Investigating the effectiveness of fuzzy logic within traffic management

A study of traffic control algorithms with applied fuzzy logic on an intersection in the south of Stockholm city

LINN BERGELID

PER NYBERG
Undersökning av effektiviteten hos oskarp logik inom trafikstyrning

En undersökning av trafikkontrolls algoritmer med applicerad oskarp logik på en korsning i den södra delen av Stockholm

LINN BERGELID AND PER NYBERG

Degree Project in Computer Science, DD143X
Supervisor: Richard Glassey
Examiner: Örjan Ekeberg

CSC, KTH 2016
Abstract

The purpose of this report is to make a comparison of how fuzzy logic rules affect the traffic flow in an intersection in the inner city of Stockholm. For the comparison, the report uses a custom written simulation to answer the question and the focus is primarily on measuring the difference in waiting time for the cars in the queues and the difference in the length of the queues. For this we have built our own fuzzy logic rules, specially adapted for the structure of the selected intersection. They were then implemented based on the current precalculated green times for the intersection. The data used for the simulation is collected by the Stockholm stad during November 2015.

Based on the work it can be concluded that the fuzzy logic rules makes the intersection more efficient, especially in terms of the average queue length which during almost the entire simulation is shorter than without the use of fuzzy logic rules. With respect to the average waiting time the result is not quite as clear, but it is possible to justify that it becomes more effective using the fuzzy logic rules. From these two statements one can fairly easily observe that using fuzzy logic can be used to streamline the flow of traffic at the intersection, and also to streamline traffic with respect to delays.
Syftet med den här rapporten är att göra en jämförelse på hur oskarpa logiska regler påverkar trafikflödet i en korsning i Stockholms innerstad. För jämförelsen använder rapporten en egenskriven simulering för att besvara frågan och riktar in sig främst på att mäta skillnaden i väntetid för bilarna i köerna samt skillnaden i längd på köerna. För detta har egna oskarpa logiska regler byggts, speciellt anpassade efter utseendet på den valda korsningen, och har därefter implementerats utifrån de förberäknade nuvarande gröntiderna som gäller i korsningen. Datan som simulationen använder sig av är framtagen av Stockholms stad i november 2015.

Utifrån arbetet kan det konstateras att de oskarpa logiska reglerna gör korsningen mer effektiv, framför allt i form av den genomsnittliga kölängden som under i princip hela simuleringen är kortare än utan de oskarpa logiska reglerna. Med avseende på den genomsnittliga väntetiden är det inte fullt lika tydligt men det går att se en övervägande motivering för att det blir mer effektivt med de oskarpa logiska reglerna. Utifrån dessa två påståenden kan det ganska enkelt konstateras att det med hjälp av oskarp logik går att effektivisera trafikflödet i korsningen och även minska förseningar.
# Contents

## Contents

1 Introduction  .................................................. 1  
1.1 Problem statement .......................................... 2  
1.2 Scope and constraints ...................................... 2  
1.3 Disposition of the report .................................. 2  

2 Background .................................................... 5  
2.1 History ......................................................... 5  
2.2 Computational intelligence .................................. 6  
2.3 Fuzzy set theory ............................................. 6  
  2.3.1 Membership function ................................. 6  
  2.3.2 Fuzzy operators .................................. 7  
2.4 Fuzzy logic systems ........................................ 8  
  2.4.1 Linguistic variables ................................. 8  
  2.4.2 Fuzzy rules .................................. 8  
2.5 Fuzzy inferencing .......................................... 8  
  2.5.1 Fuzzification .................................. 8  
  2.5.2 Inferencing .................................. 9  
  2.5.3 Defuzzification .................................. 9  
2.6 Related work .............................................. 10  

3 Method .......................................................... 13  
3.1 Data set ....................................................... 13  
3.2 Intersection .................................................. 13  
3.3 Implementation of pre-timed control system .............. 14  
3.4 Implementation of fuzzy logic control system ............ 16  
  3.4.1 Fuzzy variables and their membership functions .... 16  
  3.4.2 Fuzzy logic control rules .................................. 18  
  3.4.3 Defuzzification .................................. 19  
3.5 Simulation .................................................... 19  

4 Results .......................................................... 21  
4.1 Queue length ............................................... 21
4.2 Delay .................................................. 22

5 Discussion ................................................. 25
  5.1 Discussion of results ................................ 25
  5.2 Discussion of the interpretation of the intersection .... 26
  5.3 Discussion of the fuzzy logic ........................ 27
  5.4 Discussion of the simulation ........................ 27

6 Conclusion .................................................. 29

Bibliography .................................................. 35
Chapter 1

Introduction

In a world where vehicle traffic is becoming more intense and where people live under more crowded conditions, the need for a good traffic flow is increasing. One way to do this is to increase the amount of surfaces used, which in traffic would mean extension of current roads or construction of additional roads. Building new roads in a big city like Stockholm becomes increasingly difficult seeing how much that has been built, so the expansion of existing roads is undesirable since construction of this type is costly for society. Given a limited area like Stockholm it would be more desirable to optimize the traffic flow.

One way to optimize the traffic flow is to examine and analyze how the signaling systems that control traffic could be improved. Adjusting the existing signaling system is less expensive than building new roads and is also favourable since no roadwork is needed. About 70% of all traffic lights in Stockholm are traffic controlled. Traffic controlled in this context means that there are sensors buried under the roadway that detect vehicles and inform the connected controllers whether there is traffic incoming or not [1]. The controller can then adjust signal timing depending on this information. In cases where the traffic lights are not controlled by traffic, predetermined schedules are used and the controller is programmed to alternate between different schedules depending on what time of the day it is [1]. These type of systems will further on be referred to as pre-timed systems.

Using different schedules is effective since the traffic flow varies during the day. However, this approach carries some limitations since traffic cannot be predicted completely even if the time of the day is known. A new theory that has been tested on traffic control systems is a method called fuzzy logic which is a paradigm in the area of artificial intelligence. The main aim of using this is to make the controllers adapt to real time data by using detectors which collect the data. This way the controllers are able to optimize themselves continuously.

Mamdani et al. [2] were the first ones to implement a fuzzy logic traffic control
system. Their system was compared to a traffic controlled sensor system and the results showed better performance for the fuzzy logic controller. Other works have been done since ([3], [4], [5], [6] and [7] among others) and many of them conclude that fuzzy logic controlled systems have better performance than pre-timed and traffic controlled systems. However, this type of system has never been tested on an intersection in Sweden. Stockholm is growing in terms of population and hence the number of vehicles increases as well. [8][9] Because of this it would be interesting to study whether or not a fuzzy control system could make the traffic flow in Stockholm more efficient. This project aims to implement a fuzzy control system simulation to be tested on an intersection in Stockholm.

1.1 Problem statement

The project will be based on an intersection in the southern parts of the inner city in Stockholm in the crossing of Långholmsgatan and Hornsgatan. Based on data of the traffic volumes for that intersection we will compare a self-implemented fuzzy logic control system with a self-implemented pre-timed system. The pre-timed system is similar to the ones that are being used there today. The idea is to see which one is the most efficient and to see if the fuzzy logic control system could improve the traffic flow. We have therefore formulated our problem question thusly:

Does a fuzzy logic implementation make highly used intersections in Stockholm more efficient?

The efficiency will be measured in terms of the average queue length and the average delay for each vehicle.

1.2 Scope and constraints

This project is limited to an isolated intersection and does not pay regard to other intersections in conjunction to this one. This is due to not being able to get data for the other intersections but also to keep the project within reasonable limits for the resources given. The algorithms will be simulated using the data provided. The data is limited to afternoon rush hours only (between 15:00 and 18:00) and only includes vehicles.

1.3 Disposition of the report

The remainder of the report will be divided into five different chapters. The first chapter is a background that will cover the theory needed to understand how fuzzy logic works and to illustrate how traffic lights have evolved. The following chapter describes the method on how data for the project was collected and an in depth description of the algorithms used. The result chapter will present the result of
the simulations in forms of tables and graphs. The penultimate chapter will have a discussion on the results obtained and what factors might have affected them. Finally the report will end with a conclusion summing up the report.
Chapter 2

Background

This chapter starts by shortly covering the history of traffic control in Sweden and how it works in Stockholm today. In section 2.2-2.5 computational intelligence is covered and fuzzy logic is explained in detail. It ends with a summary of related work in section 2.6.

2.1 History

During the 1920’s, a traffic light was installed in Stockholm for the first time. Before they existed, a policeman stood in the middle of the intersection and directed the traffic using a semaphore. The increasing traffic made the introduction of traffic signals necessary and in the 1930’s automatic traffic lights started to become common. The first automatic traffic lights were controlled by a timer and could not take traffic density and variation in traffic during the day into account. This led to a lot of waiting time for the drivers despite the fact that no other vehicle was nearby. [10]

In the late 1940’s several methods for recognizing vehicles were tested. At the end of the 1950’s an inductive loop was introduced and is still being used today. [10] The purpose of the inductive loop is to recognize when a vehicle is arriving and control the traffic lights based on that. This detection system is useful for extension and termination of green time. The green time is terminated when the gap between two vehicles is larger than a decided maximum gap.

Today it is common that several intersections are coordinated to create a so called “green wave”. A green wave occurs when the coordinated traffic lights create a continuous flow over several intersections in one direction, provided that there is no other traffic waiting to cross. A coordinated traffic light often chooses one of several schedules to use on different times of the day. At night only traffic controlled lights are used because of the low flow of traffic. Some traffic signals have green light full time while others have red light full time and switch to green light when a vehicle is approaching. [1]
2.2 Computational intelligence

Recently, intelligent methods derived from computational intelligence have gained popularity for optimal traffic control. The use of machine learning techniques creates an opportunity to address unexpected situations since they have the ability to learn and adapt to new environments easily. Different areas that have been evaluated are reinforcement learning, artificial neural networks and fuzzy logic systems.

Reinforcement learning uses agents that attempt to reach a goal by trying different actions. The environment gives feedback in forms of rewards and punishment to help the agent recognize what action is the most beneficial. An experiment was carried out in 2003 in an isolated intersection and it showed that reinforcement learning is an effective approach for adaptive traffic controllers. [11]

The concept of artificial neural networks origins from biological nervous systems. Neural networks have the ability to recognize hidden patterns from complicated data and thus solve problems that cannot be solved by humans or computers. In 2006 an isolated intersection control system was developed using a combination of dynamic programming and artificial neural networks. The system made realtime decisions on whether to extend the current green time or not. The study concluded from the experimental tests that the results were nearly equal to the best solution. [12]

Fuzzy system theory is covered in chapter 2.3, 2.4 and 2.5. Earlier work on fuzzy system is covered in chapter 2.6.

2.3 Fuzzy set theory

In classic set theory an element is either a part of a set or not. This is easily implemented in a computer system which uses binary logic. Real world problems often contain vague and imprecise definitions that are difficult to translate into that kind of logic. With fuzzy set theory, an element can be a part of a set to a certain degree. Instead of using numbers it uses linguistic variables such as “there are many cars” or “the queue is very long”. This logic is closer to the way human brains work. [13]

2.3.1 Membership function

In contrast to traditional sets, fuzzy sets have membership degrees. The degree of membership specifies how certain it is that an element belongs to a specific set. Let $X$ be a defined set and $x \in X$. Then the fuzzy set $A$ is defined as a set of pairs $A = \{x, \mu_A(x)\}$ where $\mu(x)$ is the grade of membership of element $x$ in fuzzy set $A$. [14]
Thus, for all \( x \in X \), \( \mu_A \) specifies to which degree \( x \) belongs to the fuzzy set \( A \). For a binary set the only values of membership used are 0 and 1. An element can only be in the set or not. For a fuzzy set the grade of membership of an element \( x \) can be anything between 0 and 1.\[13\]

\[
\mu_A : X \rightarrow [0, 1]
\]

The membership function is often visualized using diagrams. Let fuzzy set \( A \) be defined as “queue with approximately 10 vehicles”. Then a possible membership function could be:

\[
\mu_A(x) = \begin{cases} 
0 & x < 8, x > 12 \\
\frac{x}{2} - 4 & 8 \leq x < 10 \\
6 - \frac{x}{2} & 10 \leq x \leq 12 
\end{cases}
\]

![Figure 2.1. Membership function, \( \mu_A(x) \), of fuzzy set \( A \)](image)

All elements that have a membership grade greater than zero belong to the fuzzy set \( A \). Thus, all queues of length 8 to 12 belong to the set as seen in figure 2.1. A queue of length 10 has a membership grade of 1.

It is important to notice that the membership function must be defined from 0 to 1 and that it cannot go outside these limits. For each \( x \) there is also a unique membership value, \( \mu(x) \), and each \( x \) cannot map to different membership degrees for the same fuzzy set. \[13\]

### 2.3.2 Fuzzy operators

**Complement of a fuzzy set (NOT):**

The complement of a fuzzy set \( A \) contains the same elements as \( A \) but with the
membership function:
\[ \mu_A^{-}(x) = 1 - \mu_A(x) \]

**Union of two fuzzy sets (OR):**
The union of two fuzzy sets \( A \) and \( B \) is defined as the set of elements contained in either \( A \) or \( B \) or both. Union corresponds to the operator OR.
\[ \mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \]

**Intersection of two fuzzy sets (AND):**
The intersection of two fuzzy sets \( A \) and \( B \) is defined as the set of elements contained in both \( A \) and \( B \). Intersection corresponds to the operator AND.
\[ \mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \]

### 2.4 Fuzzy logic systems

Fuzzy logic systems are systems that use fuzzy set theory for approximate reasoning. Approximate reasoning is the process of deducing an imprecise conclusion from a collection of imprecise premises. The system is built up by linguistic variables and fuzzy rules.

#### 2.4.1 Linguistic variables

Linguistic variables are variables that use words from natural language instead of numbers. Thus, linguistic variables are a great tool for approximate reasoning. An example of values that can be used are words like “many”, “few”, “long” and “short”.

#### 2.4.2 Fuzzy rules

A fuzzy system’s behavior is defined by a couple of fuzzy rules. These are generally of the form:

\[
\text{if } X = x \text{ and } Y = y \text{ then } Z = z
\]

where \( X, Y \) and \( Z \) are fuzzy sets and \( x, y \) and \( z \) are linguistic variables. [13]

### 2.5 Fuzzy inferencing

Together with the fuzzy rules and variables the system consists of a fuzzifier, an inference engine and a defuzzifier as shown in figure 2.2.

#### 2.5.1 Fuzzification

The input data to the system is most commonly real numbers. The purpose of the fuzzifier is therefore to map these numbers to fuzzy sets. This is needed so that the rules can be applied to the input. It is done by applying the membership
function of each fuzzy set onto the numbers. Assume having a fuzzy set $A$ and its corresponding membership function $\mu_A(x)$. The input to the fuzzifier would be an element $x$ where $x \in A$ and with the help of the membership function it would produce a membership grade $\mu_A(x)$. [13]

### 2.5.2 Inferencing

The purpose of the inference engine is to map the input to the fuzzy rules and produce a fuzzified output. If more than one fuzzy set is involved in the input, their relationship is described by a logic operator. The AND-operator corresponds to the minimum-function and the OR-operator corresponds to the maximum-function as described in section 2.3.2 above. Thus, if you have a rule:

$$\text{if } X = x \text{ and } Y = y \text{ then } Z = z,$$

$\mu_X(x)$ and $\mu_Y(y)$ will already be known from the fuzzification process and the intersection is calculated by taking the minimum of these values. The intersection value will be the firing strength of the rule which is the degree to which the rule is fulfilled. [13]

### 2.5.3 Defuzzification

Defuzzification is the process of finding a value for the output variable. After having calculated the firing strength for all rules in the system, the defuzzifier converts the output of the fuzzy rules to a non-fuzzy value. There are different methods for defuzzification: [13]

**Max-min method:**
The rule producing the largest firing of strength is chosen. The value is calculated by taking the horizontal coordinate of the centroid of the membership function corresponding to the output of that rule.

**The averaging method:**
The average of all firing strengths of the rules is calculated and all membership
functions are clipped at the average. Then the centroid of all clipped membership functions is calculated and the horizontal coordinate is used as output value.

**The root-sum-square method:**
Each membership function is scaled such that the peak of the function corresponds to the firing strength of that function. Then the horizontal coordinate of the centroid of the area under the scaled functions is used as output.

**The center of gravity method:**
Each membership function is clipped at the corresponding firing strength. Then the horizontal coordinate of the centroid of the area under the clipped functions is used as output.

### 2.6 Related work

Mamdani et al [2] were the first ones to implement fuzzy logic for traffic control. In 1977 they presented a fuzzy logic controller for an intersection with two one-way streets with random arrival of vehicles and no turning movements. The system had three linguistic input variables: passed time of the current interval, number of vehicles crossing during the green phase and the number of vehicles waiting from the other direction. The extension time for the green phase was the output. The results were compared to a vehicle-actuated controller and showed that the fuzzy logic controller had the best performance regardless of traffic volume.

In the beginning of the 1990's attempts were made to implement fuzzy logic systems in multi-intersection networks. The first ones to do this was Chiu and Chand. [4] Their system had multiple controllers that cooperated with each other. Each controller considered the local traffic condition and the signal timing parameters of adjacent controllers. They then used a set of fuzzy rules to adjust cycle time, phase split and offset. In 1999, Lee and Lee-Kwang [5] did a fuzzy logic controller for a group of intersections. Each intersection had its own controller that controlled its own traffic and cooperated with the other intersections. The proposed model was compared to an vehicle-actuated controller and showed a 3.5% to 4.8% improvement over the vehicle-actuated.

Two of the recent works in this field is Wenchen et al in 2012 [6] and Chiou and Huang in 2013 [7]. Wenchen et al proposed two adaptive two-stage fuzzy controllers at isolated intersections. The performance of the two models was validated using an online simulation platform and the results showed that the two proposed models offer a better performance as the flow increases compared to both pre-timed and vehicle actuated systems. In the work of Chiou and Huang a stepwise genetic fuzzy logic controller was developed. It considered traffic flow and queue lengths of cars and motorcycles to control extension of the green time to minimize total delay of
the vehicles. Comparisons were made between the fuzzy logic controller and two different pre-timed schedules and the fuzzy logic controller performed best.
Chapter 3

Method

This chapter describes the method of the project and how the algorithms were implemented and then simulated with the given data.

The study began with a literature study to get a picture of what had been done earlier in the area of traffic control combined with fuzzy logic. The study showed that a lot of research had been done and that fuzzy logic traffic control systems performed well. It had been tested mostly on theoretical intersections [2][6] but also on real intersections [3] However it had never been tested in Sweden. Therefore a single intersection in Stockholm was selected to be tested on. Getting hold of data was crucial to being able to do a realistic simulation and consequently a determining factor for the choice of the intersection used for the project. To be able to analyze the effectiveness of the fuzzy logic controller a comparison was carried out for the intersection. The fuzzy logic controller was compared to a pre-timed controller similar to the one used in the intersection today.

3.1 Data set

The data set used in this study was taken from the research done by Roadinfo Europe AB on behalf of Stockholm Stad on the 10th of November 2015. It presents the traffic flow given in vehicles per quarter for the intersection between Långholmsgatan and Hornsgatan. The collection of data took place during 06:30-09:00 and 15:00-18:00 and included data from all four directions with left, right, straight forward and u-turn movements. The vehicles counted were both lightweight- and heavyweight vehicles. [15] The data can be found in Appendix B. For this study the data for the afternoon will be used only.

3.2 Intersection

Långholmsgatan is the northern entry and has a total of four lanes: two lanes for turning left, one for straight forward and one for straight forward and right turns.
The southern entry for the intersection is Liljeholmsbron which has four lanes as well: two for driving straight forward, one for turning right and one for buses. The other two parts of the intersection are both from Hornsgatan. The western entry only has one lane which goes in all directions and the eastern entry has four lanes: two for turning left, one for going straight forward and one for turning right.

3.3 Implementation of pre-timed control system

To be able to analyse the effectiveness of the fuzzy logic control system a pre-timed control system was implemented to compare with. The aim of the pre-timed controller was to make it as similar to the real system used today as possible.

The green times could not be used directly due to the fuzzy logic needing phases to adjust the time (which will be described more in chapter 3.4). The green times (see Appendix C) therefore had to be tweaked so that we could make manageable phases that was still close enough to reality.

The pre-timed algorithm has eight different phases for green light. Down below is a summary of how the phases are structured. The direction before the arrow shows from where the traffic is coming and the directions that comes after shows where the traffic may go. E.g. in phase 2 in the green section you have North -> South, West which means that the traffic may move from north to both south and west. The opposite holds for the red section where the arrows show in which directions the cars are not allowed to move. The semicolon is used so separate the arrows from each other. The length of a phase is shown in parenthesis at the end of each phase. In the case of phase 1 it alternates between two different time-settings.

![Figure 3.1. Simplified view of the intersection](image)
all through the simulation.

The eight different phases for green and red time:

**Phase 1 (28 sec/18 sec):**
Green: North -> South, West; South -> North
Red: North -> East; East -> North, South, West; South -> East; West -> North, East, South

**Phase 2 (2 sec):**
Green: North -> South, West
Red: North -> East; East -> North, South, West; South -> North, East; West -> North, East, South

**Phase 3 (12 sec):**
Green: North -> East, South, West; East -> North
Red: East -> South, West; South -> North, East; West -> North, East, South

**Phase 4 (12 sec):**
Green: North -> East; East -> North
Red: North -> South, West; East -> South, West; South -> North, East; West -> North, East, South

**Phase 5 (12 sec):**
Green: East -> North
Red: North -> East, South, West; East -> South, West; South -> North, East; West -> North, East, South

**Phase 6 (27 sec):**
Green: East -> South, West; South -> East
Red: North -> East, South, West; East -> North; South -> North; West -> North, East, South

**Phase 7 (2 sec):**
Green: -
Red: North -> East, South, West; East -> North, South, West; South -> North, East; West -> North, East, South

**Phase 8 (7 sec):**
Green: South -> North
Red: North -> East, South, West; East -> North, South, West; South -> East; West -> North, East, South

The phases will be switched between in order. Phase 1 is followed by phase 2, phase 2 is followed by phase 3 and so on. Phase 8 is followed by phase 1. The reason for the variations in green time is that phase 2 and phase 7 are only transitions between other phases. Phase 8 is short due to that it is an extension to phase 1 placed before phase 1 (since the directions of phase 8 exists in phase 1).

The main difference between how reality's green times works and how our implementations green times works is that South -> West has two seconds added in the
beginning of it's green time and that East -> North has three seconds added in the beginning of it’s green time.

### 3.4 Implementation of fuzzy logic control system

The fuzzy logic control system is based on the same phases as the pre-timed system. What differs the fuzzy logic control system from the pre-timed is that the green time varies depending on the amount of vehicles that are arriving and the amount that are queuing.

#### 3.4.1 Fuzzy variables and their membership functions

The input variables of the fuzzy logic control system are volume (V), arrival (A) and queue (Q). V is used to consider the total amount of vehicles entering the intersection from all directions. A is used to measure the number of vehicles that have a green light during a specific phase and Q measures the number of vehicles waiting during the same phase. The output of the system will have one variable which is called time adjustment (TA).

Their membership functions will be based upon the given data and is defined as follows:

**Volume:** \{small, medium, large\}

\[
\mu_{\text{small}}(x) = \begin{cases} 
1 & x \leq 20 \\
2 - x/20 & 20 < x < 40 \\
0 & x \geq 40 
\end{cases}
\]

\[
\mu_{\text{medium}}(x) = \begin{cases} 
0 & x \leq 20, x \geq 60 \\
x/20 - 1 & 20 < x < 40 \\
3 - x/20 & 40 \leq x < 60 
\end{cases}
\]

\[
\mu_{\text{large}}(x) = \begin{cases} 
0 & x \leq 40 \\
x/20 - 2 & 40 < x < 60 \\
1 & x \geq 60 
\end{cases}
\]

**Arrival and queue:** \{zero, small, medium, many\}

Arrival and queue use the same functions and have the same values.

\[
\mu_{\text{zero}}(x) = \begin{cases} 
1 - 3x/100 & x < 100/3 \\
0 & x \geq 100/3 
\end{cases}
\]
3.4. IMPLEMENTATION OF FUZZY LOGIC CONTROL SYSTEM

Figure 3.2. Membership functions for volume. Red represents $\mu_{\text{small}}(x)$, blue represents $\mu_{\text{medium}}(x)$ and green represents $\mu_{\text{large}}(x)$

\[
\mu_{\text{small}}(x) = \begin{cases} 
3x/100 & x \leq 100/3 \\
2 - 3x/100 & 100/3 < x < 200/3 \\
0 & x \geq 200/3 
\end{cases}
\]

\[
\mu_{\text{medium}}(x) = \begin{cases} 
0 & x \leq 100/3, x \geq 100 \\
3x/100 - 1 & 100/3 < x < 200/3 \\
3 - 3x/100 & 200/3 \leq x < 100 
\end{cases}
\]

\[
\mu_{\text{many}}(x) = \begin{cases} 
0 & x \leq 200/3 \\
3x/100 - 2 & 200/3 < x < 100 \\
1 & x \geq 100 
\end{cases}
\]

**Time adjustment:** \{veryshortened, shortened, zero, extended\}

The values of the membership function for the variable time adjustment are constants.

\[\text{veryshortened} = -4\]
\[\text{shortened} = -2\]
\[\text{zero} = 0\]
\[\text{extended} = 2\]

E.g. when the output of a rule is *extended* the green phase will run for 2 seconds longer.
3.4.2 Fuzzy logic control rules

The rules of the system are divided into three groups according to the different volumes where each group has different rules. This way the system can adapt to an intersection with dynamically changing traffic volumes.

The adjusted time is calculated as follows. First the degrees of the membership functions for the volume variable are calculated. The one with the highest degree decides which group of rules to be used for the determination of the time adjustment. Secondly the degrees of membership are calculated for all rules. The degree is calculated by looking at the membership functions for arrival and queue. The degree will tell to which extent the rule is fulfilled.

Fuzzy logic control rules for **low volume group** $(V = low)$:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Adjusted Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A = \text{zero} ) and ( Q = \ast )</td>
<td>( TA = \text{veryshortened} )</td>
</tr>
<tr>
<td>( A = \text{small} ) and ( Q = \text{zero} )</td>
<td>( TA = \text{shortened} )</td>
</tr>
<tr>
<td>( A = \text{small} ) and ( Q = \text{small} )</td>
<td>( TA = \text{veryshortened} )</td>
</tr>
<tr>
<td>( A = \text{small} ) and ( Q = \text{medium} )</td>
<td>( TA = \text{veryshortened} )</td>
</tr>
<tr>
<td>( A = \text{small} ) and ( Q = \text{many} )</td>
<td>( TA = \text{veryshortened} )</td>
</tr>
<tr>
<td>( A = \text{medium} ) and ( Q = \text{zero} )</td>
<td>( TA = \text{shortened} )</td>
</tr>
<tr>
<td>( A = \text{medium} ) and ( Q = \text{small} )</td>
<td>( TA = \text{shortened} )</td>
</tr>
<tr>
<td>( A = \text{medium} ) and ( Q = \text{medium} )</td>
<td>( TA = \text{shortened} )</td>
</tr>
</tbody>
</table>
3.5 Simulation

To compare the pre-timed model and the fuzzy logic model a simulator was written. The simulation takes the data given from Stockholms stad (see Appendix B) and distributes it evenly over the interval of 15 minutes were every second gets a number dedicated to it. Since the data will not always be dividable by 900 (900 seconds = 15 minutes) the inflow may differ between two seconds by one car.

To calculate how many cars that may drive through the number for how long a car needs to leave the queue is used, [16] then the amount of cars that can drive off in one minute is calculated. A delay was used for starting when a new phase was activated due to reaction time for the driver. A random value between 0.35 seconds and 1.39 seconds was chosen which is the given standard for reaction time for drivers. [16] An algorithm was used to tell how long time it would take for a car with \( n \) cars in front of it to get through the intersection. [16] With this in mind we settled for an approximated value of 35 cars per minute that could pass each lane.

The simulation is built so that the fuzzy logic-implementation and the pre-timed-implementation both use the same base structure and therefore have the same green times from the beginning (before fuzzy logic has been applied). The time adjustment of the current phase for the fuzzy algorithm is calculated when it is about to

\[
\begin{align*}
\text{if } A = \text{medium} \text{ and } Q = \text{many} \text{ then } TA &= \text{veryshortened} \\
\text{if } A = \text{many} \text{ and } Q = \text{zero} \text{ then } TA &= \text{shortened} \\
\text{if } A = \text{many} \text{ and } Q = \text{small} \text{ then } TA &= \text{shortened} \\
\text{if } A = \text{many} \text{ and } Q = \text{medium} \text{ then } TA &= \text{shortened} \\
\text{if } A = \text{many} \text{ and } Q = \text{many} \text{ then } TA &= \text{veryshortened}
\end{align*}
\]

* stands for any and always has a membership degree of 1.

Control rules for medium volume group and large volume group appear in Appendix A.

3.4.3 Defuzzification

The method used for defuzzification is the max-min method. The reason for choosing the max-min method is because of it being less complex compared to the other ones. Due to no fuzzy logic software being available the method had to be implemented from scratch and therefore the least complex was chosen.
end. There is also a maximum of one adjustment per green phase so that one phase does not get an overly long green time.

The two performance criteria used were queue length and delay time for a vehicle. These were calculated by collecting data during the simulation. The average queue length were calculated by collecting data during every second of the simulation and at the end summing them all up and calculate the average of all those numbers. In the case of the average delay every car entering the queue was given a time stamp which was used when the car exited the queue to calculate the waiting time of the car and at the end of the simulation all of these waiting times were used to calculate the average. The result from the simulation is presented using tables and graphs in the result chapter.
Chapter 4

Results

This chapter presents the result of the simulations as described in the method. The simulation was conducted for 180 minutes, both for the pre-timed and the fuzzy logic algorithm. Queue length will be presented first followed by delay.

4.1 Queue length

The result from the simulation is shown in figure 4.1. The result shows the number of vehicles queuing each second both for the pre-timed system and for the fuzzy logic system.

![Figure 4.1](image-url)  

**Figure 4.1.** Number of vehicles queuing each second. The result of the fuzzy logic system is shown in blue and the result of the pre-timed system is shown in red.

In figure 4.2 curves are fitted after the output. This is done to make the result
clearer. The fuzzy logic system clearly has a lower amount of queuing vehicles than the pre-timed system and also has a more steady queue length through the whole simulation.

![Figure 4.2](image)

Figure 4.2. Fitted curves of the number of vehicles queuing each second. Blue curve shows the result of the fuzzy logic system and red curve shows the result of the pre-timed system.

The average queue length is displayed in table 4.1.

<table>
<thead>
<tr>
<th>System</th>
<th>Average queue length (number of vehicles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-timed</td>
<td>27.62</td>
</tr>
<tr>
<td>Fuzzy Logic</td>
<td>24.93</td>
</tr>
</tbody>
</table>

Table 4.1. Average queue length for pre-timed system and fuzzy logic system.

4.2 Delay

The result for delay is shown in figure 4.3. The results show similar delays for both the fuzzy logic system and the pre-timed system through the whole simulation. However, the fuzzy logic system has lower peaks than the pre-timed system.

The average delay is displayed in table 4.2.
4.2. DELAY

Figure 4.3. Delay for each vehicle in the simulation. Blue curve shows the result of the fuzzy logic system and red curve shows the result of the pre-timed system.

<table>
<thead>
<tr>
<th>System</th>
<th>Average delay (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-timed</td>
<td>33.23</td>
</tr>
<tr>
<td>Fuzzy Logic</td>
<td>29.72</td>
</tr>
</tbody>
</table>

Table 4.2. Average delay for pre-timed system and fuzzy logic system.

In figure 4.4 curves are fitted after the output.
Figure 4.4. Fitted curves of the delay for each vehicle in the simulation. Blue curve shows the result of the fuzzy logic system and red curve shows the result of the pre-timed system.
Chapter 5

Discussion

5.1 Discussion of results

The data we collected have high reliability due to the collecting being done by a contractor of Stockholm stad. Unfortunately we were only able to get data for the morning during two and a half hours and for the afternoon during three hours. Because of this we could not get a continuous longer period to measure from, which would have told us whether the algorithm works good over all (or during none-rush hours). Probably the most relevant period to investigate should be the rush hours since it is then traffic most often gets crowded. Therefore we do not miss the most important part, but we cannot get the whole picture. Another relevant fact is also that the data is only collected during one day in the autumn. For a more accurate result the collection of data should have been from more than one day for each season and from all the seasons. But again this doesn’t mean that the data is bad, it means that the reliability of the flow would be higher if there were more to back it up.

Data from adjacent intersections is usually positive for the control system since it gives input on how many cars that arrives. But since our fuzzy logic algorithm only bases its decisions on how many vehicles that are currently in the queues the adjacent intersection becomes less relevant and therefore we do not need the adjacent intersections data.

Regardless of the shortage of data the results will still give a theoretical picture of how a fuzzy logic system performs compared to a pre-timed system since both models use the same base structure and data.

The result is divided into two parts, queue length and delay. The results for the queue length shows that the fuzzy logic algorithm is slightly better than the pre-timed algorithm. This seems reasonable due to the fact that fuzzy logic may both extend and deduct the green time for a phase to achieve a more optimal flow. This can mainly be seen in the figure 4.2 since the fuzzy-curve is almost strictly under
the other, the only exception is when the pre-timed curve dips right before the end. This can also be seen in figure 4.1. The peaks for the pre-timed algorithm are generally higher than the fuzzy algorithms.

The results in the case of the delay is a bit less obvious. In figure 4.4 the fuzzy logic algorithm has three peaks that are higher than the pre-timed which has two peaks. On the other hand the pre-timed algorithm has a peak that is wider than the two biggest from the fuzzy logic algorithm combined. You can also get a feeling of fuzzy being better in figure 4.3 where fuzzy generally has fewer high peaks and the peaks are generally lower than those for the pre-timed algorithm. It is not quite as obvious as in the queue length-case but we feel that we still can make the statement that fuzzy is better in the delay aspect.

5.2 Discussion of the interpretation of the intersection

Because of the many factors involved in an intersection we unfortunately had to limit our working area so that it did not become to big of a project. In our restrictions we chose to not involve buses, cyclists or pedestrians because those were factors that would have made the project more complex. If we had more time and this project as a base it would be very possible to achieve a simulation which takes pedestrians, cyclist and buses into account, so that is a recommendation we have for future projects of this kind. We also disregarded all traffic going from the south to the west since that is not an allowed direction for the cars to go. This will not affect the simulation as a whole because there were only one car in our data that took that turn.

We also had to simplify some other details of the intersection to make it manageable. We have not taken into account any of the u-turns that were made, however they are only about 30 so they would not have made that high of an impact. We also did not make a distinction between heavy- and lightweight vehicles, which if we did would probably have added some seconds to all the heavy vehicles due to that they have a slower start. We argue that this difference would not affect more than one less car per phase were it was present and that it therefore would not affect the result that much.

We chose to focus on making the afternoon period better with fuzzy and has therefore disregarded the morning period. This is mainly due to that they had too big difference in green times that we had to have a different number of phases to portray those conditions properly. This means that we would have had two different fuzzy systems in order to line up with the changed conditions and we thought that would be an unnecessary time consumer since we already have a system for the afternoon period working for us. You could also argue that since we have proved that it works during the afternoon it should also work during the morning since there are gener-
ally more cars per hour during the afternoon than during the morning (according to the data we were given).

5.3 Discussion of the fuzzy logic

When it comes to defuzzification there were different methods to choose between. A lot of them are very complex, not to mention difficult to use if you don’t have access to existing fuzzy logic software. The ultimate solution would probably have been to test more than one method to see which one leads to the best result, but due to time constraints of the project and the difficulty in finding free software we chose to implement the max-min method only.

5.4 Discussion of the simulation

From the start of the project our hope was to use an existing software to implement our comparison but since most of the software requires a licence, which we did not have, we could not use any of those.

Instead we wrote our own implementation of the simulation which gave us the advantage to customize it for our own purposes and also guaranteed us that both models would be based on the same structure. Since it is a simulation there will naturally be a slight difference between reality and what we get from the simulation. In most cases we have based all the numbers in the simulation from scientific reports but unfortunately we could not find the time it takes for a car to drive through an intersection in any report. Instead we based it on how the real schedule behaves, which is that it is at lest two seconds between green lights for lanes that may collide with each other. So we have made sure that it must be at least that much time between collidable lanes, meaning that the fuzzy may not shorten phases under two seconds.

Since the data we got was in chunks of 15 minutes each we had to split it up to distribute the cars evenly over the interval. Mainly cars does not come equally spread out over an interval, but in smaller segments due to being assembled by earlier traffic lights. However, in our simulation it should not matter because it takes place during rush hours which usually have queues. This means that it does not really matter if e.g. three cars come in a cluster or if they come right after each other.
Chapter 6

Conclusion

To sum up, in this study a fuzzy logic system has been implemented and tested on data from an intersection in Stockholm. Previous work has been tested on different places in the world and has shown that fuzzy logic is a powerful tool to make traffic control more efficient. It has been shown in this report that fuzzy logic should make intersections in Stockholm more effective as well. From the results we can conclude that our fuzzy logic system performs a little better than the pre-timed system and thereby make the intersection behave more efficiently. However, since the fuzzy logic system has not been tested in the actual intersection but only in the simulated one, further tests should be carried out before it can be stated completely how effective the fuzzy logic system is. Wider studies which take adjacent intersections, public transport and pedestrians into consideration should all be evaluated in future work as well as testing in the actual intersection. Nevertheless, the simulation still shows promising results which was our goal to begin with.
Appendix A

Fuzzy logic control rules for medium volume group \((V = \text{medium})\):

if \(A = \text{zero} \) and \(Q = * \) then \(TA = \text{veryshortened} \)
if \(A = \text{small} \) and \(Q = \text{zero} \) then \(TA = \text{shortened} \)
if \(A = \text{small} \) and \(Q = \text{small} \) then \(TA = \text{veryshortened} \)
if \(A = \text{small} \) and \(Q = \text{medium} \) then \(TA = \text{veryshortened} \)
if \(A = \text{small} \) and \(Q = \text{many} \) then \(TA = \text{veryshortened} \)
if \(A = \text{medium} \) and \(Q = \text{zero} \) then \(TA = \text{zero} \)
if \(A = \text{medium} \) and \(Q = \text{small} \) then \(TA = \text{veryshortened} \)
if \(A = \text{medium} \) and \(Q = \text{medium} \) then \(TA = \text{veryshortened} \)
if \(A = \text{medium} \) and \(Q = \text{many} \) then \(TA = \text{veryshortened} \)
if \(A = \text{many} \) and \(Q = \text{zero} \) then \(TA = \text{zero} \)
if \(A = \text{many} \) and \(Q = \text{small} \) then \(TA = \text{zero} \)
if \(A = \text{many} \) and \(Q = \text{medium} \) then \(TA = \text{shortened} \)
if \(A = \text{many} \) and \(Q = \text{many} \) then \(TA = \text{veryshortened} \)

Fuzzy logic control rules for large volume group \((V = \text{large})\):

if \(A = \text{zero} \) and \(Q = * \) then \(TA = \text{veryshortened} \)
if \(A = \text{small} \) and \(Q = * \) then \(TA = \text{veryshortened} \)
if \(A = \text{medium} \) and \(Q = \text{zero} \) then \(TA = \text{zero} \)
if \(A = \text{medium} \) and \(Q = \text{small} \) then \(TA = \text{shortened} \)
if \(A = \text{medium} \) and \(Q = \text{medium} \) then \(TA = \text{veryshortened} \)
if \(A = \text{medium} \) and \(Q = \text{many} \) then \(TA = \text{veryshortened} \)
if \(A = \text{many} \) and \(Q = \text{zero} \) then \(TA = \text{extended} \)
if \(A = \text{many} \) and \(Q = \text{small} \) then \(TA = \text{extended} \)
if \(A = \text{many} \) and \(Q = \text{medium} \) then \(TA = \text{zero} \)
if \(A = \text{many} \) and \(Q = \text{many} \) then \(TA = \text{shortened} \)
The following tables show the data collected from Stockholm Stad. They present the traffic flow given in vehicles per quarter for the intersection between Långholmsgatan and Hornsgatan between 15:00 to 18:00.

### Tisdag 11/10/2015

<table>
<thead>
<tr>
<th>Starttid</th>
<th>Sluttid</th>
<th>Höger</th>
<th>Rakt</th>
<th>Vänster</th>
<th>U-Sväng</th>
<th>Höger</th>
<th>Rakt</th>
<th>Vänster</th>
<th>U-Sväng</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:00</td>
<td>15:15</td>
<td>1</td>
<td>158</td>
<td>121</td>
<td>2</td>
<td>117</td>
<td>8</td>
<td>127</td>
<td>0</td>
</tr>
<tr>
<td>15:15</td>
<td>15:30</td>
<td>3</td>
<td>186</td>
<td>139</td>
<td>5</td>
<td>122</td>
<td>11</td>
<td>140</td>
<td>0</td>
</tr>
<tr>
<td>15:30</td>
<td>15:45</td>
<td>3</td>
<td>178</td>
<td>120</td>
<td>1</td>
<td>139</td>
<td>16</td>
<td>139</td>
<td>0</td>
</tr>
<tr>
<td>15:45</td>
<td>16:00</td>
<td>2</td>
<td>192</td>
<td>118</td>
<td>0</td>
<td>123</td>
<td>13</td>
<td>158</td>
<td>0</td>
</tr>
<tr>
<td>16:00</td>
<td>16:15</td>
<td>1</td>
<td>187</td>
<td>132</td>
<td>1</td>
<td>132</td>
<td>18</td>
<td>162</td>
<td>0</td>
</tr>
<tr>
<td>16:15</td>
<td>16:30</td>
<td>0</td>
<td>186</td>
<td>146</td>
<td>1</td>
<td>117</td>
<td>17</td>
<td>137</td>
<td>0</td>
</tr>
<tr>
<td>16:30</td>
<td>16:45</td>
<td>2</td>
<td>182</td>
<td>149</td>
<td>2</td>
<td>113</td>
<td>22</td>
<td>122</td>
<td>0</td>
</tr>
<tr>
<td>16:45</td>
<td>17:00</td>
<td>2</td>
<td>117</td>
<td>125</td>
<td>10</td>
<td>113</td>
<td>14</td>
<td>111</td>
<td>0</td>
</tr>
<tr>
<td>17:00</td>
<td>17:15</td>
<td>0</td>
<td>86</td>
<td>111</td>
<td>8</td>
<td>154</td>
<td>9</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>17:15</td>
<td>17:30</td>
<td>3</td>
<td>141</td>
<td>145</td>
<td>0</td>
<td>60</td>
<td>3</td>
<td>101</td>
<td>0</td>
</tr>
<tr>
<td>17:30</td>
<td>17:45</td>
<td>2</td>
<td>131</td>
<td>122</td>
<td>0</td>
<td>97</td>
<td>22</td>
<td>103</td>
<td>0</td>
</tr>
<tr>
<td>17:45</td>
<td>18:00</td>
<td>4</td>
<td>115</td>
<td>139</td>
<td>1</td>
<td>102</td>
<td>7</td>
<td>109</td>
<td>0</td>
</tr>
</tbody>
</table>

### Södra anslutningen

<table>
<thead>
<tr>
<th>Starttid</th>
<th>Sluttid</th>
<th>Höger</th>
<th>Rakt</th>
<th>Vänster</th>
<th>U-Sväng</th>
<th>Höger</th>
<th>Rakt</th>
<th>Vänster</th>
<th>U-Sväng</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:00</td>
<td>15:15</td>
<td>97</td>
<td>131</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15:15</td>
<td>15:30</td>
<td>124</td>
<td>148</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15:30</td>
<td>15:45</td>
<td>113</td>
<td>130</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15:45</td>
<td>16:00</td>
<td>102</td>
<td>166</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16:00</td>
<td>16:15</td>
<td>103</td>
<td>179</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>16:15</td>
<td>16:30</td>
<td>113</td>
<td>198</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>16:30</td>
<td>16:45</td>
<td>123</td>
<td>207</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16:45</td>
<td>17:00</td>
<td>125</td>
<td>189</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>17:00</td>
<td>17:15</td>
<td>100</td>
<td>190</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>17:15</td>
<td>17:30</td>
<td>107</td>
<td>112</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>17:30</td>
<td>17:45</td>
<td>130</td>
<td>145</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17:45</td>
<td>18:00</td>
<td>123</td>
<td>211</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
Appendix C

On the following page the real schedule used today in the intersection is shown.

F1: West -> North, East, South (one bus-lane)
F2: North -> West, South (one bus-lane)
F3: North -> West, South (two lanes, one for only south and one split between south and west)
F4: North -> East (two lanes)
F5: East -> West (one lane)
F6: East -> North (one lane)
F7: East -> South (two lanes)
F8: South -> North, East, West (one bus-lane)
F9: South -> East (one lane)
F10: South -> North (two lanes)

Cx: Cyclist-signals
Gx: Pedestrian-signals
Bibliography


