5-2009

Neurofuzzy control to address stochastic variation in actuated-coordinated systems at closely-spaced intersections

Xiaoli Sun

University of Tennessee

To the Graduate Council:

I am submitting herewith a dissertation written by Xiaoli Sun entitled "Neurofuzzy control to address stochastic variation in actuated-coordinated systems at closely-spaced intersections." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Civil Engineering.

Thomas Urbanik II, Major Professor

We have read this dissertation and recommend its acceptance:

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting here with a dissertation written by Xiaoli Sun entitled "Neurofuzzy Control to Address Stochastic Variation in Actuated-Coordinated Systems at Closely-Spaced Intersections." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Civil Engineering.

Thomas Urbanik II, Major Professor

We have read this dissertation
and recommend its acceptance:

Lee D. Han

Christopher R. Cherry

J. Wesley Hines

Accepted for the Council:

Carolyn R. Hodges, Vice Provost and
Dean of the Graduate School

(Original signatures are on file with official student records.)

Neurofuzzy Control to Address Stochastic Variation in Actuated-Coordinated Systems at Closely-Spaced Intersections

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Xiaoli Sun
May 2009

ACKNOWLEDGEMENTS

This dissertation could not have been completed without so many people's help. I know it will be impossible to acknowledge all the people who have helped make these past three and half years the rewarding experience it has been.

Firstly, I want to thank my advisor, Dr. Thomas Urbanik II, for the vast amount of knowledge he taught me not only in the field of Transportation Engineering, but also in life. I am especially thankful to Dr. Urbanik for his enormous amount of patience and support he has given me during the whole study period. I also want to thank Dr. Lee D. Han, Dr. Christopher R. Cherry and Dr. J. Wesle Hines for being my teachers and committees. You added so much value to my experience at UT.

Then I thank my fellow students at the lab, Jun Xu, Shuguang Ji, Jianwen Si, Fang Yuan, Jan-Mou Li, and so many other friends. I will remember their help and friendships forever.

Last, and most important, I must extend my most heartfelt thank-you to my family for the encouragement and support over the years. Without having them in my life to keep me going, this dissertation would never have been completed. My daughter, Sophia, you are the most beautiful angle beside me giving me the powerful encouragement more than you will ever know.

ABSTRACT

This dissertation documents a method of addressing stochastic variation at closely-spaced signalized intersections using neurofuzzy control. Developed on the conventional actuated-coordinated control system, the neurofuzzy traffic signal control keeps the advantage of the conventional control system. Beyond this, the neurofuzzy signal control coordinates the coordinated phase with one of the non-coordinated phases with no reduction of the “green band” assigned to the coordination along the arterial, reduces variations of traffic signal times in the cycle caused by “early return to green”, hence, makes more sufficient utilization of green time at closely-spaced intersections.

The neurofuzzy signal control system manages a non-coordinated movement in order to manage queue spillbacks and variations of signal timings. Specifically, the neurofuzzy controller establishes a “secondary coordination” between the upstream coordinated phase (through phase) and the downstream non-coordinated phase (left turn phase) based on real-time traffic demand. Under the fuzzy logic signal control, the traffic from the upstream intersection can arrive and join the queue at the downstream left turn lane and be served, and hence, less possibly be blocked on the downstream left turn lane. This “secondary coordination” favors left turn progression and, hence, reduces the queue spillbacks. The fuzzy logic method overcomes the natural disadvantage of currently widely used actuated-coordinated traffic signal control in that the fuzzy

logic method could coordinate a coordinated movement with a non-coordinated movement.

The experiment was conducted and evaluated using a simulation model created using the microscopic simulation program - VISSIM. The neurofuzzy control algorithm was coded with MATLAB which interacts with the traffic simulation model via VISSIM's COM interface. The membership functions in the neurofuzzy signal control system were calibrated using reinforcement learning to further the performance. Comparisons were made between the trained neurofuzzy control, the untrained neurofuzzy control, and the conventional actuated-coordinated control under five different traffic volumes. The simulation results indicated that the trained neurofuzzy signal control outperformed the other two for each traffic case. Comparing to the conventional actuated-coordinated control, the trained neurofuzzy signal control reduced the average delay by 7% and the average number of stops by 6% under the original traffic volume; as traffic volume increasing to 120%, the reductions doubled.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

With increasing traffic volume on streets in urban areas, the safety and the effectiveness of the intersection performance are more critical. This high volume condition is especially critical where two intersections are “closely spaced,” which is defined, for the purpose of this research, as less than 800 ft. Because of the limited room between the close-spaced intersections, traffic flows affect each other significantly, and congestions occur more frequently and severely even though the demand is below the capacity.

Due to the limited space in urban area and budget constraints, adding roads or lanes is often impractical, if not impossible, at most intersections. Optimization of the traffic signal control is subsequently a more practical, and often no less effective, way for mitigating traffic problems at closely-spaced intersections. A good signal control scheme should be able to response to the traffic flow in real time and carry out the optimized control to improve the efficiency of traffic operation.

Background

Efforts have been conducted for many years to address the unique traffic characteristics at closely-spaced intersections. These efforts include using two coordinated traffic signals and single traffic signal controller to control both intersections. Since the adaptive signal control conception occurred in 1960’s

[22], researchers and engineers have developed lots of signal control methods to better respond to traffic flows in real-time. Despite to those research efforts for adaptive signal control, the effort to address traffic movements is largely to diamond interchanges, and the effort at closely-spaced intersections is limited.

Statement of the Problem

Closely-spaced intersections, including diamond interchanges, usually exist in urban area. The performance of closely-spaced intersections has a strong relationship with signal riming due to the interdependence of flows, queues, and timing at individual intersections. The traffic condition of closely-spaced intersections proposes a number of challenges for the actuated-coordinated control.

Since queue storage capacity is limited, the queue at a downstream intersection can easily spill back to the upstream intersection. Therefore vehicles which are waiting at the downstream intersection can block the normal traffic flow at the upstream intersection. This results in increased congestions on the road network even when the traffic flow is not close to the capacity.

Meanwhile, since the upstream approaches are blocked, the downstream green time is sometimes not fully utilized. This phenomenon is called “demand starvation” [10]. As mentioned in the HCM2000, pp. 26-7, “Demand starvation occurs when portions of the green at the downstream intersection are not used because conditions prevent vehicles at the upstream intersection from reaching the downstream stop line. These conditions at the upstream intersection can include delays or blockage due to queue overflow from another lane group. Demand starvation occurs in one of two ways: (1) Queues from the downstream intersection effectively block departures from the upstream intersection during part or all of the upstream green. This reduces the effective green time for flow at the upstream location during the green time at the downstream intersection. (2) Signal coordination between the two intersections is suboptimal even without downstream queuing. As a result, sometimes the upstream signal is red while unsaturated flow conditions prevail during the green at the downstream signal. ” The focus of this research is the former one, i.e., even when the two intersections are supposedly well coordinated, the demand starvation may still occur because of the downstream queue blocking the departure the upstream intersection.

The conditions causing queue spillbacks and demand starvation rise from three aspects. The first is the limited space between two intersections; the second is the fluctuating traffic on the side streets or the sporadic occurrence of pedestrian phases; the last derives from the nature of actuated-coordinated systems. Actuated control system has detectors at approaching lanes and extends green time on detection of an arriving vehicle. Actuated-coordinated control is implemented if distance between intersections along the arterial is short to synchronize multiple intersections to enhance the operation of one or more directional movements. Actuated-coordinated systems use an offset (the time relationship, expressed in seconds or percent of cycle length, determined by the difference between a defined point in the coordinated green and a system reference point) between intersections, and let unused green time from side streets transfer to the coordinated direction, which results in the coordinated phase return to green earlier than designed. Consequently, the actual time-space relationship between two adjacent intersections varies based on traffic demand and the response of the control system to that demand. This variation creates queue spillbacks and demand starvation.

In addition, if the left turn is critical at the downstream intersection, coordination with the upstream through movement becomes critical. Due to the limited queue storage capacity between closely-spaced intersections, the queues spills back to and blocks the upstream intersection easily and meanwhile causes demand starvation phenomenon at the downstream intersection. This is why many diamond interchanges are controlled by a single controller in order to closely control the time-space relationship between the closely-spaced intersections.

Currently, the most widely used traffic control schemes are the fixed-timed and the vehicle-actuated signal controls. The fixed-timed control sets up fixed green time for each direction. Different timing plans for different month, day of week, and time of day can be designed and deployed, but are still schemes preprogrammed with no consideration of fluctuations in real-time traffic demand. The vehicle-actuated signal control uses detectors to detect the arrivals of vehicles and provides certain flexibility on the green time. If intersections are closely spaced (less than 800 feet), traffic signal controllers must be coordinated in order to provide a smooth traffic flow on the arterial street to reduce delay and number of stops. However, traffic flows fluctuate during various days and the time of day, and even cycle by cycle. While vehicle actuated phases in a coordinated-actuated signal control system can partially address fluctuations in flow, their inability to closely coordinate phases at adjacent closely-spaced intersections introduces other operational problems. Therefore, there is still room for improvement for traffic signal control techniques at closely-spaced intersections. The following two sections look close to the conventional isolated traffic signal control and the conventional coordinated traffic signal control, and explain why they are short of addressing the unique traffic operation at closely space intersections.

Conventional Isolated Traffic Signal Control

An isolated intersection means the intersection is far way to the adjacent intersection. Since the distance between two intersections is large, the interaction between intersections is small and the arriving of traffic flows at isolated intersections is not in the form of platoon. For isolated signalized intersection, the controller controls the traffic flows without considering the traffic flow operations at adjacent intersections.

Both the fixed-timed control and the vehicle-actuated control can be applied at isolated intersections. The fixed-timed control is the simplest signal control scheme. It set up all times, including the cycle length, phase sequence, the green interval and change interval in each phase. Although it has different constant timing plans for month of year, day of week and time of day, and the timing plans are optimized according to off-line traffic volumes, the fixed-timed controller does not respond to the traffic flows in real time. The conventional vehicle-actuated controller at isolated intersections attempts continuously to adjust green times based on the input data collected through detectors. In this type of control, the basic time parameters at each phase are minimum green, passage time and maximum green. In the base situation, the green signal group gives at least the minimum green time. If the demand exceeds minimum green sufficiency, the green time can be extended stepwise with the lengths of the extension interval to the maximum green. After green extension, the signal group can go to red or remain as a passive green. The passive green can be terminated by conflicting signal groups. Comparing to the fixed-timed traffic signal, the advantages of the actuated signal is that it can adjust the green times and cycle length according to the arriving of traffic flows. But its advantage is limited. First, the vehicle-actuated controller can only collect traffic flow from where detectors are located and is "blind" to traffic flows outside of the range of detectors. Second, when traffic flows are oversaturated, the controller extends the green time to its maximum green value which is fixed in the controller. Under this condition, the vehicle-actuated controller works like a fixed-timed controller and lost its ability to respond to traffic flow. Third, it detects the arriving of a vehicle only when the corresponding phase is in green and does not consider the urgency level of traffic demand from the conflicting approaches. Obviously, the urgency of the traffic demand is very different between one vehicle at the conflicting approach and 30 vehicles at the conflicting approach.

Conventional Coordinated Traffic Signal Control

If two intersections are close enough (within 0.5 mile) on a major route or in a network of major routes, the traffic movement along an arterial is in a platoon. The signals along the major routes are usually coordinated in order to favor the efficiency of the prevailing traffic movement on the corridor. Signals spaced farther than 0.5 mile may also be coordinated if the platoon can be maintained. [55] The coordination system can either be liner (arterial) or in a network (such as a CBD grid). The coordination means that the traffic signals are included in a computerized signal system, all coordinated traffic signals have a common cycle, and the timing and phasing of the signals is interrelated to the settings of the adjacent intersection. The offset between adjacent intersections, i.e., the difference in start times for the through green at adjacent intersections, is mainly decided with intersection spacing (distance between intersections) and progression speed. The beginning and end of the green period on the coordinated signals create a “green band” so that the traffic signals turn green as the platoon arrives at the intersection. The benefit of coordination is the improved capacity of groups of traffic signals, higher level of service, reduction in overall network delay and number of stops, etc., than the isolated operation.

Several conventional coordinated traffic signal models have been developed in the past. Despite the differences among these models, the conventional coordination control strategies can be roughly classified into three categories.

The first is the fixed-timed coordination. This control scheme fixes up all timings including cycle, split and offsets within the coordinated signals and does not respond to traffic flows on roads. The basic assumption is that the traffic is under-saturated, traffic flow is steady, and the speed is approximately constant at the design speed. The cycle time, split and offsets are optimized according to the traffic demand in terms of reducing the average delay and providing progression. The signal timings vary with respect to time of day to accommodate peak hour and off-peak hour traffic conditions. This type of control strategy is suitable to an urban area where traffic flow does not fluctuate significantly.

The second is the semi-actuated coordination. The coordinated phases along the corridor are non-actuated with a fixed minimum split while the non-coordinated phases are fully actuated to respond to traffic flows. There are fixed yield points or permissive periods in the coordinated phases for calls on the non-coordinated phases. If there are no calls on non-coordinated phases during the permissive periods, the green stays in the coordinated phases. Otherwise, the controller goes to the first next phase(s) with calls according to the standard phase sequence. The coordinated phases receive all unused green times after serving non-coordinated movements. This phenomenon is called “early return to green” which may cause poor progression along the arterial.

The third is the actuated coordination. In this control, both the coordinated and non-coordinated phases are actuated. In the coordinated phases, both of the beginning and the end of the coordinated green can vary. Same as in semi-actuated coordination, the actuated coordination also has the “early return to green” phenomenon. If calls on the coordinated phases are received within the “actuated permissive period”, the controller will extend the coordinated green for a period of time, during which the controller searches for an appropriate gap on the coordinated phase and then commits gap out, or force off if the time exceeds the assigned split. This type of control strategy can provide a protection to vehicles in the “dilemma zone”, as well as giving the unused coordinated phase split time to the non-coordinated phases.

Comparing to the fixed-timed coordination, obviously the latter two provide more flexibility to accommodate fluctuating traffic flows. At low to medium traffic volume conditions, the semi-actuated and the actuated coordination work well, but they still have natural limitations especially under high traffic volumes. For instance, under the over-saturated condition when traffic demand from all directions is high, the semi-actuated and actuated coordination operate nearly like fixed-timed coordination. In this case, some other control strategies, such as the queue management strategies, should be applied. Pedestrian also cause issues. If a pedestrian phase occurs, the cycle at that intersection is changed to allow the pedestrian to cross, which may disturb the progression along the corridor.

Study Objectives

The effectiveness of urban traffic control systems depends heavily on its ability of reacting to changes in traffic patterns. When this ability becomes an integral part of a traffic control unit, the traffic control strategy will be able to react to changes in traffic conditions.

At closely-spaced intersections, the conventional signal control strategies are not suited to all traffic conditions due to their natural limits. The fixed-timed coordination has no flexibility and cannot respond to real-time traffic fluctuation. The semi-actuated and actuated coordination have more flexibility than the fixed one, but their flexibility is primarily in giving back time to the coordinated phase. None of them can adjust the beginning and end time of the coordinated green (i.e., “green band”) according to the real-time traffic condition on multiple approaches at both intersections, or adjust the time relationship between non-coordinated phases with coordinated phases. Furthermore, “early return to green” is one of the reasons causing the “demand starvation” phenomenon (if the “early return to green” occurs at the downstream intersection while the upstream intersection is still in red).

What needed is a strategy which can accommodate the special characteristics of the closely-spaced intersections, including being adaptive to the fluctuating traffic flow, in other words, a strategy is needed to reduce the stochastic variation of signal times in the cycle caused by fluctuating traffic flows .

The objective of this research is to develop a new traffic signal control algorithm that can improve the time relationship between phases at closely-spaced intersections to mitigate the “demand starvation” situation, provide more continuous movement through the two intersections, and reduce congestion caused by high left-turning traffic volume while keeping the two closely-spaced intersections within the coordination system along the arterial.

Study Scope

The study is based on the following hypothesis and assumptions.

The hypothesis of this study is that the delay and the number of stops can be reduced at closely-spaced intersections by adjusting the start and end time of critical phases based on neurofuzzy control.

The assumptions include:

(1) Two closely-spaced intersections.

(2) Nearly saturated (close to congestion) traffic condition with high interior left-turning volume (the left-turns from lanes between the two intersections). Under this condition, poor signal controls could result in queue spilling back to the upstream intersection.

(3) The arriving traffic is fluctuating.

(4) The base line condition is the currently running signal operation – the conventional actuated-coordinated control.

(5) The reference point of the offset is the beginning of yellow, and the offset between the two intersections has been optimized.

(6) The interior left turn movements lead to the through movement (i.e., leading left turn).

This research focus on adjusting the time relationship between the interior left turn phase (non-coordinated phase) and the upstream through phase (coordinated phase), that is, finding an appropriate beginning point of the downstream interior left turn phase in the cycle according to the real-time traffic condition at both intersections in order to reduce spillback and demand starvation, and consequently improve the effectiveness of the operation. The study is not aimed at on-line adjustment of the offset between two intersections because the study scope is to address the problems caused by interior left turns which do not belong to the arterial progression.

As mentioned before, the beginning of the coordination green band is variable due to the “early return to green”. If the downstream intersection returns to green too early i.e., the interior left turn phase occurs too early and gaps out too early, then when the upstream intersection turns to green, the downstream left turn phase will have been set to red. Consequently, the left turning volume from the upstream has to be stored in the downstream left turn bay. Due to the insufficient storage capacity of the left turn bay, the queue might spill back to the upstream intersection and block the intersection. Additionally, if the downstream interior left turn phase occurs too early, it has no chance to serve the demand from the upstream intersection resulting in “demand starvation” phenomenon. The reason comes from the variable time relationship between the downstream interior left turn phase and the upstream through phase. If we can provide a “second offset” between the these two phases, so that the upstream traffic can make sufficient use of the interior left turn phase split, we can mitigate the “early return to green” and “demand starvation” problems, improve the continuous movement throughout the two intersections and potentially reduce the delay and the number of stops.

Because of the interdependent relationship among traffic flow, queuing and timing, there is no fixed “offset” between the downstream interior left turn phase and the upstream through phase. This “offset” depends on the real-time traffic condition at both intersections. For example, if the side-street demand at the upstream intersection is very low on a specific cycle, the upstream through phase will begin very early; in this case, the downstream interior left turn phase should also begin early enough before the upstream through phase begins.

The current vehicle-actuated control strategy is designed for arterial progression and not for variations in non-coordinated phases. The natural configuration of the vehicle-actuated control strategy cannot provide the “second offset” between the coordinated phase and a non-coordinated phase. A new method/algorithm is needed to solve this issue.

The strength of the fuzzy logic is that it can import expert knowledge into the control system, and the fuzzy logic based method is explored in this research. To improve the efficiency of the fuzzy controller, the parameters in the fuzzy control system is adjusted with neural network algorithms.

This research reduces the negative effects of actuated controls on progression of non-coordinated phase without loosing the benefits of actuation and coordination.

Contribution

The contribution of this research is the development of an algorithm to progress a non-coordinated phase at two closely-spaced intersections operating as actuated-coordinated intersections.

Organization of the Dissertation

This dissertation consists five chapters. The Chapter I is an introductory chapter. The literature review is summarized in Chapter II. The development of the neurofuzzy control system is descried in Chapter III. Chapter IV illustrates the evaluation of the neurofuzzy control system. Finally, Chapter V includes conclusions and recommendations.

CHAPTER II

LITERATURE REVIEW

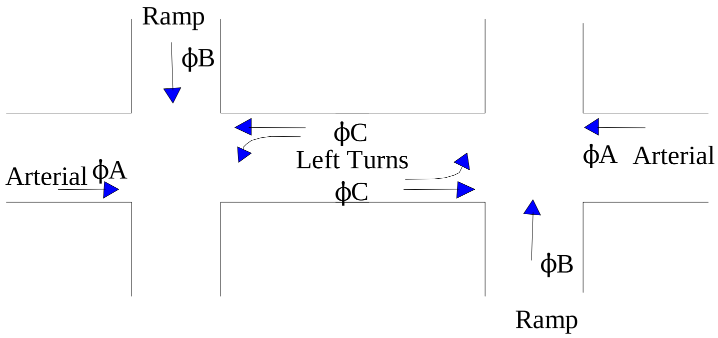
This chapter presents the literature review of signal control strategies on closely-spaced intersections, state-of-the-art of adaptive signal strategies and artificial intelligence approaches in traffic signal control. The emphasis of the review of artificial intelligence approaches in traffic signal control is on fuzzy logic and neurofuzzy methods.

Signal Control Strategies on Closely-spaced Intersections

As a typical example of closely-spaced intersections, diamond interchanges have been widely studied, they offer many insights into improved control practices for all closely-spaced intersections. Special phasing pattern and timing are often needed due to the geometry of closely-spaced intersections and the interactive traffic flow. Various signal schemes and phasing optimization software have been developed for diamond interchanges. The commonly used schemes include Texas 3-phase, Texas Transportation Institute (TTI) 4-Phase, etc. Signal control schemes at closely space intersections can be classified into two categories: one employs a single controller for both intersections while the other used two.

The commonly used single controller implementations are the three-phase and the four-phase controls. The three-phase control (Figure 1) has the interior left turns (left turns from between the two intersections) lag arterial through movements and let traffic from ramps move together. This type of control favors the progression for the arterial through movement, but does not guarantee a queue clearance on interior left-turn lanes (left turn lanes between the two intersections) and ramps. So under the condition of high left turning volumes and insufficient queue storage capacity between the two intersections, the queue spillback may occur.

The four-phase control serves the four exterior movements in a clockwise manner, with all movements progressed and the interior left turn queue kept clear, see Figure 2. This type of control provides a better left turn service and is suitable to tightly spaced intersections with high left turning volumes where queue spillback is a concern. The four-phase can reduce the number of vehicle stops and queues on the interior space. The disadvantage is a higher delay and a lower capacity comparing with the three-phase control scheme. (However, three-phase control may break down at higher volumes due to queue spill back). Progression along the arterial is also more difficult.



Left Side Phase Sequence			Left Turn Sequence	Right Side Phase Sequence		
A	B	C	Lead-Lead	A	B	C
A	C	B	Lag-Lead	A	B	C
A	B	C	Lead-Lag	A	C	B
A	C	B	Lag-Lag	A	C	B

Figure 1. Three-phase control.

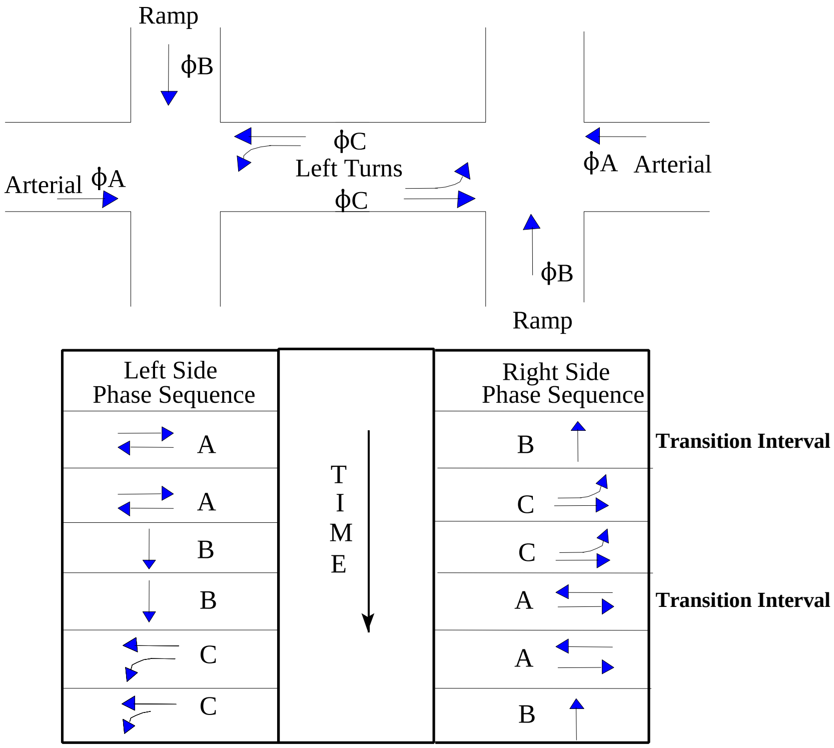


Figure 2. Four-phase control.

Considering that conventional control strategies and maintenance issues are more familiar to engineers, it is natural to have two controllers to control two intersections regardless of spacing. The two-controller implementation has the controllers operate either separately or coordinated (Figure 3). The former is equivalent to two isolated controllers each controlling one intersection independently from the other. The separate intersection mode can reduce stops at interchanges under low-volume conditions, especially when the interior left turns can operate as permissive left turns. This type of control is useful for low volume condition where the interior left turn progression is not very important. For those where the traffic volume is high or the left turn movement is a concern, it may be better to have two coordinated controllers to provide better progression. The two-controller implementation also has some disadvantages. Firstly, if not properly timed, it can cause driver expectancy problems. Also, the coordinated operation requires a background cycle.

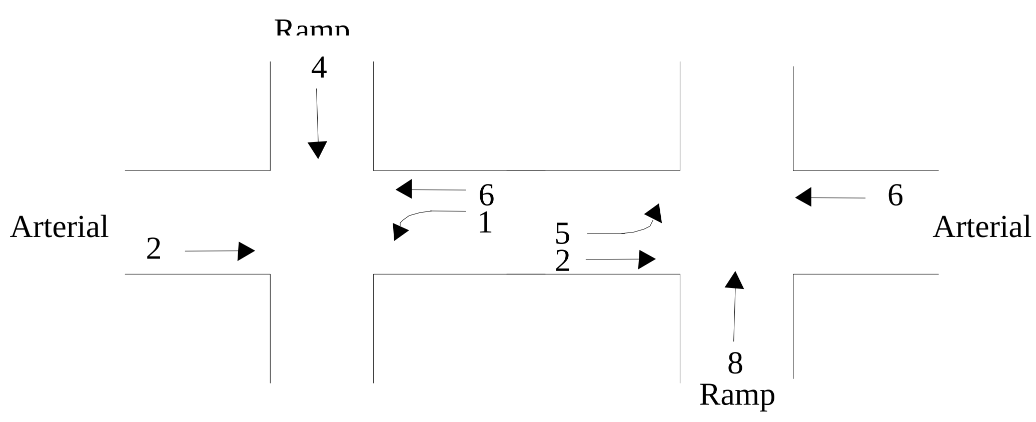


Figure 3. Two-controller control.

As to the conditions at closely-spaced intersections, though not exactly the same as diamond interchanges, they do have many similar features with diamond interchanges. These similar features include the closely-space geometry, the highly interactive traffic flow between two intersections, left turn issues, the spillback phenomenon, etc. Thus, diamond interchange control concepts can be adopted to closely-spaced intersections. However, as the number of movements increases, which is common at typical urban intersections, the “diamond solution” becomes impractical.

Traffic engineers have made many efforts on improving signals at diamond interchanges and closely-spaced intersections from both the hardware aspects and the software aspects. As for the hardware aspects, Engelbrecht et al. [17] [47] investigated the current widely used controllers including Eagle EPAC300 and Naztec 980, and verified some advanced controller features that were not often used but had the potential to improve traffic operations at signalized diamond interchanges. They also pointed out that video detection was very suitable for zone detection, but it might not be sufficiently accurate to serve as input to controller features that rely on the accurate point detection of vehicles. One of the main finding of the research was the potential usefulness of the separate intersection diamond control mode. The free separate intersection mode can significantly reduce number of stops at interchanges under low-volume conditions. The coordinated separated intersection mode has the potential to provide more efficient operation than the three-phase control or four-phase control under certain conditions that can be determined with signal optimization software such as PASSER III, Synchro, and TRANSYT-7F.

As for the software aspects, Chaudhary and Chu [18] analyzed the Texas 3-phase and TTI 4-phase under various interior distances, volume conditions and volume distributions. The authors provided technologies for analyzing and optimizing the flow of traffic in congested diamond interchange environments. These technologies included the mathematical procedure for predicting the throughput capacity for a given timing plan and a given traffic pattern, the optimum cycle length selection method for a diamond interchange, and an iterative procedure for calculating green splits for a diamond interchange with an adjacent signal. They also provided guidelines for coordinating diamond interchanges with adjacent traffic signals on the arterial.

Messer [46] described the PASSER III, a computer program specifically designed to analyze the operations of an isolated diamond interchange and to determine the best pre-timed signal plan in order to minimize the average delay. The program evaluated all basic interchange signal phasing sequences, including all possible patterns from lead-lead, lag-lead, lead-lag, to lag-lag phasing (see Figure 1), and compared the relative merits of different diamond interchange phasing schemes, such as three-phase control scheme and four-phase control scheme. The program also analyzed the protected-permissive left turn phase at signalized intersections. The author concluded that the best minimum delay, pre-timed diamond interchange signal phasing pattern can be estimated using

PASSERIII. However, it should be noted that optimization techniques are based on prescribed volumes.

Messer [28] made some simulation studies on traffic operations at oversaturated, closely-spaced signalized intersections. The author developed a calibrated microscopic traffic simulation model and investigated the nature of oversaturated systems and also under-saturated systems that might become congested because of poor signal timing and deficient spacing between the signalized intersections. The work provided additional technologies needed to develop an HCM chapter on interchanges in which oversaturated traffic demand conditions could be analyzed. These technologies included equations on minimum link travel speed during over-saturation, the average maximum link delay, the upper bounds on arterial delay, and a recommend on more advanced capacity and delay algorithms.

Messer [14] also presented extensions of work originally published by Prosser and Dunne in Australia for analyzing the operational impacts of queue spillback on the capacity and delay of closely-spaced signalized intersections. The extended model (PDX) was coded and tested by a simulation program. The author concluded that the PD model provided the first clear description of a practical method for analyzing the critical operational aspects of problems at congested closely-spaced intersections, and the PDX model was an innovative and enhanced software implementation of the general principles of PD. The author recommended implementing the queue spillback and resulting flow impediment models into the HCS and other internationally recognized signal timing optimization program.

As signalization software designed specifically for diamond interchanges, PASSER III has some limits. PASSER III is designed for under-saturated conditions and is not capable of modeling queue spillback conditions. Kovvali et al. [15] introduced the Arterial Signal Coordination Software (ASCS) to timing diamond interchanges in under-saturated and oversaturated conditional. This software used genetic algorithm to optimize signal timings, and its analysis model applied horizontal stacking of queues and shock wave analysis to estimate the performance of traffic operations. The author compared ASCS with PASSER III, and concluded that ASCS outperformed PASSER III when queue spillback occurred.

Tian et al. [16] [45] provided an integrated operations of a diamond interchange and a ramp metering system. This research concentrated on the impact of potential queue spillback from ramp metering signal to the diamond interchange signal. The study focused on the two common diamond-phasing schemes: basic three-phase and TTI-4 phase. Through implementation of special signal timings at the diamond interchange, the traffic flows feeding the ramp meters could be controlled and thus minimize ramp queues. The simulation results showed that the integrated operations were most effective under a medium traffic demand condition.

Tian et al. [19] applied the standard TTI-4 phase scheme to a site consisting of six (three pairs) closely-spaced intersections aiming at maximum progression between the closely-spaced paired signals. The simulation indicated significant reductions in the number of stops.

All the above research was concentrated on optimizing pre-timed signal control scheme at diamond interchanges, and they made great efforts to address the impact of queue spillback onto diamond interchanges. However, none of them took the real-time traffic flow into consideration, and so far all the software packages were designed for off-line optimization.

Fang and Elefteriadou [20] took the real-time traffic fluctuations at diamond interchanges into consideration and developed an optimization methodology for adaptive traffic signal control at diamond interchanges. They used the forward dynamic programming (DP) method to make the phase sequencing decision and phase duration that minimize a pre-specified performance measure over a finite forward-rolling horizon. The optimization process was based on the advanced vehicle information obtained from loop detectors installed a certain distance back from the stop line. The vehicle information included the number of vehicles passing the detectors and their speeds. Vehicle trajectories from detections to future arrivals and departures were modeled at the microscopic level to estimate the traffic flows at the stop-line for each horizon. The author compared the performance of the real-time DP algorithm and the pre-timed signal plan derived by PASSER III and TRANSYT-7F. The simulation results exhibited that the DP algorithm was superior to PASSER III and TRANSYT-7F in handling demand fluctuations for medium to high flow scenarios when the field demand was increased from the one used in off-line optimization. The performance of the three algorithms was almost identical if the simulation demand was similar to off-line demand situation and did not vary much.

Fang and Elefteriadou's research indicates that adaptive traffic signal control schemes work better than pre-timed signal optimization method since the adaptive control is capable of responding to fluctuating traffic flow and optimize signal operation on-line.

Adaptive Signal Control Strategies

In addition to the conventional fixed-timed and actuated signal control approaches, there is a traffic signal control strategy that is referred as "adaptive signal control". One definition [21] is that "adaptive signal control represents a class of signal control strategies that has the following three characteristics: a) It uses detectors placed upstream of the intersection for early detection of the arrival of vehicles, b) It utilizes that advance arrival information obtained by the detectors as a primary basis to determine and implement the optimal signal switching sequence on a real-time basis, and c) If predicted arrival data are used to supplement the arrival data obtained by the detectors, the prediction period extends in to the future only for a very short period of time." From this definition, we can see the substantial difference of adaptive signal control to the conventional signal control is that adaptive signal control is not an optimized control scheme based on off-line traffic, but rather response to the real-time fluctuating traffic and usual traffic situations.

Since the first try on developing an adaptive signal control in 1960's [22], traffic engineers and researchers have implemented many methods for adaptive signal control.

Rosdolsky [22] introduced a mathematical method for adaptive on-line signal program computation. The object was to minimize number of stops by preventing the interference of stationary queues and moving platoons. This was done by advancing green sufficiently to release the queue before the arrival of a platoon. The authors presented three adjacent one-way intersections with no turning movements. The detectors located at the departure side of intersections detected (1) the approaching platoons, and (2) the number of vehicles to be released before that platoons arrival. Mathematical algorithms were developed to estimate the arrival time of the platoon, the green time the platoon needed for the platoon to pass the intersection. This was an experiment to illustrate the application of certain mathematical techniques to traffic control.

In the 1970's and 1980's, several adaptive traffic control systems were developed, among which the most famous are SCAT, SCOOT, OPAC [23, 38, 39, 40] and RHODES [43] [25].

The Sydney Coordinated Adaptive Traffic (SCAT) system [23, 38], developed in Australia, is a totally coordinated adaptive urban traffic control system which contains a central supervisory, a series of remote regional computers, intersection computers, slave traffic signal controllers, and a communication system. In the SCAT system, an area for signal coordination is divided into smaller sub-areas of about one to ten signalized intersections that share a common cycle time. The common cycle time is updated every cycle in steps of up to 6s according to the degree of saturation of that sub-area. This cycle time is shared among various phases at each intersection according to a selected phase split plan. The offset plan within a sub-area may be, by default, one selected by an algorithm which may also be used to select an external offset for sub-system marriage, or, optionally, one which is tied to the phase split plans. This system offered a substantial improvement to movements on arterial roads in terms of delay, accident reduction, fuel consumption, air pollution, etc.

The Split, Cycle and Offset Optimization Technique (SCOOT) system [38] [39] was developed in England and has the largest market share of adaptive control software in the world wide. The objective of SCOOT is to minimize the sum of the average queue in an area. It uses real-time traffic data to model flow profiles of traffic arriving during each cycle. Based on the data, it predicts the queue size for different hypothetical changes in the signal timing parameters. A few seconds before every phase change, SCOOT used the flow profile to determine whether to delay, advance or leave it unaltered. In addition, for every cycle and every fixed minute, a similar question is asked to determine whether the offset should be advanced or delayed. Thus, SCOOT changes its timing parameters in fixed increments to optimize an explicit performance objective.

The Optimization Policies for Adaptive Control (OPAC) [40] [41] [42], sponsored by FHWA, is a fully adaptive distributed real time traffic control system. It continuously adapts signal timings to minimize total intersection delay and stops. OPAC calculates split and offset locally and provides real-time adjustments to signal timing parameters (split, cycle, and offset). OPAC provided a dual capability of distributed individual intersection control as well as coordinated control of intersections in a network.

Another FHWA developed adaptive system is RHODES [43] [25]. The system obtained traffic information from detectors, and optimized signal control (phase order, cycle, split, and offset) basing on measures of effectiveness (average delay, number of stops, throughput, etc). RHODES contained three levels of control - intersection control, network flow control, and network load control. At each level, there was an estimation/prediction component and a control component. The prediction component predicted future arrivals at the intersection by using the output of the detectors and the information on the traffic state and planned phase timings for the upstream signals. The system also estimated travel times on links, queue discharge rates, turning probabilities, and queues at the intersections and the ramps. At the network flow control level, it was needed to predict the network flow. At the intersection control level, RHODES used dynamic-programming (DP) based algorithm COP to optimize the phase sequence and splits. The network flow control logic optimized the movement of observed platoons in the sub-network to minimize an MOE. By simulation results, the RHODES was superior to semi-actuated control in terms of throughput, average delay.

Fehon [27] summarized the application of adaptive control in US up to 2004. The author introduced the successful installation of adaptive signals (e.g. SCAT and SCOOT) in other countries, and the FHWA sponsored research, RHODES and

OPAC. The author pointed out three main obstacles hold back traffic engineers in the United States from using adaptive signals. First, the traffic engineers either paid little attention to the issue or did not believe the claimed benefits of adaptive signals. The second and the third were the practical institutional and financial issues. The author suggested that US traffic engineers need a shift in attitude from the current signal control patterns and should be open minded to accept adaptive control system.

Owen and Stallard [29] developed a fuzzy rule-based approach to real-time distributed adaptive signal control.

Yu and Recker [30] provided an adaptive control model based on a discrete-time, stationary, Markove decision process. This model incorporated probabilistic forecasts of individual vehicle actuations at downstream detectors that are derived from a macroscopic link transfer function. The model was tested both on a typical isolated traffic intersection and a simple network comprised of five four-legged signalized intersections. Compared with full-actuated control, adaptive control model showed significant improvement over conventional full-actuated control.

Artificial Intelligence Approaches in Traffic Signal Control

Artificial intelligence (AI) is developed since World War II. AI involves electronics, mechanics, computer science, etc. It is the science and engineering of making intelligent machines, especially intelligent computer programs. The aim of computer programs is to create the intelligent capability to achieve goals in the world by simulate human intelligence. In the AI system, the intelligent agency is perceptive to the environment and can take actions to achieve the goal with the biggest chance. AI techniques have been applied in many areas including traffic and transportation engineering.

As to an intelligent traffic signal control system, it can improve traffic control with its capability of making pro-active decisions on the basis of temporal analysis and developments' ability of managing, learning, self-adjusting and responding to non-recurrent and unexpected events [44].

The following discusses applications of AI techniques to traffic signal control. The discussion emphasizes on expert systems (fuzzy logic), learning systems (artificial neural networks), and the combination of the above systems (neurofuzzy systems).

Fuzzy Logic in Traffic Signal Control

Traffic signal control is one of the oldest application areas of fuzzy sets in transportation engineering. The strength of fuzzy logic lies in its capability of simulating the decision-making process of a human that is often difficult to model using conventional mathematical methods. Fuzzy control has proven to be successful in problems for which an exact mathematical modeling is hard or impossible but an experienced human operator can control the process.

One basic advantage of fuzzy control is that it fires many fuzzy rules simultaneously and makes a decision with compromise. Wherefore, fuzzy control is suitable for problems with multiple conflicting objectives, such as minimization of delay between through movements and left turning movements. This trait is suitable for traffic signal control problems at an intersection where traffic flows from several approaches compete at the same time. Based on different priorities, signal control provides a desirable compromise between conflicting objectives, and assigns times to different traffic flows or movements.

Compared with conventional control, another aspect of fuzzy control worth examination is its robustness and adaptivity. The conventional signal control requires the setting of a large number of parameters, like minimum and maximum times of each signal groups and the logic of detectors. In the case of the fuzzy controller, the number of parameters can be reduced. In addition, the meaning of each parameter can be realized easily. This is possible due to the fusion of the membership function that covers a range and, as a result, the conclusions of the rules overlap. This fact makes more than one rule to fire for a given input, and the outcome is derived as a compromise of the conclusion of more than one rule.

To the best knowledge of the author, in 1977, Pappis and Mandani [1] published the first paper in which the traffic signal problem was solved using fuzzy logic. They used fuzzy logic at an isolated one-way road intersection without turning movements. The simulation results showed that the Fuzzy controller was better than a conventional vehicle-actuated controller in terms of average delay.

Based on Pappis and Mandani’s fuzzy controller concept, Nakatsuyama et al. [12] developed a fuzzy controller for two consecutive, one-way intersections without turning movements. In this model, phase lengths and offset were determined using fuzzy rules. They compared the fuzzy controller to a vehicle-actuated controller for different traffic volumes. The results indicated that the fuzzy controller had less average delay.

Tan et al. [5] designed and implemented a fuzzy controller at an isolated two-way intersection without turning movements. Using simulation models, they compared the fuzzy controller with a conventional fixed-time controller and found that the fuzzy controller had better performance and was more cost effective.

Favilla et al. [48] presented a fuzzy controller with adaptive strategies for fuzzy urban traffic control systems using two different defuzzification and decision-making criteria. The basic concept of adaptive strategies was to adjust the membership functions according to the traffic conditions in order to optimize the controller's performance. The simulation study showed that the adaptive strategies improve the efficiency of fuzzy controllers.

Chiu and Chand [24] further applied the fuzzy controller into a small network of intersections formed by six streets. In this paper, Chiu and Chand presented a distributed approach to traffic signal control, where the signal timing parameters at a given intersection were adjusted with respect to the local traffic conditions and the signal timing parameters at adjacent intersections. Thus, the signal timing parameters evolved dynamically using only local information to improve traffic flow. This distributed approach provided a fault-tolerant, highly responsive traffic management system. The author used fuzzy decision rules to adjust cycle, split and offset based only on local information (e.g., degree of saturation, traffic volume, etc). The amount of change in the timing parameters during each cycle was limited to a small fraction of the current parameters in order to ensure a smooth transition. Compared with fixed-time intersection, the simulation showed the effectiveness of this method in terms of waiting time and number of stops.

Kim [6] developed a fuzzy controller for an adaptive traffic management system to accommodate an intersection with variable traffic volumes. The fuzzy controller used variables of arrival, queue, and traffic volume and could alleviate traffic congestion. This method adaptively controlled the cycle of traffic signals even though the traffic volume varies. The effectiveness of this method was shown through simulation of a single intersection. The experimental results showed that the fuzzy controller outperformed the existing controllers in terms of number of passed vehicles, average delay of vehicles, and degree of saturation.

Since 1990's, the researchers in the Fuzzy Signal Control (FUSICO) project in Finland have been making a great contribution to the implementation of fuzzy logic to traffic control [2, 8, 33, and 35]. The objective of the FUSICO was the application of fuzzy control to traffic signals at the individual intersection level. In 1998, Niittymaki and Kikuchi [2] designed a fuzzy logic controlled pedestrian crossing signal. The controller was designed to emulate the decision process of an experienced crossing guard. The fuzzy controller found a compromise between two conflicting objectives: minimization of pedestrian delay and minimization of vehicular delay and number of stops. The evaluation was performed via simulations, showing the fuzzy controller performed equally well as or better than conventional demand actuated control. After that, Niittymaki [8] compared the differences in various traffic signal control algorithms when simulating two consequent one-way intersections with no turning traffic. The algorithms used in the comparison were coordinated fixed-time signal group control, vehicle actuated gap seeking non-coordinated signal group control, FUSICO control, and a combination of standard vehicle actuated control and a FUSICO-controller. The author found that the FUSICO-controller had better overall efficiency than conventional vehicle-actuated control. In 2000, Niittymaki [35] summarized the FUSICO project. The FUSICO included three models: (1) Traffic situation level with control policy varying with respect to different traffic situations. (2) Phase and sequence level (fuzzy phase selector), and (3) Green ending level or extension level (fuzzy green extender). The results of their past work indicated that fuzzy signal control could be the potential control method for isolated intersections. In 2001, Niittymaki [33] described the installation of a fuzzy signal controller at a real intersection. The performance of a vehicle-actuated control system with fuzzy-control system using microscopic simulation and the field implementation had been compared, indicating that the fuzzy control was very competitive against conventional vehicle-actuated control if traffic demand was high. Other important advantages included the simple algorithm structure, the savings of material costs and the low installation and maintenance costs. The above results showed that the fuzzy signal control could be installed in a real infrastructural environment and fuzzy algorithms could be more effective than conventional vehicle-actuated control.

Trabia et al. [3] designed a two-staged fuzzy logic traffic signal controller for an isolated intersection. The fuzzy controller used the vehicle loop detectors that were placed upstream of the intersection on each approach, to measure arriving flows and estimate queues. These data were used in a two-stage fuzzy logic procedure. In the first stage, observed arriving traffic flows were used to estimate relative traffic intensities in the competing approaches. These traffic intensities were then used in the second stage to determine whether the current signal phase should be extended or terminated at regular time intervals. The performance parameters were percentage of stopped vehicles and average delay per traffic cycle. The simulation results indicated that the two-stage fuzzy controller was better than a traffic-actuated controller for different traffic conditions on a four-approach intersection.

Lee et al. [34] developed a coordinated fuzzy controller for a set of intersections. The controller of an intersection controlled its own traffic and cooperated with its neighbor controllers. The controller received information from its traffic detectors and its neighbor controllers. Using this information, the fuzzy rule base system generated and displayed optimal signals. The phase sequences and phase lengths were managed adaptively to its traffic conditions and its neighbors' as well. To evaluate the controller performance, a simulator for intersections groups was developed. The method was compared with the vehicle actuated method. The simulation results showed that the fuzzy control performed better than conventional vehicle-actuated controllers in the cases of time-varying traffic patterns and heavy traffic conditions in terms of average delay.

The relevant studies were also conducted in Turkey in 2000's under the project INTAG-915. Murat [32] designed a new fuzzy traffic signal controller. In addition to fuzzy logic time controller, this controller considered a fuzzy logic phase sequencer. Performance of the fuzzy model was investigated by simulation studies and compared with vehicle actuated control method. The fuzzy model was superior in terms of average delay and number of stops. In 2005, Murat et al. [9] developed a Fuzzy Logic Multi-phased Signal Control (FLMuSiC) model for isolated signalized intersections. This model comprised of two systems which were based on fuzzy logic. One system arranged phase green times (duration) and the other one arranged phase sequences using traffic volumes. The developed FLMuSiC model was compared with the traffic-actuated control in terms of average delay. The comparison considered three and four phased controlling situations with equal and different traffic volumes on approaches of intersections. They found that the performance of the FLMuSiC was better than the vehicle actuated control, especially in the case of the higher volume variability. In addition, the FLMuSiC model was compared with earlier studies and encouraging results were reported.

Chou et al. [31] developed a more practical controller - fuzzy logic based traffic junction signal controller (FTJSC). The authors simulated an environment that considered the number of consecutive junctions, the number of lanes, the lengths of vehicles, and the lengths of streets. Compared to the fuzzy controllers in Mamdani's [1], Favilla's [48], and Nakatsuyama's [12] fuzzy controllers, the FTJSC had several advantages including applicability to any number of junctions, integrating every junction's status, requiring fewer control rules, needing fewer inference time, and taking street's distances into account. The simulations were conducted under different junction configurations and traffic conditions, and the results showed a good performance of the fuzzy controller in terms of the queue length and the average delay.

In order to control over-saturated intersections of two-way streets with left-turning movements, Zhang et al. [11] designed a fuzzy controller for oversaturated intersections. They compared the fuzzy control strategy with pre-timed and actuated control strategies using a typical intersection with varying traffic volume levels. In terms of delay, speed, percentage of stops, time in queue and throughput-to-demand ratio statistics, the fuzzy control strategy produced significant improvements over pre-timed and actuated control strategies under heavy traffic volumes.

Neurofuzzy Methods

In the researches before, the parameters as membership functions and fuzzy rules in the fuzzy controller were set up manually according to human knowledge. The fixed parameters limited the fuzzy controller's ability to accommodate changing traffic environment. In Favilla's research [48], the author found that adjusting membership functions according to the traffic conditions could optimize controller's performance.

There are various ways to adjust the parameters in fuzzy controllers, among which the neurofuzzy system is one way to fine tune parameters in the fuzzy controller. A "neurofuzzy" system is the combine of the neural network system and the fuzzy logic system, in which the fuzzy system is presented as a neural network. The neurofuzzy method has the advantage of the learning and adaptation ability of neural network and the decision-making ability of fuzzy controller.

After Favilla's work, Niittymaki [4] designed the fuzzy controller for situations involving multiple approaches and vehicle movements. The author found that the fuzzy control offered at least equal or better performance than the conventional vehicle-actuated control. The experiences and results of the field test and the calibration of membership functions with neural networks had been promising.

Patel and Ranganathan [26] integrated the artificial neural networks (ANN) and fuzzy expert system (FES) for signal control. In this system, ANN was used to model the traffic behavior and predict the traffic flow, and then FES received the predicted traffic flow and computed the cycle-time adjustment value. In this way, the FES was able to adjust itself continuously according to the dynamically changing traffic patterns during 24h of a day without having to change the fuzzy rules or membership functions of the input. The simulation results showed that the ANN+FES system performed better than the single ANN approach and the single FES approach in term of average wait time, and had lower cost (higher correct decision rate, less number of nodes compared to ANN approach).

Bingham [37] discussed the use of reinforcement learning in neurofuzzy traffic signal control. The author used the reinforcement learning algorithm of a neural network to adjust the fuzzy controller by fine-tuning the form and location of the membership functions. The neurofuzzy traffic signal controller was applied at a one-way, two-phase signalized intersection without turning movements. The author studied the neurofuzzy controller under different traffic volumes, and detector locations using the simulation software HUTSIM. The author compared simulation results between the fuzzy controllers after and before the reinforcement learning. The simulation experiments indicated that the learning algorithm was successful at constant traffic volumes.

Choy et al. [7] developed a multi-agent architecture for real-time coordinated signal control in an urban traffic network. The multi-agent architecture consisted of three hierarchical layers of controller agents: intersection layer, zone layer, and regional layer controllers. Each controller agent was implemented by applying fuzzy logic, neural network, and evolutionary algorithm. From the fuzzy rule base, each individual controller agent recommended an appropriate signal policy at the end of each signal phase. These policies were later processed in a policy repository before being selected and implemented into the traffic network. To handle the changing dynamics of the complex traffic processes within the network, an online reinforcement learning module was used to update the knowledge base and inference rules of the agents. This concept of a multi-agent system with online reinforcement learning was implemented in a network consisting of 25 signalized intersections in a microscopic traffic simulator. They found that the multi-agent system improved average delay and total vehicle stoppage time, compared with the fixed-time traffic signal control.

Summary

During the past decades, research efforts have been devoted to improved traffic control using many techniques, some have been applied to closely-spaced intersections. The previous researches focused on phase combination and sequence based on pre-timed control scheme at closely-spaced intersections, such as the Texas 3-phase and the TTI-4 control scheme. Adaptive signal control, including fuzzy logic method, provides better response to real-time fluctuation of traffic flow comparing to vehicle-actuated traffic signal controls. Bingham used reinforcement learning method of neural network to improve the performance of the fuzzy controller. This control method was applied only to an isolated one-way, two-phase signalized intersection without turning movements. The intersections in reality are more complex than that. Bingham's method provided a good reference to improve fuzzy signal control. The use of adaptive signal control method to address problems at closely space intersections is an area that has not been extensively studied. Research is still required to find ways to improve efficiency at closely-spaced intersections.

CHAPTER III

DEVELOPMENT OF THE NEUROFUZZY CONTROL SYSTEM

This chapter presents a new model for traffic signal control at closely-spaced intersections. This research focuses on developing a method to address fluctuating times in traffic signal cycles and improve the traffic operation at closely-spaced intersections.

This research utilizes fuzzy logic and its application concepts to design major components of the signal control algorithm. The fuzzy control model is calibrated using reinforcement learning algorithm with the neural network.

This chapter is divided into seven sections. The first four sections present the basic concepts of fuzzy logic, neural network, neurofuzzy, and reinforcement learning with their applications related to this research. The fifth section presents the framework of the neurofuzzy traffic signal control system. The sixth section describes the development of the neurofuzzy controller. The seventh section discusses the calibration of the fuzzy control model using reinforcement learning algorithm with the neural network.

Basic Concepts of Fuzzy Logic Theory

As a set theory based on artificial intelligence methods, Fuzzy logic was introduced in 1960 by Lutfi Askerzade at the University of California – Berkeley. Fuzzy method is useful for multi-object and multi-constraint decision situations in which the objectives and constraints are approximate.

The following introduces basic concepts of fuzzy set theory, fuzzy logic and the fuzzy control process.

Fuzzy Sets Theory

The concept of fuzzy set theory is the extension of the classical set theory. The classical set theory is built on the fundamental concept of "set". In the classical set theory, an individual either belongs to or not belong to a specified set, i.e., the answer is either of "yes" or "no". The fuzzy set theory, rather than defining an individual with a crisp description, is based on graded concepts to handle uncertainty and imprecision in a particular domain of knowledge. It means that the transition from "belong to a set" ($x \in A$) to "not belong to a set" ($x \notin A$) is gradual rather than crisp. The graded concepts are useful since real situations are not very often crisp and deterministic, and they cannot be described precisely [54]. For example, let "S" bet the set of temperatures, and consider its subset "s is cold". The definition of "cold" is vague. And hereby, fuzzy set theory is used to describe the uncertain cases.

1. Membership function

A fuzzy set is completely characterized by its membership function. Using the membership function we can specify a fuzzy set as follows:

$\mu_A : X \rightarrow [0, \quad 1]$, or $A : X \rightarrow [0, \quad 1]$.

A fuzzy set A in X is directly specified by the function $\mu_A(x)$ (or $A(x)$) or indirectly by a set of ordered pairs $(x, \mu_A(x))$ (or $(x, A(x))$) where $\mu_A(x)$ (or $A(x)$) represents the value of the "grade of membership" of x in A:

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

The value range of $\mu_A(x)$ is from 0 (totally not belong to) to 1 (totally belongs to). There are many membership functions, among which the most commonly used membership functions are triangular, trapezoidal, Gaussian, generalized bell, and sigmoid membership function, see Figure 4.

a) Triangular membership function:

$$\mu_A(x) = \begin{cases} 0, & x \le p1, \\ \dfrac{x - p1}{p2 - p1}, & p1 < x \le p2, \\ \dfrac{p3 - x}{p3 - p2}, & p2 < x \le p3, \\ 0, & p3 < x. \end{cases}$$

b) Trapezoidal membership function:

$$\mu_A(x) = \begin{cases} 0, & x \le p1, \\ \dfrac{x - p1}{p2 - p1}, & p1 < x \le p2, \\ 1, & p2 < x \le p3, \\ \dfrac{p4 - x}{p4 - p3}, & p3 < x \le p4, \\ 0, & p4 < x. \end{cases}$$

c) Gaussian membership function:

$$\mu_A(x) = e^{-\frac{(x-m)^2}{2\sigma^2}}$$

d) Generalized bell membership function:

$$\mu_A(x) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}$$

e) Sigmoid membership function:

$$\mu_A(x) = \frac{1}{1 + e^{-a(x-c)}}$$

2. Operations on Fuzzy Sets

The operations of fuzzy set include complement ($^{-}$, NOT), intersection ($\cap$, AND) and Union ($\cup$, OR).

a) The complement operation (NOT) of a fuzzy set is:

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x)$$

b) The commonly used intersection operations (AND) of a fuzzy set are:

Minimum: $\mu_s = \min\{\mu_{s1}, \quad \ldots, \quad \mu_{sn}\}$

Product: $\mu_s = \mu_{s1} \times \quad \ldots \times \mu_{sn} = \prod_{i=1}^{n} \mu_{si}$

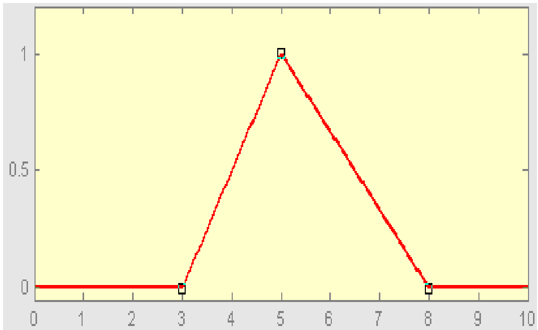
c) The commonly used Union operations (OR) of a fuzzy set are:

Maximum: $\mu_s = \max\{\mu_{s1}, \quad \ldots, \quad \mu_{sn}\}$

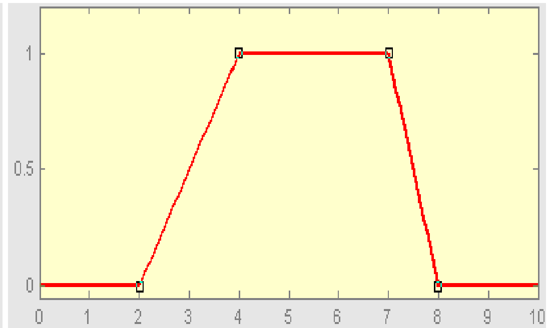
Sum: $\mu_s = \mu_{s1} + \quad \ldots + \mu_{sn} = \sum_{i=1}^{n} \mu_{si}$

Bounded sum: $\mu_s = \min\{1, \quad \sum_{i=1}^{n} \mu_{si}\}$

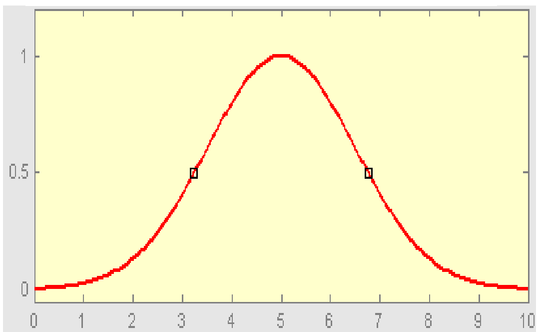
Probabilistic sum: $\mu_s = 1 - \prod_{i}^{n} (1 - \mu_{si})$



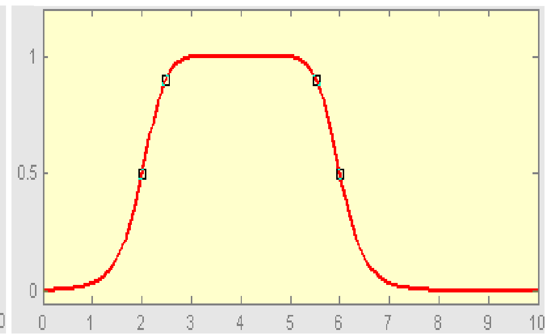
(1) Triangular membership



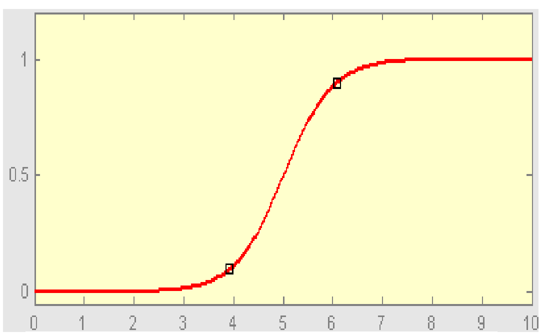
(2) Trapezoidal membership



(3) Gaussian membership



(4) Generalized Bell membership



(5) Sigmoidal membership

Figure 4. Membership functions.

Fuzzy Logic

Logic refers to the study of methods and principles of human reasoning. Similar to the classical reasoning process, fuzzy logic is to provide a foundation for approximate reasoning using imprecise propositions based on fuzzy set theory [54]. Recall the classical reasoning process is as the following:

If less than 40 F^o is cold and today's temperature is 41 F^o, then today is not cold. While this logic this is not reasonable because people can hardly feel the difference between 40 F^o and 41 F^o.

Fuzzy logic inference rules deal with imprecise logic which cannot be handle by the classical (precise) reasoning using two-valued (true or false) logic:

(i) it is cold if temperature is below 40 F^o;

it is warm if temperature is between 40 F^o and 70 F^o;

it is hot if the temperature is above 70 F^o;

(ii) Today the temperature is 41 F^o .

(iii) Today is warm but a little cold.

Fuzzy inference is an inference process based on multi-value logic: the truth values of input and the rules of the inference process are not singular (yes or no) but rather they are multi-values. The essence of this inference is the use of fuzzy sets for the representation of input and rules (relations).

Fuzzy Control System

Fuzzy control theory is a new alternative band branch of control system comparing to conventional/classical control system. This control system is developed for solving real-world problems in imprecise system which cannot be adequately managed by conventional control theories and techniques [54].

The fuzzy set theory, fuzzy logic, fuzzy control, etc. are all man-made and subjectively introduced to the scene.

The fuzzy control process consists of five steps [52]:

(1) Fuzzification of the input. The initial input is of true value. The true value of input should be converted to fuzzy linguistic descriptions (e.g. “cool”, ”warm”) with memberships before being applied into the “if…then…” fuzzy rules.

(2) Application of the fuzzy operator (AND or OR). If two and more fuzzy input variables be connected with “and” or “or” in the antecedent part of the “if…then…”fuzzy rules, the memberships should be calculated by the fuzzy operator.

(3) Implication from the antecedent to the consequent (Inference system). The fuzzy operator result (from step 2) is the firing strength of this rule, i.e., the membership of the linguistic fuzzy output. The firing strength means how strong this fuzzy rule is applied, i.e., the degree of validity of the conclusion.

(4) Aggregate all output values. Because of the overlapping fuzzy input memberships, there are usually more than one fuzzy rule is “fired” and more than one fuzzy output is derived. The fuzzy control system aggregates all output values as a union to determine the final control action.

(5) Defuzzify. Although the calculation process is based on fuzzy theory, the final action of the fuzzy control system is not vague. In other words, all the linguistic fuzzy output values should be converted to a deterministic value/action as the

final decision of the fuzzy control system. An example of a fuzzy inference process is shown in Figure 5.

1. Inference system

Fuzzy control has been developed in the context of fuzzy inference. The inference system consists of k linguistic control rules. The general format for the k_{th} rule is:

R_k: If {x_1 is $A_{1,k}$} and/or… and/or {x_i is $A_{i,k}$} and/or … and/or {x_n is $A_{n,k}$} then {y_k is B_{jk}}

where,

x_i = Fuzzy input;

$A_{i,k}$ = Linguistic description of the fuzzy input x_i;

y_k = Fuzzy output of the fuzzy rule k;

B_{jk} = Linguistic description of the fuzzy output y_k.

Under the fuzzy inference, the conclusion is drawn based on the similarity between the input (x) and the premises (A). An exact match is not necessary. The degree of similarity between them determines the degree of validity of the conclusion. The degree of validity of the conclusion is calculated by applying fuzzy operator on all fuzzy input variables.

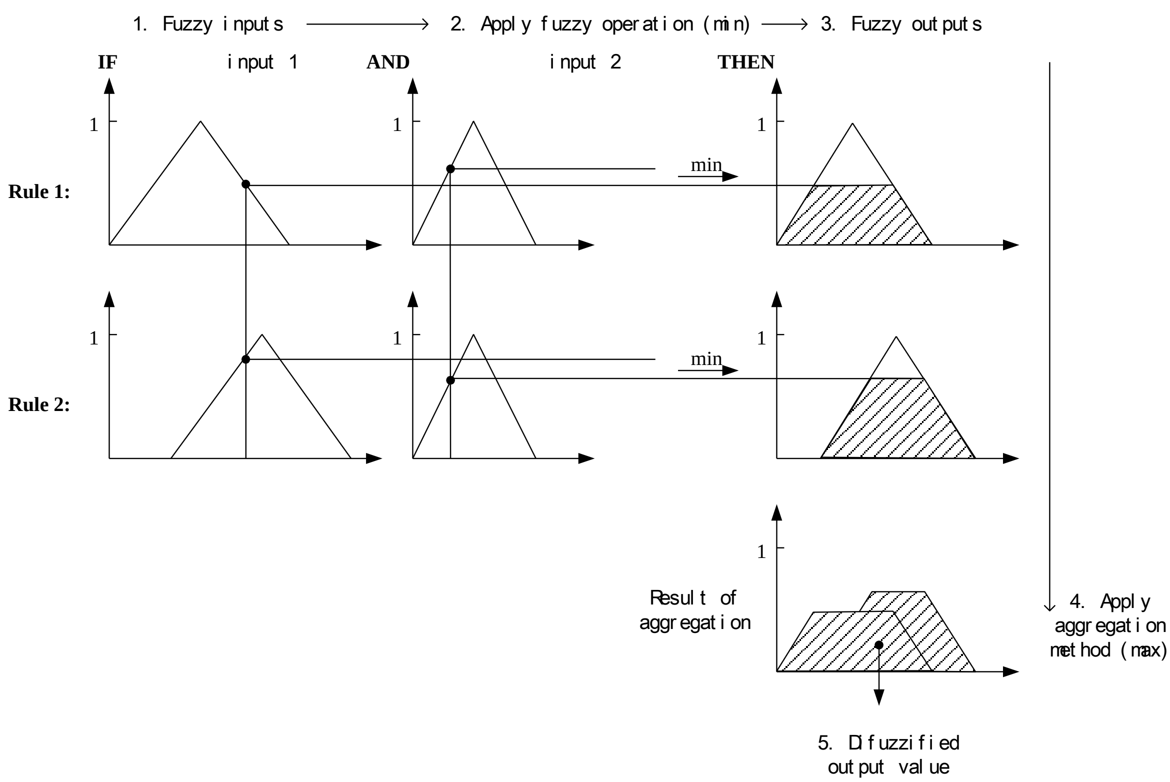


Figure 5. Fuzzy control process (max-min controller).

2. Defuzzification

In order to determine a particular action (such as whether to terminate or continue a signal phase), we still need an operation to pinpoint the specific action because the final outcome still has to be binary (yes or no). This process is called defuzzification. Some methods for defuzzifying a union of several membership functions are as follows:

a) Center of Area (COA) defuzzification

The center of area defuzzification method is to calculate the centroid of the fuzzy sets, see Figure 6.

$$y^* = \frac{\int y\mu(y)dy}{\int \mu(y)dy}$$

where,

y^* = defuzzified control action;

y = fuzzy output;

$\mu(y)$ = membership function of y.

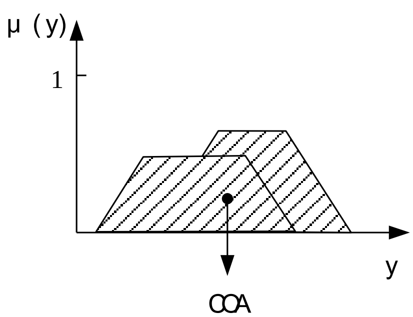


Figure 6. Center of Area method.

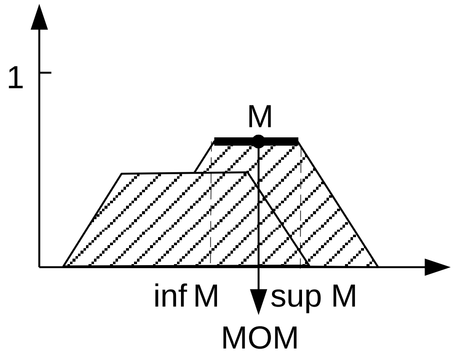


Figure 7. Mean of Maximum method.

b) Mean of Maximum (MOM) defuzzification

The mean of maximum defuzzification method calculates the average of the smallest value and the largest value of y_i for which $\mu(y_i)$ reaches its maximum, see Figure 7.

$$y^* = \frac{\inf M + \sup M}{2}$$

where,

M =the value of y for which $\mu(y)$ reaches its maximum.

$\inf M$ =smallest value of M;

$\sup M$ =largest value of M.

c) Local Center of Area (LCOA) defuzzification

The local center of area defuzzification method calculates the centroid of each output fuzzy set separately, i.e., locally defuzzify the output of each individual rule using COA method, and then calculates the weighted mean of the COAs.

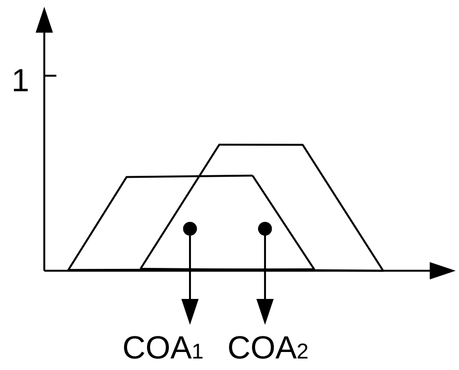


Figure 8. Local Center of Area (LCOA).

A small example of LCOA defuzzification method is as shown Figure 8. The final defuzzified action is the weighted mean of all local COAs:

$$y^* = \frac{\omega_1 COA_1 + \omega_2 COA_2}{\omega_1 + \omega_2}$$

where, ω_1 and ω_2 are weights of each fuzzy rule.

For a triangular membership function, the local COA is simply the average of the three points - p1, p2 and p3, see Figure 9 (left).

$$y = \frac{1}{3}(p1 + p2 + p3)$$

Similarly, the local COA of a trapezoidal membership function (Figure 9, right) gives:

$$y = \frac{1}{3}\frac{p_1^2 + p_2^2 + p_1 p_2 - p_3^2 - p_4^2 - p_3 p_4}{p_1 + p_2 - p_3 - p_4}$$

The final decision of the LCOA method is the weighted average of all rule output values.

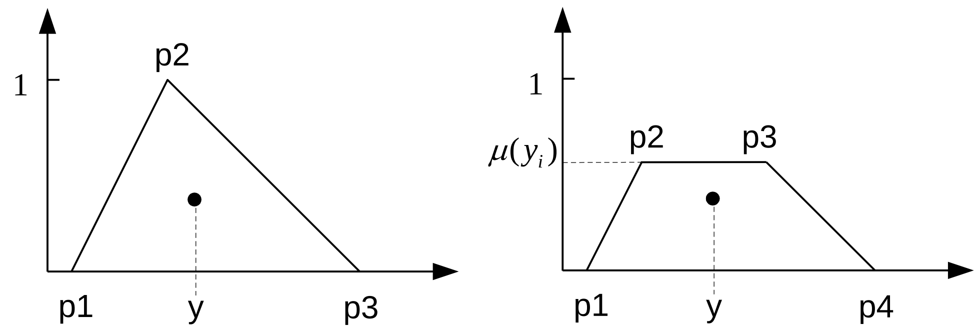


Figure 9. The Center of Area for a triangular membership function and a trapezoidal membership function.

$$y^* = \frac{\sum_{i=1}^{k} m_i \mu(y_i) V_i y_i}{\sum_{i=1}^{k} m_i \mu(y_i) V_i}$$

where,

y_i =the local COA of the membership function of rule i;

$\mu(y_i)$ =membership of the fuzzy output of rule i;

m_i =the weight of rule i;

V_i =the volume of the consequent set to which y_i belongs to.

d) Local Mean of Maximum (LMOM) defuzzification

The local mean of maximum defuzzification method locally defuzzify the output using MOM method, then calculates the weighted average of the rule output values. A small example is as shown Figure 10, the defuzzified final action is:

$$y^* = \frac{\omega_1 MOM_1 + \omega_2 MOM_2}{\omega_1 + \omega_2}$$

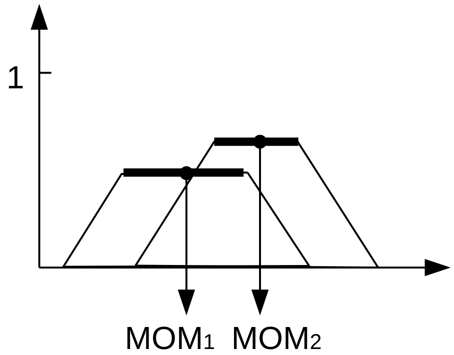


Figure 10. Local Mean of Maximum method.

To using the LMOM method, the first step is to locally defuzzify the output of each individual rule. Figure 11 illustrates the calculation of a triangular membership function. For a triangular membership function, it is truncated at level $\mu(y_i)$, the local MOM is:

$$
\begin{aligned}
y_i^* &= \frac{\inf M_i + \sup M_i}{2} \\
&= \frac{[p1 + (p2 - p1)\cdot \mu(y_i)] + [p2 + (p3 - p2)\cdot(1 - \mu(y_i))]}{2} \\
&= \frac{1}{2}(1 - \mu(y_i))(p1 + p3) + \mu(y_i)\cdot p2
\end{aligned}
$$

where,

y_i^* =the defuzzified output of rule i;

$\inf M_i$ =the smallest value of the fuzzy set y_i;

$\sup M_i$ =the largest value of the fuzzy set y_i;

$\mu(yi)$ = fire strength of the rule i, i.e., membership of the output of rule i.

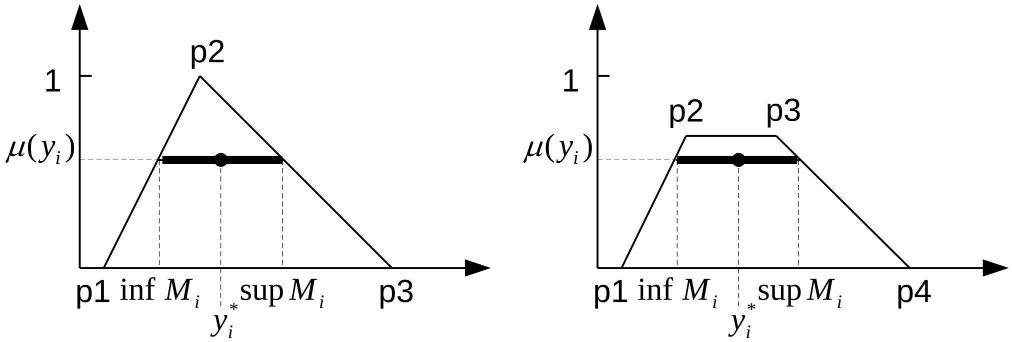


Figure 11. The LMOM method applied on a triangle membership function and a trapezoidal membership function.

Similarly, for a trapezoidal membership function truncated at level $\mu(y_i)$, shown in Figure 11 (right), the LMOM gives

$$y_i^* = \frac{1}{2}(1-\mu(y_i))(p1+p4) + \frac{1}{2}\mu(y_i)(p2+p3)$$

The final fuzzy decision of the LMOM method is the weighted average of all defuzzified output values.

$$y^* = \frac{\sum_{i=1}^{k} m_i \mu(y_i) y_i^*}{\sum_{i=1}^{k} m_i \mu(y_i)}$$

where m_i is the priority weight for each fuzzy rule. If no priority weight put on fuzzy rules, then,

$$y^* = \frac{\sum_{i=1}^{k} \mu(y_i) y_i^*}{\sum_{i=1}^{k} \mu(y_i)}$$

From all the four defuzzification method discussed above, we can see that the COA and MOM methods aggregate all output values and defuzzify output as a union, while the LCOA and LMOM methods defuzzify the output of each rule individually, and then calculate the weighted average of the already defuzzified output. It has to be pointed out that the defuzzification methods are not logic based but rather practical based depending on the specific circumstance.

Basic Concepts of Neural Network

Artificial neural network (ANN) is a system whose structure is inspired by the action of the nervous system and the human brain. Neural networks are composed of simple elements operating in parallel [53].

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

Other advantages include:

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.

- Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Due to its many advantages, ANN has been implemented in many industry and research areas covering signal processing, control, robotics, pattern recognition, medicine, speech production, vision, business, financial applications, data compression, and game playing, etc.

There are many types of neural networks including single layer NN, multiple layer NN, recurrent NN, etc. The multi-layer feedforward neural network is introduced here because this type of neural network is used in this research.

The structure of the multi-layer feedforward network with one hidden layer is as shown in Figure 12. The layers additional to the input and output layers is called “hidden layers” which is not connected externally. The network topology is constrained to be feedforward: generally connections are allowed from the input layer to the first (and possibly only) hidden layer; from the first hidden layer to the second,..., and from the last hidden layer to the output layer. The following context describes the structure, and the training process in detail.

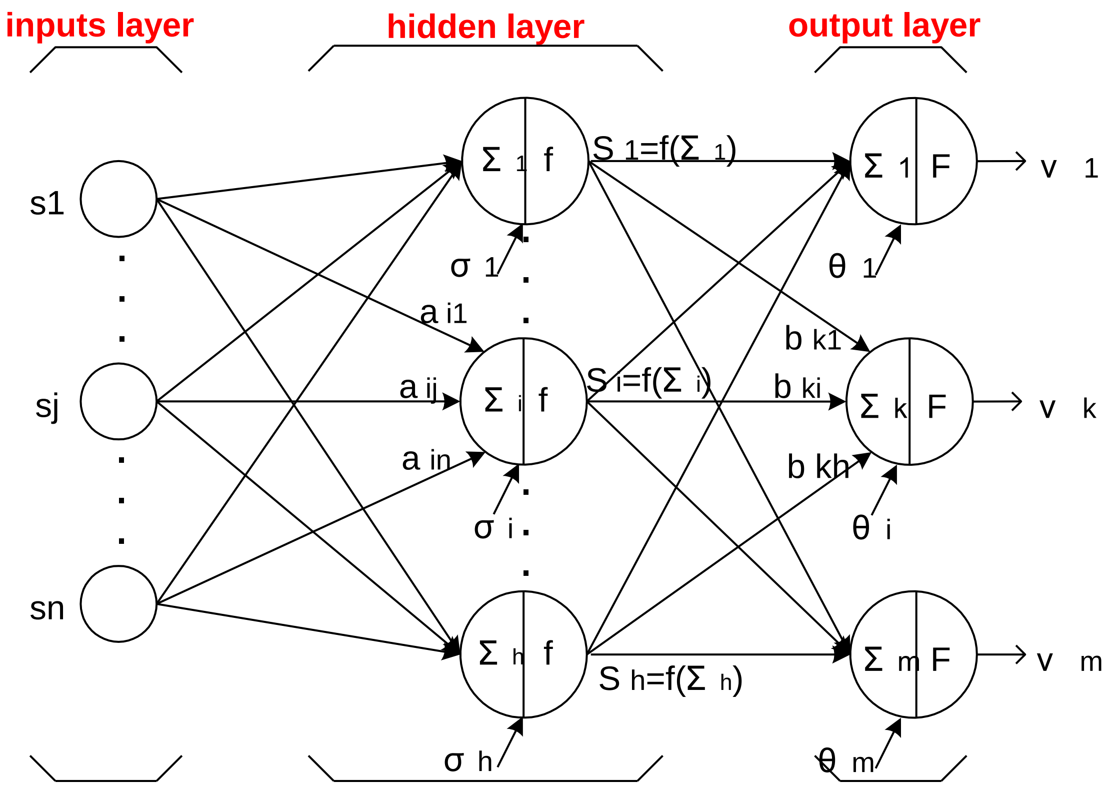


Figure 12. The structure of a neural network with one hidden layer.

In Figure 12, $s_1 ... s_j ... s_n$ are input signals transmitted to the neuron where signals are modified according to the connection weights. The neuron in the hidden layer is activated by a transfer function f on the weighted sum of input signals. The output of the i_{th} neuron is s_i.

For the neuron i in the hidden layer, the weighted sum of input variables is the liner combination of all input signals with their connection weights, as shown in the following formula:

$$\Sigma_i = s_1 a_{i1} + s_2 a_{i2} + ... + s_j a_{ij} + ... + s_n a_{in} + \sigma_i = \sum_{j=1}^{n} s_j a_{ij} + \sigma_i$$

Then based on the transfer function f, the neuron i produces an output s_i, i.e.,

$$s_i = f(\Sigma_i)$$

For all neurons in the hidden layer, the weighted input is a vector:

$$S = (s_1 \quad s_2 \quad ... \quad s_h)$$

Then based on the threshold function *F*, the system produces output υ_k:

$$\upsilon_k = F(\sum_{i=1}^{h} s_i b_{ki} + \theta_k)$$

The hidden layer and the output layer also include the constant input σ_i and θ_k, which are used to propagate the error backward during the neural network learning process.

There are many transfer functions, such as hard-limit functions, linear functions, log-sigmoid function, etc., see Figure 13.

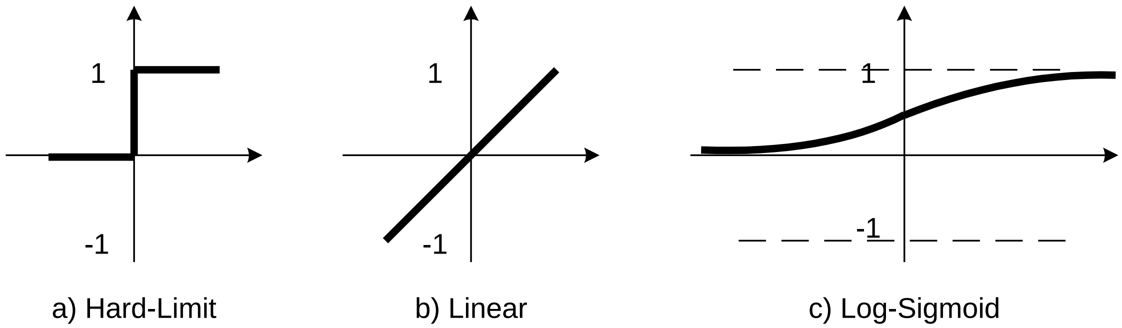


Figure 13. Transfer functions.

a) Hard-Limit Transfer Function: $f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$

The neuron produces a 1 if the net input into the transfer function is equal to or greater than 0; otherwise it produces a 0.

b) Linear Transfer Function: $f(x) = x$

The linear transfer function calculates the neuron's output by simply returning the value passed to it.

c) Log-Sigmoid Transfer Function: $f(x) = (1 + e^{-x})^{-1}$

The Log-Sigmoid Transfer Function generates output between 0 and 1 as the neuron's net input goes from negative to positive infinity. In the hidden layer of a neural network, a differentiable transfer function, such as the Log-Sigmoid Transfer Function, is selected in order to back-propagate errors in training process.

After the neural network constructed and the network weights initialized, the neural network is ready to be trained. The training process needs a series of sample input and corresponding target output. This type of training is called "supervised" learning.

The training process includes a forward pass and the backward pass. Firstly, in the forward pass process, the output is calculated and the error at the output units calculated. Secondly, in the backward pass process, the output unit error is used to alter weights on the output units. Then the error at the hidden nodes is calculated (by back-propagating the error at the output units through the weights), and the weights on the hidden nodes altered using these values. For each data pair to be learned a forward pass and backwards pass is performed. This is repeated over and over again until the error is at a low enough level (or we give up).

The error function which measures the difference between the network output and the target output are:

$$
\begin{aligned}
E &= \frac{1}{2}(V-D)^2 \\
&= \frac{1}{2}\sum_{k=1}^{m}(v_k - d_k)^2 \\
&= \frac{1}{2}\sum_{k=1}^{m}\Big(\sum_{i=1}^{h} b_{ki} S_i - d_k\Big)^2 \\
&= \frac{1}{2}\sum_{k=1}^{m}\Big(\sum_{i=1}^{h} b_{ki}\, f\big(\sum_{j=1}^{n} s_j a_{ij}\big) - d_k\Big)^2
\end{aligned}
$$

where, V and D are the neural network output vector and target vector, other symbol are as shown in Figure 12.

The training function is used to determine how to adjust the weights to minimize performance. One training function is the gradient of the performance function, etc. If use this training algorithm, the gradient is determined using a technique called “back propagation”, which involves performing computations backward through the network. In the basic back propagation algorithm, the weights are moved in the direction of the negative gradient.

Using back propagation algorithm, the weights in the hidden layer in Figure 12 are updated as:

$$b(k+1) = b(k) - \eta \ g(k)$$

$$= b(k) - \eta \ \frac{\partial E}{\partial b}$$

$$= b(k) - \eta \ \frac{\partial E}{\partial v}\frac{\partial v}{\partial b}$$

where, g_k is the current gradient.

The weights in the first layer are updated as:

$$a(k+1) = a(k) - \eta \ \frac{\partial E}{\partial a}$$

$$= a(k) - \eta \ \frac{\partial E}{\partial v}\frac{\partial v}{\partial S}\frac{\partial S}{\partial a}$$

Basic Concepts of Neurofuzzy System

Neurofuzzy system is a combination of a neural network and a fuzzy control system. In a neurofuzzy system, the fuzzy control system uses linguistic reasoning while the neural network adjusts fuzzy membership functions or fuzzy rules. Hence, the combination of the two can overcome each other's disadvantages whereas keep their advantages [51].

The comparison between fuzzy method and neural network is shown in table 1.

Table 1. Comparison between the fuzzy method and the neural network.

	Fuzzy method	Neural Network
Input technology	Expert control	Algorithm
Information	Quantity or quality (Numerical or linguistic)	Quantity (Numerical)
Perceive	Decision-making (if…then…)	Perception
Reference scheme	Heuristic search	Parallel calculation
Calculation speed	Low	High
Error toleration	Low	Very high
Learning	By induction	By modifying weights
Flexibility	High	low

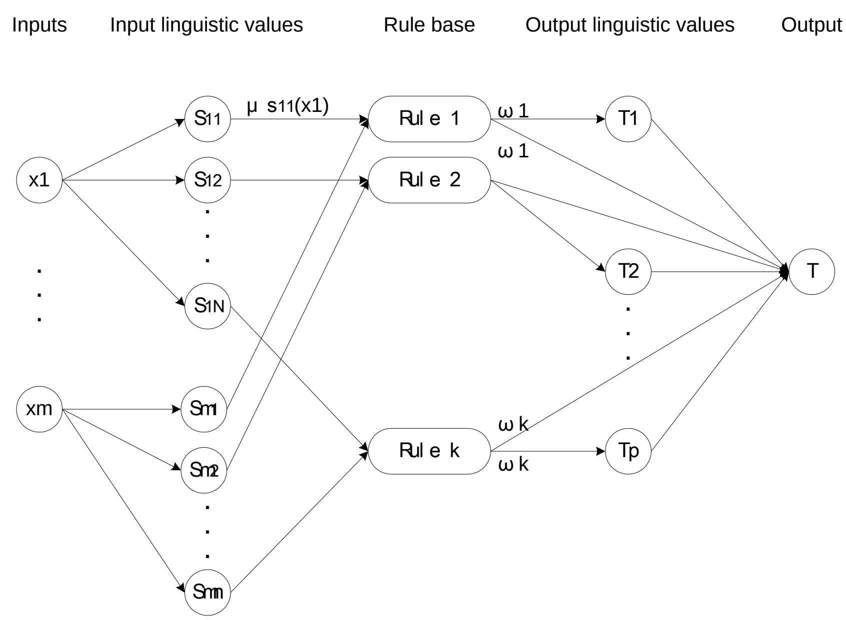


Figure 14. Fuzzy control system in neural network format.

In a neurofuzzy system, the fuzzy control system can be presented in the neural network format, see Figure 14. The real values of input are x_1, …, x_m. In the fuzzy control system, the values are converted into fuzzy linguistic values (such as "small", "medium", and "large") with relevant memberships $\mu_s(x_i)$. Then the linguistic values and memberships are sent to the fuzzy rule base. Based on the rule base, the fuzzy control system produces the linguistic output T_p (such as "long" and "short") and corresponding firing strength ω_k, i.e., the memberships of the output T_p. Then through defuzzifying output T_p, the fuzzy control system produces the final output T.

As mentioned before, in a neurofuzzy system, the neural network is used to adjust parameters in the fuzzy control system. The next section introduces the reinforcement learning method to adjust membership functions in the fuzzy control system.

Basic Concepts of Reinforcement Learning and GARIC model

Learning by interacting with our environment is probably the first to occur to us when we think about the nature of learning. Humans have no direct teachers, but we do have direct sensor-motor connection to the environment. We learn as we interact with environment which teaches us what “works” and what does not. Reinforcement learning (RL) is learning what to do – how to map situations to actions – so as to maximize a numeral reward signal [49].

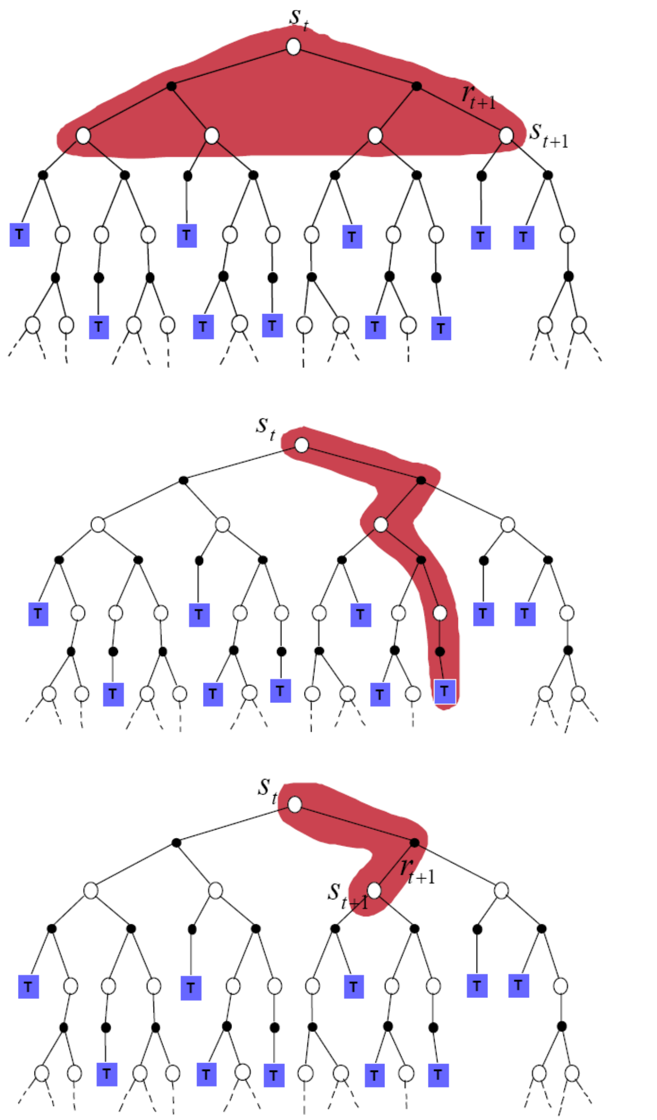
There are substantial differences between RL and both supervised and unsupervised learning. The supervised learning learns from labeled examples. It has a “teacher” which “tells” the model the “target” to achieve. The unsupervised learning does not have a “teacher” or “target”, and this type of algorithm clusters labeled examples. A typical example of unsupervised learning is pattern recognition.

RL learns from interacting with the environment which is defined by the problem. The “agent” senses its environment, produces actions that can affect the environment, and has a goal relating to its state. RL is essentially an optimization problem. There are two key components of the core of RL. One is trial-and-error, which means RL adapts internal representation based on experience to improve future performance. The other one is delay reward which means the actions are produced so as to yield long-term (not just short-term) rewards.

Beyond the agent and the environment, RL has the following four main elements:

- Policy. The policy is the learning agent's way of behavior at any given time. A policy is a mapping from perceived states of the environment to actions to be taken when in those states. The policy is usually stochastic (adapts as you go along), and is enough to determine the agent’s behavior.
- Reward function. The reward function is the goal in a RL learning problem. It maps each perceived state (or state-action pair) of the environment to a single number, a reward, indicating the intrinsic desirability of that state. Agent’s goal is to maximize the reward over time, and may be stochastic.
- Value function. Whereas a reward function indicates what is good in an immediate sense, a value function specifies what is good in the long run. Roughly speaking, the value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state. It allows the agent to look over the “horizon” that Actions are derived from value estimations, not rewards. We measure rewards, but we estimate and act upon values –corresponds to strategic/long-term thinking.
- Model: an observable entity that mimics the behavior of the environment. For example, given a state and action, the model might predict the resultant next state and next reward. Models are used for planning–any way of deciding on a course of action by considering possible future scenarios prior to them actually occurring.

There are many RL algorithms, such as Dynamic Programming, Monte Carlo Exhaustive search, Temporal Difference learning, etc. Dynamic Programming algorithms can be chosen when the model can provide description of all possibilities (of next states and rewards) and their probabilities. The Monte Carlo methods can solve RL problem without requiring a model of the environment, but need to wait until the end of the episode to update the value estimates. Temporal Difference methods do not require a model of the environment or all possibilities of next states and rewards, and does not need to wait until the end of the episode to update the value estimates. Temporal Difference methods use experience to solve the prediction problem. The idea is to have a “moving target”. The different learning processes are indicated in Figure 15 [49], where *S(t)* and *r(t)* represent the state and reward at time step t respectively.



Dynamic Programming

Monte Carlo

Temporal Difference

Figure 15. Learning process (decision tree) of Dynamic Programming, Monte Carlo and Temporal Difference methods

For the traffic signal control in this research, the traffic network system is complicated and it is impossible to identify all possibilities of next signal control state and their probabilities or define the end of the episode and there is no “end target” because the objective of using RL in this research is to reduce delay and stops but the target “minimum delay and stops” is unknown. So, Temporal Difference learning is chosen.

As one type of the Temporal Difference learning methods, the Actor-Critic methods offer a powerful framework for scalable RL systems. They are particular interesting since they operate inherently online, require minimal computation in order to select actions, and also, in Neural Networks it will be equivalent to a single feed-forward pass. The Actor-Critic methods can cope with non-Markovian environments (need not trace to the end of the decision tree) [49].

In the Actor-Critic method, the agency learns in both of the Actor and the Critic. Typically, the Critic is a state-value function. After each action selection, an evaluation error is obtained in the form of state-value function. If the function produces a positive error, which means the action improves the performance of the system, then, this action should be strengthened for the future. The Actor is the policy-making subsystem which chooses the optimal control action at each state. The architecture of the Actor-Critic Method is shown in Figure 16 [49].

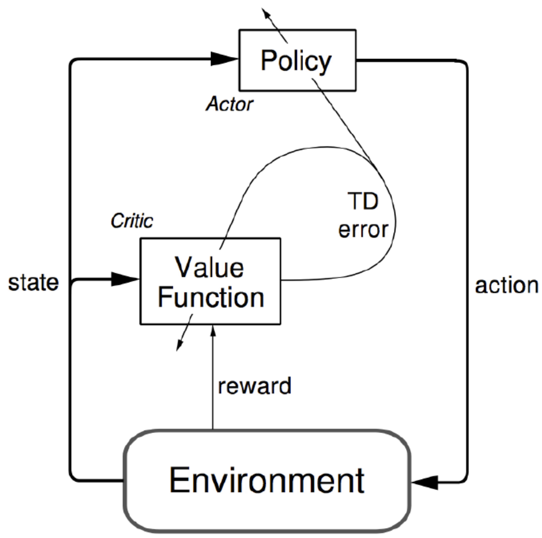


Figure 16. Actor-Critic method.

We can calibrate the membership functions using the neural networks or some other relevant systematic methods. Neural networks have recently been recognized as an important tool for constructing membership functions, constructing fuzzy inference rules, and other context-dependent entities in fuzzy set theory. Generalized Approximate Reasoning-based Intelligent Control (GARIC) model is introduced here because this method is used in the research. Firstly introduced by H. Berenji and P Khedkar in 1992 [50], GARIC is a neurofuzzy model that implements a fuzzy controller by using several specialized feedforward neural networks and the learning process is similar to the actor-critic reinforcement. In the neurofuzzy system, the fuzzy control system is the Actor who recommends actions via fuzzy rules; the neural network is the Critic who evaluates the system performance controlled by the fuzzy control system. The strength of this method is that it can learn and tune fuzzy controllers from a dynamic system even when only weak reinforcement is available. The disadvantage is the complexity of the calculation functions. The learning is achieved by integrating fuzzy inference into a feedforward neural network, which can then adaptively improve the performance of the physical system by using gradient descent methods. In H. Berenji and P Khedkar's report, GARIC method was applied to a cart-pole balancing system and demonstrated significant improvements. This method was adapted by E. Bingham into traffic signal control study [37]. But the method was applied to an isolated intersection of two one-way streets with no turning movements. In Bingham's study, there were only two phases considered for each intersection, which is not directly applicable to controlling a real-world intersection. In this research, the adapted GARIC method is used to control traffic signals at closely-spaced intersections where phases and traffic movements are more realistic.

The architecture of GARIC algorithm has three subsystems, see Figure 17.

- The action selection network (ASN), i.e., the Actor, who recommends an action "F" using fuzzy inference;
- The action evaluation network (AEN), i.e., the Crisis, who evaluates the performance of the physical system, and a good performance will be reinforced. It produces an internal reinforcement "$\hat{r}$".
- The stochastic action modifier (SAM), which uses "F" and "$\hat{r}$" to deviates the recommended action randomly, the deviated action F' is applied into the physical system in order to explore a better action.

The physical system receives the action F’, changes its state correspondingly, and then sends the state variables to both of ASN and AEN for the next step analysis. Meanwhile, AEN updates its weights, and recommends the updates of the fuzzy membership functions to ASN.

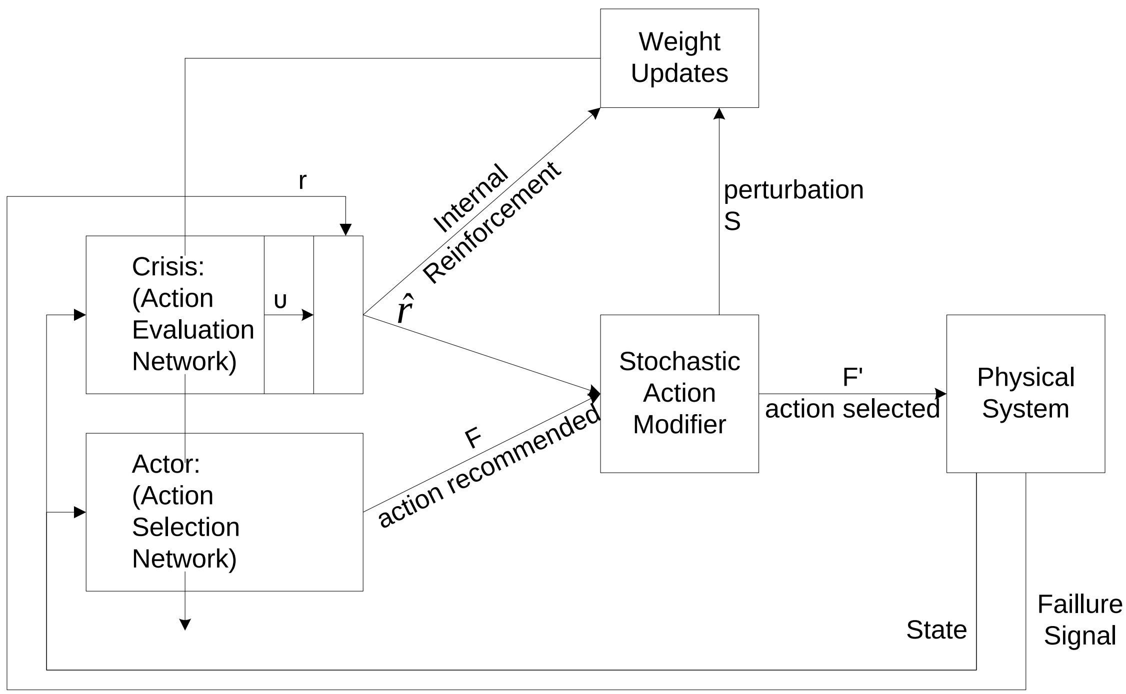


Figure 17. The architecture of GARIC.

Action Selection Network (ASN)

The Action Selection Network (ASN), i.e., the Actor, is a fuzzy control system represented in neural network form. ASN has five layers, see Figure 18. The first layer is the input layer; ASN receives the system mathematical state variables and sends them to the second layer, i.e., the “Antecedent” layer. In the second layer, ASN matches the mathematical input values with the membership functions, then, transforms the input variables into the appropriate linguistic descriptions (such as “the queue is short”) and corresponding memberships $\mu(x_i)$. Then in the third layer, ASN applies fuzzy rules, such as “If Queue #1 is medium and Queue #2 is long, then the Green Extension time is short”. The fourth layer consists of the output values from the third layer in the form of linguistic consequents (short) with the corresponding firing strengths (ω_r).

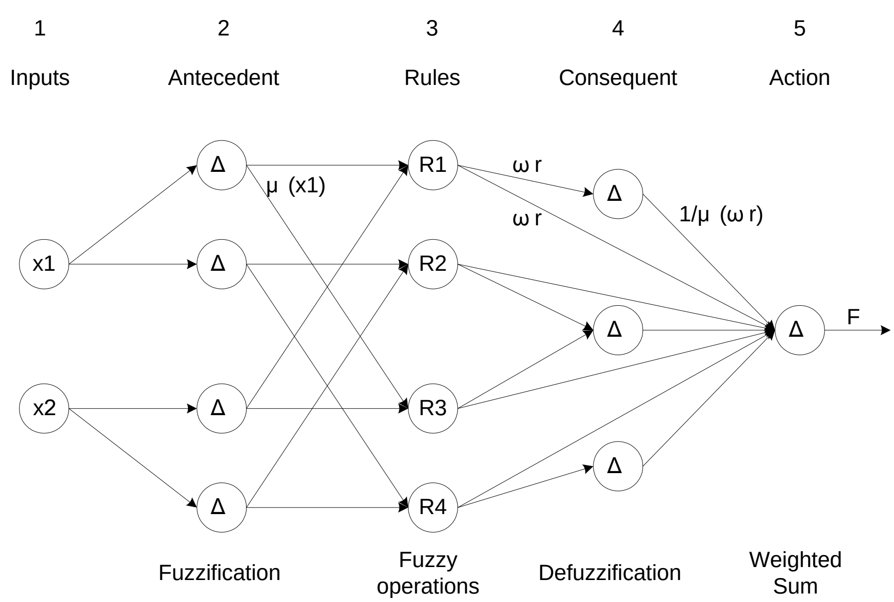


Figure 18. ASN – a fuzzy control system in neural network format.

In Berenji and Khedkar's report and Bingham's application in traffic signal control, the "softmin" method is used to do "AND" operation when apply fuzzy rules. The "min" method is not discussed because this method is not differentiable, which means this method can not "learn" in the dynamic system. In this research, the "product" method is used instead of "softmin". The reason is that the "product" method is better than the "softmin" here because the "product" method is differentiable through all membership function values in fuzzy rule and does not loss any information, whereas the "softmin" can not. The formula of the "product" operator is:

$$\omega_r = \prod_{i}^{\mathrm{m}} \mu(x_i),$$

where m is the number of rules been "fired", i goes through all "fired" rules.

In the fifth layer, all fuzzy rule output values are aggregated and defuzzified. The selection of the defuzzification method should fulfill two points: first, the method must be differentiable; second, the differentiated result is a polynomial in "x_i" (can not be a constant), which means it should be able to "learn" in the dynamic system. Berenji and Khedkar chose the LMOM method. Bingham chose the combined LMOM and LCOA method. As explained in Bingham's report, LCOA method can not be used at all, and for the LMOM and the combined LMOM and LCOA method, they can not be used on symmetric triangle or trapezoidal membership function because the differentiated result is constant for symmetric triangle or trapezoidal membership function, then the system cannot learn from different input, and no change on the size or shape of fuzzy membership

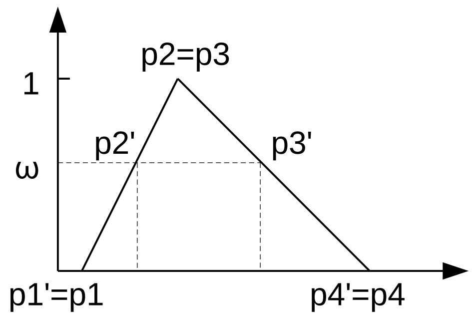


Figure 19. Triangle is truncated at the level ω.

functions can occur. The combined LMOM and LCOA method is chosen in this research because the differentiated result contains more information than LMOM as illustrated in the following context.

For a triangular membership function, given a firing strength ω, the triangle is truncated at level ω and becomes a trapezoidal shape, the parameters change as shown in Figure 19.

The new parameters changes as (for a triangular, the value of p2 and p3 is equal):

$$P1' = P1$$

$$P2' = (1-\omega)P1 + \omega P2$$

$$P3' = (1-\omega)P4 + \omega P3$$

$$P4' = P4$$

Recall that the COA of a trapezoidal membership function is:

$$y = \frac{1}{3}\frac{p_1^2 + p_2^2 + p_1 p_2 - p_3^2 - p_4^2 - p_3 p_4}{p_1 + p_2 - p_3 - p_4}$$

Then the final result of the combined LMOM and LCOA method for each fuzzy set is:

$$y_i^* = \frac{1}{3}\frac{p1'^2 + p2'^2 + p1'\,p2' - p3'^2 - p4'^2 - p3'\,p4'}{p1' + p2' - p3' - p4'}$$
$$= \frac{1}{3}\frac{(\omega^2 - 3\omega + 3)(p1^2 - p4^2) + \omega^2 (p2^2 - p3^2) + (3\omega - 2\omega^2)(p1 \cdot p2 - p3 \cdot p4)}{(2-\omega)(p1 - p4) + \omega(p2 - p3)}$$

The final output, i.e., the recommended action "$F(t)$", is the weighted average of the defuzzified output y_i^*, i.e.,

$$F(t) = \frac{\sum_{\mathrm{r}} \omega_r \cdot \mathrm{y}_i^*}{\sum_{\mathrm{r}} \omega_r}$$

where $\sum_{\mathrm{r}}$ is the sum for all rules been "fired" in the fuzzy inference system.

Action Evaluation Network (AEN)

The Action Evaluation Network (AEN), i.e., the Crisis, is a multi-layer feedforward neural network. It evaluates the system state and predicts reinforcements associated with different input states. The structure of AEN is shown in Figure 20. The input is system state variables, and the output is an evaluation of the state (a score). This output value, denoted as υ in Figure 20, is then suitably discounted

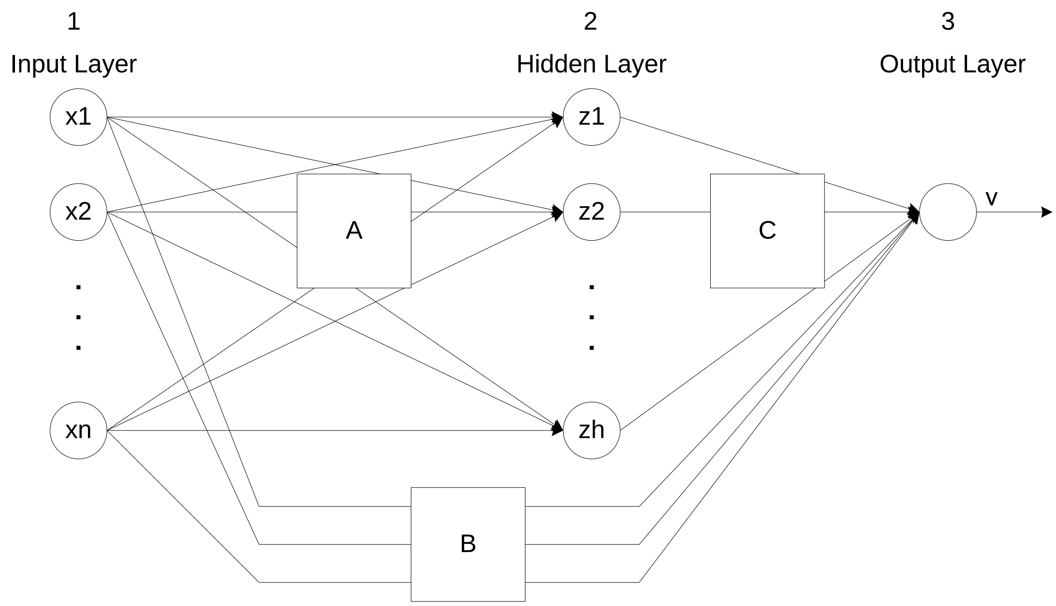


Figure 20. AEN – a multi-layer feedforward neural network.

and combined with the external failure signal to produce the internal reinforcement $\hat{r}$ (refer to Figure 17).

The output of the units in the hidden layer is calculated with the sigmoid function:

$$z_j(t,t+1) = \frac{1}{1+\exp[-\sum_{i=1}^{n} a_{ij}(t)x_i(t+1)]}; \qquad j = 1,\ldots,h$$

where,

n = the number of units in the input layer;

h = the number of units in the hidden layer;

t = the time step of the neural network weights a_{ij}, b_i and c_j in AEN;

$t+1$ = the time step of the physical system state.

There are two time indexes *(t, t+1)* in the formula of $z_j(t,t+1)$, the left time index is the time index of AEN weights, the right one is the time index of the physical system.

The final value of the unit in the output layer is calculated directly from the input layer and the hidden layer:

$$\upsilon(t,t+1) = \sum_{i=1}^{n} b_i(t) x_i(t+1) + \sum_{j=1}^{h} c_j(t) z_j(t,t+1)$$

The value of υdepends on both AEN weights time index and the physical state time index. The updating of a_{ij}, b_i and c_j only takes place after the system state is updated and a new internal reinforcement $\hat{r}(t+1)$ is gained.

The internal reinforcement at the physical system state time t +1 is:

$$\hat{r}(t+1) = \begin{cases} 0\ , & start \quad state \\ r(t+1) - \upsilon(t,t), & failure \quad state \\ r(t+1) + \gamma \cdot \upsilon(t,t+1) - \upsilon(t,t), & else \end{cases}$$

where $r(t+1)$ is the external reinforcement; a large value of $r(t+1)$ indicates a better system performance. In traffic signal control system, the external reinforcement can be reduced delay, improved speed, etc. $0 < \gamma < 1$ is a discount rate.

If the performance of the physical system is improved, AEN gains a largerυ.

Otherwise, AEN gain a smaller υ. The goal of GARIC method is to find the maximum υ as it can. But this value is discounted by γ, which means the formula gives future value ofυless importance than the current value of υ. The process of reinforcement learning is to find a larger value of the internal reinforcement $\hat{r}$ with a series of trial-and-error steps.

Stochastic Action Modifier (SAM)

For a given state, ASN always chooses an action with the largest action value $\hat{r}$. Sometimes two actions F_1 and F_2 may have approximately the same $\hat{r}$ value, and $\hat{r}_{F1}$ is just slightly larger than $\hat{r}_{F2}$. ASN will always recommend F_1. In fact, F_2 may be better than F_1, and $\hat{r}_{F2}$ will be larger than $\hat{r}_{F1}$ after just one or a few more value updates. To address this problem, a perturbation is put on the action recommend by ASN each time step to explore a better action. A Gaussian random variable with mean F and standard deviation $e^{-\hat{r}(t-1)}$ is chosen as the final action $F'(t)$ actually applied to the physical system instead of ASN recommended action $F(t)$:

$$F'(t) = F(t) + s(t)e^{-\hat{r}(t-1)}$$

From the formula above, it can be seen that when the value of $\hat{r}(t-1)$ is larger, the value of $s(t)e^{-\hat{r}(t-1)}$ is smaller, and $F'(t)$ is closer to $F(t)$. This means that the better the system performs, the smaller the perturbation is taken to explore.

Learning in AEN

AEN is a feedforward neural network. It updates its weights in the network using back propagation algorithm.

The change of the weight b_i is proportional to $\frac{\partial v}{\partial b_i} = x_i$, so b_i are updated as:

$$b_i(t+1) = b_i(t) + \beta \cdot \hat{r}(t+1) \cdot x_i(t); \qquad i = 1,\ldots,n$$

where $\beta > 0$ is a constant.

The change of the weight c_j is proportional to $\frac{\partial \nu}{\partial c_j} = z_j$, so c_j are updated as:

$$c_j(t+1) = c_j(t) + \beta \cdot \hat{r}(t+1) \cdot z_i(t,t); \qquad j = 1,\ldots,h$$

The change of the weight a_{ij} is proportional to $\frac{\partial \nu}{\partial a_{ij}} = c_j z_j (1 - z_j) x_i$, so a_{ij} are updated as:

$$a_{ij}(t+1) = a_{ij}(t) + \beta' \cdot \hat{r}(t+1) \cdot z_j(t,t) \cdot [1 - z_j(t,t)] \cdot \mathrm{sgn}[c_j(t)] \cdot x_i(t); \qquad j = 1,\ldots,h$$

where $\beta' > 0$ is a constant. The sign of c_j is used instead of its value because the algorithm is more robust [50].

Learning in ASN

ASN is a fuzzy inference system. The intent of updating of the membership functions is to maximize υ. So υ is the objective function. The updating can be done by the gradient descent method, which estimates the derivative $\frac{\partial \upsilon}{\partial p}$, and by using the learning rule:

$$\Delta p \propto \frac{\partial \upsilon}{\partial p} = \frac{\partial \upsilon}{\partial F} \frac{\partial F}{\partial p},$$

where p's are the x-coordinates of the points of the triangular or trapezoidal membership functions (as illustrated in Figure 19).

The relationship of υ and p is indirect and nonlinear, so the approximate estimate of $\frac{\partial \upsilon}{\partial F}$ is:

$$\frac{\partial \upsilon}{\partial F} \approx \mathrm{sgn}\left(\frac{\upsilon(t,t)-\upsilon(t,t-1)}{F(t)-F(t-1)}\right)$$

For the consequent labels (the fourth layer in Figure 18), the relation between F and p is direct. Recall that $F(t)=\frac{\sum_{\mathrm{r}}\omega_r \cdot \mathrm{y}_i^*}{\sum_{\mathrm{r}}\omega_r}$, and

$$y_i^* = \frac{1}{3}\frac{(\omega^2-3\omega+3)(p1^2-p4^2)+\omega^2(p2^2-p3^2)+(3\omega-2\omega^2)(p1\cdot p2-p3\cdot p4)}{(2-\omega)(p1-p4)+\omega(p2-p3)}.$$

The derivative $\frac{\partial F}{\partial p}$ is as follows.

$$\frac{\partial F}{\partial p}=\sum_r \frac{\partial F}{\partial y_i}\frac{\partial y_i}{\partial p}=\frac{1}{\sum_r \omega_i}\sum_r \omega_i \frac{\partial y_i}{\partial p}$$

In the formula above,

$$\frac{\partial y}{\partial p1}=\frac{1}{3}\frac{(\omega^2-3\omega+3)\cdot 2\cdot p1+(3\omega-2\omega^2)\cdot p2-3y\cdot(2-\omega)}{(2-\omega)(p1-p4)+\omega(p2-p3)}$$

$$\frac{\partial y}{\partial p2}=\frac{1}{3}\frac{(3\omega-2\omega^2)\cdot p1+2\omega^2\cdot p2-3y\omega}{(2-\omega)(p1-p4)+\omega(p2-p3)}$$

$$\frac{\partial y}{\partial p3}=-\frac{1}{3}\frac{2\omega^2\cdot p3+(3\omega-2\omega^2)\cdot p4-3y\omega}{(2-\omega)(p1-p4)+\omega(p2-p3)}$$

$$\frac{\partial y}{\partial p4}=-\frac{1}{3}\frac{(3\omega-2\omega^2)\cdot p3+(\omega^2-3\omega+3)\cdot 2\cdot p4-3y\cdot(2-\omega)}{(2-\omega)(p1-p4)+\omega(p2-p3)}$$

where, $p1$, $p2$, $p3$, and $p4$, mean the x-coordinates of the four points of the trapezoidal membership function.

In this research, only the membership functions of the consequent labels are updated since in many problems, this may be sufficient as well because some error in the specification of antecedent labels can be compensated for by modifying the consequent labels [50].

Framework of the Neurofuzzy Traffic Signal Control System

This research develops the neurofuzzy traffic signal controller with a simulation model. The simulation model contains the geometric layout of a pair of closely-spaced intersections, the vehicular and pedestrian traffic volumes and speeds, the virtual actuated-coordinated signal controller, etc. The neurofuzzy signal controller is coded with a programming language which interacts with the simulation model. The neurofuzzy signal control system includes a neurofuzzy controller and a microscopic simulator. The framework is shown in Figure 21. The neurofuzzy controller contains three components - a fuzzy controller (i.e., ASN), a multi-layer feedforward neural network (i.e., AEN), and a stochastic action modifier (ASN). The fuzzy controller controls the traffic signals, and the neural network and ASN are used to calibrate the membership functions in the fuzzy controller.

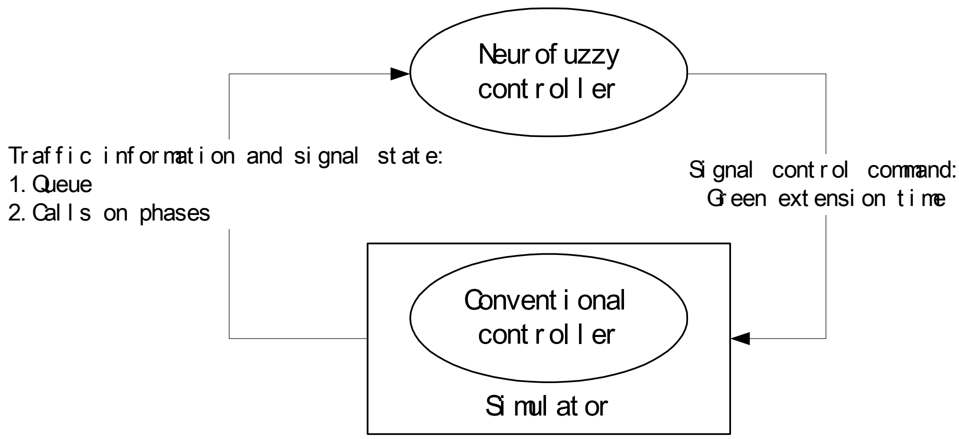


Figure 21. Framework of the neurofuzzy traffic signal control system.

Conceptually, the neurofuzzy controller is built on the conventional actuated-coordinated controller. That means the basic control parameters (min green, max green, passage time, min walk, etc.) are set up and operated by the vehicle-actuated controller in the simulator, and the fuzzy controller's commands override the commands of the conventional controller.

The microscopic simulator collects the real-time traffic data (queue in red and green, calls on phases, etc.) and signal state (green, yellow, red, walk, etc.) at the closely-spaced intersections, and sends all the data to the neurofuzzy controller; then the neurofuzzy controller analyzes the data, makes decisions, and sends a control command (green extension time) to the simulator. The command is to determine when the coordinated phase terminates based on fuzzy rules rather than typical permissive periods built in the conventional controller. This command overrides the conventional controller's control actions. At the same time, the reinforcement learning model evaluates the performance of the fuzzy control action to the system and find-tune membership functions in the fuzzy controller.

This new approach is to better "coordinate" non-coordinated phases at the downstream intersection with the upstream coordinated phase in order to improve traffic operation in a dynamic traffic environment without overly constraining the relationship between the two closely-spaced intersections. In the specific application to be tested, the objective is to postpone the beginning of the downstream interior left-turn phase to an appropriate point in the signal cycle so that the traffic from the upstream intersection can go through both intersections in an uninterrupted fashion. In order to postpone the non-coordinated phase (i.e., the downstream interior left-turn phase), the fuzzy controller needs to extend the downstream coordinated phase longer than that is controlled by the conventional controller. The fuzzy controller is designed to realize this task, i.e., extending the coordinated green phase at the downstream intersection. The fuzzy control collects traffic flow data and signal state information in real-time. And only within the green time of the downstream coordinated phase, the fuzzy controller sends signal control commands to the downstream signal controller. In detail, before the downstream coordinated green phase is terminated by the conventional controller, the fuzzy controller calculates green extension time and sends a green extension command to the downstream controller. The green extension time is decided based on fuzzy rules which look at queue lengths and vehicle/pedestrian calls from all approaches at both intersections. If the fuzzy controller decides to terminate the green time, the signal will go to the next following phase. Otherwise, if the green time is extended, then during the extension time, the fuzzy control algorithm continues to collect data and make the next decision (extend or not) after the extension time. As mentioned before, the fuzzy controller should take action before the coordinated green phase terminates; the length of the advance time depends on the neurofuzzy controller's processing speed, i.e., the faster the neurofuzzy controller works, the less the advance time is needed (Figure 22).

In addition, since the fuzzy controller is built on the conventional actuated-coordinated controller, the extension time has limits fixed up in the conventional controller, as illustrated in Figure 22.

Figure 22 illustrates a signal ring consisting 4 phases among which the phase 2 is the coordinated phase. The phase sequence is 2 - 3 - 4 -1. The "permissive window", which is portion of the cycle length during which phases other than the coordinated phases may be serviced, for each non-coordinated phase is fixed in the conventional controller ("Perm 1", "Perm 3", and "Perm 4"). This period begins timing at the coordinated phase yield point. And the beginning (yield point) and end point of the permissive windows cannot be changed by the fuzzy control algorithm. If the fuzzy controller extend the coordinated green phase time, it can only extend within the permissive windows.

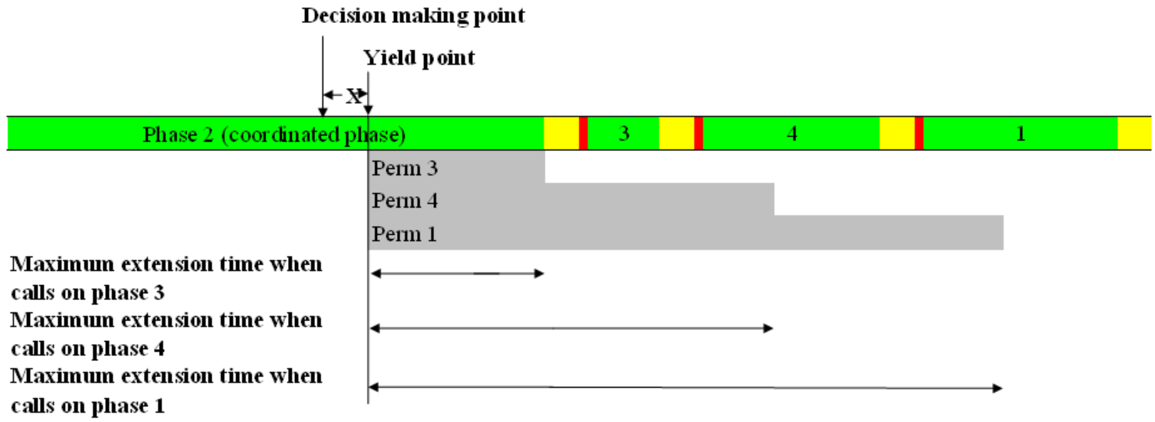


Figure 22. Extension time limits in the signal cycle.

Development of the Neurofuzzy Controller

The neurofuzzy controller is developed on the base of the actuated-coordinated signal controller. This means all other movements are controlled by the actuated-coordinated controller except that the fuzzy controller controls the downstream interior left turn movement. The fuzzy controller tries to coordinate the downstream interior left turn movement with the upstream through movement based on fuzzy rules.

For most of the previous research on fuzzy logic traffic control, the parameters of fuzzy controllers are not adjustable, i.e., the fuzzy membership functions and fuzzy rules remain consistent when traffic volume changes. However, the initial membership functions may not describe the traffic conditions and/or traffic signal control actions adequately. Calibration of fuzzy membership functions is important.

In this research, the objective is to reduce the average delay and the average number of stops. However, the fuzzy control output is fuzzy extension times, not the delays or the number of stops. Furthermore, the traffic system cannot give the “desired” or target minimum delay or number of stops. Consequently, we do not have the “desired” output at each input pattern in the neural network training sequence. So neither the “supervised” or “unsupervised” learning algorithms of neural networks can be used here.

The reinforcement learning algorithm (RL) is chosen to adjust the fuzzy membership functions. The RL algorithm evaluates whether the output improves the system; if yes, then “reinforces” this tendency, otherwise, “punishes” this tendency. This works like a human “try” and “error” learning process. The GARIC model is used to adjust the membership functions in the fuzzy controller. The Structure of the neurofuzzy control system with reinforcement learning function is shown in Figure 23.

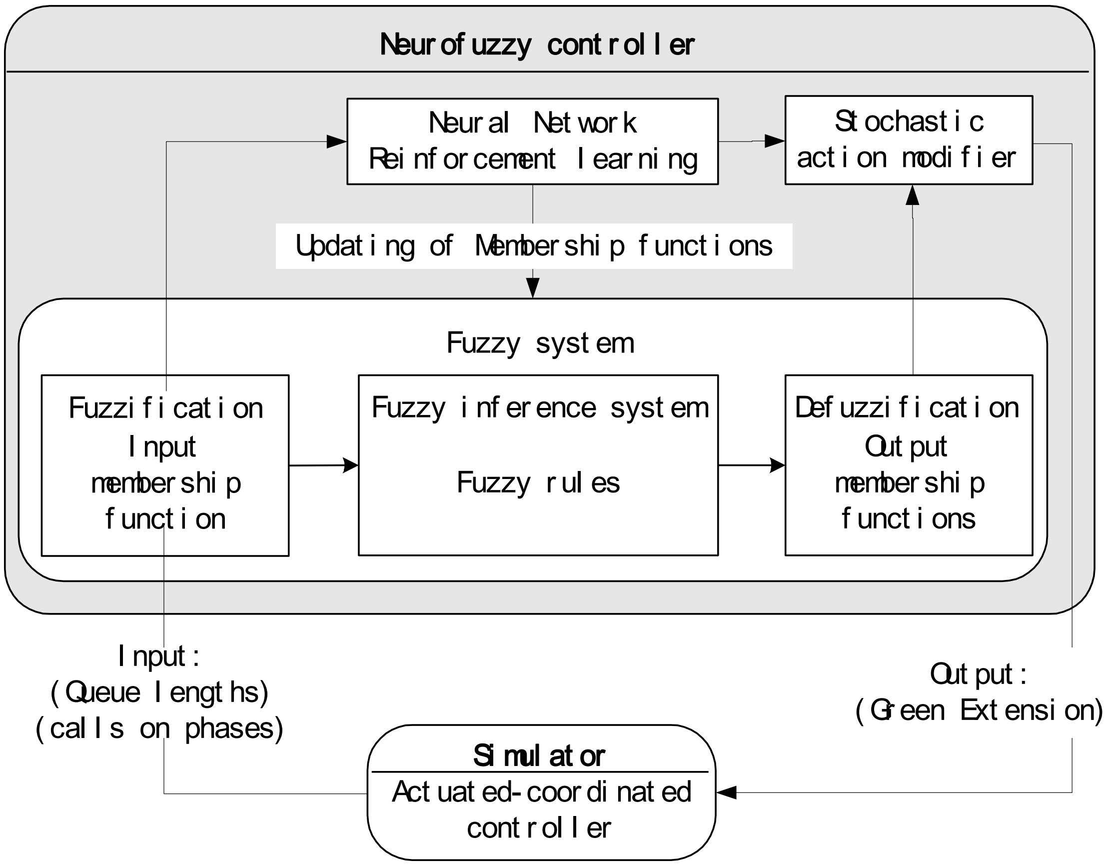


Figure 23. Structure of the neurofuzzy controller.

The following sections discuss how the traffic and signal information been processed in the fuzzy system. Three main fuzzy control steps are described: fuzzification, fuzzy inference system, and the defuzzification.

Fuzzification

(1) Input

The input variables are QU and QD which represent queue lengths at non-coordinated approaches at the upstream intersection and the downstream intersection.

The phase layout, queue measure detector locations at the closely-spaced intersections are indicated in Figure 24. Assuming that turning traffic from side streets moves with the through traffic, "QD_i" and "QU_i" represent the queue lengths at the downstream intersection and the upstream intersections respectively (i represents the corresponding phase number).

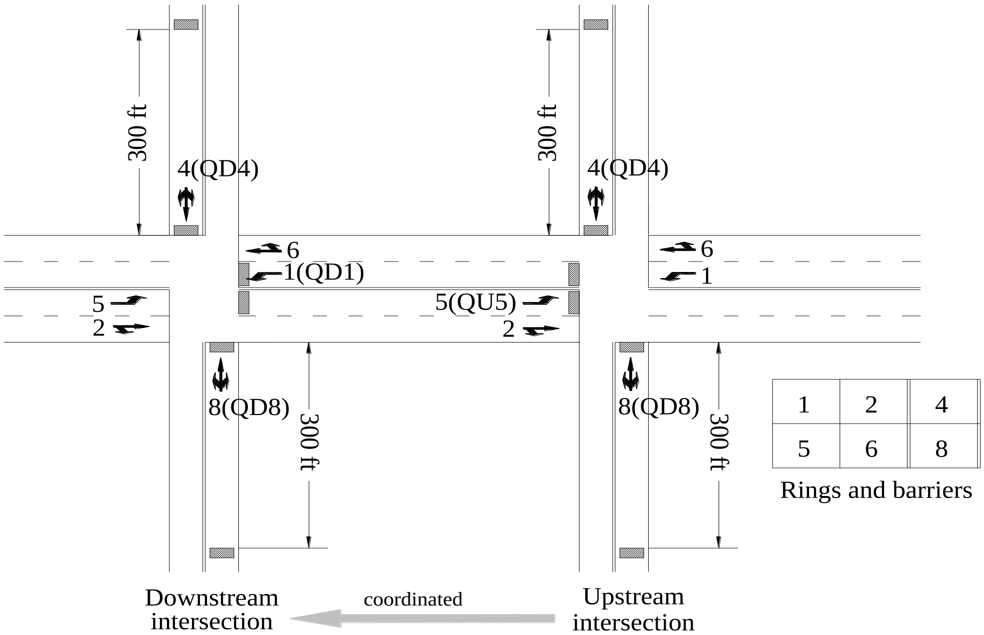


Figure 24. Phase and detector layout.

With respect to the upstream intersection in Figure 24, the beginning time of the coordinated phase (phase 6) is affected by phase 5 and phase 4/8, and the time duration of phase 5 and phase 4/8 can be predicted by the corresponding queue lengths with the formula: $QU = \{\max(QU4, QU8) + QU5\}$.

Similarly, for the downstream intersection, the ending time of phase 1 (downstream interior left turn phase) is affected by phase 4/8 and phase 1 itself, and this time duration can be predicted by queue lengths at phase 4/8 and phase 1. The time duration of phase 4/8 plus phase 1 can be estimated by the formula $QD = \{\max(QD4, QD8) + QD1\}$.

If there is no traffic demand existing on phase 4/8, then the phase 1 will begin very early because the phase 4/8 is skipped. In this case, the duration of phase 1 is estimated by the formula $QD = QD1$.

To measure queue length at each approach, two detectors are deployed each lane. One is located immediately upstream of the stop line while the other is located well upstream. The distance between the two detectors determines the maximum queue length the fuzzy controller can detect. For this research, a distance of 300 feet was used except for the interior left turning lanes where one detector is immediately upstream of the stop-line while the other is at the entrance of the left-turn lane/bay. Many closely-spaced intersections have full left-turn lanes in order to maximum the queue storage capacity. For intersections with full left turn lanes, the upstream detector is located at the departure part of the upstream intersection in order to detect the maximum queue length (storage capacity).

For a given approach *i*, the queue length at time step t is:

$$Q_i(t) = \min\{Q_i(t-1) + (A_i(t) - D_i(t)), \quad Q_{i,\lim}\}$$

where,

$Q_i(t)$ = queue length of approach i at time step t;

$Q_i(t-1)$ = queue length of approach i at time step t-1;

$A_i(t\text{-}1)$ = number of vehicles passing the upstream detector of approach i at time step t;

$D_i(t\text{-}1)$ = number of vehicles passing the stop-line detector of approach i at time step t;

$Q_{i,\lim}$ =maximum queue length, determined by the distance between two queue measure detectors.

(2) Membership Functions of input and output

The mathematical input variables (queue lengths) need to be matched with the membership functions and be fuzzified to linguistic descriptions. The membership functions of the queue length are indicated in Figure 25.

For a mathematical queue length, the fuzzification module converts it to linguistic values (“short”, “medium”, “long”, and “very long”,) with the corresponding membership. For example, given queue length of 60 ft, the fuzzification module convents it into two fuzzy values: “short” with 50% membership and “medium” with 50% membership.

(3) Output and Its Membership Functions

The output is the “Green Extension” time for the coordinated green phase. The extension time is an integer, ranging from 2 seconds to 10 seconds depending on the traffic condition at both intersections. It need be noted that if no action is needed from the neurofuzzy controller, the fuzzy controller does not generate output, which means the operation is controlled by the conventional controller and the extension time is 0 second.

The membership functions of the Green Extension are shown in Figure 26.

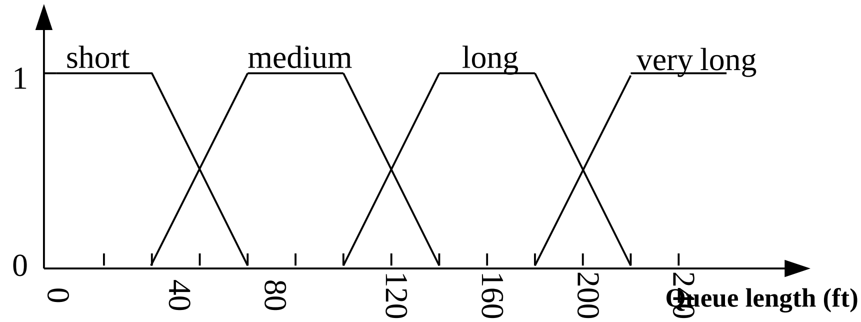


Figure 25. Membership functions of the queue length.

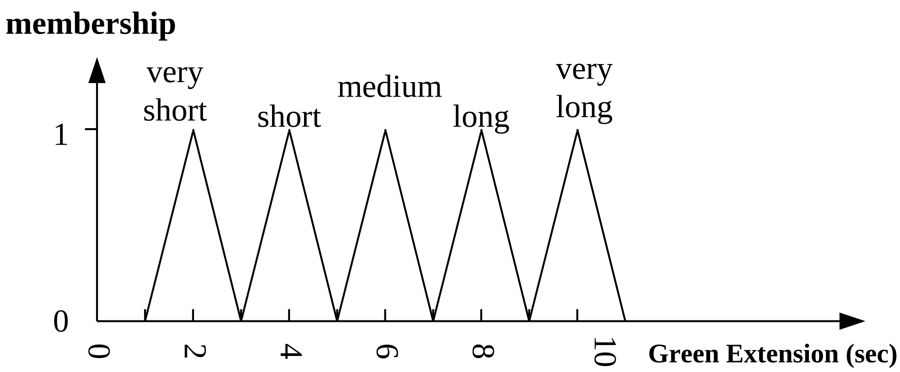


Figure 26. Membership functions of the Green Extension time.

Fuzzy Inference System

In the fuzzy inference system, the fuzzy rules are used to make the traffic from the upstream phase 6 to reach the downstream phase 1 before phase 1 “gaps out”. The fuzzy rules need to balance the traffic at the upstream coordinated phase and the downstream non-coordinated phase, and find an appropriate beginning point of the downstream phase 1 by extending the coordinated phase. The fuzzy rules look at factors that lead to the variation time relationship between the upstream intersection and the downstream interior left turn movement, and the goal is to reduce the variation.

There are two sets of fuzzy rules depending on whether there is traffic demand are on phase 4/8 or not. The fuzzy rules are shown in Table 2 and Table 3.

Table 2. Fuzzy rules – scenario 1: traffic demand on downstream phase 4/8.

		Max (QU4, QU8)+QU5			
		Short	Medium	Long	Very long
Max (QD4, QD8) + QD1	Short	S	M	L	VL
	Medium	VS	S	M	L
	Long	No action	VS	S	M
	Very long	No action	No action	VS	S

Table 3. Fuzzy rules – scenario 2: no traffic demand on downstream phase 4/8.

		Max (QU4, QU8)+QU5			
		Short	Medium	Long	Very long
QD1	Short	M	L	VL	VL
	Medium	S	M	L	VL
	Long	VS	S	M	L
	Very long	No action	VS	S	M

Note: VS, S, M, L, VL, and No action represent Green Extension time is Very Short, Short, Medium, Long, Very Long, and zero.

The general format of the fuzzy rules is:

If $\{QD$ is $Very\ long\}$ and $\{QU$ is $Very\ short\}$ then $\{Green\ Extension$ is $No\ \ action\}$

As mentioned before, the conditions that there is traffic demand on downstream phase4/8 or not is different, and the fuzzy logic model should develop two different sets of fuzzy rules for each of the two cases. The reason two sets of fuzzy rules are required is that if traffic demand exists on downstream phase 4/8, then phase 4/8 should be served before downstream phase 1, and the duration of phase 4/8 should be no less than {minimum green + yellow + all red} even if only one vehicle called for phase 4/8. Consequently, the downstream phase 1 will be postponed significantly. For example, given QD4=QD8=25 and QD1=100 in case 1 and QD1=125 in case 2, although the value of {Max (QD4, QD8 + QD1} in case 1 equals to the value of {QD1} in case 2, the ending times of phase 1 are greatly different for these two cases.

Pedestrians are also considered in the fuzzy control system, the Walk (W) and the Flashing Don't Walk (FDW) time are constructed in the fuzzy inference system. Since they are usually of fixed length in the conventional controller, pedestrian phases are not fuzzified but converted to the equivalent queue lengths with a lost time of 2 seconds, a saturation flow rate of 1800 veh/h (i.e., a saturation headway of 2 sec/veh), and an average vehicle length of 23 feet. For example, given W+FDW as 25 seconds, then the equivalent queue length will be:

$(W + FDW - lost\_time)/saturation\_headway = (25-2)/2 = 11.5$ equivalent vehicles,

and the equivalent queue length is:

$(number\_of\_vehicles) \times (vehicle\_length) = 11.5 \times 23 = 265$ feet.

The fuzzy inference system uses the larger value of the vehicle queue length and the equivalent queue length as input.

Defuzzification

The output of the fuzzy inference system is the extension time for the coordinated green phase ("short", "long", etc.) in the form of one or several linguistic descriptions of the fuzzy output with memberships. It should be noted though that the control action (extension time) is of a mathematical value. The fuzzy output values of all "fired" fuzzy rules are aggregated and defuzzified to lead to the determination of the final control action – green extension time. In this research, the combined LMOM and LCOA defuzzification method is used.

Calibration of the Fuzzy Controller

In the fuzzy traffic signal controller, the input variables of the fuzzy controller are the queue lengths at non-coordinated approaches at both intersections, and the output is the green extension time. Initially, the membership functions of the input and output are set up manually based on expert knowledge. These membership functions may not best fit the traffic situation. So calibrating membership functions is important to improve the performance of the fuzzy controller.

The GARIC reinforcement learning algorithm is used to do the calibration. In this research only the membership functions in the consequent label (i.e., membership functions of green extension time) are updated since this may be sufficient because some error in the specification of antecedent labels (i.e., membership functions of queue length) can be compensated for by modifying the consequent labels [50].

GARIC Algorithm Applied In Traffic Signal Control

When train the fuzzy controller using GARIC algorithm by neural network, the fuzzy controller is represented in the neural network format and the fuzzy controller is updated with another neural network which is the Action Evaluation Network. In GARIC, the Action Selection Network (ASN) is the fuzzy signal controller. The input of ASN includes QD and QU. The action selection process of the fuzzy controller process has been discussed in last section. After the fuzzy signal controller recommends an action, the Stochastic Action Modifier (SAM) puts in a “momentum” to the recommended action in order to explore a better action which may improve the signal timing eventually.

QD and QU are also input to the Action Evaluation Network. The value of υ and $\hat{r}$ are calculated at each observation. The purpose of the algorithm in maximizing υ in AEN, is to improve system performance through the reduction of delay and/or number of stops. To this end, the resultant changes in the values of delay and stops (the difference between the last simulation run and this simulation run) are used as the external performance measurement for the calculation of the internal reinforcement. The average delay and the number of stops are combined in the following fashion:

$$r = \alpha \cdot delay + \beta \cdot stops$$

where, *delay* and *stops* are reductions in average delay and average number of stops per vehicle, r is the external performance measurement, and α and β are coefficients. In this research $\alpha = 0.9$ and $\beta = 0.1$, which means the average delay is prior to the average number of stops when training the fuzzy controller.

If the delay of this simulation run is less than the last time run, the avoid delay is positive, the value of υ may increase. The last simulation run delay is gained after the whole simulation run is completed.

CHAPTER IV

EVALUATION OF THE NEUROFUZZY CONTROL SYSTEM

This chapter presents the performance evaluation of the neurofuzzy traffic signal control model. The microscopic simulation program VISSIM [56, 57] was used to model a pair of closely-spaced intersections. The neurofuzzy controller was trained with the simulation model to obtain the adjusted membership functions. The trained fuzzy controller was subsequently employed to evaluate the performance of signal control strategies within the simulated environment for various scenarios. Comparisons were conducted of the fuzzy controller before and after training, and of the conventional actuated-coordinated controller.

Experimental Design

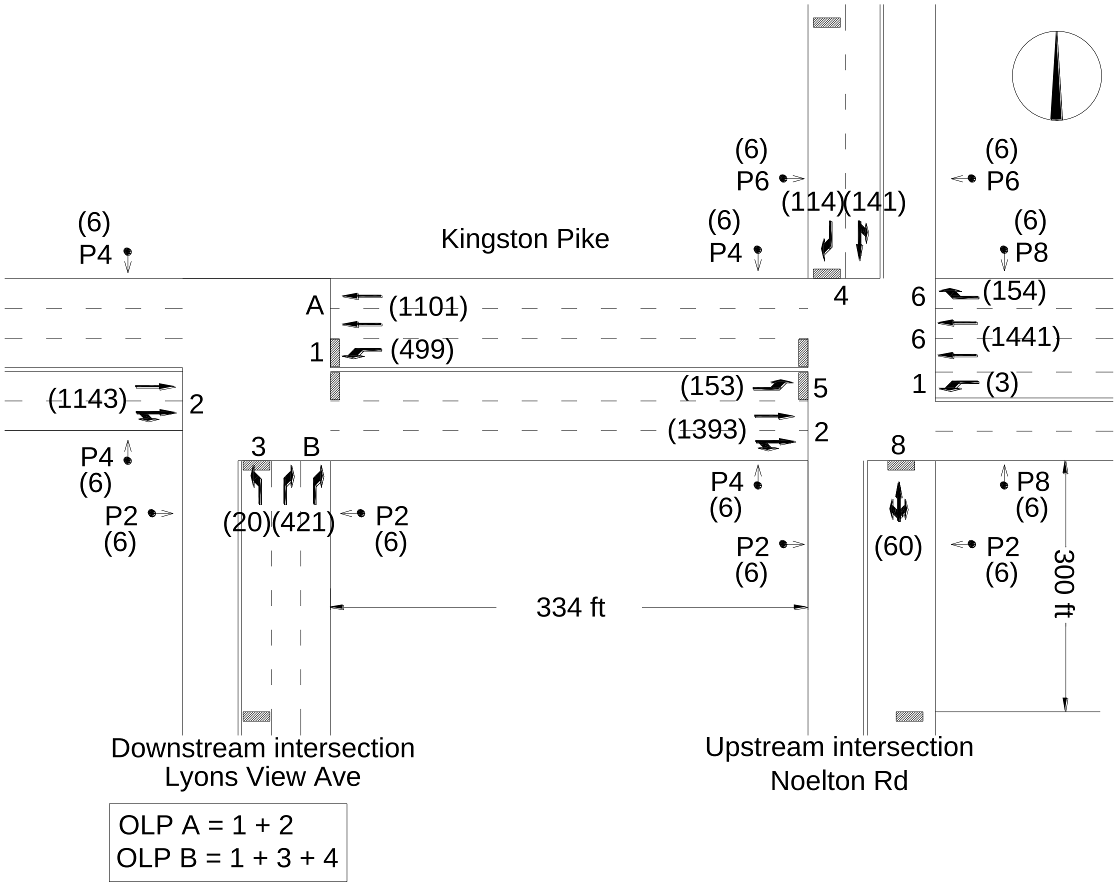
Geometric Design

The site used for this research was Kingston Pike at Noelton Road and Lyons View Avenue in Knoxville, TN (see Figure 27). The interior space between the two intersections is 334ft. Kingston Pike is a major arterial connecting the older downtown area and the city's populous suburbs to the west. During the PM peak hours, west-bound traffic is dominant. The left turn traffic from west-bound Kingston Pike to Lyons View Avenue is high, which often leads to queue spill-backs at the upstream Noelton Road intersection and aggravate the already congested west-bound through movement at that intersection. The geometric layout, traffic volumes (veh/h), and phase/overlap configurations are illustrated, not to the scale, in Figure 27.

The Conventional Controller configurations

Lyons View Avenue intersection and Noelton Road intersection are currently controlled with two actuated controllers coordinated with other signal controllers along Kingston Pike.

The traffic signal timing and settings at the two intersections are shown in Table 4 and Table 5.



A			
1	2	3	4
B		B	

Downstream intersection

1	2	4
5	6	8

Upstream intersection

Figure 27. Geometric layout, traffic volumes, and phase/overlap configurations of the experimental site.

Table 4. Signal timing and settings at Noelton Road intersection.

Phase	1	2	4	5	6	8
Initial (sec)	6	18	4	6	18	4
Split (%)	11%	71%	18%	12%	70%	18%
Passage (sec)	2.0	2.0	2.0	2.0	2.0	2.0
Yellow (sec)	4.0	4.0	4.0	4.0	4.0	4.0
Red Clear (sec)	1.0	1.0	1.0	1.0	1.0	1.0
Max 1 (sec)	15	50	25	15	50	25
Max 2 (sec)	20	55	30	20	55	30
Walk (sec)	0	8	4	0	8	4
Ped Clear (sec)	0	10	21	0	10	21
Max recall		Yes			Yes	
Cycle length (sec)	130					
Coordinated phase		Yes				
Offset	75%					
Reference point	Beginning of yellow					

Table 5. Signal timing and settings at Lyons View Avenue intersection.

Phase	1	2	3	4
Initial (sec)	6	15	6	6
Split (%)	21%	50%	10%	19%
Passage (sec)	2.0	2.0	2.0	2.0
Yellow (sec)	4.0	4.0	4.0	4.0
Red Clear (sec)	1.0	1.0	1.0	1.0
Max 1 (sec)	25	50	25	30
Max 2 (sec)	30	55	30	35
Walk (sec)	0	5	0	5
Ped Clear (sec)	0	20	0	15
Max recall		Yes		
Cycle length (sec)	130			
Coordinated phase		Yes		
Offset	85%			
Reference point	Beginning of yellow			

The controller uses fixed force off type and parallel permissive windows. The fixed force off means the phases are forced-off at fixed points in the cycle. This allows any unused time to any following phase, up to that phases' force-off (not necessarily only the coordinated phases but also the non-coordinated phases). The fixed force-off allows it possible that the downstream left turn phase receive the unused time from the side-street phases such that the left turn phase can be longer than its split time.

The parallel permissive window means the permissive windows for non-coordinated phases do not starts sequentially, but overlap and in parallel. All permissive windows can begin from the same point but not necessarily ends at the same point, and during the parallel permissive period, the controller yields to any non-coordinated phase (Figure 22). This type of setting allows the controller terminates the coordinated phase promptly when traffic demand is on any non-coordinated phase.

Neurofuzzy Controller Design

The neurofuzzy controller was designed using the MATLAB [52, 53, and 58] that interacts with the simulation model via VISSIM's COM interface [57] as presented in Figure 28.

At the beginning of the training process, the manually established membership functions were used in ASN (the fuzzy controller). In AEN, there were 20 neurons in the hidden layer, and weights of the neural network were randomly initialized between -1 and 1. The entire training process is shown in Figure 29.

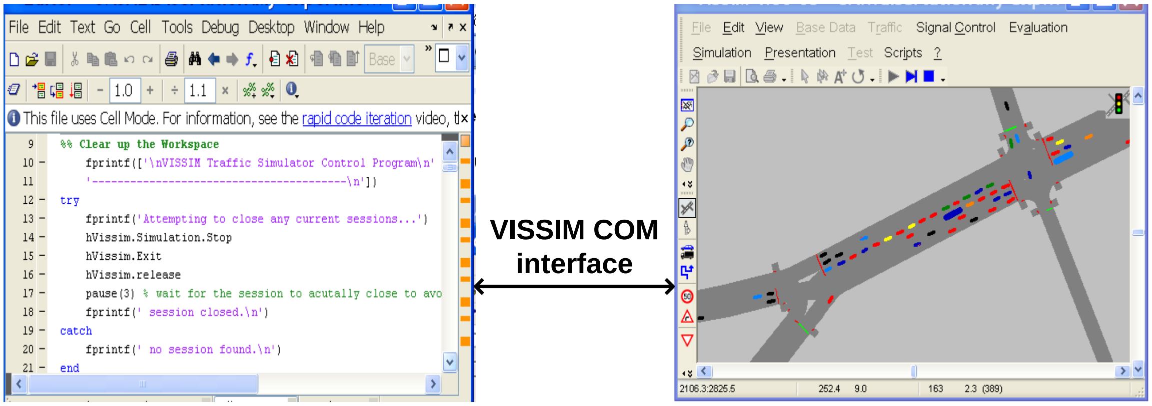


Figure 28. MATLAB controls VISSIM model via VISSIM's COM interface.

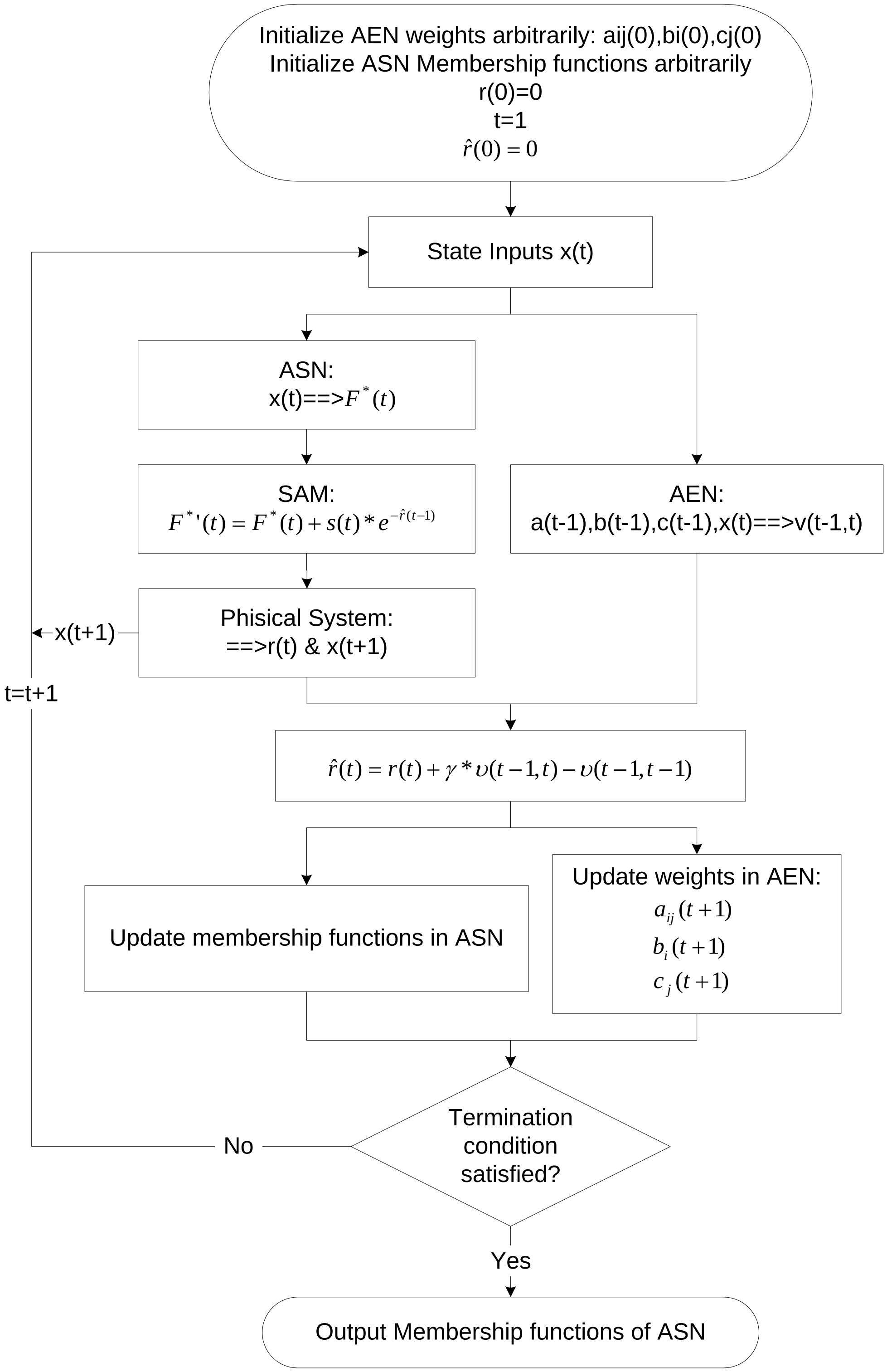


Figure 29. Training process of the neurofuzzy controller.

In the training process, the υ value is the combination of avoided delay and number of stops. The ideal method is that the system observes the fuzzy control action and calculates the avoid delay and number of stops at each observation, but the VISSIM software package calculates delay and stops only after vehicles finish their trips and exist the road network, therefore it is hard to estimate the delay and stops caused by each fuzzy action in real time. The VISSIM model produces the measure of effectiveness (delays, number of stops, etc.) after each simulation run, of which delays and stops were values for all signal control actions. So in the actually training process, the external reinforcement r is the combination of avoided average delay and stops after the simulation run. And the parameters of ASN and AEN were updated after each simulation run. The value of the stochastic perturbation of fuzzy action keeps same throughout the simulation run in VISSIM. This training process is slower than updating parameters at each observation, but the result still makes sense because a better control yields less delay and stops as a whole system.

The fuzzy control process in ASN is shown in Figure 30.

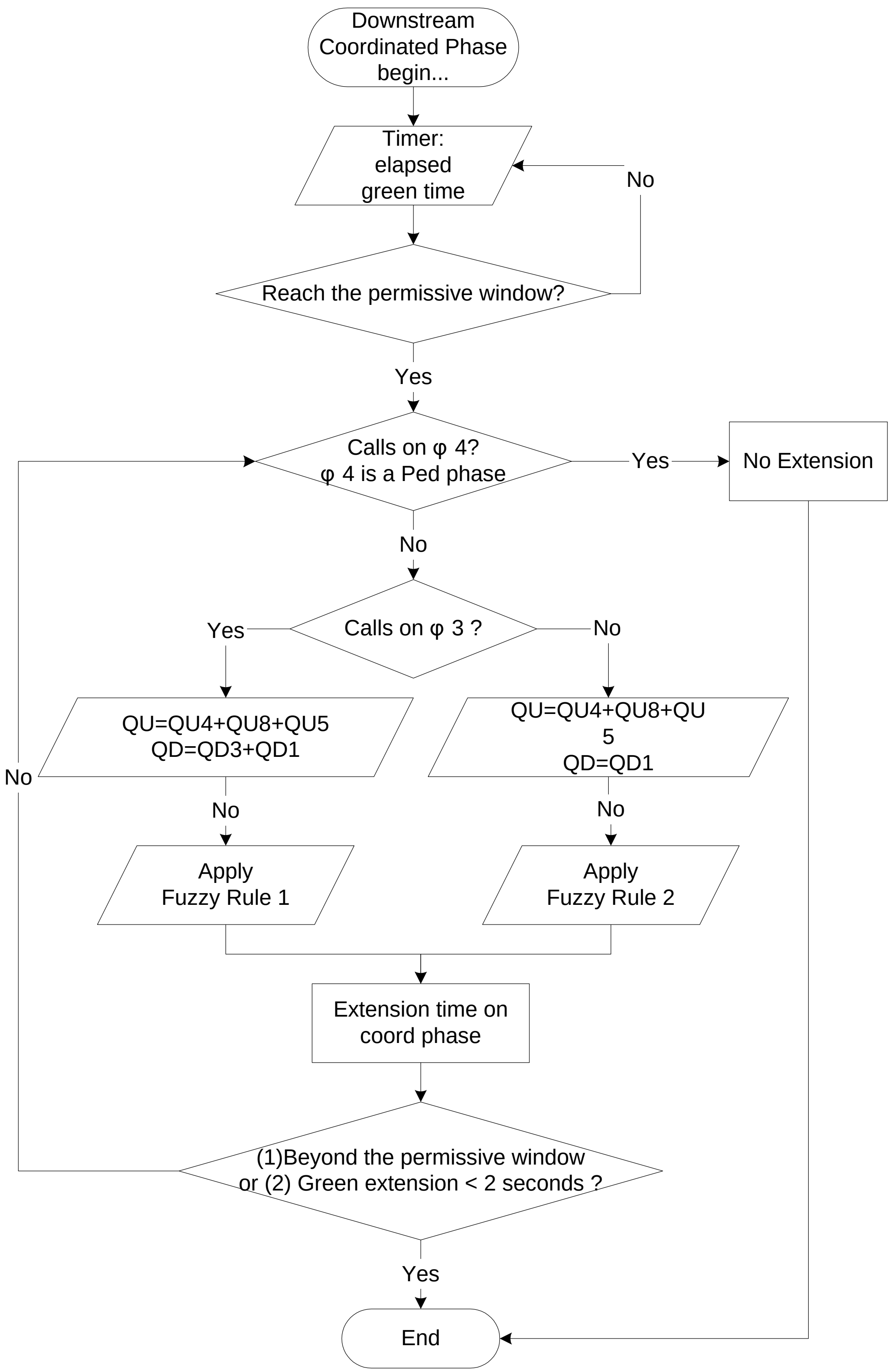


Figure 30. Fuzzy control process in ASN.

The updating process of parameters in AEN has been discussed in the "Basic Concepts of Reinforcement Learning and GARIC Method" section.

Simulation Model Design

As discussed before, the beginning of the upstream (right) phase 6 is affected by the traffic volumes of the upstream phases 5, 4 and 8, and the beginning of the downstream phase 1 is affected by the traffic volumes at downstream phases 3, 4 and 8. So in order to evaluate the fuzzy controller's ability to address stochastic fluctuations of traffic demand, different traffic volumes for these movements were used.

Five sets of volumes for upstream phases 5, 4 and 8 and downstream phases 3, 4 and 8 were used to test the performance of the neurofuzzy control system:

(1) The base case traffic volumes as manually counted at study site (Figure 27);

(2) For phases 4, 5, and 8 at the upstream intersection and phases 1, 3, 4, overlap B (which occurs with phases 1, 3, and 4) at the downstream intersection, decrease vehicular and pedestrian volumes by 20%. Keep the volumes at other approaches the same as in (1). The volumes are shown in Figure 31.

(3) For phases 4, 5, and 8 at the upstream intersection and phases 1, 3, 4, overlap B at the downstream intersection, decrease vehicular and pedestrian volumes by 10%. Keep the volumes at other approaches the same as in (1). The volumes are shown in Figure 32.

(4) For phases 4, 5, and 8 at the upstream intersection and phases 1, 3, 4, and overlap B at the downstream (left) intersection, increase vehicular and pedestrian volumes by 10%. Keep the volumes at other approaches the same as in (1). The volumes are shown in Figure 33.

(5) For phases 4, 5, and 8 at the upstream intersection and phases 1, 3, 4, and overlap B at the downstream intersection, increase vehicular and pedestrian volumes by 20%. Keep the volumes at other approaches the same as in (1). The volumes are shown in Figure 34.

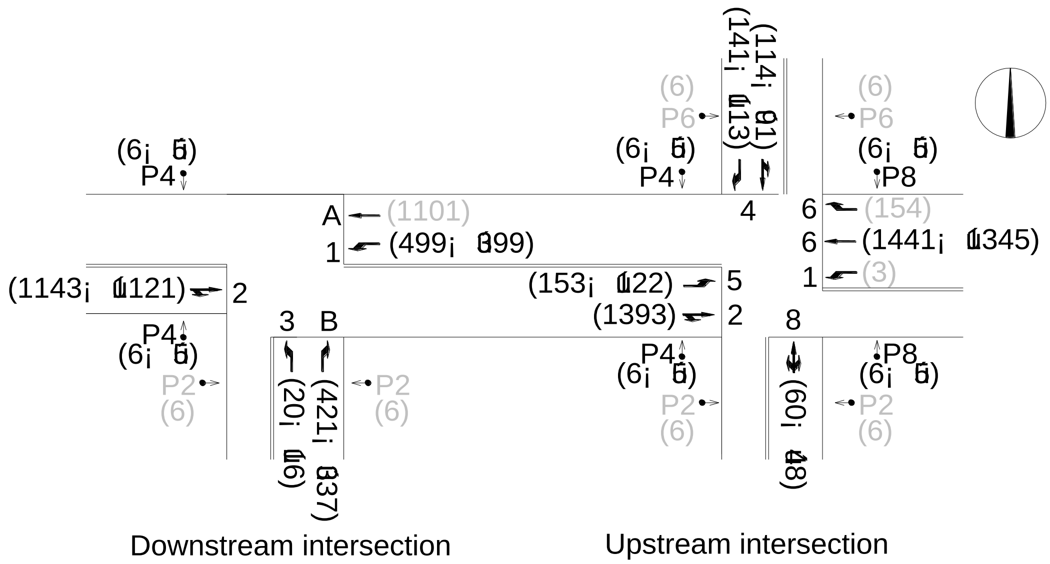


Figure 31. 80% of the original volumes.

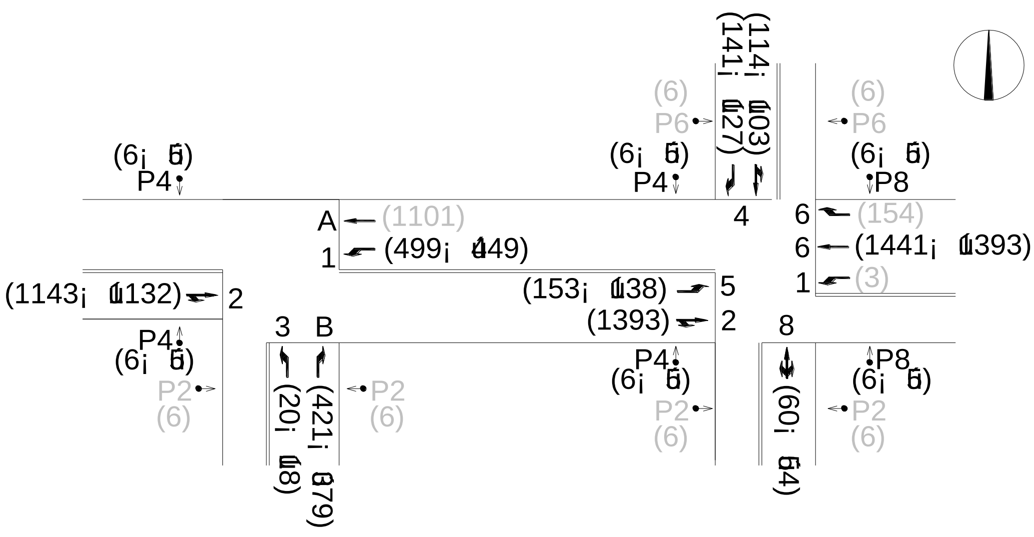


Figure 32. 90% of the original volumes.

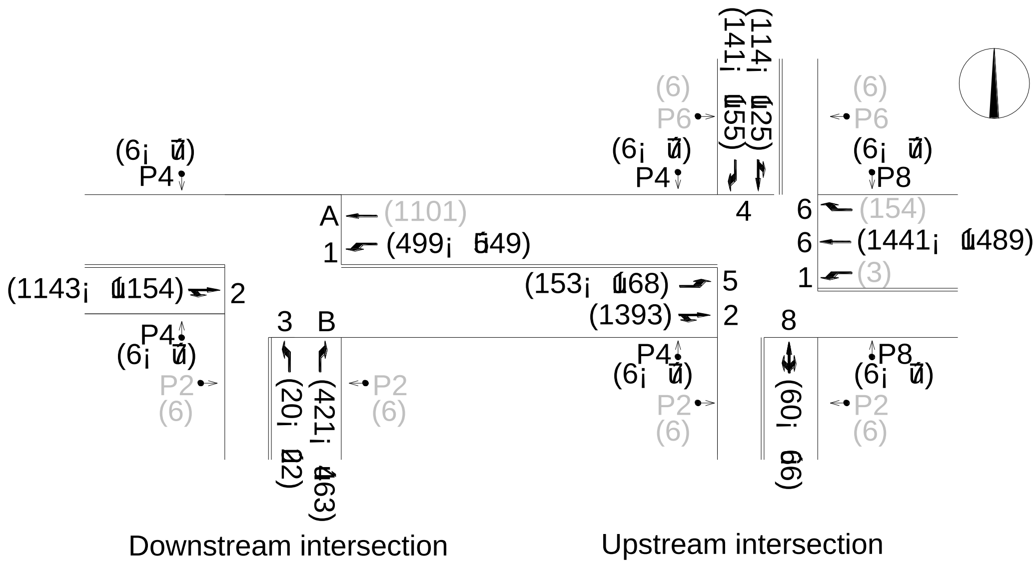


Figure 33. 110% of the original volumes.

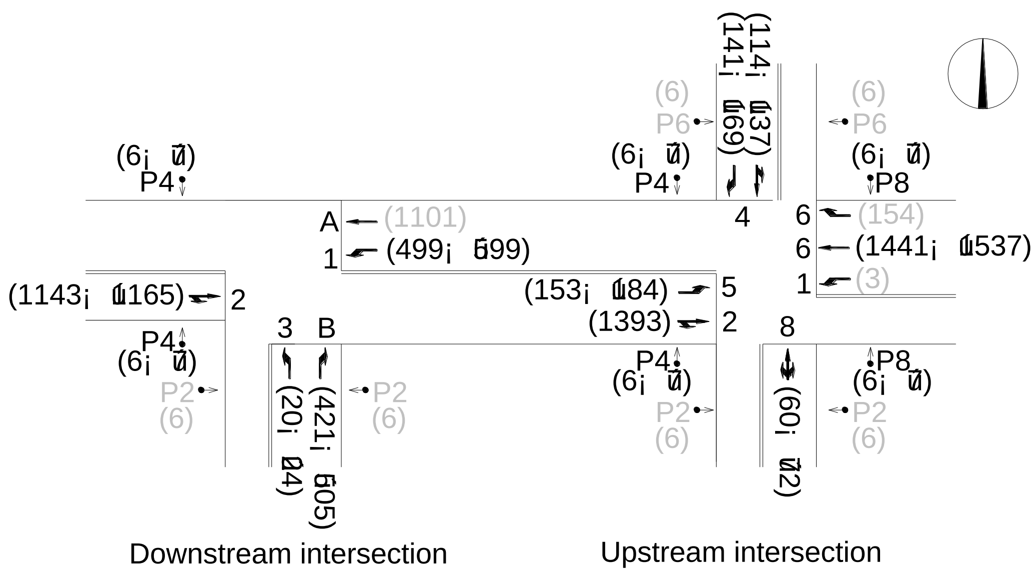


Figure 34. 120% of the original volumes.

In order to evaluate and compare the fuzzy controller with the conventional actuated-coordinated controller, all simulation runs were performed for one hour. The initial 15 minutes was treated as the "warm-up" period and discarded. Only the last 45 minutes of the simulation was used to evaluate the performance of the system.

The measurements of effectiveness (MOEs) were the number of stops per vehicle and the average delay per vehicle for the simulation model. These two MOEs are most commonly used to evaluate signal operations at intersections and can be easily collected in VISSIM. In addition, the average speed and the average stopped delay were also chosen as supplemental options to better evaluate the performance.

The original volumes counted at the study site were selected to calibrate the fuzzy membership functions. Only the membership functions for "Green Extension" time in the fuzzy controller were calibrated. The fuzzy controller was subsequently applied to all five cases using the calibrated membership functions. The objective is to see whether the calibrated membership functions can improve the traffic operations at closely space intersections under different traffic volume conditions.

Evaluation

The neurofuzzy control system after reinforcement learning was compared with the fuzzy control system before learning and also the conventional actuated-coordinated control system. For the purpose of comparison, all three control systems were simulated in the identical environment for each case. According to the Central Limit Theorem, every case was simulated 30 runs with different random seeds each of the three systems. Since different controllers were implemented to the same traffic conditions with the same random seed, paired T-test was used to compare the MOEs. A confidence level of 95% was used for the comparisons.

The following are the assumptions of the paired t-test.

1. The sampling distribution of the d_i s (differences of the paired values) is a normal distribution. ($d_i = after_i - before_i$).

2. The d_i's are independent, i.e., the pairs of observations are independent.

The summery of the paired t-test is:

Ho: case 1 (delay, number of stops, stopped delay) $\mu_d \geq 0$, i.e., $after \geq before$.

case 2 (speed) $\mu_d \geq 0$, i.e., $after \leq before$.

Ha: case 1 (delay, number of stops, stopped delay) $\mu_d < 0$, i.e., $after < before$.

case 2 (speed) $\mu_d > 0$, i.e., $after > before$.

T-test: $t = \dfrac{\overline{d}}{s_d / \sqrt{n}}$

Reject rule: For a 95% confidence level, with degree of freedom of $n-1$, reject Ho if P-value<0.05.

The Central Limit Theorem is: The sampling distribution of a sample mean from a large random sample (size n) from a population with mean μ and standard deviation σ will be approximately normal with mean μ and standard deviation

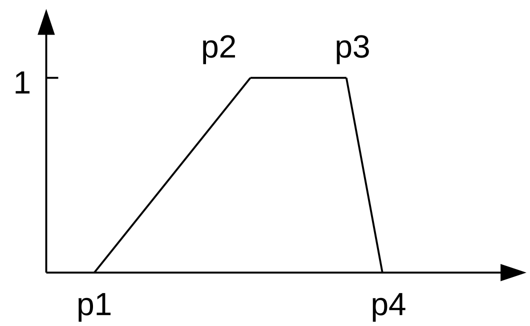
$\sigma_{\bar{x}} = \sigma / \sqrt{n}$.

Results and Discussion

New Membership Functions of the "Green Extension"

The neurofuzzy controller was trained under the original traffic volumes. The new membership functions of the green extension time after training is shown in Table 6 and Figure 35. The membership functions of the Green Extension time were trapezoids although these initial functions were reduced to triangular functions.

Table 6. Membership functions of Green Extension time after training.



Green Extension	p1	p2	p3	p4
Very Short	0	0.21	3.41	5.65
Short	0.91	5.54	8.2	8.29
Medium	4.09	5.93	8.2	8.2
Long	4.6	4.79	8.89	10.27
Very Long	8.7	11.17	13.65	14.55

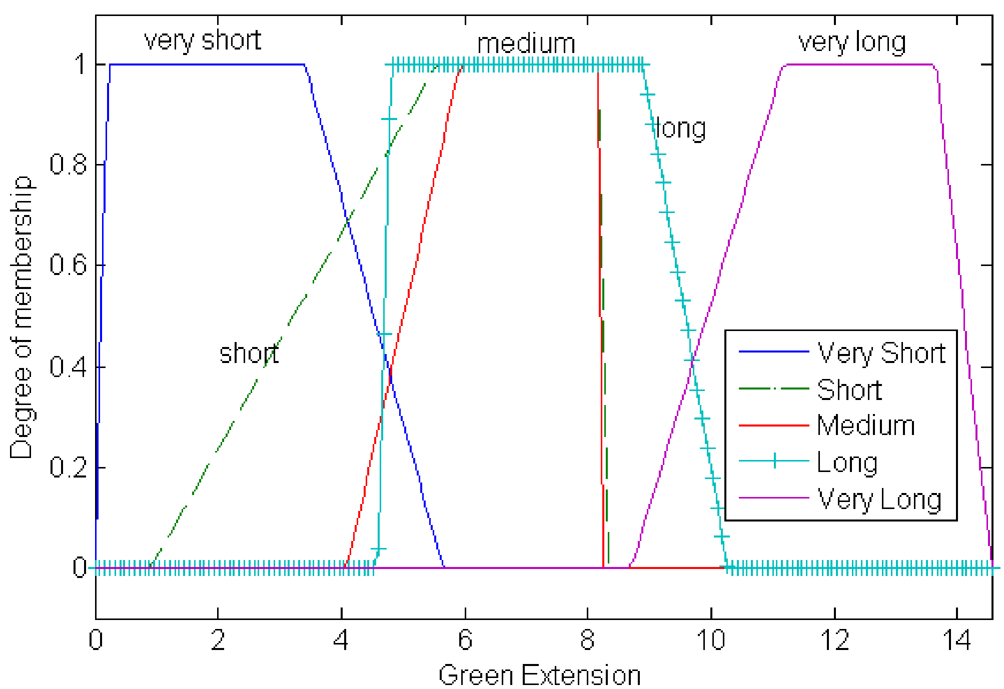


Figure 35. Membership functions of Green Extension time after training.

Comparison of the Traffic Signal Operations

Part of the signal operations under the actuated-coordinated controller and the fuzzy controller after training is indicated in Figure 36. Recall that the cycle length is 130 sec, the reference point is the beginning of yellow (phases 2 and 6), and the offset is 13 sec. Under the actuated-coordinated control, the end of phase 2 at the downstream intersection is always 13 sec later than that of the upstream intersection, and then followed by non-coordinated phases. Traffic demand from non-coordinated approaches varies vastly. Sometimes the non-coordinated phases (phases 3 and 4/8, for example) may be skipped due to a lack of any demand. Consequently, the left turn phase 1 would commence and expire too early before serving or fully serving the traffic demand from the upstream intersections. Under this condition, queue, due to later arrival from upstream, builds up and may, at times, spill over and block the through lanes at the upstream intersection.

Under the fuzzy signal control, of which the membership functions of Green Extension time are calibrated, the signal operation at the upstream intersection remains the same, whereas the end of the coordinated phase (phase 2) at the downstream intersection is postponed, i.e., the beginning of the downstream left turn phase is postponed. The duration of this postponement depends on the traffic demand from all side streets and left turning movements at both intersections. The fuzzy controller postpones the beginning of the downstream left turn movement to a time when the traffic from the upstream intersection can arrive and join the queue at the downstream left turn lane and be served. In so doing, the fuzzy controller establishes a “secondary coordination” between the upstream coordinated phase and the downstream non-coordinated phase (left-turn phase) based on real-time traffic demand. By comparing the operations under the actuated-coordinated controller and the trained fuzzy controller, see figure 36, more traffic from the upstream through approach (phase 6) goes through the downstream left turn. This “secondary coordination” favors left turn progression and, hence, reduces the delay and stops associated with this movement.

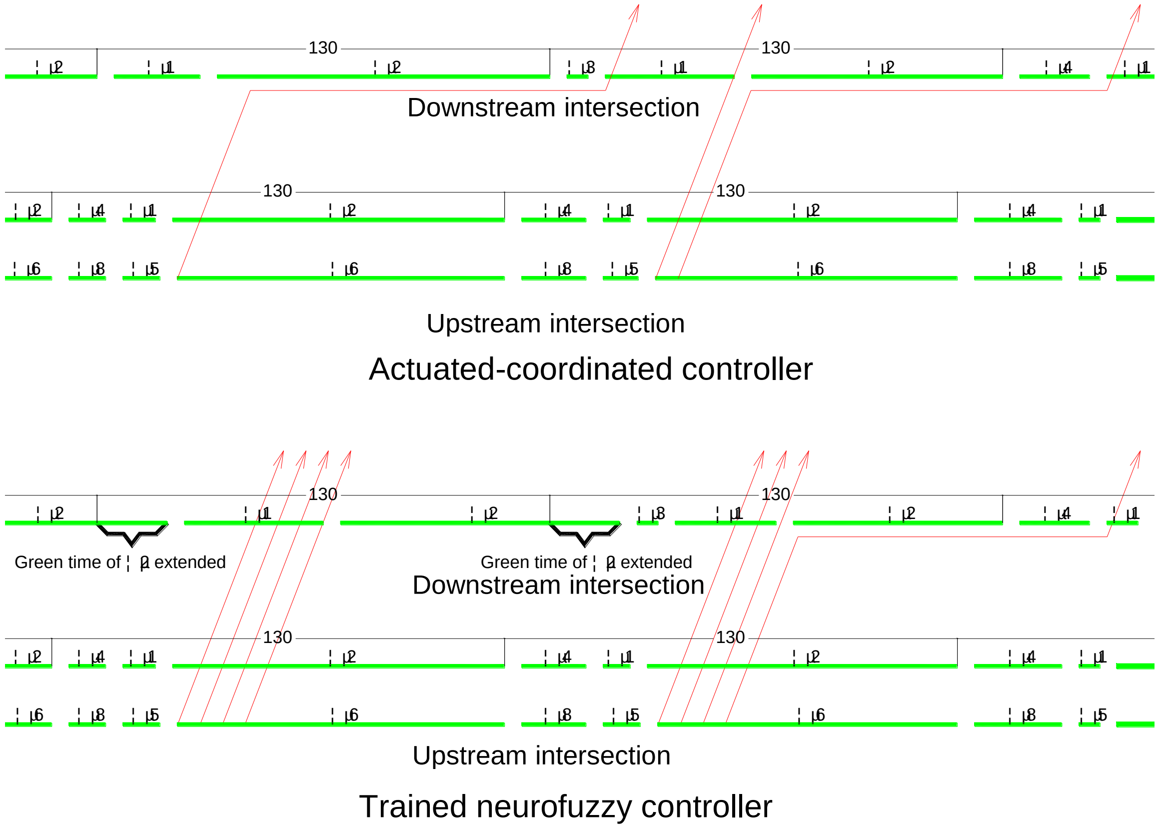


Figure 36. Signal operations under the actuated-coordinated controller and the fuzzy controller after training.

Comparison of the Model Performances

The three traffic signal control systems were applied in simulation models and compared under the five different traffic volumes as mentioned before. These three traffic signal controllers are: the actuated-coordinated signal controller, the trained fuzzy controller, and the untrained fuzzy controller.

The simulation results of the three control systems under the five traffic cases are shown in Tables 7 to 11. In a whole, the fuzzy controller outperforms the conventional actuated-coordinated control scheme in dealing with stochastic demands from side streets and non-coordinated approaches.

Figure 37 and 38 show the reductions of the average delay and the average number of stops per vehicle for the three traffic signal control systems.

Table 7. Simulation Results for the 80% of the original volumes.

	Statistics	**AC**	**UFL**	**TFL**
Delay (s/veh)	Average	23.53	23.03	22.91
	Stdev	1.31	1.35	1.37
Stops per vehicles	Average	0.79	0.76	0.75
	Stdev	0.03	0.04	0.03
Average speed (mph)	Average	26.87	26.97	26.99
	Stdev	0.28	0.29	0.29
Stopped delay (s/veh)	Average	14.13	13.79	13.69
	Stdev	0.98	1.05	1.07

Note: AC is actuated-coordinated controller:

UFL is untrained fuzzy controller; and

TFL is trained fuzzy controller.

Table 8. Simulation Results for the 90% of the original volumes.

	Statistics	AC	UFL	TFL
Delay (s/veh)	Average	26.36	25.70	25.36
	Stdev	1.76	1.63	1.65
Stops per vehicles	Average	0.86	0.83	0.83
	Stdev	0.04	0.04	0.04
Average speed (mph)	Average	26.29	26.41	26.48
	Stdev	0.36	0.34	0.34
Stopped delay (s/veh)	Average	15.98	15.56	15.26
	Stdev	1.29	1.23	1.24

Table 9. Simulation Results for the original volumes.

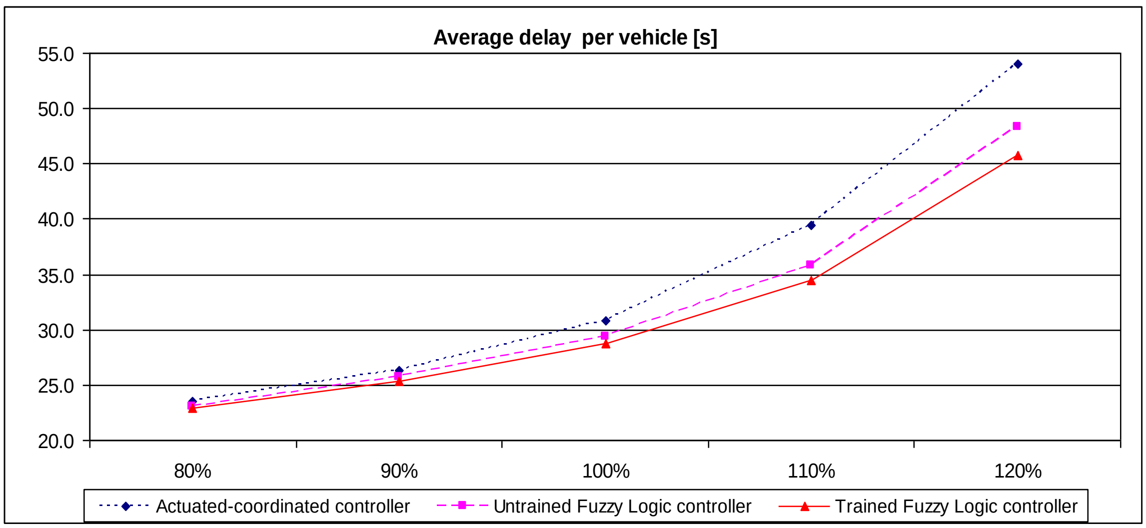
	Statistics	AC	UFL	TFL
Delay (s/veh)	Average	30.85	29.38	28.70
	Stdev	2.85	2.20	1.96
Stops per vehicles	Average	0.96	0.92	0.90
	Stdev	0.07	0.05	0.05
Average speed (mph)	Average	25.44	25.69	25.80
	Stdev	0.52	0.41	0.39
Stopped delay (s/veh)	Average	19.10	18.11	17.59
	Stdev	2.09	1.63	1.48

Table 10. Simulation Results for the 110% of the original volume.

	Statistics	AC	UFL	TFL
Delay (s/veh)	Average	39.44	35.75	34.50
	Stdev	7.36	4.78	3.69
Stops per vehicles	Average	1.12	1.04	1.01
	Stdev	0.16	0.10	0.07
Average speed (mph)	Average	24.01	24.57	24.76
	Stdev	1.07	0.75	0.62
Stopped delay (s/veh)	Average	24.78	22.50	21.73
	Stdev	3.96	3.01	2.56

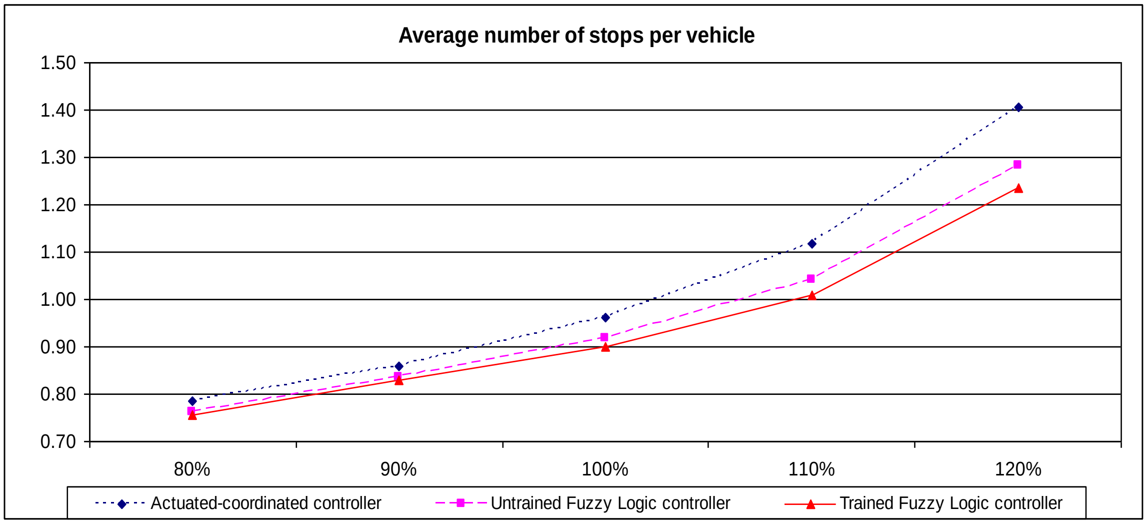
Table 11. Simulation Results for the 120% of the original volumes.

	Statistics	AC	UFL	TFL
Delay (s/veh)	Average	54.00	48.30	45.80
	Stdev	21.36	15.51	13.78
Stops per vehicles	Average	1.41	1.28	1.23
	Stdev	0.44	0.32	0.28
Average speed (mph)	Average	22.14	22.78	23.11
	Stdev	2.38	1.85	1.72
Stopped delay (s/veh)	Average	31.29	29.45	27.93
	Stdev	7.51	6.32	5.85



Reduced Avg. delay	**Case 80%**	**Case 90%**	**Case 100%**	**Case 110%**	**Case 120%**
Untrained FL	2.1%	2.5%	4.8%	9.4%	10.5%
Trained FL	2.6%	3.8%	7.0%	12.5%	15.2%

Figure 37. Average delay for the actuated-coordinated controller, the untrained fuzzy controller and the trained fuzzy controller.



Reduced avg. number of stops	Case 80%	Case 90%	Case 100%	Case 110%	Case 120%
Untrained FL	3.0%	2.8%	4.7%	6.9%	8.9%
Trained FL	3.9%	3.6%	6.5%	9.6%	12.2%

Figure 38. Average number of stops per vehicle for actuated-coordinated controller, the untrained fuzzy controller and the trained fuzzy controller.

In Figure 37 and 38, the fuzzy controller reduced the average delay and number of stops per vehicle comparing to the actuated-coordinated controller under all five traffic conditions especially under heavier traffic volumes. The trained fuzzy controller outperforms the conventional actuated-coordinated controller and the untrained fuzzy controller for all the five different traffic cases. Although the training was conducted under the original traffic condition, the trained fuzzy controller improves traffic operations universally. Under the original traffic volume, the untrained fuzzy controller reduced the average delay and the average number of stops per vehicles by 4.8% and 4.7% respectively; the trained fuzzy controller reduced the average delay and the average number of stops per vehicles by 7% and 6.5% respectively. As traffic volume increases, the benefit of the fuzzy controller becomes more pronounced and the reductions double (15.2% and 12.2% of reduction in average delay and stops respectively) when traffic volume increases to 120%. This is because with the increase of traffic volume from side streets and at left turn approaches, the traffic is usually more likely to causes congestions. The conventional actuated-coordinated controller has a limited capability of handling the fluctuating traffic flow, so queue spillbacks phenomena and congestions occur more frequently under this condition. Whereas the fuzzy controller is devised to observe the demand, with the benefit of strategically placed detectors, from longer distance upstream of the intersection, predict the arriving traffic in advance, and provide appropriate signal control strategies to reduce the frequency of queue spillbacks and congestion an, hence, average delay and stops. From Figure 37 and 38, the trained fuzzy controller after does outperform the conventional actuated-coordinated controller and the untrained fuzzy controller.

The statistical comparisons with the paired T-test results, see Table 12, indicate that under all of the five different traffic volume conditions, the untrained fuzzy controller outperforms the actuated-coordinated controller.

With respect to the case of 80% of original traffic volume, the fuzzy controller after learning does not work significantly better than the fuzzy controller before learning in terms of average delay, average speed, and average stopped delay, but yielded less average number of stops. So generally speaking, the reinforcement learning did not show its benefit significantly under this traffic condition. It is because the traffic volumes are low, the actuated-coordinated controller already works well and the fuzzy controller before/after training does not show its benefit significantly.

For the other four traffic cases, i.e., 90%, original, 110% and 120% volume cases, the fuzzy controller before learning works better than the actuated-coordinated controller, and the fuzzy controller after learning works better before learning.

From the comparison results, it can be concluded that the fuzzy controller with reinforcement learning works the best.

For detailed MOEs for simulation runs with 30 random seeds, please refer to Table 13 to 22 in APPENDIX.

Table 12. MOEs comparison results.

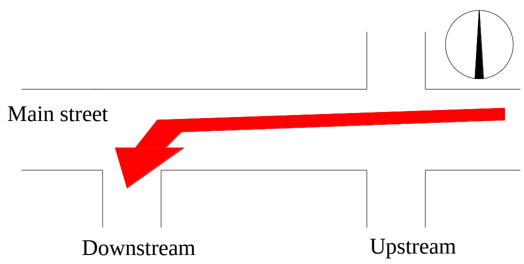
	Comparison Results and P-values	Comparison Results and P-values
80%		
Delay (s/veh)	AC > UFL P <.0001	UFL = TFL P=0.2100
Stops /veh	AC > UFL P<.0001	UFL > TFL P=0.0244
90%		
Delay (s/veh)	AC > UFL P <.0001	UFL > TFL P= 0.0060
Stops /veh	AC > UFL P<.0001	UFL = TFL P= 0.1090
100%		
Delay (s/veh)	AC > UFL P <.0001	UFL > TFL P=0.0017
Stops /veh	AC > UFL P<.0001	UFL > TFL P= 0.0072
110%		
Delay (s/veh)	AC > UFL P <.0001	UFL > TFL P= 0.0321
Stops /veh	AC > UFL P<.0001	UFL > TFL P=0.0245
120%		
Delay (s/veh)	AC > UFL P=0.0013	UFL > TFL P=0.0461
Stops /veh	AC > UFL P=0.0011	UFL > TFL P=0.0737

Note: AC is actuated-coordinated controller:

UFL is untrained fuzzy controller; and

TFL is trained fuzzy controller.

Travel time of three movements - the west-bound through movement, the east-bound through movement, and the left-turning movement from upstream west-bound to downstream south-bound movement - were also analyzed to evaluate the benefit and impact of the neurofuzzy controller, see Figure 39 to 41. The travel time of the west-bound through movement and the left-turning movement from the upstream intersection is reduced significantly. The reduction of the left-turn travel time is because the neurofuzzy controller postponed the coordinated green phase so that the left-turn phase at the downstream intersection does not return too early so that the upstream traffic could reach and join the left turn traffic at the downstream intersection and be served, and subsequently reduced the delay and number of stops. The west-bound travel time is also reduced because the neurofuzzy controller reduced the frequency of queue spillbacks, and hence reduced the chance of congestions blocking the upstream through lanes. The east-bound travel time did not increase significantly, which indicated that the postponement of the coordinate green phase did not negatively affect the east-bound through traffic along the arterial. In a whole, the neurofuzzy control system significantly reduced the variation caused by early-return-to-green, subsequently reduced delay and number of stops without losing the benefit of coordination of the conventional control.



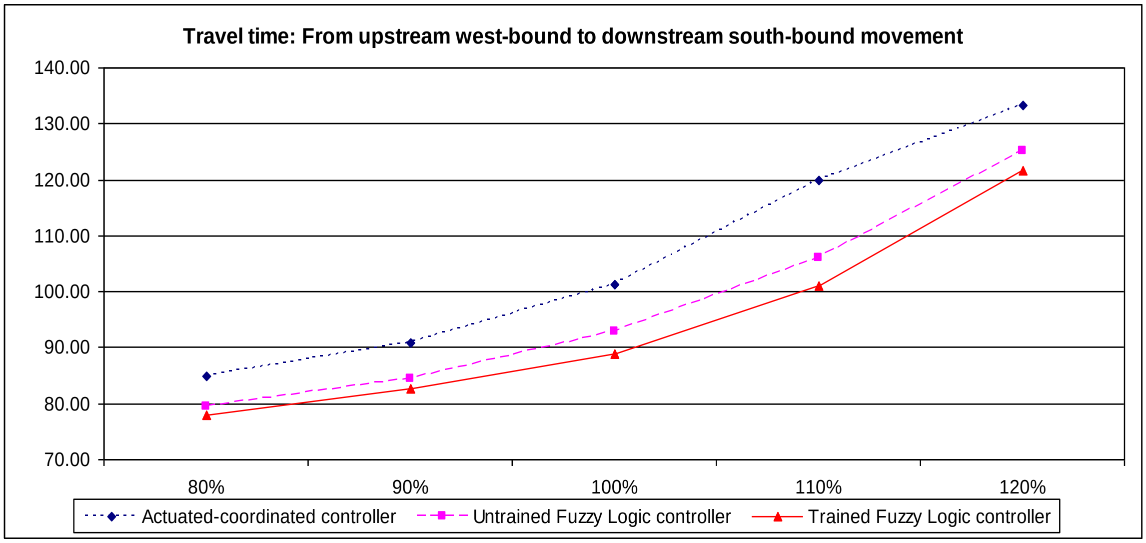
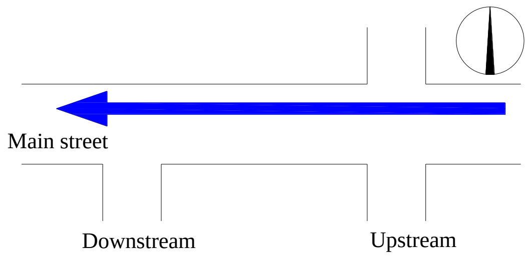


Figure 39. Travel time from upstream west-bound to downstream south-bound movement.



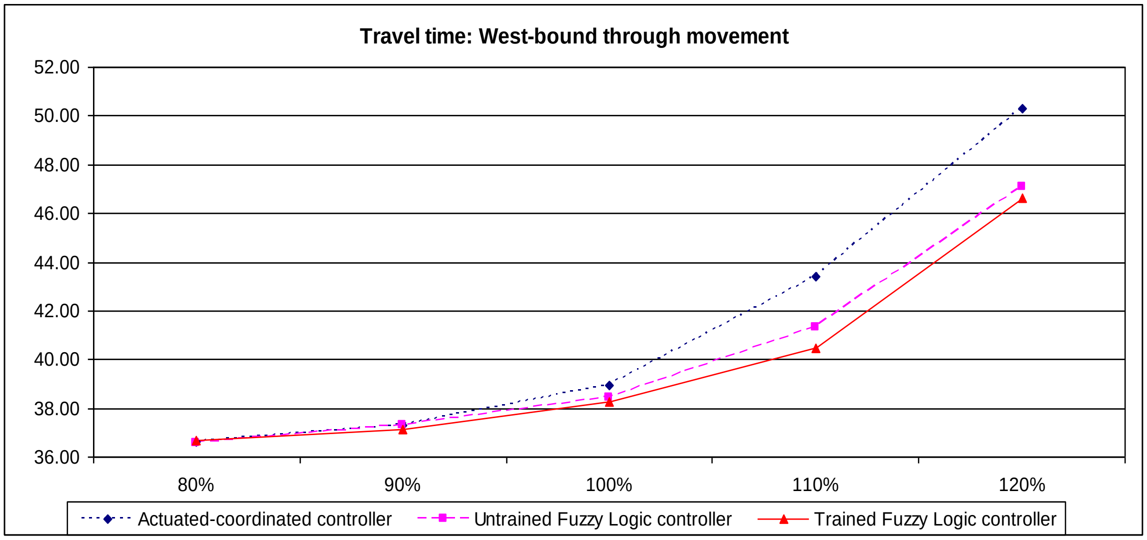
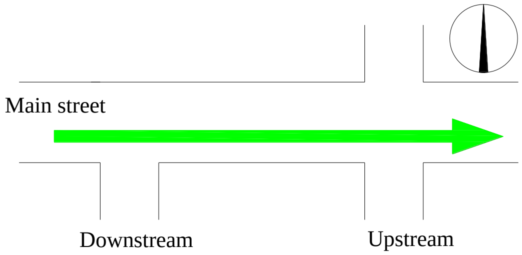


Figure 40. Travel time of west-bound through movement.



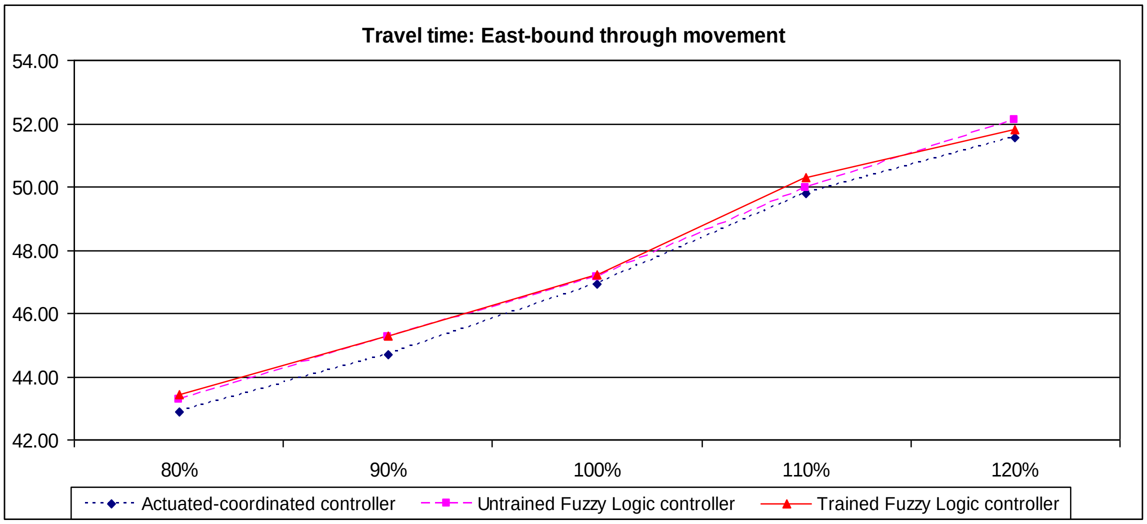


Figure 41. Travel time of east-bound through movement.

Summary

The neurofuzzy traffic signal control system was evaluated using the microscopic simulation program VISSIM. The neurofuzzy controller and reinforcement learning algorithm was coded using MATLAB which interacts with the VISSIM model via VISSIM’s COM interface.

The untrained and trained fuzzy controllers were compared with the conventional actuated-coordinated controller under five different traffic cases. For these five cases, the difference of volumes are from side streets and left turn approaches while traffic volumes of the arterial through and right turning movements keep constant. The traffic models, including geometric characters of the intersections, distributions of vehicles’ size, speed, acceleration, etc., were identical for all the three systems when doing comparisons. 30 runs with different random seeds were conducted to collect sufficient samples.

Comparing to the conventional actuated-coordinated control at closely-spaced intersections, the fuzzy control works equally well as or better than the conventional vehicle actuated traffic signal control. The fuzzy control reduced the average delay and number of stops per vehicle by 7% and 6.5% and reduces the stochastic fluctuation in the traffic flows. The variation of signal times in the cycle was reduced.

The reinforcement learning algorithm adjusted the membership functions successfully under the original observed traffic volume. In general, the adjusted membership functions can be applied under other traffic volumes and can improve the traffic operation in terms of the average delay, average number of stops per vehicle, average speed, and average stopped delay, especially under heavier traffic volumes.

The neurofuzzy traffic signal controller with reinforcement learning works better than the conventional controller and the fuzzy controller without learning.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

This research is a valuable attempt in applying fuzzy logic to address the stochastic variation of timings in the signal cycle caused by early-return-to-green when traffic demand from non-coordinated approaches is low and fluctuating in order to reduce queue spillbacks and congestions at closely-spaced intersections. Fuzzy logic provided many advantages in developing the intelligent signal control model. The following conclusions can be drawn from the results of this research:

1. The illustration of the model in Chapter III clearly showed the logic and benefits of the neurofuzzy signal control model. The model addresses technical difficulties inherent in existing pre-timed (i.e., fix-timed and actuated) signal control schemes, such as the lack of flexibility in responding to real-time traffic fluctuation and the lack of coordination between non-coordinated phases and coordinated phases. As mentioned before, the reason of queue spill-backs is an unexpected early return of the left turn phase; the neurofuzzy control system builds on the conventional actuated-coordinated control scheme to address this issue successfully. The fuzzy controller inherits the coordination designed within the conventional controller, meanwhile coordinates the downstream left turn movement with the upstream movements using fuzzy inference system so that the left turning traffic can be served at the appropriate time in the cycle, and hence reduces

the stochastic fluctuation caused by signals. Fuzzy logic is a promising approach for developing signal control system at closely-spaced intersections.

2. The fuzzy signal control system was tested under a laboratory setting, and reduced the average delay and the average number of stops per vehicle by 7% and 6.5% according to simulation running with investigated traffic volume. This fuzzy controller can be improved through a calibration process using reinforcement learning method to further its performance.

Recommendations

This research developed a fuzzy signal control system to address stochastic variations of traffic signal times in the cycle at closely-spaced intersections. Further research is recommended to enhance the model as follows:

1. The data is collected off-line, and the calibration is also based on the off-line traffic data. In other words, the traffic volume is consistent during the training process. At real intersections, the traffic flow changes in time of day and day of the week, etc. It is recommended to develop an algorithm that can calibrate the fuzzy controller under real traffic flow in real-time or different membership functions for peak-hour and off peak-hours.
2. The calibration of the fuzzy controller is only aim at membership functions. It is recommended to try calibrating the fuzzy rules also, because better fuzzy rules could yield better performance.
3. In this research, only the internal left-turn movement at one intersection (which is so called “downstream intersection”) is addressed; the internal left

turn movement at the upstream intersection was not considered. Further research can address the internal left-turn movements at both intersections.

4. In this research, the assumption is that the detectors measure queue lengths accurately and no measure errors were considered. Further research can test the robustness of the fuzzy control and the reinforcement learning ability when detection is not accurate.

LIST OF REFERENCES

[1] C. P. Pappies and Ibrahim H. Mamdani, *A Fuzzy Controller For a Traffic Junction,* IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-7, No. 10, October 1977, pp. 707-717.

[2] Jarkko Niittymaki and Shinya Kikuchi, *Application of Fuzzy Logic to the Control of a Pedestrian Crossing Signal*, Transportation Research Record 1651, 1998, pp. 30-38.

[3] Mohamed B. Trabia, Mohamed S. Kaseko, and Murali Ande, *A Two-Stage Fuzzy Controller for Traffic Signals*, Transportation Research, Part C, Vol. 7, 1999, pp. 353-367.

[4] Jarkko Niittymaki, *Using Fuzzy Logic to Control Traffic Signals at Multi-Phase Intersections*, Proceedings of the 6th International Conference on Computational Intelligence, Theory and Applications: Fuzzy Days, 1999, pp. 354-362.

[5] Kok Khiang Tan, Marzuki Khalid and Rubiyah Yusof, *Intelligent Traffic Lights Control by Fuzzy Logic,* Malaysian Journal of Computer Science, Vol. 9 No. 2, December 1996, pp. 29-35.

[6] Jongwan Kim, *A Fuzzy Logic Control Simulator for Adaptive Traffic Management*, IEEE, 1997, pp. 1519-1524.

[7] Min Chee Choy, Ruey Long Cheu, Dipti Srinivasan, and Filippo Logi. *Real-Time Coordinated Signal Control through Use of Agents with Online Reinforcement Learning*, Transportation Research Record 1836, 2003, pp64-75.

[8] Jarkko Niittymaki, *Fuzzy Logic Two-Phase Traffic Signal Control for Coordinated One-way Streets*, 1999 IEEE Midnight-Sun Workshop on Soft

Computing Methods in Industrial Applications, Kuusamo, Finland, June 16 - 18,1999, pp. 69-74.

[9] Y. Sazi Murat, and Ergun Gedizlioglu, *A Fuzzy Logic Multi-Phased Signal Control Model for Isolated Junctions*, Transportation Research Part C 13, 2005, pp. 19–36.

[10] Transportation Research Board, *Highway Capacity Manual Chapter 26*, National Research Council, Transportation Research Board, Washington, D. C. 2000.

[11] Lin Zhang, Honglong Li, and Panos D. Prevedouros, *Signal Control for Oversaturated Intersections Using Fuzzy Logic*, TRB 2005 Annual Meeting CD-ROM.

[12] M. Nakatsuyama, H. Nagahashi, and N. Nishizuka, *Fuzzy Logic Phase Controller for Traffic Functions in the One-way Arterial Road*, in Proceedings of the IFAC 9th Triennial World Congress, Pergamon Press, Oxford, 1984, pp. 2865-2870.

[13] Jurgen Branke, Peter Goldate, and Holger Prothmann, *Actuated Traffic Signal Optimization using Evolutionary Algorithms*, In Proceedings of the 6th European Congress and Exhibition on Intelligent Transport Systems and Services (ITS07). ERTICO - ITS Europe, 2007.

[14] Carroll J. Messer, *Extension and Application of Prosser-Dumme Model to Traffic Operation Analysis of Oversaturated, Closely Spaced Signalized Intersections*, Transportation Research Record 1646, 1998, pp. 106-114.

[15] Vijay G. Kovvali, Carroll J. Messer, Naddem A. Chaudhary, and Chi-leung Chu, *Program for Optimizing Diamond Interchanges in Oversaturated Conditions*, Transportation Research Record 1811, 2002, pp. 166-176.

[16] Zong Tian, Carroll J. Messer, and Kevin Balke, *Modeling Impact of Ramp Metering Queues on Diamond Interchange Operations*, Transportation Research Record 1867, 2004, pp. 172-182.

[17] Roelof J. Engelbrecht, Steven P. Venglar, and Zong Z. Tian, *Improving Diamond Interchange Operations Using Advanced Controller Features*, Project Report 0-4158, Texas Transportation Institute.

[18] Nadeem A. Chaudhary and Chi-leung Chu , *Timing Diamond Interchanges and Closely Spaced Signals,* Project Report 7-4913, Texas Transportation Institute.

[19] Zong Tian, Thomas Urbanik II, and Reed Gibby, *Application of Diamond Interchange Control Strategies at Closely Spaced Intersections*, Transportation Research Record 2035, 2007, pp. 32-39.

[20] Fang Clara Fang and Lily Elefteriadou, *Development of an Optimization Methodology for Adaptive Traffic Signal Control at Diamond Interchanges*, Journal of Transportation Engineering, August 2006, pp. 629-637.

[21] F. Lin, D. Cooke, and S. Vijaykumar, *Use of Predicted Vehicle Arrival Information for Adaptive Signal Control – an Assessment,* Transportation research board annual meeting, 1987.

[22] H. G. Rosdolsky, *A Method for Adaptive Traffic Control*, Transportation Research, Vol. 7, pp. 1-16. Pergamon Press 1973. Printed in Great Britain.

[23] A. G. Sims and K. W. Dobinson, *The Sydney Coordinated Adaptive Traffic (SCAT) System Philosophy and Benefits,* IEEE Transactions on Vehicular Technology, Vol. VT-29, No.2, May 1980, pp. 130-137.

[24] Stephen Chiu and Sujeet Chand, *Adaptive Traffic Signal Control Using Fuzzy Logic*, Fuzzy Systems, Proc. 2nd IEEE International Conference on Fuzzy systems, Vol.2, pp. 1371-1376, San Francisco, CA, US, 1993.

[25] Pitu Mirchandani and Larry Head, *A Real-Time Traffic Signal Control System: Architecture, Algorithms, and Analysis*, Transportation Research Part C, 2001, pp. 415-432.

[26] M. Patel and N. Ranganathan, *IDUTC: An Intelligent Decision-Making System for Urban Traffic-Control Applications*, IEEE Transactions on Vehiclar Technology, Vol. 50, No. 3, pp. 816-829, May 2001.

[27] Kevin Fehon, *Adaptive Traffic Signals – Are We Missing the Boat?* ITE District 6 2004 annual meeting.

[28] Carroll J. Messer, *Simulation Studies of Traffic Operations at Oversaturated, Closely Spaced Signalized Intersections*, Transportation Research Record 1646, 1998, pp. 115-123.

[29] Larry E. Owen and Charlie M. Stallard, *Rule-Based Approach to Real-Time Distributed Adaptive Signal Control*, Transportation Research Record 1683, 1999, pp. 95-101.

[30] X.-H. Yu and W. W. Recker, *Stochastic Adaptive Control Model for Traffic Signal Systems*, Transportation Research Part C 14, 2006, pp. 263-282.

[31] Chih-Hsun Chou, Jen-Chao Teng, *A Fuzzy Controller for Traffic Junction Signals*, Information Sciences 143, 2002, pp. 73-97.

[32] Yetis Sazi Murat and Ergun Gedizlioglu, *A New Approach for Fuzzy Traffic Signal Control*, Proceedings of the 13th Mini-EURO Conference, June 2002.

[33] Jarkko Niittymaki, *Installation and Experiences of Field Testing a Fuzzy Signal Controller*, European Journal of Operational Research 131, 2001, pp. 273-281.

[34] Jee-Hyong Lee and Hyung Lee-Kwang, *Distributed and Cooperative Fuzzy Controllers for Traffic Intersections Group*, IEEE Transaction on Systems, Man, and Cybernetics – Part C: Application and Reviews, Vol. 29, No. 2, May 1999, pp. 263-271.

[35] Jarkko Niittymaki and Matti Pursula, *Signal Control Using Fuzzy Logic*, Fuzzy sets and systems 116, 2000, pp. 11-22.

[36] Brungkyu Park, Carroll J. Messer, and Thomas Urbanik II, *Traffic Signal Optimization Program for Oversaturated Conditions - Genetic Algorithms Approach*, Transportation Research Record 1683, 1999, pp. 133-142.

[37] Ella Bingham, *Reinforcement Learning in Neurofuzzy Traffic Signal Control*, European Journal of Operational Research 131, 2001, pp. 232-241.

[38] J. Y. K. Luk, *Two Traffic-Responsive Area Traffic Control Methods: SCAT and SCOOT*, Traffic Engineering & Control 25, 1984, pp. 14-22.

[39] P. B. Hunt, *The SCOOT On-Line Traffic Signal Optimization Technique,* Traffic Engineering and Control, 1982, pp. 190-192.

[40] Nathan H. Gartner, *OPAC: A Demand-Responsive Strategy for Traffic Signal Control*, Transportation Research Record 906, 1983, pp. 75-81.

[41] Nathan H. Gartner, Farhad J. Pooran, and Christina M. Andrews, *Implementation of the OPAC Adaptive Control Strategy in a Traffic Signal Network*, Proceedings of the 2001 IEEE Intelligent Transportation Systems Conference, Oakland (CA), USA, August 2001, pp. 25-29.

[42] Nathan H. Gartner, Farhad J. Pooran, and Christina M. Andrews, *Optimized Policies for Adaptive Control Strategy in Real-Time Traffic Adaptive Control Systems*, Transportation research record 1811, 2002, pp. 148-156.

[43] Suvrajeet Sen and K. Larry Head, *Controlled Optimization of Phases at an Intersection*, Transportation Science, Vol. 31, No. 1, pp5-17, February 1997.

[44] Ambrosino, G., Bielli, M., Boero, M., 1994. *Artificial Intelligence Approach to Road Traffic Control*, In: Bielli, M., Ambrosino, G., Boero, M., Artificial intelligence applications to traffic engineering. VSP, Utrecht, pp.95-116.

[45] Zong Tian, Carroll Messer, Kevin Balke, and Thomas Urbanik, *Integration of Diamond Interchange and Ramp Metering Operations*, Transportation Research Record 1925, 2005, pp. 101-111.

[46] Carroll J. Messer, Daniel B. Fambro, and Stephen H. Richards, *Optimization of Pretimed Signalized Diamond Interchanges*, Transportation Research Record 644, 1997, pp. 78-84.

[47] Roelof J. Engelbrecht and Kirk E. Barnes, *Advanced Traffic Signal Control for Diamond Interchanges*, Transportation Research Record 1856, 2003, pp. 231-238.

[48] J. Favilla, A. Machion, and F. Gomide, *Fuzzy Traffic Control: Adaptive Strategies*, IEEE, 1993, pp. 506-510.

[49] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, the MIT Press, Cambridge, Massachusetts, London, England, 1998, ISBN: 0262193981. http://www.cs.ualberta.ca/~sutton/book/ebook/the-book.html

[50] Hamid R. Berenji and Pratap Khedkar, *Learning and Tuning Fuzzy Controllers Through Reinforcements*, IEEE Transactions on Neural Networks, Vol. 3, No. 5, September 1992.

[51] Detlef Nauck, Frank Klavonn, and Rudolf Kruse, *Fundations of Neuro-Fuzzy Systems*, the John Wiley & Sons Press, 1997. ISBN: 0-471-97151-0.

[52] *Fuzzy Logic Toolbox For Use with MATLAB®*, The Math Works Inc.

[53] *Neural Network Toolbox™ 6 User's Guide*, The Math Works Inc.

[54] Guanrong Chen and Trung Tat Pham, *Introduction to Fuzzy Systems*, Chapman & Hall/CRC, Taylor & Francis Group, 2006. ISBN: 1-58488-531-9.

[55] *Signalized Intersections: Informational Guide*, US department of transportation, Federal highway administration, publication No: FHWA-HRT-04-091.

[56] *VISSIM 4.30 User Manual*, PTV America, Inc.

[57] *VISSIM 4.30-00 COM Interface Manual*, PTV America, Inc.

[58] *MATLAB® 7*, the Math Works Inc.

APPENDIX

RESULTS FROM MODEL RUN

Case 1: 80% of the original traffic volume

Table 13. Delay and Stops for 30 seeds – 80% of the original traffic volumes.

	Average delay (s/veh)			**Stops per vehicles**		
Seeds	**Coord Actuated**	**Untrained FL**	**Trained FL**	**Coord Actuated**	**Untrained FL**	**Trained FL**
1	25.0	24.761	24.137	0.822	0.804	0.772
2	24.6	23.891	23.468	0.782	0.743	0.739
3	22.1	21.212	21.339	0.756	0.71	0.705
4	22.6	22.195	22.772	0.754	0.728	0.748
5	22.89	22.714	22.38	0.737	0.732	0.727
6	23.08	22.775	21.787	0.754	0.73	0.714
7	23.273	21.527	21.609	0.832	0.788	0.784
8	23.733	22.966	22.671	0.81	0.775	0.757
9	23.384	23.161	22.998	0.813	0.803	0.775
10	22.934	23.566	22.763	0.78	0.79	0.779
11	24.019	23.491	23.599	0.805	0.784	0.764
12	22.767	21.344	21.248	0.762	0.695	0.69
13	22.284	22.193	22.057	0.744	0.741	0.736
14	24.641	23.441	22.599	0.807	0.761	0.752
15	25.377	24.463	24.256	0.835	0.805	0.802
16	21.005	21.014	21.137	0.738	0.737	0.755
17	23.614	21.979	22.897	0.796	0.726	0.746
18	22.086	21.516	21.982	0.758	0.723	0.741
19	22.076	21.702	21.702	0.75	0.723	0.737
20	24.02	23.906	23.39	0.833	0.821	0.788
21	26.191	26.534	26.42	0.834	0.823	0.806
22	24.293	23.928	24.208	0.819	0.8	0.809
23	24.76	23.714	24.174	0.781	0.742	0.748
24	22.013	22.062	22.006	0.761	0.75	0.743
25	23.045	22.688	21.698	0.78	0.77	0.716
26	22.294	21.991	21.706	0.778	0.753	0.738
27	24.838	24.652	25.038	0.81	0.81	0.806
28	25.059	24.868	24.633	0.786	0.791	0.792
29	25.769	24.716	25.326	0.821	0.793	0.775
30	22.061	21.782	21.397	0.727	0.713	0.699
Averag	**23.53**	**23.03**	**22.91**	**0.79**	**0.76**	**0.75**

e						
Stdev	**1.31**	**1.35**	**1.37**	**0.03**	**0.04**	**0.03**

Table 14. Speed and Stopped Delay for 30 seeds – 80% of the original volumes.

	Average speed (mph)			**Stopped delay (s/veh)**		
seeds	**Coord Actuated**	**Untrained FL**	**Trained FL**	**Coord Actuated**	**Untrained FL**	**Trained FL**
1	26.57	26.607	26.721	15.052	15.091	14.657
2	26.651	26.789	26.872	15.292	14.88	14.552
3	27.235	27.406	27.38	13.071	12.363	12.398
4	27.118	27.186	27.082	13.589	13.204	13.564
5	27.039	27.068	27.134	14.058	14.002	13.74
6	26.944	27.003	27.19	14.073	13.855	12.952
7	26.908	27.239	27.245	13.51	12.42	12.531
8	26.819	26.969	27.019	14.069	13.495	13.415
9	26.866	26.912	26.944	13.776	13.618	13.562
10	27.008	26.89	27.04	13.744	14.163	13.312
11	26.728	26.824	26.799	14.436	14.025	14.202
12	27.032	27.297	27.312	13.443	12.457	12.492
13	27.144	27.166	27.188	13.409	13.47	13.172
14	26.691	26.914	27.071	14.914	14.088	13.281
15	26.413	26.587	26.613	15.71	14.976	14.734
16	27.365	27.365	27.34	12.323	12.183	12.248
17	26.927	27.233	27.063	14.063	12.915	13.649
18	27.192	27.295	27.209	12.941	12.628	12.792
19	27.169	27.24	27.24	13.148	12.896	12.81
20	26.744	26.766	26.86	14.203	14.174	13.784
21	26.308	26.255	26.273	16.137	16.494	16.363
22	26.634	26.714	26.646	14.657	14.467	14.572
23	26.606	26.806	26.721	15.073	14.523	14.67
24	27.226	27.212	27.216	12.848	13.013	12.904
25	26.983	27.046	27.222	13.898	13.576	12.987
26	27.117	27.174	27.229	12.888	12.716	12.584
27	26.666	26.688	26.628	15.437	15.244	15.503
28	26.54	26.569	26.61	15.003	14.936	14.826
29	26.449	26.64	26.527	15.639	14.814	15.667
30	27.144	27.209	27.285	13.359	12.949	12.733
Average	**26.87**	**26.97**	**26.99**	**14.13**	**13.79**	**13.69**
Stdev	**0.28**	**0.29**	**0.29**	**0.98**	**1.05**	**1.07**

Case 2: 90% of the original traffic volume

Table 15. Delay and Stops for 30 seeds – 90% of the original traffic volumes.

	Delay (s/veh)			Stops per vehicles		
seeds	Coord Actuated	Untrained FL	Trained FL	Coord Actuated	Untrained FL	Trained FL
1	26.5	26.494	26.033	0.817	0.817	0.801
2	27.3	26.228	25.401	0.883	0.863	0.83
3	25.6	23.868	23.619	0.842	0.797	0.785
4	25.2	24.499	24.323	0.84	0.804	0.802
5	24.34	24.451	24.417	0.838	0.822	0.815
6	25.63	25.409	25.25	0.859	0.838	0.834
7	25.102	24.719	24.15	0.847	0.834	0.808
8	26.418	26.35	24.101	0.856	0.873	0.811
9	27.574	27.463	26.73	0.886	0.906	0.869
10	26.803	24.679	24.751	0.866	0.821	0.83
11	28.166	27.744	27.776	0.873	0.862	0.866
12	26.783	26.421	24.814	0.88	0.86	0.802
13	24.46	24.51	23.614	0.794	0.803	0.782
14	24.904	24.789	24.436	0.835	0.817	0.808
15	27.645	27.297	25.874	0.889	0.838	0.828
16	23.459	22.868	22.864	0.818	0.794	0.791
17	24.944	24.061	24.108	0.818	0.782	0.781
18	24.18	23.207	23.534	0.782	0.768	0.776
19	25.254	24.952	24.499	0.841	0.809	0.809
20	27.238	26.599	25.79	0.869	0.862	0.848
21	28.603	28.13	28.409	0.896	0.881	0.901
22	29.463	28.275	28.198	0.984	0.948	0.954
23	29.563	27.518	27.244	0.928	0.862	0.884
24	23.732	23.302	23.396	0.817	0.803	0.787
25	26.547	25.051	24.615	0.876	0.797	0.796
26	26.761	26.661	26.371	0.871	0.86	0.84
27	27.139	25.777	26.098	0.852	0.836	0.859
28	28.1	27.403	28.353	0.882	0.851	0.884

29	29.489	28.317	28.181	0.938	0.872	0.9
30	24.02	23.888	23.783	0.796	0.768	0.765
Average	**26.36**	**25.70**	**25.36**	**0.86**	**0.83**	**0.83**
Stdev	**1.76**	**1.63**	**1.65**	**0.04**	**0.04**	**0.04**

Table 16. Speed and Stopped Delay for 30 seeds – 90% of the traffic volumes.

	Average speed (mph)			**Stopped delay (s/veh)**		
seeds	**Coord Actuated**	**Untrained FL**	**Trained FL**	**Coord Actuated**	**Untrained FL**	**Trained FL**
1	26.226	26.226	26.308	16.252	16.252	16.018
2	26.086	26.296	26.445	16.971	16.148	15.625
3	26.534	26.845	26.888	15.308	13.912	13.845
4	26.598	26.715	26.748	14.99	14.65	14.473
5	26.701	26.683	26.693	14.431	14.576	14.528
6	26.401	26.468	26.491	15.376	15.468	15.174
7	26.514	26.579	26.684	14.939	14.766	14.308
8	26.257	26.299	26.704	15.831	15.735	14.037
9	26.051	26.082	26.21	16.624	16.562	16.148
10	26.208	26.602	26.576	16.324	14.763	14.766
11	25.893	25.962	25.971	17.434	17.297	17.243
12	26.199	26.267	26.567	16.063	15.771	14.495
13	26.677	26.669	26.836	15.021	15.064	14.356
14	26.614	26.644	26.714	14.769	14.843	14.613
15	25.946	26.005	26.257	17.103	17.061	15.799
16	26.845	26.961	26.96	13.546	13.081	13.069
17	26.61	26.769	26.76	15.002	14.457	14.517
18	26.722	26.902	26.849	14.511	13.621	13.856
19	26.513	26.571	26.656	15.196	15.123	14.676
20	26.089	26.201	26.351	16.553	15.886	15.3
21	25.831	25.919	25.881	17.681	17.278	17.607
22	25.649	25.873	25.894	17.74	17.073	16.864
23	25.656	26.012	26.063	18.256	17.003	16.601
24	26.844	26.921	26.894	13.949	13.587	13.557
25	26.258	26.528	26.606	15.988	15.308	14.909
26	26.22	26.24	26.296	16.228	16.38	16.04
27	26.173	26.423	26.348	16.924	15.752	15.878
28	25.952	26.069	25.902	17.433	16.884	17.553
29	25.737	25.937	25.959	18.356	17.613	17.327
30	26.697	26.738	26.75	14.712	14.74	14.608

Average	**26.29**	**26.41**	**26.48**	**15.98**	**15.56**	**15.26**
Stdev	**0.36**	**0.34**	**0.34**	**1.29**	**1.23**	**1.24**

Case 3: The original traffic volume

Table 17. Delay and Stops for 30 seeds –the original traffic volumes.

	Delay (s/veh)			Stops per vehicles		
seeds	**Coord Actuated**	**Untrained FL**	**Trained FL**	**Coord Actuated**	**Untrained FL**	**Trained FL**
1	32.2	31.333	31.676	0.994	0.98	0.978
2	31.6	31.632	29.506	0.934	0.931	0.879
3	27.1	27.131	24.89	0.852	0.864	0.79
4	26.0	25.949	25.275	0.855	0.867	0.836
5	29.04	27.549	26.508	0.913	0.886	0.853
6	29.49	28.827	27.923	0.939	0.909	0.86
7	28.046	27.58	27.7	0.903	0.899	0.89
8	28.689	26.995	27.459	0.938	0.858	0.878
9	31.257	27.497	27.57	0.966	0.849	0.863
10	33.947	30.994	29.736	1.052	0.935	0.927
11	31.116	28.893	29.858	0.994	0.924	0.947
12	32.552	29.686	29.746	1.026	0.939	0.97
13	28.981	28.05	27.865	0.898	0.868	0.855
14	32.049	29.289	29.269	1.012	0.885	0.923
15	36.791	32.691	31.909	1.082	0.977	0.975
16	29.305	28.627	27.92	0.903	0.873	0.866
17	28.246	28.282	27.209	0.883	0.886	0.841
18	30.145	29.072	28.727	0.934	0.89	0.892
19	27.87	27.697	27.05	0.909	0.88	0.879
20	29.579	28.789	29.089	0.959	0.934	0.948
21	37.039	32.799	32.482	1.086	0.979	0.949
22	34.598	34.688	30.396	1.066	1.062	0.957
23	31.415	29.226	30.069	0.955	0.917	0.93
24	30.952	26.736	26.196	1.006	0.859	0.85
25	31.039	29.325	28.426	0.996	0.948	0.906
26	29.046	30.672	28.35	0.918	0.942	0.911
27	29.66	27.985	27.364	0.924	0.881	0.867

28	33.557	32.739	30.758	1.014	0.997	0.951
29	36.353	33.024	32.369	1.108	0.996	0.981
30	27.649	27.657	27.584	0.841	0.877	0.836
Average	**30.85**	**29.38**	**28.70**	**0.96**	**0.92**	**0.90**
Stdev	**2.85**	**2.20**	**1.96**	**0.07**	**0.05**	**0.05**

Table 18. Speed and Stopped Delay for 30 seeds – the original volumes.

	Average speed (mph)			**Stopped delay (s/veh)**		
seeds	**Coord Actuated**	**Untrained FL**	**Trained FL**	**Coord Actuated**	**Untrained FL**	**Trained FL**
1	25.142	25.289	25.228	20.161	19.356	19.728
2	25.265	25.278	25.634	20.524	20.457	18.651
3	26.182	26.175	26.567	16.412	16.302	14.695
4	26.349	26.361	26.481	15.334	15.378	14.926
5	25.757	26.03	26.219	18.008	16.802	15.931
6	25.657	25.769	25.952	18.175	17.738	17.311
7	25.941	26.024	25.978	16.93	16.365	16.547
8	25.803	26.129	26.021	16.801	16.341	16.239
9	25.329	25.982	25.994	19.506	16.999	16.89
10	24.897	25.386	25.6	20.53	18.965	17.865
11	25.296	25.673	25.507	19.281	17.743	18.382
12	25.129	25.618	25.618	19.915	18.088	18.013
13	25.779	25.943	25.966	18.37	17.743	17.642
14	25.277	25.756	25.756	19.686	17.983	17.865
15	24.332	24.994	25.109	23.942	20.722	20.294
16	25.704	25.821	25.946	18.068	17.779	17.196
17	25.999	25.974	26.181	17.319	17.34	16.674
18	25.591	25.774	25.839	19.006	18.271	17.843
19	25.958	25.99	26.119	16.733	16.661	16.051
20	25.615	25.757	25.698	17.82	17.305	17.484
21	24.354	25.059	25.092	23.864	20.638	20.622
22	24.704	24.689	25.41	21.822	21.855	18.923
23	25.31	25.678	25.533	19.542	17.905	18.423
24	25.492	26.217	26.319	18.68	15.822	15.483
25	25.4	25.704	25.846	19.15	18.071	17.479
26	25.75	25.462	25.862	17.928	19.145	17.394
27	25.647	25.936	26.064	18.439	17.393	16.857
28	24.944	25.079	25.4	20.696	20.183	19.066
29	24.543	25.079	25.185	22.792	20.611	19.993

30	25.946	25.953	25.963	17.417	17.319	17.34
Average	**25.44**	**25.69**	**25.80**	**19.10**	**18.11**	**17.59**
Stdev	**0.52**	**0.41**	**0.39**	**2.09**	**1.63**	**1.48**

Case 4: 110% of the original traffic volume

Table 19. Delay and Stops for 30 seeds – 110% of the original traffic volumes.

	Delay (s/veh)			**Stops per vehicles**		
seeds	**Coord Actuated**	**Untrained FL**	**Trained FL**	**Coord Actuated**	**Untrained FL**	**Trained FL**
1	64.6	54.516	46.191	1.572	1.428	1.216
2	36.2	36.187	32.828	1.071	1.061	0.966
3	32.7	30.488	29.133	0.976	0.939	0.917
4	33.3	29.401	30.907	1.037	0.89	0.948
5	35.86	34.635	33.057	1.029	1.026	0.987
6	36.44	33.07	34.68	1.053	0.957	1.006
7	33.204	32.199	31.417	1.013	1.023	0.996
8	36.573	32.304	34.66	1.056	0.988	0.991
9	39.715	39.547	37.716	1.11	1.108	1.097
10	35.975	36.206	37.262	1.048	1.067	1.06
11	45.082	41.194	41.037	1.283	1.117	1.127
12	36.522	33.473	32.723	1.059	0.986	0.981
13	35.611	35.236	31.099	1.002	1.001	0.921
14	44.581	35.776	35.339	1.268	1.114	1.076
15	56.184	42.669	36.337	1.507	1.189	1.047
16	36.326	34.705	33.81	1.072	1.057	1.015
17	34.44	33.41	28.507	0.993	0.976	0.877
18	38.413	33.823	37.43	1.055	0.971	1.044
19	35.145	34.945	32.078	1.032	1.032	0.99
20	43.676	32.006	33.705	1.226	0.959	1.001
21	40.609	39.032	36.563	1.055	1.039	0.993
22	42.7	36.004	39.502	1.229	1.049	1.125
23	39.635	36.108	35.288	1.145	1.063	1.025
24	32.919	30.795	29.883	0.985	0.926	0.913
25	34.632	34.861	32.425	1.017	1.07	1.009
26	35.118	33.853	36.186	1.03	1.016	1.051

27	37.385	35.026	34.638	1.064	1.032	0.999
28	47.191	41.385	33.007	1.313	1.176	0.976
29	50.33	37.361	36.16	1.314	1.032	1.043
30	32.27	32.14	31.548	0.905	0.913	0.906
Average	**39.44**	**35.75**	**34.50**	**1.12**	**1.04**	**1.01**
Stdev	**7.36**	**4.78**	**3.69**	**0.16**	**0.10**	**0.07**

Table 20. Speed and Stopped Delay for 30 seeds – 110% of the original volumes.

	Average speed (mph)			**Stopped delay (s/veh)**		
seeds	**Coord Actuated**	**Untrained FL**	**Trained FL**	**Coord Actuated**	**Untrained FL**	**Trained FL**
1	20.668	21.887	22.992	37.049	33.22	28.349
2	24.49	24.483	25.024	23.339	23.44	20.967
3	25.101	25.485	25.709	20.401	18.494	17.507
4	25.039	25.684	25.435	20.695	18.454	19.577
5	24.573	24.765	25.021	23.046	22.105	20.852
6	24.437	24.98	24.745	23.447	21.124	22.369
7	24.97	25.158	25.285	20.573	19.665	19.056
8	24.435	25.143	24.738	23.288	19.943	21.871
9	23.88	23.911	24.214	25.731	24.817	24.465
10	24.489	24.449	24.283	22.955	22.617	23.929
11	22.996	23.563	23.59	28.913	26.473	26.849
12	24.415	24.906	25.039	23.373	21.247	20.548
13	24.569	24.637	25.316	23.219	22.76	19.862
14	23.281	24.609	24.678	26.977	21.641	21.287
15	21.562	23.369	24.332	33.24	27.258	23.419
16	24.463	24.731	24.854	23.052	21.684	21.117
17	24.808	24.978	25.845	22.137	21.254	17.406
18	24.171	24.909	24.318	25.126	21.504	23.853
19	24.64	24.675	25.149	22.156	22.021	19.854
20	23.297	25.141	24.837	26.717	19.951	20.877
21	23.763	24.014	24.411	27.162	25.735	23.935
22	23.4	24.436	23.898	27.145	23.234	25.412
23	23.884	24.465	24.561	25.174	22.852	22.424
24	25.077	25.428	25.584	20.533	18.827	18.202
25	24.671	24.645	25.056	22.021	21.975	19.929
26	24.624	24.831	24.447	21.563	20.617	22.167
27	24.292	24.668	24.72	24.27	22.424	22.216
28	22.803	23.635	24.924	28.179	25.325	20.741

29	22.431	24.307	24.498	31.293	23.79	22.97
30	25.102	25.13	25.223	20.746	20.527	19.987
Average	**24.01**	**24.57**	**24.76**	**24.78**	**22.50**	**21.73**
Stdev	**1.07**	**0.75**	**0.62**	**3.96**	**3.01**	**2.56**

Case 5: 120% of the original traffic volume

Table 21. Delay and Stops for 30 seeds – 120% of the original traffic volumes.

	Delay (s/veh)			**Stops per vehicles**		
seeds	**Coord Actuated**	**Untrained FL**	**Trained FL**	**Coord Actuated**	**Untrained FL**	**Trained FL**
1	63.2	61.169	54.185	1.559	1.596	1.392
2	40.2	36.285	35.517	1.133	1.008	1.008
3	38.6	41.006	37.112	1.122	1.161	1.075
4	44.0	38.699	34.305	1.181	1.093	0.987
5	40.11	41.196	39.665	1.186	1.183	1.17
6	46.75	40.858	41.728	1.192	1.115	1.136
7	35.158	35.838	37.342	1.046	1.079	1.119
8	37.465	41.971	38.596	1.077	1.154	1.09
9	45.637	46.43	45.426	1.21	1.196	1.207
10	117.078	103.85	85.209	2.714	2.385	2.054
11	99.087	63.06	53.988	2.342	1.54	1.364
12	42.419	39.532	37.123	1.18	1.124	1.064
13	40.461	40.417	38.082	1.098	1.116	1.088
14	50.286	46.988	49.543	1.371	1.173	1.285
15	86.658	64.644	47.103	2.11	1.705	1.261
16	40.054	40.107	38.232	1.102	1.102	1.059
17	59.954	44.514	38.163	1.503	1.225	1.097
18	48.403	46.358	46.339	1.255	1.198	1.212
19	47.033	43.601	46.678	1.294	1.23	1.28
20	46.011	49.065	41.095	1.269	1.311	1.131
21	105.203	87.538	86.083	2.426	2.096	2.088
22	67.068	55.293	55.037	1.671	1.433	1.423
23	46.348	41.812	41.904	1.198	1.121	1.104
24	37.615	33.45	34.558	1.114	0.996	1.004
25	46.228	39.355	40.915	1.292	1.114	1.189

26	55.947	41.437	41.934	1.49	1.106	1.193
27	38.989	39.987	38.478	1.072	1.156	1.111
28	67.21	58.429	75.894	1.686	1.472	1.782
29	49.977	52.874	40.36	1.299	1.332	1.116
30	36.795	33.283	33.353	1.017	0.943	0.954
Average	**54.00**	**48.30**	**45.80**	**1.41**	**1.28**	**1.23**
Stdev	**21.36**	**15.51**	**13.78**	**0.44**	**0.32**	**0.28**

Table 22. Speed and Stopped Delay for 30 seeds – 120% of the original volumes.

	Average speed (mph)			**Stopped delay (s/veh)**		
seeds	**Coord Actuated**	**Untrained FL**	**Trained FL**	**Coord Actuated**	**Untrained FL**	**Trained FL**
1	20.737	20.959	21.855	34.247	36.148	32.323
2	23.814	24.409	24.538	26.016	23.903	23.116
3	24.117	23.75	24.324	24.218	25.952	23.332
4	23.295	24.081	24.771	28.156	24.682	21.514
5	23.824	23.669	23.89	24.518	25.816	24.27
6	22.87	23.721	23.595	32.045	27.114	27.591
7	24.61	24.493	24.256	21.975	22.443	23.339
8	24.256	23.554	24.078	23.981	27.287	24.943
9	22.966	22.88	22.989	28.625	30.431	29.447
10	15.902	16.898	18.487	48.739	47.236	40.289
11	17.142	20.643	21.762	47.583	41.514	34.305
12	23.453	23.886	24.266	27.58	25.137	23.594
13	23.704	23.723	24.103	26.211	26.144	24.482
14	22.421	22.86	22.493	29.492	31.415	29.913
15	18.261	20.473	22.687	42.938	37.825	30.898
16	23.793	23.82	24.091	25.871	25.992	24.534
17	21.189	23.195	24.127	34.891	27.64	23.571
18	22.665	22.904	22.946	31.381	30.904	29.016
19	22.77	23.263	22.837	29.921	27.867	29.769
20	22.923	22.497	23.655	28.452	30.84	26.658
21	16.651	18.19	18.359	48.208	41.325	42.511
22	20.283	21.662	21.746	39.371	33.327	34.116
23	22.9	23.55	23.549	30.628	26.837	27.281
24	24.279	24.94	24.745	23.608	20.686	21.628
25	22.851	23.866	23.629	28.299	24.948	25.739
26	21.657	23.614	23.538	33.789	26.419	25.349
27	23.909	23.76	24.04	25.277	25.445	24.667

28	20.217	21.293	19.28	37.159	35.28	42.38
29	22.445	22.077	23.779	31.132	31.629	25.954
30	24.317	24.886	24.876	24.335	21.454	21.353
Average	**22.14**	**22.78**	**23.11**	**31.29**	**29.45**	**27.93**
Stdev	**2.38**	**1.85**	**1.72**	**7.51**	**6.32**	**5.85**

VITA

Xiaoli Sun was born in Xuzhou, Jiangsu province, China in April 20, 1976. She graduated from Southeast University, Nanjing, China with a Bachelor of Science degree in Civil Engineering in 1998 and a Master of Science degree in Transportation Engineering in 2001. Following graduation, Xiaoli accepted a position as a Civil Engineer with Shanghai Urban Construction Design & Research Institute, working in Shanghai, in 2001-2002, then moving to USA in 2003-2006.

In 2006, Xiaoli enrolled in the PhD program in Civil Engineering at the University of Tennessee - Knoxville. During her graduate studies, Xiaoli was employed as a graduate research assistant and graduate teaching assistant, working both on the NCHRP 3-81, Strategies for Integrated Operation of Freeway and Arterial Corridors, and NSF 0528143, Collaborative Research: Stochastic Sensing Control Models for Safe and Efficient Traffic Signal Strategies, research projects. While at the University of Tennessee, Xiaoli served as secretary of the Institute of Transportation Engineers student chapter.