AUTOMATED INTERSECTION VOLUME COUNTS

USING EXISTING SIGNAL CONTROL DEVICES

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Civil and Environmental Engineering

by

Ali Gholami

Dr. Zong Tian/Dissertation Advisor

August, 2015
We recommend that the dissertation prepared under our supervision by

ALI GHOLAMI

Entitled

Automated Intersection Volume Counts Using Existing Signal Control Devices

be accepted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Zong Tian, Ph.D., Advisor

Hao Xu, Ph.D., Committee Member

Keith E. Dennett, Ph.D., Committee Member

Pitu Mirchandani, Ph.D., Committee Member

Yantao Shen, Ph.D., Graduate School Representative

David Zeh, Ph.D., Dean, Graduate School

August, 2015
ABSTRACT

The purpose of this dissertation was to identify and investigate the possibility of obtaining turning volumes from inductive loops and investigate the accuracy of them. A large majority of signalized intersections operate under inductive loops. Experiences in cities such as Seattle, San Antonio, and Toronto show successful usage of inductive loop detectors to obtain traffic volume at intersections.

Loop detectors are the most common method for obtaining data at intersections to operate and control traffic signals. In spite of many advantages, they have some drawbacks, including the fact that multiple detectors are usually required to monitor a location.

A macroscopic study was performed on two intersections in Reno and Sparks. Both Reno and Sparks use sequential short loops. The detector accuracy was interpreted in terms of count errors. The preferred metric for count error is the Mean Absolute Percent Error (MAPE, %). Results showed the counts were not reliable and had a very high error. At the Kietzke/Moana intersection in Reno, NV, the MAPE was 15 percent northbound, 31 percent southbound, 20 percent eastbound, and 36 percent westbound. At Sparks/Prater in Sparks, NV, the MAPE was worse with all detector groups ranging from 48 to 74 percent.

In Reno, advance detector counts could be modified because they showed a strong relationship with base (observed) counts; however, in Sparks, there was not a clear relationship between the two sets of counts. In Chapter 4, by using Genetic Programming (GP) and Adaptive Neuro-Fuzzy Inference System (ANFIS), detector counts were
modified and again MAPE was calculated. At Kietzke/Moana, all approaches after data modification had MAPE less than 14 percent. However, at Sparks/Prater, because of the loops’ wiring, there was more irregularity in count detections and as a result, models were not able to reduce detector count errors significantly.

Even when detector counts can be modified, detectors are unable to produce turning movement counts in shared lanes. Current practice involves gathering such information through manual counts, which is very costly. Chapter 5 proposes three methods to estimate turning movement proportions in shared lanes. These methods were tested using linear regression and Genetic Programming (GP). It was found that the hourly average error range at intersections was between 4 to 27 percent using linear regression and 1 to 15 percent using GP.

The proposed method for modifying detector counts did not guarantee reliable counts in all situations. In Chapter 6, a method is proposed to obtain turning movement counts only from signal information without using detector counts. To produce the required data, a simulation was performed in VISSIM with different input volumes. To change turning volumes, a code was developed in COM interface. With this code, the inputs did not have to be changed manually. In addition, the COM code stored the outputs. Data were then exported to a single Excel file. Afterwards, regression and the Adaptive Neural Fuzzy Inference System (ANFIS) were used to build models to obtain turning volumes. The accuracy of the models was defined in terms of MAPE. Results of the two case studies showed that during peak hours, there was a high correlation between actuated green time and volumes. This method does not require extensive data collection and is relatively easy
to employ. The results also showed that ANFIS produced more accurate results compared to regression.

Chapter 7 proposes mid-intersection detector (MID) concept configuration to obtain more accurate counts. MIDs are departure doctors which have moved back to middle of intersection. Under this configuration, in addition to stop bar detectors, some mid-intersection detectors also are used to obtain more reliable counts. Due to intersection operation, stop bar detectors were still required, but compared to traditional departure detector configurations, MIDs were expected to produce more reliable and accurate data while requiring same number of detectors.

Chapter 8 offers some recommendations to change the loop detector systems for the sake of improving turning movement counts. For obtaining more accurate counts, we recommend: 1) the cost-effective and non-intrusive replacements of inductive loops (Passive Infrared, Active Infrared, Radar and Passive Millimeter, Passive Acoustic, Ultrasonic-Pulse and Doppler). Several “non-intrusive” detection systems are becoming more prominent, being viewed as cost-effective replacements of inductive loops; 2) Changing the configuration and wiring of loops. Performance was significantly enhanced when the loops were connected such that the field generated by the individual loops was additive between the loops rather than subtractive. Counting results were likely to be fair to poor when the loops were separated by 10 or more feet or had a different number of turns or were connected in parallel. To obtain excellent to good counts from loops, each loop should be wired to an individual loop detector channel. If two or more are spliced together into one loop detector channel, the count accuracy would be fair to poor.
ACKNOWLEDGMENTS

The completion of this dissertation was conducted in cooperation with the Nevada Department of Transportation (NDOT), Washoe County Regional Transportation Commission (RTC), City of Reno and City of Sparks. I wish to gratefully acknowledge the contributions of several persons who made the progress of this research possible. This especially includes Dr. Rasool Andalibian, Mr. Arafat Khan, Mr. Jim Herman, Mr. Kurt M. Dietrich, and Mr. Dave Walsh. Also greatly appreciated are the comments of Dr. Reed Gibby, Dr. Hao Xu and friends at the Center for Advanced Transportation Education and Research (CATER). Special thanks to Dr. Pitu Mirchandani from Arizona State University and Mr. Srinivasa Sunkari from Texas Institute of Transportation (TTI) and Mrs. Erika P. Hutton for their helpful suggestions. Last but not least, I appreciate support and suggestions of my supervisor Dr. Zong Tian, especially Chapter 6 of this dissertation was his idea.
GLOSSARY OF ACRONYMS

ANFIS: Adaptive Neural Fuzzy Inference System
ANN: Artificial Neural Network
CATER: Center for Advanced Transportation Education and Research
EB: Eastbound
FC: Flow Characteristics (of shared lanes)
FIS: Fuzzy Inference System
GP: Genetic Programming
GUI: Graphical User Interface
ILD: Inductive Loop Detector
ITE: Institute of Transportation Engineers
LLC: Loop Detector lead-in cable
LOS: Level of Service
MAPE: Mean Absolute Percent Error
MID: Mid-Intersection Detector
NB: Northbound
NDOT: Nevada Department of Transportation
NE: Network Equilibrium
NEMA: National Electrical Manufacturers Associations
NTOC: National Transportation Operations Coalition
NTOR: No Turn on Red
RSME: Root Mean Squared Error
RTC: Regional Transportation Commission
SB: Southbound
SCOOT: Split, Cycle, Offset, Optimization Technique
SLR: Simple Linear Regression
TMC: Traffic Management Center
TTI: Texas Institute of Transportation
VQ: Volume and Queue (of shared lanes)
WB: Westbound
TABLE OF CONTENTS

ABSTRACT .......................................................................................................................... i
ACKNOWLEDGMENTS ....................................................................................................... iv
GLOSSARY OF ACRONYMS ............................................................................................. v
TABLE OF FIGURES ........................................................................................................ ix
TABLE OF TABLES ........................................................................................................... xii
1. INTRODUCTION ............................................................................................................ 1
2. LITERATURE REVIEW .................................................................................................. 6
  2.1. SUCCESSFUL USAGE OF LOOP DETECTOR COUNTS BY CITIES AND STATES .......... 7
  2.2. COMPARISON OF LOOP DETECTOR COUNTS WITH ALTERNATIVE TECHNOLOGIES .... 10
  2.3. ACCURACY OF LOOP DETECTOR COUNTS AND IMPROVEMENT ALGORITHMS ......... 10
3. LOOP DETECTOR ACCURACY FOR OBTAINING COUNTS ............................................. 13
  3.1. CASE STUDY INTERSECTIONS ................................................................................. 14
    3.1.1. Data Collection .................................................................................................. 14
    3.1.2. Data Analysis .................................................................................................. 16
  3.2. EQUATIONS FOR CALCULATING THE DETECTOR ACCURACY ................................ 24
  3.3. ANALYSIS OF DETECTOR COUNTS ......................................................................... 25
4. INCREASING THE ACCURACY OF LOOP DETECTOR COUNTS ......................................... 34
  4.1. METHODOLOGY AND SOLUTION APPROACHES .................................................. 34
    4.1.1. Genetic Programming ...................................................................................... 36
    4.1.2. ANFIS ............................................................................................................. 40
  4.2. CASE STUDY ............................................................................................................. 44
    4.2.1. Data Collection ............................................................................................... 44
    4.2.2. Data Analysis .................................................................................................. 45
    4.2.3. Modeling ........................................................................................................ 47
    4.2.4. Results ............................................................................................................ 52
  4.3. SUMMARY AND CONCLUSION .............................................................................. 54
5. DETERMINING TURNING MOVEMENT PROPORTIONS IN SHARED LANES..........................55

5.1. METHODOLOGY ........................................................................................................56
  5.1.1. Network Equilibrium (NE) ....................................................................................56
  5.1.2. Volume and Queue (VQ) of Length of Shared Lanes ........................................58
  5.1.3. Flow Characteristics (FC) of Shared Lanes .......................................................61
  5.1.4. Observed Proportions in the Field .......................................................................64

5.2. CASE STUDY .............................................................................................................64
  5.2.1. Network Equilibrium (NE) ....................................................................................64
  5.2.2. Volume and Queue Length (VQ) of Shared Lanes .............................................66
  5.2.3. Flow Characteristics (FC) of Shared Lanes .......................................................66

5.3. DATA ANALYSIS .....................................................................................................68
  5.3.1. Modeling .............................................................................................................68
  5.3.2. Analysis of Results ............................................................................................70

5.4. SUMMARY AND CONCLUSIONS ..........................................................................71

6. ESTIMATING INTERSECTION TURNING VOLUMES FROM TRAFFIC SIGNAL INFORMATION ... 73

6.1. PROBLEM STATEMENT ..........................................................................................74

6.2. METHODOLOGY ....................................................................................................76
  6.2.1. ANFIS ................................................................................................................80
  6.2.2. Count Estimation Procedure ..........................................................................84

6.3. CASE STUDIES .......................................................................................................84

6.4. RESULTS .................................................................................................................88

6.5. SUMMARY AND CONCLUSION ............................................................................92

7. MID-INTERSECTION DETECTOR (MID).................................................................94

7. 1. PROPOSED MID-INTERSECTION DETECTORS .................................................94

7.2. SIMULATION AND CASE STUDY .........................................................................103

7. 3. RESULTS ...............................................................................................................106

7. 4. SUMMARY AND CONCLUSION ...........................................................................108

8. CONCLUSIONS AND RECOMMENDATIONS ..........................................................111

8.1. COST-EFFECTIVE REPLACEMENTS OF INDUCTIVE LOOPS .........................111

8.2. CHANGING CONFIGURATION AND WIRING OF LOOPS .................................116
8.2.1. Single Loops ................................................................. 116
8.2.2. Two Loops Connected in Series ........................................... 118
8.2.3. Four Loops Connected in Series or Series/Parallel ....................... 120
8.3. SUMMARY AND CONCLUSIONS ........................................... 124
REFERENCES ............................................................................. 126
APPENDIX A: SURVEY QUESTIONNAIRE .................................. 132
APPENDIX B: RENO LOOP SPECIFICATION .................................. 135
APPENDIX C: SPARKS LOOP SPECIFICATION .................................. 138
Appendix D: R Codes .................................................................. 141
Appendix E: MATLAB Codes .......................................................... 150
## TABLE OF FIGURES

Figure 1: Case study intersections, Kietzke/Moana in Reno and Sparks/Prater in Sparks ..... 16
Figure 2: Miovision data collection software .................................................................... 17
Figure 3: Kietzke/Moana Northbound ............................................................................. 18
Figure 4: Kietzke/Moana Southbound ............................................................................ 18
Figure 5: Kietzke/Moana Eastbound .............................................................................. 19
Figure 6: Kietzke/Moana Westbound ............................................................................. 19
Figure 7: Sparks/Prater Northbound Through ................................................................. 20
Figure 8: Sparks/Prater Northbound Left ....................................................................... 20
Figure 9: Sparks/Prater Southbound Through ................................................................. 21
Figure 10: Sparks/Prater Southbound Left ..................................................................... 21
Figure 11: Sparks/Prater Eastbound Through ................................................................. 22
Figure 12: Sparks/Prater Eastbound Left ..................................................................... 22
Figure 13: Sparks/Prater Westbound Through ............................................................... 23
Figure 14: Sparks/Prater Westbound Left ..................................................................... 23
Figure 15: Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) for Kietzke/Moana ................................................................. 25
Figure 16: Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) for Sparks/Prater ................................................................................. 25
Figure 17: The scatterplot between detector and base counts Kietzke/Moana .................. 26
Figure 18: The scatterplot between detector and base counts Sparks/Prater ..................... 27
Figure 19: Fitted Line Plot and regression model in Kietzke/Moana northbound ............. 28
Figure 20: Fitted Line Plot and regression model in Kietzke/Moana southbound ............ 28
Figure 21: Fitted Line Plot and regression model in Kietzke/Moana eastbound................. 29
Figure 22: Fitted Line Plot and regression model in Kietzke/Moana westbound ............... 29
Figure 23: Fitted Line Plot and regression model in Sparks/Prater northbound through ... 30
Figure 24: Fitted Line Plot and regression model in Sparks/Prater northbound left ......... 30
Figure 25: Fitted Line Plot and regression model in Sparks/Prater southbound through ... 31
Figure 26: Fitted Line Plot and regression model in Sparks/Prater southbound left ......... 31
Figure 27: Fitted Line Plot and regression model in Sparks/Prater eastbound through ...... 32
Figure 28: Fitted Line Plot and regression model in Sparks/Prater eastbound left .......... 32
Figure 29: Fitted Line Plot and regression model in Sparks/Prater westbound through ...... 33
Figure 30: Fitted Line Plot and regression model in Sparks/Prater westbound left .......... 33
Figure 31: Methodology of modifying the detector counts ............................................ 35
Figure 32: Crossover in genetic programming ................................................................. 39
Figure 33: Adaptive Neuro-Fuzzy Inference System (ANFIS) ....................................... 44
Figure 34: sample of 15-min aggregated detector data .................................................... 45
Figure 35: Detector and observed volumes ..................................................................... 47
Figure 36: Scatterplots for detector volume and observed counts (left) and detector occupancy and observed counts (right) at Kietzke/Moana intersection, Reno, NV.........................................................51
Figure 37: Detector and observed counts versus models counts, northbound ..................................51
Figure 38: 3-D surface plot of variables Time, detector Volume and ANFIS output ...............................52
Figure 39: MAPE and RMSE of detectors and models ........................................................................53
Figure 40: Example Network with Two Intersections ........................................................................57
Figure 41: Sample Aggregated and High Resolution Data ....................................................................58
Figure 42: Intersection at 8th Street and Center Street, Reno, NV ......................................................60
Figure 43: Headway in Shared Lanes ......................................................................................................62
Figure 44: Intersection of 9th Street and Sierra Street, Reno, NV .......................................................65
Figure 45: Intersection of North McCarran Blvd and Clear Acre Ln, Reno, NV ....................................68
Figure 46: Mean Absolute Percentage Error (MAPE) of Different Methods at Case Study Intersections .................................................................................................................................72
Figure 47: MAPE and RMSE of Kietzke/Moana, Reno and Sparks/Prater, Sparks, NV ........................75
Figure 48: Signal log from intersection of Virginia St. and McCarran Blvd, Reno, NV .............................76
Figure 49: Methodology of estimating intersection turning volumes from traffic signal information ..................................................................................................................................79
Figure 50: Sample of VISSIM output for phase 2 and 8 ......................................................................79
Figure 51: Adaptive Neuro-Fuzzy Inference System (ANFIS) .................................................................84
Figure 52: Intersection of E 2nd St (east-west) and Kirman Ave (north-south), Reno, NV ....................86
Figure 53: Intersection of McCarran Blvd (east-west) and N Virginia St (north-south) .......................86
Figure 54: Scatterplots of Green Time per Cycle and Volume per Cycle in E 2nd St. during different times ........................................................................................................................................87
Figure 55: Results of intersection at E 2nd St and Kirman Ave, Reno, NV ............................................92
Figure 56: Results of intersection at McCarran Blvd and N Virginia St, Reno, NV .............................92
Figure 57: Mid-Intersection Detectors concept, Case 1. ........................................................................95
Figure 58: Mid-Intersection Detectors concept, Case 2. .......................................................................98
Figure 59: Mid-Intersection Detectors concept, Case 3. .......................................................................98
Figure 60: Mid-Intersection Detectors concept, Case 4. .....................................................................99
Figure 61: Mid-Intersection Detectors concept, Case 5. ......................................................................99
Figure 62: Obtaining counts from mid-intersection detectors ..................................................................101
Figure 63: Failed activation algorithm. ................................................................................................102
Figure 64: High-resolution data. ...........................................................................................................103
Figure 65: Case study at intersection, Oddie Blvd and Sullivan Ln in Reno, NV, Case 6 .....................106
Figure 66: Average MAPE of detectors (stop bar) in the field, Reno, NV ............................................106
Figure 67: Count errors at MID and departure detectors configurations ..............................................109
Figure 68: Approach Volumes, a sample in South Lake City, Utah [48] .............................................114
Figure 69: Various applications of single loops [15] ..........................................................................118
Figure 70: Various applications of two loops [15] .............................................................................120
Figure 71: Various applications of four loops [15]..............................122
Figure 72: Sequential four loop configuration and expected accuracy [15]..............................123
Figure 73: The process of using the methods provided in this dissertation..............................125
TABLE OF TABLES
Table 1: Attributes of loop detector systems in Reno and Sparks.............................................. 15
Table 2: Proposed detector counts modifying models................................................................. 27
Table 3: Regression and GP models for to modify detector counts ........................................... 49
Table 4: ANFIS membership function models for detector counts .......................................... 50
Table 5: Sample of Data Collected for VQ Method at Intersection of 8th Street and Center Street, Reno, NV ................................................................. 66
Table 6: Sample of Data Collected for FC Method at intersection of 8th Street and Center Street, Reno, NV ................................................................. 67
Table 7: Case Study Lanes Models ............................................................................................. 69
Table 8: Mean Absolute Percentage Error (MAPE) of Different Methods at Case Study Intersections ................................................................................................. 72
Table 9: Signal configuration of intersection at Virginia St. and McCarran Blvd, Reno, NV .......... 77
Table 10: A sample of signal information from VISSIM Simulation ........................................... 80
Table 11: Sample of fuzzy sets of intersection at McCarran Blvd and Virginia St during peak hours ................................................................................................. 89
Table 12: A sample of VISSIM output for phases 4 ................................................................. 105
1. INTRODUCTION

Traffic volume studies are typically conducted to determine the number, movements, and classifications of roadway vehicles at a given location. These data help to identify critical flow time periods, determining the influence of large vehicles or pedestrians on vehicular traffic flow. Manual counts are typically used to gather data for determination of vehicle classification, turning movements, direction of travel, and vehicle occupancy. Most applications of manual counts require small samples of data at any given location.

The automatic count method provides a means for gathering large amounts of traffic data using permanent or portable counters. The National Transportation Operations Coalition (NTOC) published the 2007 National Traffic Signal Report Card, giving the nation an overall grade of an F with automatic data collection. The NTOC’s recommendation was that more resources should be devoted to traffic signal operations. While obtaining more resources would be ideal, due to the reality of constrained resources, the primary objective of this dissertation is to investigate if reliable traffic volume information can be automatically collected using existing signal control devices.

A large majority of signalized intersections operate under some form of actuated control in that intersection approaches (or lanes) have some type of inductive loops. The new loop detector (also called loop amplifier) and signal controller equipment now provide the ability to collect traffic count information from the same loops on the intersection approaches that are used for actuated control. The potential to extract traffic counts from
an existing signalized intersection loop detection system may provide the opportunity to collect data at minimal cost.

There are many benefits of collecting traffic counts from loops at signalized intersections including the low cost of collecting counts. However, there are also several issues, including variations among transportation agencies in terms of signal loop placement, layout and wiring; potential variations in methods of data extraction based upon type of technology and/or detector manufacturer used; and loop maintenance issues.

Throughout the report, the term “loop” is used to describe the in-pavement inductive loop. The term “loop detector” is the equipment in the signal controller cabinet which is connected to the loop and monitors the inductance changes to generate a signal to the controller. It is also referred to as the “loop amplifier” or the “loop relay.” The term “loop detection system” refers to both the loop and the loop detector.

Chapter 2 provides a review of the various research about loop detectors and configurations utilized at signalized intersections and improving the level of their accuracy. Also, some cases are presented where cities or states have successfully used the existing loops at signalized intersections for obtaining traffic counts.

Chapter 3 provides the existing status of loop detector systems in Reno and Sparks. The purpose of this chapter is to identify and investigate the count accuracy of existing inductive loops. In this chapter, counts of loop detectors at two study intersections were compared with manual counts.
Results of Chapter 3 show that detector counts are not reliable. In Chapter 4, a method is proposed to improve the accuracy of these counts. The purpose of this chapter is to increase the accuracy of loop detector counts using the Adaptive Neural Fuzzy Inference System (ANFIS) and Genetic Programming (GP) based on detector volume and occupancy. ANFIS is a class of adaptive networks that is functionally equivalent to a fuzzy inference system; however, in ANFIS, the user does not need to define the rules. Rules are generated using an artificial neural system. This feature makes the ANFIS a very powerful tool to explain the relationship of variables. In this study, the ANFIS function built in MATLAB was used. GP adopted in this study is a form of regression but with the difference that the user does not need to define the structure of the data. GP, with an evolutionary process, finds the near-the-best structure for data and also similar to the regression, finds the model parameters.

These methods do not need microscopic analysis and are easy to employ. Four approaches of one intersection were used in the case study. Results showed that the models can improve intersection detector counts significantly. Results also showed that ANFIS produced more accurate counts compared to regression and GP.

One of the major challenges of obtaining turning volume counts from detectors is when there are shared lanes because in this case, additional detectors are needed to track the turns. Loops need to be located at strategic locations at the stop bar and downstream of the intersection. In cases where there are no downstream detectors, Chapter 5 proposes three methods to estimate turning movements from stop bar detectors only. These methods were tested using simple linear regression and GP. The proposed methods have the potential of
applying to locations where appropriate detectors (especially loop detectors) are installed for obtaining data at intersections.

Though Chapters 4 and 5 propose methods to improve detector counts, these methods were not successful in all situations. Chapter 6 introduces a method which attempts to obtain turning movement counts from only signal information without using detector information. To produce the required data, a simulation was performed in VISSIM. The reason for choosing this software was its ability to produce high-resolution outputs that are required in this method. In this simulation, different turning volumes were entered for each signal configuration parameters. To change turning volumes, a code was developed in COM interface. With this code, the inputs did not have to be changed manually. Then, for each phase, all green times and their corresponding volume were extracted. Afterwards, a prediction model was built for each phase/movement. Two methods were adopted to build the models: regression and the Adaptive Neural Fuzzy Inference System (ANFIS).

Chapter 7 proposes a new layout for downstream detectors that is called mid-intersection detector (MID). In this method, departure loops are moved back to the middle of the intersection in such a way that each MID can capture more than one movement.

The conclusions and recommendations are provided in Chapter 8. The first part of Chapter 8 proposes cost effective replacement for loop detectors and second part of this chapter proposes some recommendations for optimum placement and wiring of the loops to produce accurate counts.
The appendices include the questionnaire survey to solicit information on the current loop installation practices in Reno and Sparks, loop detector system configurations in Reno and Sparks, R codes and MATLAB codes.
2. LITERATURE REVIEW

Technologies for automated traffic volume counts at roadway segments are relatively mature. An increasing number of large urban freeways in the U.S. have implemented flow detectors to automatically gather volume, speed, and occupancy data. Various detection technologies exist, including inductive loops, pneumatic tubes, magnetic sensors, video, radar, and microwave detectors. However, these technologies have not been widely used for obtaining turning movement volumes at intersections. The difficulties in automated turning movement counts lie in the fact that multiple detectors need to be placed in order to track the turning traffic. In current practice, turning movement volumes are mostly obtained through manual counts, which prove to be costly and dangerous to conducting personnel. Therefore, research on automatic turning movement counts at intersections has drawn major interest over the past decade.

Very few efforts are reported in regard to the use of local traffic detectors for systematic volume data collection. Some researchers have investigated freeway loop detector errors [1-12]. However, due to the significance of speed and space headway of vehicles on loops, freeway-detecting loops have different characteristics and accuracy compared to intersection loops. In the following section, the cities and states that have used loop detector systems successfully to collect intersection traffic volumes at intersections are reviewed. The authors also have reviewed studies that have proposed alternative technologies to replace loop detector systems to obtain more accurate counts. Finally, a quick review on studies related to loop detector improvement approaches is provided.
2.1. SUCCESSFUL USAGE OF LOOP DETECTOR COUNTS BY CITIES AND STATES

Some cities, including Seattle, San Antonio, and Toronto provide real-time or stored travel information on selected freeways and arterials based on information received at their traffic management centers from their network of inductive loop detectors.

Metropolitan Toronto reported the development of a prototype transit and traffic information system [13]. The goal was to incorporate freeway and arterial SCOOT data into a complete user information data system. The system is called COMPASS and is employed on some sections of Queen Elizabeth Way (QEW) and Highway 401 [14]. In this system, data is collected at 20-second intervals and aggregated to 5-minute, 15-minute, one hour, daily and monthly time periods. Volume, occupancy and speed data are archived for the 20-second and 5-minute time intervals while only volume data is archived.

The San Antonio TransGuide program has been warehousing traffic information from over 300 detector stations located on freeway mainline segments and ramps. Speed, volume, and occupancy data are all stored in their database [14].

Institute of Transportation Engineers (ITE) reports that four cities, Nashua, NH; Fremont, CA; Minneapolis/St. Paul, MN; and Bellevue, WA are collecting traffic counts using their loop detector systems [15].

Nashua has mostly TS1 cabinets. The initial thought for collecting data at one intersection was to utilize the presence loops at the stop line of all approaches. These loops were known to be working after detailed testing by the city’s maintenance and operations staff.
However, after reviewing the signal layout plans for the intersection and comparing the functionality of available upstream 6 ft. by 6 ft. system loops with the presence loops, the conclusion was made to use the system loops. Data was extracted from the controller using a field laptop every 10 days during the desired data collection period [15].

In Fremont, CA, data is collected from the system loops and stored in and managed by the traffic signal controller. The controller is programmed to configure each system loop and determine how the collected data is grouped. Loops are typically set up to collect traffic volume and occupancy data, which are summarized in 15-minute intervals, very similar to traditional tube counts for collecting average daily traffic. Fremont has standardized its traffic signals with the use of the National Electrical Manufacturers Association’s (NEMA) TS2 traffic signal controllers and controller cabinets [15].

In 1993, the Minnesota Department of Transportation (Mn/DOT) began collecting loop detector counts on the instrumented part of the Twin Cities Metropolitan Freeway System. The system now consists of 648 directional miles and 4,300 inductive loop detectors. Both volume and occupancy are recorded and achieved in 30-second intervals. Loop detector data from traffic signals has always been available using the signal controller proprietary software, but the data was difficult to retrieve and analyze. In 2005, Mn/DOT began retrieving loop detector data from the field and then storing the data in a format that can be easily analyzed. The data was stored on a server in binary format that can be retrieved by anyone at Mn/DOT. Tools were developed to allow the user to retrieve data for numerous
loop detectors over a given period (hours to months). This data can then be averaged, smoothed, and graphed.

Bellevue, WA, also similar to Nashua and Fremont, uses advance loops located about 100 to 140 ft. from the stop line to measure the volume and occupancy data of an approach. If the approach roadway has more than one lane, the combined traffic flow of that approach is measured. At some locations with heavy turning volumes or uneven lane distribution, separate measurements for each movement are made. A remote communication unit in the signal cabinet transmits the raw data back to the central signal computer in the TMC [15].

North Carolina conducted a test on several locations in the state and concluded that there was a high level of similarity between manual counts and the 6 ft. by 6 ft. stretch loop counts. Therefore, they recommended that North Carolina begin using stretch (far) loops for traffic counts by rewiring cabinets and installing detector amplifiers with count outputs on an as-needed basis. They did not recommend the use of quadrupoles. NCDOT does not need to replace every detector amplifier with count-output units; rather, it can simply swap them out as needed for counts. Finally, they observed essentially no variation between rhombus, diamond, and square shaped loops during their 2001 field investigation. However, it was recommend that North Carolina retain the use of rectangular (square) 6 ft. by 6 ft. loop shapes [31].
2.2. COMPARISON OF LOOP DETECTOR COUNTS WITH ALTERNATIVE TECHNOLOGIES

Most vehicle detection today relies on inductive loop detectors. However, problems with installation and maintenance of these detectors have necessitated evaluation of alternative detection systems. Replacing loops with better detectors requires a thorough evaluation of the alternatives. The alternative detection technologies include: video image detection, radar, Doppler microwave, and passive acoustic. Several research have compared these technologies with loop detectors and results clearly indicate promising nonintrusive alternatives to loops, but the limitations of each type must be understood [1,16].

2.3. ACCURACY OF LOOP DETECTOR COUNTS AND IMPROVEMENT ALGORITHMS

Many cities use loop detectors, but in many cases, counts from these detectors are not accurate. Vanajakshi and Rilett [8] and Bender and Nihan [23] reviewed studies regarding the accuracy of loop detector counts and improvement algorithms. Jacobson et al. [17] divided loop detector data screening tests into two main categories: microscopic and macroscopic. At the microscopic-level, detector pulses were scanned and checked for error in the field. At the macroscopic-level, volume from detectors was collected from the sites and was compared with manual counts. Some researchers such as Dudek et al. [18], Courage et al. [19], Pinnell [20], Bikowitz and Ross [21], and Chen and May [22] have addressed loop detector data errors, its causes, and effects. Studies of loop detector data errors at the microscopic level usually require reprogramming or modification of the detector device and depend on the type of loop detector [10, 22, 24]. However, macroscopic approaches are more commonly adopted because they are independent of the sensor type
and are carried out at the data processing level [25]. Common macroscopic studies compare volumes, occupancies, or speeds with specific threshold values [17, 26, 27]. The main disadvantage of single-parameter threshold tests, which typically consider only one parameter at a time, is that they assume the acceptable range for a parameter independent of the values of the other parameters. Because combinations of parameters are not tested, single-parameter threshold tests cannot identify unreasonable combinations. Typically, the combinations of parameter tests take advantage of the relationships among the three parameters: mean speed, volume, and occupancy [17, 26–30].

A number of research efforts attempted to use videos, (either video-detection systems or videotaping) coupled with data extraction software, to estimate turning movement volumes at intersections. Tian et al. [32] developed a system called Time and Place System (TAPS), which extracts turning movement data from a low number of video detections. Miovision Technologies Inc. developed a portable intersection video recording system that extracts the turning volumes from the videos using their in-house software. Such technologies still require installation of the video equipment each time a count is desired. The cost can be high if a large number of sites are covered. Furthermore, video-based technologies heavily rely on the camera view, which could be restricted by the physical layout of an intersection. Other non-video-based tools aim to provide real-time estimates for turning movement proportions [33, 34, 35]. Such efforts focused on improving adaptive signal control systems. Various algorithms were developed based on limited detector information to derive turning movement proportions, from which turning volumes can be derived by the total link volumes.
Obtaining turning volume counts from detectors in shared lanes is a major challenge, thus multiple detectors are often needed to track the turns [36, 37]. Loops must be located at strategic locations at the stop bar and downstream of the intersection. Furthermore, based on the network traffic flow patterns, some intersections’ turning estimates can provide approximations for adjoining intersections [38].

To the best of our knowledge, all these approaches involve comparison of detector volumes with manual counts to determine the accuracy of detectors and the cause of errors. As a result, it is usually proposed to change the loop or wiring configuration which is generally very costly. In this research, existing signal control devices were used to obtain data. ANFIS and GP were tested to improve detector counts. ANFIS was able to learn the behavior of detectors and produce better results from inputs and GP was a powerful tool to find patterns of errors and consider them for future prediction. Also this research proposes a method to obtain turning movements from signal information without using loop detectors. The next chapter explains why loop detector counts are not useful without modifications.
3. LOOP DETECTOR ACCURACY FOR OBTAINING COUNTS

A survey on loop detector facilities was performed in Reno and Sparks. Appendix A shows the survey questionnaire. Both Reno and Sparks, Nevada use sequential short loops. The use of sequential short loops to emulate a long loop is the preferred treatment in many agencies. The advantages of this configuration result primarily from fewer failures because of the loop's shorter length. Thus, they are less vulnerable to problems caused by crossing pavement cracks and joints and to adjacent lane pickup (splashover). Long loops are more subject to adjacent lane splashover since the entire length of the vehicle is exposed to the side of the long loop (approximately 17 ft.) as compared to less than a third of the vehicle length of about 6 ft. for a short loop. The short loops also provide superior detection of small vehicles [7]. In Reno and Sparks, sequential short loops consist of four 6 ft. by 6 ft. square loops separated by 9 ft. or 6 ft. for left turn lanes and two 6 ft. by 6 ft. square loops for through lanes. Table 1 shows different attributes of loop detector systems in Reno and Sparks. This table was developed based on the survey presented in Appendix A.

Loops are wired together in a series circuit based on each phase. This provides for a more robust electrical connection and minimizes both false and missed calls. Under such an installation, when a vehicle passes over any of the detection zones, a call is sent. If two vehicles were to arrive simultaneously (or more precisely, within about 1/10 to 1/7 of a second of each other), the relay-based detector in the cabinet cannot differentiate between the signals. Unfortunately, this also would appear to limit the ability to count vehicles accurately. Detector units are set to presence mode; as a result, the relay will remain
actuated (that is, the “normally open” contact will stay closed) as long as a vehicle is in the detection zone.

In Sparks, there is no separate wiring for the advance detectors. The controller combines the advance detector with the stop bar detector and counts it as a single detector. As a result, it is not currently possible to get counts from these advance detectors.

3.1. CASE STUDY INTERSECTIONS

A macroscopic test was conducted at Kietzke/Moana intersection in Reno, NV and another test was conducted at Sparks/Prater intersection in Sparks, NV on June 12th and 13th, 2013 respectively. The locations of intersections are shown in Figure 1.

3.1.1. Data Collection

During specific data collection periods, data were collected using a Miovision Scout Video Collection Unit. After recording the intersections during the study hours, the videos were uploaded to Miovision data management software (Figure 2). These data was the base (observed) data for estimating the accuracy of loop detector system. Counts were collected in morning from 6:45 a.m. to 9:15 a.m., in mid-day from 11:00 a.m. to 1:45 p.m., and in afternoon from 3:45 p.m. to 6:00 p.m. At the same time counts were collected from detectors in study intersections. Appendix D shows the counts from both surveys.
<table>
<thead>
<tr>
<th>Loop Detector Attributes</th>
<th>Reno</th>
<th>Sparks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of intersections controlled by the responsible agency</td>
<td>238</td>
<td>108</td>
</tr>
<tr>
<td>Number of intersections with detection type</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Loops</td>
<td>132 (Some Intersections have both Loops and Video)</td>
</tr>
<tr>
<td></td>
<td>Video</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>Other</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>32</td>
</tr>
<tr>
<td>Loops on the major street</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Left turn lane</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Through lane</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Right turn lane</td>
<td>No</td>
</tr>
<tr>
<td>Loops on the side street</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Left turn lane</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Through lane</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Right turn lane</td>
<td>No</td>
</tr>
<tr>
<td>Separate loops in each lane of an approach</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Wiring separate for each loop back to the controller</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Spliced together by</td>
<td>Phase</td>
<td>Phase</td>
</tr>
<tr>
<td>Type of controllers existing in the field</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NEMA TS1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>TS2</td>
<td>✓ (TS2 in TS1 cabinet)</td>
</tr>
<tr>
<td></td>
<td>170</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2070</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Other</td>
<td>-</td>
</tr>
<tr>
<td>Number of loops per lane</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Left</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Through</td>
<td>2</td>
</tr>
<tr>
<td>Size of loops and distance between them in each lane</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Size</td>
<td>6'×6'</td>
</tr>
<tr>
<td></td>
<td>Distance between them in each lane</td>
<td>6' to 9'</td>
</tr>
<tr>
<td></td>
<td>Distance back from stop line</td>
<td>-2' (Advance detectors are based on table in Appendix B)</td>
</tr>
<tr>
<td>Are counts collected at traffic signals from detectors on the approaches?</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Does detector data go back to TMC or a central computer?</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Is the data maintained in a historical data base?</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Advance detectors</td>
<td>Yes</td>
<td>Yes (Spliced to stop bar detectors)</td>
</tr>
<tr>
<td>Any plan for replacing the loop detectors with other detectors?</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
3.1.2. Data Analysis

The purpose of this section was to identify and investigate the count accuracy of inductive loops. The volume scatterplots, shown for both the base and detector counts in Figures 3 to 14, provide a visual representation difference between base and detector counts during different time periods. Each point on a scatterplot represents a 15-minute traffic volume as measured by the base or loops on a defined approach.
Figure 2: Miovision data collection software
Figure 3: Kietzke/Moana Northbound

Figure 4: Kietzke/Moana Southbound
Figure 5: Kietzke/Moana Eastbound

Figure 6: Kietzke/Moana Westbound
Figure 7: Sparks/Prater Northbound Through

Figure 8: Sparks/Prater Northbound Left
Figure 9: Sparks/Prater Southbound Through

Figure 10: Sparks/Prater Southbound Left
Figure 11: Sparks/Prater Eastbound Through

Figure 12: Sparks/Prater Eastbound Left
Figure 13: Sparks/Prater Westbound Through

Figure 14: Sparks/Prater Westbound Left
3.2. EQUATIONS FOR CALCULATING THE DETECTOR ACCURACY

Accuracy can be expressed using one of the following two error quantity values [1]:

1. Mean Absolute Percent Error (MAPE) (see Equation 1) or

2. Root Mean Squared Error (RMSE) (see Equation 2).

\[
MAPE(\%) = \frac{\sum_{i=1}^{n} \left| \frac{D_i - B_i}{B_i} \right|}{n}
\]

Equation 1

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (D_i - B_i)^2}{n}}
\]

Equation 2

where:

MAPE: Mean Absolute Percentage Error

RMSE: Root Mean Squared Error

\(D_i\): the detector data value

\(B_i\): the reference (base) data value

\(n\): the total number of time intervals

The detector accuracy is interpreted in terms of count error. The preferred metric for count error was the Mean Absolute Percent Error (MAPE, %) as its range of values is larger. The RMSE for counts was also performed. RMSE is a useful statistic for evaluating the deviation between loop and base volumes. The RMSE were calculated for 15-minute volumes. Note that the RMSE was sensitive to the relative volume of traffic being observed. The RMSE for devices at one test site cannot be directly compared to RMSE values for devices at all other intersection test site. Similarly, the RMSE from one approach at the intersection cannot be compared to the RMSE values from another approach.
These metrics allow an easy comparison of detector and base counts. Figures 15 and 16 summarize count performance attributes of the detectors tested at two intersections during study hours. The figures utilize the MAPE and RMSE to describe these results.

**Figure 15: Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) for Kietzke/Moana**

**Figure 16: Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) for Sparks/Prater**

### 3.3. ANALYSIS OF DETECTOR COUNTS

As seen in Figures 3 through 6 (Kietzke/Moana) detector counts showed a strong association with base counts. In almost all graphs the detectors showed lower volume at
almost the same rate. In Figures 7 through 14 (Sparks/Prater) detectors also undercounted the volumes. Therefore it might be possible to modify detector counts with regression model for each detector group. In this study a simple linear regression (SLR) was performed in Minitab.

To get an idea of possible relationships between detector and base counts, a scatterplot was constructed in Minitab (Figures 17 and 18). Some of the scatterplots in Figure 17 indicated that the pairs (detector and base counts) have a mild linear association except for west bound. However scatterplots in Figure 18 did not show association between pairs. This might be because stop bar detectors are wired together. To determine the relationship between detector and base counts a SLR analysis was performed for detector groups. Figures 19 through 30 show the results. Table 2 also shows the estimated regression models. The next chapter explains a method to increase the accuracy of detector counts.

![Scatterplot of NB-B vs NB-D, SB-B vs SB-D, EB-B vs EB-D, WB-B vs WB-D](image-url)

*Figure 17: The scatterplot between detector and base counts Kietzke/Moana*
Figure 18: The scatterplot between detector and base counts Sparks/Prater

Table 2: Proposed detector counts modifying models

<table>
<thead>
<tr>
<th>Intersection</th>
<th>Phase</th>
<th>Modifying Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kietzke/Moana</td>
<td>Northbound</td>
<td>NB-B = 6.18 + 1.133 NB-D*</td>
</tr>
<tr>
<td></td>
<td>Southbound</td>
<td>SB-B = 27.50 + 1.240 SB-D</td>
</tr>
<tr>
<td></td>
<td>Eastbound</td>
<td>EB-B = 22.51 + 1.106 EB-D</td>
</tr>
<tr>
<td></td>
<td>Westbound</td>
<td>WB-B = 141.7 + 0.6829 WB-D</td>
</tr>
<tr>
<td></td>
<td>Northbound Through</td>
<td>NB-T-B = 419.7 - 6.800 NB-T-D</td>
</tr>
<tr>
<td></td>
<td>Northbound Left</td>
<td>NB-L-B = 13.52 + 1.378 NB-L-D</td>
</tr>
<tr>
<td></td>
<td>Southbound Through</td>
<td>SB-T-B = 362.7 - 3.786 SB-T-D</td>
</tr>
<tr>
<td></td>
<td>Southbound Left</td>
<td>SB-L-B = 5.251 + 1.786 SB-L-D</td>
</tr>
<tr>
<td></td>
<td>Eastbound Through</td>
<td>EB-T-B = 103.4 - 0.2391 EB-T-D</td>
</tr>
<tr>
<td></td>
<td>Eastbound Left</td>
<td>EB-L-B = 22.68 + 2.913 EB-L-D</td>
</tr>
<tr>
<td></td>
<td>Westbound Through</td>
<td>WB-T-B = 55.48 + 0.3648 WB-T-D</td>
</tr>
<tr>
<td></td>
<td>Westbound Left</td>
<td>WB-L-B = 33.60 + 0.7024 WB-L-D</td>
</tr>
</tbody>
</table>

* The last letter of variables refer to D: Detector count, and B: Base count
Figure 19: Fitted Line Plot and regression model in Kietzke/Moana northbound

Figure 20: Fitted Line Plot and regression model in Kietzke/Moana southbound
Figure 21: Fitted Line Plot and regression model in Kietzke/Moana eastbound

Figure 22: Fitted Line Plot and regression model in Kietzke/Moana westbound
Figure 23: Fitted Line Plot and regression model in Sparks/Prater northbound through

Figure 24: Fitted Line Plot and regression model in Sparks/Prater northbound left
Figure 25: Fitted Line Plot and regression model in Sparks/Prater southbound through

Figure 26: Fitted Line Plot and regression model in Sparks/Prater southbound left
Figure 27: Fitted Line Plot and regression model in Sparks/Prater eastbound through

Figure 28: Fitted Line Plot and regression model in Sparks/Prater eastbound left
Figure 29: Fitted Line Plot and regression model in Sparks/Prater westbound through

Figure 30: Fitted Line Plot and regression model in Sparks/Prater westbound left
4. INCREASING THE ACCURACY OF LOOP DETECTOR COUNTS

Loop detectors are the most commonly used devices for obtaining data at intersections. Multiple detectors are usually required to monitor a location, and this reduces the accuracy of detectors for collecting traffic volumes. The purpose of this chapter is to increase the accuracy of loop detector counts using the Adaptive Neural Fuzzy Inference System (ANFIS) and Genetic Programming (GP) based on detector volume and occupancy. These methods do not need microscopic analysis and are easy to employ. Four approaches of one intersection were used in the case study. Results showed that the models can improve intersection detector counts significantly. Results also showed that ANFIS produced more accurate counts compared to regression and GP.

4.1. METHODOLOGY AND SOLUTION APPROACHES

The methodology of Figure 31 was applied to the case study intersection. In this method, the response variable for training and testing was observed counts which were manually obtained and independent variables were detector counts, time and occupancy which were obtained through central computers or directly from a signal controller. A portion of this data was used for formulating the models and the rest for testing and evaluating the models. Errors of estimated counts were calculated in terms of Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE). These measures may prove useful for future applications of these models. As long as each application requires a different level of accuracy, models may be useful for one application and inappropriate for another one.
Due to uncertainties and nonlinearities that exist between variables, developing a mathematical model that can explain the relationship between detector counts, observed counts, occupancy and time is challenging. ANFIS and Genetic Programming have been used successfully for problems where the development of analytical models proved to be problematic (Teodorovic and Vukadinovic [40] and Negnevitsky [41]). The following sections explain these two solution approaches briefly. A detailed description and discussion can be found in [41] and [42].

Figure 31: Methodology of modifying the detector counts
4.1.1. Genetic Programming

Genetic programming was greatly stimulated in the 1990s by John Koza. Genetic programming applies the same evolutionary approach as genetic algorithms. However, genetic programming is no longer breeding bit strings that represent coded solutions but is a complete computer program that solves a problem at hand. In making a model using conventional methods like regression, the user needs to define the structure of the model. For nonlinear and complex data, this task is challenging especially when there is more than one variable. The user does not need to know the structure of data and evolutionary process since GP does this task.

Solving a problem by GP involves determining the set of variables, selecting the set of functions, defining a fitness function to evaluate the performance of created computer programs, and choosing the method for designating a result of the run.

Before applying GP to a problem, five preparatory steps must be accomplished [44]:

Step 1: Determining the set of terminals: The terminals correspond to the inputs of the computer program to be discovered. The current program requires three inputs, time (x1), detector volume (x2), and occupancy (x3). Step 2: Selecting the set of primitive functions. The functions can be presented by standard arithmetic operations, standard programming operations, standard mathematical functions, logical functions or domain-specific functions. The current program used four standard arithmetic operations: plus, minus, multiplication and division; and mathematical functions: square, root square, and logarithm. Terminals and primitive functions together constitute the building blocks from
which GP constructs a computer program to solve the problem. Step 3: Defining the fitness function. A fitness function evaluates how well a particular computer program can solve the problem. For the current problem, the fitness of the computer program was measured by the mean absolute percentage error (MAPE) of the results produced by the program and the observed counts. The closer MAPE was to zero, the better the computer program. Step 4: Deciding on the parameters for controlling the run. For controlling a run, GP uses the same primary parameters as those used for Genetic Algorithm. They include the population size, the maximum number of generations to be run, crossover and mutation probability and elitism rate. Step 5: Choosing the method for designating a result of the run. It is common practice in GP to designate the best-so-far generated program as the result of a run.

Once these five steps are completed, a run can be performed. The run of GP starts with a random generation of an initial population of computer programs. Each program is composed of functions +, -, × and ÷, square, root square, and logarithm; and terminals x1, x2, and x3.

In the initial population, all computer programs usually have poor fitness, but some individuals are more fit than others [41]. Just as a fitter chromosome is more likely to be selected for reproduction, a fitter computer program is more likely to survive by copying itself into the next generation. In GP, the crossover operator functions on two computer programs which are selected on the basis of their fitness. These programs can have different sizes and shapes. The two offspring programs are composed by recombining randomly
chosen parts of their parents. For example, Figure 32 shows two solutions where crossover makes two offspring. For mutation, a function or terminal will be changed randomly.

After completing the five preparatory steps, the following eight steps will be executed [41]:

Step 1: Assigning the maximum number of generations to be run and probabilities for cloning, crossover and mutation. The sum of the probability of cloning, the probability of crossover and the probability of mutation must be equal to one. Step 2: Generating an initial population of computer programs of size N by combining randomly selected functions and terminals. Step 3: Executing each computer program in the population and calculating its fitness with MAPE and designating the best-so-far individual as the result of the run. Step 4: With the assigned probabilities, a genetic operator will be selected to perform cloning, crossover or mutation. Step 5: If the cloning operator is chosen, one computer program is selected from the current population of programs and will be copied into a new population. If the crossover operator is chosen, a pair of computer programs is selected from the current population, and creates a pair of offspring programs. If the mutation operator is chosen, one computer program from the current population is selected to mutate and it will be placed into the new population. All programs are selected with a probability based on their fitness (i.e., the higher the fitness, the more likely the program is to be selected). Step 6: Step 4 will be repeated until the size of the new population of computer programs becomes equal to the size of the initial population, N. Step 7: The current (parent) population is replaced with the new (offspring) population. Step 8: Program goes to Step 3 and repeats the process until the termination criterion is satisfied.
\[(x^2 - x^3)^2 - x^2 + (x_1x_2)^2\]  
\[\frac{(x_1x_2 + (x_1 - x_3))^2 - x_1}{x_1x_2}\]

\[(x_1x_2 - x^2) \quad \frac{(x_1x_2 + x_1x_3)^2 - x_1}{(x^2 - x^3)^2}\]

Figure 32: Crossover in genetic programming
Four consecutive loop detectors in each lane usually produce systematic under counting during different times. This section describes the application of the Adaptive Neuro-Fuzzy Inference System (ANFIS) that is able to learn these under counting patterns and reduce the cumulative errors.

ANFIS combines the Fuzzy Inference System (FIS) and Artificial Neural Networks (ANN) where the FIS is used to model the relationship between non-linear variables and ANN is used to optimize input and output membership function parameters. FIS can be defined as a process of mapping from a given input to an output using the theory of fuzzy sets and ANN is an artificial neural network consisting of a number of very simple and highly interconnected processors, also called neurons. The neurons are connected by weighted links passing signals from one neuron to another. ANN adjusts the weights to bring the network input/output behavior into line with that of the training data. There are two well-known fuzzy inference systems: Mamdani-style inference and Sugeno-style inference [41].

The Sugeno fuzzy model was used for a systematic approach to generate fuzzy rules from a given input-output data set [41]. A typical Sugeno fuzzy rule can be expressed in the following form:

IF Detector Volume is Medium
AND Detection Time is Peak-hour
THEN Observed Volume is High

The ANFIS adopted in this chapter is represented by a six-layer feedforward neural network [41]. Figure 33 shows the ANFIS architecture used in this research.
Layer 1 is the input layer. Neurons in this layer simply pass external crisp signals to Layer 2 [41]. That is,

\[ y_1^{(1)} = x_1^{(1)}, \quad \text{Equation 3} \]

where \( x_1^{(1)} \) is the input and \( y_1^{(1)} \) is the output of input neuron \( i \) in Layer 1. Layer 2 is the fuzzification layer. Neurons in this layer perform fuzzification. For sake of simplicity in the diagram, Figure 33 shows only two fuzzy members for each variable. For example, two fuzzy members of variable Time (\( x_1 \)) can be defined as Peak (A1) and off-peak (A2). However, the actual members for all variables are more than two. In this chapter, for fuzzification neurons, the bell activation function and the trapezoid activation function were tested.

A bell activation function, which has a regular bell shape, is specified as [41]:

\[ y_1^{(2)} = \frac{1}{1 + \left( \frac{x_1^{(2)} - a_i}{c_i} \right)^2 b_i }, \quad \text{Equation 4} \]

where \( x_1^{(2)} \) is the input and \( y_1^{(2)} \) is the output of neuron \( i \) in Layer 2; and \( a_i, b_i \) and \( c_i \) are parameters that control, respectively, the center, width and slope of the bell activation function of neuron \( i \). Trapezoid activation function is specified by its four corners [41].

Layer 3 is the rule layer. Each neuron in this layer corresponds to a single Sugeno-type fuzzy rule. A rule neuron receives inputs from the respective fuzzification neurons and calculates the firing strength of the rule it represents. In an ANFIS, the conjunction of the
rule antecedents is evaluated by the operator product. Thus, the output of neuron \( i \) in Layer 3 is obtained as,

\[
y^{(3)}_i = \prod_{j=1}^k x^{(3)}_{ji}, \quad \text{Equation 5}
\]

where \( x^{(3)}_{ji} \) are the inputs and \( y^{(3)}_i \) is the output of rule neuron \( i \) in Layer 3.

For example in Figure 33,

\[
y^{(3)}_{\Pi 1} = \mu_{A1} + \mu_{B1} = \mu_1, \quad \text{Equation 6}
\]

where the value of \( \mu_1 \) represents the firing strength, or the truth value, of Rule 1 [41].

Layer 4 is the normalization layer. Each neuron in this layer receives inputs from all neurons in the rule layer, and calculates the normalized firing strength of a given rule. The normalized firing strength is the ratio of the firing strength of a given rule to the sum of firing strengths of all rules. It represents the contribution of a given rule to the final result.

Thus, the output of neuron \( i \) in Layer 4 is determined as [41],

\[
y^{(4)}_i = \frac{x^{(4)}_i}{\sum_{j=1}^n x^{(4)}_{ji}} = \frac{\mu_i}{\sum_{j=1}^n \mu_j} = \bar{\mu}_i, \quad \text{Equation 7}
\]

where \( x^{(4)}_{ji} \) is the input from neuron \( j \) located in Layer 3 to neuron \( i \) in Layer 4, and \( n \) is the total number of rule neurons. For example,

\[
y'^{(4)}_{N1} = \frac{\mu_1}{\mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6} = \bar{\mu}_1 \quad \text{Equation 8}
\]
Layer 5 is the defuzzification layer. Each neuron in this layer is connected to the respective normalization neuron, and also receives initial inputs, \( x_1, x_2, \) and \( x_3 \). A defuzzification neuron calculates the weighted consequent value of a given rule as [41],

\[
y^{(5)}_i = x^{(5)}_i [k_{i0} + k_{i1}x_1 + k_{i2}x_2 + k_{i3}x_3] = \mu_1 [k_{i0} + k_{i1}x_1 + k_{i2}x_2 + k_{i3}x_3].
\]

Equation 9

where \( x^{(5)}_i \) is the input and \( y^{(5)}_i \) is the output of defuzzification neuron \( i \) in Layer 5, and \( k_{i0}, k_{i1}, k_{i2}, \) and \( k_{i3} \) is a set of consequent parameters of rule \( i \).

Layer 6 is represented by a single summation neuron. This neuron calculates the sum of outputs of all defuzzification neurons and produces the overall ANFIS output, \( y \) [41],

\[
y = \sum_{i=1}^n x^{(6)}_i = \sum_{i=1}^n \mu_1 [k_{i0} + k_{i1}x_1 + k_{i2}x_2 + k_{i3}x_3] \quad \text{Equation 10}
\]

It is often difficult or even impossible to specify a rule consequent in a polynomial form. Conveniently, it is not necessary to have any prior knowledge of rule consequent parameters for an ANFIS to deal with a problem [41]. An ANFIS learns these parameters and tunes membership functions. An ANFIS uses a hybrid learning algorithm that combines the least-squares estimator and the gradient descent method (see [43] for more details).
4.2. CASE STUDY

A test was conducted at the Kietzke/Moana intersection in Reno, NV on June 12, 2013. The attributes of this intersection are shown in Table 1.

4.2.1. Data Collection

During specific time periods, data was collected using a Miovision Scout Video Collection Unit. This data was the observed data for calculating the accuracy of loop detectors and to validate proposed models. For the same time periods, volume and occupancy were also obtained from detectors. Figure 34 shows a sample of detector data at the intersection.
4.2.2. Data Analysis

The volume time series, shown for both the observed and detector counts in Figure 35, provide a visual representation of the differences between the observed and the detector counts during different time periods. In all approaches, detector counts were lower than the observed counts. Especially during peak hours in the southbound and the westbound approaches, the differences between the detector and the observed counts were significant. This happened mainly because loops were spliced together and as a result, all cars that are within the detection zone at each time were counted as a single vehicle.

The accuracy of these detectors can be expressed using one of the following two error quantity values [1]:

1. Mean Absolute Percent Error (MAPE) (Equation 11) or
2. Root Mean Squared Error (RMSE) (Equation 12).

\[
MAPE(\%) = \frac{\sum_{i=1}^{n} |D_i - B_i|}{n} \quad \text{Equation 11}
\]

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (D_i - B_i)^2}{n}} \quad \text{Equation 12}
\]

where:

MAPE: Mean Absolute Percentage Error

RMSE: Root Mean Squared Error

D_i: detector data value

B_i: reference (observed) data value

n: total number of time intervals

The detector accuracy was interpreted in terms of count errors. The preferred measure for count error was the Mean Absolute Percent Error (MAPE, %) as its range of values was large. Note that the RMSE was sensitive to the relative volume of traffic being observed.

The RMSE for devices at one intersection test site cannot be directly compared to RMSE values for devices at other intersection test sites. Similarly, the RMSE from one approach at an intersection cannot be compared to the RMSE values from another approach. However, when the number of vehicles matters, RMSE is a better measure. For example, if observed volume was 10 veh/min and the detectors report of nine veh/min, then MAPE produced a 10 percent error while RMSE indicated only one veh/min error. Therefore, in comparison to RMSE, MAPE does not show the quantity of errors.
4.2.3. Modeling

To get an idea of possible relationships between detector volume and occupancy with observed counts, scatterplots were constructed in Minitab (Figures 36). Except in the case of the westbound approach, the detector counts indicated a strong association with observed counts.
To identify possible associations between detector and observed counts, ANFIS and GP were performed for each approach. For a better comparison of results, regression models were also developed.

Table 3 shows the results of regression and GP models. These models were simple and as long as there was a strong association between variables, both GP and regression produced similar accuracy. However, when association of variables became complex, like for the westbound approach, GP produced better models compared to regression. ANFIS models could not be represented as mathematical models. Table 4 demonstrates fuzzy sets of two variables, Time and Detector Volume. Occupancy was not significantly improving the results; therefore, for sake of simplicity, it was omitted from the modeling process. Figure 37 shows the input of training data into the different methods’ models. Figure 38 also shows 3-D surface plot of variables Time, Detector Volume and ANFIS output based on a training data set. This diagram shows the non-linearity relationship of variables.

In general using ANFIS models was not as easy as regression and GP models. However, fortunately there are applications that can be used to facilitate the usage of ANFIS models. One of them is “anfisedit” Graphical User Interface (GUI) in MATLAB. For each approach, all information of fuzzy sets should be entered into the anfisedit GUI. Both bell-shaped and trapezoid membership functions were tested for approaches. Bell-shaped membership functions, despite of their complexity, did not significantly improve the models compared to trapezoid membership functions. In Table 4, northbound and westbound variables had three members in their fuzzy sets and eastbound and southbound
had four and five respectively. These numbers represent the number of categories for each variable. For example, in the southbound approach, variable “Detector Volume” categories can be named as: Very Low, Low, Medium, High, and Very High. The numbers inside the brackets in Table 4 show the four corners of trapezoid members. For example, member Medium in southbound Detector Volume has been defined by 132.8, 146.6, 167.5, and 181. This means that membership of volumes less than 132.8 and bigger than 181 are considered zero in this category, and assigned value of one from 146.6 to 167.5. Other volume ranges have a membership between zero and one. However, users do not need to engage in these calculations since anfisedit GUI generates all of the output results.

### Table 3: Regression and GP models for to modify detector counts

<table>
<thead>
<tr>
<th>Approach</th>
<th>Regression Models</th>
<th>GP Models</th>
<th>R-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB</strong></td>
<td>$NBB = -1.2 + 1.07 NBD + 0.144 NBO^*$</td>
<td>$NBB = -\frac{4NBD^3}{10^5} + 0.02NBD^2 - 0.6NBD + 59.9$</td>
<td>0.885</td>
</tr>
<tr>
<td><strong>EB</strong></td>
<td>$EBB = 10.7 + 1.04 EBD + 0.336 EBO$</td>
<td>$EBB = 3.3 EBD - 2.5EBO - 0.02EBD \times EBO + 0.02EBO^2 + 53.2$</td>
<td>0.879</td>
</tr>
<tr>
<td><strong>SB</strong></td>
<td>$SBB = 8.7 + 1.04 SBD + 0.170 SBO$</td>
<td>$SBB = -\frac{8SBD^3}{10^5} + 0.04SBD^2 - 4.3SBD + 0.15SBO + 253$</td>
<td>0.646</td>
</tr>
<tr>
<td><strong>WB</strong></td>
<td>$WBB = 107 + 0.263 WBD + 0.665 WBO$</td>
<td>$WBB = 8WBO - 0.13WBD - 0.01(WBD - WBO)^2 - 0.03WBO^2 - 315$</td>
<td>0.199</td>
</tr>
</tbody>
</table>

* The two first letters of variables refer to approach, and the last letter of variables refer to D: Detector count, and B: Observed count, and O: Occupancy
Table 4: ANFIS membership function models for detector counts

<table>
<thead>
<tr>
<th>Time</th>
<th>Detector Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NB</strong></td>
<td></td>
</tr>
<tr>
<td>[-10.2, -3.8, 10, 18.51]</td>
<td>[5.918, 13, 22.83, 27.94]</td>
</tr>
<tr>
<td>[23.67, 29.28, 37.8, 44.2]</td>
<td></td>
</tr>
<tr>
<td><strong>EB</strong></td>
<td></td>
</tr>
<tr>
<td>[-6.467, -2.2, 2.717, 8.839]</td>
<td>[2.005, 6.964, 16.17, 19]</td>
</tr>
<tr>
<td>[14.09, 19.89, 28.37, 32]</td>
<td>[172.6, 211.4, 270.3, 309.3]</td>
</tr>
<tr>
<td><strong>SB</strong></td>
<td></td>
</tr>
<tr>
<td>[-4.6, -1.4, 3.326, 6.417]</td>
<td>[3.355, 6.527, 11.19, 14.44]</td>
</tr>
<tr>
<td>[11, 14.4, 19.29, 22.78]</td>
<td>[63.85, 77.65, 98.37, 112.2]</td>
</tr>
<tr>
<td>[27.47, 30.57, 35.4, 38.6]</td>
<td>[132.8, 146.6, 167.5, 181]</td>
</tr>
<tr>
<td><strong>WB</strong></td>
<td></td>
</tr>
<tr>
<td>[-10.2, -3.8, 6.633, 11]</td>
<td>[34.3, 58.7, 95.3, 119.7]</td>
</tr>
<tr>
<td>[-10.2, -3.8, 6.633, 11]</td>
<td>[34.3, 58.7, 95.3, 119.7]</td>
</tr>
<tr>
<td>[20.45, 25.96, 37.8, 44.2]</td>
<td>[157.7, 181.4, 217.3, 241.7]</td>
</tr>
</tbody>
</table>
Figure 36: Scatter plots for detector volume and observed counts (left) and detector occupancy and observed counts (right) at Kietzke/Moana intersection, Reno, NV

Figure 37: Detector and observed counts versus models counts, northbound
4.2.4. Results

Based on the models presented in Table 3 and ANFIS models, detector counts were modified and MAPE and RMSE were calculated (Figure 39). Westbound and eastbound detector errors were more than 30 percent, and northbound and southbound 15 percent and 20 percent respectively. However, after data modification, all approaches had MAPE less than 12 percent with all methods. While GP produced slightly better results compared to regression, ANFIS accuracy was significantly higher among the selected methods.

![3-D surface plot of variables Time, detector Volume and ANFIS output](image)
Figure 39: MAPE and RMSE of detectors and models
4.3. SUMMARY AND CONCLUSION

In this chapter, the accuracy of detectors in regard to producing counts was studied. Results showed that the counts were not reliable and had high errors. However, detector counts were able to be modified since there was a relationship with observed counts. Calibration models for one intersection were made based on volume and occupancy. All methods produced errors less than 14 percent.

For modeling, in addition to regression, genetic programming (GP) and ANFIS were used. GP uses an evolutionary process to develop a good model for a data set and ANFIS uses fuzzy logic for modelling and neural networks for optimizing its input and output membership functions. Results showed that by using existing signal control devices, acceptable intersection volume counts could be obtained without any further cost. Also, results showed that ANFIS models provided better results than either regression or GP.
5. DETERMINING TURNING MOVEMENT PROPORTIONS IN SHARED LANES

Turning vehicle volumes at intersections are critical inputs for various transportation engineering studies such as Level of Service (LOS), signal timing and traffic safety studies. There are various types of detectors, including loop detectors, which are able to produce volume estimates in addition to providing intersection control. However, such detectors are unable to produce turning movement counts in shared lanes. Current practice is gathering such information through manual counts, which is very costly. The purpose of this chapter is to provide three methods to estimate turning movement proportions in shared lanes. These methods were tested using simple linear regression and Genetic Programming (GP).

A review of prior studies and current technologies indicate that there is no automated method to calculate the proportion of turning movement volumes in shared lanes without downstream detectors (also known as departure or exit detectors). Most studies relied on additional downstream detectors to estimate the shared lane turning volume while many intersections do not have such detectors [32, 33, 34, 35, 37, 38]. Modern signal controllers are enhanced with advanced data monitoring and logging capabilities, making it possible to record high resolution detector and signal timing information. The methods proposed in this chapter can be used to estimate turning movement volumes in shared lanes using such information. This allows continuous traffic volume counts without incurring additional costs.
5.1. METHODOLOGY

Three methods proposed in this chapter to estimate shared lane proportions based on stop bar detectors include: 1) Network Equilibrium (NE) which utilizes the actual upstream and downstream counts at other intersections, 2) Volume and Queue (VQ) length of shared lanes compared to adjoining lanes, and 3) Flow Characteristics (FC) of shared lanes. These methods can be used together in a network to obtain better results. However, obtaining proportions that have been observed in the field remains an alternative method at intersections where there is no possibility of using these methods.

5.1.1. Network Equilibrium (NE)

Network equilibrium is a method which has been mentioned by Gentili and Mirchandani [39]. However, it is explained in this chapter to complete the list of possible methods. Based on volume equilibrium, it is possible to estimate shared lane proportion if there are enough equations for every unknown movement. For example, Figure 40 displays a simple network where there are two intersections. The following equation can be applied to calculate west bound through at intersection $i$:

$$WT_i^t = -NL_i^t - SR_j^{t+\Delta t} + WR_j^{t+\Delta t} + WT_j^{t+\Delta t} + WL_j^{t+\Delta t} - \delta_{ij}$$  

Equation 13

where:

$WT$: westbound through
$NL$: northbound left
$SR$: southbound right
$WR$: westbound right
$WL$: westbound left
\( t \): time interval \( t \), for example 7:00:00 to 7:15:00

\( i,j \): intersection numbers

\( \Delta t \): travel time between intersections \( i \) and \( j \)

\( \delta_{ij}^{t+\Delta t} \): additional trips generated between intersections \( i \) and \( j \) during time interval \( t \)

If the distance between two intersections is short, or there are no significant volume fluctuations during different time intervals, \( \Delta t \) can be assumed as zero. Also \( \delta_{ij}^{t+\Delta t} \) can be considered zero if there is not a major trip generator between intersections \( i \) and \( j \). The advantage of this method is that high resolution detector data and traffic signal information are not required. Fifteen-minute counts can be used for accurate calculations. A low-cost approach to obtaining detector and signal timing information is through advanced signal control software. Technologies in signal control systems have advanced dramatically in recent years. The latest signal control hardware and software are equipped with enhanced features for providing high-resolution detector and signal timing information. Figure 41 (a) displays automatically reported detector information from an existing intersection in Reno. This data sample shows the vehicle counts from the detectors aggregated in 15-minute intervals. Figure 41 (b) displays high resolution data extracted from signal controller using toolbox developed by TTI [36].

![Figure 40: Example Network with Two Intersections](image-url)
5.1.2. Volume and Queue (VQ) of Length of Shared Lanes

There are some cases where a shared lane has some adjoining lanes with similar movements. For this condition, the probability that vehicles that have two or more options
(using shared lane or adjoining lane(s)) use the shared lane when the queue length in the
shared lane is more than adjoining lane(s) is very low. This probability depends on driver
habits, upstream and downstream intersection configuration and distance to the
intersection. Figure 42 demonstrates the intersection at 8th Street and Center Street, Reno,
NV. At the westbound approach, there are two exclusive lanes and one shared lane.
Observations show that when there are several vehicles in the shared lane while the two
other through lanes experience less traffic, probably most vehicles in shared lane would
turn right. There might be an association between the ratio of shared lane volume to the
adjoining lane volumes and the ratio of right turns to through volumes in shared lanes. The
following equations demonstrate this association:

$$r_{s,a}^t = \frac{v_s^t}{v_a^t}$$  \hspace{1cm} \text{Equation 14}

$$r_{r,t}^t (\text{or } r_{l,t}^t) = \frac{v_r^t (\text{or } v_l^t)}{v_t^t}$$  \hspace{1cm} \text{Equation 15}

$$r_{r,t}^t (\text{or } r_{l,t}^t) = f(r_{s,a}^t)$$  \hspace{1cm} \text{Equation 16}

where:

$r_{s,a}^t$: ratio of shared lane volume to volume of adjoining lanes with same direction at time interval $t$

$r_{r,t}^t (\text{or } r_{l,t}^t)$: ratio of right (or left) turns to through volume in shared lane at time interval $t$

$v_s^t$: total volume in shared lane at time interval $t$

$v_a^t$: total volume at adjoining lane(s) at time interval $t$, for example if shared lane is right and through, then
all adjoining through lanes should be considered

$v_r^t (\text{or } v_l^t)$: total right (or left) turns in shared lane at time interval $t$
$v_i^t$: total through volume in shared lane at time interval $t$

After developing Equation 16 for each shared lane, the turning volumes may be estimable. Note that this method is applicable only for shared lanes that have at least one adjoining lane with similar movement. In addition, some shared lanes are wide enough for right turn vehicles to overpass stopped vehicles to make their turns. At these shared lanes, this method cannot be used.

The advantage of this method is that high resolution detector data and signal timing information are not required. However, shorter time intervals (for instance, one minute), provide better results. Usually this method can be used if the downstream intersection is not close to the study intersection, because some vehicles use more congested lanes due to their turning movement at the next intersection.

Figure 42: Intersection at 8th Street and Center Street, Reno, NV
5.1.3. Flow Characteristics (FC) of Shared Lanes

If there is high resolution data from stop bar detectors and traffic signal data, then it would be possible to develop an estimation model based on vehicle headways in shared lanes. The turning movement can be related to headway, the position of a vehicle in the queue, and vehicle type. Also, intersection geometry can affect the significance of headway on the turning movements. Here, the geometry can be summarized as the turning radius. The turning radius depends on whether the shared lane is a right or left turn, the number of lanes accessible for turning vehicles, width of each lane, angle of turning, and number of opposing lanes. Figure 43 demonstrates the problem. Therefore:

\[ P(t_{i,r}) \text{ or } P(t_{i,l}) = f(h_i, h_{fi}, cp_i, ct_i, cf_i) \]  \hspace{1cm} \text{Equation 17}

Where:

\[ P(t_{i,r}) \text{ or } P(t_{i,l}) \]: probability of vehicle \( i \) turning right (or left)

\( h_i \): headway, the time difference of vehicle \( i \) from its front vehicle when passing stop bar

\( h_{fi} \): summation of each headway with previous car headway

\( cp_i \): position of vehicle \( i \) in the line

\( ct_i \): type of vehicle \( i \)

\( cf_i \): type of front vehicle of vehicle \( i \)
In this method, when the turning radius is large, the difference between the headways of through and turning vehicles is not significant. Therefore, the first step is to check the applicability of this method for an intersection based on turning radius. In other words, to use this method, there must be an association between turning movement and headway.

One critical step of any volume counting systems is to attain signal timing and detector information. Most of the previous studies relied on additional data recording devices in order to obtain detailed data. The Texas Transportation Institute (TTI) developed a data acquisition toolbox that can be directly connected to a TS2 controller cabinet [36]. The TTI toolbox has a hardware and software tool to automatically download detector
information and summarize various performance measures. SMART Signal Inc. also has been developing the AdaptiTrolDCU device, a signal controller data collection and communication adaptor. This adaptor can record high resolution detector and signal timing data and work with any NEMA signal control cabinet. Sample data was collected using the device as shown in Figure 41 (b). As can be seen, the on/off times of each detector and signal phase are recorded at the milliseconds resolution.

An issue regarding this method is the vehicles which arrive during green time when there is not a queue. These vehicles are not lined up during the red time. As a result, their headway is random and generates noise in the models. One method to deal with this issue is to share these vehicles with ratio of vehicles which have headway significantly related to turning vehicles. Furthermore, the first vehicle at the queue should be divided between through and right (or left) turn based on this ratio.

If the shared lane is a through and right turn and the intersection is not a No Turn on Red (NTOR), all vehicles that activate the detector during red time are categorized as a right turn. At these intersections, if the pedestrian volume at cross street is high, the through and turning headways lose their significance. Similar to the VQ method, some shared lanes are wide enough for right turn vehicles to overpass stopped vehicles in order to make their turns. At these shared lanes, this method cannot be utilized.

One of the factors that makes headway a significant parameter on turning vehicles is driver behavior. In some cities, drivers are more conservative and keep a slower speed during turning, while in some cities the difference between speeds of turning and through vehicles is not significant. In addition, observations show that, during the peak hour, the
behavior of drivers change. However, the effects of these factors can be summarized as the significant difference of through vehicles compared with turning vehicles, and should be tested by a pilot study.

### 5.1.4. Observed Proportions in the Field

When none of these three methods can be applied, observed proportions in the field can be applied for future estimates. This means the proportion of turning volumes at each time interval would be applied in the future as long as there is an insignificant difference between turning proportions during different days, weeks and months.

### 5.2. CASE STUDY

The methods provided in this chapter were applied on three different intersections in Reno, Nevada. In the following sections, the results of each method are described.

#### 5.2.1. Network Equilibrium (NE)

The intersection at 9th Street and Sierra Street, Reno, NV shown in Figure 44 was selected for the NE method. At this intersection, eastbound through and eastbound right share a lane. All required movements for calculating the proportion of this shared lane have a loop detector. The two unknown movements can be calculated using the following equations:

\[
ER_i^t = -SR_j^t - ST_j^t + ST_i^t + WL_i^t \\
ET_i^t = -ET_k^t - EL_k^t - ER_k^t + SL_i^t + NR_i^t
\]

where:

ER: eastbound right
$SR$: southbound right

$ST$: southbound through

$WL$: westbound left

$ET$: eastbound through

$EL$: eastbound left

$SL$: southbound left

$NR$: northbound right

$t$: time interval $t$, for example 7:00:00 to 7:15:00

$i, j, k$: intersection numbers

*Figure 44: Intersection of 9th Street and Sierra Street, Reno, NV*
The travel time between intersections \(i\) and \(j\) or \(k\) is very short. As a result, \(\Delta t\) has been considered zero. In addition, there is not a significant trip generation between intersections \(i\) and \(j\) or \(k\). Therefore, additional trips generated between intersections \(i\) and \(j\) or \(k\) during time interval \(t\), \(\delta_{ij}^{t+\Delta t}\) or \(\delta_{ik}^{t+\Delta t}\), has been ignored.

### 5.2.2. Volume and Queue Length (VQ) of Shared Lanes

The intersection of 8th Street and Center Street, Reno, NV displayed in Figure 42 was selected for the VQ method. For the westbound approach, there are two exclusive lanes and one shared lane for right and through movements. A prediction model was made for this shared lane. Table 5 shows the sample of data gathered for this intersection.

#### Table 5: Sample of Data Collected for VQ Method at Intersection of 8th Street and Center Street, Reno, NV

<table>
<thead>
<tr>
<th>TIME INTERVAL (1 MINUTE) (t)</th>
<th>SHARED LANE VOLUME (v_s^t)</th>
<th>ADJOINING LANE(S) VOLUME (v_a^t)</th>
<th>RATIO OF SHARED LANE TO ADJOINING LANES (r_{s,a}^t)</th>
<th>NO. OF RIGHT Turner (v_r^t)</th>
<th>RATIO OF RIGHT TURNS (r_{r,t}^t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>25</td>
<td>0.24</td>
<td>3</td>
<td>0.50</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>18</td>
<td>0.39</td>
<td>5</td>
<td>0.71</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>13</td>
<td>0.46</td>
<td>4</td>
<td>0.67</td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>21</td>
<td>0.86</td>
<td>18</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>16</td>
<td>0.44</td>
<td>5</td>
<td>0.71</td>
</tr>
</tbody>
</table>

### 5.2.3. Flow Characteristics (FC) of Shared Lanes

To test the validity of the FC method, several shared lanes with different turning radii were required, because turning radius is one of the factors that affects the significance of headways. These intersections are: westbound shared lane at the intersection of 8th Street
and Center Street in Figure 42, with a turning radius equal to 16 ft.; eastbound shared lane at the intersection of 9th Street and Sierra Street in Figure 44, with a turning radius equal to 35 ft.; and the northbound shared lane at the intersection of North McCarran Blvd and Clear Acre Ln in Figure 45, with a turning radius equal to 100 ft.

For each of the shared lanes, collected data was tabulated and the sample can be seen in Table 6. In these tables, Headway, $h_i$, is the time difference between the vehicle $i$ and its front vehicle when passing the stop bar. The next column, $h_{fi}$, is the summation of each headway with the previous vehicle’s headway. The reason for defining this variable is that the front car headway can affect the headway of its following car. Car Position, $cp_i$, means the position of the car in the line. Car Type, $ct_i$, was also added as a variable because it is likely that heavy vehicles have lower speeds, thus resulting in a higher headway. Due to propagation of this effect to following cars, Front Car Type variable, $cf_i$, was also added. Lastly, $td_i$, shows the turning direction of cars. This variable was used as the response variable, while others remain independent variables.

<table>
<thead>
<tr>
<th>CYCLE NO. $cn_j$</th>
<th>HEADWAY $h_i$</th>
<th>FRONT CAR HEADWAY $h_{fi}$</th>
<th>CAR POSITION $cp_i$</th>
<th>CAR TYPE $ct_i$</th>
<th>FRONT CAR TYPE $cf_i$</th>
<th>TURNING DIRECTION $td_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>c*</td>
<td>c</td>
<td>T</td>
</tr>
<tr>
<td>1</td>
<td>2.24</td>
<td>2.24</td>
<td>2</td>
<td>c</td>
<td>c</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>2.31</td>
<td>4.55</td>
<td>3</td>
<td>t</td>
<td>c</td>
<td>T</td>
</tr>
<tr>
<td>1</td>
<td>3.01</td>
<td>5.32</td>
<td>4</td>
<td>c</td>
<td>t</td>
<td>T</td>
</tr>
</tbody>
</table>

* c: private car, t: truck, T: through, R: right turn
5.3. DATA ANALYSIS

5.3.1. Modeling

The NE method needs direct calculation and did not require prediction models. For the VQ and FC methods, linear regressions were performed in R® (see Appendix D). VQ was modeled as a continuous variable and FC as discrete. R® was able to generate models for both continuous and discrete variables. Genetic Programming (GP) was also used for modeling. GP is an evolutionary algorithm-based methodology that can develop a model based on a set of data. GP was generally able to find a more accurate model compared to other conventional methods. The main advantage of GP, in comparison with traditional
regression methods, is that it finds both parameters and the best model structure for
the data set. While in traditional regression methods, users need to determine the model
structure. Another advantage of GP was its ability to find relatively respectable models for
complex data sets. In this chapter, GPTIPS was used for generating the models. GPTIPS is
a Genetic Programming tool for use with MATLAB™. This package enables users to
identify hidden and non-linear relationships in data sets and automatically creates compact
and accurate non-linear equations to predict the behavior of physical systems [45, 46].
Table 7 summarizes the models developed with regression and GP for the shared lanes of
this case study.

<table>
<thead>
<tr>
<th>MODELLING METHOD</th>
<th>N MCCARRAN BLVD/CLEAR ACRE LN</th>
<th>9TH ST/ SIERRA ST</th>
<th>8TH ST/ CENTER ST</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NE</strong></td>
<td>Regression: Not applicable</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>GP</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>Analytical Calculation: Not</td>
<td>Not applicable</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>applicable due to un-controlled southbound right turn</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ ER_i^T = -SR_i^T - ST_i^T + ST_i^T + WL_i \]
\[ ET_i^T = -ET_k^T - EL_k^T - ER_k^T + SL_i^T + NR_k^T \]

Not applicable due to downstream un-signalized intersection (9th St/ Center St)

| **VQ**           | Regression: Not applicable    | Not applicable   | Not applicable   |
|                  | GP                            | Not applicable   | Not applicable   |
|                  | Analytical Calculation: Not   | Not applicable   | Not applicable   |
|                  | applicable due to no existing same-direction adjoining lane | | |

\[ r_{r,t}^f = 0.2 + 0.6 r_{s,\alpha} \]
\[ r_{r,t} = 4.55 + 2.5 r_{s,\alpha} - 3.75 e^{-r_{s,\alpha}} + 0.25 r_{s,\alpha}^2 \]

| **FC**           | Regression: P(t_l) = \( \frac{e^{12.6 - 3.6h_l}}{1 + e^{12.6 - 3.6h_l}} \) | P(t_r) = \( \frac{e^{2.7 - 2.7h_l}}{1 + e^{2.7 - 2.7h_l}} \) | P(t_r) = \( \frac{e^{3.2 - 2.2h_l}}{1 + e^{3.2 - 2.2h_l}} \) |
|                  | GP                            | P(t_r) = -3 + 2.8√[h_l - 3.8] + 0.1 e^{h_l - 0.1 h_l^2} | P(t_r) = 1.5 + \frac{12.2}{h_l - 5.6} |
|                  | Analytical Calculation: Not   | Not applicable   | Not applicable   |

* Other variables due to lack of significance have been omitted
5.3.2. Analysis of Results

The accuracy of models was interpreted in terms of Mean Absolute Percentage Error (MAPE, %):

\[
MAPE(\%) = \frac{\sum_{i=1}^{n} \left| \frac{M_i - B_i}{B_i} \right|}{n} \quad \text{Equation 20}
\]

where:

\textit{MAPE}: Mean Absolute Percentage Error

\(M_i\): model estimation at lane \(i\)

\(B_i\): reference (base) count at lane \(i\)

\(n\): total number of intervals

Table 8 summarizes the MAPE calculations of different methods for the case study intersections. MAPE was calculated for vehicle by vehicle and hourly average, except for the NE method, where vehicle by vehicle MAPE calculation cannot be applied. The hourly average MAPEs for all methods with GP modeling are less than 7 percent except at the McCarran Blvd/Clear Acre Ln intersection due to a large turning radius (100 ft). At this intersection, models showed low significance for through and turning headways and as a result, errors were higher than the two other intersections. In the FC method, when turning radius began to increase, the reliability of models decreased. However, a higher turning radius usually means shared lanes have intersected with a major street. In these cases, downstream intersections of shared lanes are usually signalized. Therefore, it is feasible to use the highly accurate NE method.
In all cases, GP models were more accurate than regression, except for the hourly average of the FC method at the intersection of 9th St. and Sierra St. where the calculated MAPE for the GP model was 2 percent higher than the regression method. In other cases, GP MAPE was up to 7 percent less. For a better demonstration, Figure 46 was created based on MAPE calculations for different methods. Hourly average MAPE showed that methods produced accurate turning volumes except when the turning radius was large in the FC method. In all methods, the accuracy of hourly average was greater than 85 percent at the case study intersections using the GP.

5.4. SUMMARY AND CONCLUSIONS

Current practice for automated calculation of the proportion of turning vehicle movements in shared lanes is to use downstream detectors in addition to stop bar detectors and traffic signal information. However, many cities do not have downstream detectors. In this chapter, three different methods were proposed to calculate the proportion of turning vehicle movements in shared lanes at signalized intersections. These methods, wherever conditions were met, are easy to apply and did not need considerable investment. The methods were applied at three intersections in Reno, NV. Results indicated that these methods can be applied to produce accurate counts. Genetic Programming (GP) was used for modeling, and was found to generate more accurate models compared to conventional regression. Many DOT divisions and regional transportation agencies will likely benefit from this study in terms of traffic operations, safety, intermodal planning, traffic information, and performance analysis. These methods can be used for almost every kind of detectors including loop detectors.
Table 8: Mean Absolute Percentage Error (MAPE) of Different Methods at Case Study Intersections

<table>
<thead>
<tr>
<th>MODELING METHOD</th>
<th>INTERSECTION</th>
<th>TURNING RADIUS (FT)</th>
<th>TIME INTERVAL</th>
<th>MAPE (%)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Regression</td>
<td>GP</td>
<td>Analytical Calculation</td>
<td></td>
</tr>
<tr>
<td>FLOW CHARACTERISTICS (FC)</td>
<td>8th St and Center St</td>
<td>16</td>
<td>Vehicle by Vehicle</td>
<td>20</td>
<td>14</td>
<td>na</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hourly Average</td>
<td>8</td>
<td>7</td>
<td>na</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9th St and Sierra St</td>
<td>35</td>
<td>Vehicle by Vehicle</td>
<td>21</td>
<td>20</td>
<td>na</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hourly Average</td>
<td>3</td>
<td>5</td>
<td>na</td>
<td></td>
</tr>
<tr>
<td></td>
<td>McCarran Blvd and Clear Acre Ln</td>
<td>100</td>
<td>Vehicle by Vehicle</td>
<td>28</td>
<td>25</td>
<td>na</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hourly Average</td>
<td>27</td>
<td>15</td>
<td>na</td>
<td></td>
</tr>
<tr>
<td>VOLUME AND QUEUE (VQ)</td>
<td>8th St and Center St</td>
<td></td>
<td>Vehicle by Vehicle</td>
<td>17</td>
<td>15</td>
<td>na</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hourly Average</td>
<td>4</td>
<td>1</td>
<td>na</td>
<td></td>
</tr>
<tr>
<td>NETWORK EQUILIBRIUM (NE)</td>
<td>9th St and Sierra St</td>
<td></td>
<td>Vehicle by Vehicle</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hourly Average</td>
<td>na</td>
<td>na</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Figure 46: Mean Absolute Percentage Error (MAPE) of Different Methods at Case Study Intersections
6. ESTIMATING INTERSECTION TURNING VOLUMES FROM TRAFFIC SIGNAL INFORMATION

Actuated traffic signals usually use loop detectors. The current practice in many cities is to install four consecutive loop detectors in each lane to reduce the chance of undetected vehicles. Due to practical reasons, all four loop detectors in each lane and other detectors referring to the same phase are usually spliced together. Thus, it is possible for several vehicles to be counted as one single car. This way of detector wiring to the cabinet generally reduces the accuracy of detectors for collecting traffic volumes. The preliminary studies in this research show cases with an error greater than 75 percent. Therefore, the purpose of this chapter is to provide a simple method to obtain turning volumes from signal information in actuated non-coordinated traffic signals without using loop detector data.

To generate the required data, a simulation was performed in VISSIM with different input volumes. To change turning volumes, a code was developed in COM interface. With this code, the inputs did not have to be changed manually. In addition, the COM code stored the outputs. Data were then exported to a single Excel file. Afterwards, regression and the Adaptive Neural Fuzzy Inference System (ANFIS) were used to build models to obtain turning volumes. The accuracy of these models was defined in terms of Mean Absolute Percent Error (MAPE, %). Results of the two case studies showed that during peak hours, there was a high correlation between actuated green time and volumes. This method did not need extensive data collection and was easy to employ. The results also showed that ANFIS produced more accurate models compared to regression.
6.1. PROBLEM STATEMENT

The best source to obtain intersection-turning volumes is signal-controlling detectors. They are in place for operation of the traffic signals so they can be used for obtaining counts without any extra cost. An ITE report [15] proved that loop detectors can produce excellent counts if location and wiring of loops are appropriate. However, they usually do not have these ideal configurations and as a result, count errors are significant. To evaluate the accuracy of loop detectors for collecting turning movement counts, a study was conducted on two intersections in Reno and Sparks, NV. The accuracy of detectors can be expressed using one of the following two error quantity values [1]:

1. Mean Absolute Percent Error (MAPE) (see Equation 21) or

2. Root Mean Squared Error (RMSE) (see Equation 22).

\[
MAPE(\%) = \frac{\sum_{i=1}^{n} \left| \frac{D_i - B_i}{B_i} \right|}{n}
\]  
(Equation 21)

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (D_i - B_i)^2}{n}}
\]  
(Equation 22)

where:

MAPE: Mean Absolute Percentage Error

RMSE: Root Mean Squared Error

\(D_i\): detector data value

\(B_i\): reference (base) data value
n: total number of time intervals

At the Kietzke Ln and Moana Ln intersection in Reno, MAPE was up to 35 percent and in the Sparks intersection (Sparks Blvd and Prater Way), it was up to 75 percent (Figure 47). The RMSE shown in Figure 47 is based on volume per 15 minutes (veh/15-min). In the Reno intersection, RMSE was as high as 100 veh/15-min and in Sparks, it was up to 180 veh/15-min. This measures shows very high errors which suggested that loop detectors are not reliable for obtaining turning movements.

The unreliability of loop detectors for producing turning movements was the incentive to develop a method to obtain automated intersection turning volumes without
using detector data. Except for detector data, the only remaining source of available automated data is signal logs. Figure 48 shows a signal log sample from the intersection of Virginia St. and McCarran Blvd. in Reno, NV. Table 9 also shows the signal configuration for this intersection. The following sections address the research question: Can turning volumes be estimated based on this signal information without using loop detector data?

### Split History

**ID:** 164  
**Name:** VIRGINIA & MCCARRAN NORTH  
**Begin Date/Time:** 06/23/2014 12:00 am  
**End Date/Time:** 06/23/2014 11:59 pm

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Pattern</th>
<th>Cycle</th>
<th>SP1</th>
<th>SP2</th>
<th>SP3</th>
<th>SP4</th>
<th>SP5</th>
<th>SP6</th>
<th>SP7</th>
<th>SP8</th>
<th>SP9</th>
<th>SP10</th>
<th>SP11</th>
<th>SP12</th>
<th>SP13</th>
<th>SP14</th>
<th>SP15</th>
<th>SP16</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/23/2014 00:16 AM</td>
<td>25a</td>
<td>24</td>
<td>12</td>
<td>16</td>
<td>10</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:17 AM</td>
<td>25a</td>
<td>24</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:18 AM</td>
<td>25a</td>
<td>24</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:19 AM</td>
<td>25a</td>
<td>24</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:20 AM</td>
<td>25a</td>
<td>24</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:21 AM</td>
<td>25a</td>
<td>33</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:22 AM</td>
<td>25a</td>
<td>45</td>
<td>11</td>
<td>11</td>
<td>12</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:23 AM</td>
<td>25a</td>
<td>24</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:24 AM</td>
<td>25a</td>
<td>36</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:25 AM</td>
<td>25a</td>
<td>27</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:26 AM</td>
<td>25a</td>
<td>24</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:27 AM</td>
<td>25a</td>
<td>23</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:28 AM</td>
<td>25a</td>
<td>34</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:29 AM</td>
<td>25a</td>
<td>24</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:30 AM</td>
<td>25a</td>
<td>23</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6/23/2014 00:31 AM</td>
<td>25a</td>
<td>34</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 48:** Signal log from intersection of Virginia St. and McCarran Blvd, Reno, NV

### 6.2. METHODOLOGY

The proposed methodology of estimating intersection turning volumes from traffic signal information is depicted in Figure 49. To produce the required data, a simulation must be performed in VISSIM. The reason for choosing this software is its ability to generate high-resolution outputs that are required for this method. In this simulation, different turning volumes should be entered for each signal configuration parameter. To change turning
volumes, a code can be developed in a COM interface. With this code, the inputs do not have to be changed manually.

Table 9: Signal configuration of intersection at Virginia St. and McCarran Blvd, Reno, NV

<table>
<thead>
<tr>
<th>ID: 164</th>
<th>Name: VIRGINIA &amp; MCCARRAN NORTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration: Standard</td>
<td></td>
</tr>
<tr>
<td>Param</td>
<td>Phs 1</td>
</tr>
<tr>
<td>Walk</td>
<td>0</td>
</tr>
<tr>
<td>Ped Clearance</td>
<td>0</td>
</tr>
<tr>
<td>Min Green</td>
<td>6</td>
</tr>
<tr>
<td>Passage</td>
<td>2.5</td>
</tr>
<tr>
<td>Max1</td>
<td>38</td>
</tr>
<tr>
<td>Max2</td>
<td>23</td>
</tr>
<tr>
<td>Yellow</td>
<td>3</td>
</tr>
<tr>
<td>Red</td>
<td>1</td>
</tr>
<tr>
<td>Red Revert</td>
<td>4</td>
</tr>
<tr>
<td>Added Initial</td>
<td>0</td>
</tr>
<tr>
<td>Max Initial</td>
<td>4</td>
</tr>
<tr>
<td>Time Before Reduce</td>
<td>4</td>
</tr>
<tr>
<td>Cars Before Reduce</td>
<td>0</td>
</tr>
<tr>
<td>Time To Reduce</td>
<td>21</td>
</tr>
<tr>
<td>Reduce By</td>
<td>0</td>
</tr>
<tr>
<td>Min Gap</td>
<td>1.5</td>
</tr>
<tr>
<td>Dynamic Max Limit</td>
<td>48</td>
</tr>
<tr>
<td>Dynamic Max Step</td>
<td>10</td>
</tr>
<tr>
<td>Startup</td>
<td>RED</td>
</tr>
<tr>
<td>Enable</td>
<td>On</td>
</tr>
<tr>
<td>Auto Entry</td>
<td>Off</td>
</tr>
<tr>
<td>Auto Exit</td>
<td>Off</td>
</tr>
<tr>
<td>Non Act1</td>
<td>Off</td>
</tr>
<tr>
<td>Non Act2</td>
<td>Off</td>
</tr>
<tr>
<td>Lock Call</td>
<td>Off</td>
</tr>
<tr>
<td>Min Recall</td>
<td>Off</td>
</tr>
<tr>
<td>Max Recall</td>
<td>Off</td>
</tr>
<tr>
<td>Ped Recall</td>
<td>Off</td>
</tr>
<tr>
<td>Soft Recall</td>
<td>Off</td>
</tr>
<tr>
<td>Dual Entry</td>
<td>Off</td>
</tr>
<tr>
<td>Sim Gap Enable</td>
<td>Off</td>
</tr>
<tr>
<td>Guar Passage</td>
<td>Off</td>
</tr>
<tr>
<td>Rest In Walk</td>
<td>Off</td>
</tr>
<tr>
<td>Cond Service</td>
<td>Off</td>
</tr>
<tr>
<td>Add Init Calc</td>
<td>Off</td>
</tr>
<tr>
<td>Ring</td>
<td>1</td>
</tr>
</tbody>
</table>

A sample of VISSIM output is shown in Figure 50. In this output for each phase, one column shows the state of signal (green by |, yellow by \, and red by a dot). Other columns show the state of detectors (occupied by ? and otherwise by a dot). Then, for each phase,
all green times and their corresponding volumes can be extracted. Each row in this data set includes the traffic volume passing by during green time (gv), the cycle length (cl), the green time (gt), the minimum green (mg), the vehicle extension (ve), the min recall (discrete variable with yes or no as values), the max recall (discrete variable with yes or no as values), and the side street traffic volume (sv). Because the side street hourly volume (sv) is unknown in reality, this variable should be replaced with the time of day or different time intervals regarding traffic condition (i.e. night, off-peak, and peak). Table 10 shows a sample of the prepared data set. In this table, some variables are removed because they are the same for all data sets. For example, minimum green (mg) is not usually necessary because it does not change during different times. To prepare these data sets, several codes were developed in MATLAB to extract the required information from the VISSIM output (see Appendix E). The next step was to develop a model for each phase/movement. For these models, the green time volume (gv) was selected as a dependent variable while the other parameters were defined as independent variables. Afterwards, a prediction model was developed for each phase/movement. Two methods were adopted to build the models: regression and Adaptive Neural Fuzzy Inference System (ANFIS). ANFIS is a class of adaptive networks that is functionally equivalent to a fuzzy inference system; however, in ANFIS, the user does not need to define the rules. Rules are generated using an artificial neural system. This feature makes the ANFIS a very powerful tool to explain the relationship of variables. In this study, the built in ANFIS function of MATLAB was used. The following section explains the ANFIS approach briefly. A detailed description and discussion can be found in references [41] and [43].
Dash box shows model development for movement $i$. It should be repeated for all movements.

Collecting manual counts at the same time
Calculating Mean Absolute Percent Error (MAPE) of estimated counts

Use the model for future count estimations

Getting signal configuration

This method is not appropriate for this movement

Is The results satisfactory?

No

Making prediction model for movement $i$

Estimating turning volumes

Collecting manual counts at the same time

Calculating Mean Absolute Percent Error (MAPE) of estimated counts

Getting signal log from central computer

Developing a code in COM interface (VISSIM tool) with different parameters

Modeling the intersection in VISSIM

Figure 49: Methodology of estimating intersection turning volumes from traffic signal information

Figure 50: Sample of VISSIM output for phase 2 and 8
6.2.1. ANFIS

ANFIS combines the Fuzzy Inference System (FIS) and Artificial Neural Networks (ANN) where the FIS is used to model relationships between non-linear variables and ANN is used to optimize input and output membership function parameters. FIS can be defined as a process of mapping from a given input to an output using the theory of fuzzy sets and ANN is an artificial neural network that consists of a number of very simple and highly interconnected processors, also called neurons. The neurons are connected by weighted links passing signals from one neuron to another. ANN adjusts the weights to bring the network input/output behavior into line with that of the training data. There are two well-known fuzzy inference system: Mamdani-style inference and Sugeno-style inference [41]. The Sugeno fuzzy model was used for a systematic approach to generating fuzzy rules from a given input-output data set. A typical Sugeno fuzzy rule can be expressed in the following form:

\[
\text{IF} \quad \text{Green Time} \quad \text{is} \quad \text{Medium} \\
\text{AND} \quad \text{Minor (Intersecting) Street Volume (Time Intervals)} \quad \text{is} \quad \text{Peak-hour} \\
\text{THEN} \quad \text{Green Time Volume} \quad \text{is} \quad \text{High}
\]
The ANFIS adopted in this chapter is represented by a six-layer feedforward neural network [41]. Figure 51 shows the ANFIS architecture that corresponds to the first-order Sugeno fuzzy model.

Layer 1 is the input layer. Neurons in this layer simply pass external crisp signals to Layer 2. That is [41],

\[ y_i^{(1)} = x_i^{(1)}, \]  

(Equation 23)

where \( x_i^{(1)} \) is the input and \( y_i^{(1)} \) is the output of input neuron \( i \) in Layer 1. Layer 2 is the fuzzification layer. Neurons in this layer perform fuzzification. For sake of simplicity in the diagram, Figure 51 shows only two fuzzy members for each variable. For example, two fuzzy members of variable Green Time (\( x_1 \)) can be defined as Low (A1) and High (A2). However, the actual members for both variables are more than two. In this chapter, for fuzzification neurons, the bell-activation function and the trapezoid-activation function were tested.

A bell activation function, which has a regular bell shape, was specified as [41]:

\[ y_i^{(2)} = \frac{1}{1 + \left( \frac{x_i^{(2)} - a_i}{c_i} \right)^2 b_i}, \]  

(Equation 24)

where \( x_i^{(2)} \) is the input and \( y_i^{(2)} \) is the output of neuron \( i \) in Layer 2; and \( a_i, b_i \) and \( c_i \) are parameters that control the center, width and slope, respectively, of the bell activation function of neuron \( i \). The trapezoid activation function was specified by its four corners.
Layer 3 is the rule layer. Each neuron in this layer corresponds to a single Sugeno-type fuzzy rule. A rule neuron receives inputs from the respective fuzzification neurons and calculates the firing strength of the rule it represents. In an ANFIS, the conjunction of the rule antecedents is evaluated by the operator product. Thus, the output of neuron $i$ in Layer 3 was obtained as follows [41]:

$$y_i^{(3)} = \prod_{j=1}^{k} x_{j|i}^{(3)},$$

(Equation 25)

where $x_{j|i}^{(3)}$ are the inputs and $y_i^{(3)}$ is the output of rule neuron $i$ in Layer 3.

For example, in Figure 51,

$$y_{11}^{(3)} = \mu_{A1} + \mu_{B1} = \mu_1,$$

(Equation 26)

where the value of $\mu_1$ represents the firing strength, or the truth value, of Rule 1.

Layer 4 is the normalization layer. Each neuron in this layer receives inputs from all neurons in the rule layer and calculates the normalized firing strength of a given rule. The normalized firing strength is the ratio of the firing strength of a given rule to the sum of firing strengths of all rules. It represents the contribution of a given rule to the final result.

Thus, the output of neuron $i$ in Layer 4 was determined as follows [41]:

$$y_i^{(4)} = \frac{x_i^{(4)}}{\sum_{j=1}^{n} x_j^{(4)}} = \frac{\mu_i}{\sum_{j=1}^{n} \mu_j} = \bar{\mu}_i,$$

(Equation 27)
where $x_{ji}^{(4)}$ is the input from neuron $j$ located in Layer 3 to neuron $i$ in Layer 4, and $N$ is the total number of rule neurons. For example,

$$y_{N1}^{(4)} = \frac{\mu_1}{\mu_1 + \mu_2 + \mu_3 + \mu_4} = \bar{\mu}_1$$  \hspace{1cm} \text{(Equation 28)}$$

Layer 5 is the defuzzification layer. Each neuron in this layer was connected to the respective normalization neuron and also received initial inputs, $x_1$ and $x_2$. A defuzzification neuron calculates the weighted consequent value of a given rule as follows [41]:

$$y_i^{(5)} = x_i^{(5)} [k_{i0} + k_{i1}x_1 + k_{i2}x_2] = \bar{\mu}_i [k_{i0} + k_{i1}x_1 + k_{i2}x_2],$$  \hspace{1cm} \text{(Equation 29)}$$

where $x_i^{(5)}$ is the input and $y_i^{(5)}$ is the output of defuzzification neuron $i$ in Layer 5, and $k_{i0}$, $k_{i1}$, and $k_{i2}$ is a set of consequent parameters of rule $i$.

Layer 6 is represented by a single summation neuron. This neuron calculated the sum of outputs of all defuzzification neurons and produced the overall ANFIS output, $y$ [41],

$$y = \sum_{i=1}^{n} x_i^{(6)} = \sum_{i=1}^{n} \bar{\mu}_i [k_{i0} + k_{i1}x_1 + k_{i2}x_2]$$  \hspace{1cm} \text{(Equation 30)}$$

It is often difficult or even impossible to specify a rule consequence in a polynomial form. Conveniently, it is not necessary to have any prior knowledge of rule consequent parameters for an ANFIS to deal with a problem. An ANFIS learns these parameters and tunes membership functions.
6.2.2. Count Estimation Procedure

After developing the models for each phase/turning movement, green time of signal logs (Figure 48) was used as the input for the models. For each green time, models estimated a volume. Then these volumes were summed to produce 15-minute or hourly counts.

To verify the models for each phase, real turning volumes were compared with model outputs. The detector accuracy was defined in terms of Mean Absolute Percent Error (MAPE, %). Then, if MAPEs are satisfactory, the models can be used to predict future turning movements.

6.3. CASE STUDIES

The intersection at E. 2nd St. (east-west, as major street) and Kirman Ave (north-south, as minor street), (Figure 52) and the intersection at McCarran Blvd (east-west) and N. Virginia St. (north-south) in Reno, NV (Figure 53) were selected for case studies. The first intersection represented a major-minor intersection and the second one represented a major-major intersection. Figure 54 shows scatterplots of Green Time per Cycle and
Volume per Cycle at E. 2nd St. during different times. Twenty-four hours were categorized into five different time intervals from Very Low Volume, which refers to midnight hours, to Very High Volume, which refers to peak hours. This figure shows that during off-peak hours, there was not a high correlation between actuated green time and volumes. This is because the signal continues in green time until the maximum green and a call from the side street. Close to peak hours, the flow rate becomes closer to saturation flow rate and green time shows more correlation with volume. During peak hours, there are enough calls from the side street to terminate green after gap out on the major street. Therefore, in almost all cycles, a certain number of vehicles passed through the intersection within a given green time before gap out happened. In the side street, because green terminated after gap out or maximum green, there was high association between green time and volume at all times.

Similar scatterplots were generated for the intersection of McCarran Blvd. and N. Virginia St. However, in this intersection, both streets were major streets and therefore, it was only during peak hours or close to peak hours that green time showed only correlation with volume. The next section describes the results of applying the proposed method at these intersections.
Figure 52: Intersection of E 2nd St (east-west) and Kirman Ave (north-south), Reno, NV

Figure 53: Intersection of McCarran Blvd (east-west) and N Virginia St (north-south)
Figure 54: Scatterplots of Green Time per Cycle and Volume per Cycle in E 2nd St. during different times
6.4. RESULTS

Table 11 demonstrates a sample of fuzzy sets of variable Green Time for the intersection at McCarran Blvd. and Virginia St. during peak hours. Intersecting Street Volume did not significantly improve the results. Therefore, for sake of simplicity, it was omitted from the modeling process. There are applications that can be used to facilitate the usage of ANFIS models. One of them is the anfisedit Graphical User Interface (GUI) in MATLAB. For each approach, all information of fuzzy sets were entered into the anfisedit GUI. Both bell-shaped and trapezoid membership functions were tested for approaches. The bell-shaped membership functions, despite of their complexity, did not significantly improve the models compared to those using the trapezoid membership function. In Table 11, all phases had four members in their fuzzy sets. Having four members meant that each variable was categorized into four categories that were: Very Low, Low, Medium, and High. The numbers inside the brackets show the four corners of trapezoid members. For example, member Medium in Phase 1 was defined by 17.3, 23.46, 27.55, and 30.72. This meant that the membership of volumes lower than 17.3 and larger than 30.72 were assigned a value of zero in this category, and assigned a value of one from 23.46 to 27.55. Other volume ranges had a membership between zero and one. However, users did not need to engage in this calculation since anfisedit GUI generated all of the output results.
Table 11: Sample of fuzzy sets of intersection at McCarran Blvd and Virginia St during peak hours

<table>
<thead>
<tr>
<th>Membership Functions</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phase 1</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[3.747 6.92 12.61 16.84]</td>
</tr>
<tr>
<td></td>
<td>[10.14 15.05 20.6 24.6]</td>
</tr>
<tr>
<td></td>
<td>[17.3 23.46 27.55 30.72]</td>
</tr>
<tr>
<td></td>
<td>[27.59 30.72 35.48 38.65]</td>
</tr>
<tr>
<td><strong>Phase 2</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[10.21 15.21 21.19 25.2]</td>
</tr>
<tr>
<td></td>
<td>[17.3 23.07 29.39 33.71]</td>
</tr>
<tr>
<td></td>
<td>[29.4 32.1 35.48 46.53]</td>
</tr>
<tr>
<td><strong>Phase 3</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[2.91 6.39 12.3 18.04]</td>
</tr>
<tr>
<td></td>
<td>[9.1468 20.96 26.5]</td>
</tr>
<tr>
<td></td>
<td>[18.35 24.7 29.5 33.26]</td>
</tr>
<tr>
<td></td>
<td>[26.77 32.64 37.71 41.19]</td>
</tr>
<tr>
<td><strong>Phase 4</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[12.03 19.27 30.2 36.7]</td>
</tr>
<tr>
<td></td>
<td>[30.13 37.43 48.23 55.46]</td>
</tr>
<tr>
<td></td>
<td>[48.23 55.47 66.33 73.57]</td>
</tr>
<tr>
<td></td>
<td>[66.33 73.57 84.43 91.67]</td>
</tr>
<tr>
<td><strong>Phase 5</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[6.503 7.93 10.08 11.43]</td>
</tr>
<tr>
<td></td>
<td>[8.999 11.39 13.98 15.27]</td>
</tr>
<tr>
<td></td>
<td>[12.57 14.88 17.22 18.66]</td>
</tr>
<tr>
<td></td>
<td>[17.2 18.64 20.77 22.2]</td>
</tr>
<tr>
<td><strong>Phase 6</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[6.117 12.05 19.42 29.18]</td>
</tr>
<tr>
<td></td>
<td>[17.78 24.66 38.34 44.7]</td>
</tr>
<tr>
<td></td>
<td>[35.44 43.87 50.36 56.4]</td>
</tr>
<tr>
<td></td>
<td>[50.21 56.3 65.45 71.38]</td>
</tr>
<tr>
<td><strong>Phase 7</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[4.45 7.05 11.11 15.34]</td>
</tr>
<tr>
<td></td>
<td>[9.001 13.4 17.49 20.99]</td>
</tr>
<tr>
<td></td>
<td>[15.88 19.93 23.87 26.57]</td>
</tr>
<tr>
<td></td>
<td>[23.71 26.48 30.45 33.05]</td>
</tr>
<tr>
<td><strong>Phase 8</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[8.81 17.89 29.32 40.38]</td>
</tr>
<tr>
<td></td>
<td>[29.81 37.8 54.2 63.32]</td>
</tr>
<tr>
<td></td>
<td>[54.09 63.28 76.91 85.99]</td>
</tr>
<tr>
<td></td>
<td>[76.91 85.99 99.61 108.7]</td>
</tr>
</tbody>
</table>

Figure 55 shows the accuracy of applying the proposed method to the intersection at E. 2nd St. and Kirman Ave. in Reno, NV. The first diagram in this figure illustrates the Mean Absolute Percentage Error of regression and ANFIS for both the training and test
data sets during different conditions. The training data was a data set from which models were developed and the test data was used to validate the models. The dashed line demonstrates the regression results and the bold line shows the ANFIS results. In almost all conditions, ANFIS produced better results. Two extreme conditions were at major streets during Low Volume hours. While regression produced 53.8 and 55.7 percent MAPE for training and test data sets, respectively, ANFIS MAPEs were 7.6 and 7.4 percent. This showed that when there are enough training sets, ANFIS was able to learn the hidden patterns of data and generated much better models compared to regression. As it was expected, during peak hours, errors were lower than other hours and decreased to less than 15 percent. The second diagram of Figure 55 shows ANFIS improvements over regression. As it can be seen, ANFIS produced results which were up to 48 percent better than the regression results.

Figure 56 contains similar diagrams for the intersection at McCarran Blvd. and N. Virginia St. in Reno, NV. Here, models were built for eight phases. Time intervals (i.e., intersecting street volume replacement) were categorized into Night, Off-peak, and Peak hours. Because both streets are major streets, during Night hours errors were extremely high. This was because green times were not based on volume. However, by increasing the volume during off-peak and peak hours, the accuracy of models also increased. ANFIS produced the following MAPE for phase 1 to 8 during peak hours: 10, 19, 9, 20, 7, 22, 8, and 25 percent. Therefore, phases 1, 3, 5, and 7 had errors less than 10 percent while phases 2, 4, 6, and 8 had errors close to 20 percent. This suggests that all left turn phases had almost half the error compared to the through phases. The reason for this was the fact that
left-turn green times were based on gap out. That means they were highly related to volume. The bar diagram within Figure 56 also shows that the improvement of ANFIS over regression can be more than 25 percent. ANFIS improvement was more significant during the night and off peak hours. The reason was that during these hours, there were more irregularities in the data sets and ANFIS was able to learn and consider them.
6.5. SUMMARY AND CONCLUSION

Current detectors that are spliced together generally produce unreliable counts of turning volumes. In this study, a method was proposed to estimate turning volumes from signal...
information without using detector data. In this method, a simulation model was developed in VISSIM with different volume inputs. Then, based on this simulation a data set was generated which contained green times in each cycle during the simulation period and their corresponding volumes. A model was developed for each phase/turning movement based on this data set and if errors from these models were acceptable, they could be used for future count estimations. For modeling, regression and the Adaptive Neural Fuzzy Inference System (ANFIS) were used. Results showed that, during peak hours, there was a high correlation between actuated green time and volumes at the major street. The minor street green terminated either after gap out or maximum green. Therefore, it was feasible to estimate volume from prediction models at all times. From the results, it was also concluded that when there were enough records for modeling, ANFIS produced more accurate models compared to regression. MATLAB has a built-in toolbox for ANFIS that facilitated utilization of this powerful modeling method.

The method proposed in this chapter does not require extensive data collection and due to VISSIM's detailed outputs and capabilities, it is relatively easy to employ. Also, there is no need to install new equipment or change and modify existing facilities.
7. MID-INTERSECTION DETECTOR (MID)

Due to many advantages, loop detectors are the most common practice for obtaining data to control intersections. However, they have some drawbacks, including the fact that multiple detectors are usually required to monitor a location. The current practice in many cities is to install four consecutive loop detectors per lane, or two at the stop bar and one as an advanced detector. In some cities, there are also departure detectors. All these configurations have some practical problems and do not produce accurate counts. In this chapter, a new placement configuration for departure detectors is proposed and named the mid-intersection detector (MID). In this configuration, departure detectors are moved back to the middle of the intersection in such a way that they can be activated by more than one movement at different times. In some cases, departure detectors lack equations for calculating turning movements, a problem solved by MIDs since each movement passes more detectors along its path (without increasing the number of required loops). MIDs can produce more accurate and reliable data for obtaining turning movement counts.

7.1. PROPOSED MID-INTERSECTION DETECTORS

This chapter recommends the new concept of a mid-intersection detector (MID). With this placement configuration, the number of equations is more than the number of turning movements. In this method, departure loops are moved back to the middle of the intersection (Figures 57 to 61), in such a way that each MID can capture more than one movement. For example, in Figure 57, detector M11 can capture southbound (SB) movements, westbound through (WBT) and northbound left (NBL). Therefore, in this
example, while departure detectors generate only one equation, the equivalent MID can generate three equations.

Due to intersection operation, stop bar detectors are still required, but compared to other detector configurations, MIDs may produce more reliable and accurate data with fewer detectors even with a non-split scheme.

Extracting counts from detectors is based on sequences of detector activations. For example, in Figure 57, westbound right (WBR) activates only detector $S_{11}$ (depending on the intersection geometry, it can also activate $M_{41}$); westbound through (WBT) first activates detector $S_{11}$, then $M_{41}$ and then $M_{11}$, and westbound left (WBL) first activates detector $S_{11}$, then $M_{41}$ and then $M_{21}$. These sequences of detector activations are unique and as a result, all turning movements can be obtained.

![Figure 57: Mid-Intersection Detectors concept, Case 1.](image-url)
At time $t$ that $S_{11}$ has been activated, the following function can be applied for westbound (WB) movements:

$$S_{11}^t = \begin{cases} 
WBL & \text{if } \left(M_{41}^{t+\Delta t_{S_{11}-M_{41}} \pm \delta}\right) \text{ and } \left(M_{21}^{t+\Delta t_{S_{11}-M_{21}} \pm \delta}\right) \\
WBT & \text{if } \left(M_{41}^{t+\Delta t_{S_{11}-M_{41}} \pm \delta}\right) \text{ and } \left(M_{11}^{t+\Delta t_{S_{11}-M_{11}} \pm \delta}\right) \\
WBR & \text{if } \left(M_{41}^{t+\Delta t_{S_{11}-M_{41}} \pm \delta}\right)^* 
\end{cases}$$

Equation (31) denotes that if there is activation at time $t$ at the stop bar, then at time $t + \Delta t_{S_{11}-M_{41}} \pm \delta$ detector $M_{41}$ is activated. Finally, at time $t + \Delta t_{S_{11}-M_{21}} \pm \delta$ detector $M_{21}$ is activated. Therefore, this movement should be added to WBL counts.

The variable $\delta$ indicates a time range that is possible for vehicles at a certain movement to activate a detector. For example, if the left turns are not protected, then westbound left turns before activating $M_{21}$ should yield to eastbound through (EBT), which may take several seconds (up to green time of the phase). As a result, if $M_{21}$ is activated after $t + \Delta t_{S_{11}-M_{21}} \pm \delta$ seconds, it could not be WBL and may be, for example, southbound through (SBT). If the movement is protected, then $\delta$ defines the time variations that can be estimated based on speed variations. For example, if the distance between $S_{11}$ and $M_{21}$ is 50 feet and speed range is 20 mph to 40 mph, then vehicles reach $M_{21}$ from $S_{11}$ in 0.85 to 1.7 seconds; therefore $\delta$ is $\pm 0.85$ seconds, if WBL is protected.
For WBT, a sequence of detectors must be $S_{11}$, then $M_{41}$, and then $M_{11}$. Due to channelizing, WBR can be only $S_{11}$, or $S_{11}$ then $M_{41}$, so the detector $M_{41}$ is shown with an asterisk in Equation 31, which indicates that even without this activation, it is possible for a vehicle to turn right. The $\Delta t$s are a function of speed and distance and can be calculated as follows:

\[
\Delta t_{S_{11}-M_{41}} = \frac{x_{S_{11}-M_{41}}}{V} \times 0.68 \quad \text{Equation (32)}
\]

\[
\Delta t_{S_{11}-M_{11}} = \frac{x_{S_{11}}-M_{11}}{V} \times 0.68 \quad \text{Equation (33)}
\]

\[
\Delta t_{S_{11}-M_{21}} = \frac{x_{S_{11}}-M_{21}}{V} \times 0.68 \quad \text{Equation (34)}
\]

where:

$x_{S-M}$ : Distance between detector $S$ to detector $M$ (ft)

$\Delta t_{S-M}$ : Travel time from detector $S$ to detector $M$ (sec)

$V$ : Average speed from detector $S$ to detector $M$ (mph)

Figures 63 to 66 show four other intersections. Based on the geometry of each intersection, different numbers of mid-intersection detectors can be activated by a movement. For example, WBL and SBT in intersection of Figure 57 activate two MIDs, while in Figure 58, they activate one and three MIDs, respectively.
Figure 58: Mid-Intersection Detectors concept, Case 2.

Figure 59: Mid-Intersection Detectors concept, Case 3.
Figure 60: Mid-Intersection Detectors concept, Case 4.

Figure 61: Mid-Intersection Detectors concept, Case 5.
The flowchart depicted in Figure 62 shows the process of obtaining counts from MID. This process is based on functions similar to Equation 31. In other words, it extracts counts from the sequences of activations. The simplest case is when all left turns are protected. However, even when movements are permitted, still the correct movements can be extracted based on the sequence of activations. For example, in Figure 57, both WBL and EBL activate M_{21} and M_{41}. However, WBL first activates M_{41} and then M_{21} while EBL first activates M_{21} and then M_{41}.

This process should be repeated for all approaches and for all S_{ij} (stop bar detectors) activations.

There is a possibility, however, that some vehicles do not pass or do not activate all detectors (stop bar detectors and MID) along their path because of detection failure. At these situations, due to a lot of detectors and different activation sequences, most of the movements can be estimated based on the remaining detector activations. The flowchart depicted in Figure 68 (failed activation algorithm) shows this process. After the process of obtaining data from the flowchart in Figure 62, all activations that are not related to S_{ij} activations will be transferred to the failed activation algorithm (Figure 63). In this flowchart, based on the sequence of detector activations, an appropriate movement would be assigned to a sequence of activations. If no logical sequence can be extracted, then one count would be divided between all movements that may have activated the detectors based on the proportions of previous counts. For example, if one activation of M_{41} has remained undetermined, then one count can be shared between WBT, SBT, and EBL based on their previous counts. Suppose up to now the total number of counts for these movements are
300, 600, and 100, respectively, then 0.3, 0.6, and 0.1 would be added to counts of these movements, respectively.

Starting from the time that the first S detector is activated,

\[ k = k + \Delta t \]

Constructing the \( S_{ij}^k \) functions of MID activations in the range of valid interval

Determine the turning movement

Adding 1 to the turning counts of approach \( i \)

Deleting the activated records of selected movement

Are all \( S_{ij} \) records used?

Yes

Are all MID activations used?

No

Using Failed Activation Algorithm

No

Yes

Counts

Figure 62: Obtaining counts from mid-intersection detectors.
Figure 63: Failed activation algorithm.
7.2. SIMULATION AND CASE STUDY

Many modern controllers are capable of storing high resolution data from detectors. This high-resolution data includes time of detector activations and the state of the signal at that time. Figure 64 shows a sample of high-resolution data obtained from a controller. However, because the MID concept has not yet been implemented, it is not possible to analyze it without simulation.

A simulation was performed for the intersection at Oddie Blvd and Sullivan Ln in Reno, Nevada (Figure 65) in VISSIM (a simulation software by PTV Group). VISSIM does not have the ability to place MIDs because in VISSIM, each detector should be defined for one movement while MIDs cover several movements. To resolve this problem for each of the MID movements, a separate detector was placed and then all movements at each location were combined. The simulation runs under ideal situations and cannot simulate different problems related to loop detectors. To make the simulation more realistic, some noises (errors) were introduced into the data randomly. These noises
represent a variety of problems such as the failure of a detector to detect any movement, double detecting of a single vehicle, and detectors that may be out of service. An algorithm was developed to introduce these noises into the data set. Each data set was produced with a different probability of noise. For example, a data set with a probability of five percent noise means five percent of activations were wrong. Table 12 shows a sample of this data set. To enter noises into simulation data, it is necessary to know how many activations were incorrect under normal operations at a real intersection, or in other words, what was the overall accuracy of detectors. For this purpose, another study was performed at an intersection in Reno, NV. Figure 66 shows the accuracy of detectors in this intersection in terms of Mean Absolute Percent Error (MAPE) (see Equation 35).

\[
MAPE(\%) = \frac{\sum_{i=1}^{n} \frac{|D_i - B_i|}{B_i}}{n}
\]

Equation (35)

where:

MAPE: Mean Absolute Percentage Error

D_i: the detector data value

B_i: the reference (base) data value

n: the total number of intervals

As it is evident in Figure 66, the stop bar detector counts were unreliable since there were MAPEs more than 35 percent. The reason behind these high errors was the way that detectors were wired to the controller. Loops were spliced together and then a single wire goes to the controller. This structure is adequate for intersection control but reduces the accuracy of counts significantly. Based on studies performed on the accuracy of actual
detectors, noises of up to 40 percent were entered for simulation stop bar detectors. However, MIDs are capable of producing a much lower error rate. The ITE report [15] concluded that if loop detectors are placed and wired properly, their counts would be “Excellent”, which means that counts tend to have an error of less than five percent. Some other common practices have “Good” and “Fair” accuracy, which mean 10 and 25 percent error, respectively. Based on these numbers, noises with a probability of five, 10, and 25 percent were entered into the MIDs data set. Then, data extraction algorithms were applied to these data sets to evaluate the ability of MIDs under different conditions.

Table 12: A sample of VISSIM output for phases 4.

<table>
<thead>
<tr>
<th>SG4</th>
<th>S41</th>
<th>M41</th>
<th>M42</th>
<th>M43</th>
<th>S42</th>
<th>M22</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* This column refers to state of signal, “ |” means green, “/” means yellow and point means red

** These columns refer to state of stop detectors, “ ?” means activation, and point means no activation

*** These columns refer to state of MIDs, “ ?” means activation, and point means no activation

To compare MIDs with departure detector configurations, another case was defined for the same intersection. In this case, instead of MIDs, departure detectors were placed at the beginning of exit lanes. Then, the same levels of noise were also considered for this case. The same process was applied for Cases 1 to 5 (Figures 57 to 61). The next section compares and analyzes the results of MIDs accuracy with departure detectors.
7.3. RESULTS

Figure 67 summarizes the results of estimating counts from MID placement configuration and its equivalent departure detector configuration. When all detectors were perfect, the two placement configurations produced almost 100 percent accuracy. However, when detectors perform similarly to actual detectors in the field (i.e., with errors including failure
to detect or inaccurate counts because of splicing the stop bar detectors), the MID configuration produced better counts. The reason for better accuracy of MID configuration was due to more detectors for each movement. As a result, if one detector failed to detect a vehicle, there was still a good chance that another detector sensed it.

Note that Figure 67 shows only the improvements of MIDs over departure detectors regarding the second problem (i.e., when departure detectors fail to detect). There are some cases where departure detectors were not able to produce enough equations and, therefore, there was not a straightforward method for estimation of the counts. This figure did not reflect MID’s improvements for this problem.

In Figure 67, each diagram shows 12 different conditions regarding the stop bar and MID noises. For example, in Case 1, when the stop bar and MID noises were both five percent, estimated turning volume error was almost zero for both MID and departure detector configurations. On the other end of this diagram, when MID and stop bar detector noises were 25 and 40 percent, respectively, the errors of estimated turning volume of MIDs and departure detectors were 2.5 and 10 percent, respectively. Figure 67 shows that especially when detectors are not accurate, MID configuration can have up to four times fewer errors compared to departure detectors. Note also, as it was stated previously, that it was only one of MID’s advantages and this simulation did not reflect other advantages of MIDs including giving more equations for unknown movements, especially when there are shared lanes or permitted left turns. Another advantage of MIDs compared to departure detectors was when permitted left turns trap in the middle of the intersection after the
termination of green time. Using MIDs, these vehicles can be detected and safely exit the intersection before the end of their green time.

7.4. SUMMARY AND CONCLUSION

Loop detectors are the most common devices for obtaining traffic data at intersections. In spite of many advantages, they have some drawbacks, including the fact that multiple detectors are usually required to monitor a location. The typical practice in many cities is to install four consecutive loop detectors per lane, or two at the stop bar and one as an advanced detector. In some cities, there are also departure detectors. Each of these configurations, has some practical problems and are not able to produce reliable turning counts. In this chapter, a new placement configuration was proposed and named as the mid-intersection detector (MID). Mid-intersection detectors can produce more accurate and reliable data for generating intersection Origin-Destination counts because they activate more detectors per movement. For a departure detector configuration, there is only one activation per movement in addition to the stop bar detector activations. When using MIDs, it is possible to increase the number of detector activations per movement up to the number of perpendicular lanes plus one. By increasing the number of activations per movement: 1) the number of equations would be more than the number of unknown movements; 2) if some of the detectors fail to record a vehicle, there is the possibility that the vehicle be recorded by other detectors.
Figure 67: Count errors at MID and departure detectors configurations.

Also, MIDs can solve another problem when left turns are permitted. Drivers that pass the stop line beyond a detection zone and wait for a gap in the opposing traffic may be left undetected if a gap does not occur or a vehicle ahead prevents the turn. In this case, the controller may skip the turn arrow in the next cycle because the vehicle is positioned ahead of the sensor's detection zone. Some agencies extend the loop beyond the stop line.
to prevent this situation; however, it may interfere with other movements. These left turn movements can be detected using MID.
8. CONCLUSIONS AND RECOMMENDATIONS

Loop detectors are the most common detectors used for obtaining data at intersections. In spite of many advantages, they have some drawbacks, including the fact that multiple detectors are usually required to accurately monitor a location. The current practice in many cities is to install four consecutive loop detectors per lane, or two at the stop bar and one as an advanced detector. In some cities, there are also departure detectors. Each of these configurations, in addition to some practical problems in installation and maintenance, typically result in unreliable counts.

This research studied the accuracy of detectors in terms of producing counts. Results showed that the counts were not reliable and had very high errors. In some cases detector counts could be modified. In addition to modifying detector counts, it is feasible to obtain turning movement counts from only traffic signal information. This research also has provided three methods to obtain turning movement counts in shared lanes. All proposed methods for obtaining turning movement volumes are based on existing traffic signal devices. To obtain more accurate and reliable counts, in the following section, the authors recommend cost effective replacements of inductive loops. However, if loop detectors be placed in certain ways and be wired properly, they can also produce accurate counts. The last section of this report explains these layouts and configurations.

8.1. COST-EFFECTIVE REPLACEMENTS OF INDUCTIVE LOOPS

Until the last decade, the methods for collecting historical traffic data over a wide area were essentially limited to a mixture of fixed counting locations using inductive loop detectors,
road tube counts and manual counts. Each of these methods has limitations that make the collection of urban traffic data a significant challenge. Fixed counting locations with inductive loop detectors can provide a baseline for traffic data collection. However, problems regarding the wiring of loops, failed loops that are not replaced, and difficulties in extracting data from controllers often prevent effective data collection from loop detectors. Also, there are counts, such as turning movement counts, complex weaving section movements, and vehicle occupancy counts, where fixed inductive loop detectors typically cannot provide the data needed.

These problems have resulted in a number of new technologies being employed in devices for collecting traffic data in urban areas. These technologies are considered to be non-intrusive because they can be deployed without the need to close lanes to traffic or to expose staff to unsafe conditions.

Several “non-intrusive” detection systems are becoming more prominent, being viewed as cost-effective replacements of inductive loops [1, 16, 47].

South Lake City, UT is now using Wavetronix SmartSensors to collect and systematically store approach volume counts. UDOT has introduced a site that uses upstream detectors (350-400 feet upstream of stopbar) to collect data and shows approach volume counts (Figure 68).

Some of these technologies are [47]:

1. Passive Infrared
Passive infrared devices detect the presence of vehicles by comparing the infrared energy naturally emanating from the road surface with the change in energy caused by the presence of a vehicle. Since the roadway may generate either more or less radiation than a vehicle depending on the season, the contrast in heat energy is what is detected.

2. Active Infrared

Active infrared devices detect the presence of vehicles by emitting a low-energy laser beam(s) at the road surface and measuring the time for the reflected signal to return to the device. The presence of a vehicle is measured by the corresponding reduction in time for the signal return.
3. Magnetic (Passive and Active)

Passive magnetic devices measure the change in the earth’s magnetic flux created when a vehicle passes through a detection zone. Active magnetic devices, such as inductive loops, apply a small electric current to a coil of wires and detect the change in inductance caused by the passage of a vehicle.

4. Microwave (Doppler, Radar and Passive Millimeter)
Doppler microwave devices transmit low-energy microwave radiation at a target area on the pavement and then analyze the signal reflected back to the detector. According to the Doppler principle, the motion of a vehicle in the detection zone causes a shift in the frequency of the reflected signal. This can be used to detect moving vehicles and to determine their speed. Radar devices use a pulsed, frequency-modulated or phase-modulated signal to determine the time delay of the return signal, thereby calculating the distance to the detected vehicle. Radar devices have the additional ability to sense the presence of stationary vehicles and to sense multiple zones through their range finding ability. A third type of microwave detector, passive millimeter, operates at a shorter wavelength than other microwave devices. It detects the electromagnetic energy in the millimeter radiation frequencies from all objects in the target area.

5. Passive Acoustic

Passive acoustic devices consist of an array of microphones aimed at the traffic stream. The devices are passive in that they are listening for the sound energy of passing vehicles.

6. Ultrasonic (Pulse and Doppler)

Pulse devices emit pulses of ultrasonic sound energy and measure the time for the signal to return to the device. Doppler devices emit a continuous ultrasonic signal and utilize the Doppler principle to measure the shift in the reflected signal.

7. Video

Video devices use a microprocessor to analyze the video image input from a video camera. Two basic analysis techniques are used: tripline and tracking. Tripline techniques monitor
specific zones on the video image to detect the presence of a vehicle. Video tracking techniques employ algorithms to identify and track vehicles as they pass through the field of view. The video devices use one or both of these techniques.

For a detailed cost of these technologies refer to Appendix A of [47].

The ITE Traffic Detector Field Manual reports that magnetometers “are better for counting than is an ILD (inductive loop detector).” Since magnetometers are passive devices without a radiated detection “field”, the magnetometer could distinguish between vehicles that arrive one foot apart [49].

8.2. CHANGING CONFIGURATION AND WIRING OF LOOPS

Based on a survey on several cities, ITE concluded that if the loops have certain configurations, they can produce reliable counts [15]. In the following section, these configurations are explained.

8.2.1. Single Loops

The loop type considered here is a single loop in presence mode. The size of the loop is not critical; however, most agencies used a 6 ft. loop. In general, 4 ft., 5 ft. and 6 ft. square loops will provide similar vehicle counting results. Agencies use one of three different types: square, circular and elongated diamond. While there is no significant difference in their count accuracy, it has been known that elongated diamonds (as compared to circles, squares and rotated squares) will tend to count two vehicles as a single vehicle when the two vehicles are closely spaced. Figure 58 shows several different applications of a single presence loop, which include:
1. Exit loops. Loops 6A, 6B, 6C are examples. These are generally 6 ft. by 6 ft., square or diamond, or 6 ft. diameter three-turn\textsuperscript{1} or four-turn loops. These loops will generally give excellent\textsuperscript{2} results as long as each loop is wired to an individual loop detector channel. These loops have good count accuracy across a range of vehicle types except for bicycles and motorcycles.

2. Queue loops. Loops 5D and OvA (overlap A) should provide excellent to good counts for the lanes in which they are located. It is important to locate these loops appropriately to improve count accuracy. For example, if Loop 5D is located prior to the point where vehicles change lanes, then the counts from this loop may not be accurate.

3. Advance loops. Loops 2A, 2B, 2C as well as 2D, 2E and 2F all provide excellent to good counts for either by lane or for the total approach (or mid-block).

4. STOP line loops: Loops 2J through 2L, loops 2G through 2I, or loop 5E should give excellent to good lane counts.

To get excellent to good counts from single loops, each loop should be wired to an individual loop detector channel. If two or more are spliced together into one loop detector channel, the count accuracy is expected to be fair to poor [15].

\textsuperscript{1} For calculating the minimum number of loop turns see [50], chapter 2.
\textsuperscript{2} The ratings given in the ITE report [15] are qualitative and based upon field experience and the knowledge of committee members and the industry and are defined as follow: 1) Excellent: Counts tend to have an error of less than 5 percent. 2) Good: Counts tend to have an error of less than 10 percent. 3) Fair: Counts tend to have an error less than 25 percent. 4) Poor: Counts tend to have an error greater than or equal to 25 percent.
8.2.2. Two Loops Connected in Series

Two loops in a series that have the same number of turns, are buried to the same depth, are separated by approximately 9 ft., and are sized such that the distance between the outside edges of those two loops is different than the typical spacing between vehicles, may be reasonably expected to provide good counts. Performance can be significantly enhanced.
when the loops are connected such that the field generated by the individual loop is additive
between the loops rather than subtractive. Counting results are likely to be fair to poor
when the loops are separated by 10 or more feet, have a different number of turns or are
connected in parallel. The most common spacing between these loops is 9 ft., although 8
ft. and 10 ft. are also used. Figure 59 shows various applications of two loops in series.

1. STOP line loops. Loops in a series separated by 9 ft. or less, such as 5A and 5B; 2G and
2H; and 2K and 2L, may be expected to give excellent to good counting performance as
long as both loops have the same number of turns. In the illustration, loops 2I and 2J are
spaced 12 ft. apart and may be expected to give fair to poor performance. Similarly, loop
OvA, shown in the illustration to be spaced 24 ft. apart, is expected to give poor lane count
performance.

2. Extension loops. Loop pairs 2A-2D, 2B-2E and 2C-2F connected in a series and spaced
50 ft. or further apart, are expected to give poor lane count performance. The important
point to remember when using two loops connected in a series for traffic counting is that
both loops should have an equal number of turns and the two loops should be spaced 9 ft.
apart.

This configuration will result in excellent to good count accuracy. If, however, the two
loops have a different number of turns, are connected in parallel or are spaced greater than
10 ft., then their count accuracy is expected to be poor [15].
8.2.3. Four Loops Connected in Series or Series/Parallel

Figure 71 shows multiple connection methods and multiple placement locations of four 6 ft. by 6 ft. loops hooked in a series or in series/parallel to emulate long loops. Counting
performance is significantly enhanced when the loops are connected such that the fields generated by the individual loops are additive between the loops rather than subtractive. Counting using these loop combinations assumes the use of loop detectors that either have long loop and/or multiple loop counting modes. Loops in combination 5A, 5B, 5C, 5D are connected in a series and are separated between 8 and 10 ft. Excellent to good counting results are expected.

Loops in combination 5E, 5F, 5G, 5H are connected in series/parallel and are separated by about 9 ft. Again, excellent to good counting results are expected.

Loops in combination 2A, 2B, 2C, 2D are connected in series/parallel. This combination is further connected in parallel with another series/parallel combination of 2F, 2G, 2H, 2I. This parallel combination of two series/parallel combinations is connected in series with a third series/parallel combination (2K, 2L, 2M, 2N). The loops in each of the three combinations are separated by 9 ft. Poor to fair counting performance is anticipated when this total loop combination is used for counting. Any time a loop combination detects the traffic in more than one lane, the ability of that loop’s combination to accurately count traffic decreases.

In general, if four regular 6 ft. by 6 ft. loops on a lane are connected in a series or series/parallel and are spaced equally at about 9 ft. apart, then the count accuracy should be excellent to good. If, however, the loops are not equally spaced, have a separation distance of greater than 10 ft. or are connected to a four-loop combination from another lane, then the count accuracy is expected to be fair to poor [15].
Figure 72 shows different wiring configurations for four sequential loops and their consequent expected accuracy.

Figure 71: Various applications of four loops [15]
Figure 72: Sequential four loop configuration and expected accuracy [15]
8.3. SUMMARY AND CONCLUSIONS

The automatic count method provides a means for obtaining large amounts of traffic data using permanent or portable counters. While obtaining more resources would be ideal, due to the reality of constrained resources, the primary objective of this dissertation was to investigate if reliable traffic volume information can be automatically collected using existing signal control devices.

A large majority of signalized intersections operate under some form of actuated control in that intersection approaches (or lanes) have some type of inductive loops. The new loop detector (also called loop amplifier) and signal controller equipment now provide the ability to collect traffic count information from the same loops on the intersection approaches that are used for actuated control. The potential to extract traffic counts from an existing signalized intersection loop detection system may provide the opportunity to collect data at minimal cost.

Figure 73 shows how to use methods provided in this research. First, Chapter 3 provides the process and the required equations to identify and investigate the accuracy of existing inductive loops to obtain automated turning movement volumes. When detector counts are not reliable, the method in Chapter 4 can be used to modify the detector counts.

One of the major challenges of obtaining turning volume counts from detectors is when there are shared lanes because in this case, additional detectors are needed to track the turns. Loops need to be located at strategic locations at the stop bar and downstream of the intersection. In cases where there are no downstream detectors, Chapter 5 proposes three methods to estimate turning movements from stop bar detectors only.
Chapter 6 introduces an alternative method which attempts to obtain turning movement counts from only signal information without using detector information.

Chapter 7 proposes a new layout for downstream detectors that is called mid-intersection detector (MID). In this method, departure loops are moved back to the middle of the intersection in such a way that each MID can capture more than one movement.

![Flowchart](image)

**Figure 73:** The process of using the methods provided in this dissertation
REFERENCES


[16] Middleton D., R. Parker, Vehicle Detector Evaluation. Texas Transportation Institute, Texas A&M University System. FHWA/TX-03/2119-1


APPENDIX A: SURVEY QUESTIONNAIRE
Survey Questionnaire

Center for Advanced Transportation Education and Research (CATER)

University of Nevada, Reno

Using Existing Loop Detectors at Signalized Intersections for Automated Volume Counts

The purpose of this project is to explore using existing loop detectors at signalized intersections for automated volume counts in Reno and Sparks. To assist CATER (Center for Advanced Transportation Education and Research) in determining current practices followed in Reno/Sparks, we are requesting that you complete this form and return it to the address below with any attachments. We thank you for your cooperation and input.

1. Total number of intersections controlled by your agency?

2. Number of intersections with detection type:

   Loops  Video  Other  None

3. On the major street, are loops placed in the:
   
   - Left turn lane
   - Through lane
   - Right turn lane

4. If there are loops on the side street, are loops placed in the:
   
   - Left turn lane
   - Through lane
   - Right turn lane

5. Do you typically provide separate loops in each lane of an approach?  Yes  No

6. Is the wiring separate for each loop back to the controller?  or spliced together?

7. If spliced together, is it by lane  by approach  by phase

8. Type of controllers existing in the field (check all that apply)
   
   - NEMA TS1
   - TS2
   - 170
   - 2070
   - Other

9. How many loops do you use per lane?
10. What is the size of loops and distance between them in each lane?

<table>
<thead>
<tr>
<th>Size</th>
<th>Distance between them in each lane</th>
</tr>
</thead>
</table>

11. Are counts collected at traffic signals from detectors on the approaches? Yes ☐ No ☐

   If yes, at what percent of intersections?
   If yes, how much is the accuracy of counts?
   If no, why not?

12. Does detector data go back to TMC or a central computer? Yes ☐ No ☐

   If yes, what is the name of software that manages data?
   If yes, is this software able to record counts based on: Vehicle per vehicle ☐ x minutes intervals?
   If it stores data based on minute intervals, what is the minimum interval (second, 15 minutes, etc.)?

13. Is the data maintained in a historical data base? Yes ☐ No ☐

14. Do you have any plan for replacing the loop detectors with other detectors? Yes ☐ No ☐

   Comments:

15. Are separate system detectors utilized for traffic counts? Yes ☐ No ☐

   If yes what type of detector?
   And how many approaches?

16. Please endorse a sample plan of your loop details and configuration in a typical intersection.

Please email your response to:

Ali Gholami
ali.gholami32@gmail.com

Thank you for your cooperation and helping us in this survey
APPENDIX B: RENO LOOP SPECIFICATION
NOTES:

1. Where new pavement surfaces are placed, all traffic signal loop detectors shall be installed prior to the placement of the final "top" lift of the plantmix bituminous pavement material. Placement of slurry seal does not negate this requirement.

2. Slots shall be washed, blown out and thoroughly dried before installing loop conductors.

3. The additional length of each conductor for each loop shall be twisted together into a pair (at least two turns per foot) before being placed in the slot and conduit to termination pull box.

4. Loops shall be centered in lanes.

5. Left turn loop farthest from stop bar shall be on dedicated channel.

6. Where loops are to be overlaid with asphalt, the loop sealant shall be Sakrete, or approved equal, and compacted.

7. Distance between side of loop and a lead-in saw cut from adjacent detectors shall be two feet minimum. Distance between lead-in saw cuts shall be six inches minimum. Distance between lead-in saw cuts shall be twelve inches from any curb, gutter, pan lip or pavement edge.

8. All wires shall be identified in pull boxes, with loop wires as follows: blue tape indicates left turn lane loops with one band identifying loop one, two bands loop two, etc. White tape indicates through lane loops and red tape indicates right turn lanes. If there are two left turn lanes, yellow tape indicates the lane closest to the center lane of the street.

9. All inductive loops on a given channel shall be connected in series. No more than one lead in cable shall be connected to a cabinet channel termination. No more than six individual loops are to be connected to one channel.

10. For loop lead in cables greater than 500 feet, loop in length shall be installed with four (4) turns instead of three (3).

11. Loops cut into the street surface shall be sealed with "Crafco" loop sealant, or an approved equal.

12. Listed below are the minimum distances for advance loop detector placement as a function of posted speed limits, measured from the stop bar to the rear of the loop. See table below.

<table>
<thead>
<tr>
<th>Speed Limit (MPH)</th>
<th>Distance (Feet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>150</td>
</tr>
<tr>
<td>30</td>
<td>200</td>
</tr>
<tr>
<td>35</td>
<td>255</td>
</tr>
<tr>
<td>40</td>
<td>285</td>
</tr>
<tr>
<td>45</td>
<td>330</td>
</tr>
<tr>
<td>50</td>
<td>355</td>
</tr>
<tr>
<td>55</td>
<td>390</td>
</tr>
</tbody>
</table>
APPENDIX C: SPARKS LOOP SPECIFICATION
Loop detector wire shall be I.M.S.A. Spec. 51-5-1984 #14 AWG to #20 AWG with Polyethylene (XLPE/XHHW) encasing tube or approved equal. Loop Detector lead-in cable (LLC) shall be I.M.S.A. Spec 50-2-1984 #16 AWG or approved equal. Splicing of the LLC is not permitted between the signal cabinet and loop detector wire. Loops shall be a minimum of 6’ x 6’ with 3 turns per loop. Loops in adjacent lanes shall be wound in opposite directions. Preformed loops shall be encased in PVC and shall be placed between the structural and wearing course unless otherwise approved by the City Transportation Manager. All new roadways shall have the loops placed between the structural and wearing course. Loops shall be placed in accordance to the Nevada Department of Transportation, Standard Drawing T-30.1.4 called “Loop Detectors”. The LLC shall have 4 to 6 turns per foot between the end of the saw slot and the detector to prevent separation of the wires. When making the final connections in the controller cabinet, crimp type connectors should be soldered for additional security and the screws on the terminal strip securely tightened. The shield must be floated (left unconnected and insulated) at the splice end and shall be grounded to earth ground at the cabinet end only. All loops shall have a 1” long non metallic backer rod placed at 2’ intervals to secure the wires and lead-ins during the sealing operation. The backer rod shall be at least 1” below the top of pavement. All loops shall be sealed with hot applied asphalt sealant Craftco 34272 or approved equal. Quadrupole loop layouts shall not be used. Loop amplifiers shall be EDILMD622T Loop Amplifier or approved equal. One Loop Amplifier shall be EDI-ORACLE 2E or approved equal. The saw cut depth for all home runs shall be a minimum of 2½ inches. There shall be no more than 2 twisted pairs per sawed loop slot. The maximum area of asphalt allowed to be
displaced shall be no more than 6” x 6” when installing loops. LLC shall be connected to no more than 4 loops and be clearly marked by permanent means (tag and marker) in the pull box splice point and the traffic signal cabinet.
Appendix D: R Codes
# Estimation of Shared Lane Proportion

# File name: sl

al <- read.csv("C:/Users/Ali/Desktop/al.csv")
Ratio <- al$Ratio.of.Shared.Lane.to.Adjoining.Lanes
RofTurns <- al$Ratio.of.Right.Turns

boxplot(Ratio,RofTurns)
aggregate(Ratio~RofTurns,FUN=mean)

# Linear Modeling

reg1<-lm(RofTurns~Ratio)
summary(reg1)

# Generalized Linear Modeling

gl1<-glm(RofTurns~Ratio,family=binomial)
summary(gl1)

summary(gl2)

gl3<-glm(Direction~H1*H2*H3*Car.Position*Phase*Car.Type*Front.Car.Type,data=sl,family=binomial)
summary(gl3)

plot(sl$Direction, col=3, type="l"); points(predict(gl1, newdata = sl), col=4, type="l")

# How can I get predict values?

pl1<-predict(gl1,sl$Direction~sl$H1,sl$H2,sl$H3,sl$Car.Position,sl$Phase,sl$Car.Type,sl$Front.Car.Type),type='response')

pl1<-predict(gl1,sl$Direction),type='response')
pl1<-predict(gl1, newdata = data1)

# GLM, Non-Linear Terms

H1 <- sl$H1
H2 <- sl$H1
Car.Position <- sl$Car.Position
Direction <- sl$Direction

plot(Direction,H1,pch=19,col='blue')
plot(H1,Direction,pch=19,col='blue')

plot(Direction,H2,pch=19,col='blue')
plot(H2,Direction,pch=19,col='blue')

plot(Direction,Car.Position,pch=19,col='blue')
plot(H1,Direction,pch=19,col='blue')
install.packages('rgp')
require("rgp")
vignette("rgp_introduction")

functionSet1 <- functionSet("+", "+", "-")
inputVariableSet1 <- inputVariableSet("x")
constantFactorySet1 <- constantFactorySet(function() rnorm(1))
interval1 <- seq(from = -pi, to = pi, by = 0.1)
fitnessFunction1 <- function(f) rmse(f(interval1), sin(interval1))
set.seed(1)
gpResult1 <- geneticProgramming(functionSet = functionSet1,
inputVariables = inputVariableSet1,
constantSet = constantFactorySet1,

fitnessFunction = fitnessFunction1,
stopCondition = makeTimeStopCondition(5 * 60))

bestSolution1 <-
gpResult1$population[[which.min(gpResult1$fitnessValues)]]

plot(y = bestSolution1(interval1), x = interval1, type = "l", lty = 1,
xlab = "x", ylab = "y")
lines(y = sin(interval1), x = interval1, lty = 2)

#GP Example using data set

install.packages('rgp')
require("rgp")
vignette("rgp_introduction")

x1 <- seq(0, 4*pi, length.out=201)
x2 <- seq(0, 4*pi, length.out=201)
y <- sin(x1) + cos(2*x2)
data1 <- data.frame(y=y, x1=x1, x2=x2)
newFuncSet <- functionSet("+","-","*","/","sin","sqrt","^")#,"exp","ln")

#arithmeticFunctionSet (+,-,*,/),
#expLogFunctionSet (sqrt, exp, ln),
#trigonometricFunctionSet (sin, cos, tan),
#mathFunctionSet (all above)
numericConstantSet <- constantFactorySet(function() rnorm(1))

result1 <- symbolicRegression(y ~ x1 + x2,
data=data1, functionSet=newFuncSet,
stopCondition=makeStepsStopCondition(2000))

plot(data1$y, col=1, type="l"); points(predict(result1, newdata =
data1, col=18, type="l")

bf <- result1$population[[which.min(sapply(result1$population,
result1$fitnessFunction))]]
wf <- result1$population[[which.max(sapply(result1$population,
result1$fitnessFunction))]]

bf
wf

#Symbolic Regression using GP

install.packages('rgp')
require("rgp")
vignette("rgp_introduction")

H1 <- sl$H1
H2 <- sl$H1
Direction <- sl$Direction
data1 <- data.frame(Direction=Direction, H1=H1, H2=H2)
newFuncSet <- functionSet("+","-","*","/","sqrt","^")#,"exp","ln","sin","cos","tan")
#arithmeticFunctionSet (+, -, *, /),
#expLogFunctionSet (sqrt, exp, ln),
#trigonometricFunctionSet (sin, cos, tan),
#mathFunctionSet (all above)
numericConstantSet <- constantFactorySet(function() rnorm(1))
result1 <- symbolicRegression(Direction ~ H1 + H2,
data=data1, functionSet=newFuncSet,
stopCondition=makeStepsStopCondition(5000))

plot(Direction, col=3, type="l"); points(predict(result1, newdata =
data1), col=4, type="l")
bf <- result1$population[[which.min(sapply(result1$population,
result1$fitnessFunction))]]
wf <- result1$population[[which.max(sapply(result1$population,
result1$fitnessFunction))]]
bf
wf

 #######################################################################
## Estimation of Shared Lane Proportion
#######################################################################
# 8th Street
s1_8th <- read.csv("C:/Users/Ali/Desktop/sl_8th.csv")

#Generalized Linear Modeling

gl1<- glm(Direction~H1+H2+H3+Car.Position+Phase+Car.Type+Front.Car.Type,data=
s1_8th,family=binomial)
summary(gl1)

s1_8th,family=binomial)
summary(gl2)

gl3<- glm(Direction~H1*H2*H3*Car.Position*Phase*Car.Type*Front.Car.Type,data=
s1_8th,family=binomial)
summary(gl3)

#GLM, Non-Linear Terms

gl4<-glm(Direction~H1+H2+H3+Car.Position,data=s1_8th,family=binomial)
summary(gl4)

gl5<-

# Generalized Linear Modeling

# GLM, Non-Linear Terms
gl6 <-
  glm(Direction ~ I(H1^2) + H2 + I(Car.Position^2) + I(H1*Car.Position), data = sl_8th, family = binomial)
summary(gl6)

gl7 <-
  glm(Direction ~ H1 + I(Car.Position^2), data = sl_8th, family = binomial)
summary(gl7)

gl8 <-
  glm(Direction ~ H1 + H2, data = sl_8th, family = binomial)
summary(gl8)

gl9 <-
  glm(Direction ~ H1, data = sl_8th, family = binomial)
summary(gl9)

gl10 <-
  glm(Direction ~ H1 + Car.Position, data = sl_8th, family = binomial)
summary(gl10)

gl11 <-
  glm(Direction ~ H1 + I(Car.Position^.5), data = sl_8th, family = binomial)
summary(gl11)

gl12 <-
  glm(Direction ~ H1 + I(H1*Car.Position), data = sl_8th, family = binomial)
summary(gl12)

gl13 <-
  glm(Direction ~ H1 + H2 + Car.Position, data = sl_8th, family = binomial)
summary(gl13)

9th Street

sl_9th <- read.csv("C:/Users/Ali/Desktop/sl_9th.csv")

glm(Direction ~ H1 + H2 + H3 + Car.Position + Phase + Car.Type + Front.Car.Type, data = sl_9th, family = binomial)
summary(gl1)

summary(gl2)

summary(gl3)
### GLM, Non-Linear Terms

```r
gl4 <- glm(Direction~H1+H2+H3+Car.Position, data=s1_9th, family=binomial)
summary(gl4)

gl5 <- glm(Direction~H1+H2+I(Car.Position^2)+I(H1*Car.Position), data=s1_9th, family=binomial)
summary(gl5)

gl6 <- glm(Direction~I(H1^2)+H2+I(Car.Position^2)+I(H1*Car.Position), data=s1_9th, family=binomial)
summary(gl6)

gl7 <- glm(Direction~H1+I(Car.Position^2), data=s1_9th, family=binomial)
summary(gl7)

gl8 <- glm(Direction~H1+H2, data=s1_9th, family=binomial)
summary(gl8)

gl9 <- glm(Direction~H1, data=s1_9th, family=binomial)
summary(gl9)

gl10 <- glm(Direction~H1+Car.Position, data=s1_9th, family=binomial)
summary(gl10)

gl11 <- glm(Direction~H1+I(Car.Position^.5), data=s1_9th, family=binomial)
summary(gl11)

gl12 <- glm(Direction~H1+I(H1*Car.Position), data=s1_9th, family=binomial)
summary(gl12)

gl13 <- glm(Direction~H1+H2+Car.Position, data=s1_9th, family=binomial)
summary(gl13)

gl14 <- glm(Direction~H1+Car.Position, data=s1_9th, family=binomial)
summary(gl14)

gl15 <- glm(Direction~H1, data=s1_9th, family=binomial)
summary(gl15)

gl16 <- glm(Direction~H1+I(H1^2), data=s1_9th, family=binomial)
summary(gl16)

gl17 <- glm(Direction~I(H1^2), data=s1_9th, family=binomial)
summary(gl17)
```

# McCarran Street

```r
sl_McCarran <- read.csv("C:/Users/Ali/Desktop/sl_McCarran.csv")
```

### Generalized Linear Modeling

```r
gl18 <- glm(Direction~H1+H2, data=sl_9th, family=binomial)
summary(gl18)
```
```r
# GLM Non-Linear Terms

gl1 <- glm(Direction ~ H1 + H2 + H3 + Car.Position + Phase + Car.Type + Front.Car.Type, data = sl_McCarran, family = binomial)
summary(gl1)

summary(gl2)

gl3 <- glm(Direction ~ H1*H2*H3*Car.Position*Phase*Car.Type*Front.Car.Type, data = sl_McCarran, family = binomial)
summary(gl3)

#GLM, Non-Linear Terms

#gl4 <- glm(Direction ~ H1 + H2 + H3 + Car.Position, data = sl_McCarran, family = binomial)
summary(gl4)

#gl5 <- glm(Direction ~ H1 + H2 + I(Car.Position^2) + I(H1*Car.Position), data = sl_McCarran, family = binomial)
summary(gl5)

#gl6 <- glm(Direction ~ I(H1^2) + H2 + I(Car.Position^2) + I(H1*Car.Position), data = sl_McCarran, family = binomial)
summary(gl6)

#gl7 <- glm(Direction ~ H1 + I(Car.Position^2), data = sl_McCarran, family = binomial)
summary(gl7)

#gl8 <- glm(Direction ~ H1 + H2, data = sl_McCarran, family = binomial)
summary(gl8)

#gl9 <- glm(Direction ~ H1, data = sl_McCarran, family = binomial)
summary(gl9)

#gl10 <- glm(Direction ~ H1 + Car.Position, data = sl_McCarran, family = binomial)
summary(gl10)

#gl11 <- glm(Direction ~ H1 + I(Car.Position^.5), data = sl_McCarran, family = binomial)
summary(gl11)

#gl12 <- glm(Direction ~ H1 + I(H1*Car.Position), data = sl_McCarran, family = binomial)
summary(gl12)

#gl13 <- glm(Direction ~ H1 + H2 + Car.Position, data = sl_McCarran, family = binomial)
summary(gl13)

#gl14 <- glm(Direction ~ H1 + Car.Position, data = sl_McCarran, family = binomial)
summary(gl14)
```
gl15<-glm(Direction~H1, data=sl_McCarran, family=binomial)
summary(gl15)

gl16<-glm(Direction~H1+I(H1^2), data=sl_McCarran, family=binomial)
summary(gl16)

gl17<-glm(Direction~I(H1^2), data=sl_McCarran, family=binomial)
summary(gl17)
Appendix E: MATLAB Codes
% This program extracts Green time and its relevant volume per cycle from
% VISSIM output. After making these Green and Volume records they will be
% aggregated in another program to make them ready for making models
clc
clear
sim_res = 10; % Simulation Resolution (e.g. 10 means time intervals are
% 0.1 sec)
detector_distance = 1 * sim_res; % Distance from stop bar (sec)
nlanes = 2; % Number of lanes per phase, this means number of detectors
% per phase
nphase = 5; % Number of phases, this data is used for reading the excel
% file
warmup = (1 * sim_res) - detector_distance; % Warmup period
nintersections = 6;

 [~,~,output] = xlsread('test.xlsx');

 total_volume = [];
gvpc = [];

 for i = 1 : nphase * nintersections
   disp ('Calculating G/V per phase:');
   disp (i);
   phase_info = output(:,((i-1)*(nlanes+1)+1):((i-1)*nlanes+1)+nlanes+1));

   phase_info = phase_info(1:end-warmup+1:end,:); % Removing warmup records

   if nlanes > 2
     phase_info1 = phase_info(1:end-(detector_distance),1);
   else
     phase_info1 = [phase_info(1:1:end); phase_info];
   end
   phase_info1 = [phase_info1, phase_info(:,2:end)];
   phase_info = phase_info(11:end,:);

   volume_links = [];
   for k = 2 : nlanes+1
     output1 = phase_info(:,1);
     output2 = phase_info(:,k);
     output3 = [output1 output2];
     gvpc1 = sv(output3);
     volume_linek = gvpc1(:,2);
     volume_links = [volume_links volume_linek];
   end
   volume_links1 = volume_links(:,1); % If number of lanes per phase is more or less than 2 then change it here
   volume_links2 = volume_links(:,2);
   volume_phasei = volume_links1 + volume_links2;
total_volume_phasei = sum(volume_phasei);
total_volume = [total_volume total_volume_phasei];

gvpc_phasei = [gvpc1(:,1) volume_phasei];
gvpc.([n', num2str(i)]) = gvpc_phasei;

end

for i = 1 : nphase * nintersections
disp ('Saving phase:')
disp (i);
z = gvpc.([n', num2str(i)]);
x = z (1:end,1);
y = z (1:end,2);
figure
scatter(x,y);
title ([Total Volume:', num2str(total_volume(1,i))]);
xlabel ('Green Time (sec)');
ylabel ('Volume per Green Time (veh)');
saveas(gcf,([SideStreet_100_ ', num2str(i+12)]), 'tiffn')
grid on;
y = z';
xlswrite ('gvpc.xlsx', y, 'Sheet1','[a', num2str(i*2)]);
end

%sv.m
function gvpc = sv(output)
% This function calculate volume at each green+yelloow time. The output is
% a matrix that one column shows green+yelloow time and another column the
% total number of vehicles which have passed during this time.
% clc
% clear
% [~,~,output] = xlsread('test.xlsx');

output = cell2mat (output);
s_output = size (output);
output1 = output(:,1);
output2 = output(:,((s_output(1,2)) ./ 2)+1));
output = [output1 output2];
%Changing the texts to number
for i = 1 : s_output(1,1)
    if output(i,1) == 'I' %Green
        output(i,1) = 1;
    elseif output(i,1) == '/' %Yellow
        output(i,1) = 2;
    elseif output(i,1) == '.' %Red
        output(i,1) = 3;
    end
end
for i = 1 : s_output(1,1)
    if output(i,2) == '|' %Detector Activation
        output(i,2) = 1;
    elseif output(i,2) == '?' %Detector Empty
        output(i,2) = 1;
    elseif output(i,2) == '.' %Detector Empty
        output(i,2) = 0;
    elseif output(i,2) == '+' %Detector Errors
        output(i,2) = 1;
    end
end
% output = cell2mat (output);

%Total Volume
% gcount = [];
% for i=1:(s_output(1,1)) - 1
%     if (output(i,1)==1 || 2) && output(i,2)==1 && output(i+1,2)==0
%         gcount = [gcount 1];
%     end
% end
%End of Yellow

for i=1:(s_output(1,1)) - 1
    if (output(i,1)== 2) && output(i+1,1)==3
        g_end = [g_end i];
    end
end
if output(s_output(1,1),1)== 1
    g_end = [g_end s_output(1,1)];
elseif output(s_output(1,1),1)== 2
    g_end = [g_end s_output(1,1)];
end
%Start of Green
if (output(1,1)== 1)
    g_start = 1;
elseif (output(1,1)== 2)
    g_start = 1;
else
    g_start = [];
end
for i = 1:(s_output(1,1)) - 1
    if (output(i,1) == 3) && output(i+1,1) == 1
        g_start = [g_start i+1];
    end
end

% Green and Yellow Seconds
g_sec = g_end - g_start + 1;

% Green and Yellow Volume
vpc = [];
for i = 1:length(g_end);
    s = g_start(1,i);
    e = g_end(1,i);
    Veh1 = [0 output(s:e,2)];
    Veh2 = [output(s:e,2) 0];
    x = [find(Veh == 1)
         0];
    y = [0
         x(1:(length(x)-1))];
    numV = length(find((Veh1 - Veh2)==1));
    disp(['for cycle ' num2str(i) ':
          num2str(numV)])
    if i ~= (length(g_end));
        if output(e,2) == 1 && output(e+1,2) == 1
            numV = numV - 1;
        end
    end
    vpc = [vpc numV]; % Vehicle per Cycle
end

% Vehicle per Cycle

% Vehicle per Cycle

% "aggregate.m"
% After extracting data from VISSIM output and saving them in gvpces.xlsc
using the signal_info.m program, this program prepares data for making models.
"aggregate2.m" is for volumes that are more realistic for example there is not major 200 and minor 1800 (if during simulation all volumes were simulated, otherwise simply use aggregate.m). "aggregate.m" has all these volumes.

clear
clc
% [~,~,output] = xlsread('gvpc_non major street.xlsx');

[output1] = xlsread('gvpc.xlsx','Phase 1');
[output2] = xlsread('gvpc.xlsx','Phase 2');
[output3] = xlsread('gvpc.xlsx','Phase 3');
[output4] = xlsread('gvpc.xlsx','Phase 4');
[output5] = xlsread('gvpc.xlsx','Phase 5');
[output6] = xlsread('gvpc.xlsx','Phase 6');
[output7] = xlsread('gvpc.xlsx','Phase 7');
[output8] = xlsread('gvpc.xlsx','Phase 8');
% [output1800] = xlsread('gvpc_non major street.xlsx','1800');
% [output2000] = xlsread('gvpc_non major street.xlsx','2000');
% [output2200] = xlsread('gvpc_non major street.xlsx','2200');
% [output2400] = xlsread('gvpc_non major street.xlsx','2400');

s = size(output1);
gvpc = [];
for j = 1 : 2 : s(1,1)
    for i = 1 : s(1,2)
        major_v = nansum(output1(j+1,1:end));
        if isnan(output1(j,i))
            gvpc = gvpc;
        else
            gvpc = [gvpc; [output1(j,i),output1(j+1,i),major_v,1]];
        end
    end
end

s = size(output2);
for j = 1 : 2 : s(1,1)
    for i = 1 : s(1,2)
        major_v = nansum(output2(j+1,1:end));
        if isnan(output2(j,i))
            gvpc = gvpc;
        else
            gvpc = [gvpc; [output2(j,i),output2(j+1,i),major_v,2]];
        end
    end
end
end

s = size(output3);
    for j = 1 : 2 : s(1,1)
        for i = 1 : s(1,2)
            major_v = nansum(output3(j+1,1:end));
            if isnan(output3(j,i))
                gvpc = gvpc;
            else
                gvpc = [gvpc; [output3(j,i),output3(j+1,i),major_v,3]];
            end
        end
    end

s = size(output4);
    for j = 1 : 2 : s(1,1)
        for i = 1 : s(1,2)
            major_v = nansum(output4(j+1,1:end));
            if isnan(output4(j,i))
                gvpc = gvpc;
            else
                gvpc = [gvpc; [output4(j,i),output4(j+1,i),major_v,4]];
            end
        end
    end

s = size(output5);
    for j = 1 : 2 : s(1,1)
        for i = 1 : s(1,2)
            major_v = nansum(output5(j+1,1:end));
            if isnan(output5(j,i))
                gvpc = gvpc;
            else
                gvpc = [gvpc; [output5(j,i),output5(j+1,i),major_v,5]];
            end
        end
    end

s = size(output6);
    for j = 1 : 2 : s(1,1)
        for i = 1 : s(1,2)
            major_v = nansum(output6(j+1,1:end));
            if isnan(output6(j,i))
                gvpc = gvpc;
            else
                gvpc = [gvpc; [output6(j,i),output6(j+1,i),major_v,6]];
            end
        end
    end

s = size(output7);
    for j = 1 : 2 : s(1,1)
        for i = 1 : s(1,2)
major_v = nansum(output7(j+1,1:end));
if isnan(output7(j,i))
    gvpc = gvpc;
else
    gvpc = [gvpc; [output7(j,i),output7(j+1,i),major_v,7]];
end
end

end

s = size(output8);
for j = 1 : 2 : s(1,1)
    for i = 1 : s(1,2)
        major_v = nansum(output8(j+1,1:end));
        if isnan(output8(j,i))
            gvpc = gvpc;
        else
            gvpc = [gvpc; [output8(j,i),output8(j+1,i),major_v,8]];
        end
    end
end

% s = size(output1800);
% for j = 1 : 2 : s(1,1)
%     for i = 1 : s(1,2)
%         major_v = nansum(output1800(j+1,1:end));
%         if isnan(output1800(j,i))
%             gvpc = gvpc;
%         else
%             gvpc = [gvpc; [output1800(j,i),output1800(j+1,i),major_v,1800]];
%         end
%     end
% end

% s = size(output2000);
% for j = 1 : 2 : s(1,1)
%     for i = 1 : s(1,2)
%         major_v = nansum(output2000(j+1,1:end));
%         if isnan(output2000(j,i))
%             gvpc = gvpc;
%         else
%             gvpc = [gvpc; [output2000(j,i),output2000(j+1,i),major_v,2000]];
%         end
%     end
% end

% s = size(output2200);
% for j = 1 : 2 : s(1,1)
%     for i = 1 : s(1,2)
%         major_v = nansum(output2200(j+1,1:end));
%         if isnan(output2200(j,i))
%             gvpc = gvpc;
% else
gvpc = [gvpc; [output2200(j,i),output2200(j+1,i),major_v,2200]]; end
% end

% s = size(output2400);
% for j = 1 : 2 : s(1,1)
%     for i = 1 : s(1,2)
%         major_v = nansum(output2400(j+1,1:end));
%         if isnan(output2400(j,i))
%             gvpc = gvpc;
%         else
%             gvpc = [gvpc; [output2400(j,i),output2400(j+1,i),major_v,2400]]; end
% end
% end
xlswrite ('gvpc_for_R.xlsx', gvpc, 'Example', 'A2');
x = gvpc(1:end,1);
y = gvpc(1:end,3);
z = gvpc(1:end,2);
plot3(x,y,z,'ro')

%Ranking Major Street Volume
gvpc_ranked = gvpc;
for i = 1 : length(gvpc_ranked)
    if gvpc_ranked(i,3) < 300
        gvpc_ranked(i,3) = 1;
    elseif gvpc_ranked(i,3) >= 300 && gvpc_ranked(i,3) < 600
        gvpc_ranked(i,3) = 2;
    elseif gvpc_ranked(i,3) >= 600 && gvpc_ranked(i,3) < 900
        gvpc_ranked(i,3) = 3;
    elseif gvpc_ranked(i,3) >= 900 && gvpc_ranked(i,3) < 1200
        gvpc_ranked(i,3) = 4;
    elseif gvpc_ranked(i,3) >= 1200
        gvpc_ranked(i,3) = 5;
    end
end

%Ranking Minor Street Volume

% for i = 1 : length(gvpc_ranked)
%     if gvpc_ranked(i,4) < 500
%         gvpc_ranked(i,4) = 1;
%     elseif gvpc_ranked(i,4) >= 500 && gvpc_ranked(i,4) < 1000
%         gvpc_ranked(i,4) = 2;
%     elseif gvpc_ranked(i,4) >= 1000 && gvpc_ranked(i,4) < 1500
%         gvpc_ranked(i,4) = 3;
%     elseif gvpc_ranked(i,4) >= 1500 && gvpc_ranked(i,4) < 2000
% gvpc_ranked(i,4) = 4;
% elseif gvpc_ranked(i,4) >= 2000
%      gvpc_ranked(i,4) = 5;
% end
% end

xlswrite ('gvpc_for_R_ranked.xlsx', gvpc_ranked, 'Example', 'A2');

gvpc_1 = [];
gvpc_2 = [];
gvpc_3 = [];
gvpc_4 = [];
gvpc_5 = [];
for i = 1 : length(gvpc_ranked)
  if gvpc_ranked(i,3) == 1
    gvpc_1 = [gvpc_1; gvpc_ranked(i,:)];
  elseif gvpc_ranked(i,3) == 2
    gvpc_2 = [gvpc_2; gvpc_ranked(i,:)];
  elseif gvpc_ranked(i,3) == 3
    gvpc_3 = [gvpc_3; gvpc_ranked(i,:)];
  elseif gvpc_ranked(i,3) == 4
    gvpc_4 = [gvpc_4; gvpc_ranked(i,:)];
  elseif gvpc_ranked(i,3) == 5
    gvpc_5 = [gvpc_5; gvpc_ranked(i,:)];
  end
end

xlswrite ('gvpc_for_R_1.xlsx', gvpc_1, 'Example', 'A2');
 xlswrite ('gvpc_for_R_2.xlsx', gvpc_2, 'Example', 'A2');
 xlswrite ('gvpc_for_R_3.xlsx', gvpc_3, 'Example', 'A2');
 xlswrite ('gvpc_for_R_4.xlsx', gvpc_4, 'Example', 'A2');
 xlswrite ('gvpc_for_R_5.xlsx', gvpc_5, 'Example', 'A2');

x = gvpc(1:end,1);
y = gvpc(1:end,2);
figure
scatter(x,y);
title ('All Times');
xlabel ('Green Time (sec)');
ylabel ('Volume per Green Time (veh)');
saveas(gcf, ('all'), 'tiffn')

x = gvpc_1(1:end,1);
y = gvpc_1(1:end,2);
figure
scatter(x,y);
title ('Very Low Volume');
xlabel ('Green Time (sec)');
ylabel ('Volume per Green Time (veh)');
saveas(gcf, ('1'), 'tiffn')
x = gvpc_2(1:end,1);
y = gvpc_2(1:end,2);
figure
scatter(x,y);
title ('Low Volume');
xlabel ('Green Time (sec)');
ylabel ('Volume per Green Time (veh)');
saveas(gcf,('2'), 'tiffn')

x = gvpc_3(1:end,1);
y = gvpc_3(1:end,2);
figure
scatter(x,y);
title ('Medium Volume');
xlabel ('Green Time (sec)');
ylabel ('Volume per Green Time (veh)');
saveas(gcf,('3'), 'tiffn')

x = gvpc_4(1:end,1);
y = gvpc_4(1:end,2);
figure
scatter(x,y);
title ('High Volume');
xlabel ('Green Time (sec)');
ylabel ('Volume per Green Time (veh)');
saveas(gcf,('4'), 'tiffn')

x = gvpc_5(1:end,1);
y = gvpc_5(1:end,2);
figure
scatter(x,y);
title ('Very High Volume (Peak Hour)');
xlabel ('Green Time (sec)');
ylabel ('Volume per Green Time (veh)');
saveas(gcf,('5'), 'tiffn')

%aggregate2.m
%aggregate2 is for volumes that are more realistic for example there is not
%major 200 and minor 1800. "aggregate" has all these volumes
clear
clc
% [~,~,output] = xlsread('gvpc_non major street.xlsx');

[output200] = xlsread('gvpc_non major street2.xlsx','200');
[output400] = xlsread('gvpc_non major street2.xlsx','400');
[output600] = xlsread('gvpc_non major street2.xlsx','600');
[output800] = xlsread('gvpc_non major street2.xlsx','800');
[output1000] = xlsread('gvpc_non major street2.xlsx','1000');
[output1200] = xlsread('gvpc_non major street2.xlsx','1200');
```matlab
[output1400] = xlsread('gvpc_non major street2.xlsx','1400');
[output1600] = xlsread('gvpc_non major street2.xlsx','1600');
[output1800] = xlsread('gvpc_non major street2.xlsx','1800');
[output2000] = xlsread('gvpc_non major street2.xlsx','2000');
[output2200] = xlsread('gvpc_non major street2.xlsx','2200');
[output2400] = xlsread('gvpc_non major street2.xlsx','2400');

s = size(output200);
gvpc = [];
for j = 1 : 2 : s(1,1)
    for i = 1 : s(1,2)
        major_v = nansum(output200(j+1,1:end));
        if isnan(output200(j,i))
            gvpc = gvpc;
        else
            gvpc = [gvpc,
                    output200(j,i),output200(j+1,i),major_v,200];
        end
    end
end

s = size(output400);
for j = 1 : 2 : s(1,1)
    for i = 1 : s(1,2)
        major_v = nansum(output400(j+1,1:end));
        if isnan(output400(j,i))
            gvpc = gvpc;
        else
            gvpc = [gvpc,
                    output400(j,i),output400(j+1,i),major_v,400];
        end
    end
end

s = size(output600);
for j = 1 : 2 : s(1,1)
    for i = 1 : s(1,2)
        major_v = nansum(output600(j+1,1:end));
        if isnan(output600(j,i))
            gvpc = gvpc;
        else
            gvpc = [gvpc,
                    output600(j,i),output600(j+1,i),major_v,600];
        end
    end
end

s = size(output800);
```
for j = 1 : 2 : s(1,1)
for i = 1 : s(1,2)
    major_v = nansum(output800(j+1,1:end));
    if isnan(output800(j,i))
        gvpc = gvpc;
    else
        gvpc = [gvpc;
    [output800(j,i),output800(j+1,i),major_v,800]];
end
end

s = size(output1000);
for j = 1 : 2 : s(1,1)
for i = 1 : s(1,2)
    major_v = nansum(output1000(j+1,1:end));
    if isnan(output1000(j,i))
        gvpc = gvpc;
    else
        gvpc = [gvpc;
    [output1000(j,i),output1000(j+1,i),major_v,1000]];
end
end

s = size(output1200);
for j = 1 : 2 : s(1,1)
for i = 1 : s(1,2)
    major_v = nansum(output1200(j+1,1:end));
    if isnan(output1200(j,i))
        gvpc = gvpc;
    else
        gvpc = [gvpc;
    [output1200(j,i),output1200(j+1,i),major_v,1200]];
end
end

s = size(output1400);
for j = 1 : 2 : s(1,1)
for i = 1 : s(1,2)
    major_v = nansum(output1400(j+1,1:end));
    if isnan(output1400(j,i))
        gvpc = gvpc;
    else
        gvpc = [gvpc;
    [output1400(j,i),output1400(j+1,i),major_v,1400]];
end
end

s = size(output1600);
for j = 1 : 2 : s(1,1)
for i = 1 : s(1,2)
    major_v = nansum(output1600(j+1,1:end));
    if isnan(output1600(j,i))
        gvpc = gvpc;
    else
        gvpc = [gvpc;
                output1600(j,i),output1600(j+1,i),major_v,1600]];
    end
end

s = size(output1800);
for j = 1 : 2 : s(1,1)
    for i = 1 : s(1,2)
        major_v = nansum(output1800(j+1,1:end));
        if isnan(output1800(j,i))
            gvpc = gvpc;
        else
            gvpc = [gvpc;
                    output1800(j,i),output1800(j+1,i),major_v,1800]];
        end
    end
end

s = size(output2000);
for j = 1 : 2 : s(1,1)
    for i = 1 : s(1,2)
        major_v = nansum(output2000(j+1,1:end));
        if isnan(output2000(j,i))
            gvpc = gvpc;
        else
            gvpc = [gvpc;
                    output2000(j,i),output2000(j+1,i),major_v,2000]];
        end
    end
end

s = size(output2200);
for j = 1 : 2 : s(1,1)
    for i = 1 : s(1,2)
        major_v = nansum(output2200(j+1,1:end));
        if isnan(output2200(j,i))
            gvpc = gvpc;
        else
            gvpc = [gvpc;
                    output2200(j,i),output2200(j+1,i),major_v,2200]];
        end
    end
end

s = size(output2400);
for j = 1 : 2 : s(1,1)
    for i = 1 : s(1,2)
major_v = nansum(output2400(j+1,1:end));
if isnan(output2400(j,i))
    gvpc = gvpc;
else
    gvpc = [gvpc; output2400(j,i),output2400(j+1,i),major_v,2400];
end
end
end
xlswrite ('gvpc_for_R2.xlsx', gvpc, 'Example', 'A2');
x = gvpc(1:end,1);
y = gvpc(1:end,3);
z = gvpc(1:end,2);
plot3(x,y,z,'ro')

%Ranking Major Street Volume

gvpc_ranked = gvpc;
for i = 1 : length(gvpc_ranked)
    if gvpc_ranked(i,3) < 500
        gvpc_ranked(i,3) = 1;
    elseif gvpc_ranked(i,3) >= 500 && gvpc_ranked(i,3) < 1000
        gvpc_ranked(i,3) = 2;
    elseif gvpc_ranked(i,3) >= 1000 && gvpc_ranked(i,3) < 1500
        gvpc_ranked(i,3) = 3;
    elseif gvpc_ranked(i,3) >= 1500 && gvpc_ranked(i,3) < 2000
        gvpc_ranked(i,3) = 4;
    elseif gvpc_ranked(i,3) >= 2000
    end
end

%Ranking Minor Street Volume

for i = 1 : length(gvpc_ranked)
    if gvpc_ranked(i,4) < 500
        gvpc_ranked(i,4) = 1;
    elseif gvpc_ranked(i,4) >= 500 && gvpc_ranked(i,4) < 1000
        gvpc_ranked(i,4) = 2;
    elseif gvpc_ranked(i,4) >= 1000 && gvpc_ranked(i,4) < 1500
        gvpc_ranked(i,4) = 3;
    elseif gvpc_ranked(i,4) >= 1500 && gvpc_ranked(i,4) < 2000
        gvpc_ranked(i,4) = 4;
    elseif gvpc_ranked(i,4) >= 2000
        gvpc_ranked(i,4) = 5;
end
end
xlswrite ('gvpc_for_R_ranked2.xlsx', gvpc_ranked, 'Example', 'A2');
gvpc_1 = [];
gvpc_2 = []; 
gvpc_3 = []; 
gvpc_4 = []; 
gvpc_5 = [];
for i = 1:length(gvpc_ranked)
    if gvpc_ranked(i,3) == 1
        gvpc_1 = [gvpc_1; gvpc_ranked(i,:)];
    elseif gvpc_ranked(i,3) == 2
        gvpc_2 = [gvpc_2; gvpc_ranked(i,:)];
    elseif gvpc_ranked(i,3) == 3
        gvpc_3 = [gvpc_3; gvpc_ranked(i,:)];
    elseif gvpc_ranked(i,3) == 4
        gvpc_4 = [gvpc_4; gvpc_ranked(i,:)];
    elseif gvpc_ranked(i,3) == 5
        gvpc_5 = [gvpc_5; gvpc_ranked(i,:)];
    end
end

xlswrite ('gvpc_for_R_12.xlsx', gvpc_1, 'Example', 'A2');
xlswrite ('gvpc_for_R_22.xlsx', gvpc_2, 'Example', 'A2');
xlswrite ('gvpc_for_R_32.xlsx', gvpc_3, 'Example', 'A2');
% xlswrite ('gvpc_for_R_42.xlsx', gvpc_4, 'Example', 'A2');
% xlswrite ('gvpc_for_R_52.xlsx', gvpc_5, 'Example', 'A2');

x = gvpc(1:end,1);
y = gvpc(1:end,2);
figure
scatter(x,y);
title ('All Times');
xlabel ('Green Time (sec)');
ylabel ('Volume per Green Time (veh)');
saveas(gcf, ('all2'), 'tiffn')

x = gvpc_1(1:end,1);
y = gvpc_1(1:end,2);
figure
scatter(x,y);
title ('Very Low Volume');
xlabel ('Green Time (sec)');
ylabel ('Volume per Green Time (veh)');
saveas(gcf, ('12'), 'tiffn')

x = gvpc_2(1:end,1);
y = gvpc_2(1:end,2);
figure
scatter(x,y);
title ('Low Volume');
xlabel ('Green Time (sec)');
ylabel ('Volume per Green Time (veh)');
saveas(gcf, ('22'), 'tiffn')

x = gvpc_3(1:end,1);
$y = \text{gvpc}_3(1:\text{end},2);$  
figure  
scatter($x,y$);  
title ('Medium Volume');  
xlabel ('Green Time (sec)');  
ylabel ('Volume per Green Time (veh)');  
saveas(gcf,('32'), 'tiffn')

$x = \text{gvpc}_4(1:\text{end},1);$  
y = \text{gvpc}_4(1:\text{end},2);$  
figure  
scatter($x,y$);  
title ('High Volume');  
xlabel ('Green Time (sec)');  
ylabel ('Volume per Green Time (veh)');  
saveas(gcf,('42'), 'tiffn')

$x = \text{gvpc}_5(1:\text{end},1);$  
y = \text{gvpc}_5(1:\text{end},2);$  
figure  
scatter($x,y$);  
title ('Very High Volume (Peak Hour)');  
xlabel ('Green Time (sec)');  
ylabel ('Volume per Green Time (veh)');  
saveas(gcf,('52'), 'tiffn')