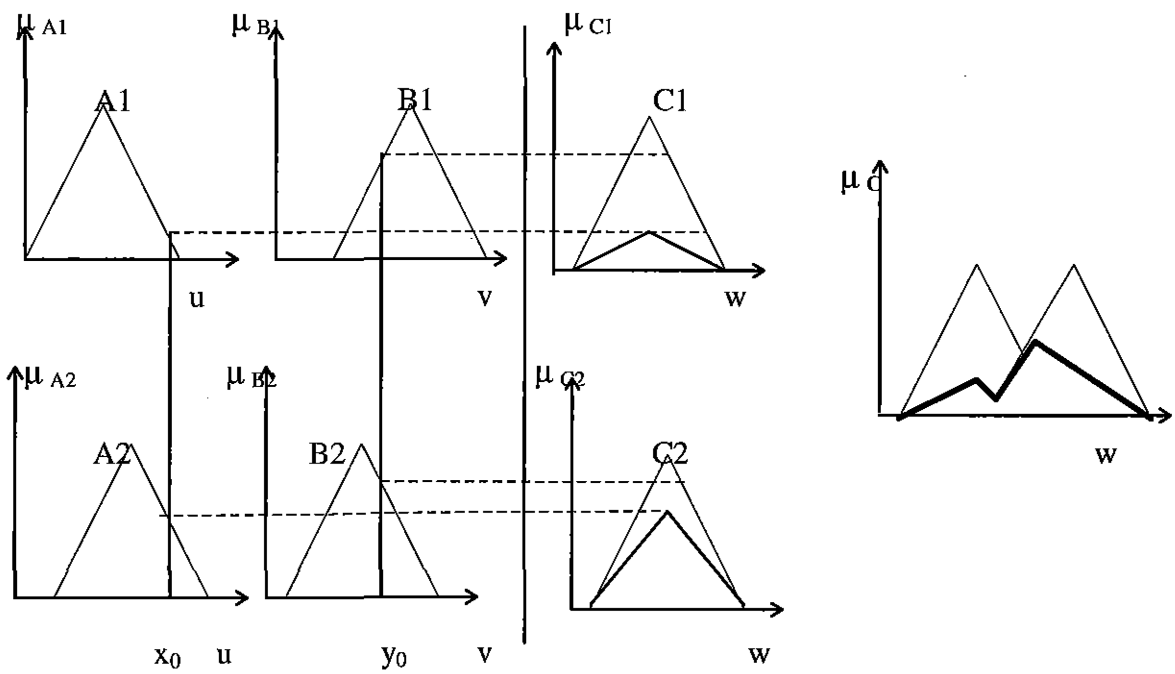


Figure 2.5 MAX-DOT fuzzy inference method

In MAX-DOT fuzzy reasoning, Larsen's product operation rule is used as the fuzzy implication function. The control decision can be expressed as $\alpha_i \cdot \mu_{ci}(w)$. The membership function is given by

$$\mu_c(w) = (\alpha_1 \cdot \mu_{c1}(w)) \vee (\alpha_2 \cdot \mu_{c2}(w))$$

Figure 2.5 shows the MAX-DOT inference process for the crisp inputs x_0 and y_0.

2.3.6.8 *Defuzzification*

Defuzzification is the process of mapping the fuzzy control actions into crisp control actions. The purpose of defuzzification is to produce a nonfuzzy control action that best represents the possibility distribution of the inferred fuzzy control action. There is no systematic procedure for choosing a defuzzification strategy. Three methods that are often applied are described here (Wen-Ruey H., 1993).

1. *Centre of Area (COA) method*

The centre of area method calculates the centre of gravity of the distribution of the control action. For a fuzzy control action with a membership function μ_c, the control action is given by:

$$W = \frac{\sum_{i=1}^{q} \mu_c(z_i) z_i}{\sum_{i=1}^{q} \mu_c(z_i)}$$

where q is the number of quantisation levels of the output, W is the crisp output, $\mu_c(z_i)$ is the membership grade of z_i and z_i is the amount of control action at the quantification level *I*.

2. *Mean of Maximum (MOM) method*

The mean of maximum method generates a control action which represents the mean value of all the control actions whose membership functions reach the maximum. The control action is given by:

$$W = \sum_{i=1}^{n} \frac{\alpha_i H_i W_i}{\alpha_i H_i}$$

where H_i is the maximum height of the membership function of the fuzzy set defined for the *i*th rule output control, α_i is the firing strength of the ith rule and W_i is the crisp control value at which the membership function reaches the maximum H_i, which is usually equal to one.

3. *Centroid Method*

The centroid method is a simplification of the COA method where the control action is the average of the weighted rules.

$$W = \frac{\sum_{i=1}^{n} w_i z_i}{\sum_{i=1}^{n} w_i}$$

where n is the number of rules fired, w_i is the firing strength and z_i is the amount of control action recommended by rule *i*.

2.3.7 *Self Organising Fuzzy Logic Controller*

Fuzzy logic control has been applied successfully to many industrial processes and it has been found to be appropriate when process models are either unknown, non-linear or variable in structure. Since fuzzy control is based on a set of fuzzy rules which are derived from people's experience and common sense, it is sometimes difficult to obtain an adequate rule base for the controller especially when complicated dynamic processes are concerned. Rule elicitation can be performed by interviews with operators, on-line logging of control action, etc. But this is a lengthy process and specific to each application. To overcome this problem, the concept of Self Organising Fuzzy Logic Control (SOFLC) was introduced (Procyk T. J., et al., 1979).

A SOFLC achieves a better performance of the controlled process by developing and improving the fuzzy rules through a learning process and structures itself automatically by monitoring the process's performance on-line, and is thereby able to obtain a predetermined quality output. The learning process of the SOFLC consists of algorithms that allow the controller to assess its own performance on the basis of a set of predetermined rules.

The SOFLC observes the environment while issuing the appropriate control actions and uses the results of these control actions to improve them further, that is, learn from them. The function of the controller is one of system identification and control. (Procyk T. J., et al., 1979).

2.3.7.1 *Structure of a SOFLC*

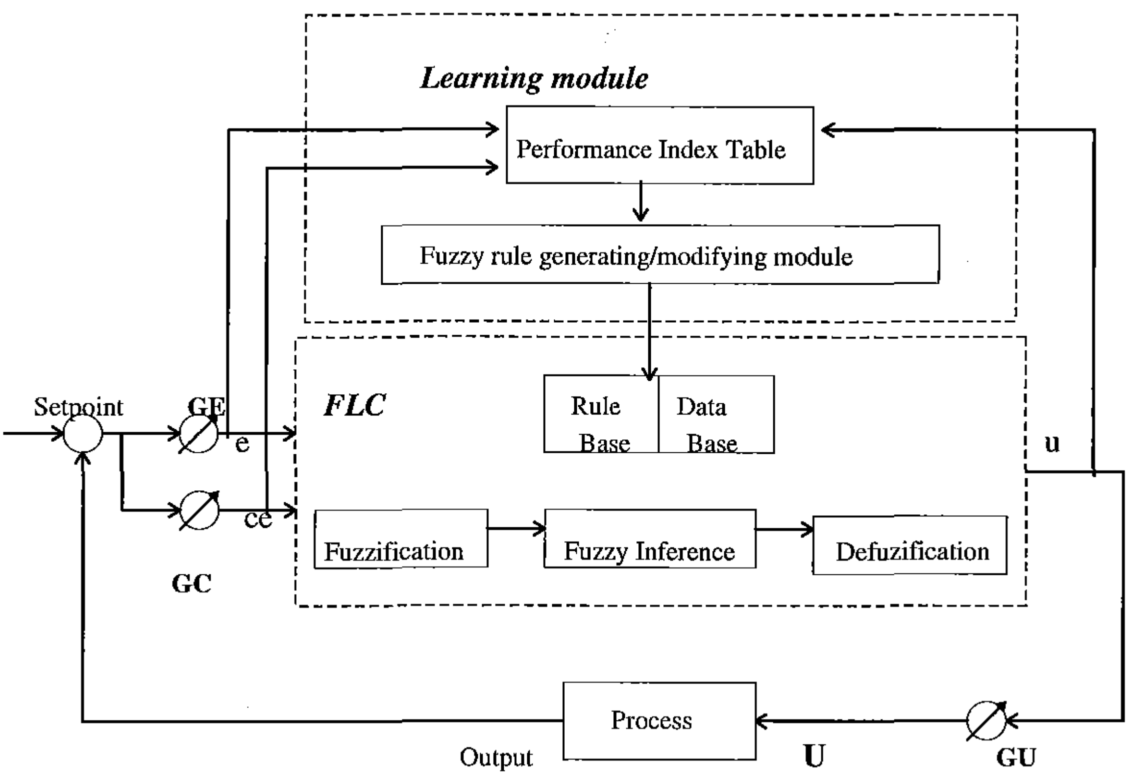


Figure 2.6 Basic structure of a SOFLC (Yan J., et al, 1994)

A SOFLC is a two level, hierarchical, rule-based controller in which the fuzzy control rule base of the FLC is created and modified by a learning module which comprises of a set of fixed performance rules. As illustrated in Figure 2.6, a SOFLC consists of the following:

1. An ordinary fuzzy logic controller at the basic level.
2. A learning module at the top level.

1. *The basic level*

The basic level of the SOFLC consists of a simple fuzzy logic controller. In this level, at each sampling instant, the input signal to the controller is taken and the error and change in error are calculated. The error (e) is given by the difference between the process output and the set point or the reference point and the change in error (ce) is the difference between the present error and the previous error. These two values are mapped from real values to normalised values and sent to the self organising level to generate new control rules and modify the old ones.

2. *The Self organising level*

The self organising level contains a performance index table (also called a learning rule base table) and a rule generation and modification algorithm, which is responsible for creating new rules or modifying existing ones. In a SOFLC, the control state of a process is monitored by the learning module. When an undesired output of the process is detected, the fuzzy control rules are created or modified based on the corrections given by the performance index table. The performance index table acts as a meta rule base that generates the strategy on how the control rules in the rule base can be amended. When the control state of a process deviates from its desired behaviour, the performance index table assigns a credit or reward value to the control actions that contributed to the present state.

2.3.8 *Adaptive Fuzzy Logic Controller*

A conventional fuzzy system converts its crisp inputs into fuzzy sets, invokes the rules relevant to the inputs based on some inference scheme, and defuzzifies the resultant output fuzzy sets into crisp outputs which acts as the input to the process to be controlled. It can adjust its behaviour during an execution cycle based on the results of the previous cycle but it does not reorganise itself or modify its rules to accommodate the changes in the environment (Cox E., 1993). Thus, the initial design parameters such as fuzzy sets, membership functions, universe of discourse and control rules have to be modified and adjusted to meet the design objectives.

While a self organising fuzzy logic controller is only responsible for creating and modifying fuzzy control rules, an adaptive fuzzy logic controller has self-tuning and learning capabilities. It can respond to environmental changes and adjusts itself to these changes. An adaptive system has the ability to learn and explain its reasoning and has the capacity to modify and extend it structure, thus making it very robust and extensible for a variety of problems.

An adaptive fuzzy logic controller consists of the following:

1. A supervising and tuning module at the top level.

2. A performance measurement module at the top level.

3. An ordinary FLC at the low level.

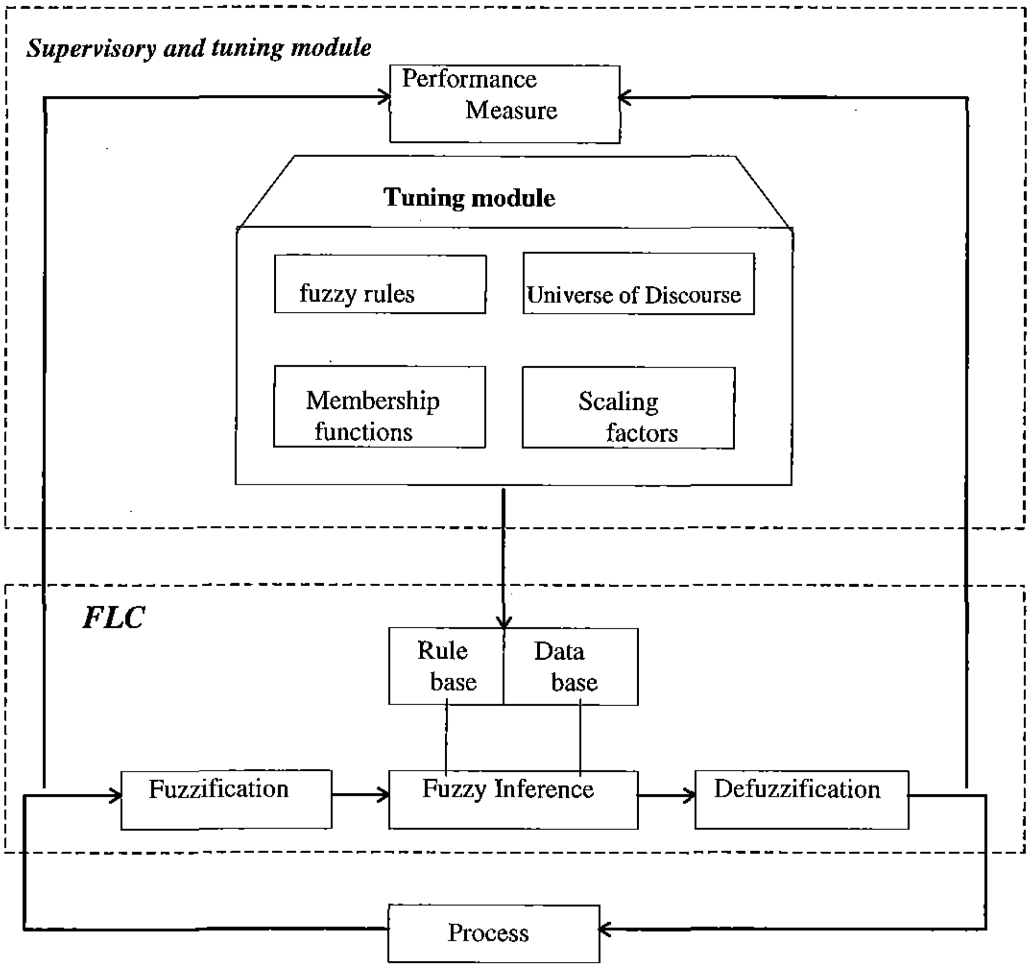


Figure 2.7 An adaptive fuzzy logic controller (Yan J., et al, 1994)

The kernel of an adaptive fuzzy logic controller is the supervising and tuning module which determines the required modifications or adjustments to the corresponding parameters, based on the system performance measures.

The performance measurement module determines the system performance, including the process error, the change of error, the least square error, the least mean square error, etc. The process error and the change of error are more frequently used as performance measures in most implementations, since the other algorithms are computationally expensive (Yan J., et al, 1994).

A fuzzy logic controller can be made adaptive by several ways. Various adaptive fuzzy logic controllers have been designed and implemented. They can be classified into four types: Adaptive fuzzy logic controllers with membership function tuning, universe of discourse tuning, scaling factor tuning, and fuzzy rule tuning (Yan J., et al, 1994). Procyk and Mamdani (1979) suggested a method for modifying the defuzzification technique to determine which rules are responsible for the present poor performance of the system and then delete those rules and add new rules. Making modifications to the scale factors by condensing or enlarging the spacing of the terms around the equilibrium point also improves the performance of the system (Ragot J., et. al, 1993).

Before implementing any system, obtaining the necessary training data for the problem is crucial to optimise the performance of the system. The self organising fuzzy logic controller developed by Procyk and Mamdani achieves a better performance of the controlled process by improving the fuzzy rules through a learning process. However, self organising fuzzy logic controllers require a valid model of the system. The desired output of the system is known and the controller is required to converge the system to the desired output. This type of learning, called supervised learning is not viable when the desired output of the system is not known.

Another learning technique that is becoming very popular is the use of Genetic Algorithms (GAs) to develop the fuzzy knowledge base for the fuzzy logic controller. The fuzzy rules are generated via genetic evolution. GAs do not require a model of the system nor do they need to know the desired output of the system.

A Genetic Algorithm starts with an initial population of solutions. Each of these solutions is evaluated by the fuzzy logic controller and ranked according to a fitness function. The fitness function gives an indication of the goodness of the solution and is system specific. Pairs of solutions are selected and combined based on the GA operators to create new solutions.

2.4 Genetic Algorithms

Genetic Algorithms (GAs) are search algorithms based on the mechanics of natural selection and natural genetics. They use operations found in natural genetics and guide their way through a search space. Genetic Algorithms have been shown to find the optimum or a near optimum solution to problems which have large and complex parameter space. They are probabilistic in nature and exhibit global search capabilities.

2.4.1 *An overview of GAs*

Genetic Algorithms were developed by Prof. John Holland (Holland J.H., 1975) at the University of Michigan to mimic some of the processes observed in natural evolution. The concept of Genetic Algorithms comes from the Darwinian theory of natural selection and survival of the fittest. Genetic Algorithms were developed to explain the adaptive processes of natural systems and design artificial systems that emulate the characteristics of the natural system.

Genetic Algorithms work on a population of genetically coded solutions. They use a bit string to encode a solution and each genetic code (a bit string of 0 and 1) represents a member or an individual. Each individual string is a member of a population.

A simple Genetic Algorithm is composed of three operators.

1. *Selection and Reproduction* - This is a process where individual strings are assigned copies according to their fitness function. Strings with a higher fitness value get more copies and have a higher probability of contributing to one or more offspring in the next generation.

2. *Crossover* - The selection stage of Genetic Algorithms reproduces a pair of the best existing individual parent strings for each offspring that is to be included in the next generation. But, it does not create any new individuals. In order to create new individuals from a mating pool, a simple one point crossover is applied where two strings are mated at random and are cut randomly between any two bits in the string. These pieces are swapped to form new individuals. As a result of crossover, valuable information from both parents are obtained and combined and is expected to form a highly fit individual.

Crossover can be illustrated by the following example:

Consider the following two parent strings:

$$A = \{011\ 100\}$$

$$B = \{101\ 010\}$$

An integer position R is selected between one and the string length less one. If $R = 3$, then

$$A = \{011 \mid 100\}$$

$$B = \{101 \mid 010\}$$

where the separator symbol (|) is the crossover point and the crossover yields the following two children.

$$A_{child} = \{011\ 010\}$$

$$B_{child} = \{101\ 100)$$

3. *Mutation* - Reproduction and crossover give genetic algorithms much of their power by searching towards a better solution in the local area. Mutation is a mechanism for maintaining genetic diversity in a population of strings and insuring against the premature loss of information. In this operation, each bit of the binary string is flipped with a very small probability. This prevents certain bits becoming fixed at a specific value. Otherwise, every string in the population might have the same value resulting in a premature convergence to a non-optimal solution.

Mutation can be illustrated as follows:

Consider the following parent string:

$$\{011\underline{1}00\}$$

After the mutation of the fourth bit, the child string obtained is as follows:

$$\{011\underline{0}00\}$$

The standard Genetic Algorithm searching steps can be summarised as follows (Goldberg D.E., 1989):

1. Define the representation of an individual.
2. Define a fitness function based on the desired objective. The fitness value evaluated from the fitness function gives the goodness of the individual.

3. Set the size of the population, the crossover rate, and the mutation rate.

4. Initialise the population.

5. Evaluate the fitness of each individual.

6. Select mates based on the fitness value of the population.

7. Create offsprings via reproduction, crossover and mutation.

8. Repeat steps 5 through 7 until a satisfactory solution is obtained.

2.4.2 *Differences between Genetic Algorithms and traditional search techniques*

Genetic Algorithms are a class of optimisation procedures whose mechanics are based on those of natural genetics. They combine survival of the fittest among string structures with a structured yet randomised information exchange to form a search algorithm with some of the innovative flair of human search.

Random search techniques have been increasing in popularity over the traditional search schemes like calculus-based and enumerative methods because of the robustness of the search algorithm. Calculus-based schemes are local in scope, that is, the optima they seek are best in the neighbourhood of the current point and they tend to climb false peaks in a mutimodal (multi-peaked) search space. Furthermore, these methods need derivatives, which are calculated analytically or numerically, to climb the current peak. But many functions contain discontinuities and vast multimodal noisy search spaces and these search spaces are unsuitable for search by calculus-based search schemes (Goldberg D.E., 1989).

Unlike calculus-based schemes which require derivatives to perform a search, Genetic Algorithms do not require any such auxiliary information: they are blind. To perform an effective search for better and better solutions, they only require the objective function values associated with individual strings. This characteristic makes Genetic Algorithms a more flexible method than other search schemes.

The search algorithms in enumerative schemes look at objective function values one at a time. Although this algorithm is simple and very human kind of a search, this search scheme lacks efficiency. Many practical spaces are too large to search one at a time.

In enumerative search techniques, we move from a single point in the search space to the next using some transition rule to determine the next point. But, this point to point method is dangerous because there is a high probability of locating false peaks in a multimodal search space. By contrast, Genetic Algorithms work from a rich database of points simultaneously (a population of strings), climbing many peaks in parallel, thus reducing the probability of finding a local maximum.

The differences between genetic algorithms and normal optimisation and search procedures can be summarised as follows:

1. Genetic Algorithms work with a coding of the parameter set, not the parameter themselves.

2. Genetic Algorithms search for a population of points, not a single point .

3. Genetic Algorithms use payoff (objective function) information, not derivatives or other auxiliary information.

4. Genetic Algorithms use probabilistic transition rules, not deterministic rules.

The use of probabilistic transition rules by Genetic Algorithms to guide their search does not suggest that this method is some simple random search. On the contrary, genetic algorithms use random choice as a tool to guide a search towards regions of the search space with likely improvement. Thus, direct use of coding, search from a population, blindness to auxiliary information, and randomised operators contribute to a genetic algorithm's robustness and the resulting advantage over other search schemes.

One of the most tedious and painstaking tasks in the design of a fuzzy logic controller is the construction of an appropriate rule base to control a process effectively. Genetic algorithms alleviate this by automatically generating the fuzzy rule base. The integration of the properties of Genetic Algorithms and Fuzzy Logic optimises the performance of fuzzy logic controllers.

In this thesis, we use Genetic Algorithms to learn the fuzzy control rules for a fuzzy logic controller which coordinates the traffic flow approaching two adjacent intersections and the traffic flow approaching a set of three intersections. The decision making ability of fuzzy logic controllers and the learning capability of genetic algorithms attempt to improve the overall design of the traffic environment.

2.5 Discussion

In this chapter, we introduced the basic concepts of fuzzy logic and genetic algorithms and provided a brief overview of urban road traffic planning and traffic light control. Fixed time control strategies and traffic responsive control strategies were discussed and some of the prevailing methods of automated urban traffic signal control systems were introduced.

Traffic control in large cities is a difficult and a non-trivial problem. The increasing number of vehicles and passengers often causes delays, congestion and accidents in roads. To overcome these difficulties, automated urban traffic control systems (AUTCS) are in use. These systems can be classified into two categories - Fixed time and Traffic responsive control systems. Traffic responsive systems are a better alternative to fixed time control because they change their signal parameters based on the prevailing traffic while fixed time control plans are done off-line and then switched into operation depending on the time of day.

Even though traffic responsive control strategy regulates the traffic flow at the intersections based on the prevailing traffic, it is not able to deal with unforseen changes in the traffic environment such as congestion, accidents, etc. This control strategy fails to adapt itself to the dynamic changes in the traffic situation. As a result, the whole network is affected since each intersection is linked to its neighbouring intersection.

The application of artificial intelligence techniques to transport problems has been an area of intense research for the past few years. The use of fuzzy logic for controlling traffic at intersections has been studied by various researchers and different approaches have been proposed.

Fuzzy set theory and fuzzy logic, the logic for manipulating the fuzzy sets, is a very powerful mathematical device for treating uncertainty, subjectivity, ambiguity, and indetermination. Fuzzy logic perceives the environment in a manner which is not very different from the human way of thinking. It is not practically possible to accurately define human features like thinking and reasoning, and fuzzy logic attempts to emulate these features, an ability which traditional methods lack. These characteristics make fuzzy logic a powerful tool for making decisions in an uncertain environment.

Fuzzy logic control is an effective tool for decision making in an uncertain and imprecise environment. It can be employed as an alternative to the conventional methods of traffic control. The randomness and unevenness inherent in the flow of traffic can be handled more efficiently by fuzzy logic controllers than conventional controllers. They regulate the traffic by making on-line adjustments to the signal timing parameters of the traffic light.

The design of a fuzzy logic controller is based on the construction of an appropriate rulebase for controlling the system. These rules are generally derived from an operator's knowledge and experience. In certain cases, the process to be controlled may depend on a large number of input parameters, thereby making it difficult to determine an appropriate control action for a given set of input conditions. This problem of rule-elicitation can be overcome by employing Genetic Algorithms to learn the fuzzy rules for controlling a system.

Genetic Algorithms (GAs) are search techniques that define a global search by simultaneously considering many points in the search space. Most search and optimisation techniques require derivative information or complete knowledge of the problem structure under consideration. But, genetic algorithms require only information about the quality of the solution produced by each parameter set. These characteristics make genetic algorithms an attractive tool for the generation of fuzzy knowledge bases for a fuzzy logic controller.

Chapter 3

Fuzzy control of an Isolated Intersection

Fuzzy logic control can be used as an alternative approach to conventional control for the control of traffic environments. A typical traffic environment includes the lanes to and from the intersection, the intersection, the vehicle traffic and the traffic signals at the intersection. A traffic signal should be able to handle the fluctuations in the traffic flow efficiently and utilise the green phase periods to achieve maximum throughput.

In this chapter, a fuzzy logic traffic controller to control the traffic signal at an isolated intersection, is presented. The traffic flow approaching the intersection is regulated by a set of fuzzy decision rules to adjust the green phase split of the signal. Depending upon the traffic volume at the north-south and east-west approaches, the green phase is adjusted accordingly to minimise congestion at the intersection.

3.1 The Model

It is assumed that each signalised intersection uses sensors that count the number of vehicles instead of proximity sensors which only indicates the presence of a vehicle. The vehicle densities are taken from two sensors placed on the road. One is at the intersection and the other at 100 metres from the intersection. The rear sensor increments a counter every time a vehicle passes over it, while the forward sensor decrements the same counter. This gives a count of the number of vehicles waiting 100 metres before the light and a count of the number of vehicles that pass through the intersection when the light is green. Figure 3.1 shows an isolated signalised intersection.

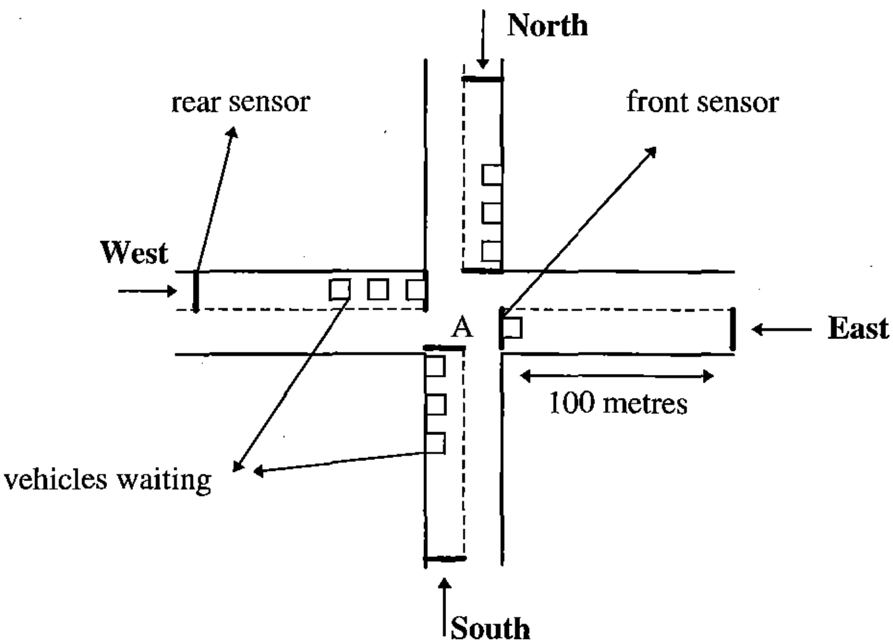


Figure 3.1 An isolated intersection

The maximum number of vehicles that can be detected by the sensors is twenty since we are assuming that each vehicle is about four metres in length and there is a spacing of one metre between each vehicle when they are stationary, and waiting for the light to turn to green. If the number of vehicles waiting at the intersection exceeds twenty, the fuzzy logic controller makes a decision based only on the first twenty vehicles in the queue. As a result, the queue length is reduced gradually by the fuzzy logic controller.

3.1.1 *Vehicle Movement*

The traffic flow is modelled to mimic reality, specifically the motion of vehicles relative to one another. Physical equations are used to describe the motion of a vehicle based on the velocity and position of the vehicle in front of it. The equations used in the simulation are given as follows:

$$a_f(t + T) = k\ (\ v_l(t) - v_f(t)\ )\ /\ (\ x_l(t) - x_f(t)\ ) \qquad (3.1)$$

$$v(t) = v(t\text{-}1) + a(t)dt \qquad (3.2)$$

$$x(t) = x(t\text{-}1) + v(t)dt + \frac{1}{2}a(t)dt^2 \qquad (3.3)$$

$$t = d\ /\ v(t) \qquad (3.4)$$

Equation (3.1) describes the acceleration of a vehicle based on the velocities and positions of itself and the vehicle immediately in front of it. This equation is obtained from the traffic flow theory proposed by Haight (1963) and Barwell (1973). Equations (3.2) and (3.3) are classical physics equations for determining the velocity and position of an object based on the object's acceleration. Equation (3.4) is used to determine the time taken by a vehicle to travel a distance d with a velocity v(t).

$a_f\,(t + T)$ is the acceleration of the following vehicle after a time lag of T seconds. $v_l(t)$ and $x_l(t)$ are the velocity and position, respectively, of the leading vehicle at time t, and $v_f(t)$ and $x_f(t)$ are the velocity and position of the following vehicle. T is the time lag between the leading vehicle and the following vehicle. Time lag is the gap distribution to the next vehicle in time, also called headway, and is assumed to be equal to one and a half seconds. k is the characteristic speed which is set to 8.0 meter/second (Barwell F.T., 1973). The acceleration value for the first vehicle in the queue of vehicles is assigned a value equal to 4.9 metres/second^2.

In equations (3.2) and (3.3), v(t) is the final velocity of a vehicle, v(t-1) is the initial velocity of a vehicle, a(t) is the acceleration of a vehicle, x(t-1) is the initial position of a vehicle and x(t) the final position. In equation (3.4), t is the time taken by a vehicle to travel a distance d with a velocity v(t).

When the traffic signal changes to green for a particular approach, the velocity and position of the first car in the queue is determined from equations (3.2) and (3.3) after a time lag of one and half seconds. The acceleration of the next vehicle in the queue is determined from the velocity and position of the leading vehicle using the equation (3.1). After a time lag of one and a half seconds, the next vehicle moves forward based on the velocity of the vehicle immediately in front of it. In this way, each vehicle crosses the intersection with a velocity which is dependent on the velocity of the leading vehicle. The time taken by each vehicle to cross the intersection is determined from equation (3.4).

3.1.2 *Delay time*

The arrival time of vehicles at the intersection is considered as being random. At each successive time unit (one second), a random number is generated and compared with the mean vehicle arrival rate (number of vehicles/hour) and the arrival of a vehicle is decided.

If q_n denotes the arrival of a vehicle at the *n*th time interval,

then $q_n = 1$, if a vehicle arrived during the *n*th time interval

else $q_n = 0$.

If Q_G is the number of vehicles not cleared during the previous green phase, then queue $Q_{n,R}$ at the *n*th time unit after the beginning of the red phase is given by:

$$Q_{n,R} = Q_G + \sum_{n1=1}^{n} q_{n1} \qquad (3.4)$$

The total waiting time of the vehicles, $D_{n,R}$, in the queue at the *n*th time unit after the beginning of the red phase would be:

$$D_{n,R} = \sum_{n2=1}^{n} \left(Q_G + \sum_{n1=1}^{n2} q_{n1} \right) \qquad (3.5)$$

Equations (3.4) and (3.5) are obtained from the model proposed by Pappis and Mamdani (1977).

If P_G is the number of vehicles that has passed during the green phase, the number of vehicles not cleared at the *n*th time unit after the beginning of the green phase is given by:

$$Q_{n,G} = Q_R + \sum_{n1=1}^{n} q_{n1} - P_G \qquad (3.6)$$

These vehicles are subjected to a delay of:

$$D_{n,G} = \sum_{n1=1}^{n}\left(Q_G - P_G + \sum_{n2=1}^{n1} q_{n2}\right) \qquad (3.7)$$

Thus, during a cycle, the total delay experienced by the vehicles travelling in all four directions is given by:

$$D = D_{n,R} + D_{n,G} \qquad (3.8)$$

The average delay per vehicle would be:

$$d = \frac{D}{\sum_{n=1}^{R+G} q_n} \qquad (3.9)$$

3.1.3 *Assumptions*

Some simplifying assumptions are used in the simulation model:

- There is only single lane traffic.
- A vehicle cannot turn at the intersection.
- There is no pedestrian traffic.

3.2 Fuzzy control rules for an Isolated intersection

A fuzzy logic traffic controller comprising twenty five fuzzy rules is used to adjust the green phase split of a traffic signal. The number of vehicles waiting (queue length) at the end of a red phase and the number of vehicles that passed through in the previous green phase are the deciding factors in adjusting the green phase split of the signal.

When traffic volume is high for a particular approach, the current green phase for that approach is increased to allow more vehicles to flow through, thus reducing the number of vehicles waiting at that approach. When traffic volume is low, the green periods are reduced resulting in shorter cycle time and thereby reducing the delay in waiting for phase changes.

The input variables of the fuzzy logic traffic controller are :

(i) *The ratio of queue length to number of vehicles that passed through, in the east-west approaches.*

(ii) *The ratio of queue length to number of vehicles that passed through, in the north-south approaches.*

The input control variables are determined in the following fashion: the queue length (number of vehicles waiting) of an approach is compared with that of the opposite approach, and the greater of the two queue lengths is chosen. This queue length is then divided by the number of vehicles that passed through the intersection during the previous green phase. For example, if the queue length at the north approach is greater than that of the south approach, the number of vehicles waiting at the north approach is divided by the number of vehicles, from the north approach, that passed through the intersection during the previous green phase. The ratio of queue length to number of vehicles that passed through, in the east-west approaches is also determined in a similar way.

The output variables of the fuzzy logic traffic controller are :

(i) *The amount of adjustment to the current green phase of the north-south approach.*

(ii) *The amount of adjustment to the current green phase of the east-west approach.*

In the approach presented by Chiu and Chand (1993) for the fuzzy control of a traffic signal, the input control variables are the degree of saturation in the north-south and east-west approaches. The degree of saturation is the ratio of the number of vehicles that has passed through the intersection to the maximum number of vehicles that can pass through during that period, determined from the saturation flow, in the north-south and east-west approaches. The degree of saturation shows the effectiveness of the green phase. The effectiveness of the current green period gives a measure of how long the next green phase should be. This control scheme is based on prediction as the length of the current green phase is determined from the vehicle flow during the previous green phase.

In this thesis, the queue length and the number of vehicles that has passed through the intersection during the previous green phase are considered as the deciding factors for the extension/reduction to the green phase time. The queue length gives an accurate measure of the waiting traffic, and the ratio of queue length to number of vehicle that passed through gives a measure of both the vehicles waiting and the vehicle flow. If the ratio is high, then the green phase should be extended to allow more vehicles to flow through thereby reducing the queue length.

The degree of saturation introduced by Chiu and Chand (1993) gives an indication of the effectiveness of the previous green phase which, in turn determines the length of the current green phase where as, the ratio of queue length to number of vehicles that passed through, determines the length of the green periods based on the actual traffic situation.

The linguistic input and output variables for adjusting the green phase for the North-South and East-West approach of a traffic signal are listed below. Each fuzzy set numerically represents a linguistic term. The membership functions for the input and output variables are shown in Figure 3.2 and 3.3.

The fuzzy linguistic terms for the input variables (ratio of queue length to vehicles passed) are:

VL : *Very Low*

LO : *Low*

MD : *Medium*

HI : *High*

VH : *Very High*

The fuzzy linguistic terms for the output variables (amount of adjustment to the green phase) are:

NB : *Negative Big*

NM : *Negative Medium*

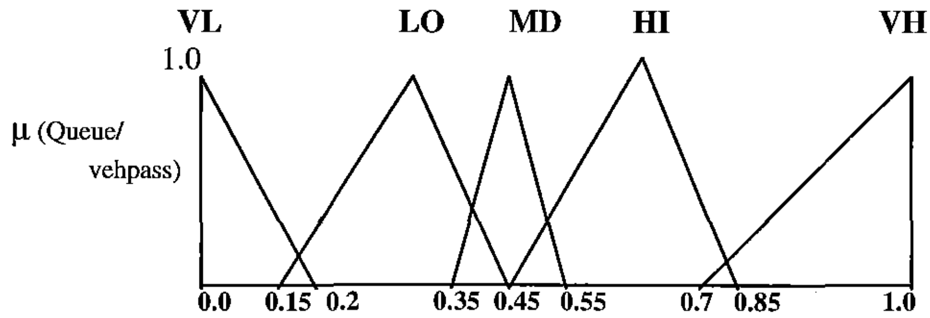
NS : *Negative Small*

ZE : *Zero*

PS : *Positive Small*

PM : *Positive Medium*

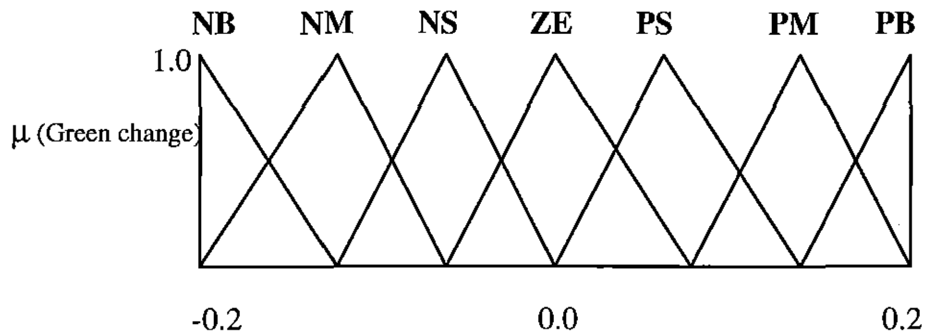
PB : *Positive Big*



Ratio of queue length to vehicles passed (Queue/vehpass) for

north-south and east-west approaches

Figure 3.2 Membership functions for the input fuzzy sets



Green phase change for N-S and E-W approaches

Figure 3.3 Membership functions for the output fuzzy sets

Construction of the Rulebase

Once the precise numeric conditions of the system are categorised into fuzzy sets, a process for determining the appropriate control action must be constructed. This involves writing a rule set that provides a fuzzy action for every possible condition that could exist in the problem environment. The following rules are developed for adjusting the green phase of the north-south and east-west approaches. They are arranged in a matrix relationship of input and output fuzzy variables which is called a Fuzzy Knowledge Base. Table 3.1 shows a Fuzzy Knowledge Base. Each entry in the table is made up of two components. The first is the green phase adjustment to the traffic signal at the North-South approach and the second is the green phase adjustment to the traffic signal at the East-West approach. The rules in the rule set are of the form '*If ratio in North-South is **MD** and ratio in East-West is **HI** then green_change_NS is **ZE** and green_change_EW is **PM**.*

The rules are evaluated at every phase change; the maximum green phase adjustment allowed in one step is 20% of the current green period so that any change in the traffic volume results in a gradual change to the green phase of the signal (Chiu S., et al, 1993). The green phase for any approach can be extended to a maximum of 28 seconds.

		Ratio in east -west aproaches				
		VL	**LO**	**MD**	**HI**	**VH**
Ratio in north-south approaches	**VL**	NB NB	NB NS	NB PS	NB PM	NB PB
	LO	NS NB	NS NS	NS PS	NM PM	NS PB
	MD	PM NB	PS NS	ZE ZE	ZE PM	NS PB
	HI	PM NB	PM NM	PM ZE	PM PM	PM PB
	VH	PB NB	PB NM	PB NS	PB PM	PB PB

Table 3.1 - Fuzzy Knowledge Base

Inference mechanism and Defuzzification

The fuzzy outputs of the fuzzy logic traffic controller are determined by the *MAX-MIN fuzzy reasoning* mechanism (see section 2.3.6.7). Mamdani's 'min' implication function is adopted to map the input fuzzy sets to the output fuzzy sets. The fuzzy control action of each rule is decided by its firing strength and the fuzzy sets. The fuzzy control action inferred from the complete set of fuzzy rules is equivalent to the aggregated result derived from individual rules.

The *Centroid defuzzification* scheme is employed to determine the crisp outputs from the fuzzy output sets. The green phase change to the North-South and East-West approaches is computed by invoking the following formula:

$$W = \frac{\sum_{i=1}^{n} W_i Z_i}{\sum_{i=1}^{n} W_i} \tag{3.10}$$

where n is the number of rules fired, w_i is the firing strength and z_i is the amount of control action recommended by rule i. The Centroid method of defuzzification yields a single control action which is applied to the physical system.

3.3 Software used for the simulations

The simulations are performed by developing software packages for the Fuzzy Logic Controller (FLC) and the Traffic model. The software packages are developed in 'C' using the Borland C++ V4.0 compiler. The process of fuzzification, inference, and defuzzification is performed by the Fuzzy Logic Controller simulator. The software simulating the traffic flow provides the inputs to the Fuzzy Logic Controller and receives the outputs from the Fuzzy Logic Controller to regulate the traffic flow.

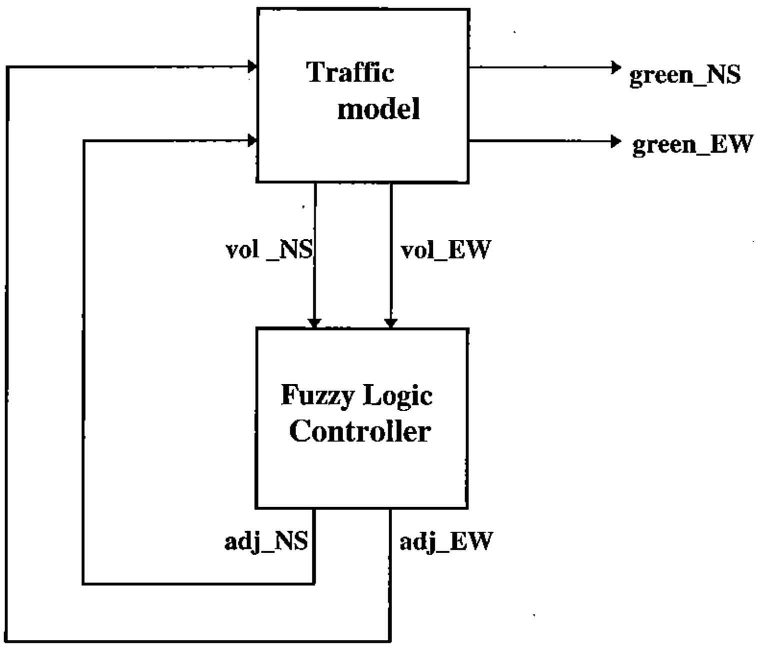


Figure 3.4 Model used for the simulation

The traffic density approaching the intersection is determined by the software simulating the flow of traffic (The traffic model). This data is sent to the Fuzzy Logic Controller which determines the amount of adjustment to be made to the green phase. An increase in the flow of traffic results in an increase in the time duration of the corresponding green phase.

Vehicle movements are simulated using the standard laws of motion and equations to determine the velocity and position of a vehicle based on the velocity and position of the leading vehicle.

3.4 Simulation Results

Simulation was performed to establish the effectiveness of this fuzzy logic control scheme. An isolated intersection with four approaches, as shown in Figure 3.4 is considered. The green phase for the North-South and the East-West approaches is set initially to 20 seconds. The green phase of the signal for any approach cannot exceed 28 seconds. The fuzzy control rules are evaluated at every phase change and the maximum green phase adjustment allowed at any time step is 20% of the current green period.

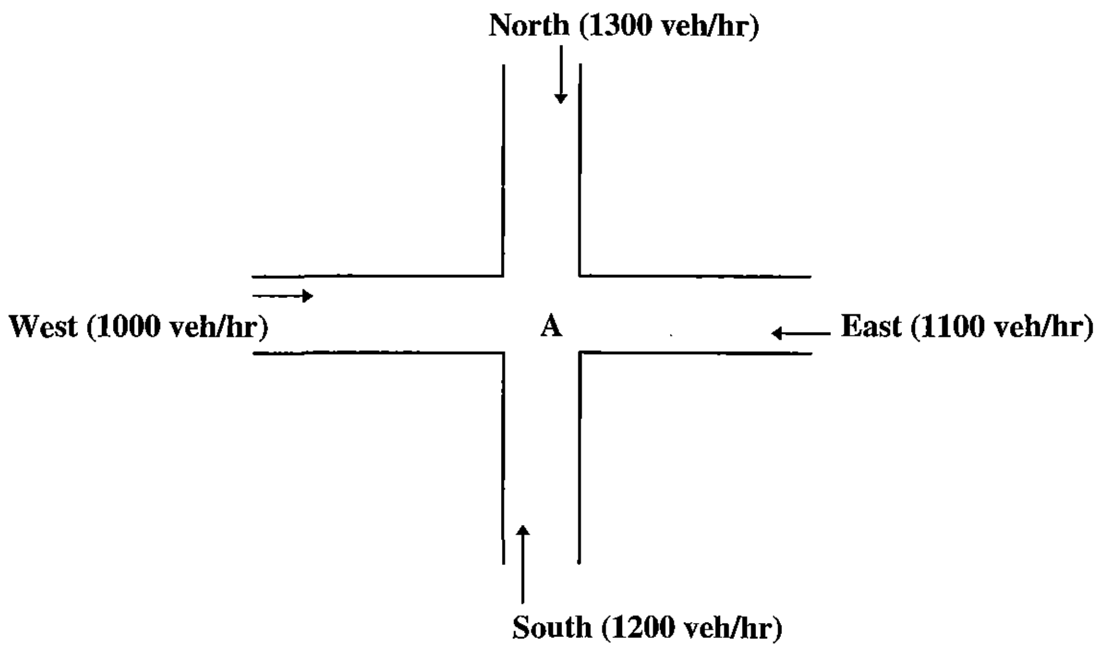


Figure 3.5 An isolated intersection '*A*'

Road sensors are assumed to be installed at the intersection and at 100m from the intersection to determine the number of vehicles waiting and to keep a count of the number of vehicles that passes through the intersection. A mean vehicle arrival rate is assigned to each end of a street as shown in Figure 3.5. At each simulation time step, a random number is generated for each approach and compared with the mean vehicle arrival rate (vehicles/hour) to determine whether a vehicle should be added to the end of the lane. If the mean arrival rate is high at the north-south approaches, the traffic flowing from those directions is high thus resulting in extended durations of green phase for the north-south approach and reduced durations for the east-west approach.

The input variables, ratio of the vehicles waiting to the vehicles passed during the previous cycle for the north-south and east-west approaches are sent to the fuzzy logic traffic controller. The fuzzy decision rules given in Table 3.1 are then applied to determine whether the green phase is to be extended/reduced for an approach.

When the traffic volume is high for both approaches, the green phase for both the approaches is increased thus minimising the number of stops at the intersection. As the number of vehicles waiting at the approaches reduces, the green phase is also reduced thus minimising the delay as a result of waiting.

Figure 3.6 shows the number of vehicles waiting (queue length) at the north approach of the isolated intersection (solid line) and the green phase duration of the north-south approach of the traffic signal (dashed line). The queue length is the number of vehicles at the end of a complete cycle. From Figure 3.6, it can be seen that the fuzzy logic traffic controller follows a trend similar to that of the queue length. An increase in the queue length results in a corresponding increase in the length of the green phase thus allowing more vehicles to pass through the intersection. The fuzzy logic traffic controller tries to keep the number of vehicles waiting to a minimum and thereby reducing the average time spent by a vehicle on waiting. The change in the green phase duration is gradual rather than abrupt in relation to the queue length.

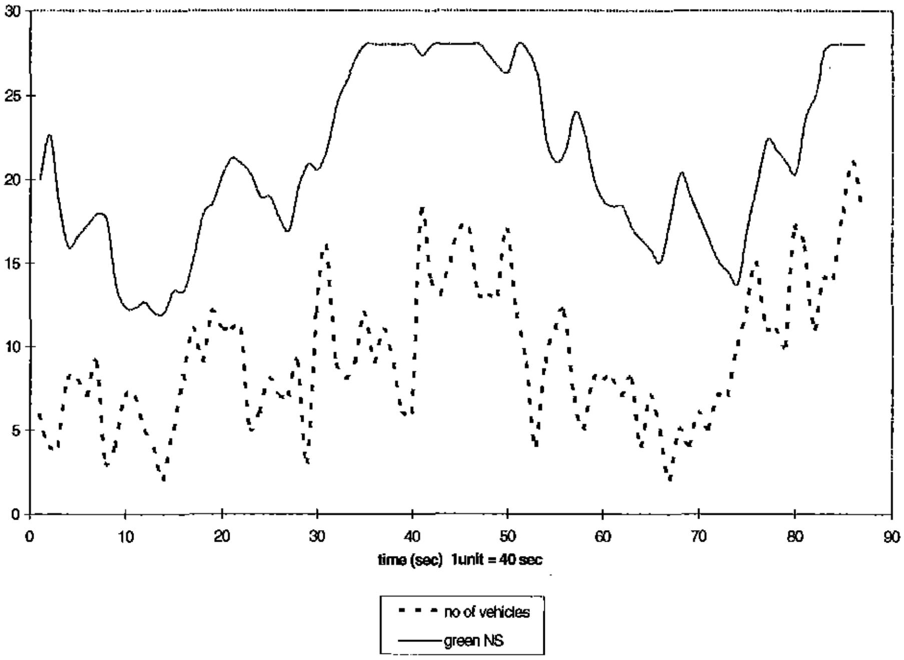


Figure 3.6 Queue length at North approach and green phase North-South

Figure 3.7 shows the number of vehicles waiting at the south approach of the isolated intersection and the green time in the north-south approach. Similar to the previous figure, the controller tries to maintain the queue length to a minimum. When there is a heavy burst of traffic, as from time instant 30 - 50, in the Figure 3.5, the fuzzy logic traffic controller extends the green phase period to increase the vehicle flow and thus reducing the number of vehicles waiting.

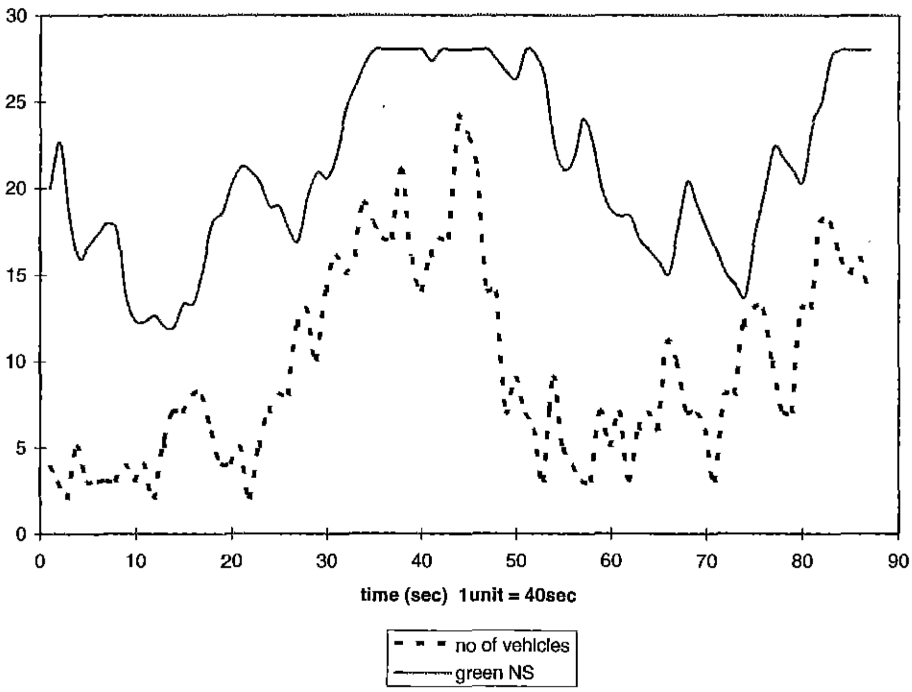


Figure 3.7 Queue length at the South approach and green North-South

Figure 3.8 shows the number of vehicles waiting at the east approach and the green phase duration of the east-west approach.. In this figure, the green phase duration during the time period 75 - 80 is high, eventhough there are no vehicles waiting at the east approach. This is because, the approach with the greater of the two queue lengths acts as the input to the fuzzy logic traffic controller. If there is a heavy traffic flow at the west approach, the green phase for the east-west approach is increased to accomodate more vehicles to pass through from the west approach.

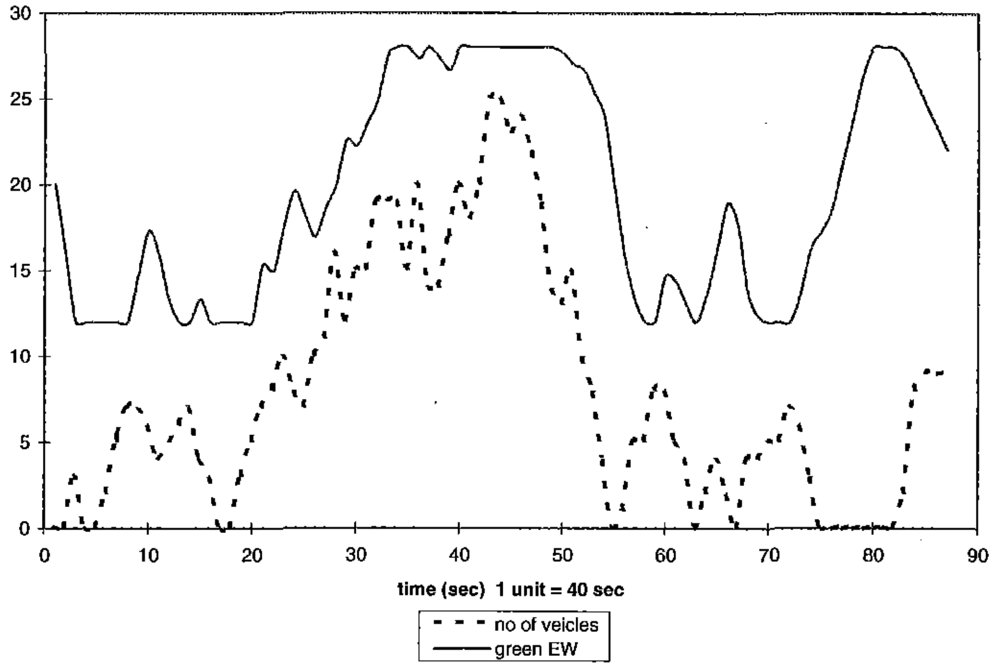


Figure 3.8 Queue length at the East approach and green phase East-West

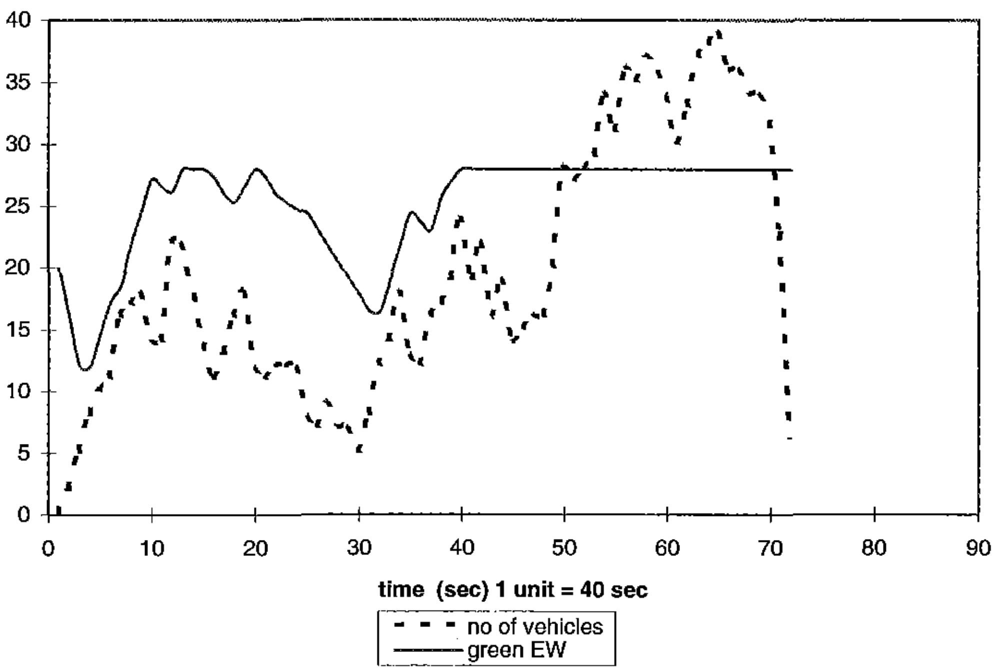


Figure 3.9 Queue length at the West approach and green phase East-West

In Figure 3.9, an increase in the length of the queue at the west approach of the intersection results in a corresponding increase in the green phase of the east-west approach of the signal. Once the green phase attains the saturation value, which is 28 seconds, further increase in the queue length does not affect the green phase of the east-west approach of the signal. The duration of the green phase is maintained at 28 seconds till there is a reduction in the traffic flow approaching the west approach. This figure shows that due to the constraint on the upper limit of the green phase duration, the fuzzy logic traffic controller is unable to respond to very heavy traffic flows. To avoid congestion at intersections with very heavy traffic flows at all approaches, the maximum value of the green phase of the north-south and east-west approaches should be high.

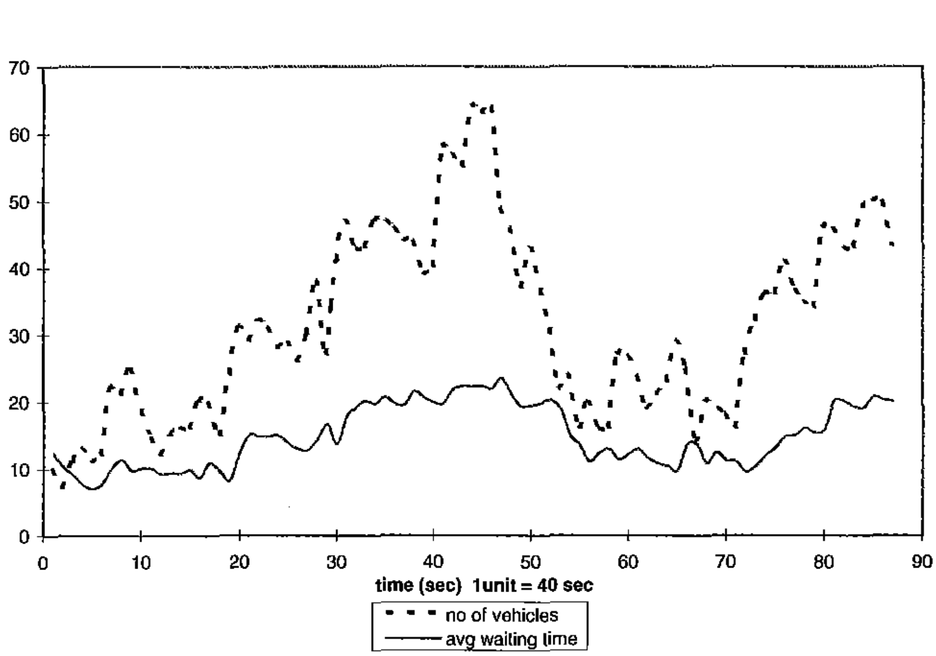


Figure 3.10 Queue length and average waiting time

Figure 3.10 shows the total number of vehicles waiting at all four approaches at the end of a cycle and the average waiting time per vehicle. When there are a greater number of vehicles waiting at the approaches, the time spent in waiting by each vehicle for the light to change to green increases.

Figure 3.11 shows the effect of of the vehicle arrival rate on the average waiting time spent by a vehicle in the queue. The delay time per vehicle increases as the vehicle arrival rate is increased for all the approaches. This is due to increased cycle time periods and more vehicles spending time at the intersection waiting for the phase to change from red to green.

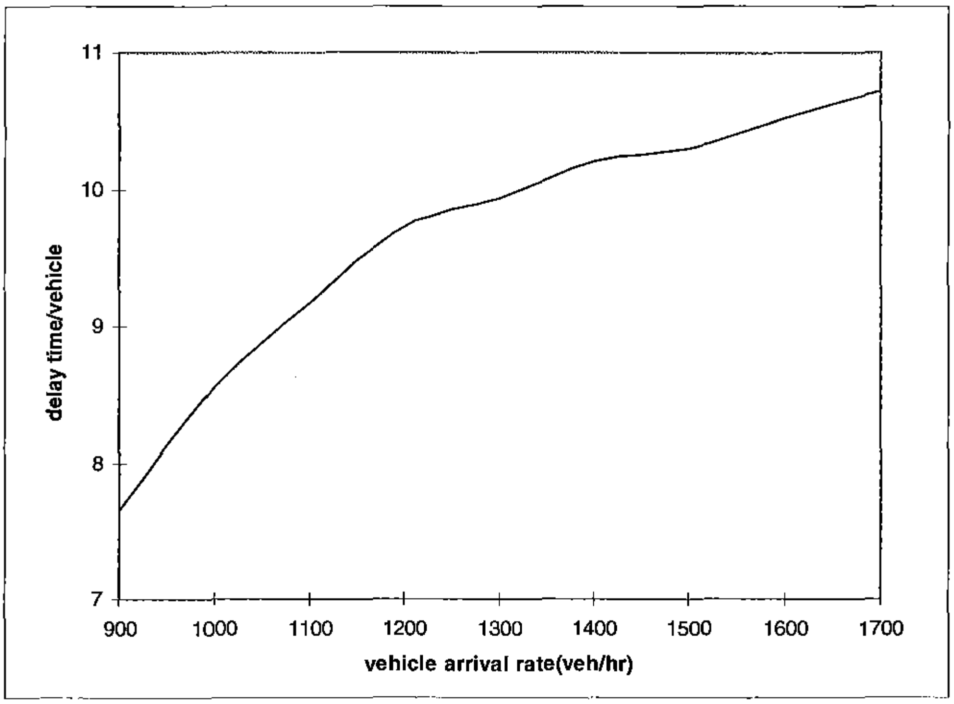


Figure 3.11 Average waiting time/vehicle

It can be seen from Figure 3.12 that the flow of vehicles passing the intersection increases with an increase in the vehicle arrival rate. When the arrival rate of vehicles is not high, the traffic signal at the north-south and east-west approaches have reduced green phase durations since the number of vehicles waiting at these approaches is quite minimal and there is no need for extended durations of the green phase. An increase in the vehicle arrival rate for all the approaches, results in an increase in the green phase periods for all the approaches and as a consequence more vehicles flow through the intersection from all directions. After a stage, the vehicle flow reaches a saturation level as the vehicle arrival rate does not have any effect once the green phase durations of the signal reaches the maximum value, that is, 28 seconds. The vehicle flow tends to remain a constant irrespective of the vehicle arrival rates.

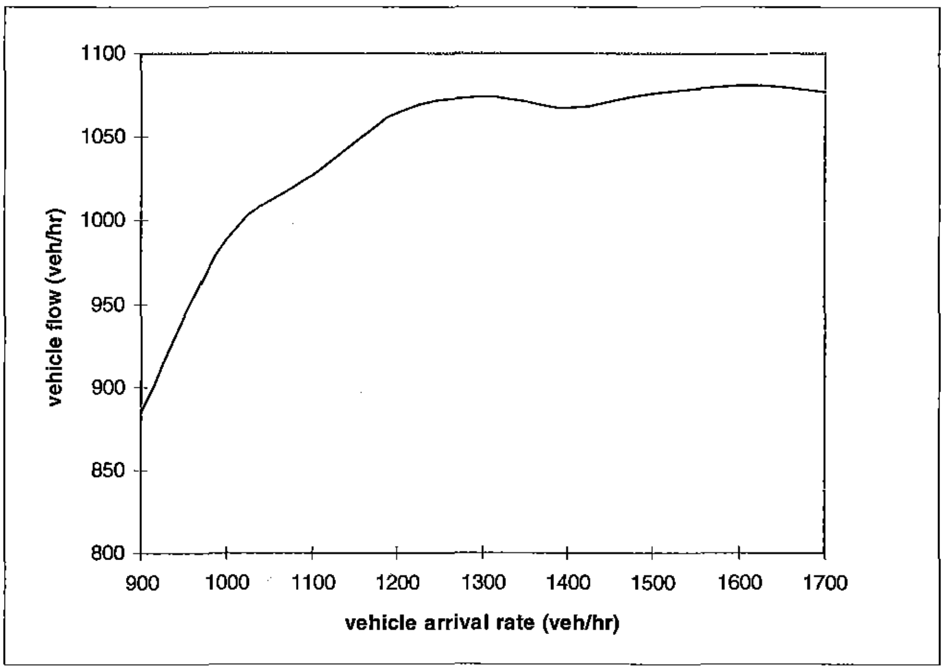


Figure 3.12 vehicle flow (veh/hr)

3.5 Discussion

The fuzzy logic traffic controller controlling an isolated traffic signal is responsive to abrupt changes in the traffic situations. An increase in the number of vehicles waiting at the north-south and east-west approaches results in an increase in the green phase durations for the respective approaches. Any change in the traffic conditions results in a corresponding change in the time duration to the green phase of the traffic signal. An increase in the queue length results in an extension to the green phase until it reaches its maximum value, after which it cannot be extended any more.

The on-line adaptation of the fuzzy logic traffic controller is instrumental in minimising the number of vehicles waiting at the intersection as well as the average time spent on waiting by each vehicle. The performance of the controller is still of high quality even when the traffic flow is highly uneven. It can change the traffic signal lights as necessary to achieve maximum throughput, rather than be limited to a preset cycle time as it is the case with a fixed time traffic controller.

The use of sensors to determine the traffic densities in the lanes rather than just detecting the presence of a vehicle provides the fuzzy logic traffic controller with a better assessment of the traffic patterns. Thus, the controller developed is expected to improve the standard of traffic regulation at signalised intersections in a dynamic environment by adapting itself to the changes in the traffic situation.

This chapter discussed the regulation of traffic flow approaching an isolated traffic signal. The signal is controlled by a fuzzy logic traffic controller comprising twenty five fuzzy rules. In an urban road traffic network, the performance of a traffic junction usually affects that of its neighbouring traffic junctions. Each intersection is coordinated with its adjacent intersections based on the signal timing parameters. In the next chapter, two adjacent traffic signals are coordinated by adjusting their respective offsets.

Chapter 4

Fuzzy control of two adjacent intersections

In chapter 3, a fuzzy logic scheme to control a traffic signal at an isolated intersection is presented. The adjustments to the green phase splits of the north-south and east-west approaches are made on-line depending on the number of vehicles waiting at the respective approaches and the number of vehicles that passed through the intersection during the previous green phase.

Traffic signal coordination is one of the most widely used and cost effective means of improving the efficiency of traffic flow. An Area Traffic Control (ATC) system does not consist of just a single traffic signal but a number of traffic signals linked together. If any one of these traffic signals are oblivious to the traffic volume at the other intersections, then the traffic flowing through the system is not optimised. This is because of the lack of coordination in phase splits and offset between the intersections.

The signal timing changes at each intersection should be dependent upon the prevailing conditions at the other intersections in order to optimise the traffic flowing through the system. In an ATC system, the signals at two or more intersections are coordinated on a common cycle time and the offsets are adjusted in such a way that the vehicles passing one intersection arrive at the downstream intersection when the light is green. As a result, the vehicles arriving at the downstream intersection pass through unstopped.

In this chapter, two adjacent signalised intersections are coordinated in the north-south direction. The traffic signal at each intersection is controlled by a fuzzy logic traffic controller which adjusts the green phase splits based on the local traffic. In addition to the green phase adjustment, the offset at each intersection is also adjusted in order to coordinate it with the adjacent intersection. Two control schemes using fuzzy logic are investigated for adjusting the offset at each signal so that number of stops at the second intersection is minimised.

4.1 The Model

The model to regulate the traffic flow approaching two adjacent intersections is similar to the one that is proposed in chapter 3 for the isolated intersection. In the model proposed for the isolated intersection, the motion of vehicles after they cross the intersection is not taken into account since only one intersection was being considered. In the case of two intersections, each vehicle passing the first intersection moves at a velocity which is dependent on the velocity of the vehicle immediately in front, till it reaches the second intersection. The distance between the two intersections is assumed to be 100 metres.

Figure 4.1 shows the two adjacent intersections *A* and *B*. Road sensors are placed at each intersection and 100 metres away from each intersection. A mean vehicle arrival rate is assigned to each end of the street. The mean vehicle arrival rate for the north, south, east, and west approaches of the two intersections are shown in Figure 4.1. A vehicle is added to the queue based on this mean vehicle arrival rate. At each simulation step, a random number is generated and compared with the vehicle arrival rate to determine whether a vehicle should be added to the queue. The arrival rate at the north and south directions are higher than those of the east and west directions thereby making north-south the dominant directions of traffic flow.

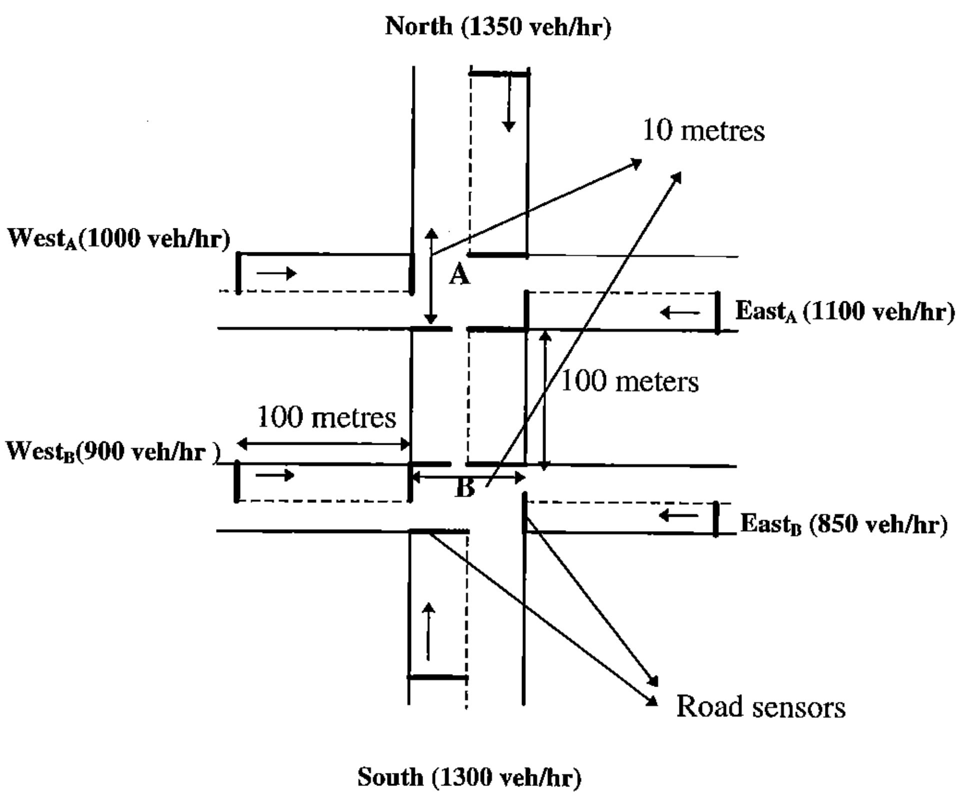


Figure 4.1 Two adjacent intersections used in the simulation

The two traffic signals operate in such a way that the green phase for both the signals begin simultaneously. When the traffic signal changes to green at the first intersection, each vehicle in the queue crossing the traffic junction accelerates for two seconds and then moves forward at a constant velocity. The initial velocity for all the vehicles is zero. The acceleration is determined from equation (3.1). The final velocity and final position of each vehicle after two seconds is determined from equations (3.2) and (3.3). There is a time lag of approximately one and half seconds between any two vehicles since each vehicle crosses the junction giving a headway of one and half seconds to the vehicle in front. The time taken by each vehicle to reach the second intersection is calculated from equation (3.4) since the distance between the two intersections and velocity for each vehicle are known.

If there are no vehicles waiting at the second intersection and if the light is still green at the second traffic signal, then each vehicle crosses the second traffic junction unstopped. Else if the light is red or if there are vehicles waiting at the second intersection, then the vehicles leaving the first intersection form a queue at the second intersection. The maximum number of vehicles that can be accommodated in the stretch between the two intersections is twenty. Once the number reaches twenty, no more vehicles are passed across the first intersection until the queue length becomes less than twenty. For example, if the number of vehicles waiting at the north approach of intersection B exceeds twenty, the vehicles at the north approach of intersection A do not pass through intersection A, even if the light is green, until the number of vehicles at the north approach of intersection B reduces below twenty.

4.2 Two intersections with no offset adjustment

Each of the two traffic signals placed at adjacent intersections in the north-south direction is controlled by a fuzzy logic traffic controller which uses a set of twenty five fuzzy decision rules. These rules are the same as those that were introduced in chapter 3 for controlling the traffic signal at an isolated intersection, (see Table 3.1). The fuzzy sets and the membership functions of the input and output variables are the same as those used in chapter 3. The green phase splits at the north-south and the east-west approaches are adjusted based on the ratio of the traffic density to the number of vehicles that passed through during the previous green phase. Both traffic signals operate in an isolated mode, based only on the local traffic information.

The fuzzy logic traffic controllers controlling the individual traffic signals operate independently of each other. There is no coordination between them, that is, neither one of the two controllers has any knowledge about the traffic situation at the other intersection. This might lead to chaos if there is a heavy flow of traffic from either one or from both the directions resulting in a traffic congestion at the approaches leading up to the intersections.

4.2.1 *Simulation Results*

Figures 4.2 and 4.3 show the number of vehicles waiting at intersection *A* and intersection *B* respectively. With no coordination between the two intersections, there is a tendency for the queue length to increase at the intersections A and B. The fuzzy logic traffic controllers that adjust the green phase of the north-south and east-west approaches of the traffic signals A and B based on the local traffic flow, are unable to cope with this increase in queue length.

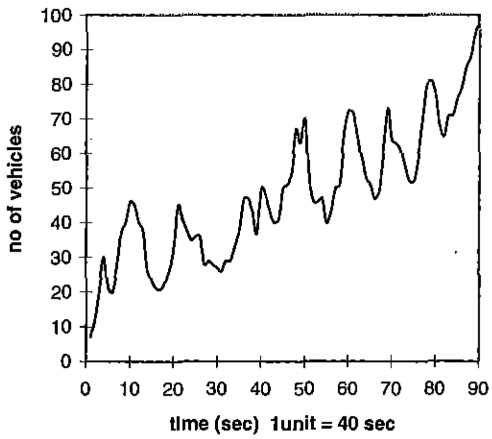


Figure 4.2 Queue length at intersection A (all approaches) - no offset adjustment

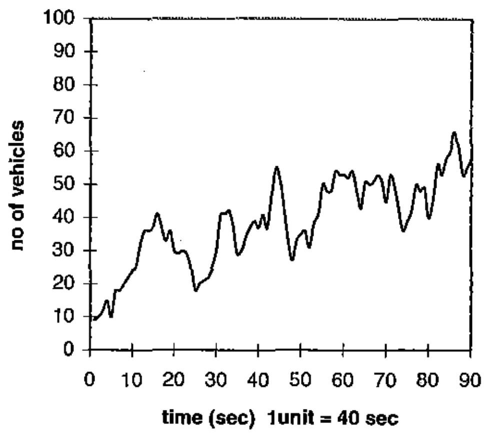


Figure 4.3 Queue length at intersection B (all approaches) - no offset adjustment

In order to obviate such a situation, the two intersections A and B are coordinated with each other by adjusting their respective offsets. The offset is adjusted by using local fuzzy logic controllers, which adjusts the offset at each intersection based on the traffic volume at the upstream intersection.

4.3 **Offset adjustment with two local fuzzy logic controllers**

In this section, in addition to the fuzzy logic traffic controller which adjusts the green phase of the traffic signal based on the queue length and the number of vehicles that passed through the intersection during the previous green phase, another set of fuzzy decision rules is used to adjust the offset between the two adjacent intersections.

Offset is the time relationship between the start of each phase, among adjacent intersections. Offset is adjusted to coordinate adjacent signals in a way that minimises stops in the direction where the traffic flow is heavy. A local fuzzy logic controller with a set of five rules is developed for adjusting the offset of an intersection. This local fuzzy logic controller adjusts the offset based on the traffic volume at the upstream intersection.

The local fuzzy logic controller receives the vehicle count at the north-south approach of the upstream intersection and the vehicle count at the east-west approaches of the local intersection. If the vehicle count at the upstream intersection is greater than the average vehicle count at the east-west approaches of the local intersection, the offset is adjusted by giving an extension to the green phase of the north-south approach of the local traffic signal. Figure 4.4 shows a block diagram of the two fuzzy logic traffic controllers and the two local fuzzy logic controllers. FLC1 is the fuzzy logic traffic controller and FLC2 is the local fuzzy logic controller.

In Figure 4.4, Vol_diff_A and Vol_diff_B are given by equations (4.1) and (4.2) respectively. V_NS_A and V_EW_A are the ratio of queue length to vehicles passed in the north-south and east-west approaches of intersection A. V_NS_B and V_EW_B are the ratio of queue length to

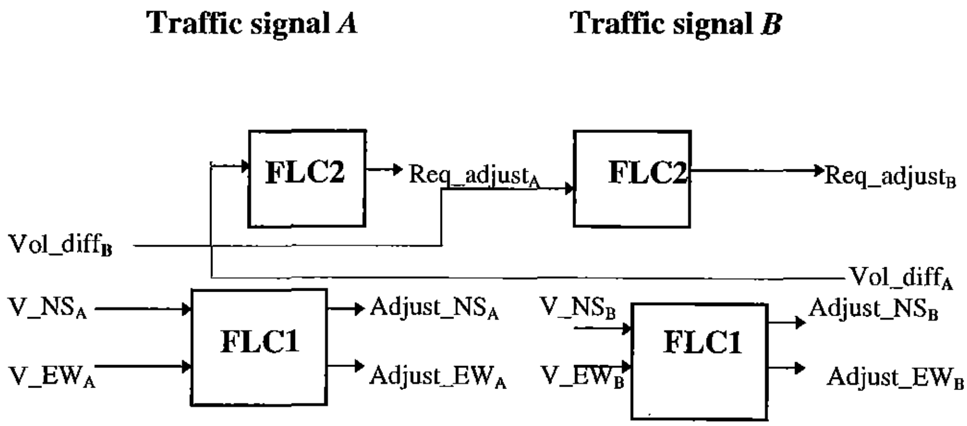


Figure 4.4 Block diagram of two traffic signals whose offset is adjusted by local fuzzy logic controllers

vehicles passed in the north-south and east-west approaches of intersection *B*. Adjust_NS$_A$ and Adjust_EW$_A$ are the adjustments to the green phase of the north-south and east-west approaches of the traffic signal *A*. Adjust_NS$_B$ and Adjust_EW$_B$ are the adjustments to the green phase of the north-south and east-west approaches of the traffic signal *B*. Req_adjust$_A$ and Req_adjust$_B$ are the offset adjustments to the north-south approach of intersections *A* and *B* respectively.

The input variable to the local fuzzy logic controller is the difference between the traffic volume at the upstream intersection and the average volume at the east-west approaches of the local intersection. The difference in volume, called *Vol_diff*, for intersection *A* is the difference between the number of vehicles waiting at the south approach of intersection *B* and the average number of vehicles waiting at the east-west approach of intersection *A*.

$$\text{Vol\_diff} = V_{SB} - ((V_{EA} + V_{WA}) /2) \qquad (4.1)$$

In equation (4.1), V_{SB} is the traffic volume at the south approach of intersection *B*, and V_{EA} and V_{WA} are the traffic volumes at the east and west approaches of intersection *A*.

Similarly, the *Vol_diff* for intersection *B* is the difference between the number of vehicles waiting at the north approach of intersection *A* and the average number of vehicles waiting at the east-west approach of intersection *B*.

$$\text{vol\_diff} = V_{NA} - ((V_{EB} + V_{WB}) /2) \qquad (4.2)$$

In equation (4.2), V_{NA} is the traffic volume at the north approach of intersection *A*, and V_{EB} and V_{WB} are the traffic volumes at the east and west approaches of intersection *B*.

The output of the local fuzzy logic controller determines whether the green phase of the north-south approach of the local traffic signal is to be further extended. This extension to the green phase of the local traffic signal is the offset adjustment. This adjustment, called *Req_adjust*, either extends the green phase or leaves it unchanged depending on the value of *Vol_diff*.

The local fuzzy logic controller at each intersection operates as a local controller coordinating the local intersection with the upstream intersection. If the local signal becomes green, then the vehicles will pass through the local intersection unstopped. Thus, each traffic signal is controlled by two fuzzy logic controllers - (i) a fuzzy logic traffic controller, comprising twenty five fuzzy rules, which adjusts the green phase of the north-south and east-west approaches of the signal based on the queue length and the vehicles passed during the previous green phase and (ii) a local fuzzy logic controller, comprising five fuzzy rules, which adjusts the offset of the intersection by adjusting the green phase of the north-south approach of the signal based on the traffic volume at the upstream intersection and the average volume at the east-west approaches of the local intersection.

The fuzzy linguistic terms for the input variable, *Vol_diff*, of the local fuzzy logic controller are:

VL : *Very Low*
LO : *Low*
MD : *Medium*
HI : *High*
VH : *Very High*

The fuzzy linguistic terms for the output variable, *Req_adjust*, of the local fuzzy logic controller are:

VS : *Very Small*
SM : *Small*
MD : *Medium*
HI : *High*
VH : *Very High*

The fuzzy sets and the membership functions for the input variable, *Vol_diff,* and the output variable, *Req_adjust,* are shown in Figures 4.5 and 4.6 respectively. The fuzzy rules for adjusting the offset is given in Table 4.1.

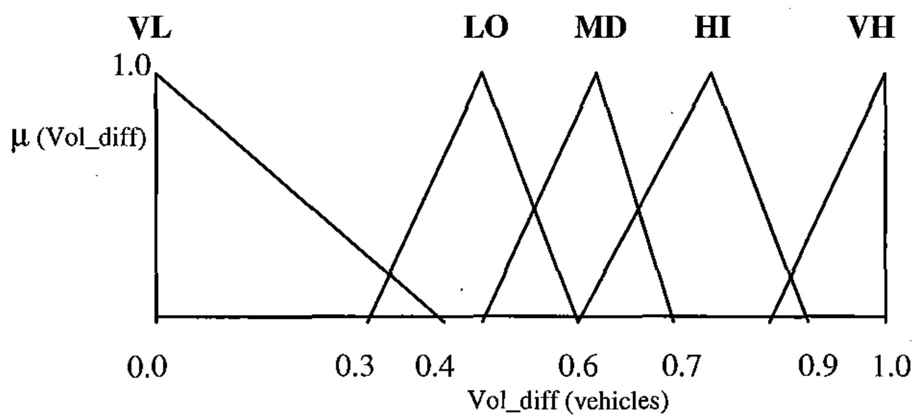


Figure 4.5 Membership functions for Vol_diff

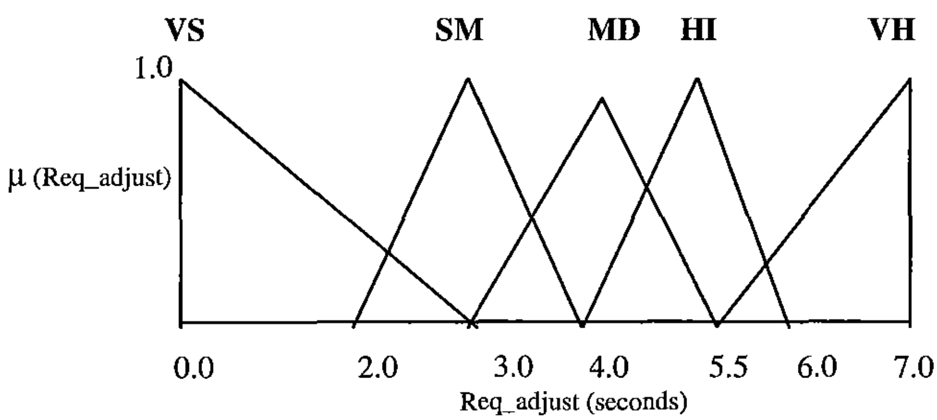


Figure 4.6 Membership functions for req_adjust

If the traffic volume at the either the north or south approach of the upstream intersection is high and the average volume at the east-west approaches of the local intersection is low, then the offset at the local intersection is adjusted by extending the green phase of the north-south approach of the local traffic signal so that the vehicles leaving the upstream intersection pass through the local intersection unstopped. The main purpose of adjusting offset is to coordinate the two intersections in such a way that the number of stops at both the intersections is minimised.

The fuzzy rules for the local fuzzy logic controller are given in Table 4.1:

if Vol_diff is Very Low (VL) **then Req_adjust is Very Small (VS)**
if Vol_diff is Low (LO) **then Req_adjust is Small (SM)**
if Vol_diff is Medium (MD) **then Req_adjust is Medium (MD)**
if Vol_diff is High (HI) **then Req_adjust is High (HI)**
if Vol_diff is Very High (VH) **then Req_adjust is Very High (VH)**

Table 4.1 Fuzzy Knowledge base for the local fuzzy logic controller

4.3.1 *Simulation Results*

Figures 4.7 and 4.8 show the number of vehicles waiting at at four approaches of intersections *A* and *B* respectively. By adjusting the offset at intersections *A* and *B*, the queue length does not escalate as in the case where there is no offset adjustment, see Figures 4.2 and 4.3. In Figure 4.7, when there is a heavy burst of traffic during the time instant 60-90, the fuzzy logic controller adjusting the offset does not allow the queue to build up and tries to minimise the number of vehicles waiting at intersection A. The queue length at intersection B, as shown in Figure 4.8, is less than that of intersection A. This is because the traffic flow approaching the south approach and the east-west approaches of intersection B is not heavy. The fuzzy logic controller adjusting the offset maintains the queue length at the same level throughout the simulation.

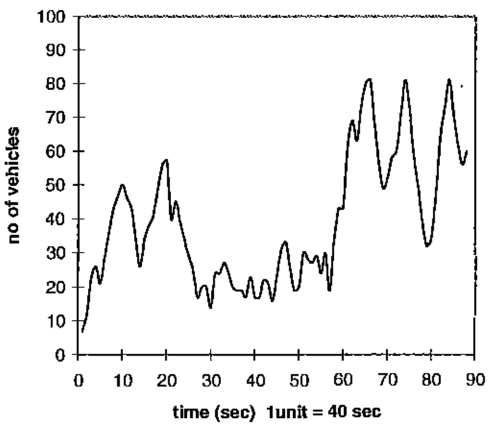


Figure 4.7 Queue length at intersection *A* (all four approaches) - 2 local FLC

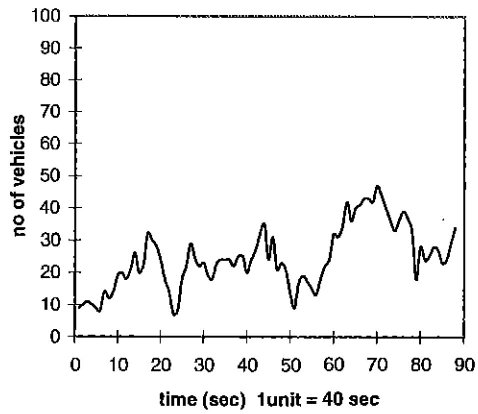


Figure 4.8 Queue length at intersection *B* - (all four approaches) - 2 local FLC

Figures 4.9 shows the effect of the local fuzzy logic controller on the number of vehicles waiting at the north approach of intersection A. By adjusting the offset of the traffic signal at intersection A, there is a reduction of about 15% in the number of vehicles waiting at the end of a cycle. Cycle is the total time duration of the green phase of north-south approach and green phase of east-west approach. The difference is not vast because the traffic volume arriving at the south approach of intersection B is not heavy and hence, the volume difference between the traffic volume at the south approach of B and the average volume at the east and west approaches of A is small, thereby resulting in very small adjustments to the offset at intersection A.

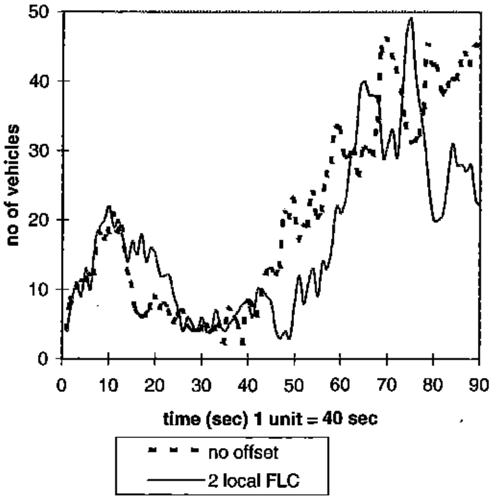


Figure 4.9 Queue length at the north approach of intersection A

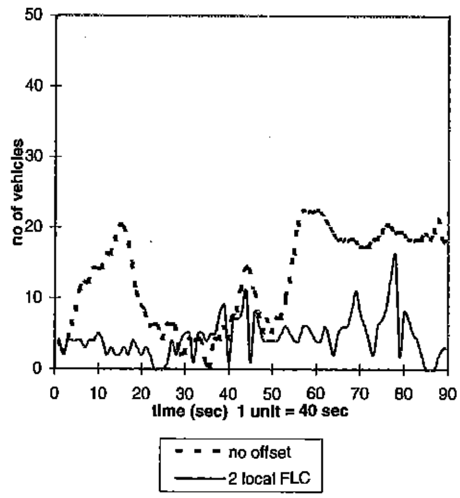


Figure 4.10 Queue length at the north approach of intersection B

Figures 4.10 shows the queue length at the north approach of intersection B. The effect of the local fuzzy logic controller is very pronounced in this case. The volume difference between the traffic volume at the north approach of intersection A and the average volume at east-west approaches of intersection B is large due to the high vehicle arrival rate in the north direction and a low arrival rate at the east and west approaches of intersection B (see Figure 4.1 for vehicle arrival rates). This large difference in volume results in extended durations to the green phase at the north-south approaches of intersection B. Thus, more vehicles coming from intersection A cross intersection B without stopping, thereby reducing the queue length at the north approach of intersection B by 65%.

In Figure 4.11, the outcome of the offset adjustment by the local fuzzy logic controller is a reduction of about 35% in the number of vehicles waiting at the south approach of intersection A. The unevenness of the traffic flow is illustrated in this figure when there is a sudden increase in the queue length when the offset is not adjusted. Figure 4.12 shows the number of vehicles waiting at the south approach of intersection B. An adjustment to the offset of signal at intersection B reduces the queue length by 33%.

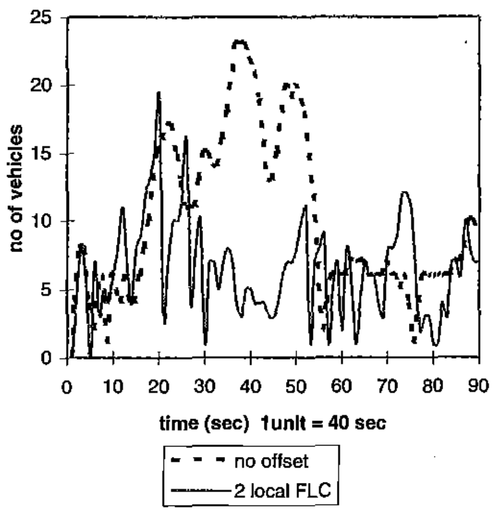


Figure 4.11 Queue length at the south approach of intersection A

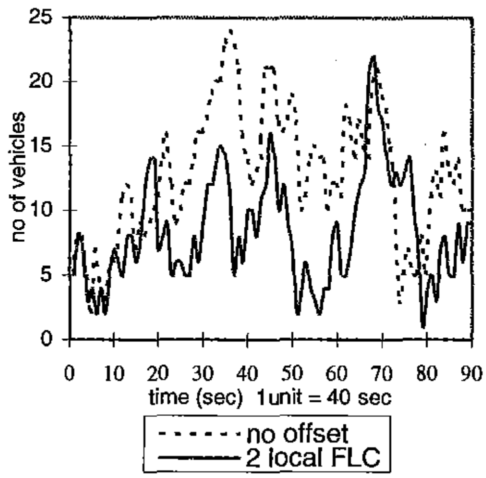


Figure 4.12 Queue length at the south approach of intersection B

In this section, two adjacent intersections are coordinated by adjusting the offset using a local fuzzy logic controller, comprising five fuzzy rules, at each intersection. This local controller coordinates each intersection with only its upstream intersection based on the traffic volume at the upstream intersection. Simulation results show the effectiveness of the local fuzzy logic controllers in making appropriate amounts of offset adjustment to the north-south approaches of intersections A and B thereby reducing the queue lengths at these approaches.

When more than two intersections are to be coordinated, using a local fuzzy logic controller at each intersection to adjust the offset may not be very effective. The local fuzzy logic controller receives information about the traffic conditions at only its upstream intersection. It has no knowledge of the traffic volumes at the other neighbouring intersections. In order to coordinate a network of intersections, a supervisory fuzzy logic controller is proposed, which adjusts the offset at each intersection based on the traffic volumes at all the neighbouring intersections. The supervisory fuzzy logic controller is expected to improve the flow of vehicles and minimise the number of vehicles waiting at all the intersections.

In the next section, a supervisory fuzzy logic controller is used to adjust the offset at intersections A and B. The supervisory fuzzy logic controller receives the traffic volume at the north approach of intersection A and that at the south approach of intersection B and decides on a course of action for adjusting the offset at intersections B and A respectively.

4.4 Offset adjustment with a supervisory fuzzy logic controller

In the previous section, the flow of vehicular traffic between two adjacent intersections was studied. Each intersection is coordinated with its upstream intersection using a local fuzzy logic controller with a knowledge base comprising five rules to adjust the offset between the two intersections. The offset is adjusted to minimise the number of stops between the traffic junctions.

Here, a supervisory fuzzy logic controller is developed to adjust the offset between the two adjacent intersections. The offset for both intersections is adjusted by a single supervisory fuzzy logic controller instead of two local fuzzy logic controllers as discussed in the previous section. A comparison of both techniques is made. Figure 4.13 shows a block diagram of a supervisory fuzzy logic controller and the two fuzzy logic traffic controllers.

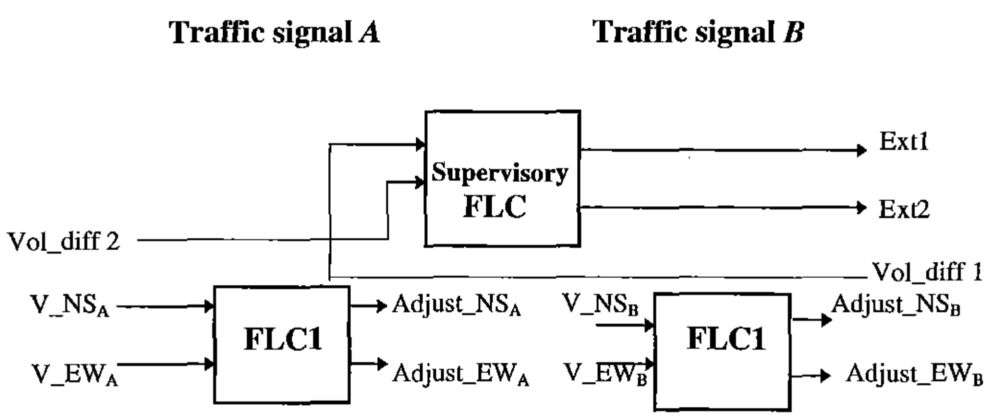


Figure 4. 13 Block diagram of two traffic signals whose offset is adjusted by a supervisory fuzzy logic controller

In Figure 4.13, Vol_diff2 and Vol_diff1 are given by equations (4.3) and (4.4) respectively. Ext1 and Ext2 are the offset adjustments to the north-south approach of intersections *A* and *B* respectively.

The supervisory fuzzy logic controller uses a set of twenty five rules to adjust the offset at each intersection. The inputs to the controller are:

(i) The volume difference, *Vol_diff 2*, between the traffic volume at the north approach of intersection *A* and the average volume in the east-west direction of intersection *B*.

$$\text{Vol\_diff } 2 = V_{NA} - \left((V_{EB} + V_{WB}) /2 \right) \qquad (4.3)$$

In equation (4.3), V_{NA} is the traffic density at the north approach of intersection *A*, and V_{EB} and V_{WB} are the traffic density at the east and west approaches of intersection *B*.

(ii) the volume difference, *Vol_diff 1,* between the traffic volume at the south approach of intersection *B* and the average volume at the east-west approaches of intersection *A*

$$\text{vol\_diff } 1 = V_{SB} - \left((V_{EA} + V_{WA}) /2 \right) \qquad (4.4)$$

In equation (4.4), V_{SB} is the traffic density at the south approach of intersection B, and V_{EA} and V_{WA} are the traffic density at the east and west approaches of intersection A.

The two outputs of the supervisory fuzzy logic controller are the extensions that have to be made to the green phase of the north-south approach of the two traffic signals *A* and *B*. *Ext1* is the extension to the green phase of the north-south approach of signal *A* and *Ext2* is the extension to the green phase of the north-south approach of signal *B*.

The fuzzy linguistic terms and the membership functions for the input and output variables are the same as those used by the two local fuzzy logic controllers to adjust the offset between the intersections (see Figures 4.5 and 4.6).

The fuzzy knowledge base of the supervisory fuzzy logic controller is shown in the form of a rule matrix in Table 4.2. Each entry in the table is made up of two components. The first is the offset adjustment given to the north-south approach of intersection *A* and the second is the offset adjustment given to the north-south approach of intersection *B*.

		Vol_diff 2				
		VL	**LO**	**MD**	**HI**	**VH**
Vol_diff 1	**VL**	VS VS	VS SM	VS MD	MD MD	MD HI
	LO	SM VS	SM SM	SM MD	MD HI	MD VH
	MD	MD VS	MD SM	MD MD	MD HI	MD VH
	HI	MD MD	HI MD	HI MD	HI HI	HI VH
	VH	HI MD	VH MD	HI MD	VH HI	VH VH

Table 4.2 - Fuzzy Knowledge base used by supervisory FLC for adjusting offset

In this section, a supervisory fuzzy logic controller is proposed to adjust the offset at two adjacent intersections. The supervisory fuzzy logic controller takes into account the traffic volume at all the neighbouring intersections rather than just considering the traffic volume at the upstream intersection. It is expected to be a better alternative to the local fuzzy logic controllers when more than two intersections are to be coordinated.

4.4.1 *Simulation Results*

Figures 4.14 to 4.17 show a reduction in the number of vehicles when the offset is adjusted by a supervisory fuzzy logic controller. There is a significant reduction in the queue length at the north approach of intersection *B* and at the south approach of intersection *A*. This is because, by adjusting the offset at intersection *B* based on the traffic volume at the north approach of intersection *A*, most of the vehicles leaving intersection *A* also pass through intersection *B* unstopped. Similarly, by adjusting the offset at intersection *A* based on the traffic volume at the south approach of intersection *B*, the vehicles leaving intersection *B* pass through intersection *A* unstopped.

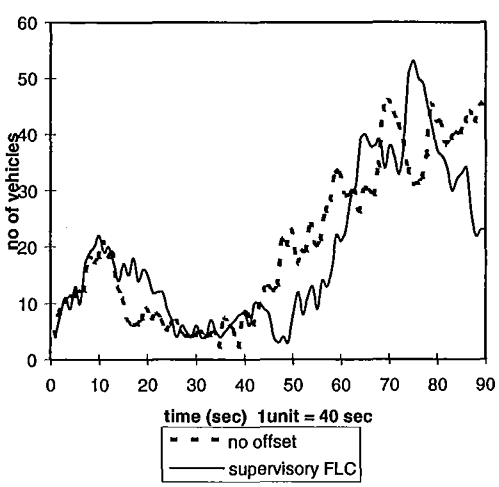


Figure 4.14 Queue length at the north approach of intersection *A*

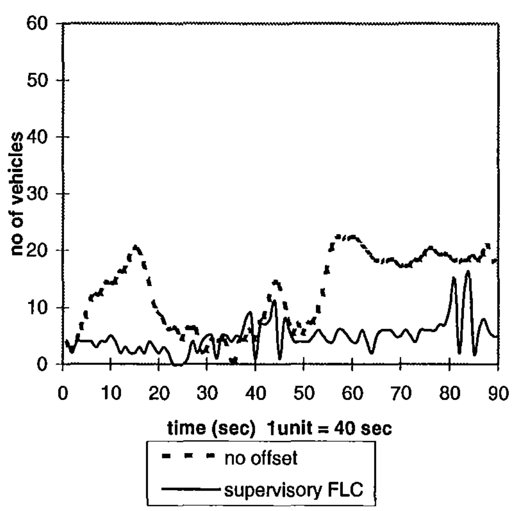


Figure 4.15 Queue length at the north approach of intersection *B*

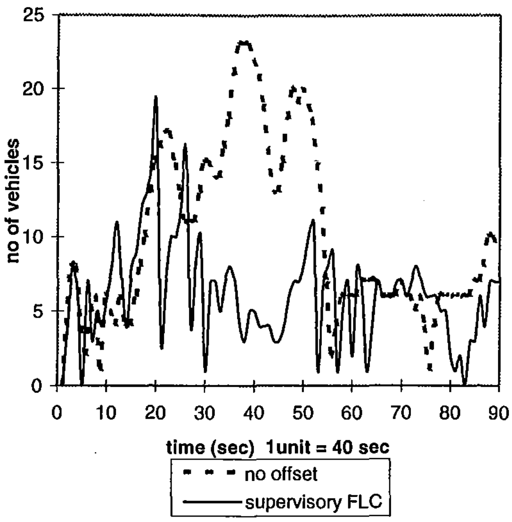


Figure 4.16 Queue length at the south approach of intersection *A*

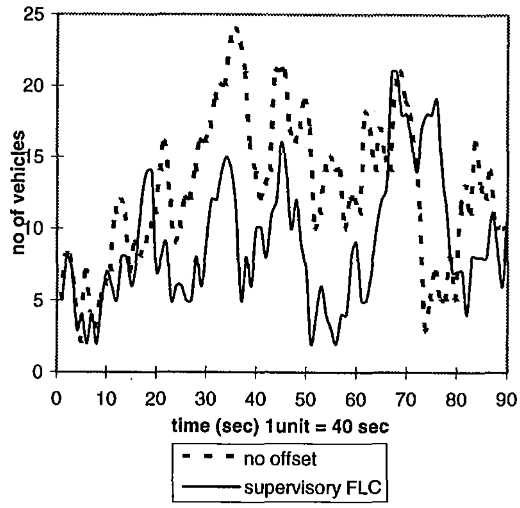


Figure 4.17 Queue length at the south approach of intersection *B*

Figures 4.18 and 4.19 show the number of vehicles waiting at intersections *A* and *B*. When the two signals are coordinated by the supervisory fuzzy logic controller, the number of stops at both intersections is reduced. When there is an increase in the queue length at

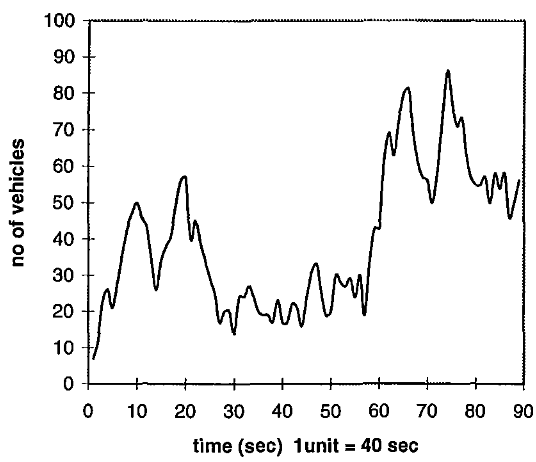


Figure 4.18 Queue length at intersection *A* (all four approaches) - supervisory FLC

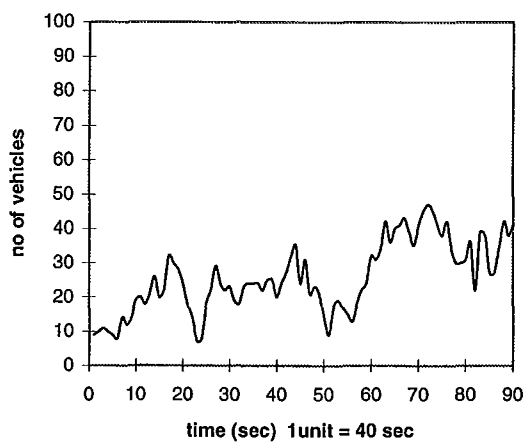


Figure 4.19 Queue length at intersection *B* (all four approaches) - supervisory FLC

intersection A, as can be seen in Figure 4.18, the fuzzy logic controller adjusting the offset at intersection A gives large extensions to the green phase of the north-south approach of the signal, thereby reducing the queue length. Thus, the queue lengths at intersections A and B are restricted by adjusting the offset.

The supervisory fuzzy logic controller attempts to minimise the queue length at the intersections A and B by adjusting the offset. But, it does not show any marked improvement over the performance of the local fuzzy logic controllers. Both the supervisory and the local fuzzy logic controllers adjust the offset by a similar amount. This is because both the controllers receive the same input information from the sensors. When more than two intersections are to be coordinated, the performance of the supervisory and the local fuzzy logic controllers differs because the supervisory controller adjusts the offset based on the traffic volume at all the intersections while the local fuzzy logic controller only recieves the traffic volume at the north-south approach of the upstream intersection.

4.5 Simulation results

Further simulations were performed to establish the effectivenss of offset adjustment at intersections A and B. The above figures show the improvement brought into the queue length by adjuting the offset at intersections A and B using either two local fuzzy logic controllers or a supervisory fuzzy logic controller. The offset adjustment reduces the queue length at the north-south approaches of both intersections.

The number of vehicles waiting at the north and south approaches of intersection B is less compared to that of intersection A when the offset is adjusted at both the signals. This is due to the high vehicle arrival rate at the north direction resulting in a heavy vehicle volume at the north approach of intersection A. Since the offset at intersection B is adjusted based on the traffic volume at the north approach of intersection A, a heavy traffic volume at the north approach of intersection A results in extended durations to the green phase of the signal at the north-south approach of intersection B, thereby reducing the number of stops at the north approach of intersection B.

Figures 4.20 - 4.25 show the average time spent by each vehicle from any approach at the intersections A and B. As the queue length increases, so does the average waiting time. The

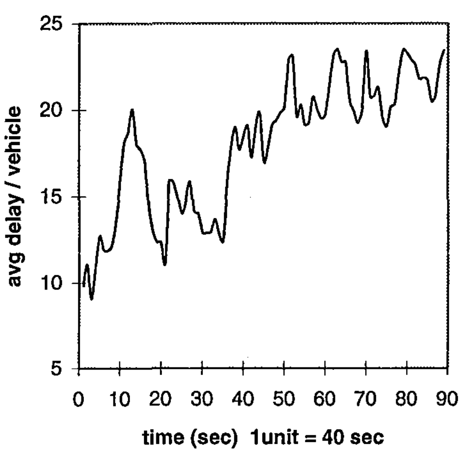


Figure 4.20 Average delay / vehicle for all four approaches at intersection A - no offset

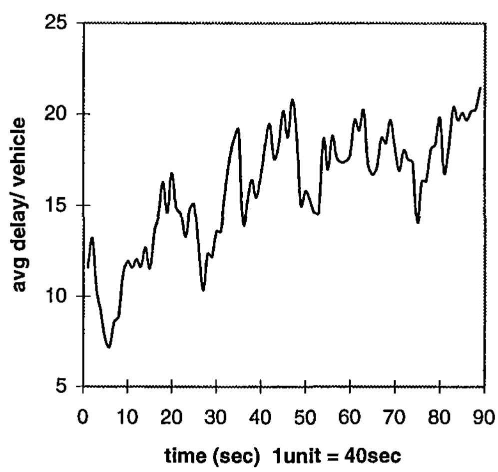


Figure 4.21 Average delay/vehicle for all four approaches at intersection B - no offset

time spent in waiting is more at intersection A than at intersection B because of the high vehicle arrival rates at the north approach of intersection A and the east-west approaches of

A in comparison to the arrival rate at the south approach of intersection *B* and east-west approaches of intersection *B*.

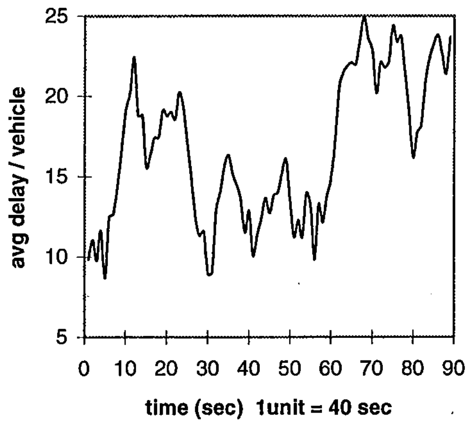


Figure 4.22 Average delay / vehicle for all four approaches at intersection *A* - local FLC

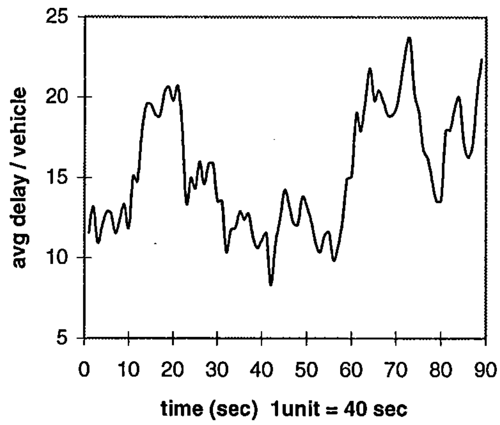


Figure 4.23 Average delay/vehicle for all four approaches at intersection *B* - local FLC

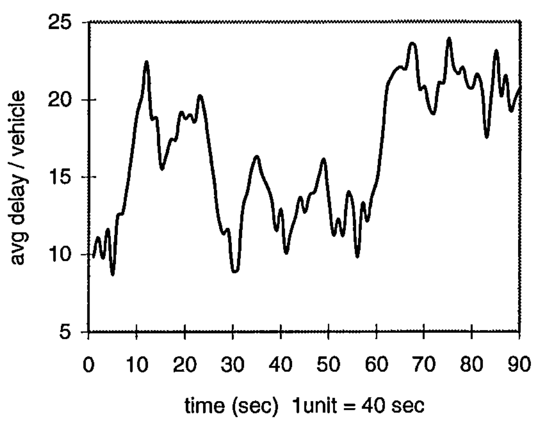


Figure 4.24 Average delay / vehicle for all four approaches at intersection *A* - supervisory FLC

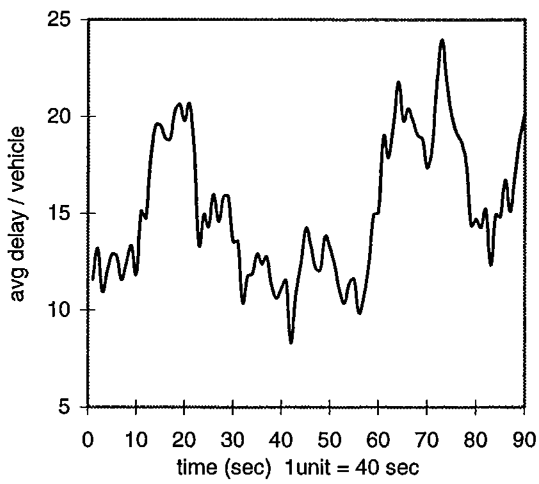


Figure 4.25 Average delay/vehicle for all four approaches at intersection *B* - supervisory FLC

The following statistics were arrived from the above figures:

<u>Average waiting time at intersection *A* :</u>

No offset adjustment - 17.92 sec

Two local FLC - 16.7 sec

Supervisory FLC - 16.5 sec

Average waiting time at intersection *B* :

No offset adjustment - 15.9 sec

Two local FLC - 15.39 sec

Supervisory FLC - 15.2 sec

The average time spent by a vehicle, at both the intersections, waiting for the signal to change to green is reduced when there is an offset adjustment at both the intersections. As an outcome of the adjustment, the green phase duration is increased, thereby allowing more vehicles to flow across the intersection. A decrease in the number of stops at the intersection constitutes a reduction in the average waiting time. From the statistics, it can also be seen that the supervisory fuzzy logic controller is more effective in reducing the waiting time than the local fuzzy logic controllers.

4.6 Discussion

The fuzzy control of two traffic signals placed at adjacent intersections was investigated. Two fuzzy logic control schemes were introduced to coordinate the adjacent traffic signals by adjusting the offset. Simulations were run to establish the effectiveness of the two control schemes. The simulation results showed that when there is no offset adjustment at the intersections, the number of vehicles waiting tends to accumulate resulting in longer queues at all approaches. A significant reduction in the queue length is noticed when the offset is adjusted by either the local fuzzy logic controller or the supervisory fuzzy logic controller.

The number of vehicles waiting at all four approaches of intersections A and B, and the average time spent by a vehicle in waiting at the intersections are reduced by adjusting the offset. The performance of the supervisory fuzzy logic controller and the local fuzzy logic controller, which adjusts the offset, are equally good with neither one performing better than the other. This is because the control of only two adjacent traffic signals is considered.

The effectiveness of the supervisory fuzzy logic controller is evident in chapter 5 where the traffic flow approaching a set of three intersections is studied. A set of three traffic signals is coordinated by a supervisory fuzzy logic controller which makes adjustements to the offset at all three intersections.

Chapter 5

Fuzzy control of a set of three intersections

In chapter 3, a fuzzy logic controller to regulate the flow of traffic approaching an isolated traffic junction, is studied. The controller responds to fluctuations in the traffic flow in an effective way by making appropriate on-line adjustments to the green phase splits of the traffic signal based on the traffic densities at the north-south and east-west approaches. Any change in the traffic conditions results in a corresponding change in the time duration to the green phase of the traffic signal.

In chapter 4, the traffic flow approaching two adjacent intersections is regulated by coordinating the two intersections. Each intersection is coordinated with its adjacent intersection by making adjustments to the offset of each signal. Two fuzzy control schemes for adjusting the offset at an intersection, based on the traffic volume at the adjacent intersections, were discussed. In the first scheme, the offset of a traffic signal is adjusted by a local fuzzy logic controller which makes the adjustment based on the volume difference between the traffic volume at the upstream intersection and the average volume at the east-west approaches of the local intersection. The local fuzzy logic controller coordinates each intersection with only its upstream intersection. In the second control scheme, the offset adjustment is made by a supervisory fuzzy logic controller which makes the offset adjustments based on the traffic volume at both the intersections rather than that of just the upstream intersection.

In this chapter, the two fuzzy control schemes that were introduced in the previous chapter will be applied to the control of traffic signals at three intersections. The three intersections are in a straight line in the north-south direction, with the north and south directions being the dominant directions of traffic flow. The purpose of adjusting the offset is to minimise the queue length at the north and south approaches of the three intersections.

5.1 The Model

The model that was introduced in chapter 4 is now extended to the control of three signalised intersections. The three intersections are denoted by *A*, *B*, and *C*. A mean vehicle arrival rate is assigned to each end of the street as shown in Figure 5.1. The mean vehicle arrival rate is assigned to the north approach of intersection *A*, the south approach of intersection C and the east and west approaches of intersections *A*, *B*, and *C*. At each simulation step, a random number is generated and a vehicle may be added to the queue based on the mean vehicle arrival rate.

If the traffic signal is red for an approach at any of the three intersections, each vehicle arriving at that approach adds to the queue of vehicles at that approach.The time spent by each vehicle in waiting for the phase to change is determined from the length of the red phase and the time instant the vehicle joined the queue.

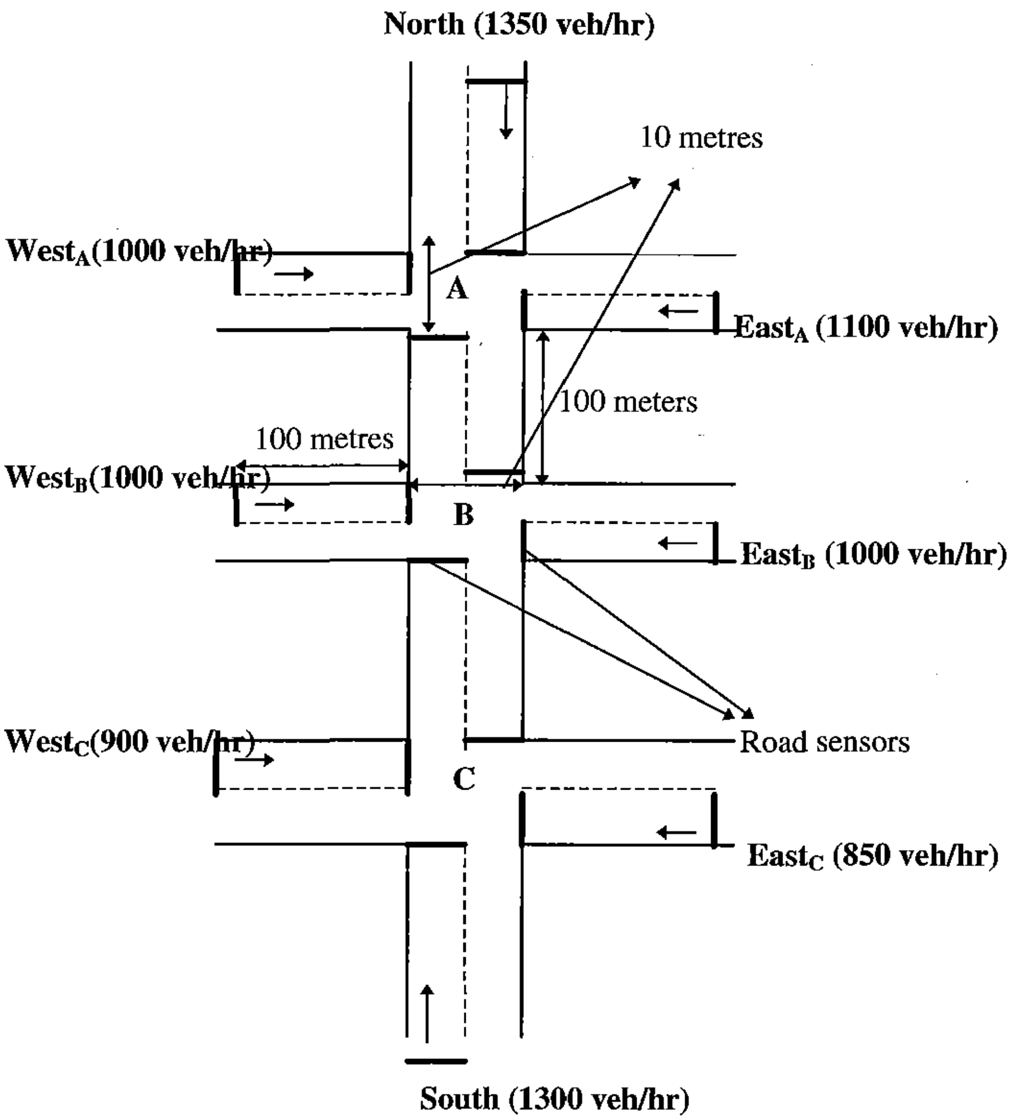


Figure 5.1 A set of three intersections

When a vehicle from the north approach of intersection *A* crosses this traffic junction, the time taken by the vehicle to reach intersection *B* is calculated from the distance between the two intersections, which is known and velocity of the vehicle, which is determined from equation (3.2). If this time period is less than the length of the green phase of the north-south approach at intersection *B*, then the vehicle passes through intersection *B* unstopped. Similarly, the time taken by a vehicle crossing intersection *B* and reaching intersection *C* is calculated and if it is less than the length of the green phase of the north-south approach at intersection *C*, the vehicle passes through unstopped. The number of vehicles passing through intersection *C* and reaching intersection *A* is also determined in a similar way.

5.2 Three intersections with no offset adjustment

The three intersection *A*, *B*, and *C* are aligned in a straight line in the north-south direction as shown in Figure 5.1. The vehicle flow approaching each of the three intersections is regulated by a set of twenty five fuzzy decision rules which adjust the green phase splits of the traffic signal for the north-south and east-west approaches. The adjustments are made to the green phase based on the ratio of the queue length at the respective approaches to the number of vehicles that passed through the intersection during the previous green phase.

The fuzzy decision rules that adjust the green phase have been explained in detail in chapter 3. The fuzzy sets and the membership for the input and output variables are the same as those used in chapter 3. The three signals are not coordinated with each other since their respective offsets are not adjusted. All the three signals operate individually based only on the local traffic information.

5.2.1 *Simulation Results*

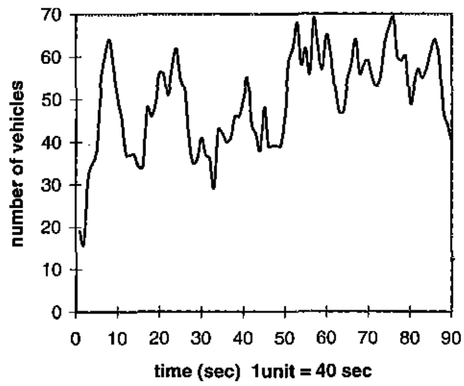


Figure 5.2 Queue length at all four approaches of intersection *A* - no offset adjustment

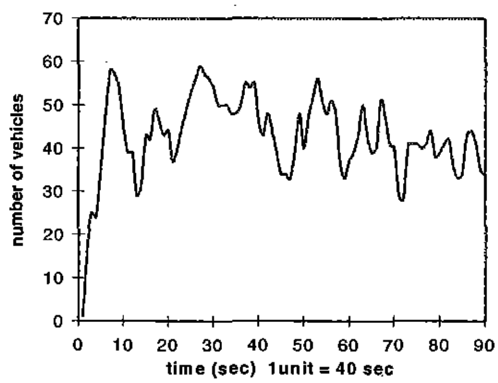


Figure 5.3 Queue length at all four approaches of intersection *B* - no offset adjustment

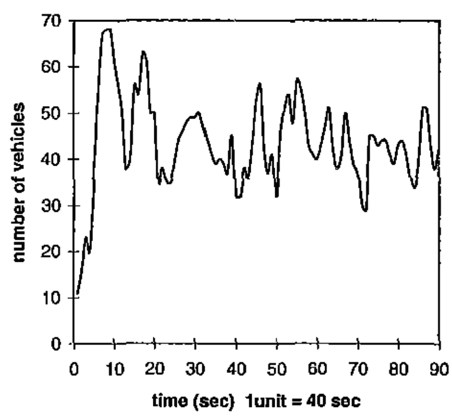


Figure 5.4 Queue length at all four approaches of intersection *C* - no offset adjustment

Figure 5.2, 5.3, and 5.4 show the number of vehicles waiting at the intersections *A*, *B*, and *C*. With no coordination between the three intersections, the queue length at the approaches to the intersections is not reduced. The average queue length at the end of a cycle at intersection A is 49, at intersection B is 42, and at intersection C is 43. If the queue is allowed to build up, it might hamper the traffic flow at the other intersections resulting in congestion and increased delay time.

In order to avoid such a situation, each intersection is coordinated with its neighbouring intersections by adjusting the offset of each traffic signal. The offset at each intersection is adjusted by a local fuzzy logic controller which coordinates each intersection with only its upstream intersection.

5.3 Offset adjustment with a local fuzzy logic controller

Offset is adjusted to coordinate the local signal with its adjacent signals in such a way that the vehicles arriving at the local intersection pass through unstopped. In this section, the offset at each intersection is adjusted by a local fuzzy logic controller, comprising five fuzzy rules, located at each intersection. Each intersection has two upstream intersections - one in the north and one in the south. The upstream intersection having the greater queue length is determined and the local fuzzy logic controller coordinates the local intersection with this upstream intersection by adjusting the offset at the local intersection. For example, the offset at intersection B is adjusted based either on the traffic volume at the north approach of intersection A or on the traffic volume at the south approach of intersection C. If the traffic volume at the north approach of intersection A is greater than that at the south approach of intersection *C*, then intersection *B* is coordinated with intersection *A*, else with intersection *C*.

If the traffic volume at the upstream intersection is high and the average volume at the east-west approaches of the local intersection is low, then the green phase for the local controller is extended by adjusting the offset, so that the vehicles leaving the upstream intersection pass through the local intersection unstopped. The main purpose of the offset adjustment is to minimise the number of stops at the intersection thereby reducing the waiting time for each vehicle.

Thus, each of the three traffic signals is controlled by two fuzzy logic controllers. The first controller consists of twenty five fuzzy decision rules for adjusting the green phase splits based on the local traffic volume and the number of vehicles that passed through the intersection during the previous green phase (explained in chapter 3). The second controller consists of a set of five decision rules for adjusting the offset of the signal based on the traffic volume at the upstream intersection and the average volume at the east-west approaches of the local intersection.

The local fuzzy logic controller, adjusting the offset at intersection *A* coordinates this intersection with intersection *B*, the fuzzy logic controller at intersection *B* coordinates this intersection with either intersection *A* or intersection *C*, depending upon which intersection has a greater waiting traffic, and the fuzzy logic controller at intersection *C* coordinates intersection C with intersection *B*.

The input variable to the local fuzzy logic controller is the difference in volume, *Vol_diff*, between the traffic volume at the upstream intersection and the average traffic volume at the east-west approaches of the local intersection. The output of the fuzzy logic controller is the amount of adjustment that has to be made to the current green phase of the signal, *Req_adjust*.

The volume difference, *Vol_diff*, at intersection *A* is given by:

$$\text{Vol\_diff} = V_{SB} - (V_{EA} + V_{WA})/2 \qquad (5.1)$$

In equation (5.1), V_{SB} is the traffic volume at the south approach of intersection *B*, V_{EA} is the traffic volume at the east approach of intersection *A*, and V_{WA} is the traffic volume at the west approach of intersection *A*.

The volume difference, *Vol_diff*, at intersection *B* is given by:

$$\text{Vol\_diff} = V_{NA} - (V_{EB} + V_{WB})/2 \qquad (5.2\text{-a})$$

$$\text{Vol\_diff} = V_{SC} - (V_{EB} + V_{WB})/2 \qquad (5.2\text{-b})$$

In equations (5.2-a and 5.2-b), V_{NA} is the traffic volume at the north approach of intersection *A*, V_{SC} is the traffic volume at the south approach of intersection *C*, V_{EB} is the traffic volume at the east approach of intersection *B*, and V_{WB} is the traffic volume at the west approach of intersection *B*.

If the traffic volume at the north approach of intersection *A* is greater than the traffic volume at the south approach of intersection *C*, equation (5.2-a) is used to determine the volume difference else equation (5.2-b) is used.

The volume difference, *Vol_diff*, at intersection *C* is given by:

$$\text{Vol\_diff} = V_{NB} - (V_{EC} + V_{WC})/2 \qquad (5.3)$$

In equation (5.3), V_{NB} is the traffic volume at the south approach of intersection *B*, V_{EC} is the traffic volume at the east approach of intersection *C*, and V_{WC} is the traffic volume at the west approach of intersection *C*.

The input and output fuzzy membership functions are shown in Figures 5.5 and 5.6 respectively.

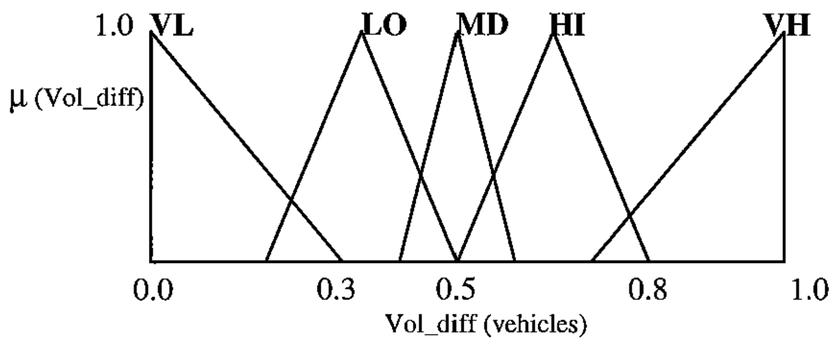


Figure 5.5 Membership functions for Vol_diff

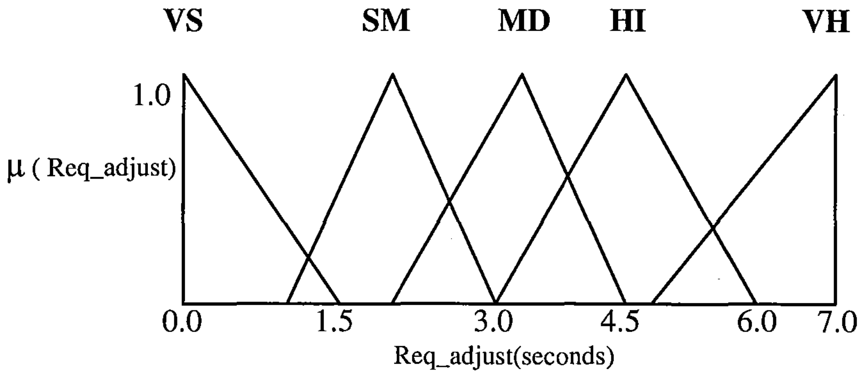


Figure 5.6 Membership functions for Req_adjust

The knowledge base of the local fuzzy logic controller comprises five fuzzy decision rules as shown in Table 4.1 in chapter 4. The input and the output linguistic terms are same as the ones used for the control of two intersections (see chapter 4).

5.3.1 *Simulation Results*

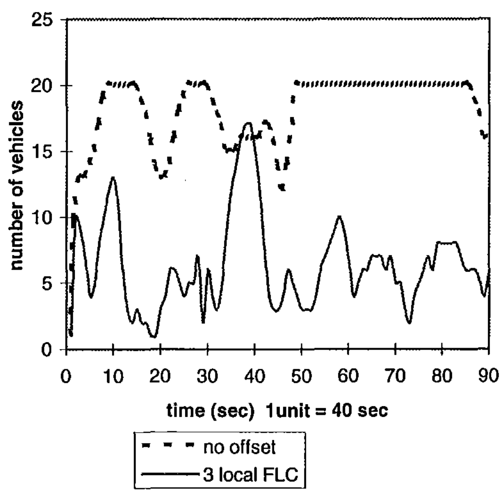


Figure 5.7 Queue length at the north approach of intersection *B*

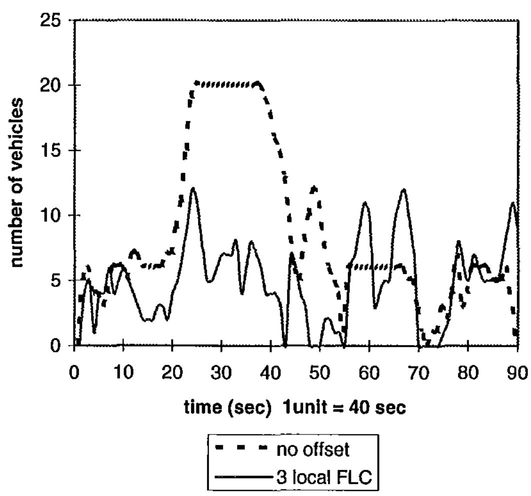


Figure 5.8 Queue length at the south approach of intersection *B*

The queue length at the north approach and the south approach of intersection B is shown in Figures 5.7 and 5.8 respectively. The queue length is reduced by 65% at the north approach of intersection B as a result of the offset adjustment at intersection B. This large reduction is due to the heavy traffic volume at north approach of intersection A. Since the offset adjustment at intersection B is made based on this traffic volume, the traffic flow is improved and the queue length is reduced.

In Figure 5.8, the queue length at the south approach of intersection B is reduced by 42%. The reduction is not as large as that of the north approach of intersection B because the traffic flowing from intersection C to the south approach of intersection B is less compared to the traffic flowing from intersection A to the north approach of intersection B.

In this section, a local fuzzy logic controller adjusts the offset at an intersection based only on the traffic volume at the upstream intersection. It does not have any knowledge of the traffic situation at the intersections other than that at the upstream intersection. For example, the traffic signal at intersection A is aware only of the conditions prevailing at intersection B and not that of intersection C. As a result, the queue lengths at the north and south approaches of intersection B are minimised but there is not much reduction in the queue length at the north-south approaches of intersections A and C. Thus, the traffic flow is optimised at only intersection B when the offset is adjusted by a local fuzzy logic controller.

In the next section, a supervisory fuzzy logic controller is proposed for adjusting the offset at the three intersections based on the traffic volume at all the intersections. It is expected to achieve a better performance at all the intersections rather than just improving the traffic flow at a single intersection.

5.4 Offset adjustment with a supervisory fuzzy logic controller

In the previous section, the traffic flow approaching a set of three intersections was discussed. Each intersection is coordinated with only its upstream intersection by adjusting the offset at each intersection. The offset is adjusted by a local fuzzy logic controller whose knowledge base consists of five fuzzy control rules to adjust the offset.

In this section, a supervisory fuzzy logic controller comprising twenty seven rules is developed to coordinate the three intersections. The supervisory fuzzy logic controller adjusts the offset of the three traffic signals by evaluating the traffic volume at all three intersections and then decides on the amount of extension that is to be made to the green phase of each signal.

The supervisory fuzzy logic controller has three input and three output variables. The input variables are :

(i) the volume difference (*Vol_diff1*) between the traffic volume at the south approach of intersection *C* (V_{SC}) and the average volume at the east-west approaches of intersection *A* (V_{EA} and V_{WA});

$$\text{Vol\_diff1} = V_{SC} - ((V_{EA} + V_{WA}) /2) \qquad (5.4)$$

(ii) the volume difference (*Vol_diff2*) between the traffic volume at the south approach of intersection *C* (V_{SC}) or at the north approach of intersection *A* (V_{NA}) depending on which direction the traffic flow is high) and the average volume at the east-west approaches of intersection *B* (V_{EB} and V_{WB});

$$\text{Vol\_diff2} = V_{NA} - ((V_{EB} + V_{WB}) /2) \qquad (5.5\text{-a})$$

$$\text{Vol\_diff2} = V_{SC} - ((V_{EB} + V_{WB}) /2) \qquad (5.5\text{-b})$$

(iii) the volume difference (*Vol_diff3*) between the traffic volume at the north approach of intersection *A* (V_{NA}) and the average volume at the east-west approaches of intersection *C* (V_{EC} and V_{WC});

$$\text{Vol\_diff3} = V_{NA} - ((V_{EC} + V_{WC}) /2) \qquad (5.6)$$

The fuzzy output variables are the extensions that are to be made to the green phase of the north-south approaches of the three traffic signals. The outputs are represented as: *Ext1*, *Ext2*, and *Ext3*.

Each of the three input variables of the supervisory fuzzy logic controller is divided into three fuzzy sets and each output variable is divided into five fuzzy sets. Each input variable is divided into three fuzzy sets to simplify the construction of the knowledge base. An increase in the number of input variables and fuzzy sets results in a knowledge base of very high dimensionality which might be difficult to construct. To ease the task of constructing a knowledge base, each input variable is divided into three fuzzy sets thereby yielding 27 fuzzy rules. The sensitivity of the fuzzy logic controller is certainly reduced as a result of restraining the number of fuzzy sets of the input variables to three but the effectiveness with which the supervisory fuzzy logic controller deals with the fluctuations in the traffic pattern is still commendable when compared to the performance of the local fuzzy logic controller.

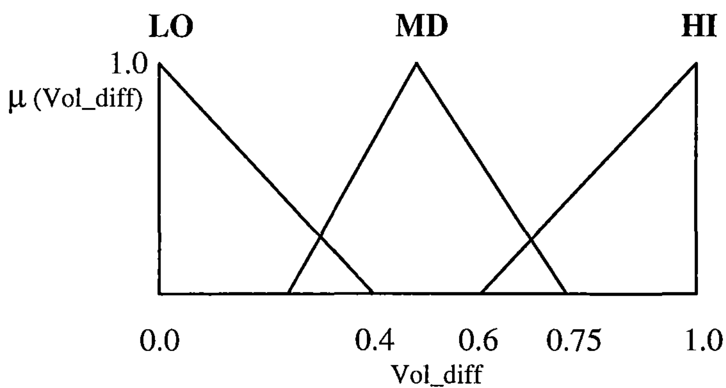


Figure 5.9 Membership functions for the input fuzzy sets, Vol_diff1,Vol_diff2,Vol_diff3 of the supervisory FLC

The membership functions for the input fuzzy sets is shown in Figure 5.9. The membership functions of the output fuzzy sets, same as those used for the local fuzzy logic controller, is shown in Figure 5.6. The fuzzy knowledge base comprising 27 rules is shown in Appendix A.

5.4.1 *Simulation Results*

Figure 5.10 shows the effect of the local fuzzy logic controller and the supervisory fuzzy logic controller on the queue lengths at the north approach of intersection *C*. The superiority of the supervisory fuzzy logic controller is evident in this case. The local fuzzy logic controller adjusts the offset at intersection *C* based on the traffic volume at the north approach of intersection *B*. The supervisory fuzzy logic controller adjusts the offset based on the traffic volume at the north approach of intersection *A* and the volume difference at the other two approaches..

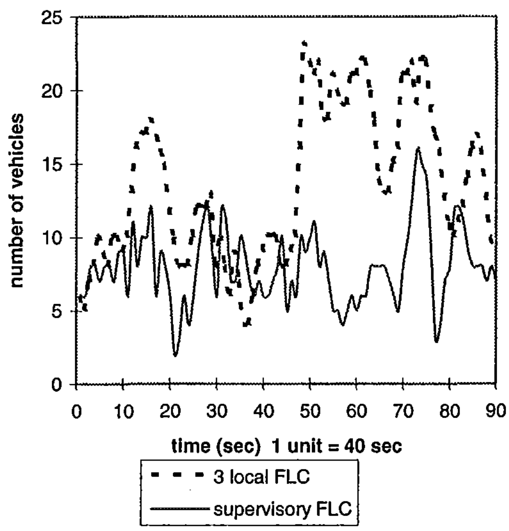


Figure 5. 10 Queue length at the north approach of intersection *C*

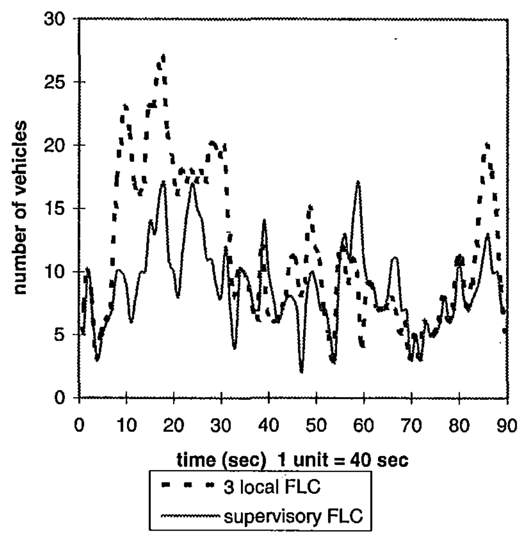


Figure 5. 11 Queue length at the south approach of intersection *C*

Since the vehicle arrival rate at the north approach of intersection *A* is higher than that at the north approach of intersection *B*, offset adjustment using a supervisory fuzzy logic controller results in extended time durations to the the green phase of the north-south approach of the signal at intersection *C*. As a result, the traffic flowing across intersection

C is increased and the number of vehicles waiting at the north approach of intersection C is reduced by 41%. The extended time durations to the green phase of the north-south approach of signal C also reduces the queue length at the south approach of intersection C by 22%, as shown in Figure 5.11. In can be inferred from the above two figures, that offset adjustment using a supervisory fuzzy logic controller is a better option than using local fuzzy logic controllers, when more than two traffic signals are to be coordinated.

5.5 Simulation results

Further simulations were performed to study the behaviour of the local fuzzy logic controllers and the supervisory fuzzy logic controller in adjusting the offset of the three intersections.

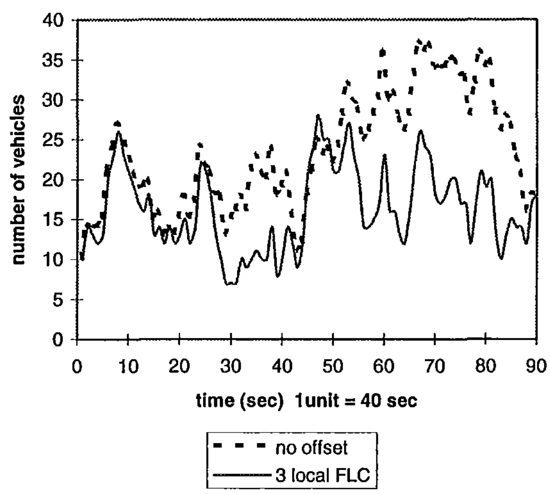


Figure 5.12 Queue length at the north approach of intersection *A*

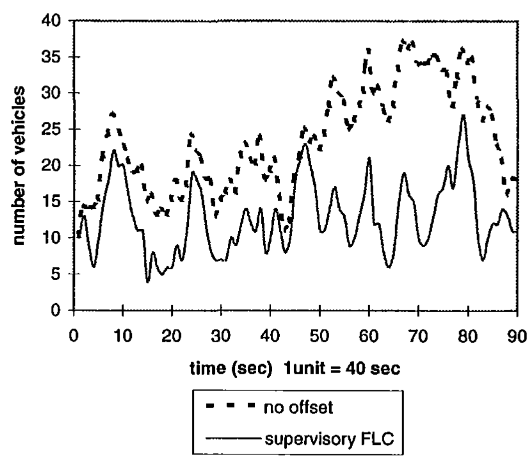


Figure 5. 13 Queue length at the north approach of intersection *A*

Figures 5.12 and 5.13 show the number of vehicles waiting at the north approach of intersection A. When there is no offset adjustment at this intersection, the queue length increases with time. Since there is no controller adjusting the offset at intersection A, knowledge regarding the traffic volume at the other intersections is not available and the green phase can be increased to a maximum of only 28 seconds by the fuzzy logic traffic controller. When a local fuzzy logic controller is used to adjust the offset at intersection A, the reduction in the number of vehicles waiting is 30% while the supervisory fuzzy logic controller brings about a reduction of 45% by adjusting the offset at the north-south of this intersection, see Figures 5.12 and 5.13. The performance of the supervisory fuzzy logic controller is better than the local fuzzy logic controller because the latter only considers the traffic volume at the upstream intersection, which in this case is south approach of intersection B, while the supervisory controller takes into consideration the traffic volume at all the intersections before invoking a decision regarding the offset adjustment.

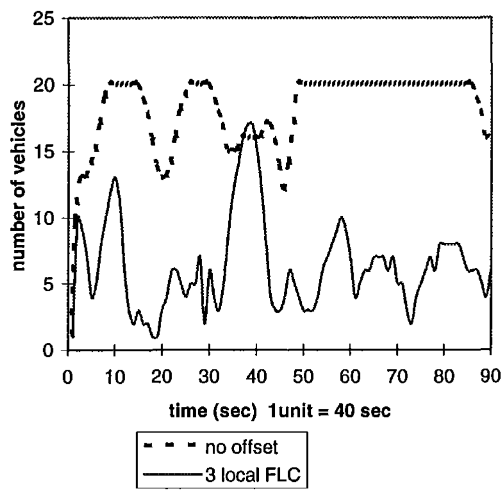


Figure 5. 14 Queue length at the north approach of intersection B

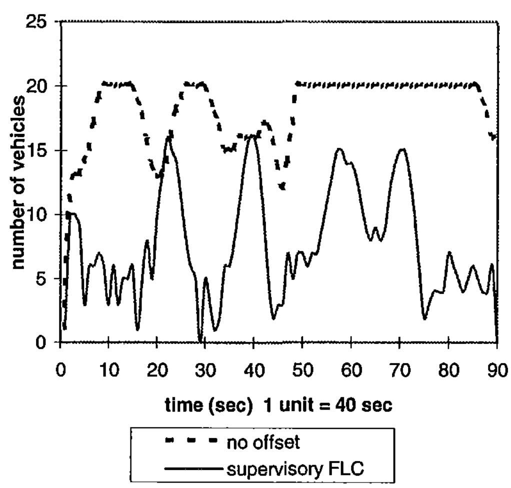


Figure 5. 15 Queue length at the north approach of intersection B

The queue length at the north approach of intersection B is shown in the Figures 5.14 and 5.15. As expected, there is a marked reduction in the queue length when the offset is adjusted by the local fuzzy logic controller or by the supervisory fuzzy logic controller. Even though the offset at the north-south approach of intersection B is adjusted based on the traffic volume at either the north approach of intersection A or at the south approach of intersection C for both the local fuzzy logic controller and the supervisory fuzzy logic controller, the queue length is reduced by about 65% in the former instance and only by 58% in the latter case. This could be due to the reduced number of fuzzy sets of the input parameters of the supervisory controller. The input variable is divided into three fuzzy sets, in order to facilitate the construction of the rule base. As a result, the supervisory controller is less sensitive to the abrupt changes in the queue length.

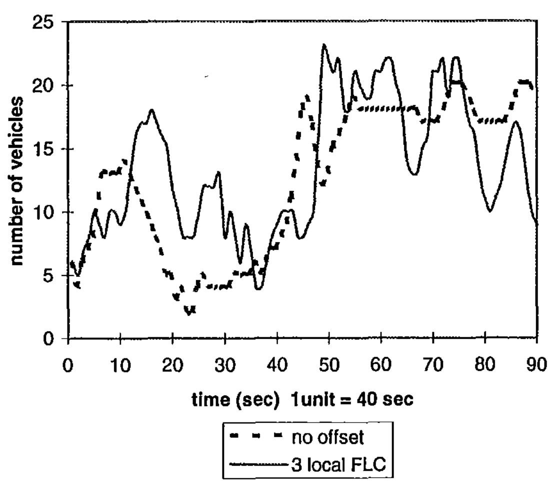


Figure 5. 16 Queue length at the north approach of intersection C

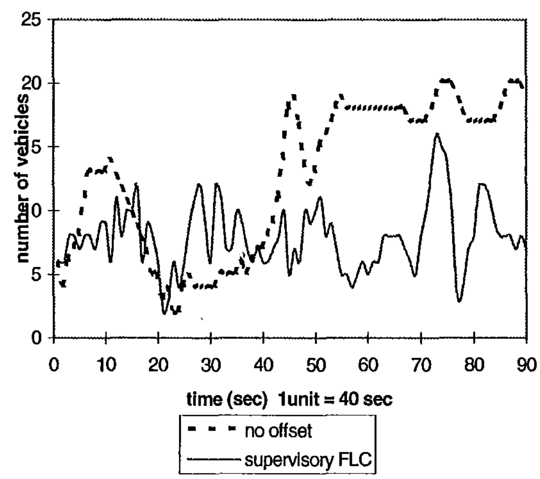


Figure 5. 17 Queue length at the north approach of intersection C

The above two figures, Figures 5.16 and 5.17, show the number of vehicles waiting at the north approach of intersection C. The offset adjustment by the local fuzzy logic controller does not result in a reduction in the number of vehicles waiting. This is because of the low vehicle volume at the north approach of intersection *B* as can be seen from Figure 5.14. The supervisory fuzzy logic controller reduces the traffic volume at the north approach of intersection *C* by 37% due to the high vehicular traffic at the north approach of intersection *A*.

The Figures 5.18 and 5.19 show the vehicular volume at the south approach of intersection *A*. In Figure 5.18, the local fuzzy logic controller at intersection *A* which coordinates this intersection with intersection *B*, causes a 22% increase in the number of vehicles waiting. The traffic volume at the south approach of *B* which affects the offset adjustment at intersection *A* is low, thus resulting in a reduction in the green phase, thereby increasing the queue length. When a supervisory fuzzy logic controller is used to adjust the offset, as shown in Figure 5.19, the average queue length is not affected.

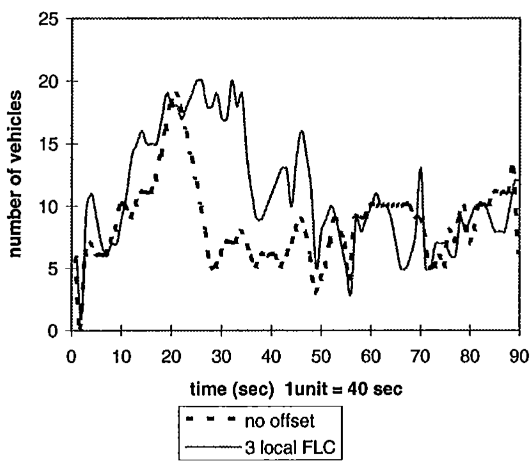


Figure 5. 18 Queue length at the south approach of intersection *A*

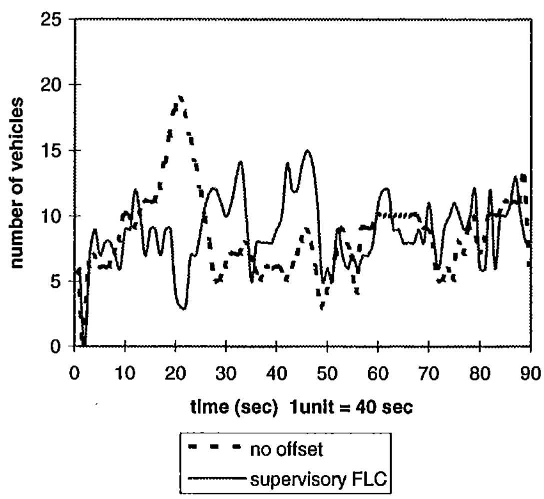


Figure 5. 19 Queue length at the south approach of intersection *A*

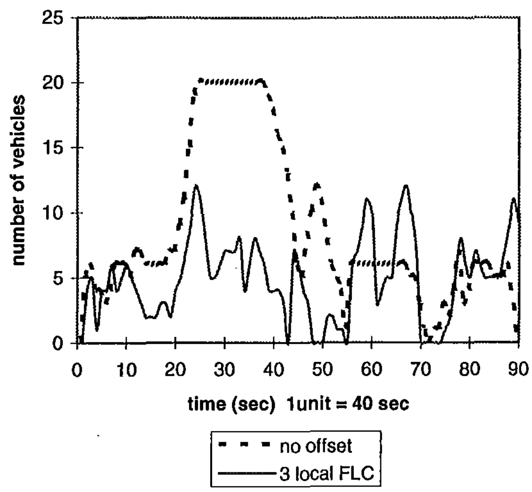


Figure 5. 20 Queue length at the south approach of intersection *B*

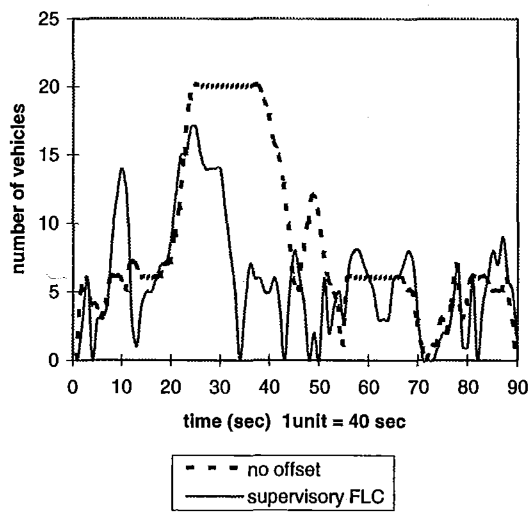


Figure 5. 21 Queue length at the south approach of intersection *B*

The Figure 5.20 shows a reduction of 42% in the queue length at the south approach of intersection *B* when the offset is adjusted by the local fuzzy logic controller. Figure 5.21 shows only a 31% reduction in the queue length for the supervisory fuzzy logic controller. This may be a consequence of restricting the number of fuzzy sets of the input parameters of the supervisory controller to three fuzzy sets.

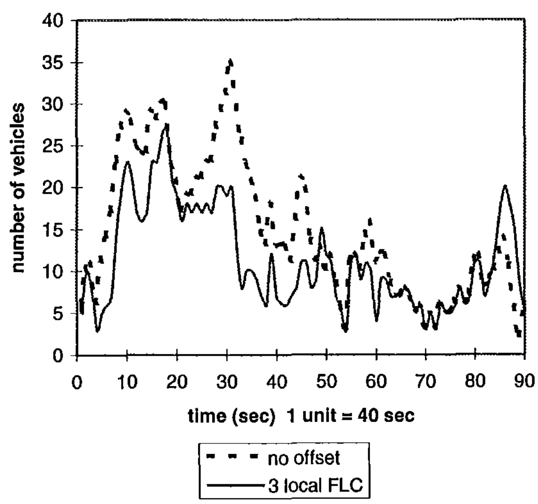


Figure 5. 22 Queue length at the south approach of intersection *C*

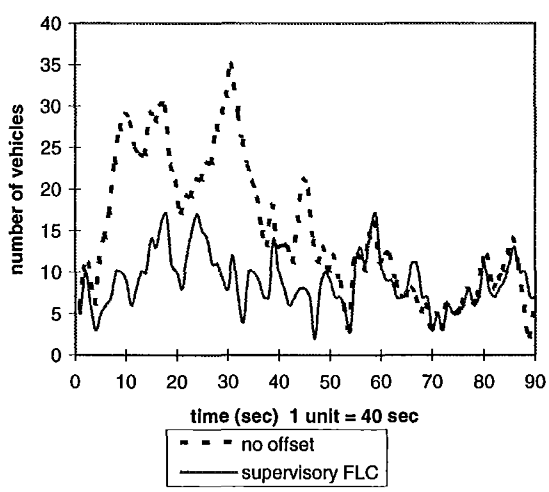


Figure 5. 23 Queue length at the south approach of intersection *C*

Figure 5.23 shows a significant reduction of about 40% in the number of vehicles waiting at the south approach of intersection *C* when using a supervisory fuzzy logic controller while the local fuzzy logic controller brings about only a 23% reduction in the queue length (Figure 5.22). Due to the high vehicle arrival rate at the north approach of intersection *A* and the low vehicle arrival rate at the east and west approaches of intersection *B*, the volume difference is high thus giving large extensions to the green phase of the north-south approach of intersection *C*. When a local fuzzy logic controller is used to adjust the offset, there is not much reduction in the queue length as the the traffic volume at intersection *B* is not as high as that of the north approach of intersection *A*.

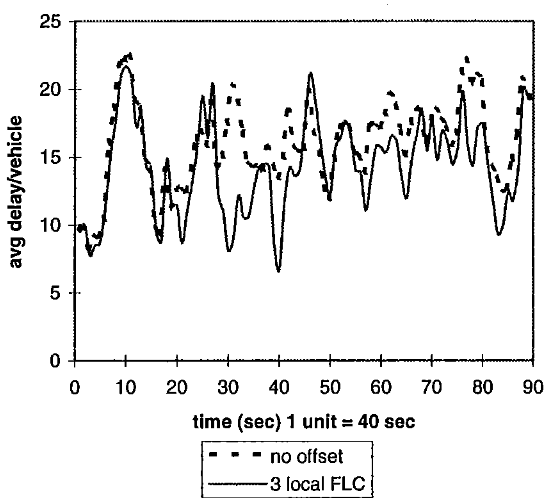


Figure 5.24 Avg delay of vehicles at the north approach of intersection *A*

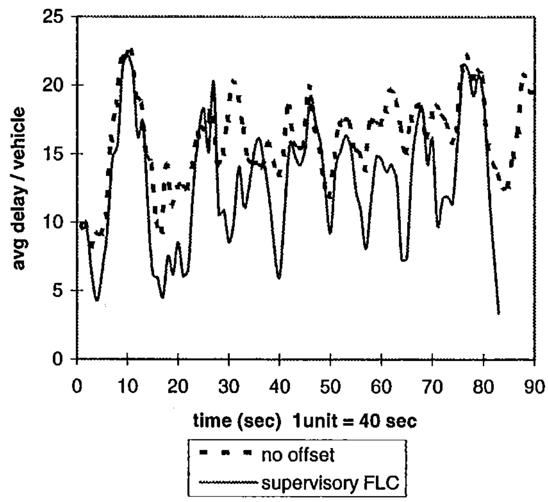


Figure 5.25 Avg delay of vehicles at north approach of intersection *A*

The average time spent by a vehicle in waiting at the north approach of intersection *A* is reduced when the offset is adjusted by either a local fuzzy logic controller or a supervisory fuzzy logic controller as illustrated in the Figures 5.24 and 5.25.

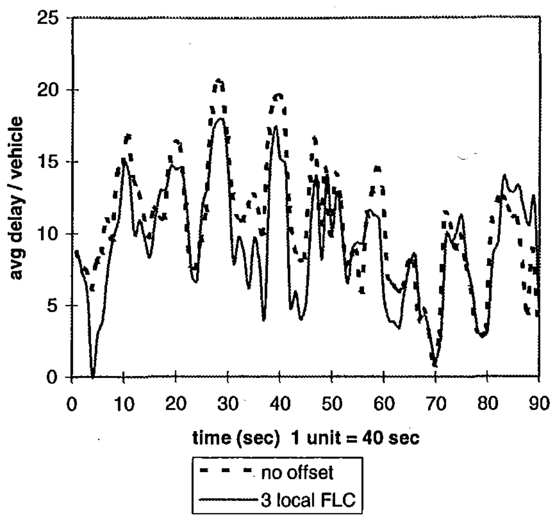


Figure 5.26 Avg delay of vehicles at the south approach of intersection *C*

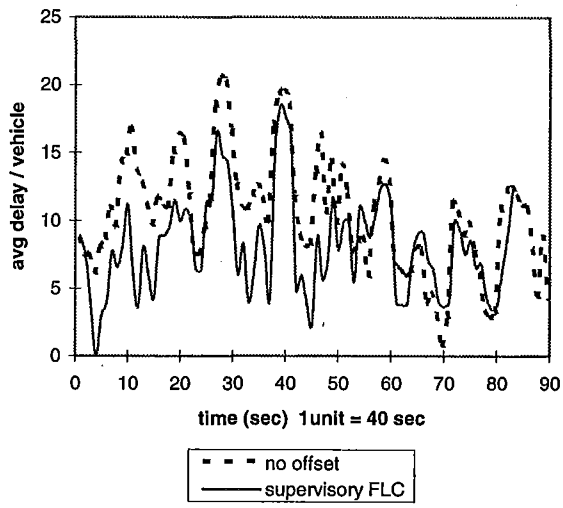


Figure 5.27 Avg delay of vehicles at south approach of intersection *C*

Similarly, the average delay per vehicle at the south approach of intersection *C* is also reduced as a consequence of the offset adjustment at this intersection, see Figures 5.26 and 5.27. An offset adjustment at the north approach of intersection *A* and at the south approach of intersection *C* results in increased green phase durations at these two approaches resulting in more number of vehicles flowing through the junction without stopping, thereby reducing the time spent in waiting.

<u>*The delay statistics are given below:*</u>

<u>Average delay per vehicle at the north-south approach of intersection *A*</u>

No offset adjustment between the intersections	- 17.5 sec
Local FLC used for offset adjustment at each intersection	- 16.6 sec
Supervisory FLC for adjusting offset at all three intersections	- 15.94 sec

<u>Average delay/vehicle at the north-south approach of intersection *B*</u>

No offset adjustment between the intersections - 18.6 sec

Local FLC used for offset adjustment at each intersection - 17.8sec

Supervisory FLC for adjusting offset at all three intersections - 17.7 sec

<u>Average delay/vehicle at the north-south approach of intersection *C*</u>

No offset adjustment between the intersections - 13.3 sec

Local FLC used for offset adjustment at each intersection - 12.7 sec

Supervisory FLC for adjusting offset at all three intersections - 12.1 sec

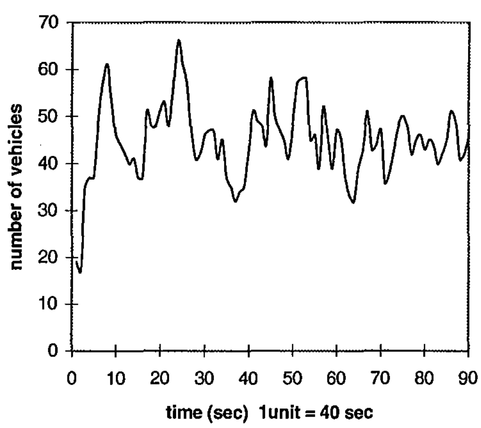


Figure 5.28 Queue length at all four approaches of intersection *A* - using three local FLC

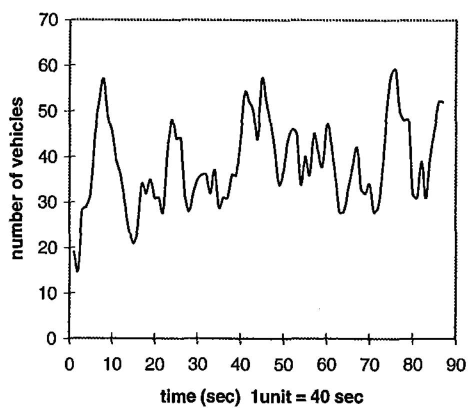


Figure 5.29 Queue length at all four approaches of intersection *A*- using supervisory FLC

Figures 5.28 and 5.29 show the number of vehicles waiting at all four approaches of intersection *A* when offset is adjusted with a local fuzzy logic controller and when offset is adjusted using a supervisory fuzzy logic controller respectively. The average queue length at the end of each cycle is equal to <u>38</u> vehicles when the offset is adjusted using a supervisory controller and 44 vehicles when the offset is adjusted by a local fuzzy logic controller.

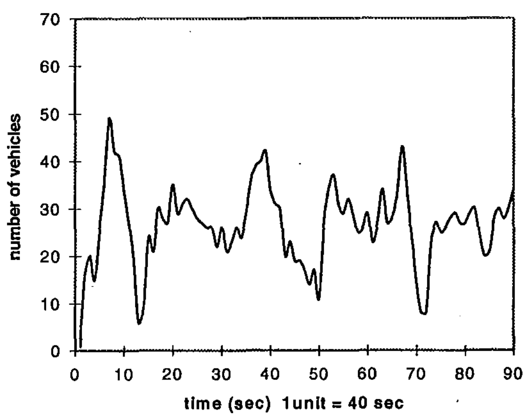


Figure 5.30 Queue length at all four approaches of intersection *B* - using three local FLC

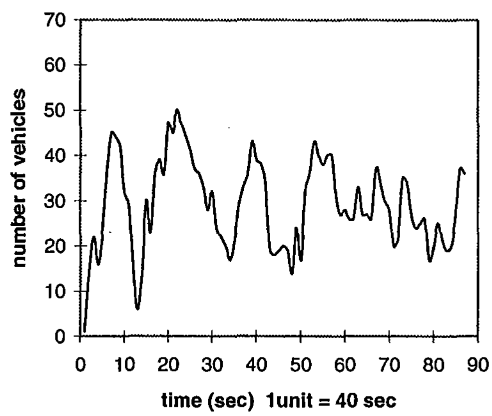


Figure 5.31 Queue length at all four approaches of intersection *B* - using supervisory FLC

Figures 5.30 and 5.31 show a significant reduction in the number of vehicles waiting at all four approaches of intersection *B* when there is an offset adjustment at the north-south approach of this intersection. When the offset is adjusted by a local fuzzy logic controller, the average queue length is 26 at the end of a cycle, while the queue length when a supervisory controller is used is 29.

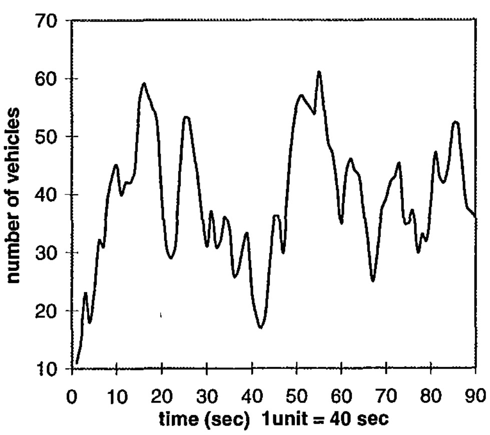


Figure 5.32 Queue length at all four approaches of intersection *C* - using three local FLC

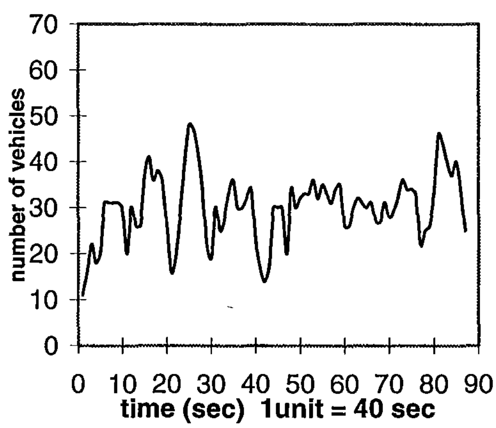


Figure 5.33 Queue length at all four approaches of intersection *C* - using supervisory FLC

Figures 5.32 and 5.33 show the traffic volume at all four approaches of intersection *C*. There is a considerable reduction in the traffic volume waiting at intersection *C* when the offset is adjusted by a supervisory fuzzy logic controller (Figure 5.33). The average number of vehicles at the end of a cycle is 30 in this case. When a local fuzzy logic controller is used, the average queue length is 38.

The simulation results establish the effectiveness of the supervisory fuzzy logic controller in improving the traffic flow through the three intersections. The queue lengths at the north and south approaches of the three intersections are reduced to a greater extent by the supervisory fuzzy logic controller than by the local fuzzy logic controllers.

5.6 **Discussion**

The traffic flow approaching a set of three intersections situated in the north-south direction was studied. The three intersections are coordinated by adjusting their respective offsets. The simulation results show that by using a local fuzzy logic controller to coordinate each intersection with only its upstream intersection, the queue length at the local intersection is reduced when the traffic volume at the upstream intersection is high. Otherwise, the green phase duration of the local traffic signal is not long enough to allow the traffic coming from the upstream intersection to pass through unstopped, thereby not reducing the queue length at the local intersection.

When a supervisory fuzzy logic controller is used to adjust the offset of the three traffic signals, the queue length at the north-south approaches of all three intersections is reduced. This is due to the adjustment of offset of each traffic signal based on the traffic volume at all the intersections rather than that of just the upstream intersection.

Adjusting the offset of a traffic signal using a supervisory fuzzy logic controller has a more pronounced effect on the traffic flow passing through the three traffic junctions compared to the effect of three local fuzzy logic controllers. It is possible to achieve a better throughput at all the intersections rather than just maximising the traffic flow at a single intersection. This is because the fuzzy control scheme involving the supervisory controller is a more coupled architecture, where each signal is coordinated with its neighbouring signals thus making it a cohesive network.

However, there is a limitation with using the single supervisory fuzzy logic controller. When a large number of traffic signals is to be coordinated, a supervisory fuzzy logic controller is not feasible because of the large number of input and output variables thus making it computationally expensive and not very cost effective. In such a case, where a large number of signalised intersections is considered, coordinating each intersection with its upstream intersection might be a better proposition.

In developing the supervisory fuzzy logic controller, each of the three input variables is divided into three fuzzy sets to facilitate the construction of the rulebase. This resulted in a fuzzy rulebase comprising twenty seven rules. By restricting the number of fuzzy sets to only three, the fuzzy logic controller is made less sensitive to abrupt changes in the queue length of vehicles.

In chapter 6, the traffic flow approaching a set of three intersections is regulated by adjusting the offset at each intersection using a supervisory fuzzy logic controller whose input variables are divided into five fuzzy sets each and whose fuzzy rules are developed using a Genetic Algorithm. The increase in the number of fuzzy sets should improve the sensitivity of the supervisory fuzzy logic controller resulting in a better performance.

Chapter 6

Genetic Algorithms for fuzzy rule generation

6.1 Introduction

Fuzzy Logic Control (FLC) is a control methodology based on the theory of fuzzy sets. It attempts to model the human rule of thumb approach to problem solving. Fuzzy logic controllers are rule based systems whose inference mechanism is accomplished by fuzzy rules. These rules are normally obtained by interviewing human operators, extracting knowledge from experts in the area, and trial and error. But this could be a lengthy process and does not always provide reliable information for complex and ill-defined processes.

To overcome the problem of rule-elicitation, Procyk and Mamdani (1979) proposed the Self Organising Fuzzy Logic Controller (SOFLC) to develop and improve the fuzzy rules of a system and automatically structure itself by monitoring the performance of the process so as to obtain a predetermined quality. SOFLC, being a supervised learning technique, needs a valid model of the system that is to be controlled. If the desired output of the system is not known, it is not possible to obtain a satisfactory knowledge base for the system.

Another technique that can be employed to obtain an adequate rulebase is by using Genetic Algorithms (GAs) (Wen-ruey H., 1993, Mohammadian M., et al, 1994a). There is no need to know the desired output of the system that is to be controlled. With the aid of GAs, optimal fuzzy rules could be obtained without human operators' experience or control engineers' knowledge.

In this chapter, Genetic Algorithms (GAs) are employed to learn the fuzzy rules to adjust the offset of a set of traffic signals. The approach presented here generates the fuzzy rules to regulate the traffic flow approaching two adjacent intersections and a set of three intersections. The models considered here are the same as the ones described in chapters 4 and 5 respectively.

6.2 Genetic Algorithms

Genetic Algorithms are search algorithms based on the mechanics of natural selection and evolution (Goldberg, D.E., 1989). There are three basic genetic operations, associated with a simple GA which are performed on a population of possible solutions. They are reproduction, crossover and mutation.

In reproduction, a population of possible solutions (strings) is usually represented as a string of binary numbers. Each individual string is decoded and applied to the problem and the performance of the string is assessed by assigning it a fitness value. Thus, each string is assigned a fitness value depending on how well it has performed its task.

The next step involves recombination where the fittest strings are chosen to form the next generation. This process involves the crossover operation where a point is randomly chosen in the two strings and the segments of the two strings are switched to the right of this point. Crossover occurs with a certain probability called the crossover probability.

Mutation is performed to maintain genetic diversity within a small population of strings and avoid premature convergence to a non-optimal solution. The mutation operator performs a random change of the value of a string position according to a small probability called the mutation probability.

An additional feature commonly included in GAs is the automatic inclusion of the best performing string of the parent generation in the new offspring generation. This procedure prevents a good string being lost due to the probabilistic nature of reproduction.

6.3 Fuzzy Rule generation using GAs

The design of the knowledge base of a fuzzy logic controller is based upon a human operator's knowledge and experience. The fuzzy rules are formulated by a trial and error method which is not only time consuming but also does not guarantee optimal fuzzy rules for the system. Incorporating Genetic Algorithms into the design of a fuzzy logic controller ensures automatic generation of fuzzy rules for controlling a system. In this chapter, a fuzzy-GA rule generator architecture is used to automatically generate the fuzzy rules for the supervisory fuzzy logic controller. A block diagram of the fuzzy-GA rule generation architecture is shown in Figure 6.1.

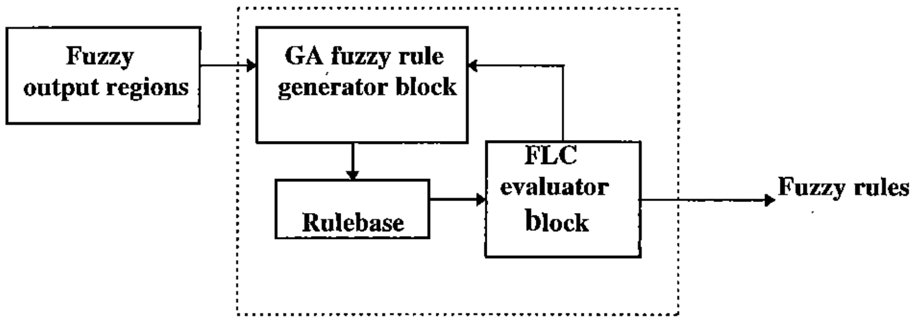


Figure 6.1 Fuzzy-GA rule generator architecture (Mohammadian M., 1994)

The structure of a fuzzy-GA rule generator architecture is similar to that of a traditional fuzzy logic controller except that in a traditional fuzzy logic controller, the fuzzy rules are determined by expert's knowledge while in a fuzzy-GA rule generator system, the fuzzy rules are obtained by doing a random yet iterative search in the output fuzzy regions based on a system specific performance criterion to search for the best fuzzy rule base for the fuzzy logic controller. The fuzzy rules are obtained by evaluating the performance of the system for each set of rules that is generated by the GAs, until some system specific performance criterion is met.

Fuzzy-GA architecture

Let us consider a fuzzy logic controller with two inputs (x and y) and a single output (z). As a first step to generating the fuzzy rules, the domain intervals of the input and output variables are divided into different regions, called fuzzy sets. The number of fuzzy sets is application dependent. It is assumed that x, y and z are all divided into five fuzzy regions each, with x and y denoted by the linguistic terms **VL, LO, MD, HI, VH** and z denoted by the linguistic terms **VS, SM, MD, HI, VH**. A fuzzy membership function is assigned to each fuzzy set. Since x and y are divided into five fuzzy sets each, a maximum of twenty five fuzzy rules can be written for the fuzzy logic system.

y

		VL	LO	MD	HI	VH
	VL					
	LO					
x	**MD**					
	HI					
	VH					

Table 6.1 An empty fuzzy rule matrix

The fuzzy rule base can be constructed as 5 x 5 rule matrix as shown in Table 6.1, with cells to hold the corresponding action that must be taken for every possible combination of the input variables x and y.

The consequent for each fuzzy rule, the value of each cell in the Table 6.1, is determined by genetic evolution. In order to do so, the input and the output fuzzy sets need to be encoded. However, it is not necessary to encode the input fuzzy sets because the input fuzzy sets are static and do not change. Only the output fuzzy sets are encoded. The fuzzy rules relating the input variables (x and y) to the output variable (z) have twenty five possible combinations. The consequent of each fuzzy rule can be any one of the five output fuzzy sets. The output fuzzy sets are encoded by assigning 1 = **VS** (very small), 2 = **SM** (small), 3 = **MD** (medium), 4 = **HI** (high), and 5 = **VH** (very high). GAs randomly encode each output fuzzy set into a number ranging from 1 to 5 for all possible combinations of the input fuzzy variables. Each encoded parameter can be considered to be a gene and the string formed by the concatenation of all the encoded parameters is called a genotype. Each genotype is an individual string which is a member of a population which in this case, is a set of fuzzy rules for the fuzzy logic controller.

An individual string can be represented in the following way:

4	3	5	3	---------- ----------------	1
1	2	3	4		25

which translates to:

if x = **VL** and y = **VL** *then* z = **HI**

if x = **VL** and y = **LO** *then* z = **MD**

if x = **VL** and y = **MD** *then* z = **VH**

if x = **VL** and y = **HI** *then* z = **MD**

-

-

if x = **VH** and y = **VH** *then* z = **VS**

Each string is a member of a population and a population of size *n* has *n* number of individuals strings randomly encoded by GA. A population comprising of *n* individual strings can be represented as:

	1	2	3	---------- ------------------------	25
1	4	3	5		5
2	1	5	3		3
3	4	4	2		1
-					
-					
-					
-					
n	3	1	4		3

Each individual string is then decoded into the ouput linguistic terms. The set of fuzzy rules thus developed is evaluated by the fuzzy logic controller, based upon a fitness value which is specific to the system. The fitness value is application dependent. At the end of each generation, two copies of the best performing string from the parent generation is included in the next generation to ensure that the best performing strings are not lost. GA then performs the process of selection, crossover and mutation on the rest of the individual strings. Selection and crossover are the same as a simple genetic algorithm while the mutation operation is modified.

Crossover and mutation take place based on the probability of crossover and mutation respectively. For crossover, two individual strings from the parent generation are selected at random and an integer position *k* is selected randomly between 1 and the string length less one, which is 24. Two new strings are created by swapping all integers between positions *k+1* and 25 inclusively. For example, consider the following two strings, strings 1 and 2 from the initial population:

	1	2	3	4	5 ---------	--------------------------	25
1	4	3	5	2	3		5
2	1	5	3	2	2		3

If k = 2, the new strings obtained as a result of crossover is given by:

	1	2	3	4	5 ----------	------------------	25
1	4	3	3	2	2		3
2	1	5	5	2	3		5

The process of mutation takes place depending on the mutation rate. An integer position in the individual string is selected at random and the encoded parameter at that position is replaced by a random number ranging from 1 to 5. The mutation operation can be illustrated in the following way:

Before mutation:

	1	2	3	4	5 ----------	---------------- 25
1	4	3	5	<u>2</u>	3	5

After mutation:

	1	2	3	4	5 ----------	---------------- 25
1	4	3	5	<u>4</u>	3	5

The process of selection, automatic inclusion, crossover and mutation are repeated for a number of generations until a satisfactory fuzzy rule base is obtained. We define a satisfactory rule base as one whose fitness value differs from the desired output of the system by a very small value.

Genetic Algorithms perform the task of generating high performance fuzzy rules quickly. They run automatically without any need for operational guidance other than the fitness value supplied by the fuzzy logic controller. GAs perform a self-directed search, learning new fuzzy rules for the fuzzy logic controller (Mohammadian M., et al., 1994a).

6.4 Control of two adjacent intersections using fuzzy rules generated by the Fuzzy-GA rule generator architecture

The traffic flow approaching an intersection is random and highly uneven. Regulating this fluctuating traffic flow requires common sense reasoning and knowledge pertaining to the pattern of traffic flow. If we assume a policeman directing traffic at an intersection, he lets traffic pass through in one direction and then stops traffic from that direction and lets the traffic pass from the other direction. He makes his decision based on the traffic density at each approach and also on the time spent by vehicles waiting to pass through. He makes his decisions using the rule of thumb and knowledge gained through years of experience.

However, if the traffic flow approaching a set of intersections is to be regulated, the policeman's knowledge and experience may not be sufficient. Fuzzy logic, which emulates the human way of thinking is an useful tool for dealing with problems which are ill-defined and uncertain as in the case of traffic regulation which is highly uneven and unpredictable.

In chapter 3, a fuzzy logic traffic controller to regulate the vehicular flow approaching an isolated intersection from all four directions (North, South, East and West), is presented. Based on the current traffic volume and the number of vehicles that passed through during the previous green phase, the current green phase of the north-south and east-west approaches of the traffic signal is adjusted. The simulation results showed the ability of the fuzzy logic controller in handling a wide range of varying traffic patterns.

In chapter 4, a supervisory fuzzy logic controller is developed to coordinate two adjacent intersections by adjusting the offset of the traffic signals in order to minimise the number of stops at each intersection. A set of twenty five fuzzy control rules is used to adjust the offset. These rules, developed manually based on common sense reasoning and trial and error, are effective in reducing the number of vehicles waiting at the two intersections. However, it is difficult to determine whether these fuzzy rules are the 'optimal' fuzzy rules for adjusting the offset.

In order to procure a better set of fuzzy decision rules, a genetic algorithm is employed to acquire the fuzzy rule base for adjusting the offset at the two intersections *A* and *B*, see Figure 4.1. The input variables are *Vol_diff 1* and *Vol_diff2* where

$$\text{Vol\_diff 1} = V_{SB} - (V_{EA} + V_{WA}) / 2 \qquad (6.1)$$

$$\text{Vol\_diff 2} = V_{NA} - (V_{EB} + V_{WB}) / 2 \qquad (6.2)$$

V_{SB} is the queue length at the south approach of intersection *B*, V_{NA} is the queue length at the north approach of intersection *A*, V_{EA} and V_{WA} are the queue lengths at the east and west approaches of intersection *A*, and V_{EB} and V_{WB} are the queue lengths at the east and west approaches of intersection *B*.

The output variables are the adjustments to the green phase of the two traffic signals, *Ext1* and *Ext2*. The linguistic fuzzy sets for the input and output variables and their corresponding membership functions are the same as used in chapter 4.

The fuzzy-GA rule generator architecture shown in Figure 6.1 is employed to acquire the fuzzy rule base for adjusting the offset. GA initialises a population of individual strings by encoding the output regions in a random manner. Since the supervisory fuzzy logic controller has two output variables, *Ext1* and *Ext2*, each individual string is constructed as a two dimensional array represented by:

	1	2	3	4	5 --------	------------------------	25
Ext1	4	3	5	2	2		3
Ext2	5	5	1	3	1		2

Each of these strings is then decoded into the output linguistic terms and for each combination of the input fuzzy sets, an output linguistic term is assigned. This fuzzy rule base is evaluated by the supervisory fuzzy logic controller based on a fitness function. The fitness function is the sum of all the vehicles waiting at the north and south approaches of the two intersections during the simulation. It is desired to generate a fuzzy rule base that minimises the fitness function. The GA operators - reproduction, crossover, and mutation are then applied to the individual strings of the population based on the fitness value. This process is repeated for a number of generations till a suitable fuzzy rule base is obtained. A suitable fuzzy rule base is one which minimises the queue length at the north and south approaches of both intersections.

In generating the fuzzy control rules for adjusting the offset of the two traffic signals, the following set of data is used.

Number of generations = 300

Population size = 30

Length of chromosome = maximum number of rules = 25

Crossover probability = 0.6

Mutation probability = 0.015

	Fuzzy rules generated by hand	Fuzzy rules generated by GA
Queue length at the north approach of intersection *A*	2323	802
Queue length at the south approach of intersection *A*	562	436
Queue length at the north approach of intersection *B*	441	543
Queue length at the south approach of intersection *B*	1529	778
Queue length at all four approaches of intersection *A*	4253	2595
Queue length at all four approaches of intersection *B*	2974	2375

Table 6.2 Comaprison of the fuzzy rules constructed by hand and rules generated by GA for the supervisory fuzzy logic controller

Table 6.2 shows the effectiveness of the fuzzy rules generated by Genetic Algorithms over the fuzzy rule base constructed by hand. When the fuzzy rules are determined by hand, the number of vehicles waiting at the south approach of intersection A and those at the north approach of intersection B is less when compared to the other two approaches. When the fuzzy rules are generated manually, there is a tendency to restrict the flow of vehicles passing through the north approach of intersection A and the south approach of intersection B in order to avoid the possibility of congestion between the two intersections. As a result, the number of vehicles waiting at the approaches between the intersections, that is, north approach of intersection B and south approach of intersection A, is kept to a mimimum. However, the queue length at the north approach of intersection A and that at the south approach of intersection B increases to a high value whenever there is a heavy burst of traffic.

On the other hand, Genetic Algorithms attempt to optimise the overall performance of the system. They minimise the number of vehicles waiting at all the approaches to intersections A and B rather than just reducing the queue length at the north approach of intersection B and the queue length at the south approach of interscetion A. When the fuzzy rules are generated by genetic algorithms, the queue length at the north approach of intersection A is reduced by 62%, the queue length at the south approach of intersection A is reduced by 22%, the queue length at the north approach of intersection B is increased by 18% and the queue length at the south approach of intersection B is reduced by 49%. The total number of vehicles waiting at intersection A is reduced by 39% and the total number of vehicles waiting at intersection B is reduced by 20%.

The fuzzy rule base obtained after 300 generations for a population size of 30 is shown in Table 6.3. The offset adjustments to the north-south approaches of intersection A and B, which in effect is the green phase extensions, are the output variables. Each entry in the table is made up of two components. The first is the green phase extension in the north-south approach of intersection A and the second is the green phase extension to the north-south approach of intersection B.

		Vol_diff2				
		VL	LO	MD	HI	VH
Vol_diff 1	VL	SM VS	VH VS	SM SM	VH HI	SM HI
	LO	VS VH	SM VH	SM VH	VS SM	VH HI
	MD	VS SM	VS VH	MD SM	VH HI	SM SM
	HI	MD HI	MD MD	HI MD	VS VH	MD MD
	VH	MD MD	SM HI	VH VH	VS VH	HI HI

Table 6.3 Fuzzy rule base supervisory FLC adjusting offset

Figures 6.2 and 6.3 show the queue length at the north and south approaches of intersection *A*. It can be seen that the traffic density is reduced as a result of the fuzzy rules generated by Genetic Algorithms.

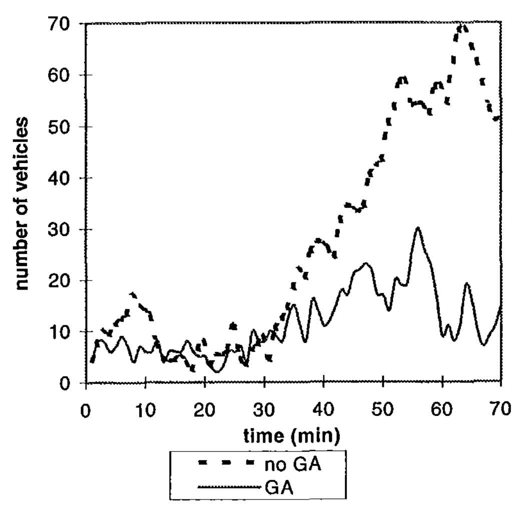


Figure 6.2 Queue length at the North approach of intersection *A*

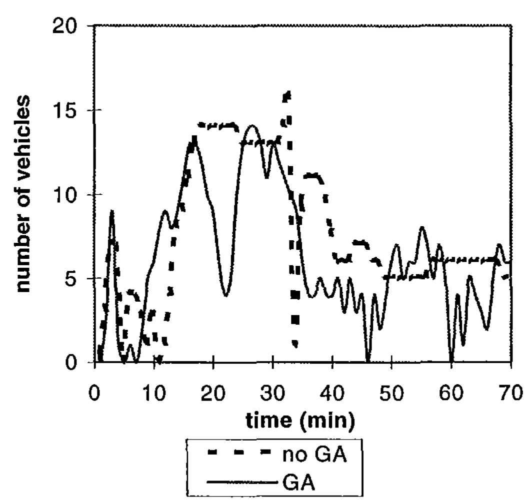


Figure 6.3 Queue length at the south approach of intersection *A*

In Figure 6.2, when the fuzzy rules are developed by hand, the queue of vehicles tends to build up and there is no indication of any change in the trend, as the vehicles keep accumulating. The fuzzy rules generated by Genetic Algorithms maintain the queue length, without letting it increase, throughout the simulation. In Figure 6.3, the fuzzy rules generated by genetic algorithms shows only a slight improvement in the queue length at the south approach of intersection *A* because the queue length is already minimised to a small value.

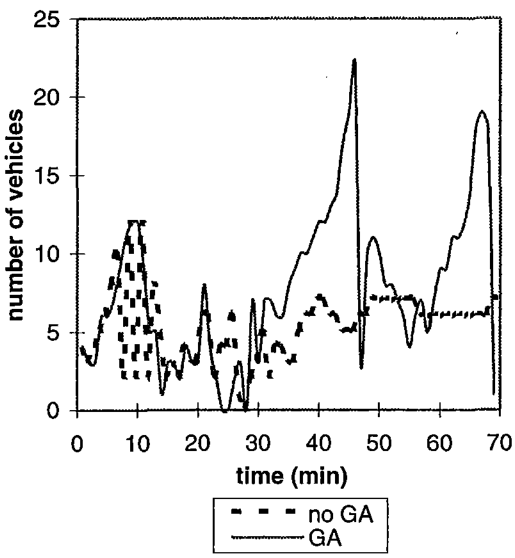


Figure 6.4 Queue length at the North approach intersection *B*

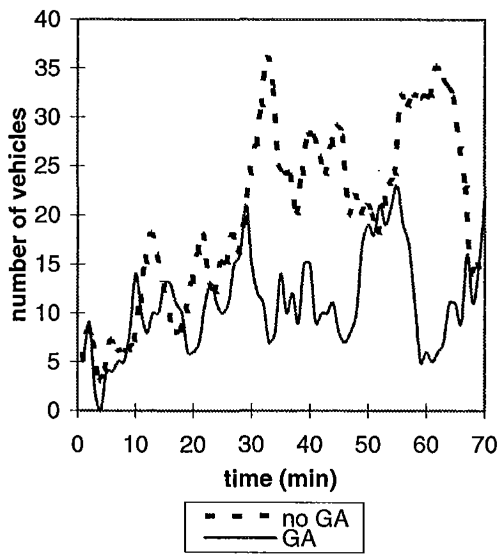


Figure 6.5 Queue length at the south approach intersection *B*

Figure 6.4 shows the number of vehicles waiting at the north approach of intersection *B*. The fuzzy rules generated by Genetic Algorithms is not successful in reducing the traffic volume at the north approach of intersection *B*. This is because, the number of vehicles waiting at the north approach of intersection *B* is already minimised to a great extent and it is not possible to further reduce the queue length without affecting the traffic flow at the other approaches of intersections A and B. The fuzzy rules generated by genetic algorithms tend to minimise the traffic volume at all the approaches, to the same extent. In contrast, the fuzzy rules generated by hand are only effective in reducing the queue lengths between the two intersections.

The queue length at the south approach of intersection B is reduced, as shown in Figure 6.5. The number of vehicles waiting does not increase beyond 24 even as the vehicle arrival rate increases. The fuzzy rules generated by genetic algorithms extends the green phase duration of the north-south approach of intersection B in such a way that the queue length is maintained below this level.

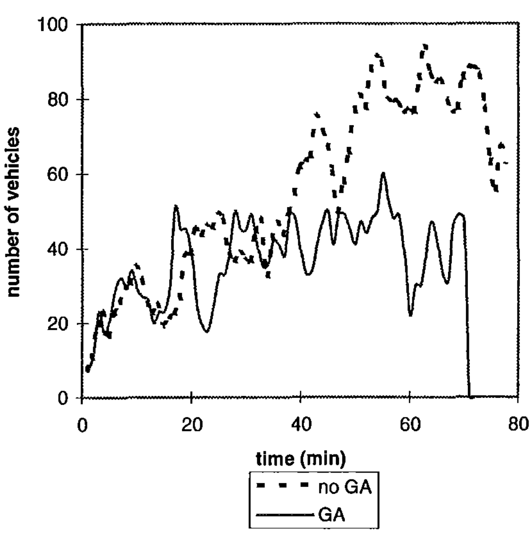


Figure 6.6 Queue length at all four approaches of intersection A

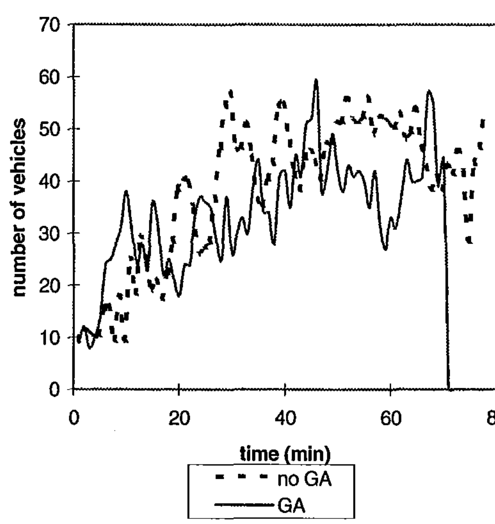


Figure 6.7 Queue length at all four approaches of intersection B

Figures 6.6 and 6.7 show the number of vehicles waiting at all four approaches of intersections A and B. These figures show a reduction in the traffic density at these intersections when the fuzzy rules generated by Genetic Algorithms are used. These rules are effective in reducing the number of vehicles waiting at intersection A to a greater extent than the number of vehicles waiting at intersection B. This is because the fuzzy rules generated by the genetic algorithms reduce the traffic volume at all the approaches to the same extent, while the fuzzy rules developed by hand are biased towards the north approach of intersection B because of the high vehicle arrival rate at the north approach of intersection A. Hence, the overall reduction in the traffic volume at intersection B is not significant.

6.5 Control of a set of three intersections using the fuzzy rules generated by the Fuzzy-GA rule generator architecture

The fuzzy-GA rule generator architecture shown in Figure 6.1 is now used to automatically generate the fuzzy rules for regulating the traffic flow approaching a set of three traffic signals. In chapter 5, a supervisory fuzzy logic controller is developed to coordinate a set of three intersections and minimise the number of vehicles at each intersection. A set of twenty seven fuzzy control rules is used to adjust the offset of all three traffic signals A, B, and C, see Figure 5.1. The supervisory fuzzy logic controller comprising three input variables and three output variables was employed for this purpose. Each of the three input variables was divided into three fuzzy regions, instead of five regions, in order to facilitate the formulation of the fuzzy rule base. Limiting the number of fuzzy input sets results in a smaller rule base. However, the sensitivity of the fuzzy logic controller tends to reduce because of the wider input fuzzy regions (Yan J., et al 1994),.

In this chapter, the offset of the three traffic signals is adjusted using a supervisory fuzzy logic controller whose input variables, *Vol_diff1*, *Vol_diff2*, and *Vol_diff3* are divided into five fuzzy regions to make the fuzzy logic controller more sensitive to abrupt changes in traffic flow. A control action is required for every possible condition that exists in the system. The number of possible combinations of the input fuzzy sets is 125. The deduction of fuzzy rules by hand, for a knowledge base of such high dimensionality is a difficult task. To facilitate the formulation of the fuzzy rule base, the rules are generated using Genetic Algorithms. The fitness function is the sum of the queue lengths at the north-south approaches of the three intersections.The fuzzy membership functions for the three inputs are shown in Figure 6.8. The output fuzzy sets and their corresponding membership functions are the same as those used in chapter 5, see Figure 5.6.

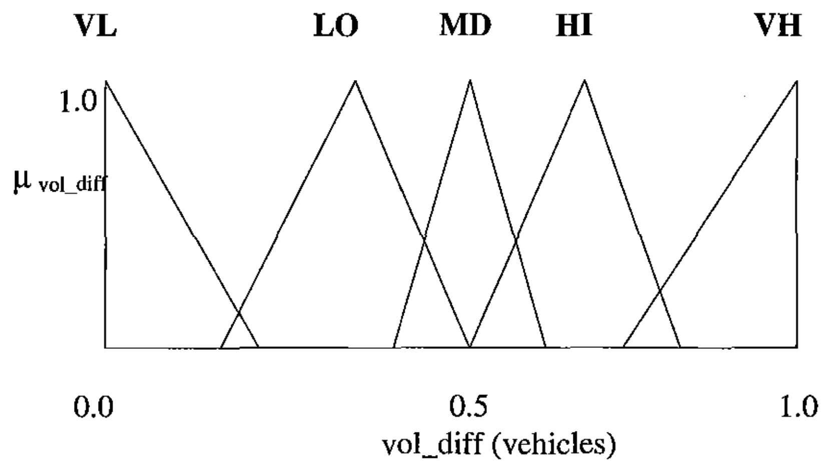


Figure 6.8 Membership function for the input fuzzy variables, vol_diff1,vol_diff2,vol_diff3

We start with the following initial set of data in order to generate the fuzzy rules for adjusting the offset.

Population size = 30

Length of chromosome = maximum number of rules = 125

Crossover probability = 0.6

Mutation probability = 0.001

The genetic algorithm was run for 100 and 200 generations and the results obtained is shown in Table 6.4.

	Fuzzy rules generated by hand	**Fuzzy rules after 100 generations**	**Fuzzy rules after 200 generations**
Queue length at the north approach of intersection *A*	**1167**	**1231**	**1052**
Queue length at the south approach of intersection *A*	**839**	**775**	**959**
Queue length at the north approach of intersection *B*	**685**	**968**	**239**
Queue length at the south approach of intersection *B*	**547**	**743**	**240**
Queue length at the north approach of intersection *C*	**718**	**280**	**628**
Queue length at the south approach of intersection *C*	**789**	**691**	**684**
Queue length at the north approach of intersection *A*	**3509**	**3512**	**3533**
Queue length at the north approach of intersection *B*	**2616**	**2802**	**1859**
Queue length at the north approach of intersection *C*	**2683**	**2117**	**2477**

Table 6.4 Comparison of fuzzy rules generated by hand and fuzzy rules generated using GAs for offset adjustment of three traffic signals

The fuzzy rules generated by the genetic algorithms after 200 generations reduce the number of vehicles waiting at the north-south approaches of all three intersections except the south approach of intersection *A*. This fuzzy rule base can be considered to be a satisfactory rule base for offset adjustment at the three intersections because of the extent to which it has reduced the queue lengths at the north and south approaches of the three intersections. The fuzzy rules are given in Appendix B. The fuzzy rules obtained after 100 generations are not very good as can be seen from the statistics for 100 generations in Table 6.4. This is could be due to a poor selection of individual strings in the initial population and not enough generations for the genetic algorithm to evolve a good fuzzy rule base.

Figures 6.9 - 6.14 show the number of vehicles waiting at the north-south approaches of intersections *A*, *B* and *C*. The fuzzy rules generated by Genetic Algorithms reduces the queue length at the north approach of intersection *A* by 9%, the queue length at the north approach of intersection *B* by 65%, the queue length at the south approach of intersection *B* by 56%, the queue length at the north approach of intersection *C* by 12% and the queue

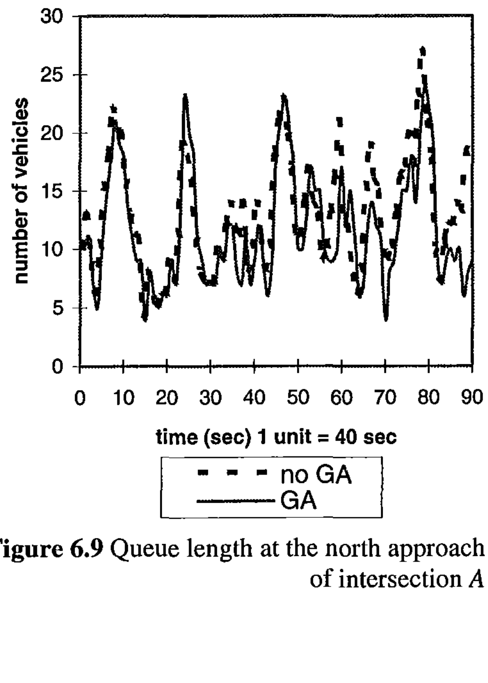


Figure 6.9 Queue length at the north approach of intersection *A*

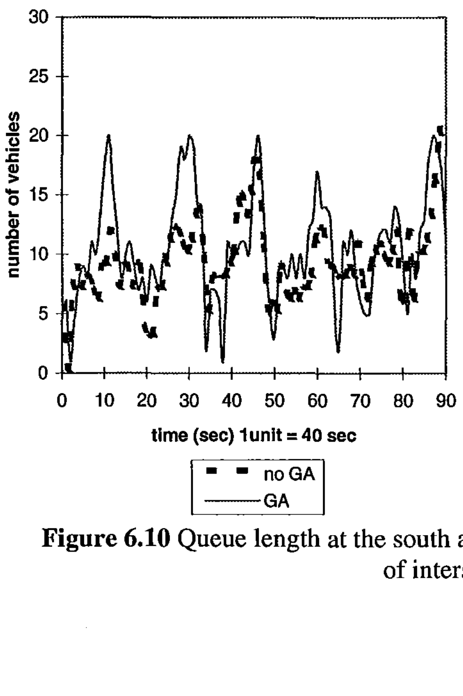


Figure 6.10 Queue length at the south approach of intersection *A*

length at the south approach of intersection *C* by 13%. However, the queue length at the south approach of intersection *A* in increased by 12%. But the overall performance of the supervisory fuzzy logic controller is improved as a result of the fuzzy rules generated by Genetic Algorithms.

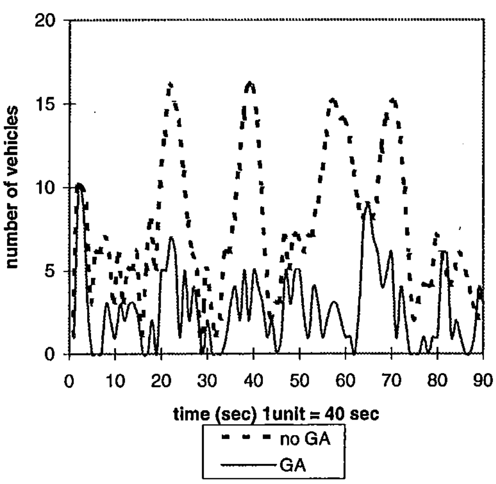


Figure 6.11 Queue length at the north approach of intersection *B*

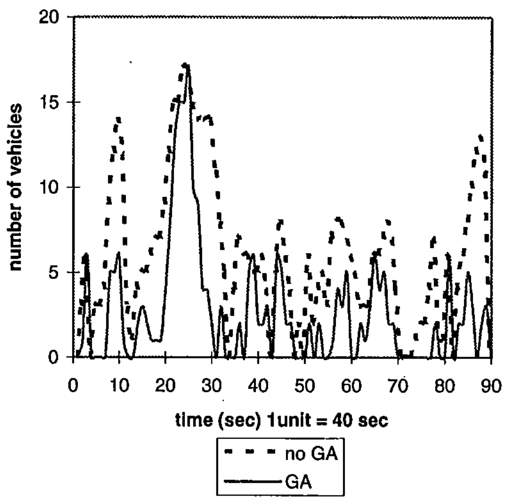


Figure 6.12 Queue length at the south approach of intersection *B*

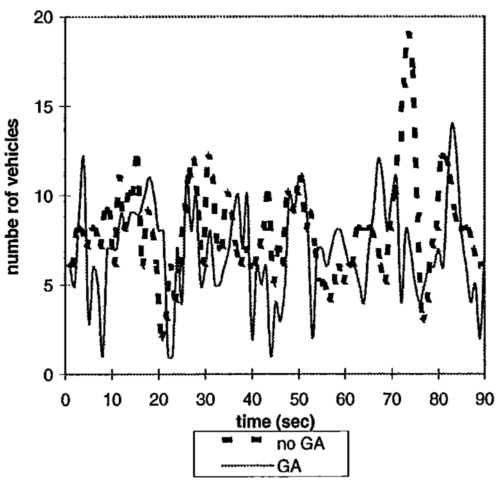


Figure 6.13 Queue length at the north approach of intersection *C*

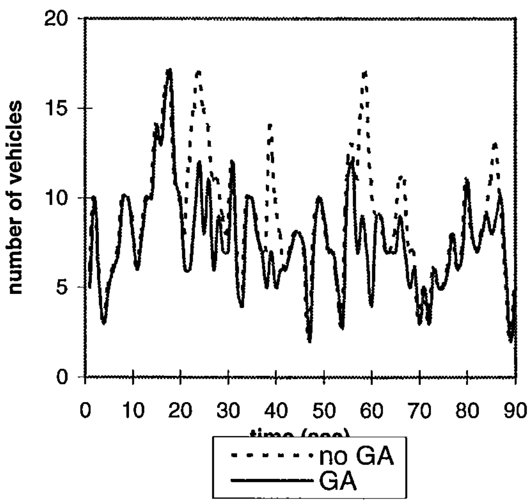


Figure 6.14 Queue length at the south approach of intersection *C*

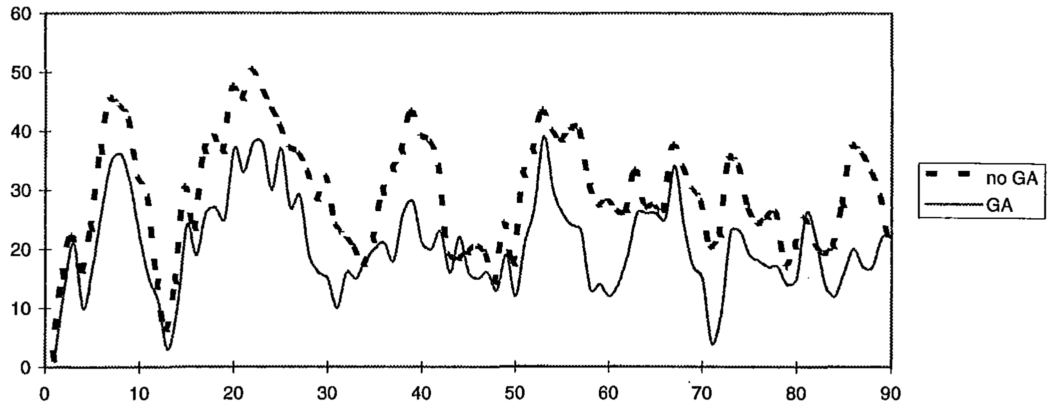


Figure 6.15 Queue length at all four approaches of intersection *B*

Figure 6.15 shows the number of vehicles from all approaches waiting at intersection *B*. There is a reduction of 30% in the total number of vehicles waiting at intersection *B* when the offset at intersection *B* is adjusted by the fuzzy rules generated by the genetic algorithm. From the simulation results, it can be inferred that the fuzzy rules generated by genetic algorithms is expected to reduce the queue lengths at all the intersections irrespective of vehicle arrival rates. Genetic algorithms are unaware of the internal workings of the system; they strive to evolve the best possible solution to the task using the fitness as a measure of the performance of individuals in the population. Hence, the overall queue length is reduced.

6.6 Analysis of Genetic Algorithms (GAs)

A genetic algorithm is a search procedure that uses random choice as a tool to guide a search through a space of candidate solutions (Goldberg D. E., 1989). It strives to improve the performance of a process or a system to an 'optimal' point. The mechanics of GAs are quite simple, involving the copying of strings and swapping of partial strings.

The population of individual strings may converge to an 'optimal' value after a series of generations, based on the fitness function. The fitness of individual strings improves as the number of generations increases. The best fitness for each generation is the encoded individual parameter string of a population with the least fitness value.

In this section, we investigate the effectiveness of Genetic Algorithms and how they can evolve a fuzzy rule base that can improve the performance of the system being controlled. The control of two adjacent traffic signals using a supervisory fuzzy logic controller, is considered. The initial set of population strings is decoded into fuzzy rules and evaluated by the supervisory fuzzy logic controller based on a fitness function which is the sum of all the vehicles waiting at the north-south approaches of the two intersections during the simulation. The genetic operators - selection, crossover and mutation are applied to the initial population string in order to minimise the fitness function.

The Genetic Algorithm was run for 100, 200 and 300 generations for a population size equal to 10 and the results obtained are shown in Table 6.5.

In Table 6.5, the number of vehicles waiting at the north approach of intersection *A* and the vehicles waiting at the south approach of intersection *B* are reduced when the Genetic

Algorithm is run for a greater number of generations. The queue length at the north approach of intersection B and the queue length at the south approach of intersection A are not reduced because the volume at these approaches is already low and it is not possible to reduce it further.

	100 gen	200 gen	300 gen
Queue length at north approach of intersection *A*	**832**	**780**	**766**
Queue length at south approach of intersection *A*	**348**	**401**	**384**
Queue length at north approach of intersection *B*	**495**	**561**	**564**
Queue length at south approach of intersection *B*	**1481**	**1256**	**1240**
Total number of vehicles waiting at intersection *A*	**2580**	**2693**	**2603**
Total number of vehicles waiting at intersection *B*	**2920**	**2864**	**2840**

Table 6.5 Simulation results using GAs for two adjacent intersections with population size = 10 and number of generations = 100, 200, 300

The number of vehicles waiting at all of the four approaches of intersections A and B is reduced as the Genetic Algorithm is run for many generations. Even though the queue lengths at the north approach of intersection B and that at the south approach of intersection A are not reduced, the overall performance of the system is improved with an increase in the number of generations.

The Genetic algorithm tries to converge to an 'optimal' solution by doing a random search in the output fuzzy regions. They do not have any information pertaining to the actual flow of traffic. GA aims to optimise the performance of the entire system as a whole. It tries to obtain the best possible solution by generating a fuzzy rule base which minimises the total queue length at all the intersections. As a result, it is quite possible that the fuzzy rules adjusting the offset at one intersection may not be as good as the rules controlling some other intersection.

The fitness value returned by the supervisory fuzzy logic controller is the sum of the queue lengths at the approaches of all the intersections. Hence, the queue length at one approach may be reduced by a great extent while the queue length at some other approach may not be reduced by the same amount. This is quite satisfactory because the purpose of using GAs is to optimise the traffic flowing through all three intersections and minimise the total number of vehicles waiting at the intersections.

Further analysis to establish the effectiveness of using genetic algorithms for constructing fuzzy control rules was done by running the simulation for different population sizes. The number of generations was set to 100 and the Genetic Algorithm was run for population sizes of 10, 20 and 30. The probablity of crossover and mutation were the same as used before.

Table 6.6 shows the effect of population sizes on the overall performance of the system. It can be seen from the table that an increase in the population size does not always enhance the performance of the system. For a population size of 20, the queue length at the south approach of intersection *B* is reduced significantly but there is no improvement at the other approaches. The overall performance of the system does not become better.

	popsize 10	popsize 20	popsize 30
Queue length at north approach of intersection *A*	**832**	**990**	**1088**
Queue length at south approach of intersection *A*	**348**	**512**	**357**
Queue length at north approach of intersection *B*	**495**	**537**	**452**
Queue length at south approach of intersection *B*	**1481**	**842**	**1191**
Total number of vehicles waiting at intersection *A*	**2580**	**2885**	**3124**
Total number of vehicles waiting at intersection *B*	**2920**	**2482**	**2706**

Table 6.6 Simulation results using GA for two adjacent intersections with different population sizes and number of generations = 100

When the genetic algorithm is run for a population size of 30, there is a slight improvement in the queue length at the south approach of intersection *A* and the north approach of intersection *B*, but there is a heavy build up of traffic at the other approaches. The reason could be due to the selection of poor strings in the initial set of population and not enough generations for GAs to evolve into an optimal solution. It can thus be inferred from this table that, increasing the population size does not evolve a better solution if the GA is run for fewer generations, which in this case is, 100.

In an attempt to acquire a better set of fuzzy rules, the number of generations is increased to 300 for population sizes 20 and 30. Table 6.7 shows the effect of population size on the performance of the system when the Genetic Algorithm is run for 300 generations.

	popsize 10	popsize 20	popsize 30
Queue length at north approach of intersection *A*	**766**	**861**	**802**
Queue length at south approach of intersection *A*	**384**	**474**	**436**
Queue length at north approach of intersection *B*	**564**	**327**	**543**
Queue length at south approach of intersection *B*	**1240**	**832**	**778**
Total number of vehicles waiting at intersection *A*	**2603**	**2825**	**2595**
Total number of vehicles waiting at intersection *B*	**2840**	**2382**	**2375**

Table 6.7 Simulation results using GA for two adjacent intersections with different population sizes and number of generations = 300.

Table 6.7 shows a reduction in the overall number of vehicles waiting at the north-south approaches of the two intersections when the population size is increased to 30 and when GA is run for 300 generations. For a population size of 20, an increase in the number of generations reduces the queue length at all the north-south approaches other than the north approach of intersection *A*. The overall number of vehicles waiting at the three intersections is reduced with an increase in the population size. The fuzzy rulebase obtained after 300 generations for a population size of 30 is the best fuzzy rule base generated by the genetic algorithm.

Population size	Number of generations	Best fitness
10	100	2978
10	200	2873
10	300	2848
20	100	2648
20	300	2488
30	100	2927
30	300	2477

Table 6.8 Best fitness values for different population sizes and generations

Table 6.8 shows the best fitness values for different population sizes and different generations. GAs converge rapidly when the size of the population is small. For a population size of 10, GAs do not have enough individual strings to perform the genetic operations in a viable manner. With an increase in population size, the convergence of the search space is increased as the GAs have more number of strings to process. This achieves a better solution at a slower rate of convergence.

Figures 6.16 to 6.18 show the rate of convergence for population sizes equal to 10, 20, and 30.

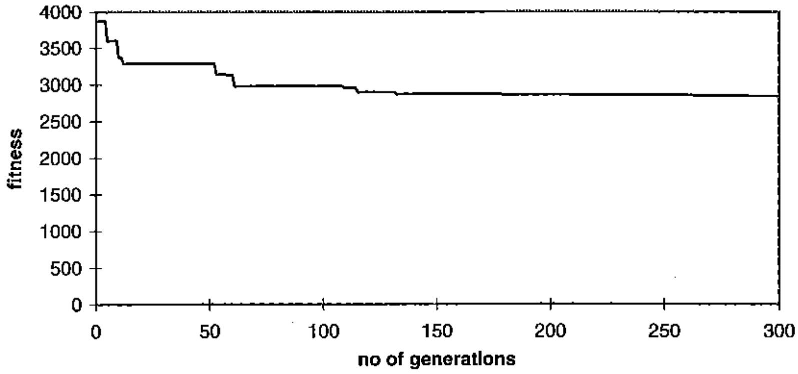


Figure 6.16 Best fitness for population size = 10

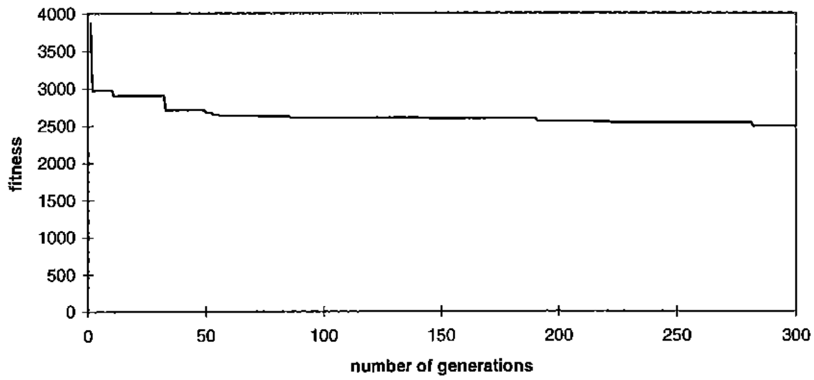


Figure 6.17 Best fitness for population size = 20

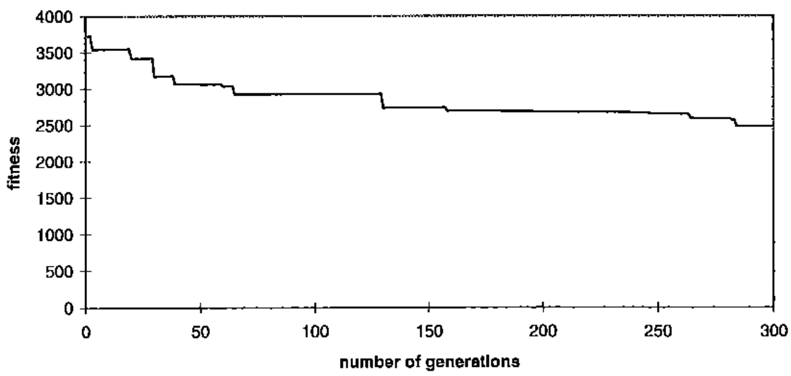


Figure 6.18 Best fitness for population size = 30

In Figure 6.16, there is no marked improvement in the fitness function after 130 generations. GAs converge to a value rather quickly because of the small population size. A new population of strings is produced, at the end of each generation, via the genetic operations which involve copying of strings, swapping portions of strings and generating random numbers. In a search space containing fewer number of individual strings, the operations of selection and crossover after a few generations may result in a population of strings with similar combinations. In other words, GAs converge prematurely to a local optimum, which might not be the global optimum.

For a population size of 20, the performance of GAs is distinct in the first 50 generations. (Figure 6.17). This could be due to the selection of very good individual strings in the initial set of population. But after 50 generations, the small size of the population restricts the search algorithm's quest for better solutions. In Figure 6.18, where the GA is run for a population size of 30, the rate of convergence is slow but it yields a better solution.

It can be inferred from the simulation results that an increase in the population size, and in the number of generations, results in the evolution of a better solution. Increasing the size of the population has a more pronounced effect than increasing the number of generations. The fitness value obtained for a population size of 20, after 100 generations, is better than the fitness value obtained after 300 generations for a population size of 10. Using a population size of larger magnitude leads to a better solution.

If the initial population consists of good individual strings, genetic algorithms achieve a better solution after a few generations. But, if the size of the population is small, all the candidate solutions in the population of strings may become similar after a few generations with no marked improvement in the fitness function.

6.7 Discussion

In this chapter, a Genetic Algorithm is used to generate the fuzzy control rules for adjusting the offset of a set of traffic signals. The fuzzy rule base for coordinating two adjacent intersections and a set of three intersections is developed using GA as a search algorithm.

From the simulation results, it was found that Genetic Algorithms are capable of generating fuzzy rules which effectively reduce the queue lengths at all the approaches of the three intersections. Since GAs consider many points from the search space simultaneously, it has a greater chance of converging to global optima.

However, the Fuzzy-GA architecture used to develop the knowledge base might not be a feasible option for complex networks involving more than three intersections. An increase in the number of intersections results in an increase in the number of input parameters to the Fuzzy Logic Controller thereby resulting in an exponential increase in the number of fuzzy rules. The use of GAs to generate the fuzzy knowledge base in such a case would be a computationally intensive task. Hence, a more feasible and an effective method needs to be applied to generate the fuzzy knowledge base for controlling a complex network of intersections. This issue is beyond the scope of this thesis and further work needs to be done in this area.

An analysis of GAs is also done to illustrate the effect of population size and number of generations on its performance. The simulation results showed the achievement of better solutions when there is an increase in size of the population and number of generations with population size having a more pronounced effect on the convergence to a better solution.

It can be concluded that Genetic Algorithms (GAs) are an effective tool for the generation of fuzzy rules for controlling a set of three traffic signals. The Fuzzy-GA rule generator architecture generates the fuzzy rules automatically. It is a quick, effectual and a cost efficient method of rule generation. The fuzzy decision rules obtained enhance the performance of the system and achieve better results than the fuzzy rules derived by hand. Incorporating Genetic Algorithms into fuzzy logic makes it possible to develop fuzzy logic controllers for controlling systems which are dependent on a high number of input parameters.

Chapter 7

Conclusions and future research

Traffic light control is used to resolve conflicts among vehicle movements at intersections. The existing traffic control techniques have been successful in regulating traffic in urban road networks. However, they cannot respond adequately to unpredictable changes in the traffic demand. These techniques strive to minimise the delay at a single intersection and do not optimise the traffic flow of the entire network.

In this thesis, a fuzzy control scheme for controlling a set of three urban traffic signals is presented. The effectiveness of fuzzy logic controllers in controlling the signals is established through simulations. The fuzzy logic controller makes adjustments to the green phase of the north-south and east-west approaches and the offset of the north-south approach of the traffic signal, based on actual traffic flow data. Sensors which measure the traffic densities in the lanes approaching the intersection provides the fuzzy logic controller with a good assessment of the varying traffic patterns.

In chapter 3, a fuzzy logic traffic controller for controlling an isolated traffic signal is studied. The controller makes adjustments to the green phase of an approach based on the ratio of the queue length at the approach to the number of vehicles that passed through the approach during the previous green phase. The fuzzy logic traffic controller is effective in reducing the queue length and the average waiting time per vehicle. Fluctuations in the traffic flow result in a corresponding change in the time duration of the green phase of the signal.

In chapter 4, two adjacent intersections are coordinated by adjusting the offset at each intersection. Each intersection is coordinated with only its upstream intersection by a local fuzzy logic controller placed at each intersection. A supervisory fuzzy logic controller is proposed to adjust the offset, based on the traffic volume, at both intersections. Simulation results show a reduction in the queue length and in the average delay per vehicle when the offset is adjusted. The performance of the supervisory fuzzy logic controller is similar to that of the local fuzzy logic controllers, since only two intersections were considered.

The traffic flow approaching a set of three intersections is studied in chapter 5. The superiority of the supervisory fuzzy logic controller over the local fuzzy logic controllers in adjusting the offset is established through simulations. The local fuzzy logic controllers achieve a reduction in the queue length only if the traffic volume at the upstream intersection is high. The supervisory fuzzy logic controller reduces the queue length at all the north-south approaches of the three intersections because each intersection is coordinated with all its neighbouring intersections rather than with just its upstream intersection as it is the case with a local fuzzy logic controller.

The formulation of fuzzy rules and selection of appropriate consequents for a set of input conditions can be a time consuming and an arduous task. Genetic Algorithms (GAs) have been gaining in popularity as a learning technique for the generation of fuzzy rule bases. In chapter 6, GAs are used for the automatic generation of the fuzzy rule base for a supervisory fuzzy logic controller. A supervisory fuzzy logic controller, comprising 25 fuzzy rules, is used for adjusting the offset at two adjacent intersections and a supervisory fuzzy logic controller consisting of 125 fuzzy rules is used to adjust the offset at a set of three intersections. Simulation results showed a marked improvement in the overall performance of the system when the offset is adjusted by the fuzzy rules generated by GAs. Using GAs as an unsupervised learning scheme for designing fuzzy rules not only simplifies the construction of the rule base but also provides an adequate control action for coordinating a set of two and three intersections.

The proposed fuzzy control scheme is not without its limitations. For the sake of simplicity, turning traffic was not considered in this research. The inclusion of turning traffic might result in a large number of input and output parameters, increasing the complexity of the fuzzy control scheme. Another assumption was the inclusion of only single lane traffic. However, the fuzzy control system does not lose its generality as it can be extended to include multi lane traffic without any changes to the control scheme.

There is much that can be done to further improve the present fuzzy logic control scheme. Some of the suggestions for future research are:

1. Control of a network of four intersections placed in the form of a square. In such a case, the intersections would have to be coordinated in both the north-south and east-west directions.

2. Inclusion of turning traffic and muti-lane traffic.

3. Using a hierarchical fuzzy logic controller to adjust the signal timing parameters instead of using a fuzzy logic traffic controller and a supervisory fuzzy logic controller operating independently of each other. A hierarchical fuzzy control scheme with the fuzzy logic traffic controller at the bottom level and the supervisory fuzzy logic controller at the top level is expected to enhance the performance of traffic control systems.

In this research, artificial intelligence techniques have been applied to the control of a set of urban traffic signals. The use of fuzzy logic controllers to adjust the signal timing parameters, green phase splits and offset, of the traffic signals improve the traffic flow across the set of intersections. They make the adjustments automatically in response to traffic situations. This fuzzy logic control scheme can detect occurrences of traffic congestion quickly and exactly by using sensors that determines the traffic density..

The integration of Fuzzy Logic and Genetic Algorithms is effective in the generation of fuzzy control rules which are superior compared to the fuzzy rules developed manually. These rules are effective in improving the performance of the fuzzy logic control system. The fuzzy logic control scheme proposed in this thesis can be effectively applied to on-line traffic control because of its ability to handle extensive traffic situations.

BIBLIOGRAPHY

Al-Khalili A.J., "Urban Traffic Control - A General Approach", IEEE Transactions on Systems, Man and Cybernetics, Vol SMC-15, NO 2, March 1985, pp 260-271.

Akcelik R., "Traffic Signals Capacity and Timing Analysis" Research report ARR 123, Australian Road Research Board, Nunawading, 1981.

Ambrosino G., Bielli M., Boero M., "Artificial Intelligence Approach to Road Traffic Control", Artificial Intelligence Applications to Traffic Engineering, VSP, Utrecht, The Netherlands, 1994, pp 95-116.

Barriere J., Farges J., Henry J., "Decentralisation vs Hierarchy in Optimal Traffic Control", IFAC/IFIP/IFORS Conf. on Control in Transportation Systems, Vienna, Austria, 1986.

Barwell F.T., "Automation and Control in Transport", Pergamon Press, 1973.

Bernard J.A., "Use of a Rule-Based System for Process Control", IEEE Control Systems Magazine, 1988, pp 3-13.

Bielli M., Ambrosino G., Boero M., Mastretta M., "Artificial Intelligence Techniques for Urban Traffic Control", Transportation Research, vol 25, no 5, sep 1991, pp 319-325.

Boverie S., Demaya B., Titli A., "Fuzzy Logic Control Compared with Other Automatic Control Approaches", Proceedings of the 30th Conference on Decision and Control, Dec 1991, pp 1212-1216.

Bruno G., Improta G., "Urban Traffic Control: Current Methodologies", Artificial Intelligence Applications to Traffic Engineering, VSP, Utrecht, The Netherlands, 1994, pp 69-94.

Caldwell C., Johnston V.S., "Tracking a Criminal Suspect through Face-Space with a Genetic Algorithm", Proc. of the Fourth International Conf. on Genetic Algorithms, 1991, pp 416-421.

Chang Chir-Ho, "Designing and Tuning a Fuzzy Logic Controller via Genetic Input/Output Mapping Factors", Phd Thesis, University of Oklahoma, 1993.

Chiu S., Chand S., "Self-Organising Traffic Control via Fuzzy Logic", Proceedings of the 32nd Conference on Decision and Control, Dec 1993, pp 1897-1902.

Chiu S., Chand S., "Adaptive Traffic Signal Control using Fuzzy Logic", Intelligent Vehicles '92 Symposium, Proceedings, IEEE, 1992, pp 98-107.

Chiu S., Chand S., “A Development Environment for Fuzzy Rule Based Traffic Control”, Robotics and Computer Integrated Manufacturing, vol 11, part 3, September 1994, pp 167-176.

Cox E., “Fuzzy Fundamentals”, IEEE Spectrum, October 1992, pp 58-61.

Cox E., "Adaptive Fuzzy Systems", IEEE Spectrum, February 1993, pp 27-31.

Decker K., “Distributed Problem Solving Techniques: a Survey”, IEEE Transactions in Systems, Man and Cybernetics, vol 17, no 4, 1987.

Evans, G.W., Karwowski, W., and Wilhelm, M.R., "Application of the Fuzzy Set Methodolgies in Industrial Engineering", Elsevier Netherlands, 1989.

Favilla, J., Machion, A., Gomide, F., and Gudwin, R., “ Adaptive Fuzzy Logic Based Urban Traffic Control”, Third Annual IAKE Symposium, November 16 - 19, Washington, 1992.

Favilla, J., Machion, A., and Gomide, F., “Fuzzy Traffic Lights: Adaptive Strategies”, 2nd IEEE International Conference on Fuzzy Systems, New York, 1993, pp 506 - 511.

Fitzpatrick J.M., Geefenstette J.J., Van Gucht D., "Image Registration by Genetic Search" Proc. IEEE Southeastern 1984, pp 460-464.

Fleming P.J., Fonseca C.M., "Genetic Algorithms in Control Systems Engineering", Communications of the ACM, March 1994, pp 383-390.

Fujitec F., "FLEX-8800 Series Elevator Group Control System", Fujitec Co. Ltd., Osaka, Japan, 1988.

Gartner N.H., Little J.D.C., Gabay H., "Optimisation of Traffic Signal Settings in a Network by Mixed-Integer Linear Programming", Technical Report no. 91, Operations Research Center, MIT, March 1974.

Gartner N.H., Little J.D.C., Gabay H., "Simulatneous Optimisation of Offsets, Splits. and Cycle Time", Tranportation Res. Rec., vol 596, 1976, pp 6-15.

Gartner N.H., Gershwin S.B., "Analytical Models of Transportation System Management", Proc. 1983 AIRO Conf., Guida Editori Napoli, 1983, pp 793-804.

Gegov A., "Multilevel Intelligent Fuzzy Control of Oversaturated Urban Traffic Networks", International Journal of Systems Science, June 1994, pp 967-978.

Goldberg D.E., "Genetic Algorithms in Search, Optimisation and Machine Learning", Addison Wesley, 1989.

Goldberg D.E., "Genetic and Evolutionary Algorithms Come of Age", Communications of the ACM, March 1994, pp 113-119.

Goldberg D.E., Lingle Jr R., "Alleles, Loci and the Travelling Salesman Problem", Proc. Int. Conf. on Genetic Algorithms and their Applications, 1985, pp 154-159.

Gupta M.M., Yamakawa T., " Fuzzy Computing: Theory, Hardware and Applications", Elsevier Science Publishers, North-Holland, 1988.

Haight F.A., "Mathematical Theories of Traffic Flow", Academic Press, 1963.

Hillier J.A., "The Area Traffic Control Experiment in Glasgow", Proc. 35th Annual Meeting of the Institute of Traffic Engineers, Boston , Massachusetts, 1965.

Holland J.H., "Adaptation in Natural and Artificial Systems", Ann Arbor, The University of Michigan Press, 1975.

Hoyer, R., and Jumar, U., " Fuzzy Control of Traffic Lights", 3rd IEEE International Conference on Fuzzy Systems, Florida, June 1994, pp 1526 - 1531.

Hunt P.B., Robertson D.I., Bretherton R.D., Winton R.I., "SCOOT - A Traffic Responsive Method of Coordinating Signals. TRRL Report LR1014, Transport and Road Research Laboratory, Crowthorne, 1981.

Jang, J.S.R., "ANFIS : Adaptive-Network-Based Fuzzy Inference System", University of California, Berkeley, Report, Department of Electrical Engineering and Computer Science, 1991.

Kandel A., "Fuzzy Techniques in Pattern Recognition", New York: Wiley Interscience, 1982.

Kandel, A.,"Fuzzy Expert Systems", CRC Press, Boca Raton, FL, 1992.

Karr C.L., "Design of an Adaptive Fuzzy Logic Controller using a Genetic Algorithm", Proc. of the Fifth International Conf. on Genetic Algorithms, 1991, pp 450-457.

Kelsey, R., Bisset, K., and Jamshidi, M., " A Simulation Environment for Fuzzy Control of Traffic Systems", 12th IFAC-World Congress, Sydney, Australia, July 1993, pp 553 - 556.

Kelsey, R., Bisset, K., " Simulation of Traffic Flow and Control using Fuzzy and Conventional Methods", 1993, pp 262-278.

Kickert W.J.M., Van Naute Lemke H.R., "Application of a Fuzzy Controller in a Warm Water Plant", Automatica, vol 12, no 4., 1976, pp 301-308.

King P.J., Mamdani E.H., "The Application of Fuzzy Control Systems to Industrial Processes", Automatica, vol 13, 1977, pp 235-242.

Kong S.G., Kosko B., "Adaptive Fuzzy System for Backing up a Truck and Trailer", IEEE Transactions on Neural Networks, vol 3, iss 2, March 1992, 211-223.

Kosko, B., "Neural Networks and Fuzzy Systems: A Dynamic System", Prentice-Hall, Englewood Cliff, 1992.

Koza J., "Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT press, Cambridge, Massachusetts, 1992.

Kumar K.K., Goldberg D.E., "Genetic Algorithms in Control systems Optimisation" AIAA Guidance, Navigation and Control Conference, August 20-22, 1990, Portland, OR.

Lee C.C., "Fuzzy Logic in Control Systems: Fuzzy Logic Controller", IEEE Transactions on Systems, Man and Cybernetics, Vol 20, No 2, March/April 1990, pp 404-418.

Li, Y.F., and Lan, C.C., "Development of Fuzzy Algorithms for Servo Systems", IEEE

Control System Magazine, vol 9, no 3, 1989, pp 65-73.

Linkens, D.A., and Abbod, M.F., "Self-Organising Fuzzy Logic Control and the Selection of its Scaling Factor", Transactions of the Institute of Measurement, 1992, vol 14, no 3, pp 114-125.

Luk J.Y.K., "Two Traffic Responsive Area Traffic Control Methods: SCAT and SCOOT", Traffic Engineering and Control, 1984, pp 14-20.

Mamdani E.H., Gaines B.R., "Fuzzy Reasoning and its Applications", London: Academic, 1981.

Mohammadian, M., and Stonier, R.J., "Generating Fuzzy Rules by Genetic Algorithms", 3rd IEEE International Workshop on Robot and Human Communication : RO-MAN'94, Nagoya, Japan, July, 1994a.

Mohammadian, M., and Stonier, R.J., "Tuning and Optimisation of Membership Functions of Fuzzy Logic Controllers by Genetic Algorithms", 3rd IEEE International Workshop on Robot and Human Communication : RO-MAN'94, Nagoya, Japan, July, 1994b.

Nakamiti G., Freitas R., Prado J., Gomide F., "Fuzzy Distributed Artificial Intelligence Systems", Technical Report RT-DCA 15/93, Department of Computer Engineering and Industrial Automation, University of Campinas-SP, Brazil, 1994, pp 462-467.

Nakatsuyama, M., Nagahashi, H., and Nishizura, N., " Fuzzy Logic Controller for a Traffic junction in a One Way Arterial Road", 9th IFAC-World Congress, Budapest, Hungary, 1984, pp 13 - 18.

Pappis, C.P., and Mamdani, E.H., " A Fuzzy Logic Controller for a Traffic Junction", IEEE Transactions Systems, Man, Cybernetics, 1977, SMC-7, pp 707 - 717.

Patriksson M., "The Traffic Assignment Problem : Models and Methods", VSP, Utrecht, The Netherlands, 1994.

Pedrycz, W., "Fuzzy Control and Fuzzy Systems", Wiley, New York, 1989.

Procyk T.J., Mamdani E.H., " A Linguistic Self-Organising Process Controller", Automatica, vol 15, 1979, pp 15-30.

Ragot J., Lamote M., "Fuzzy Logic Control", International Journal of Systems Science, vol 24 no 10, October 1993, 1825-1848.

Raju G.V.S., "Adaptive Hierarchical Fuzzy Controller", IEEE Transactions on Systems, Man and Cybernetics, vol 23, no 4, July/August 1993, pp 973-980.

Robertson D.I., "TRANSYT: A Traffic Network Study Tool", Ministry of Transport, RRL Report LR 253, Crowthorne, England, 1969.

Robertson D.I., "Traffic Models and Optimum Strategies of Control - A Review", Proceedings International Symposium on Traffic Control Systems, California, vol 1, 1979, pp 262-288.

Scemama G., "Traffic Control Practices in Urban Areas", Annual Report of the Nati Res, Inst. on Transport and Safety, Paris, France, 1990.

"SIGOP: Traffic Signal Optimisation Program", Traffic Research Corporation, New York, 1966.

Siljak D., "Complex Dynamic Systems: Dimensionality, Structure and Uncertainity", Large Scale Systems, vol 4, 1983.

Sims A.G., "The Sydney Co-ordinated Adaptive Traffic System. In Urban Transportation Div. of ASCE", Proc. Engineering Foundation Conf. on Research Priorities in Computer Control of Urban Traffic Systems, ASCE, New York, 1979, pp 12-27.

Smith M.J., Ghali M.D., "Two New Models for Assessing Urban Traffic Control and Road Pricing Strategies", Traffic Engineering and Control, vol 33, no 4, April 1992, pp 245-249.

Sugeno, M.,editor, "Industrial Applications of Fuzzy Control", Elsevier Science Pub. Co., 1985.

Takagi, T., and Sugeno, M., "Derivation of Fuzzy Control Rules from Human Operators Control Actions", Proceeding of the IFAC Symposium on Fuzzy Information Representation and Decision Analysis, July 1983, pp 55-60.

Tanscheit R., Scharf E.M., "Experiments with the Use of a Rule-Based Self Organising Controller for Robotics Applications", Fuzzy Sets and Systems, vol 26, 1988, pp 1085-1087.

Teodorovic D., "Fuzzy Sets Theory Applications in Traffic and Transportation", European Journal of Operational Research, Vol 74, No 3, 1994, pp 379-390.

Welstead S.T., "Neural Network and Fuzzy Logic Applications in C/C++", John Wiley & Sons, Inc, 1994.

Wen-Ruey H., "Intelligent Control Based on Fuzzy Algorithms and Genetic Algorithms", Ph.D Thesis., New Mexico State University, 1993.

Wu J., Heydecker B., "A Knowledge Based System for Road Accident Remedial Work", Int. Conf. on Artificial Intelligence (CIVIL-COMP 91), Oxford, UK, 1991.

Yan, J., Ryan, M., and Power, J., " Using fuzzy logic - Towards Intelligent Systems", Prentice-Hall International, UK, 1994.

Yang D., Huhns M., "An Architecture for Control and Communications in Distributed Artificial Intelligence Systems" IEEE Trans. Systems, Man, Cybernetics, vol 15, 1985.

Yasanobu S., Myamoto S., "Automatic Train Control by Predictive Fuzzy Control", Industrial Applications of Fuzzy Control, Sugeno M., ed. Amsterdam, North-Holland, 1985.

Zadeh, L.A., "Fuzzy Sets", Information and Control, vol 8, no.3, 1965, pp 338-353.

Zadeh L.A., "Fuzzy Logic", IEEE Computer, vol 1, 1988, pp 83-93.

Zadeh L.A., "The Calculas of Fuzzy If/Then Rules", AI Expert, March 1992, pp 23-27.

Zhang, B.S., and Edmunds, J.M., "Self-Organising Fuzzy Logic Controller", IEE Proceedings, Part d, Control Theory and Applications, vol 139, no 5, 1992, pp 460-464.

Appendix A

The fuzzy control rules used by the supervisory fuzzy logic controller to adjust the offset at the three intersections, are contained here. A set of 27 fuzzy rules is employed for this purpose. Each fuzzy rule consists of three antecedents shown by their corresponding fuzzy sets and three consequents represented as fuzzy linguistic terms. The input variables are *Vol_diff1*, *Vol_diff2*, and *Vol_diff3*. The output variables are *Ext1*, *Ext2*, and *Ext3*.

[0.00 0.40] [0.00 0.40] [0.00 0.40]
VS VS VS

[0.00 0.40] [0.00 0.40] [0.25 0.75]
SM SM MD

[0.00 0.40] [0.00 0.40] [0.60 1.00]
MD MD HI

[0.00 0.40] [0.25 0.75] [0.00 0.40]
VS SM VS

[0.00 0.40] [0.60 1.00] [0.00 0.40]
SM MD SM

[0.00 0.40] [0.25 0.75] [0.60 1.00]
SM HI HI

[0.00 0.40] [0.60 1.00] [0.25 0.75]
MD HI HI

[0.00 0.40] [0.25 0.75] [0.25 0.75]
SM MD MD

[0.00 0.40] [0.60 1.00] [0.60 1.00]
MD HI HI

[0.25 0.75] [0.00 0.40] [0.00 0.40]
MD SM SM

[0.25 0.75] [0.00 0.40] [0.25 0.75]
MD SM MD

[0.25 0.75] [0.00 0.40] [0.60 1.00]
HI MD HI

[0.25 0.75] [0.25 0.75] [0.00 0.40]
MD MD SM

[0.25 0.75] [0.25 0.75] [0.25 0.75]
HI HI HI

[0.25 0.75] [0.25 0.75] [0.60 1.00]
HI HI VH

[0.25 0.75] [0.60 1.00] [0.00 0.40]
MD HI SM

[0.25 0.75] [0.60 1.00] [0.25 0.75]
HI HI HI

[0.25 0.75] [0.60 1.00] [0.75 1.00]
HI HI VH

[0.60 1.00] [0.00 0.40] [0.00 0.40]
HI MD SM

[0.60 1.00] [0.00 0.40] [0.25 0.75]
MD SM MD

[0.60 1.00] [0.00 0.40] [0.60 1.00]
HI MD HI

[0.60 1.00] [0.25 0.75] [0.00 0.40]
MD MD SM

[0.60 1.00] [0.25 0.75] [0.25 0.75]
HI HI MD

[0.60 1.00] [0.25 0.75] [0.60 1.00]
HI HI HI

[0.60 1.00] [0.60 1.00] [0.00 0.40]
HI MD MD

[0.60 1.00] [0.60 1.00] [0.25 0.75]
HI VH HI

[0.60 1.00] [0.60 1.00] [0.60 1.00]
VH VH VH

Appendix B

The fuzzy knowledge base, generated using Genetic Algorithms (GAs), for adjusting the offset at three adjacent intersections is included here. The fuzzy rules shown below are obtained after 200 generations for a population size of 30. These rules are used by the supervisory fuzzy logic controller to adjust the offset of the three traffic signals. The supervisory fuzzy logic controller has three input and three output variables. The input variables are *Vol_diff1*, *Vol_diff2*, and *Vol_diff3*. The output variables are *Ext1*, *Ext2*, and *Ext3*. The three antecedents of each fuzzy rule are represented by their corresponding fuzzy sets and the three consequents are represented as fuzzy linguistic terms.

[0.00 0.20] [0.00 0.20] [0.00 0.20]
VS HI VS

[0.00 0.20] [0.00 0.20] [0.15 0.50]
VS VS HI

[0.00 0.20] [0.00 0.20] [0.40 0.60]
VS HI HI

[0.00 0.20] [0.00 0.20] [0.50 0.80]
VS VH MD

[0.00 0.20] [0.00 0.20] [0.70 1.00]
MD VS SM

[0.00 0.20] [0.15 0.50] [0.00 0.20]
VS VS HI

[0.00 0.20] [0.15 0.50] [0.15 0.50]
SM VH HI

[0.00	0.20]		[0.15	0.50]		[0.40	0.60]	
VS	**VH**	**VS**						
[0.00	0.20]		[0.15	0.50]		[0.50	0.80]	
VS	**HI**	**VS**						
[0.00	0.20]		[0.15	0.50]		[0.70	1.00]	
VS	**VS**	**SM**						
[0.00	0.20]		[0.40	0.60]		[0.00	0.20]	
MD	**MD**	**HI**						
[0.00	0.20]		[0.40	0.60]		[0.15	0.50]	
VS	**VH**	**HI**						
[0.00	0.20]		[0.40	0.60]		[0.40	0.60]	
VS	**SM**	**MD**						
[0.00	0.20]		[0.40	0.60]		[0.50	0.80]	
VS	**HI**	**VH**						
[0.00	0.20]		[0.40	0.60]		[0.70	1.00]	
MD	**HI**	**MD**						
[0.00	0.20]		[0.50	0.80]		[0.00	0.20]	
VS	**HI**	**VH**						
[0.00	0.20]		[0.50	0.80]		[0.15	0.50]	
VS	**SM**	**HI**						
[0.00	0.20]		[0.50	0.80]		[0.40	0.60]	
VS	**HI**	**VS**						
[0.00	0.20]		[0.50	0.80]		[0.50	0.80]	
VS	**VH**	**VH**						
[0.00	0.20]		[0.50	0.80]		[0.70	1.00]	
VS	**VH**	**SM**						
[0.00	0.20]		[0.70	1.00]		[0.00	0.20]	
SM	**SM**	**SM**						
[0.00	0.20]		[0.70	1.00]		[0.15	0.50]	
VS	**MD**	**SM**						

[0.00	0.20]	[0.70	1.00]	[0.40	0.60]
SM	**VH**	**HI**			
[0.00	0.20]	[0.70	1.00]	[0.50	0.80]
VS	**VS**	**VH**			
[0.00	0.20]	[0.70	1.00]	[0.70	1.00]
VS	**SM**	**VH**			
[0.15	0.50]	[0.00	0.20]	[0.00	0.20]
VS	**VH**	**MD**			
[0.15	0.50]	[0.00	0.20]	[0.15	0.50]
SM	**SM**	**MD**			
[0.15	0.50]	[0.00	0.20]	[0.40	0.60]
VS	**SM**	**VS**			
[0.15	0.50]	[0.00	0.20]	[0.50	0.80]
SM	**VH**	**VH**			
[0.15	0.50]	[0.00	0.20]	[0.70	1.00]
MD	**VS**	**SM**			
[0.15	0.50]	[0.15	0.50]	[0.00	0.20]
MD	**HI**	**VS**			
[0.15	0.50]	[0.15	0.50]	[0.15	0.50]
VS	**HI**	**HI**			
[0.15	0.50]	[0.15	0.50]	[0.40	0.60]
SM	**HI**	**MD**			
[0.15	0.50]	[0.15	0.50]	[0.50	0.80]
MD	**VH**	**MD**			
[0.15	0.50	[0.15	0.50]	[0.70	1.00]
VH	**SM**	**MD**			
[0.15	0.50]	[0.40	0.60]	[0.00	0.20]
MD	**HI**	**SM**			
[0.15	0.50]	[0.40	0.60]	[0.15	0.50]
VS	**VH**	**VH**			

[0.15	0.50]		[0.40	0.60]		[0.40	0.60]	
VH	**VH**	**VS**						
[0.15	0.50]		[0.40	0.60]		[0.50	0.80]	
VH	**VS**	**VS**						
[0.15	0.50]		[0.40	0.60]		[0.70	1.00]	
VH	**HI**	**HI**						
[0.15	0.50]		[0.50	0.80]		[0.00	0.20]	
MD	**HI**	**SM**						
[0.15	0.50]		[0.50	0.80]		[0.15	0.50]	
VH	**MD**	**VS**						
[0.15	0.50]		[0.50	0.80]		[0.40	0.60]	
VS	**HI**	**HI**						
[0.15	0.50]		[0.50	0.80]		[0.50	0.80]	
SM	**VS**	**VS**						
[0.15	0.50]		[0.50	0.80]		[0.70	1.00]	
VH	**VH**	**SM**						
[0.15	0.50]		[0.70	1.00]		[0.00	0.20]	
MD	**VH**	**HI**						
[0.15	0.50]		[0.70	1.00]		[0.15	0.50]	
MD	**MD**	**HI**						
[0.15	0.50]		[0.70	1.00]		[0.40	0.60]	
HI	**HI**	**SM**						
[0.15	0.50]		[0.70	1.00]		[0.50	0.80]	
VS	**SM**	**VS**						
[0.15	0.50]		[0.70	1.00]		[0.70	1.00]	
HI	**VS**	**VH**						
[0.40	0.60]		[0.00	0.20]		[0.00	0.20]	
VH	**SM**	**VS**						
[0.40	0.60]		[0.00	0.20]		[0.15	0.50]	
VH	**VS**	**HI**						

[0.40 0.60] [0.00 0.20] [0.40 0.60]
SM VS SM

[0.40 0.60] [0.00 0.20] [0.50 0.80]
VH MD VH

[0.40 0.60] [0.00 0.20] [0.70 1.00]
SM HI SM

[0.40 0.60] [0.15 0.50] [0.00 0.20]
VS SM VH

[0.40 0.60] [0.15 0.50] [0.15 0.50]
SM HI SM

[0.40 0.60] [0.15 0.50] [0.40 0.60]
VS HI VH

[0.40 0.60] [0.15 0.50] [0.50 0.80]
VS VH HI

[0.40 0.60] [0.15 0.50] [0.70 1.00]
HI SM VS

[0.40 0.60] [0.40 0.60] [0.00 0.20]
VH VS HI

[0.40 0.60] [0.40 0.60] [0.15 0.50]
HI VS VS

[0.40 0.60] [0.40 0.60] [0.40 0.60]
VS HI HI

[0.40 0.60] [0.40 0.60] [0.50 0.80]
MD VS VS

[0.40 0.60] [0.40 0.60] [0.70 1.00]
VS VH HI

[0.40 0.60] [0.50 0.80] [0.00 0.20]
VH VH HI

[0.40 0.60] [0.50 0.80] [0.15 0.50]
VH MD HI

[0.40 0.60] [0.50 0.80] [0.40 0.60]
HI SM MD

[0.40 0.60] [0.50 0.80] [0.50 0.80]
HI MD VS

[0.40 0.60] [0.50 0.80] [0.70 1.00]
VH VH VS

[0.40 0.60] [0.70 1.00] [0.00 0.20]
MD MD SM

[0.40 0.60] [0.70 1.00] [0.15 0.50]
VS VS VS

[0.40 0.60] [0.70 1.00] [0.40 0.60]
HI MD VS

[0.40 0.60] [0.70 1.00] [0.50 0.80]
VH MD VH

[0.40 0.60] [0.70 1.00] [0.70 1.00]
HI HI SM

[0.50 0.80] [0.00 0.20] [0.00 0.20]
MD HI VS

[0.50 0.80] [0.00 0.20] [0.15 0.50]
MD MD VH

[0.50 0.80] [0.00 0.20] [0.40 0.60]
VH VH MD

[0.50 0.80] [0.00 0.20] [0.50 0.80]
HI MD VS

[0.50 0.80] [0.00 0.20] [0.70 1.00]
VS HI VH

[0.50 0.80] [0.15 0.50] [0.00 0.20]
MD VH VS

[0.50 0.80] [0.15 0.50] [0.15 0.50]
MD SM MD

[0.50	0.80]	[0.15	0.50]	[0.40	0.60]	
VS	**MD**	**HI**				
[0.50	0.80]	[0.15	0.50]	[0.50	0.80]	
MD	**SM**	**MD**				
[0.50	0.80]	[0.15	0.50]	[0.70	1.00]	
SM	**MD**	**VS**				
[0.50	0.80]	[0.40	0.60]	[0.00	0.20]	
HI	**VS**	**VH**				
[0.50	0.80]	[0.40	0.60]	[0.15	0.50]	
HI	**MD**	**SM**				
[0.50	0.80]	[0.40	0.60]	[0.40	0.60]	
VH	**VS**	**VH**				
[0.50	0.80]	[0.40	0.60]	[0.50	0.80]	
MD	**VH**	**MD**				
[0.50	0.80]	[0.40	0.60]	[0.70	1.00]	
VH	**MD**	**MD**				
[0.50	0.80]	[0.50	0.80]	[0.00	0.20]	
VH	**VH**	**HI**				
[0.50	0.80]	[0.50	0.80]	[0.15	0.50]	
VS	**SM**	**MD**				
[0.50	0.80]	[0.50	0.80]	[0.40	0.60]	
HI	**HI**	**SM**				
[0.50	0.80]	[0.50	0.80]	[0.50	0.80]	
VS	**SM**	**SM**				
[0.50	0.80]	[0.50	0.80]	[0.70	1.00]	
MD	**HI**	**MD**				
[0.50	0.80]	[0.70	1.00]	[0.00	0.20]	
VS	**VH**	**MD**				
[0.50	0.80]	[0.70	1.00]	[0.15	0.50]	
MD	**HI**	**SM**				

[0.50	0.80]	[0.70	1.00]	[0.40	0.60]
SM	**HI**	**VH**			
[0.50	0.80]	[0.70	1.00]	[0.50	0.80]
MD	**VH**	**VS**			
[0.50	0.80]	[0.70	1.00]	[0.70	1.00]
MD	**HI**	**VS**			
[0.70	1.00]	[0.00	0.20]	[0.00	0.20]
SM	**VS**	**VS**			
[0.70	1.00]	[0.00	0.20]	[0.15	0.50]
HI	**MD**	**VH**			
[0.70	1.00]	[0.00	0.20]	[0.40	0.60]
HI	**SM**	**MD**			
[0.70	1.00]	[0.00	0.20]	[0.50	0.80]
VH	**VH**	**HI**			
[0.70	1.00]	[0.00	0.20]	[0.70	1.00]
MD	**VH**	**MD**			
[0.70	1.00]	[0.15	0.50]	[0.00	0.20]
HI	**VH**	**SM**			
[0.70	1.00]	[0.15	0.50]	[0.15	0.50]
HI	**HI**	**VS**			
[0.70	1.00]	[0.15	0.50]	[0.40	0.60]
HI	**VS**	**HI**			
[0.70	1.00]	[0.15	0.50]	[0.50	0.80]
HI	**SM**	**SM**			
[0.70	1.00]	[0.15	0.50]	[0.70	1.00]
MD	**VH**	**VH**			
[0.70	1.00]	[0.40	0.60]	[0.00	0.20]
VH	**VS**	**VH**			
[0.70	1.00]	[0.40	0.60]	[0.15	0.50]
HI	**MD**	**VS**			

[0.70	1.00]		[0.40	0.60]		[0.40	0.60]	
SM	**SM**	**MD**						
[0.70	1.00]		[0.40	0.60]		[0.50	0.80]	
VH	**SM**	**VS**						
[0.70	1.00]		[0.40	0.60]		[0.70	1.00]	
VH	**MD**	**HI**						
[0.70	1.00]		[0.50	0.80]		[0.00	0.20]	
MD	**VS**	**VS**						
[0.70	1.00]		[0.50	0.80]		[0.15	0.50]	
HI	**VH**	**VS**						
[0.70	1.00]		[0.50	0.80]		[0.40	0.60]	
VH	**VH**	**VH**						
[0.70	1.00]		[0.50	0.80]		[0.50	0.80]	
VH	**HI**	**VH**						
[0.70	1.00]		[0.50	0.80]		[0.70	1.00]	
HI	**MD**	**SM**						
[0.70	1.00]		[0.70	1.00]		[0.00	0.20]	
HI	**SM**	**VH**						
[0.70	1.00]		[0.70	1.00]		[0.15	0.50]	
MD	**VS**	**MD**						
[0.70	1.00]		[0.70	1.00]		[0.40	0.60]	
HI	**VH**	**MD**						
[0.70	1.00]		[0.70	1.00]		[0.50	0.80]	
HI	**HI**	**HI**						
[0.70	1.00]		[0.70	1.00]		[0.70	1.00]	
VH	**VH**	**HI**						