Technical Report Documentation Page

1. Report No. FHWA/TX-06/0-4770-1	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AN INVESTIGATION INTO THE EVALUATION AND OPTIMIZATION OF THE AUTOMATIC INCIDENT DETECTION ALGORITHM USED IN TXDOT TRAFFIC MANAGEMENT SYSTEMS		5. Report Date October 2005 6. Performing Organization Code
7. Author(s) Robert E. Brydia, Jeremy D. Johnson, and Kevin N. Balke		8. Performing Organization Report No. Report 0-4770-1
9. Performing Organization Name and Address Texas Transportation Institute The Texas A&M University System College Station, Texas 77843-3135		10. Work Unit No. (TRAIS) 11. Contract or Grant No. Project 0-4770
12. Sponsoring Agency Name and Address Texas Department of Transportation Research and Technology Implementation Office P.O. Box 5080 Austin, Texas 78763-5080		13. Type of Report and Period Covered Technical Report: September 2003 – August 2005 14. Sponsoring Agency Code

15. Supplementary Notes
Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration.
Project Title: Evaluation and Optimization of Automatic Incident Detection Algorithm Used in TxDOT Traffic Management Systems
URL: http://tti.tamu.edu/documents/0-4770-1.pdf

16. Abstract
TxDOT traffic management systems have the capability to monitor real-time traffic flow data for automatic incident detection. The faster an incident is detected, the more rapid the response, which decreases congestion on the roadways. This detection capability is centered on an existing algorithm that utilizes loop occupancy from roadway loop detectors.
This research proposes a minimal modification to the incident detection algorithm which decreases false alarms and increases the detection rate, as determined by a multi-year assessment of its performance characteristics, using archived data.
The project delivers a revised algorithm, a procedure for setting incident detection thresholds, a logic flow for an automated tool, and recommendations for improving the incident detection process and data archives.

17. Key Words Incident Detection, Algorithm, Freeway Traffic Management. Lane Occupancy, Detector Data		18. Distribution Statement No restrictions. This document is available to the public through NTIS: National Technical Information Service Springfield, Virginia 22161 http://www.ntis.gov	
19. Security Classif.(of this report) Unclassified	20. Security Classif.(of this page) Unclassified	21. No. of Pages 110	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

An Investigation into the Evaluation and Optimization of the Automatic Incident Detection Algorithm Used in TxDOT Traffic Management Systems

by

Robert E. Brydia
Associate Research Scientist
Texas Transportation Institute

Jeremy D. Johnson
Engineering Research Associate
Texas Transportation Institute

and

Kevin N. Balke
Center Director
Texas Transportation Institute

Report 0-4770-1
Project 0-4770
Project Title: Evaluation and Optimization of Automatic Incident Detection Algorithm Used in TxDOT Traffic Management Systems

Performed in cooperation with the
Texas Department of Transportation
and the
Federal Highway Administration

October 2005

TEXAS TRANSPORTATION INSTITUTE
The Texas A&M University System
College Station, Texas 77843-3135

DISCLAIMER

This research was performed in cooperation with the Texas Department of Transportation (TxDOT) and the Federal Highway Administration (FHWA). The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official view or policies of the FHWA or TxDOT. This report does not constitute a standard, specification, or regulation. The researcher in charge was Robert E. Brydia.

ACKNOWLEDGMENTS

This project was conducted in cooperation with TxDOT and FHWA. The authors gratefully acknowledge the contributions of numerous persons who made the successful completion of this guidebook possible.

Project Coordinator
- Al Kosik, P.E., Traffic Operations Division, TxDOT

Project Director
- Joyce Seebock, Traffic Operations Division, TxDOT

Project Monitoring Committee
- Chris Harris, Traffic Operations Division, TxDOT
- Brian Burk, Austin District, TxDOT
- Charlie Cardenas, Corpus Christi District, TxDOT
- John Gold, Austin District, TxDOT

RTI Engineer
- Wade Odell, P.E., Research and Technology Implementation Office, TxDOT

TABLE OF CONTENTS

Page

List of Figures .... **ix**
List of Tables .... **xi**
CHAPTER 1 – INTRODUCTION **1**
BACKGROUND 1
GOALS OF AND OBJECTIVES OF PROJECT 1
HISTORY OF PROJECT 2
PROJECT DELIVERABLES 3
FORMAT OF THIS REPORT 3
CHAPTER 2 – AN OVERVIEW OF INCIDENT DETECTION **5**
INCIDENT DETECTION ALGORITHMS 5
Comparative Algorithms 5
Statistical Algorithms 5
Time-series and Smoothing Algorithms 6
Modeling Algorithms 6
ASSESSING INCIDENT DETECTION ALGORITHM PERFORMANCE 7
CALIBRATION OF AN INCIDENT DETECTION ALGORITHM 9
Errors in Detector Data 9
ANALYSIS PROCEDURES 10
CHAPTER 3 – THE TXDOT INCIDENT DETECTION ALGORITHM **11**
STANDARD IMPLEMENTATION 11
AUSTIN IMPLEMENTATION 12
THRESHOLDS AND PROFILES 13
ESTABLISHING THRESHOLDS 14
Manual Method 14
Automated Method 15
CHAPTER 4 –PRELIMINARY DATA MANIPULATION **17**
INTRODUCTION 17
OPERATOR's LOGS 17
Step 1: Incident Log Reduction (Manual) 17
Step 2: Reduce Log File 19
DETECTOR THRESHOLDS 20
Step 1: Threshold Data Files Creation (Manual) 20
SCU DATA FILES 22
Step 1: Uncompress Raw Data Files from .ZIP Files 22
Step 2: Create Unique File Names for Every Data File 24
Step 3: Create Individual Detector Data Files 26
Step 4: Make 365 Day Matrix 27
AUSTIN DATA 28
CHAPTER 5 – THE CROSS-LANE COMPARISON ALGORITHM **31**
INTRODUCTION 31
BASIS FOR CROSS-LANE ANALYSIS 31
CLC Thresholds 37

CHAPTER 6 – DETERMINATION OF CROSS-LANE COMPARISON THRESHOLDS 39
OVERVIEW OF CLC THRESHOLD PROCESS 39
ESTABLISHING TIME POINTS 39
Overview of Time Point Selection Process 39
Data Manipulation for Time Point Selection Process 40
Manual (Visual) Process for Selecting Time Points 44
ESTABLISHING CLC ThRESHOLD VALUES 46
Manual (Visual) Process for Selecting CLC Threshold Values 47
AUTOMATING THE CLC THRESHOLD PROCESS 48
Automated Process for Selecting CLC Threshold Values 48
LOGIC FLOW FOR AUTOMATED THRESHOLD TOOL 51
CHAPTER 7 – CROSS-LANE COMPARISON PERFORMANCE ASSESSMENT 57
INTRODUCTION 57
CLC PERFORMANCE ASSESSMENT USING 2002 DATA 58
False Alarms 58
Incident Detection 64
CLC Performance Summary 68
MULTI-YEAR SUSTAINABILITY ASSESSMENT 68
Data Requirements 69
False Alarm Comparison 70
Incident Detection Comparison 71
OPERATORS' ASSESSMENT OF AUTOMATIC INCIDENT DETECTION 71
CHAPTER 8 – RECOMMENDATIONS FOR IMPROVEMENTS 81
CHAPTER 9 – CONCLUSIONS 83
INTRODUCTION 83
CROSS-LANE COMPARISON ALGORITHM 83
ASSESSMENT OF CAPABILITIES 83
PROJECT DELIVERABLES 83
REFERENCES 85
APPENDIX 89
AUTOMATIC INCIDENT DETECTION SURVEY 89

LIST OF FIGURES

Page

Figure 1. General Relationship between Detection Rate, Detection Time, and False Alarm Rate for Incident Detection Algorithms. 9
Figure 2. TxDOT Incident Detection Algorithm. 11
Figure 3. Austin Implementation of TxDOT Incident Detection Algorithm. 12
Figure 4. Typical Threshold Graph for TxDOT Incident Detection Algorithm. 13
Figure 5. Thresholds Created Under Austin Inter-Agency Contract. 15
Figure 6. Input Incident File. 18
Figure 7: Output Incident Log File. 19
Figure 8. Incident Summary File. 19
Figure 9. Detector Inventory Text File. 21
Figure 10. Detector Profile Text File. 21
Figure 11. Profile Information File. 22
Figure 12. Directory and Filename Structure Prior to Renaming. 24
Figure 13. Directory and Filename Structure After Renaming Process. 25
Figure 14. Input File for Step 3 of SCU Data Manipulation. 27
Figure 15. Output File for Step 3 of SCU Data Manipulation. 27
Figure 16. Detector Occupancy File. 28
Figure 17. 3-Minute Rolling Average and Cross-Lane Comparison for Non-Incident Conditions. 32
Figure 18. 3-Minute Rolling Average and Cross-Lane Comparison for Incident Conditions. 33
Figure 19. CLC Values for Station 63 - Non-Incident Day. 34
Figure 20. CLC Values for Station 67 - Non-Incident Day. 34
Figure 21. CLC Values for Station 63 - Incident Day. 35
Figure 22. CLC Values for Station 67 - Incident Day. 36
Figure 23. CLC Algorithm. 37
Figure 24. Station CLC Data. 41
Figure 25. 1-minute LAG of MAD. 42
Figure 26. 30-Minute Average LAG. 43
Figure 27. Slope of 30-Minute Average LAG Data. 43
Figure 28. Slope Indicator. 44
Figure 29. New Threshold Time Periods. 44
Figure 30. Selection of Time Points for CLC Thresholds. 45
Figure 31. Selection of CLC Values for Thresholds. 47
Figure 32. Overview of Automated Logic Flow. 51
Figure 33. Detailed Flow Chart for Module 1 – Import and Parse Raw Data. 52
Figure 34. Detailed Flow Chart for Module 2 – Perform Calculations. 53
Figure 35. Detailed Flow Chart for Module 3 – Determine New Time Periods. 54
Figure 36. Detailed Flow Chart for Module 4 – Determine New Time Thresholds. 55
Figure 37. False Alarm Summary. 60
Figure 38. Station False Alarms. 60
Figure 39. False Alarm Times. 61
Figure 40. Detection Times. 65

Figure 41. Operator Survey – Detection Algorithm Parameters.. 73
Figure 42. Operator Survey – Roadway Collection Time Period... 73
Figure 43. Operator Survey – Incident Detection Time Period.. 74
Figure 44. Operator Survey – Critical Factor in Incident Detection.. 75
Figure 45. Operator Survey – Desired Speed for Incident Detection.. 76
Figure 46. Operator Survey – Maximum Wait Time for 100% Detection................................ 76
Figure 47. Operator Survey – Acceptable False Alarms. 77
Figure 48. Operator Survey – Simultaneous False Alarms.. 78
Figure 49. Operator Survey – False Alarms Per Detection Rate... 79

LIST OF TABLES

Page

Table 1. US183 Average False Alarms Per Detector, Per Day, For 2002 Data. 63
Table 2. Average Daily False Alarms By Roadway For 2002 Data. 64
Table 3. Incident Detection Performance for US 183, 2002 Data. 67
Table 4. Incident Detection Rate for Austin Roadways, 2002 Data. 68
Table 5. Incident Detection Time for Austin Roadways, 2002 Data. 68
Table 6. False Alarm Comparison for Multi-Year Analysis (Average Daily Alarms Across All Austin Roadways). 70
Table 7. Incident Detection Rate Comparison for Multi-Year Analysis (Average Percenatage of Detected Incidents Across All Austin Roadways). 71
Table 8. Incident Detection Time Comparison for Multi-Year Analysis (Average Incident Detection Time Lag to Operator's Logs Across All Austin Roadways). 71

CHAPTER 1 – INTRODUCTION

BACKGROUND

According to the Texas Transportation Institute's (TTI) recent Urban Mobility Report, incidents are responsible for somewhere between 53 and 58 percent of the total delay experienced by motorists in all urban areas (*1*). An important goal of freeway traffic management is the rapid detection of incidents and the management of traffic during incident conditions. In Texas, freeway management systems are utilized in many major urban areas including Houston, San Antonio, Ft. Worth, Dallas, Austin, El Paso, and soon to be Laredo. Freeway management systems are also planned for other major metropolitan areas in Texas such as Amarillo and Corpus Christi.

Although the control centers in Texas utilize different software and systems, the mission of all the centers is the same: to reduce the amount of congestion and improve traffic safety by minimizing the impact of incidents on the freeways through their rapid detection and clearance. A primary tool in the center's arsenal to accomplish this mission is the use of automatic incident detection algorithms.

In many control centers in Texas, the system software in use has been developed by TxDOT. Known as Advanced Traffic Management System (ATMS), the software has the capability to monitor real-time traffic flow data for automatic incident detection. The faster an incident is detected, the more rapid the response, which decreases congestion on the roadways.

Unfortunately, most implementations of the TxDOT systems operate with their incident detection algorithm turned off, as there are no established procedures for optimizing the settings to ensure the rapid detection of incidents. Because of the nature of detection algorithms and the trade-off between rapid detections and false-alarms, the lack of procedures makes it difficult for current implementations to achieve an acceptable compromise among the performance parameters

GOALS OF AND OBJECTIVES OF PROJECT

The key to optimizing performance of an incident detection algorithm is adjusting the parameters to achieve an acceptable level of performance. However, a procedure to accomplish this for the TxDOT incident detection algorithm has never been developed.

The overall goal of this research project is to develop these capabilities for TxDOT. In order to accomplish this goal, the following project objectives have been established:

1. through a survey of district freeway management operations, establish desired criteria for assessing the performance of incident detection algorithms for the state of Texas in terms of detection rate, detection time, and false alarm rates;
2. quantify the real-world performance of the TxDOT incident detection algorithm currently being used in ATMS, and
3. develop a practical procedure for setting the thresholds and profiles in TxDOT's ATMS necessary for the algorithm adjustment and optimization.

The first objective defines the level of performance TxDOT operators expect from an algorithm. Currently, most implementations of ATMS operate with the incident detection algorithm turned off. This obviously impacts the ability to detect incidents, which has an effect on both the operator workload and the driving public. What are the acceptable limits or boundaries for how the algorithm is expected to perform, to maximize incident detection, but minimize false alarm rate?

The second objective focuses on quantifying how well the current algorithm embedded in TxDOT's ATMS is truly performing, regardless of TxDOT's expectations. The results of this investigation will ultimately provide guidance to TxDOT on the future of the incident detection algorithm development and its use.

Optimizing performance of the algorithm is the focus of the third objective. While the key is developing a procedure for identifying the thresholds and profiles, many questions must be answered. Crucial among them are questions such as how much data is required for this process? How lengthy is this procedure? Can it be automated in the future? Are there different requirements and procedures necessary for different traffic conditions?

HISTORY OF PROJECT

Over the course of this project, the path by which the goals and objectives were accomplished changed from the initial plan. As initially proposed, the scope included examining data from the Austin district, formulating a threshold calibration procedure, and then calibrating that procedure with data from the El Paso district.

However, during the course of investigating the different foundations on which to build a threshold calibration procedure, a small change to the existing TxDOT algorithm was discovered that had the potential to improve on its effectiveness and performance. The project was modified to allow the research team to pursue this algorithm change. Due to this new path of research, the tasks related to the algorithm calibration using data from El Paso were removed from the project. Instead, the research focused on the new algorithm, dubbed the Cross-Lane Comparison (CLC) to quantify the potential improvements to TxDOT's incident detection capabilities within ATMS.

PROJECT DELIVERABLES

The results of this research project are implemented in four products, all of which will have an immediate use for TxDOT. Project modifications did not change the products of the project. These products are more specifically described as follows:

1. *A documented assessment of incident detection capability* which will provide TxDOT with a clear understanding of the capabilities of their existing algorithm, the new algorithm, and the operators' desires (Deliverable P1).
2. *Recommendations for modifications to the incident detection algorithm* that provide methods by which TxDOT can improve their overall incident detection capabilities (Deliverable P2).
3. *A threshold calibration procedure* that details a systematic and scientific process for creating thresholds for the new algorithm (Deliverable P3).
4. *A logic flow for an automated threshold calibration tool* to provide TxDOT with a roadmap for implementing the revised algorithm in future versions of ATMS (Deliverable P4).

FORMAT OF THIS REPORT

This report documents the investigations and results of the research performed in support of this project. Each of the deliverables above is contained within this report. The following chapters contain the deliverables detailed above:

- Deliverable 0-4770-P1 is contained in Chapter 7.
- Deliverable 0-4770-P2 is contained in Chapter 8.
- Deliverable 0-4770-P3 is contained in Chapter 6.
- Deliverable 0-4770-P4 is contained in Chapter 6.

CHAPTER 2 – AN OVERVIEW OF INCIDENT DETECTION

INCIDENT DETECTION ALGORITHMS

Over the past three decades, numerous research studies have been performed on incident detection algorithms, resulting in a large number of individual algorithms. Most incident detection algorithms can however, be grouped into four categories:

- comparative algorithms,
- statistical algorithms,
- time-series / smoothing algorithms, and
- modeling algorithms *(2,3)*.

Comparative Algorithms

Comparative algorithms are generally the simplest of all the algorithms. Comparative algorithms (sometimes called pattern recognition algorithms) generally compare speed, volume and/or occupancy measurements from a single detector station or between two detector stations against thresholds that define when incident conditions are likely. These comparisons may be temporal (comparing measures from the same detector over time) or spatial (comparing measures between two adjacent detectors). Some of the more well-known algorithms of this type include the following:

- the ten Modified California Algorithms *(4, 5)*,
- the Pattern Recognition Algorithm (PATREG) *(6)*, and
- the All-Purpose Incident Detection Algorithm *(7)*.

Statistical Algorithms

Statistical algorithms use statistical techniques to determine whether observed detector data differ statistically from historical or defined conditions (3). Most of the algorithms in this category use data from normal (i.e., non-incident) traffic conditions to develop mean and expected values for traffic conditions. Confidence intervals are then used to account for "normal" fluctuations (or variance) in traffic patterns. Real-time detector data is then compared with these "typical" values to determine if they fall outside the normal range of data for that particular station. Examples of this type of incident detection algorithm include the following:

- the Standard Normal Deviate Algorithm *(8)*, and
- the Bayesian Algorithm *(9)*.

Time-series and Smoothing Algorithms

Time-series and smoothing algorithms compare short-term predictions of traffic conditions to measured traffic conditions. This approach generally uses more complicated procedures. The concept is to look at general trends in the data to determine if what is currently being observed is what typically can be expected for that time of day, or if the observed data is "abnormal". In other words, these techniques generally track what is happening at a detection station over time, looking for large or "abnormal" changes in the traffic data. The algorithms in this class generally use techniques to "smooth" or "filter" the data to remove "noise" (or minor fluctuations) in the detector data stream. Some examples of time-series / smoothing algorithms include the following:

- the ARIMA Algorithm *(10)*,
- the Exponential Smoothing Algorithm *(11)*, and
- the Low-Pass Filter Algorithm *(12)*.

Modeling Algorithms

The final class of incident detection algorithms is the modeling algorithm. Modeling algorithms use traffic flow theories and computer simulation models to model expected traffic conditions, based on the current data coming from the detector. Generally, these models predict what traffic conditions would be like under incident conditions and then examine the detector data to determine if the predicted values are similar. Modeling algorithms include the following:

- the McMaster Algorithm *(13)* and
- the Dynamic Model Algorithm *(14)*.

In more recent years, researchers have again been tackling the issues of automatic incident detection algorithms, applying new approaches such as wavelet energy theory, fuzzy logic, and neural networks in attempt to improve the performance of incident detection algorithms (*15, 16, 17*). Also, as technology advances, new technologies such as video imaging and automatic vehicle identifications systems have been applied to finding better means of detecting freeway incidents (*18*).

ASSESSING INCIDENT DETECTION ALGORITHM PERFORMANCE

The most commonly used measures to evaluate the performance of different incident detection algorithms are:

- detection rate,
- detection time, and
- false alarm rate *(19)*.

Detection rate is typically defined as the number of incident that have been detected as a percentage of the total number of capacity-reducing incidents that occur on the system. Detection time is defined as the elapsed time from when an incident occurred on the freeway to when it is detected by the incident detection algorithm. The false alarm rate is generally used to provide an indication of how many times an algorithm incorrectly issues an incident alarm when in fact no incident is present. The two types of false alarm rates generally used to evaluate incident detection algorithm performance are on-line and off-line *(2)*.

Most incident algorithms and research use the off-line measure when reporting on the performance of their algorithms. In these situations, the false alarm rated is defined as the percentage of incorrect (or false) declarations of an incident condition out of all possible declarations including true incidents, false incidents and incident-free declarations. In other words, the off-line incident detection rate is the percentage of false detections divided by the total number of iterations (or checks) that the algorithm executes in the evaluation period. The on-line rate generally used by practitioners is defined as the percentage of false incident alarms to the total number of incident declarations, both true and false.

The following provides an example to help illustrate the difference between the two false alarm rates. In this example, we will assume that there were four true detections and one false alarm, for a total of five alarms, during a three-hour period. In this example, the "on-line" false alarm rate would be as follows:

$$False\ Alarm\ Rate_{On-line} = \frac{1\,False\ Alarm}{5\,Total\ Incident\ Alarms} \times 100\% = 20\%$$

If we assume that the incident detection algorithm executes once every minute, then the "off-line" false alarm rate would be as follows:

$$False\ Alarm\ Rate_{Off-line} = \frac{1\,FalseAlarm}{1\,Iteration/Minute \times 60\,Minutes/Hour \times 3\,Hours} \times 100\% = 0.55\%$$

Because the off-line false alarm rate includes periods where the algorithm is running, but not issuing an alarm, the off-line false alarm rate is often significantly lower than what the operator might typically experience in the control center. Therefore, when evaluating different types of incident detection algorithm, it is important to know what type of false alarm rate is being used: the on-line or off-line version.

Figure 1 shows the general relationship that typically exists between detection rate, false alarm rate, and detection time (*19*,*2*). For most algorithms, especially the comparative incident detection algorithms, the detection rate is directly proportional to detection time. In other words, the detection rate tends to increase the longer the algorithm takes in evaluating the detector data.

Conversely, the false alarm rate generally tends to be inversely proportional to the detection time. In other words, as the detection time increases, the false alarm rate decreases. Therefore, in order to maximize the detection time while minimizing the false alarm rate, the detection time must increase.

The problem comes in the fact that the goal of most incident detection algorithms is to minimize the detection time. As the algorithm thresholds are adjusted to detect less severe incidents more quickly, minor fluctuations in traffic demands can cause the false alarm rate to rise. Therefore, with many existing incident detection algorithms, there is a trade-off between detection rate, detection time, and false alarm rate. In order to minimize detection time, agencies must be willing to live with detecting only those incidents that have a major impact on traffic demands and more false alarms.

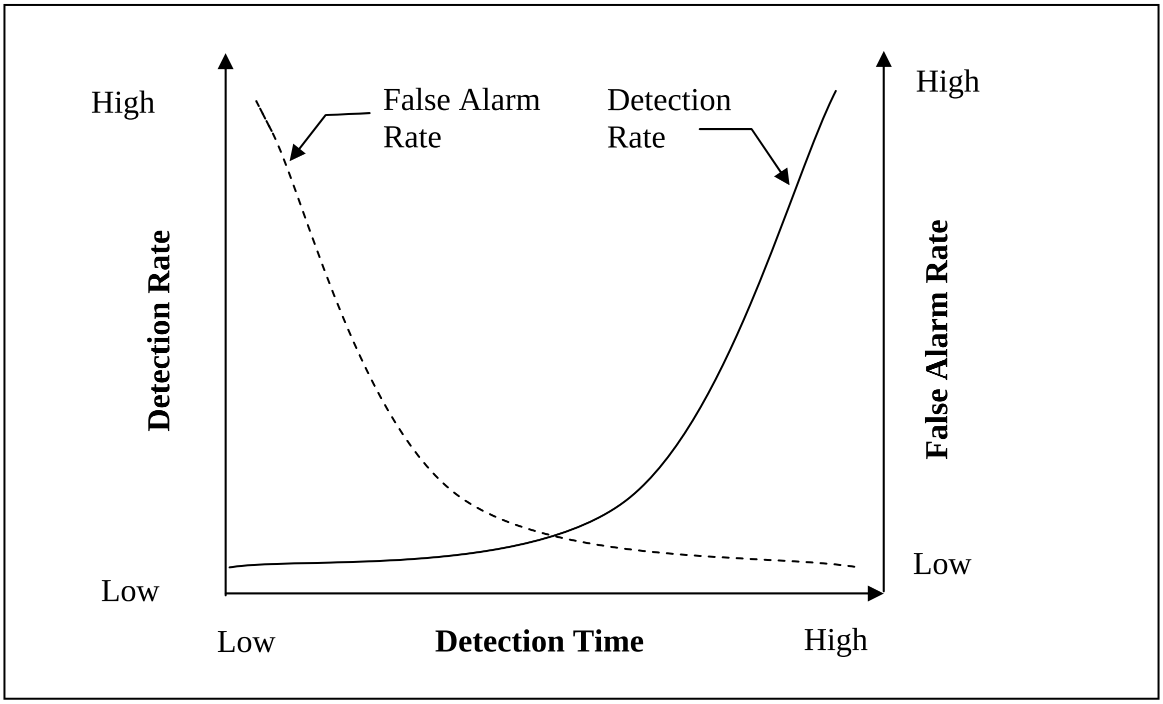


Figure 1. General Relationship between Detection Rate, Detection Time, and False Alarm Rate for Incident Detection Algorithms.

CALIBRATION OF AN INCIDENT DETECTION ALGORITHM

Most of the problems that many agencies experience with their current algorithms are due, in part, to the need for calibration to the particular site *(20)*. Local geometries and traffic operations within a freeway can significantly affect the normal traffic patterns on that freeway such that detection thresholds need to be specific to that particular section of freeway *(21)*. Generally, proper calibration of each detection zone requires extensive data collection and analysis. Data from both incident and non-incident data must be secured. Raw speed, volume and occupancy data must also be free of malfunctioning detectors (*21*).

Errors in Detector Data

Although many ITS deployments include some rudimentary data error checking, many detector failures and data errors remain undetected. A more advanced automated statistical quality control procedures to clean an ITS database has been developed by Park et al.*(22)*. These procedures were an outgrowth of work performed on DataLink, one of the first ITS Data archiving systems *(23,24)*. DataLink utilized extensive empirical rules to clean the data as it arrived, to ensure the validity of the reported results. These concepts are directly applicable to cleaning the data supplied to incident detection algorithms.

Cleaning procedures attempt to detect erroneous/faulty observations that emerge as outliers by examining each data record in relation to other temporarily close data records. While a variety of techniques can be utilized, Park et. al. used a mean curve representing underlying traffic trends fitted for each location and took into account the variability of each observation *(22)*. The variability was used to measure the degree of abnormality associated with each reading *(25.)*

Although one data type is often the key attribute in the context of an incident detection algorithm, it is important to consider all the attributes in the stage of cleaning the database because an observation that looks typical in terms of any single variable — speed, volume, or occupancy — could still be an outlier.

ANALYSIS PROCEDURES

To compound the issues of measuring performance and cleaning the data, analysis procedures to establish thresholds, deviations, or predictions are frequently unclear, poorly documented, or, in many cases, nonexistent. As a result of all these factors, properly calibrating an algorithm's performance is often impractical for operations personnel, and thus limits the effectiveness of these algorithms (*20*).

CHAPTER 3 – THE TXDOT INCIDENT DETECTION ALGORITHM

STANDARD IMPLEMENTATION

The TxDOT incident detection algorithm is a comparative mechanism. The basis for the algorithm is a 3-minute rolling average of loop occupancy, which is calculated on a per-lane basis. Occupancy is a measure of the amount of time a loop detector is occupied in a given time period, expressed as a percentage. The algorithm simply determines if the loop occupancy value is over a pre-established threshold. If it is, an alarm is raised. If not, the rolling average is refreshed with a new value and the process begins anew. Figure 2 shows the algorithm details.

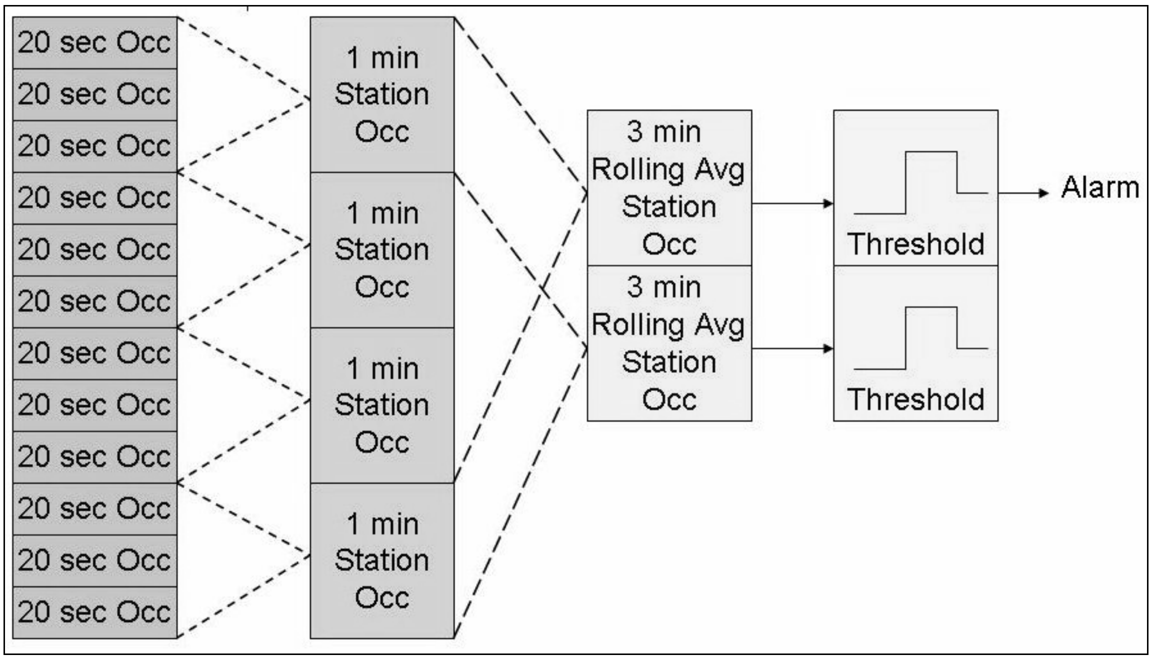


Figure 2. TxDOT Incident Detection Algorithm.

TxDOT uses a 20-second time period for loop occupancy calculations. The 20-second values of loop occupancy are read from the roadway loop detectors and sent to the local control unit (LCU), a field device which reads data from multiple field devices. The 20-second values are averaged across all lanes in the cross-section. Then, every three values of 20-second data are combined to produce a 1-minute average of occupancy across the entire cross section. This calculation is done at the level of the system control unit (SCU), a device which interfaces with multiple LCUs.

Once the 1-minute lane occupancies are calculated, every three values are averaged together into a 3-minute rolling average. This average updates once a minute, by dropping off the oldest 1-minute value and averaging in the latest 1-minute value. The use of this rolling average technique helps to smooth out normal fluctuations in the occupancy data.

Once a minute, the 3-minute rolling average is compared to a threshold to see if the average loop occupancy value is higher than a predetermined limit. If so, an alarm is triggered for the operator's attention. If the occupancy value is lower than the threshold, the algorithm waits for the next time period and again performs the calculations and comparisons.

AUSTIN IMPLEMENTATION

It is important to note that the Austin district utilizes a slightly different implementation of the TxDOT algorithm than other ATMS implementations. As will be evident later in this report, it was this different implementation that led to the basis of the investigation into the modified algorithm and potential improvements in the incident detection capabilities.

In Austin, all calculations and comparisons are performed on a per-lane basis, rather than across the entire-cross section. Figure 3 shows the revised algorithm Note that the changes from Figure 2 are minimal, reflecting simply that the calculations are done on a per-lane basis.

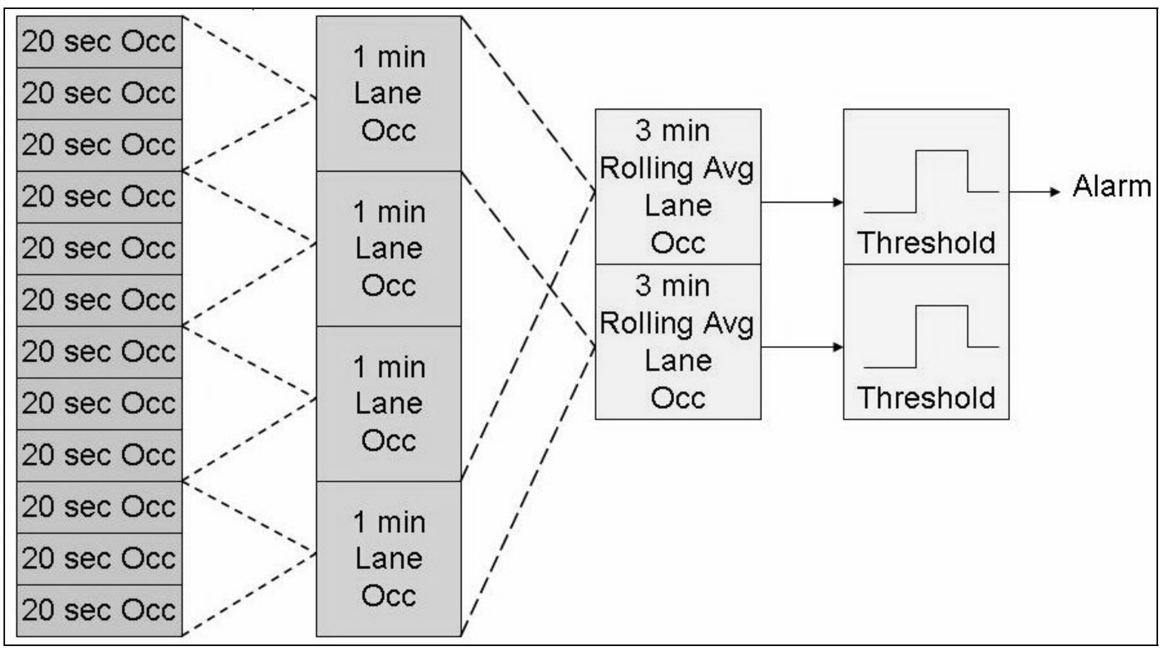


Figure 3. Austin Implementation of TxDOT Incident Detection Algorithm.

THRESHOLDS AND PROFILES

In the TxDOT algorithm, the concept of thresholds as a comparison measure to identify traffic conditions outside the norm, i.e., an incident. A threshold is nothing more than a predefined limit for loop occupancy. Exceeding the threshold indicates that a possible incident has occurred. In recognition that traffic conditions vary across the day, the TxDOT implementation allows up to six thresholds and corresponding time periods to be defined for a given day. Figure 4 shows a typical threshold graph for a detector in the TxDOT ATMS. The horizontal axis represents the time of day in hours while the vertical axis represents the loop occupancy.

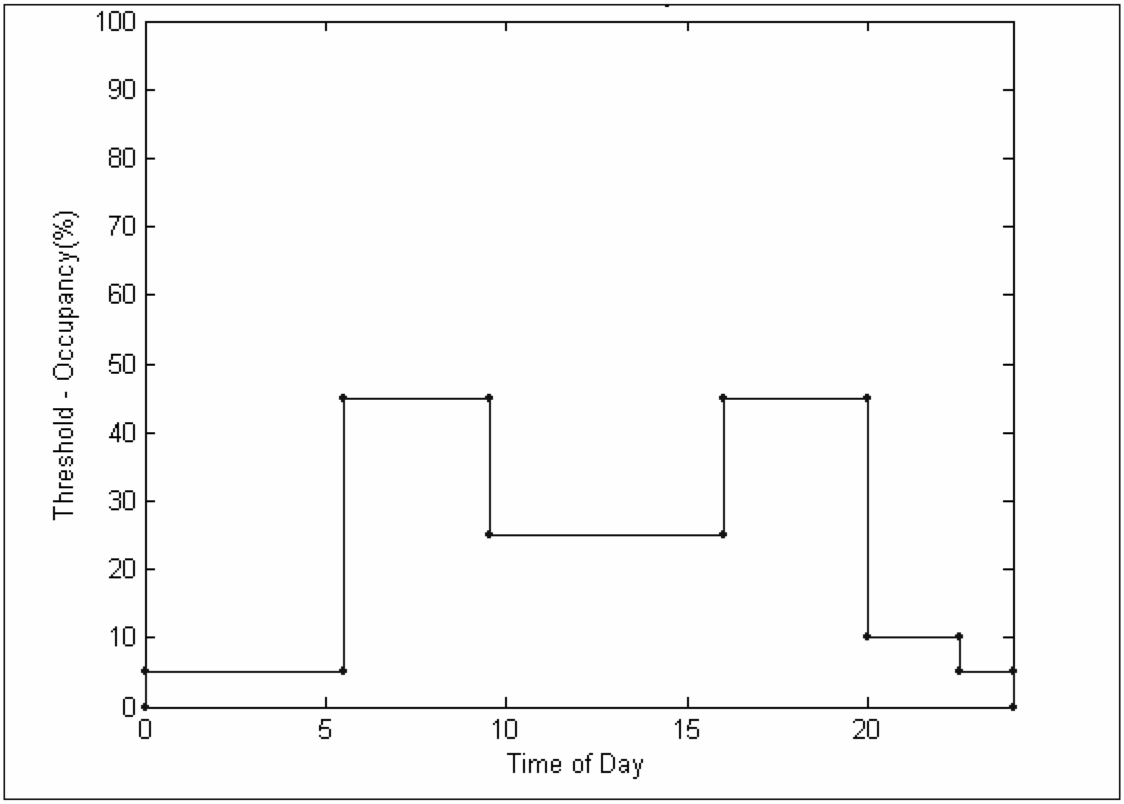


Figure 4. Typical Threshold Graph for TxDOT Incident Detection Algorithm.

Thresholds can be set at any level of occupancy, and time intervals can be established at any point throughout the 24 hours of a day. The 24 hours of a typical profile start of 12:00 AM and proceed to 11:59 PM. The only limitation is that there can only be six non-overlapping time intervals in a profile. Multiple profiles exist to account for situations such as weekdays (Monday through Friday), weekends (Saturday through Sunday), special events and inclement weather.

The inclement weather and special events profiles typically would take place for less than 24 hours and would be specifically designated by an operator to be in effect.

It should also be noted that each station is independent of all other stations in terms of the profiles. This allows implementations to tailor the detection capabilities by location. In the Austin implementation, because stationing is done on a per-lane basis, it is possible for each lane in the cross section to have a unique profile. In practice however, the same profile is kept for all lanes in the cross section.

ESTABLISHING THRESHOLDS

The key to optimizing performance of the algorithm is determining the thresholds and adjusting them so that there is an effective compromise between the parameters of detection time, detection rate, and false alarm rate. However, a procedure to accomplish this for the TxDOT incident detection algorithm has never been developed

Manual Method

Currently, the procedure used in the Austin District to establish thresholds is labor intensive and manual. The basic steps are described below:

1. Pick a weekday or weekend (depending on threshold being established) that is representative of normal conditions for the station.
2. Once the day is determined, assemble a data archive for analysis. The final data archive will contain data for a 24-hour time period. The archive is created from the comma delimited ASCII file from the SCU that contain 1-minute data for one hour organized by station (in Austin's case, each station is a detector).
3. Merge individual 1-hour data files ASCII editor to complete 24 hours of data. This file is imported into Microsoft Excel ™. Because Excel will only input 256 columns of data at a time, the Austin district developed macros to import successive groups of 256 columns of data.
4. Once the data is in Excel, create charts to show the 24-hour occupancy data for each station.
5. Review the occupancy charts for each station and manually determine the time periods corresponding to significant pattern changes in the occupancy.

6. For each time period, establish a threshold occupancy level.
7. Compile the profile for each station and enter it into the ATMS

Automated Method

Because of the difficult and time-consuming nature of the manual process above, the Austin District contracted with researchers at TTI to create a more automated methodology for establishing thresholds. This was done via an interagency contract.

The methodology created under that work utilized concepts of time-series analysis and operator workload to minimize the number of false alarms. Approximately 30 false alarms per station per year was chosen as a target. Figure 5 shows a sample of the thresholds created under this contract. The shaded areas represent individual time points, while the number at the top of each time section corresponds to the threshold value that should be set.

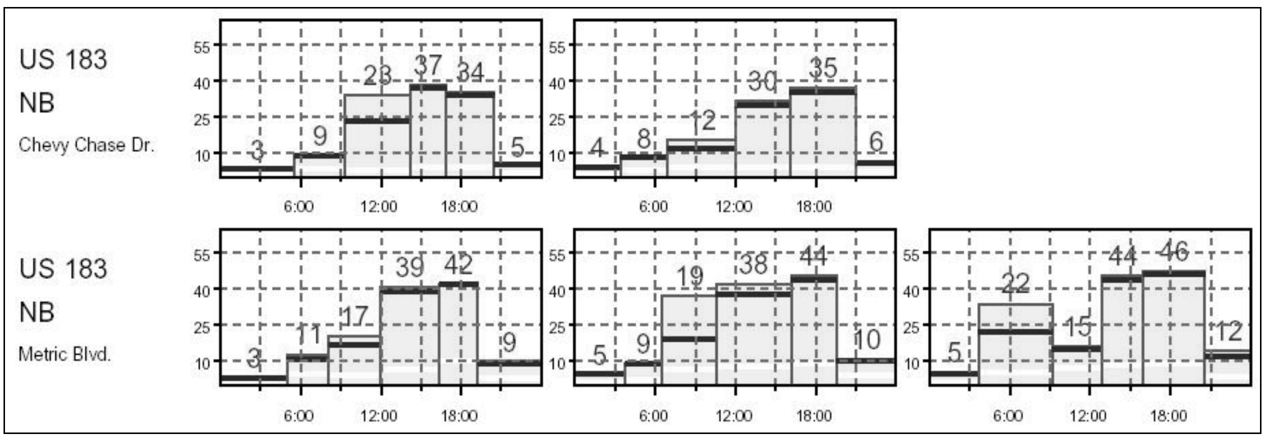


Figure 5. Thresholds Created Under Austin Inter-Agency Contract.

While the thresholds and time periods created using this process were effective, there were a number of considerations associated with their use.

- The data manipulation process was intensive and time consuming.
- The data manipulations could not be performed in a typical desktop computer using standard, off-the-shelf software.
- The data manipulations could not be performed by someone without specialized knowledge and training on the specific software used to create the thresholds.
- The process was developed for Austin and would require modifications to expand to other areas of the state using ATMS.

- The thresholds were very sensitive to increases in traffic, requiring revisions whenever substantial traffic growth or changes took place.

CHAPTER 4 –PRELIMINARY DATA MANIPULATION

INTRODUCTION

The data logging structure in use by ATMS and the SCU are not conducive to easy use for other purposes. Significant effort is necessary to prepare the data for use. This chapter describes the data manipulations utilized to prepare data in three areas:

- operator's logs,
- detector thresholds, and
- SCU data files.

Each of the following sections describes in detail the steps in the data manipulation process. Each section below uses a standardized format consisting of:

- MatLab File(s) – the names of the MatLab custom programming file utilized in this data manipulation step;
- Input File(s) – the names of the input files utilized in this step;
- Output File(s) – the names of any output files created as a result of this step;
- Procedure – a summary of the data manipulation procedures accomplished in this step of the process; and
- File Descriptions – a brief summary of the makeup of the output file.

Where appropriate, graphics are included to supplement the text discussion and illustrate the data manipulation process.

OPERATOR'S LOGS

The goal of any incident detection algorithm is of course to detect incidents. Operators at ATMS implementations record incident information that is stored in the ATMS database. This data is copied in to an Excel worksheet format which can then be reduced to extract incidents of specific or known types.

Step 1: Incident Log Reduction (Manual)

MatLab File: N/A

Input Files: Incident Files

Output Files: IncidentLog.txt

Procedure: Extract the following fields from each incident log file, and save them to a new output file, called IncidentLog.txt.

- Nearest Station(s)
- Archive Number
- Incident Number
- Direction
- Roadway
- Detection Data Start Time
- Detection Data End Time

File Descriptions:

Incident File: Figure 6 shows a portion of the incident file. This is the raw data received from the operators at an ATMS implementation and the input file for this step of the data manipulation process. The file contains numerous items of information pertaining to each incident including roadway and cross-street location.

1	archive_number	ident_num	direction_str	roadway_str	roadway_	tion_descri	cross_street_str
2	5542	7	Southbound	IH 0035	-10		Rundberg Lane
3	5543	4	Northbound	IH 0035	-10		Cesar Chavez St /
4	5544	17	Southbound	IH 0035	-10		51st Street
5	5577	19	Northbound	IH 0035	-10		32nd Street
6	5589	1	Northbound	IH 0035	-10		Rundberg Lane
7	5593	1	Northbound	IH 0035	-10		US 183 NB / Anderson
8	5594	44	Northbound	IH 0035	-10		51st Street
9	5612	4	Northbound	IH 0035	-10		US 183/Anderson Lane
10	5614	21	Southbound	IH 0035	-10		Rundberg Lane
11	5615	20	Northbound	IH 0035	-10		Rundberg Lane
12	5643	31	Southbound	IH 0035	-10		51st Street
13	5644	33	Northbound	IH 0035	-10		Braker Lane
14	5668	25	Northbound	IH 0035	-10		US 183/Anderson Lane

Figure 6. Input Incident File.

Incident Log: Figure 7 shows a portion of the output file for this step in the data manipulation process. The process, which is completely manual, extracts pertinent information from the input incident file. In addition, the file contains the two closest detector stations for use in subsequent steps related to incident testing.

It should be noted that the incident log contains incidents of all types and durations. For the purposes of this research, the concern was for major incidents that significantly affect conditions. For that reason, the operator's log was also pared down during this step to reduce the incidents only to those listed as a "collision".

29,25	5466	61	Northbound	US 0183	12/10/2003 8:43:00 PM	12/10/2003 9:02:00 PM
31,36	4495	82	Southbound	US 0183	2/26/2003 4:36:00 PM	2/26/2003 4:46:00 PM
31,36	4706	8	Southbound	US 0183	6/10/2003 12:53:00 PM	6/10/2003 1:45:00 PM
33,29	4670	14	Northbound	US 0183	5/22/2003 4:02:00 PM	5/22/2003 4:27:00 PM
40,33	4694	3	Northbound	US 0183	6/3/2003 5:52:00 AM	6/3/2003 7:47:00 AM
42,50	4478	36	Southbound	US 0183	2/20/2003 9:57:00 AM	2/20/2003 10:33:00 AM
42,50	4516	5	Southbound	US 0183	3/6/2003 2:31:00 PM	3/6/2003 2:50:00 PM
42,50	4617	1	Southbound	US 0183	4/28/2003 9:58:00 AM	4/29/2003 5:59:00 AM
50,53	4325	3	Southbound	US 0183	1/2/2003 3:12:00 PM	1/2/2003 4:07:00 PM
50,53	4664	3	Southbound	US 0183	5/15/2003 11:53:00 AM	5/15/2003 12:12:00 PM
57,60	5480	23	Southbound	US 0183	12/19/2003 7:00:00 PM	12/19/2003 7:30:00 PM
57,60	5479	24	Southbound	US 0183	12/19/2003 7:01:00 PM	12/19/2003 7:26:00 PM
57,60	5481	22	Southbound	US 0183	12/19/2003 6:45:00 PM	12/19/2003 7:31:00 PM
59,55	4460	5	Northbound	US 0183	2/12/2003 9:56:00 AM	2/12/2003 10:58:00 AM
64,69	4679	1	Southbound	US 0183	5/29/2003 10:21:00 AM	5/29/2003 11:41:00 AM
84,87	4351	27	Southbound	US 0183	1/15/2003 4:00:00 PM	1/15/2003 5:02:00 PM
84,87	5238	43	Southbound	US 0183	9/23/2003 8:30:00 AM	9/23/2003 9:29:00 AM
101,105	4487	8	Southbound	US 0183	2/25/2003 9:15:00 AM	2/25/2003 10:04:00 AM
101,105	4491	2	Southbound	US 0183	2/26/2003 8:19:00 AM	2/26/2003 9:18:00 AM
101,105	4594	5	Southbound	US 0183	4/14/2003 7:58:00 AM	4/14/2003 8:46:00 AM

Figure 7: Output Incident Log File.

Step 2: Reduce Log File

MatLab File: ReduceLog.m

Input Files: IncidentLog.txt

Output Files: IncidentLog_Sum2_Up.txt

Procedure: This step takes the incident file created in the initial step and reduces the incidents into the correct format to be used in later steps.

File Descriptions:

IncidentLog_Sum2_Up.txt:

This step takes the output file from step 1 and formats the data for use in subsequent steps of the data manipulation and analysis process. The primary purpose of this step is to parse the time stamp information from the incidents and separate into years, months, days, hours, and minutes, each in a separate column. Figure 8 shows an example of the output file.

Nearest Sta.	Arc. No.	Inc. No	Start Time (YY, MM, DD, HH, Min)					End Time (YY, MM, DD, HH, Min)				
29,25	5466	61	2003	12	10	20	43	2003	12	10	21	2
31,36	4495	82	2003	2	26	16	36	2003	2	26	16	46
31,36	4706	8	2003	6	10	12	53	2003	6	10	13	45
33,29	4670	14	2003	5	22	16	2	2003	5	22	16	27
40,33	4694	3	2003	6	3	5	52	2003	6	3	7	47
42,50	4478	36	2003	2	20	9	57	2003	2	20	10	33
42,50	4516	5	2003	3	6	14	31	2003	3	6	14	50
42,50	4617	1	2003	4	28	9	58	2003	4	29	5	59
50,53	4325	3	2003	1	2	15	12	2003	1	2	16	7
50,53	4664	3	2003	5	15	11	53	2003	5	15	12	12
57,60	5480	23	2003	12	19	19	0	2003	12	19	19	30
57,60	5479	24	2003	12	19	19	1	2003	12	19	19	26
57,60	5481	22	2003	12	19	18	45	2003	12	19	19	31
59,55	4460	5	2003	2	12	9	56	2003	2	12	10	58
64,69	4679	1	2003	5	29	10	21	2003	5	29	11	41
84,87	4351	27	2003	1	15	16	0	2003	1	15	17	2
84,87	5238	43	2003	9	23	8	30	2003	9	23	9	29
101,105	4487	8	2003	2	25	9	15	2003	2	25	10	4
101,105	4491	2	2003	2	26	8	19	2003	2	26	9	18
101,105	4594	5	2003	4	14	7	58	2003	4	14	8	46

Figure 8. Incident Summary File.

DETECTOR THRESHOLDS

One of the keys to evaluating performance of any algorithm is to know the existing threshold for the detector locations. Researchers were provided with an Excel worksheet that comtained the threshold information for the Austin district. This is a 1-step process.

Step 1: Threshold Data Files Creation (Manual)

MatLab File: N/A

Input Files: Threshold Spreadsheet
Detector List

Output Files: Daily.txt
Row.txt
Inventory.txt

Procedure: The threshold spreadsheet contains three worksheets. These worksheets contain descriptions of the thresholds in a format that can be read by a database program and imported into the ATMS system. For the purposes of this research, the threshold information was extracted to text files for use in subsequent manipulation and analysis steps.

The inventory file is then created by listing each detector for the corresponding roadway, along with the detectors station, direction, and lane. This information is found from the list of detectors in Austin.

File Descriptions:

Inventory.txt Contains the number, station, direction, and lane for each detector; Figure 9 shows a portion of the file. The file contains the detector identification number, the station, the direction, and the lane number.

Detector	Station	Direction	Lane
2000411	1	NB	1
2000412	1	NB	2
2000413	1	NB	3
2000415	2	NB	1
2000421	3	SB	1
2000422	3	SB	2
2000423	3	SB	3
2000427	4	SB	1
2000511	5	NB	1
2000512	5	NB	2
2000513	5	NB	3
2000515	6	NB	1
2000521	7	SB	1
2000522	7	SB	2
2000523	7	SB	3
2000527	8	SB	1
2001011	9	NB	1

Figure 9. Detector Inventory Text File.

Daily.txt Contains the information relating to what profile is currently being used for each detector. Figure 10 illustrates the file, which contains a link between the detector identification number and the particular profile that will be used at that location.

DetID	Day	Profile	Status
1512	2	151	3
1513	2	151	3
1514	2	151	3
1521	2	152	3
1522	2	152	3
1523	2	152	3
1524	2	152	3
4711	2	471	3
4712	2	471	3
4713	2	471	3

Figure 10. Detector Profile Text File.

Row.txt Contains the time points and occupancy levels for the current profiles being used by detectors. The correspondence between the 'daily.txt' file and the 'row.txt' file is the profile number. For each profile, the corresponding start time (in hours and minutes), end time, and occupancy threshold is identified.

Profile Row	Profile	StartHour	StartMin	EndHour	EndMin	Occupancy
1	151	0	1	5	45	45
1	152	0	1	6	15	45
1	471	0	1	5	45	45
1	472	0	1	5	30	45
1	474	0	1	5	30	5
1	475	0	1	5	30	10
1	476	0	1	5	30	5
1	477	0	1	5	30	10

Figure 11. Profile Information File.

SCU DATA FILES

The main thrust of the research performed during this project was to develop a calibration process for the TxDOT incident detection algorithm. This objective required, at minimum, the use of a 24-hour data file per detector. However, because this is not the manner in which TxDOT stores data, creating those files required an extensive amount of data manipulation.

The TxDOT system records data in 1-hour files, so an assembly of 24 files is necessary to create a single day's data. In addition, because each file contains all detectors connected to a particular SCU, significant data manipulation has to be performed in order to assemble individual files per detector. The vast majority of the data manipulation accomplished in this research was performed using MatLab, a statisitical analysis package that allows for custom programming and data manipulation functions.

Step 1: Uncompress Raw Data Files from .ZIP Files

MatLab File: Not Applicable

Input Files: 030101.ZIP
….
031231.ZIP

Output Files: US 0183 SCU Wednesday 0000.DET
….
US 0183 SCU Wednesday 2300.DET

Procedure: The TxDOT ATMS system stores data files from the SCU using a filename structure that will be overwritten in 1 week. Typically, operators copy the data from the SCU and create an off-line data repository. For each day of the year, researchers were provided with a compressed file (in

ZIP format) containing the 24 individual 1-hour data files from the SCU. Typically, an SCU is utilized per roadway. The roadway name is not included in the ZIP file name.

This step is a manual procedure where each ZIP file is copied to a directory unique for an individual day on a particular roadway and expanded to uncompress the 24 individual 1-hour data files.

File Descriptions:

The file naming convention followed by the operators is to name the file by YEAR, MONTH, DAY, with no spaces in between. Thus, a filename of 030101.ZIP represents data from January 1, 2003.

The 24 files inside the ZIP file have the following typical format:

US 0183 SCU Wednesday 2300.DET

Variable items are the roadway name (US 0183, LP 0001 and IH 0035) and the day of the week (Monday through Sunday)

Figure 12 shows a graphic of the results of this step in the data manipulation process. The directory is entitled 030301 which equates to March 1, 2003. The files in the directory are for Loop 1 and correspond to 24 hours of data.

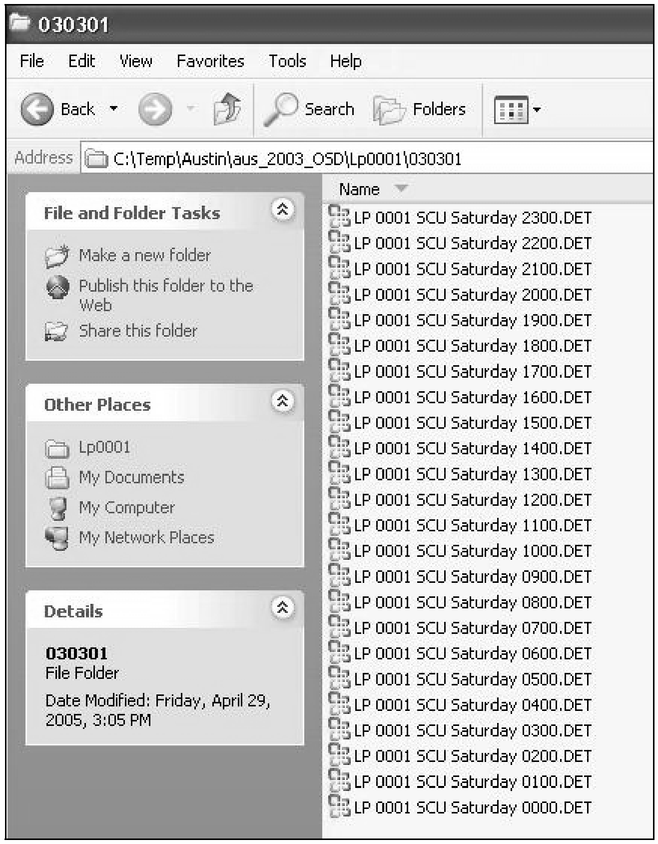


Figure 12. Directory and Filename Structure Prior to Renaming.

Step 2: Create Unique File Names for Every Data File

MatLab File: Not Applicable

Input Files: US 0183 SCU Wednesday 0000.DET
….
US 0183 SCU Wednesday 2300.DET (example)

The roadway name portion of the filename can vary as follows:
US 0183
LP 0001
IH 0035

The day of the week portion of the filename varies from Monday through Sunday.

Output Files: US 0183 SCU 030101 0000.DET
….
US 0183 SCU 030101 2300.DET

Procedure: With the data files now sorted into individual directories according to the YEAR, MONTH, and DAY, the 24 data files have the exact same name as the 24 data files in any other directory. Each data file in each directory must be renamed to include the YEAR, MONTH, and DAY so that there are unique data files for any hour of the day, any day of the year, across any roadway.

This manual step uses a utility file renaming program to replace the day of week portion of the filename with YEAR, MONTH, DAY equivalent. For example, since January 1, 2003, was a Wednesday, the 'Wednesday' in each of the 24 filenames in the directory would be replaced with '030101'..

File Descriptions:

This step creates 8,760 unique filenames for each roadway. Each data file represents one hour of data from the SCU.

Figure 13 shows the results of this step in the data manipulation process. The directory is entitled IH35_Raw_Data and contains a full year of 1-hour data with unique filenames.

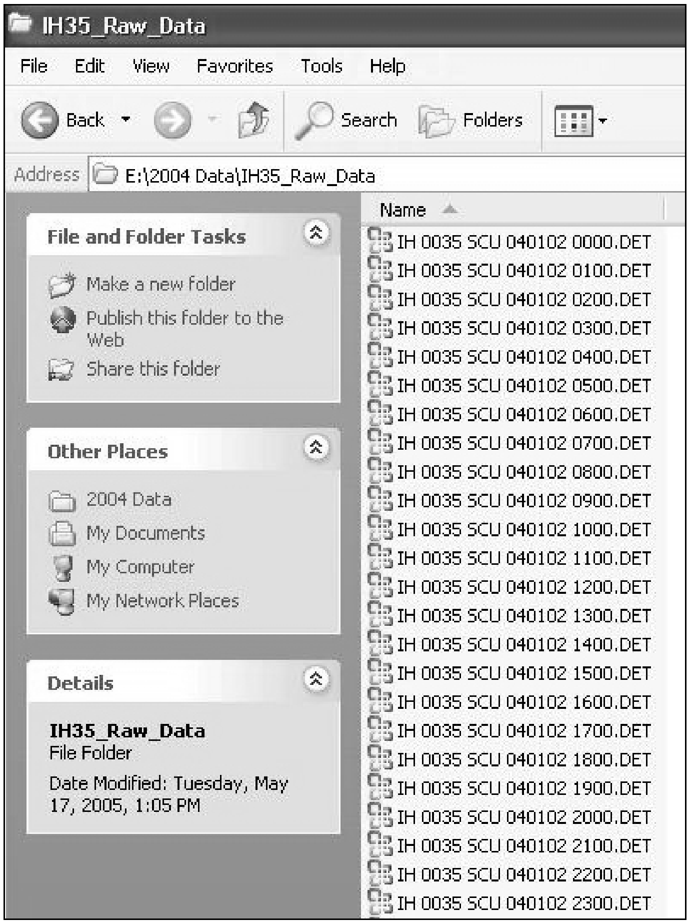


Figure 13. Directory and Filename Structure After Renaming Process.

Step 3: Create Individual Detector Data Files

MatLab File: Reduce1.m

Input Files: US 0183 SCU 030101 0000.DET
….
US 0183 SCU 031231 2300.DET

Output Files: US 1083 SCUall_*DETID*.txt

Procedure: In order to work with the raw data provided in the DET files the data must be separated by detector. Each DET file contains one hour of data, separated into sixty 1-minute intervals. Each interval contains occupancy, volume, speed, and truck percentage data for each detector on the roadway.

This step reads these intervals one at a time and separates the data by detector. This data is then written to a separate file for each detector. This is repeated for every interval in each DET file for the entire year. This creates a text file for each detector, containing the volume, occupancy, and speed data for that detector.

File Descriptions:

Input File: The first line of the input file contains the total number of detectors in the file and the detector identification number, cross street location, and freeway lane description for each detector. This information allows the MatLab program to parse the comma delimited data and construct individual data files for each detector number.

The remaining lines each contain 1-minute of data for each detector. The following format is utilized:

- time in hour, minute, second format (HHMMSS);
- detector identification number,
- volume in number of cars,
- occupancy in percent,
- speed in miles per hour, and
- portion of trucks in the vehicle stream in percent.

This information repeats for each detector reporting in the 1-minute timeframe.

Figure 14 shows an example of the input detector file.

```
258,   2000411,Guadalupe St.  F1              ,   2000412,Guadalupe St.  F2
000047,2000411,1,0,68,0,2000412,3,1,66,0,2000413,3,1,62,33,2000415,1,0,43,0,2000421
000147,2000411,1,0,60,0,2000412,3,1,64,0,2000413,3,1,60,33,2000415,2,0,50,0,2000421
000247,2000411,4,1,65,25,2000412,3,1,59,0,2000413,5,2,55,0,2000415,0,0,0,0,2000421,
000347,2000411,3,1,78,0,2000412,3,1,61,0,2000413,4,2,57,0,2000415,2,1,55,0,2000421,
000447,2000411,3,1,72,0,2000412,1,0,68,0,2000413,2,1,64,50,2000415,2,1,54,0,2000421
000547,2000411,1,0,77,0,2000412,0,0,0,0,2000413,6,3,59,33,2000415,0,0,0,0,2000421,1
000647,2000411,1,0,77,0,2000412,1,0,57,0,2000413,5,3,62,20,2000415,2,0,53,0,2000421
000747,2000411,2,0,70,0,2000412,3,1,66,0,2000413,3,1,62,33,2000415,4,2,52,0,2000421
000847,2000411,4,2,66,0,2000412,6,2,61,0,2000413,5,2,58,40,2000415,0,0,0,0,2000421,
000947,2000411,1,0,68,0,2000412,4,1,57,0,2000413,4,1,66,0,2000415,2,1,42,0,2000421,
001047,2000411,2,0,70,0,2000412,3,1,64,0,2000413,5,2,67,20,2000415,2,0,51,0,2000421
001147,2000411,1,0,72,0,2000412,4,1,71,0,2000413,5,2,64,20,2000415,2,0,70,0,2000421
001247,2000411,1,0,60,0,2000412,3,1,63,0,2000413,2,1,62,0,2000415,5,2,48,0,2000421,
```

Figure 14. Input File for Step 3 of SCU Data Manipulation.

Output File : SCUall_*DETID*.txt: The resulting file contains the volume, occupancy, and speed data for an entire year, separated into 1-minute intervals. Each detector has a unique file. Figure 15 shows an example of one of the detector output files. The first column is the time period, expressed as MONTH, DAY, HOUR, MINUTE (MMDDHHMM). The next three columns represent the Volume, Occupancy, and Speed values. The percent truck values were not written to the detector file as they were not required for the data analysis.

MMDDHHMM	Vol	Occ	Spd
01010000	0	36	14
01010001	0	36	14
01010002	0	36	14
01010003	0	36	14
01010004	0	36	14
01010005	0	36	14
01010006	0	36	14

Figure 15. Output File for Step 3 of SCU Data Manipulation.

Step 4: Make 365 Day Matrix

MatLab File: Make365daysMatrix2.m

Input Files: IncidentLog.txt
Inventory.txt
SCUall_XXX.txt

Output Files: DET*XXX*_OCC_STA*YYY*.txt
DET*XXX*_VOL_STA*YYY*.txt
DET*XXX*_SPD_STA*YYY*.txt

Where *XXX* is detector number, *YYY* is station number

Procedure: This step in the process takes the data compiled in the previous step and formats it into a 365 x 1440 matrix. A separate data file is created for the speed, volume, and occupancy information for each detector. This ensures that the data files for use in later steps are a small as possible and contain only the necessary information. The matrix size of 365 by 1440 represents 365 days in a year and 1440 minutes in a day.

File Descriptions:

Output Files: DET*XXX*_OCC_STA*YYY*.txt
This file contains a 365x1440 matrix of occupancy values for the specified detector. The same format is used to create a speed and volume file as well. *XXX* is the detector number, and *YYY* is the station number. Figure 16 shows a portion of the output file for this step. The first column contains the minutes of the day, from 1 to 1440. Each subsequent column contains 1-minute detector occupancy values for one day of the year.

1	-2	0	2	1	0	-3	-3	-3	0	0	2	1	0	0	0	0	0	2	0	1	1	1	0	1	1
2	-2	2	0	0	1	-3	-3	-3	0	2	0	0	0	0	0	0	2	1	0	2	2	0	2	1	1
3	-2	0	1	1	0	-3	-3	-3	0	1	1	1	0	0	1	0	1	2	2	1	0	0	0	1	1
4	-2	1	2	2	0	-3	-3	-3	1	2	0	1	1	2	1	1	2	1	0	0	0	1	1	0	1
5	-2	0	3	0	1	-3	-3	-3	0	1	0	0	1	0	0	0	1	1	0	1	1	1	0	0	0
6	-2	0	3	1	1	-3	-3	-3	0	1	0	0	0	1	0	1	1	2	0	0	0	1	0	1	1
7	-2	0	1	0	0	-3	-3	-3	0	2	1	0	1	1	0	0	2	2	0	0	0	0	1	1	0
8	-2	0	1	1	1	-3	-3	-3	1	0	0	1	1	1	1	0	0	1	2	0	0	0	0	2	1
9	-2	1	1	1	0	-3	-3	-3	1	1	0	0	0	0	0	1	1	1	0	1	1	1	0	2	0
10	-2	0	1	2	2	-3	-3	-3	0	0	0	0	2	0	1	2	0	3	1	0	0	0	1	0	2
11	-2	0	0	3	1	-3	-3	-3	0	1	0	0	1	1	0	0	1	0	2	0	0	0	1	0	0
12	-2	0	2	3	0	-3	-3	-3	1	1	3	0	0	0	0	1	1	1	2	0	0	1	0	1	2
13	-2	1	2	2	0	-3	-3	-3	0	1	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0
14	-2	0	1	1	0	-3	-3	-3	1	2	1	0	0	1	0	0	2	1	1	0	0	1	0	1	0
15	-2	0	2	0	0	-3	-3	-3	0	0	1	0	0	1	1	0	0	1	0	0	0	1	1	2	0
16	-2	1	0	2	1	-3	-3	-3	1	0	1	0	0	0	0	0	0	2	0	1	0	0	1	0	0
17	-2	0	0	0	0	-3	-3	-3	0	2	1	0	0	0	0	0	2	1	0	1	1	1	0	2	0
18	-2	1	1	2	0	-3	-3	-3	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0
19	-2	2	1	1	0	-3	-3	-3	1	1	1	0	1	0	0	1	1	0	1	0	0	2	1	2	0
20	-2	0	0	1	1	-3	-3	-3	1	0	2	1	0	1	0	0	0	3	0	0	0	1	1	1	0
21	-2	0	0	0	1	-3	-3	-3	0	2	2	0	0	1	0	0	2	1	0	0	0	0	0	2	0
22	-2	1	0	2	1	-3	-3	-3	1	1	0	1	1	0	0	0	1	1	1	0	1	0	0	2	3
23	-2	0	1	2	1	-3	-3	-3	1	1	0	0	1	1	1	0	1	1	0	1	2	1	0	0	0
24	-2	0	1	1	0	-3	-3	-3	1	2	0	0	0	1	0	0	2	1	0	0	0	0	0	2	3
25	-2	0	0	1	0	-3	-3	-3	0	0	1	0	1	0	1	0	0	1	0	0	1	2	0	0	3
26	-2	0	1	1	0	-3	-3	-3	1	1	2	1	0	0	0	1	1	0	0	1	0	1	0	0	0
27	-2	0	1	0	1	-3	-3	-3	0	3	1	0	1	1	0	1	3	3	2	0	0	1	1	0	0
28	-2	0	1	2	0	-3	-3	-3	0	1	0	0	0	1	1	0	1	1	0	1	0	0	0	2	0
29	-2	0	1	0	1	-3	-3	-3	0	1	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0
30	-2	0	0	2	0	-3	-3	-3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
31	-2	0	0	2	1	-3	-3	-3	0	1	1	1	1	0	0	1	1	2	0	0	0	0	1	1	1
32	-2	1	0	1	0	-3	-3	-3	1	0	0	0	0	0	1	0	0	3	0	0	0	0	0	0	0
33	-2	0	0	0	0	-3	-3	-3	0	1	0	0	0	0	0	1	1	2	0	0	0	1	1	1	0
34	-2	0	0	0	0	-3	-3	-3	1	1	0	0	0	0	0	0	1	1	0	0	1	0	1	1	1
35	-2	0	1	1	0	-3	-3	-3	0	1	1	0	0	0	0	0	1	1	1	0	0	0	3	1	1
36	-2	0	1	0	0	-3	-3	-3	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0
37	-2	0	0	1	0	-3	-3	-3	0	0	2	0	0	1	1	1	0	1	0	1	0	2	1	0	1
38	-2	0	0	2	0	-3	-3	-3	0	0	2	1	0	0	1	0	0	1	0	1	1	0	0	2	1
39	-2	0	1	0	0	-3	-3	-3	0	1	2	0	1	0	0	1	1	1	0	0	0	0	1	0	0

Figure 16. Detector Occupancy File.

AUSTIN DATA

The procedures detailed above result in a comprehensive set of data files that can be used as the basis for any subsequent data analyses. It should be noted that many of these initial steps are manual and time consuming. This is due to the large volume of data as well as the current file formats and structures used by ATMS. Preparing a year's worth of data for a city the size of Austin, Texas, can take several weeks of preparation and computer time. In addition, the size of the data files is significant, resulting in the need for a large amount of storage space to catalog and store the data.

CHAPTER 5 – THE CROSS-LANE COMPARISON ALGORITHM

INTRODUCTION

One of the primary problems with a comparative algorithm is that it is difficult to detect changing conditions with a simple test. Occupancy, for example, rises not only during an incident, but also at several times during the day when traffic tends to increase, such as the AM (morning) and PM (afternoon) peak periods. With a comparison based essentially on three minutes of data in one lane, it can be challenging to tell the difference. After all, an incident detection algorithm simply detects congestion. It does not determine the cause of the congestion.

Because the existing TxDOT algorithm suffered in this regard, a slight change was made to the algorithm in an attempt to improve on its capability to detect congestion related to incidents and not normal congestion conditions. The goal was to create a more discerning detection algorithm for which thresholds could be determined and utilized to improve the overall incident detection performance.

BASIS FOR CROSS-LANE ANALYSIS

The basis for a cross-lane analysis comes from looking at the different traffic reactions to congestion. In normal, recurring congestion, at most locations, traffic tends to be evenly split across the freeway lanes. This causes the occupancy values to be relatively similar across all lanes in the cross-section.

In incident conditions however, traffic tends to leave the affected lane and shift into the other lanes in the cross-section. This causes an occupancy shift between lanes. Researchers theorized that if this shift could be detected, it could form the basis for a new detection mechanism that would be more sensitive to incident congestion. The Austin data was well-suited to this type of analysis as it was already stationed by individual lanes. The standard mechanism for determining the relative difference between values is to subtract the minimum value from the maximum value.

Figure 17 shows the result of this analysis for non-incident conditions. The 24-hour plots of 3-minute rolling average occupancy values for each lane show a normal progression, starting off very small at 12 AM, rising through the morning peak, reaching the highest occupancy in the PM, and slowly decreasing as the evening progresses.

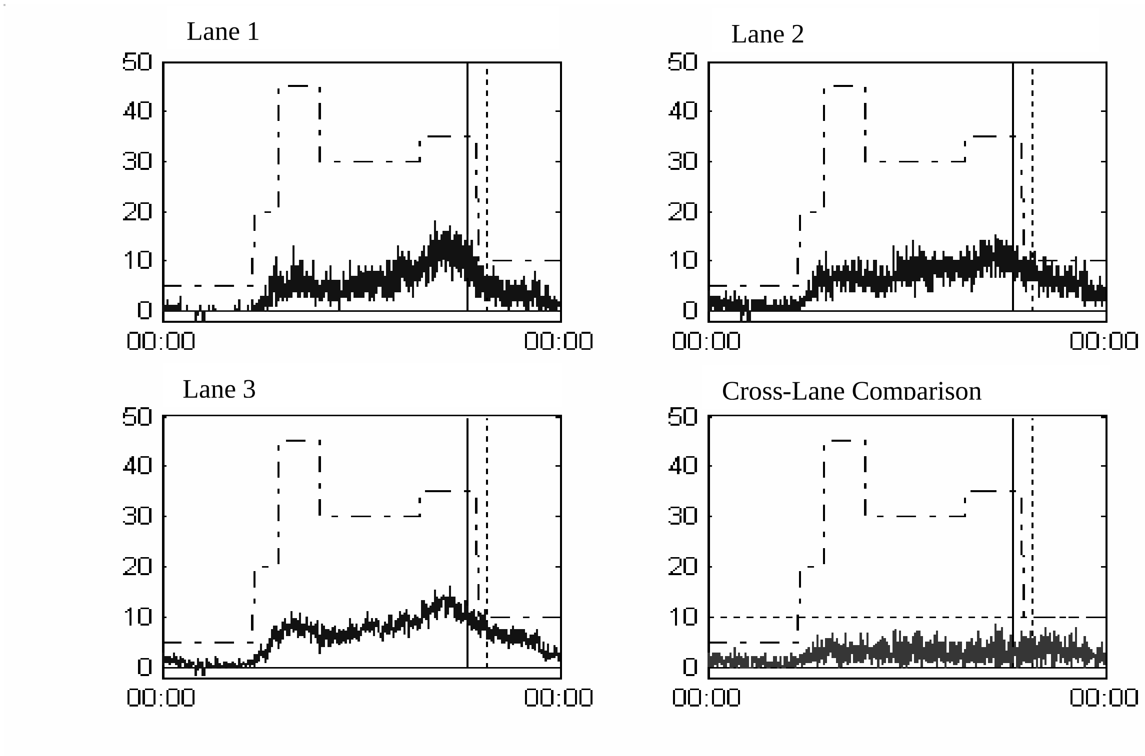


Figure 17. 3-Minute Rolling Average and Cross-Lane Comparison for Non-Incident Conditions.

The dashed lines in each of the lane occupancy graphs are the thresholds in place for the existing TxDOT algorithm. Finally, the series of two vertical lines represent an incident timeframe for comparison against subsequent plots.

The most important portion of Figure 17 is the graph in the bottom right-hand corner. This graph shows the results of subtracting the minimum occupancy values across all lanes in the cross section from the maximum occupancy value across all lanes in the cross section. This is repeated for each minute of the day using the 3-minute rolling average occupancy values. Of particular note is the fact that the graph is relatively flat, showing little movement, even though the occupancy values show significant deviation throughout the day.

Figure 18 provides a direct comparison by examining the corresponding set of graphs for a known incident condition. This incident day was determined from the operator's logs and the data was extracted to create the same graphs as in Figure 17. The incident occurred on Friday, November 1, 2002, at Station 29. As in Figure 17, the dashed lines represent the existing TxDOT thresholds, while the series of two vertical lines represent the actual timeframe of the incident.

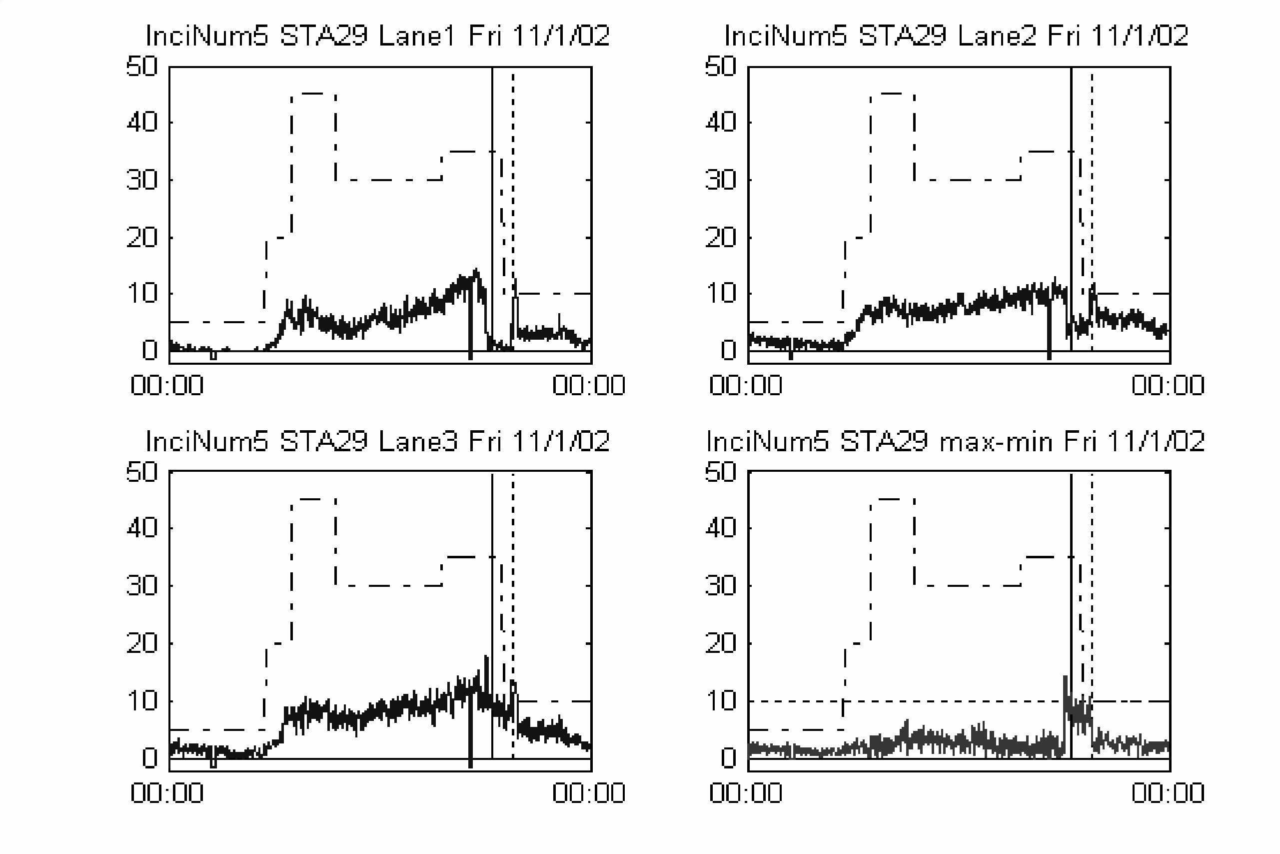


Figure 18. 3-Minute Rolling Average and Cross-Lane Comparison for Incident Conditions.

Of particular note in Figure 18 is the cross-lane comparison graph in the bottom right-hand corner. This graph clearly shows a marked increase in the cross-lane values of loop occupancy during the timeframe of the incident. In addition, this variation has a rapid vertical rise indicating that the detection capability utilizing this mechanism may be timely when compared to the actual occurrence time of the incident.

The CLC value is subject to variability, however. Figure 19 and Figure 20 show a closer view of some representative CLC data. Figure 19 shows the CLC values for the 24-hour period of May 21, 2002, from Station 63 in Austin, Texas, while Figure 20 shows the same data but for Station 67. In each graph, the existing TxDOT thresholds are represented by the dashed-dotted line, while a reference line for the CLC threshold is shown as a dashed line. Of particular note between the figures is the marked difference in CLC variability. Station 63 values remain relatively constant across the entire 24-hour time period, while Station 67 values show a significant rise in CLC variability between the morning and evening peaks.

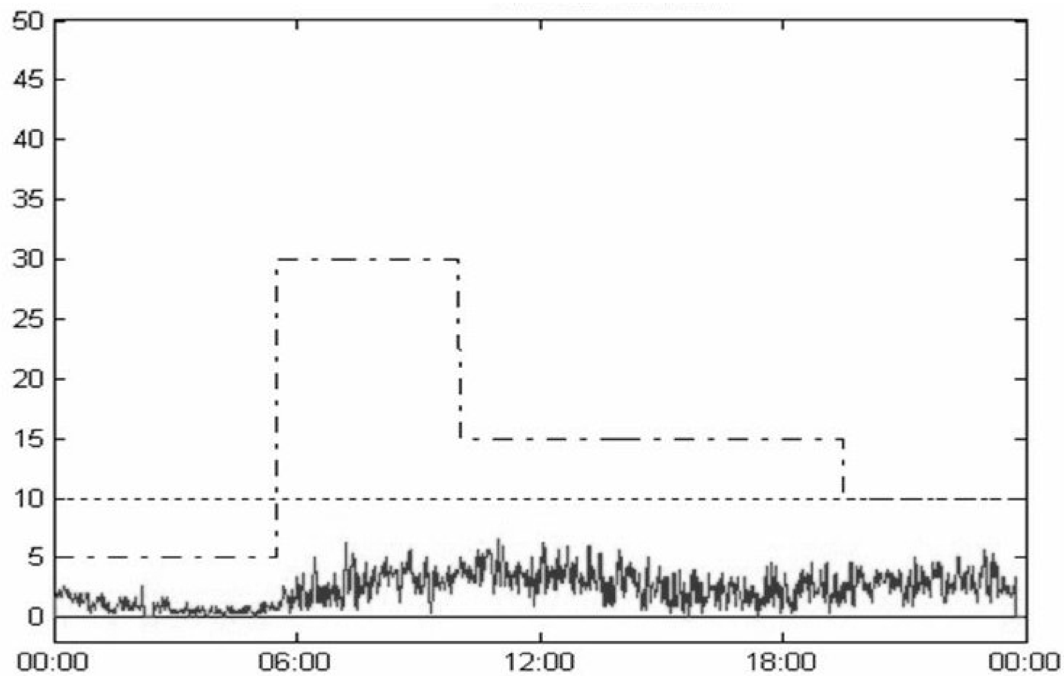


Figure 19. CLC Values for Station 63 - Non-Incident Day.

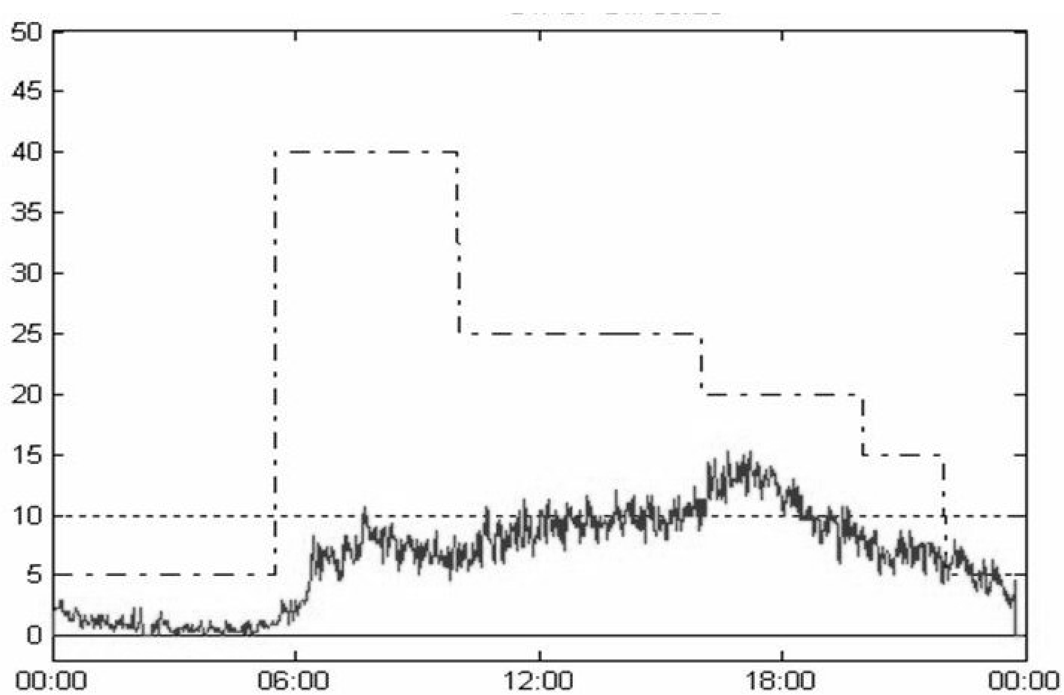


Figure 20. CLC Values for Station 67 - Non-Incident Day.

Although variability exists at different locations, the CLC algorithm can still be used to determine the onset of congestion from incident conditions. Figure 21 and Figure 22 show the same locations as before, but the data now represents an incident day. While Station 63 does not show a dramatic increase in variability during normally congested timeframes, Station 67 does. At both locations, however, the CLC values show a significant spike at the time of the incident, suggesting that the CLC technique for determining congestion due to incident conditions has potential. It should also be noted that the amount of the increase in the CLC at the onset of the incident is different at each location, which reflects the normal variability of traffic across the freeway.

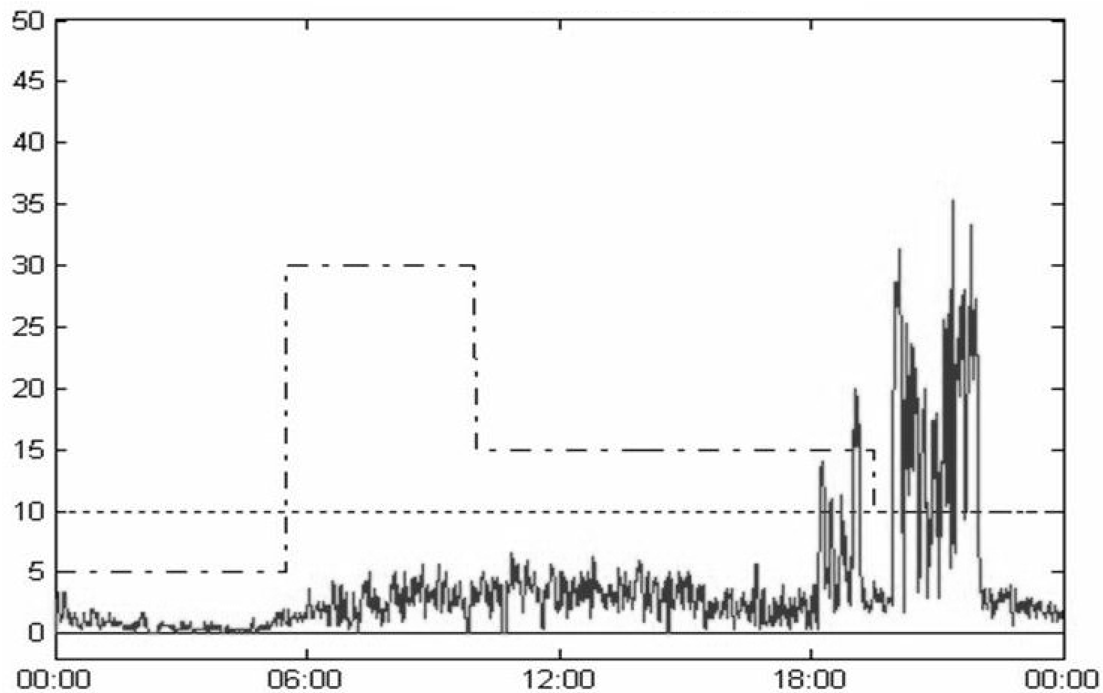


Figure 21. CLC Values for Station 63 - Incident Day.

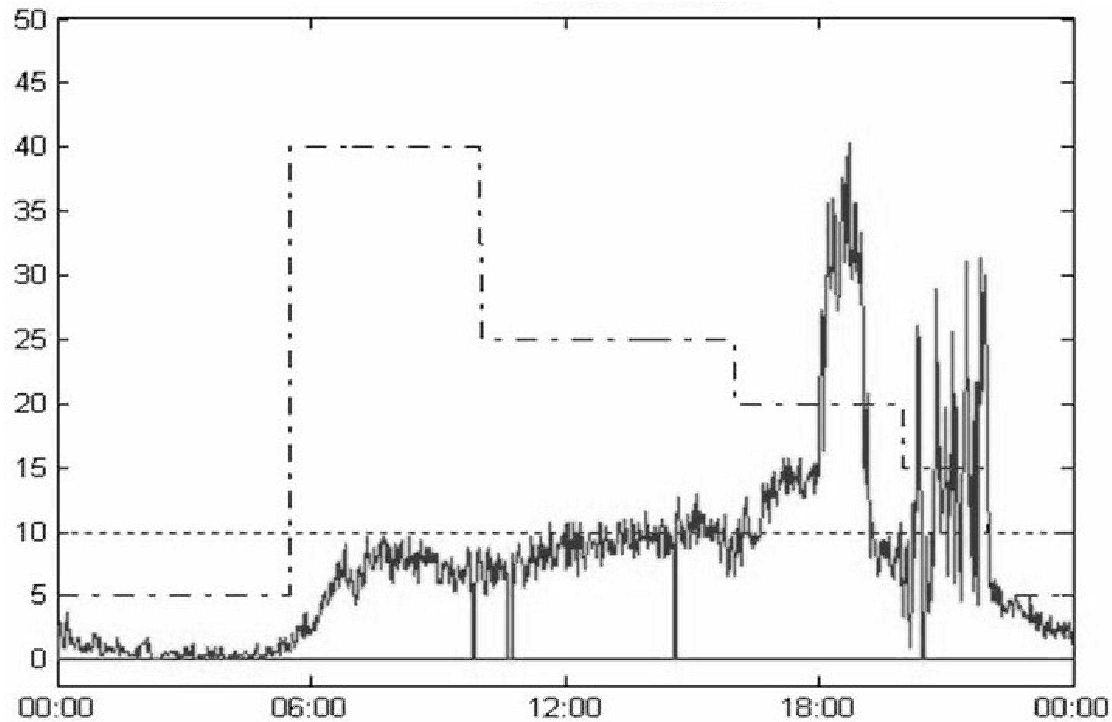


Figure 22. CLC Values for Station 67 - Incident Day.

When this same analysis was performed on all major incidents in the Austin 2002 data, Figure 17 through Figure 22 were found to be representative of the trends in the loop occupancy values. However, this report does not reproduce the charts for every incident in the dataset.

As a result of these analyses and the promising approach to detecting the onset of incident conditions even during periods of congestion, the TxDOT incident detection algorithm was modified to include a cross-lane comparison component prior to comparing it to thresholds.

Figure 23 shows the TxDOT algorithm modified to incorporate a CLC component. As shown in the figure, a step has been inserted after the calculation of the 3-minute rolling average for each lane. This additional step determines the maximum occupancy value, the minimum occupancy value, and the difference between the two values. This difference is then utilized in the traditional threshold comparison step. As per the original algorithm, a value over the threshold will trigger an alarm. A value that does not exceed the threshold will be discarded and the algorithm will return and repeat the process during the next time step.

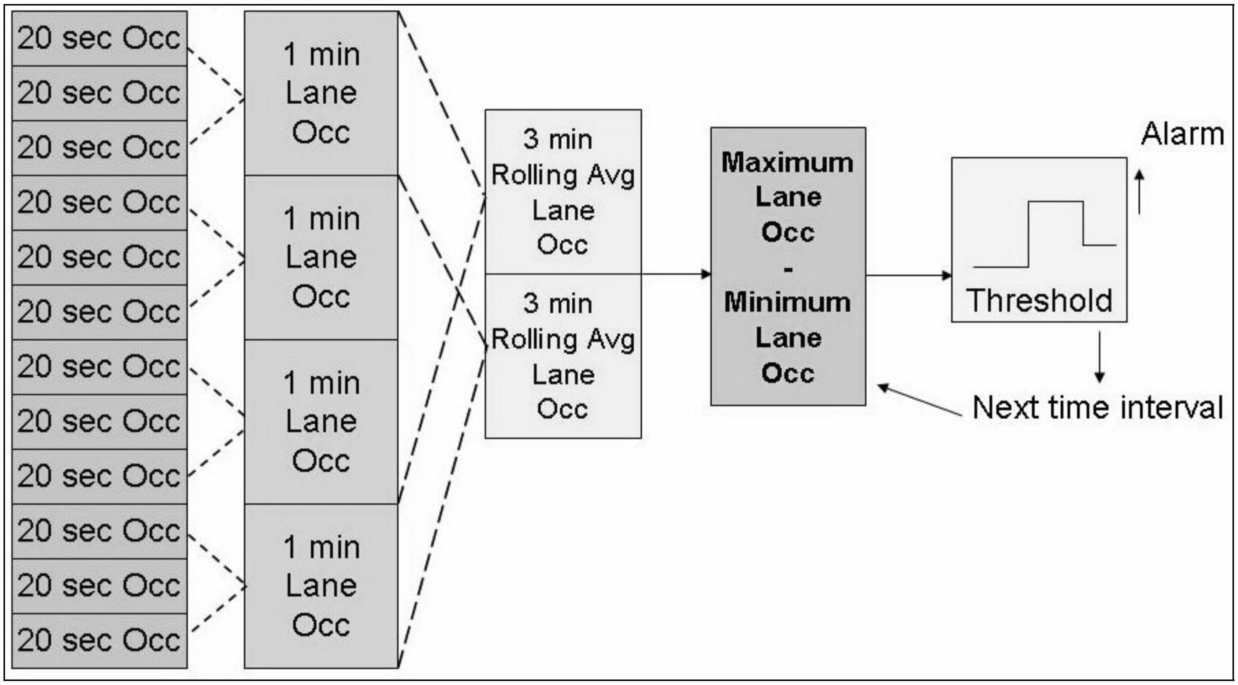


Figure 23. CLC Algorithm.

CLC Thresholds

While the changes to the TxDOT algorithm are relatively minor, the complicated portion of the process is that the concept of thresholds has changed. Whereas thresholds under the original algorithm were a loop occupancy value, they represent a difference in conditions across all lanes in the cross section under the new algorithm. This demands an entirely new and different process for creating the thresholds utilized in this algorithm.

CHAPTER 6 – DETERMINATION OF CROSS-LANE COMPARISON THRESHOLDS

OVERVIEW OF CLC THRESHOLD PROCESS

There are two steps in the determination of thresholds for the CLC algorithm. The first step involves setting the time points where the threshold value will change. While these time points can be determined visually, the challenge is to construct a process that can be automated. The second step in the threshold identification process is setting the actual threshold values, based on the time points. The process of setting thresholds is Deliverable P3 of this project.

ESTABLISHING TIME POINTS

The primary challenge of creating the time points for the CLC algorithm is the preponderance of data. The CLC is calculated on a per-minute basis, which results in 1440 data points per day. In addition, to determine normal conditions, the best technique is to utilize a sufficiently large number of days to account for minor variations in the loop occupancy (and therefore CLC) data. For the purposes of this research, all non-incident days throughout the year were utilized to set CLC time points. Plotting this amount of data for a visual inspection tends to blur any discernable trends and produce inaccuracies in the selection of time points. This problem can be solved by establishing a scientific and statistically sound, data driven approach for establishing time points.

Overview of Time Point Selection Process

An overview of the steps necessary to select time points based on the CLC data are as follows:

1) Determine the CLC at every minute of the day, for all non-incident days. – as an example, if there are 200 non-incident days in the year, there will be 200 CLC values for each minute of the day.
2) Utilize a statistical approach known as Mean Absolute Deviation (MAD) to determine the variability between all the CLC points. This will reduce the number of data points in use and result in a single data point per day.

3) Establish continuity between the data points by finding the difference between the current 1-minute data point and the previous 1-minute data point. This difference is called the 1-minute LAG.
4) Calculate a 30-minute average of the 1-minute LAG values to smooth the variability of the data.
5) Determine the slope between each 30-minute LAG data point.
6) Identify significant differences in slope values by comparing to a cut-off value. This cut-off value establishes slope indicator points that have a value of 1, -1, or 0.
7) At this point in the procedure, time points can be visually chosen from the slope indicator graph. Alternatively, a subsequent section of this report will detail an automated technique for determining the time points based on the slope indicator values.

It should be noted that the procedure listed above is the final result of an intensive and time-consuming research process. The steps in the procedure represent the best results and methods for clearly indicating time points. For example, in addition to the 1-minute LAG values utilized in the procedure, researchers also utilized 2-minute, 5-minute, and 10-minute values during the project. The 1-minute values were found to provide the best definition and original CLC representation, while still reducing the overall need for data.

Data Manipulation for Time Point Selection Process

The data manipulation steps necessary to select time points based on the CLC data are detailed below. The primary output of this data manipulation process is a series of five graphs that show the sequence of data reduction. The final graph contains the slope indicator points which can be used as part of a manual (visual) time point identification process. The description of the process follows the same format utilized in Chapter 4.

MatLab File: 5figures.m

Input Files: Inventory.txt
Daily.txt
Row.txt
IncidentLog_Sum2_Up.txt
DET*XXX*_OCC_STA*YYY*.txt

Output Files: STA_*YYY*_Plots.bmp
newTimeChunkSta*YYY*.txt (manually created file)

Where:

XXX is detector number
YYY is station number

Procedure: After creating a matrix for each detector (Step 4 in SCU Data Files Manipulation from Chapter 4), the data can be analyzed and new threshold time points can be determined.

The process begins by taking a 3-minute rolling average of each detector's occupancy, taking into consideration weekdays where no accidents were reported. Once this is complete the process calculates a Cross-Lane Comparison value for each minute of the day for each station. Figure 24 shows the CLC values calculated in this step.

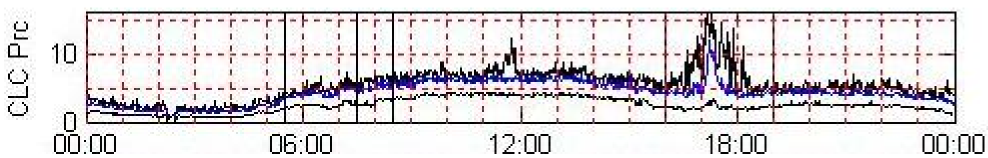


Figure 24. Station CLC Data.

The three lines in Figure 24 represent the 99th, 95th, and 50th percentile CLC values. While the calculation for this step in the process uses the 100th percentile values, the corresponding graph shows three CLC lines to enable end-users to determine the appropriate percentile value for their particular situation.

Once the CLC has been calculated for each minute of the day, the mean absolute deviation (MAD) is determined by the following formula.

$$\frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}|$$

Where:
x_i is the value of CLC at each interval
$\bar{x}$ is the average of the CLC across all time intervals
n is the number of time intervals

The result of the MAD process is a reduction in data points to one per minute for each minute of the day.

The next step in the process is to take the 1-minute LAG of the MAD values, as identified in Step 4 of the overview. The 1-minute LAG is calculated by the following formula.

$$1-\min LAG = [MAD_{(time\ x)} - MAD_{(time\ (x-1))}]$$

Where:

$MAD_{(time\ x)}$ is the mean absolute deviation at a time interval

$MAD_{(time\ (x-1))}$ is the mean absolute deviation at the immediate past time interval

$1-\min LAG$ is the resulting calculation at each time interval

This process establishes continuity between successive data points. Figure 25 shows the graph that results from this step in the process. Note that Figure 25 shows the same trends, valleys, and peaks as Figure 24 but does so with one data point per minute instead of a compilation of all non-incident days in the dataset.

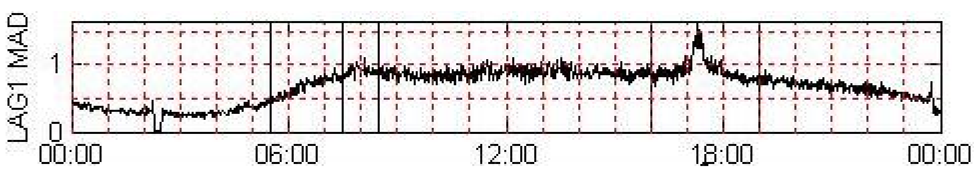


Figure 25. 1-minute LAG of MAD.

The next step in the process is to determine the 30-minute slope value of the 1-minute LAG data points. This step smoothes out the incremental variability in LAG data and makes it easier to calculate the slope values in the next step.

The 30-minute average is calculated according to the following formula:

$$30-\min\ average\ LAG = [(\sum_{time\ x}^{time\ x+30} 1-\min LAG)/30]$$

Where:

$1-\min LAG$ is the 1 minute LAG calculation for each time interval

Figure 26 shows the graphical result of this step. Observe again, how the trends, peaks and valleys of the original CLC plot are reflected in this plot of 30-minute average points.

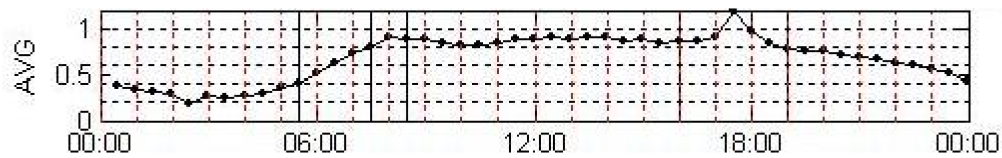


Figure 26. 30-Minute Average LAG.

Although Figure 26 is highly representative of the original CLC values, it does not clearly delineate where time points should be located. In order to provide this delineation, the threshold procedure calculates the slope between each successive point on the 30-minute average LAG graph.

Figure 27 shows the resultant slope graph. At this point in time, the data no longer visually represents the same information as the original CLC graph. Instead, it represents the variability of the CLC data at each 30-minute juncture in a typical 24-hour period.

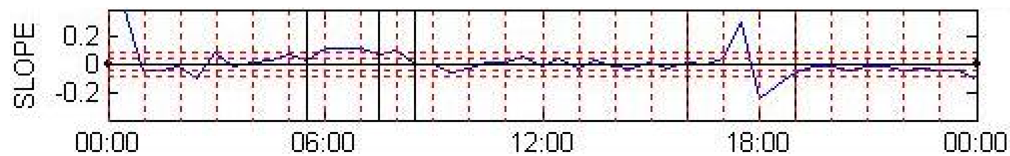


Figure 27. Slope of 30-Minute Average LAG Data.

The average of all the slope points is also calculated during this step in the process. The dashed horizontal lines surrounding the horizontal midpoint axis of 0 identify deviations from the average slope. Each of the first dashed lines surrounding the 0-axis represents one deviation from the average, either positive or negative. Each of the second dashed lines surrounding the 0-axis represents two deviations from the average, either positive or negative.

The final step in the process is identifying the slope indicators on the basis of the deviations from the 0-axis of the slope curve in Figure 27. As shown in Figure 28 the values of this final graph are separated into 30 minute time intervals and have values of -1, 0, or 1 (indicated as INDICIA on the Y-axis). If the slope at the 30-minute point is greater than two times the slope average, the value is 1. If the indicator is less than two times the negative average of the slope, the value is -1. Anything other than 2 times the average, either positive or negative, is classified as a 0, representing no significant deviation from the average slope.

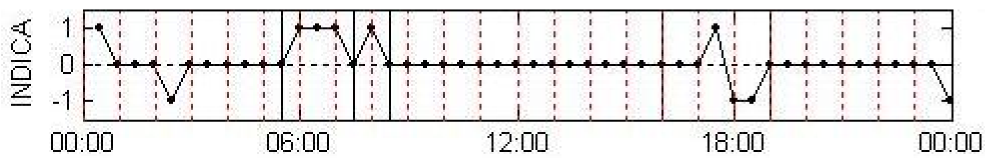


Figure 28. Slope Indicator.

Once this graph is created the user can create the new threshold time chunks for each station.

File Descriptions:

STA_YYY_Plots.bmp:
Graphic containing the 5 graphs created in this step. Each graph shows the progression from CLC data to the slope indicator graph for determining time points for the CLC thresholds.

newTimeChunkStaYYY.txt:
This file contains the new time chunk values from the data produced from this step. This file is manually created by looking at the slope indicator graph (Figure 28) from the bitmap file created for each station. This process is explained in detail in the section below.

Time Period

```
7.5
11.5
12.5
19.5
20.5
```

Figure 29. New Threshold Time Periods.

Manual (Visual) Process for Selecting Time Points

The technique for determining the time points corresponding to changes in the CLC threshold levels are centered on the slope indicator points. Each significant deviation, which is $\pm$ 2 times the average slope, is a potential change point. However, the process is slightly more complicated as there are several factors to consider.

- Because of the mathematics of the technique, the first point of the day on the slope indicator graph will always be a +1. The second point is almost always 0. This does not indicate that there should be a time point established at 12:30AM.

- A jump between -1 and +1 generally does not indicate a time point change. The jump in slope indicator values is simply a measure of variability in the CLC data, so a single time period which surrounds both values will account for the variability.
- A number of short-timeframe jumps between 0 and +1 or -1 generally indicate a rise or fall in the CLC profile. These trends are best captured by surrounding the entire timeframe with a single time period.
- As time periods are visually identified, working across the graph from left to right, a significant number of changes may be identified. While each change should be marked, keep in mind that there are only six allowable time changes per profile. It may be necessary to return to some of the earlier time points and combine close points in order to fall within the allowable limit for the number of time points.

In addition to the indications of the slope indicator graph, the selection of time points in this procedure also calls upon experience and intuition to make some of the necessary decisions.

Figure 30 illustrates the results of the time point selection process for a representative slope indicator curve. The top graph shows the slope indicator curve. As reference for this illustration only, solid vertical lines have been drawn at the existing time points for the Austin thresholds. Each dashed vertical line represents a 60-minute time increment, again for reference purposes only.

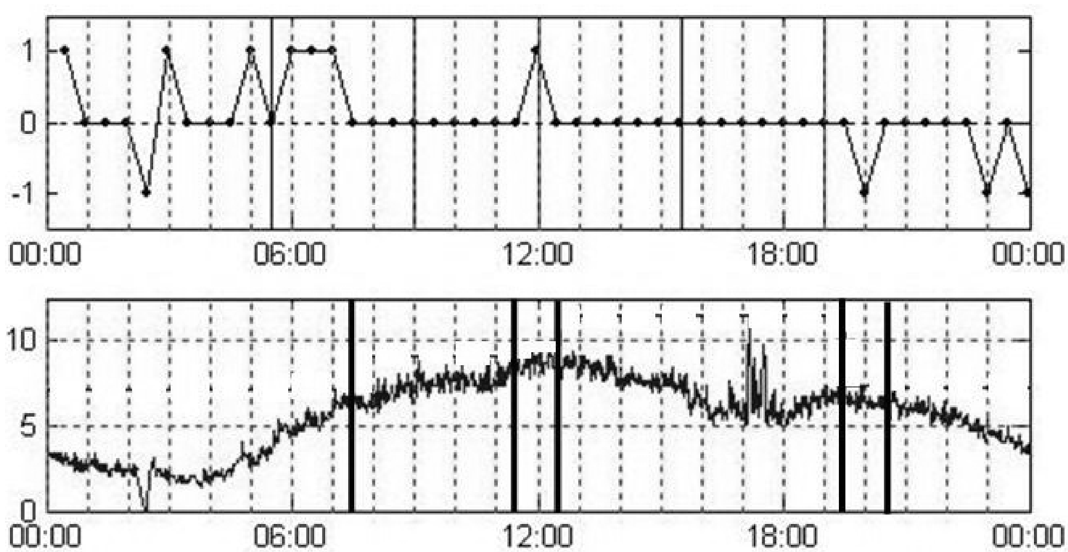


Figure 30. Selection of Time Points for CLC Thresholds.

Still focusing on the topmost graph in Figure 30, the initial points illustrate the discussion points above which stated that the slope indicator always starts at +1 and quickly returns to 0. This does not indicate a need for a time point at 1:00 AM.

At 2:30 AM, the slope indicator starts a period of variability and ranges between -1, 0, and +1. Because of the rapid fluctuation in the slop indicator, it would be prudent to consider putting the first time point at 7:30 AM, when the slope indicator returns to 0 and stays there for an extended period of time.

At 12:00 PM, the slope indicator jumps to 1 and the returns to 0 in the successive time period. While this is a short timeframe, it would be prudent to block this variability by establishing a time point at 11:30 AM and another one at 12:30 PM.

In a similar fashion, the slope indicator experiences a shift to -1 at 8:00 PM (labeled 20:00 on the axis utilized in Figure 30). This variability can be blocked by establishing a time point at 7:30 PM and another one at 8:30 PM.

Toward the end of the 24 hours covered by the graph, the slope indicator varies between 0 and -1. Because this occurs in a timeframe where this is likely to be little to no significant traffic on the roadway, it is not necessary to block these variations as individual time points.

The bottom graph in Figure 30 shows the results of the time point selection process. Notice that the time points are similar to the existing TxDOT algorithm time points but not exactly the same. The fact that the two procedures produce similar time points is actually a good validation of the new process. Some variation is to be expected since the foundation on which the time points are picked are different. However, because both algorithms attempt to measure the variability in traffic, the expectation is that they should produce similar time points.

ESTABLISHING CLC THRESHOLD VALUES

The primary challenge of the CLC threshold procedure is establishing the time points. With that step complete, the remaining task of establishing CLC threshold values is relatively simple.

Recall that the original CLC graphic in Figure 24 illustrated the use of a 99th, 95th, and 50th percentile CLC value for each representative station. These different values were provided as a mean of extending the procedure to accommodate all needs. The higher the percentile used, the more the CLC variability is covered. In essence, using a CLC threshold established at the

50th percentile range will create more false alarms than the 99th percentile, although it is likely that more incidents will be detected. Once again, the trade-offs between the three principles governing incident detection are inviolate and a decision must be made as to which one will be the most useful to the freeway management system. In practice however, it is reasonable to assume that most implementations would utilize a 99th percentile level to reduce false alarms as much as possible.

Manual (Visual) Process for Selecting CLC Threshold Values

The manual procedure for determining threshold values contains two decision points.

1) Determine what percentile level of CLC is desired for use in thresholds. This decision could vary by station, although it would be more consistent to establish the same percentile level for each station.
2) Enter the CLC figure at the desired percentile level and identify the CLC level for each time point established in the previous step.

Figure 31 illustrates the result of this step of the process. The top graph in the figure is the same slope indicator graphic utilized to set the time points. The bottom part of the figure now shows the horizontal lines corresponding to the CLC values in use for each time period. The first time period, from 12:00 AM to 7:30 AM, uses a CLC value of 7. The next time period uses a CLC value of 9 and so on.

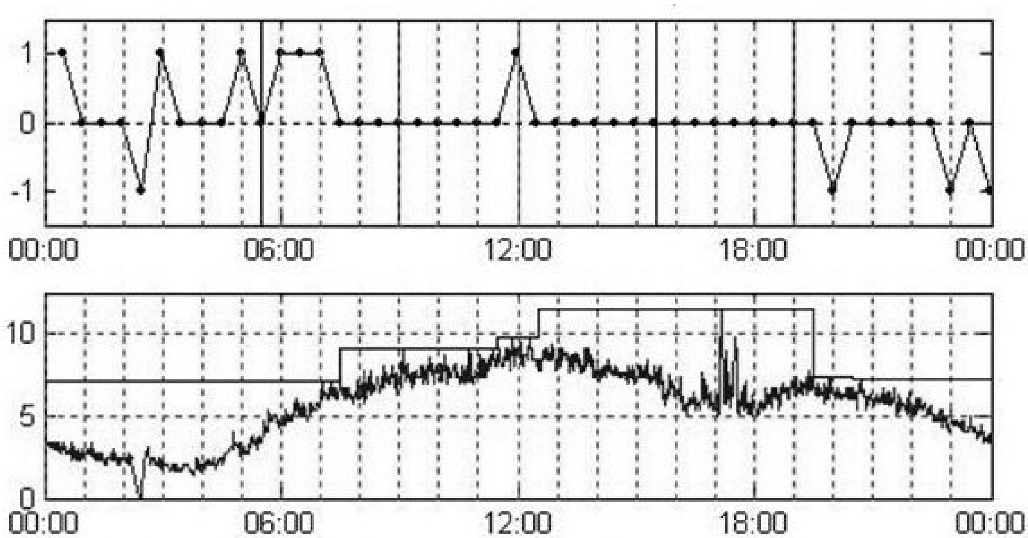


Figure 31. Selection of CLC Values for Thresholds.

AUTOMATING THE CLC THRESHOLD PROCESS

The complete process of setting the CLC thresholds is time-consuming and requires a tremendous amount of data. This is the same drawback identified for the other procedures. However, an advantage of the CLC algorithm is that it can be automated.

Automation carries a caveat. Because of the data requirements, traditional desktop software has limitations which reduce their effectiveness at carrying out the procedures described above. Significant effort was devoted to the task of automating the entire process during this research project. The task was successful using higher-end software than TxDOT typically has access to or has the expertise to use. Unfortunately, there were significant limitations for commonly available software such as Excel.

It is clear that if the desire is to use the more commonly available desktop software, pre-processing of the data will have to take place to prepare specific data files for use. During the course of the project, an automation tool to automatically pick the time periods and corresponding CLC values was prototyped for Excel. However, the two input files of CLC values by minute of the day and slope indicator by 30-minute time increments had to be prepared outside of Excel.

Automated Process for Selecting CLC Threshold Values

The following sections describe a general procedure for automating the CLC threshold process.

Step 1 - Import Raw Data

The initial step in calibrating a new threshold is to import and parse the raw data files. Each raw data file should be named according to a set standard. The standard for the file names shall be "*Roadway SCU YYMMDD HHmm.DET*", where YY is year, MM is month, DD is day, HH is hour, and mm is minute. Using this standard for naming raw data files will enable the automated tool to check for missing data as it establishes a new threshold.

Each raw data file contains one hour of data for each detector on the corresponding roadway, separated into 60 one-minute segments. The automation process begins with importing the first raw file. For the first minute of data, the process will create a new file for each detector and insert the occupancy data for the first minute into that file. The process will then move to the second minute, and insert the occupancy values into the existing files for each detector. This will continue for each of the 60 minutes in the file.

Once the file is complete, the automated process checks to see if any other raw data files exist. If more files are found the process imports the next file. It validates that the current file is the next file in the time sequence (one hour after the previous one). If the file is correct it repeats the process of appending the occupancy values into each detector file. If the file is not the correct file the process must determine how many files were missing, and in each detector file substitute a value of 0 for the missing data. This process will continue for each raw data file.

When complete, each detector file will contain a 365*1440 (366*1440 for leap year) matrix that contains an entire year of 1-minute occupancy data.

Step 2 - Perform Calculations

In order to compute a new threshold, the process must perform a series of calculations for each detector and station. The raw data, which is 1-minute occupancies for each detector for an entire year, must be evaluated to create the 30-minute slope indicator values. This is done for each station for a time period of 24 hours.

Step 2a calculates a 3-minute rolling average for each detector for each minute throughout the year.

Step 2b computes the cross-lane comparison for each station. A station's CLC is defined as the difference between the maximum and minimum rolling occupancy of all lanes at that station. In preparation for subsequent steps in the process, Step 2b should also calculate the 99th, 95th, and 50th percentile CLC value for each minute of the day and store each result as a matrix of 1 x 1440, representing a CLC percentile value at each minute.

Step 2c calculates the mean absolute deviation of the CLC. This leaves a resultant matrix for each station consisting of 1440 data points or one for each minute of a day

Step 2d in the calculations is to determine the 1-minute LAG for each station. The lag is the difference between the current minute and the preceding minute. The resultant matrix is 1 x 1440, representing a LAG1 value for each minute of the day.

Step 2e computes a 30-minute average for each station's 1-minute LAG values. This step creates 48 data points for each station, one for every 30 minute period in the day.

Step 2f computes the average slope of each 30-minute time segment in Step 2e.

Step 2g computes the slope indicator values for each station. The slope indicator data will consist of 48 data points for each station, one point every 30 minutes, starting at 12:00 AM. Each data point will have one of three values: 1, 0, or -1. For each station a boundary is calculated for the average slope. The values of the boundary are calculated using the range and average of the lag for each station. If the value of the average slope at a given point is greater than the higher end of the boundary (2x the average slope) the slope indicator is 1. If the average slope is less than the lower end of the boundary (2x the average slope) the slope indicator is -1. If the average slope falls within the boundary conditions, the slope indicator is 0. The result of this step is a matrix of the 48 slope indicator points, one for each 30-minute time period throughout the 24-hour time analysis timeframe.

Step 3 - Determine New Time Points

Time points are determined using the slope indicator points that resulted from Step 2g. The iterative process begins by looking for the first non-zero slope indicator value. Once it is found a time period boundary is put at that position. The process keeps looking from this point until if finds three consecutive slope indicator values of zero and places the next boundary. The use of three zero values ensures that the process brackets those situations where the slope indicators vary between -1 and 1 within a short timeframe. This process repeats until all points of the slope indicator have been checked.

The next step in this portion of the process checks to see if the number of time points created is greater than the allowable number of six. If there are more than six, some time period boundaries must be removed, in order to reduce the overall number of time periods. This is achieved by looking at the amount of time between boundaries. If these boundaries are less than a specified amount of time apart, the two time periods will be merged into one.

Step 4 - Determine New CLC Threshold Values

Once there are six or fewer time periods created, the Step 3 will terminate and move to the final step in the process. Step 4 identifies the corresponding CLC value based on the time point boundary. This process assumes a 99th percentile CLC value but would be the same for any other percentile value that was specified.

LOGIC FLOW FOR AUTOMATED THRESHOLD TOOL

While the previous section explained the general steps in the automation procedure, a flow chart provides perhaps the best mechanism for the exact logic flow for an automated tool. Figure 32 through Figure 36 present the logic flow that could be used to build an automated capability. These flow charts, which represent Deliverable P4 of the project, could be used to build a software module within ATMS, a stand-alone software program, or to simply replicate the logic in a data analysis program.

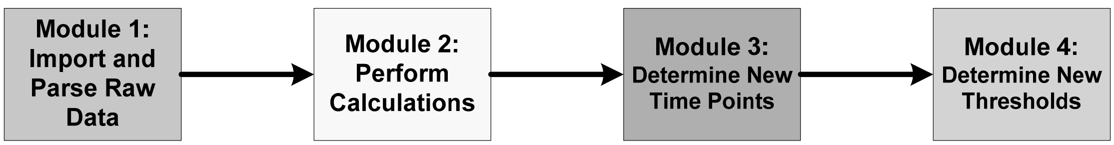


Figure 32. Overview of Automated Logic Flow.

As shown in Figure 32, the overview of the automated logic flow identifies four modules or components. This corresponds to the stepwise text listing in the previous section.

Figure 33 provides the detailed flow chart diagram for Module 1. The process starts with the first raw data file which is parsed for detectors and occupancy values. Logic checks are included to check for the presence of additional data files so that the process is iterative until all data files have been read. The logic also includes a component to check for missing data and substitute for the missing data if found. The resulting output is a stored table of occupancy values for each detector, at which point the logic passes to Module 2 of the process.

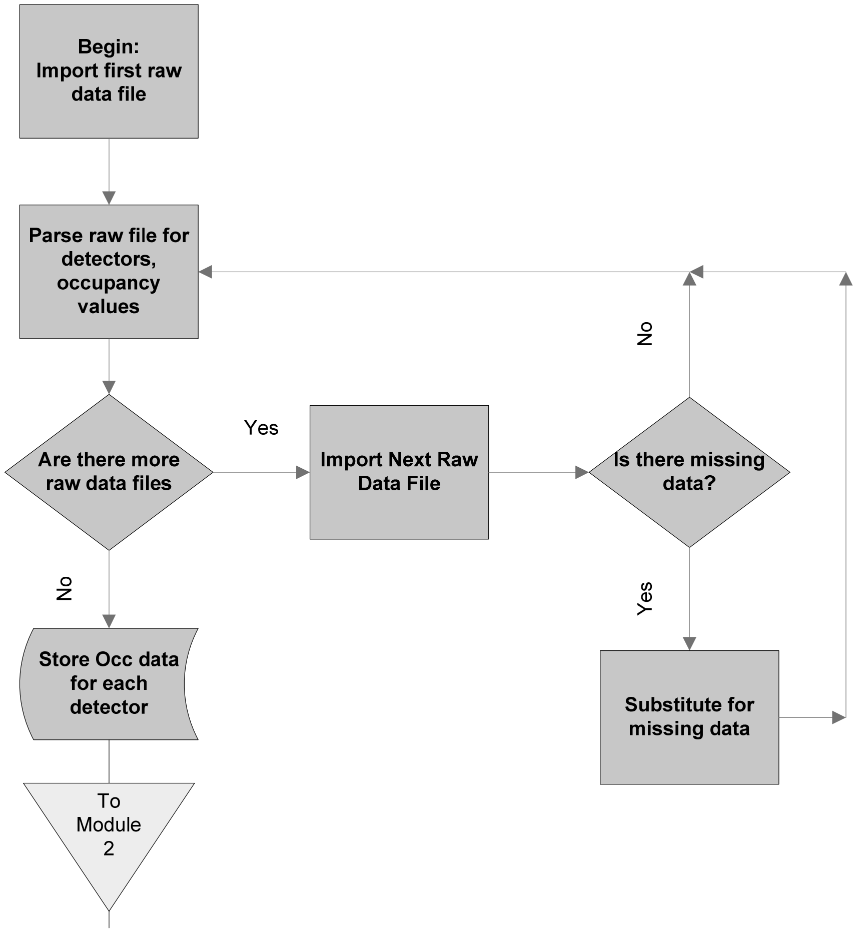


Figure 33. Detailed Flow Chart for Module 1 – Import and Parse Raw Data.

Figure 34 details the logic flow for the second module of the overall process. This is a straightforward module consisting of a sequence of computational steps using the formulas identified in the previous sections. The process starts with the detector occupancy files created in Module 1 and terminates with the calculation of the slope indicator values for each station. At this point, logic control is passed to the third module in the process.

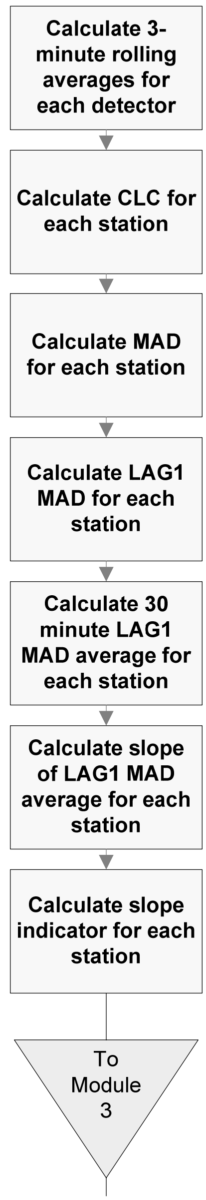


Figure 34. Detailed Flow Chart for Module 2 – Perform Calculations.

Figure 35 highlights the logic flow for Module 3, the determination of time periods for the CLC thresholds. As indicated in the text in previous sections, the logic flow utilizes the slope indicator points as a means to measure the departure, in any given time period, from the same conditions that existed in the previous time period. While the procedure utilizes a time period basis of 30 minutes, corresponding to 48 increments during the 24-hour timeframe, the procedure could be easily modified to support other time increments.

Take note that Modules 1 and 2 perform their portion of the procedure for every station in the data set. Starting with Module 3, the process takes place for one station at a time and proceeds to Module 4. Module 4 will contain a test for additional stations and process the logic flow handoff accordingly.

There are two exit points for the time period process in Module 3. The first occurs if the establishment of time periods is complete and less than six have been identified. The second exit point occurs after a time period combination component if more than six time periods have been observed. The exit point for this module sends logic control to Module 4.

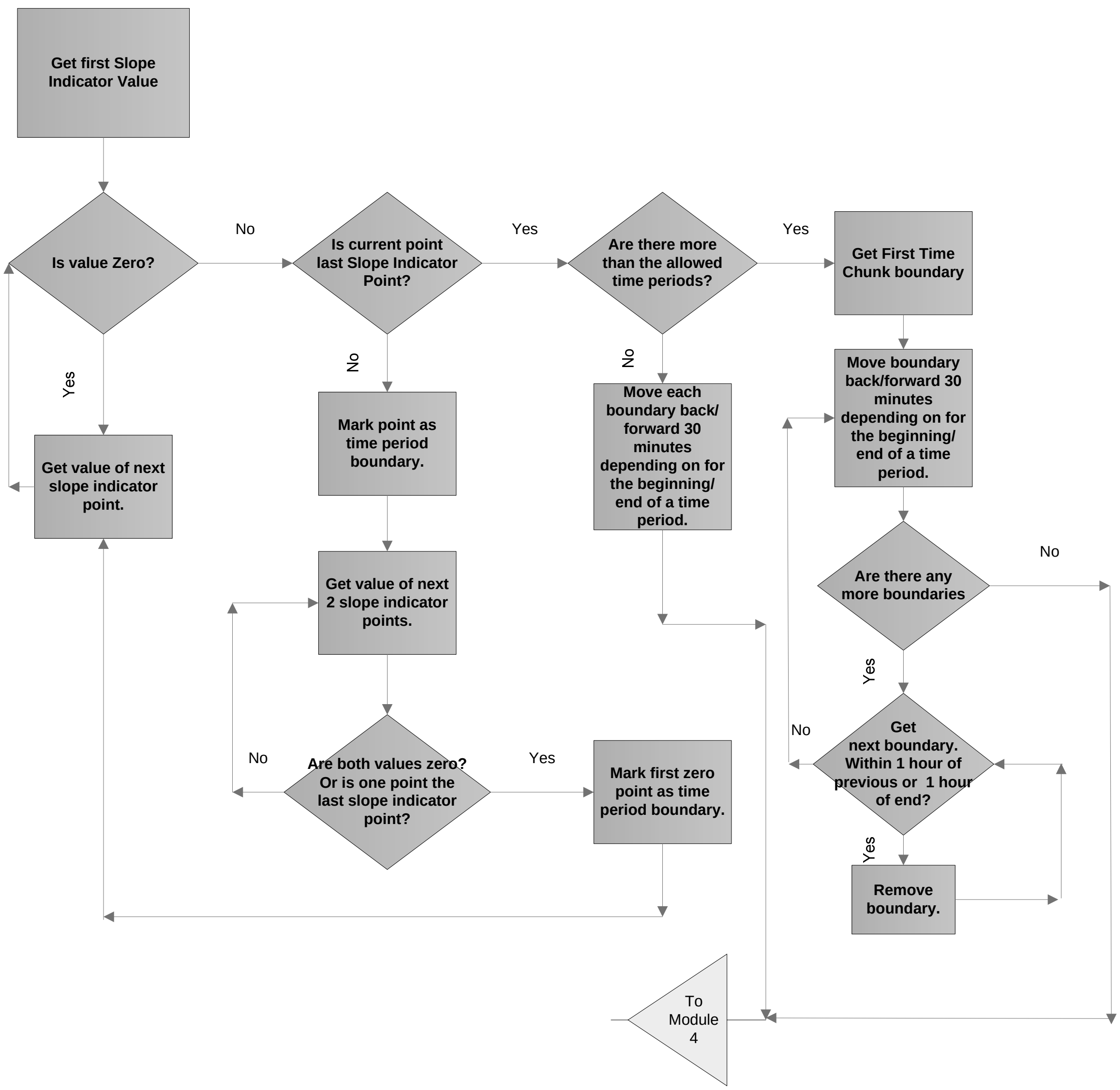


Figure 35. Detailed Flow Chart for Module 3 – Determine New Time Periods.

Figure 36 details the procedures included in Module 4 of the automated procedures. The focus of this module is obtaining the new threshold values based on the time periods identified in Module 3. It should be noted that Figure 36 assumes the use of the 99th percentile CLC values for calculating thresholds. The logic flow could easily be modified to support the use of 95th, 50th, or other defined percentile values, as desired for a specific implementation.

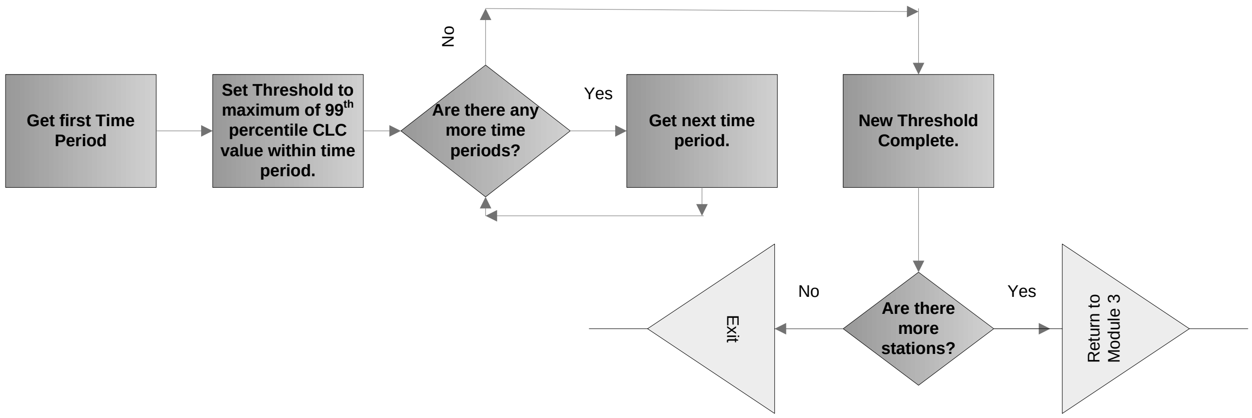


Figure 36. Detailed Flow Chart for Module 4 – Determine New Time Thresholds.

At the conclusion of Module 4, the logic flow tests for the presence of additional stations that have not yet been processed to determine threshold values. If this test is found to be true, the logic flow returns to Module 3, in an iterative process until all stations have been analyzed. If this is the final station in the process, the logic flow exits the procedure.

CHAPTER 7 – CROSS-LANE COMPARISON PERFORMANCE ASSESSMENT

INTRODUCTION

Chapter 2 discussed the typical methods of assessing incident detection algorithm performance. Algorithms are typically judged on their effectiveness of:

- minimizing false alarms (false alarm rate),
- maximizing detections (detection rate), and
- minimizing the time to each detection (detection time).

Unfortunately, as illustrated in Figure 1, these parameters are not mutually exclusive. In fact, minimizing false alarms tends to inversely affect on both the detection rate and detection time. Likewise, a focus on maximizing the detection rate and/or minimizing the detection time tends to greatly increase the number of false alarms. In short, there is no perfect solution that allows for the best of all possible performance. The reality is that because of the trade-offs, agencies typically have to focus on one parameter and achieve the best compromise with the performance of the other parameters.

In order to assess the performance of the CLC algorithm, the performance parameters were determined using 2002 data from Austin, Texas. At each step in the performance assessment, the results were compared to the performance parameters of the existing TxDOT algorithm, again using 2002 data from Austin, Texas.

As a second assessment of performance, the sustainability of the CLC thresholds across multiple years was examined. Additional datasets from 2003 and 2004 from Austin, Texas, were obtained, prepared according to the data reduction procedures described in Chapter 4, and utilized to determine the same performance parameters.

Investigation into the sustainability of the thresholds centered on how long the thresholds would last. Consider for example that CLC thresholds were created in 2002. Would those same thresholds continue to work well in 2003? What would the performance be in 2004? In order to answer these questions, a matrix of performance parameters using CLC thresholds from previous years was created, providing significant insight into the question of sustainability.

Finally, as part of the assessment, operators throughout the state were contacted and questioned to determine their opinions and desires with regard to automatic incident detection algorithms.

The information presented in this chapter represents Deliverable P1 of the research project, a comprehensive assessment of the performance capabilities of the incident detection algorithm.

CLC PERFORMANCE ASSESSMENT USING 2002 DATA

False Alarms

The first assessment of CLC performance was performed using false alarms. Similar to the procedures identified in other chapters of this report, a MatLab program was written to calculate the false alarms for each station. Because the algorithm utilizes a simple comparative threshold, the flagging of false alarms is essentially an accounting process. At each minute of the day, the CLC value is calculated as the difference between the maximum and minimum values of the 3-minute rolling average for all detectors in the cross-section. This CLC value is compared to the CLC thresholds, the creations of which were detailed in Chapter 6.

The false alarm accounting utilized in these assessments was a simple accumulation of total number of false alarms. This method was used instead of the on-line and off-line false alarms rates because of a potential misleading problem with the rates. In both of the rate calculations, the total number of false alarms is utilized. In the case of a simple comparative algorithm, the threshold value is likely to be exceeded at the beginning of an incident and stay above the threshold for some period of time. Because a comparison occurs each minute, the number of false alarms would be greatly magnified, and the rate would not represent the true performance of either algorithm.

Although this same problem would occur in the simple accounting procedure, the accounting procedure was modified to skip forward 30 minutes after each false alarm detection. This moves beyond the timeframe of the typical incident and avoids multiple counting and inflation of the number of false alarms. The information pertaining to the false alarm assessment procedure follows.

Step 1: Calculate False Alarm Parameters

MatLab File: False_Alarms_WW_Pct.m

Input Files: Inventory.txt
Daily.txt
Row.txt
IncidentLog_Sum2_Up.txt
DET*XXX*_OCC_STA*YYY*.txt
newTimeChunkSta*YYY*.txt
*WW*_PCT_Thresh_Sta*YYY*.txt

Output Files: *WW*_PCT_False_Alarm_Counts.txt
Station_*YYY*_*WW*_PCT_False_Alarms.txt
Station_*YYY*_*WW*_PCT_False_Alarm_Times.txt

Where:

WW is 50, 95, or 99 percentile
XXX is detector number
YYY is station number

Procedure: This step calculates the false alarms generated by the new CLC thresholds and the existing TxDOT algorithm thresholds.

File Descriptions:

WW_PCT_False_Alarm_Counts.txt:
Shows a summary of each station's false alarms counts separated by old and new threshold values.

Figure 37 shows a sample from an output file that is created as a result of this procedure. The file details, by station number, the total number of false alarms accumulated by both the new (CLC) and old (existing TxDOT) algorithms. Also included is the total number of days in the dataset. Because the accounting is cumulative across the entire year, dividing the number of false alarms by the number of days provides an average number of false alarms per day, per detector. The false alarm summary procedure was conducted on a subset of the 2002 data consisting of non-incident, non-holiday weekdays.

Station	New	Old	Days
1	117	0	261
13	38	11	261
23	84	2	261
25	55	3	261
29	75	4	260
31	70	19	259
33	57	13	260
36	1764	40	261
40	3	59	260

Figure 37. False Alarm Summary.

Station_YYY_WW_PCT_False_Alarms.txt:

Summary of false alarms for a particular station, broken down by day.

Figure 38 shows a sample from an additional output file that is created as a result of this procedure. The file details, by date, the false alarms detected by each of the algorithm procedures. One file of this type is created for each station in the dataset.

Day	New	Old
01/06/2003	1	0
01/09/2003	1	0
01/20/2003	1	0
02/13/2003	2	0
02/17/2003	1	0
02/20/2003	1	0
02/21/2003	1	0

Figure 38. Station False Alarms.

*Station_YYY_WW_PCT_False_Alarm_Times.tx*t:

Shows the time that each false alarm occurred for a particular station.

Figure 39 shows a sample from the last output file that is created as a result of this procedure. The file contains, by date, the detailed false alarms information for each detection. This file was used as a double check to ensure that multiple false alarms were not being flagged for the same incident on any given day. One file of this type is created for each station in the dataset.

TH	Day	Time
New	01/06/2003	7:08
New	01/09/2003	7:09
New	01/20/2003	21:12
New	02/13/2003	7:16
New	02/13/2003	9:52
New	02/17/2003	7:29
New	02/20/2003	7:16

Figure 39. False Alarm Times.

The information contained in the output files from this data analysis was parsed into Excel for easy tabulation and accounting.

Table 1 shows the results of the false alarm tabulations for US 183 for 2002 data. The values shown in the table are average false alarms per day, obtained by dividing the total number of false alarms by the number of non-incident, non-holiday weekdays.

The comparison of individual detector stations is interesting. There are some locations where the CLC algorithm performed better than the existing TxDOT algorithm, such as Station 97. However, other locations, such as Station 69, performed worse under the CLC algorithm.

Table 1. US183 Average False Alarms Per Detector, Per Day, For 2002 Data.

	Algorithm	
Station Number	**CLC**	**Existing TxDOT**
1	.39	0
3	.20	0
13	.13	.02
16	.14	.22
21	.4	.07
22	.29	.04
23	.30	.01
24	.37	.01
25	.31	.02
26	.2	.05
29	.35	.03
31	.15	.09
33	.39	.1
42	.34	.004
44	.23	8.05
46	.32	0
48	.26	.15
50	.27	.18
55	.23	.15
59	.18	.19
60	.37	.3
61	.44	.7
62	.42	0
63	.21	.22
64	.27	.12
66	.58	1.6
67	2.58	2.44
69	1.01	0
72	.01	.03
73	.27	.71
74	.18	.5
76	.31	0
81	.65	1.27
82	.38	0
84	.21	2.64
85	.47	.06
87	.6	2.23
89	.26	.94
90	.20	2.41
91	.28	0.03
92	.26	.003
93	.30	0
95	.26	.46
97	.39	6.56
98	.28	1.38
101	.32	.02

What is perhaps more revealing is the total number of false alarms on average per day delineated by roadway. Table 2 details the average daily false alarms for each roadway in the Austin 2002 dataset. On most roadways, the CLC algorithm resulted in fewer false alarms than the existing TxDOT algorithm.

Table 2. Average Daily False Alarms By Roadway For 2002 Data.

	Algorithm	
Roadway	**CLC**	**Existing TxDOT**
US 183 (46 Stations)	16.97	34.04
Loop 1 (28 Stations)	43.4	52.69
IH 35 (2 Stations)	1.1	0.18

Incident Detection

The second assessment of the CLC algorithm's performance centered around incident detection. Incident detection performance is measured in terms of both the rate, i.e. percentage of incidents detected, and the incident detection time, the lag between the real-time occurrence of the incident and the detection by the algorithm.

Similar to the procedures identified in other chapters of this report, a MatLab program was written to tabulate the incident detection performance. The dataset used for this procedure was the known set of major incidents culled from the operator's log. The basic procedure is to utilize the occupancy values from a known incident day and run them against the algorithm thresholds to determine if (a) the incident is detected, and (b) when it is detected. This information is then tabulated for each incident and the results analyzed to determine the overall performance. The methodology for tabulating the incident detection performance is as follows:

Step 1: Calculate Incident Detection Parameters

MatLab File: IncidentDetection_2.m

Input Files: Inventory.txt
Daily.txt
Row.txt
IncidentLog_Sum2_Up.txt
DET*XXX*_OCC_STA*YYY*.txt
newTimeChunkSta*YYY*.txt
*WW*_PCT_Thresh_Sta*YYY*.txt

99_PCT_Thresh_Sta*YYY*.txt
95_PCT_Thresh_Sta*YYY*.txt
50_PCT_Thresh_Sta*YYY*.txt

Output Files: Incident_Times.txt

Where:

XXX is detector number
YYY is station number

Procedure: This step looks at each recorded incident in the reduced dataset and determines if and when the thresholds would have logged the incident.

File Descriptions:

Incident_Times.txt
Contains the detections times for each incident recorded in the incident log.

Figure 40 shows a sample of the output from this procedure. Some lines indicate that the incident was detected and show the corresponding detection times for the 99th, 95th, and 50th percentile thresholds. The file also tabulates the detection success and time of the existing TxDOT algorithm.

Date	Incident Station	Reported Time	99% Thresh	95% Thresh	50% Thresh	Old Thresh
12/10/2003	There is no CLC data for this detector on this day.					
02/26/2003	31	17:06	16:45	15:57	Undetected	
06/10/2003	31	13:23	Undetected	Undetected	Undetected	
05/22/2003	33	16:32	15:56	15:52	15:56	
06/03/2003	There is no CLC data for this detector on this day.					
02/20/2003	42	10:27	Undetected	10:01	Undetected	
03/06/2003	42	15:01	14:48	14:47	14:48	
04/28/2003	42	10:28	Undetected	Undetected	Undetected	
01/02/2003	50	15:42	Undetected	Undetected	Undetected	

Figure 40. Detection Times.

The performance of the algorithms is somewhat difficult to see in the raw output file. For this reason, the information was parsed into Excel, where data manipulation could be performed to more easily quantify the performance.

Table 3 details the incident detection performance for US 183 and the known major incidents. A review of the data shows that the CLC algorithm detected more incidents than the existing TxDOT algorithm.

It is worthwhile to note at this juncture that the data pertaining to detection time is considered suspect at best. In most operations centers, when an incident occurs, the focus is on verification, dispatching aid, and managing upstream traffic. It is typically only after these tasks are accomplished that operators record the information into the incident logs. The time occurrence is therefore highly suspect as there may be a several minute delay between the actual time of the incident and the recorded time in the operator's log. This tends to give a false impression about the capabilities of either algorithm in terms of how quickly incidents can be detected. While it is reported for completeness, the researchers do not have sufficient faith in the validity of the incident detection time data to draw accurate conclusions.

Table 3. Incident Detection Performance for US 183, 2002 Data.

			Algorithm			
			CLC		Existing TxDOT	
Incident Date	Incident Time	Station	Detection Time	Delay	Detection Time	Delay
1/18/02	16:50	42	16:42	-8	16:44	-6
1/31/02	7:54	3	7:46	-8	Undetected	
3/25/02	8:11	85	8:17	6	Undetected	
5/7/02	6:08	42	Undetected		Undetected	
5/7/02	7:51	74	8:00	9	7:50	-1
5/21/02	20:17	59	20:58	41	Undetected	
5/21/02	18:21	67	18:13	-8	18:02	-19
6/6/02	8:07	42	Undetected		Undetected	
6/7/02	13:58	42	Undetected		Undetected	
7/3/02	18:30	89	18:23	-7	17:58	-32
7/17/02	7:49	50	Undetected		Undetected	
7/18/02	16:35	13	16:10	-25	16:19	-16
7/18/02	9:55	31	10:03	8	Undetected	
7/23/02	18:34	50	Undetected		Undetected	
7/24/02	13:14	48	13:10	-4	Undetected	
7/30/02	8:20	13	8:13	-7	8:17	-3
7/31/02	13:24	69	13:24	0	Undetected	
8/6/02	7:27	16	Undetected		Undetected	
8/8/02	9:07	42	Undetected		Undetected	
8/12/02	11:15	50	Undetected		Undetected	
8/20/02	12:44	60	Undetected		Undetected	
8/29/02	17:34	85	17:04	-30	17:22	-8
9/11/02	7:50	3	Undetected		Undetected	
9/25/02	17:55	31	17:55	0	Undetected	
10/1/02	15:25	1	Undetected		Undetected	
10/22/02	17:25	1	17:26	1	Undetected	
10/24/02	14:43	26	14:48	5	Undetected	
11/1/02	18:28	23	18:23	-5	Undetected	
11/6/02	18:35	3	18:31	-4	Undetected	
12/6/02	8:27	24	8:30	3	8:31	1
12/6/02	8:43	31	8:05	-38	8:05	-38
12/12/02	9:48	50	Undetected		Undetected	

Table 4 and Table 5 detail the summary result for Austin roadways using the 2002 dataset. As can be seen from Table 4, the CLC algorithm had a higher incident detection rate than the existing TxDOT algorithm, signifying an increase in performance.

Table 4. Incident Detection Rate for Austin Roadways, 2002 Data.

	Algorithm	
Roadway	**CLC**	**Existing TxDOT**
US 183	62.5%	28.13%
Loop 1	50%	25%
IH 35	NA	NA

Table 5. Incident Detection Time for Austin Roadways, 2002 Data.

	Algorithm	
Roadway	**CLC**	**Existing TxDOT**
US 183	-3.55	-13.55
Loop 1	-10.25	20
IH 35	NA	NA

Table 5 shows a mix of results. The negative numbers indicate that the algorithm detected the incident prior to the recorded time in the operator's log. A value of -3.55 minutes, as is recorded for the CLC performance on US 183 sounds plausible, indicating a 3- to 4-minute delay between responding to an incident and information being recorded in the operator's log. However, the detection time values in the table show a wide range of performance, from the existing TxDOT algorithm detecting incidents more than 13 minutes *prior* to any record keeping on one roadway to detecting incidents 20 minutes *after* the recorded time on another roadway. These trends cannot be explained in the data and as indicated earlier, are questionable results.

CLC Performance Summary

In summary, the results of assessing the CLC algorithm are promising. Using the 2002 dataset from Austin shows a decrease in the number of false alarms and an increase in the detection rate. Future changes in the mechanism for recording incident information may lead to some substantive conclusions regarding incident detection time, but none are possible at this time.

MULTI-YEAR SUSTAINABILITY ASSESSMENT

One of the issues with the existing TxDOT algorithm is the fact that thresholds have to be "refreshed" or updated periodically. This is due to increases or changes in traffic volume. As volumes increase, the occupancy levels used as thresholds must be adjusted as well.

Because the CLC algorithm relies on a difference in occupancy values rather than an absolute value, an important question was raised as to the sustainability of the CLC thresholds. Would they also have to be refreshed on a periodic schedule or would they have a longer 'shelf life'?

To investigate this question, a series of comparisons were planned to compare the performance of the CLC algorithm across multiple years. Consider that the starting point was the known assessment of 2002 data and thresholds created from this data. Additional performance results could be cataloged for the following comparisons:

- 2002 thresholds with 2003 data,
- 2002 thresholds with 2004 data,
- 2003 thresholds with 2003 data,
- 2003 thresholds with 2004 data,
- 2004 thresholds with 2004 data,
- TxDOT 2002 thresholds with 2003 data, and
- TxDOT 2003 thresholds with 2004 data.

Together, these comparisons would provide a comprehensive assessment of the capability of the CLC thresholds for multi-year sustainability.

Data Requirements

In order to perform the analyses listed above, a significant amount of additional data and processing time were required. The research team obtained 2003 and 2004 data from the Austin district. This data was comprised of the same SCU Detector files files that formed the starting point for 2002 data manipulation as well as the operators' logs.

The data manipulation steps detailed in Chapter 4 were run on each of the new datasets to create the basic files necessary for performing this comparison. When complete, the data analysis steps detailed in the earlier part of this chapter were also run, resulting in a complete set of performance parameters for all roadways across 2002, 2003, and 2004, for both the CLC algorithm and the existing TxDOT algorithm. The following tables show summary statistics only for these analyses. The detailed information is exactly analogous to the results reported in Table 1 and Table 3.

False Alarm Comparison

Table 6 shows the results of the false alarm comparison for the multi-year sustainability analysis. Cells in which the year of the data corresponds to the year of the thresholds is presented in the lightest shade of grey. An example would be a cell such as Year 2004 data with CLC thresholds from year 2004. The value of this cell is approximately 36.5, signifying that on average; operators could expect 36 to 37 false alarms per day, cumulative, across all the roadways in Austin.

Table 6. False Alarm Comparison for Multi-Year Analysis (Average Daily Alarms Across All Austin Roadways).

	THRESHOLDS			
Year	**CLC-2004**	**CLC-2003**	**CLC-2002**	**TxDOT**
2004	36.45	142.31	179.95	94.47
2003	***	29.31	39.68	77.75
2002	***	***	60.37	86.73

Similarly, the number of daily false alarms for the TxDOT algorithm, when using 2002 thresholds and 2002 data, is expected to be, on average, between 86 and 87.

The values in Table 6 also show the results of not refreshing the thresholds for a time period of one year. This would be the case for the medium-grey cells, such as CLC-2002 thresholds and 2003 data. This value shows 39.68 which again is the anticipated average number of daily false alarms that would be expected when using thresholds that were one year old. This is actually a decrease over the total value of 60.37 reported for CLC-2002 thresholds and 2002 data.

The corresponding value for the existing TxDOT algorithm and 1-year old threshold values was 77.75, showing that the expected average number of daily false alarms with the TxDOT algorithm is nearly double the number forecast by the CLC algorithm.

Finally, Table 6 also shows the result of waiting two years before redoing the thresholds. Using CLC-2002 thresholds and 2004 data, the expected number of average daily false alarms jumps to nearly 180. By comparison, the TxDOT algorithm with the corresponding 2-year old thresholds shows an average daily false alarm expectation of approximately 95.

The results of the false alarm comparison clearly show that the performance of the CLC algorithm is best when the thresholds are created from the current year's data. While some degradation in performance is seen with 1-year old thresholds, performance is noticeably poorer with 2-year old thresholds, as evidenced by the rapid rise in average daily false alarms.

Incident Detection Comparison

Table 7 summarizes the incident detection rates that can be expected based on the multi-year sustainability analysis. In all cases, the use of current year thresholds with current year data exceeds the performance of the TxDOT algorithm. Generally speaking, Table 7 shows a trend of decreasing incident detection rates as thresholds are aged past one year.

Table 7. Incident Detection Rate Comparison for Multi-Year Analysis (Average Percenatage of Detected Incidents Across All Austin Roadways).

	THRESHOLDS			
Year	**CLC-2004**	**CLC-2003**	**CLC-2002**	**TxDOT**
2004	29.63	41.98	47.06	27.16
2003	***	52.63	55.56	31.58
2002	***	***	60	27.50

Table 8 summarizes the information pertaining to detection time. As has been previously discussed, a number of questions related to the record keeping practices associated with incidents make it difficult to draw any substantive conclusions. While the results appear to be within the realm of reason, the underlying questions raise an important concern relative to the validity of the data.

Table 8. Incident Detection Time Comparison for Multi-Year Analysis (Average Incident Detection Time Lag to Operator's Logs Across All Austin Roadways).

	THRESHOLDS			
Year	**CLC-2004**	**CLC-2003**	**CLC-2002**	**TxDOT**
2004	-20.79	-19.03	-22.44	-16.18
2003	***	-22.5	-8.4	-22.33
2002	***	***	-4.67	-7.45

OPERATORS' ASSESSMENT OF AUTOMATIC INCIDENT DETECTION

While the research for this project focused primarily on improving and cataloging the performance of the incident detection algorithm, an important aspect of the overall assessment was cataloging the operators' expectations for automatic incident detection algorithms. If these expectations cannot be met, regardless of the algorithm in use, then a discrepancy exists between perception and reality that must be addressed.

Although not every management center within Texas uses the TxDOT ATMS platform, the survey was broad enough to solicit opinions from across the state. This provided a wider base of responses and also served to clarify expectations for other incident detection mechanisms.

The appendix of this report contains a full copy of the questionnaire that was used for the operator survey. This questionnaire was administered in person and took approximately 15 minutes to complete.

The survey consisted of three sections:

- TMC Background Information (Questions 1 through 7),
- Incident Detection Information (Questions 8-20), and
- General Comments.

The first section served mainly to gather information about center staffing and hours of operation. These issues are not germane to the operators' expectations of incident detection performance and will not be summarized. Questions 9 through 20, which highlight the incident detection performance expectations across Texas are summarized below. A total of 15 responses from operators in five cities were received.

Figure 41 shows that more than half of the respondents indicated that occupancy is the primary performance parameter on which their incident detection capability is based. This value is probably somewhat misleading, as there were multiple respondents for cities that do not utilize ATMS.

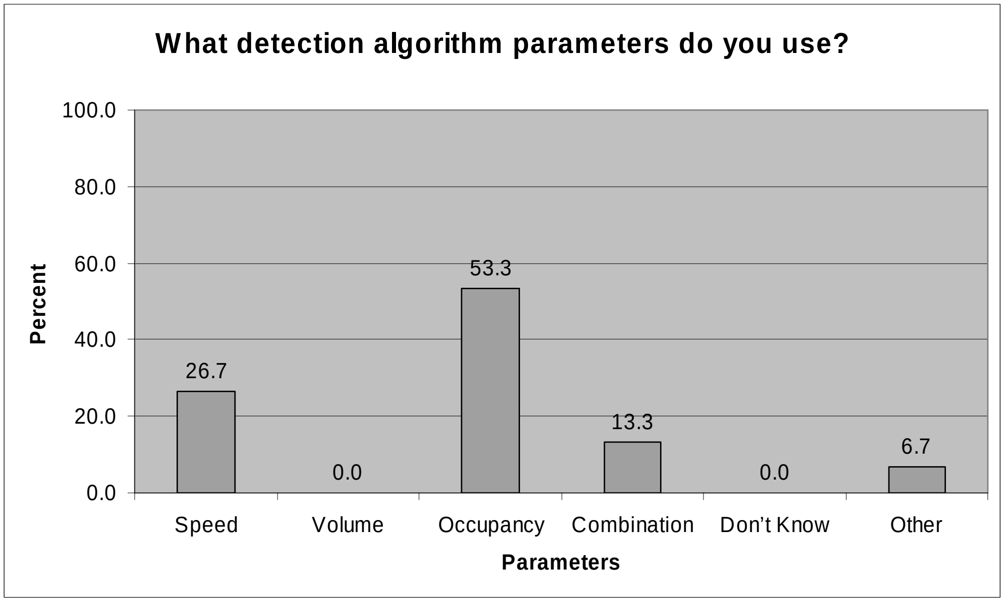


Figure 41. Operator Survey – Detection Algorithm Parameters.

Figure 42 identifies that more than half of the respondents come from systems where the data collection time period is 20-seconds. The basis for both the existing TxDOT algorithm and the CLC are the 20-second loop occupancy values. This indicates that occupancy is the primary performance parameter on which their incident detection capability is based. This value is probably somewhat misleading, as there were multiple respondents for cities that do not utilize ATMS.

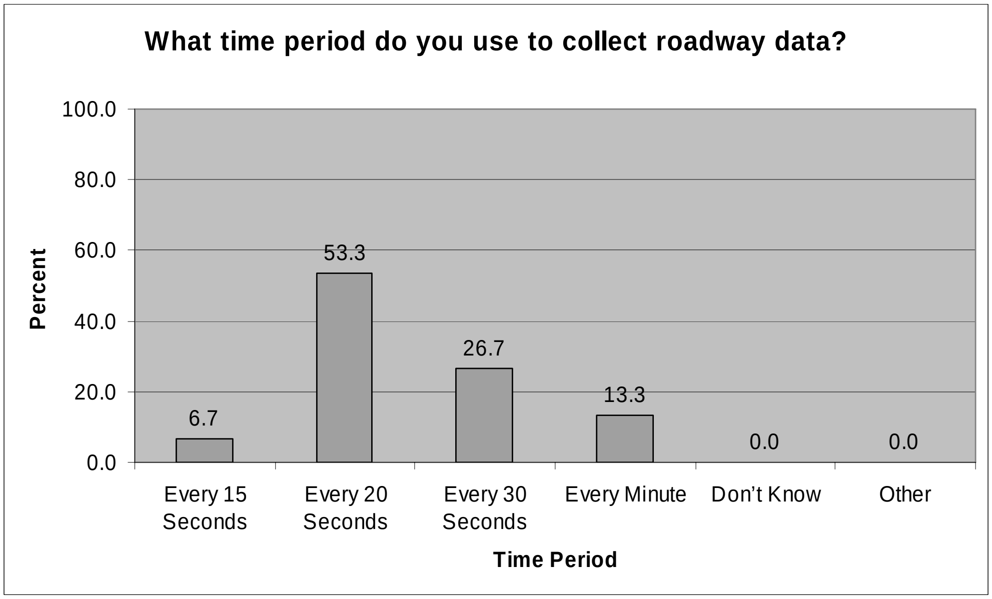


Figure 42. Operator Survey – Roadway Collection Time Period.

Figure 43 reflects the responses from operators regarding the incident detection time period in use for their incident detection implementation. Because all of the Austin operators answered the survey and the center utilizes a 3-minute rolling average, there is a preponderance of answers at that data point.

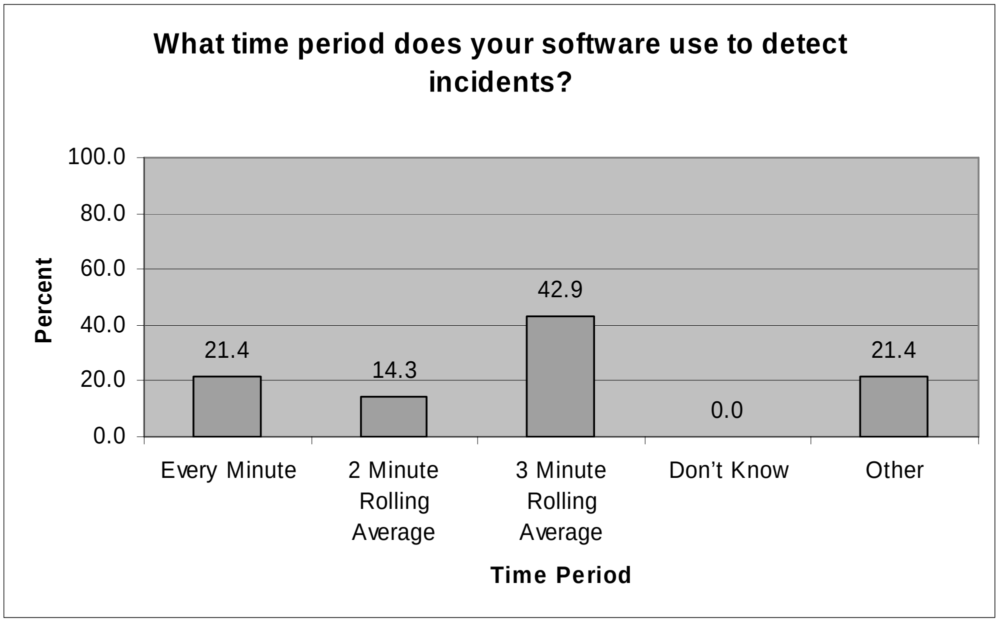


Figure 43. Operator Survey – Incident Detection Time Period.

Figure 44 summarizes the results of what operators feel is the most significant factor in incident detection. Of particular interest is that even though the responses indicate either detection time or detection rate, informal conversations suggest that operators have a strong negative reaction to a high number of false alarms and do not appear to carry the same level of concern for the detection parameters.

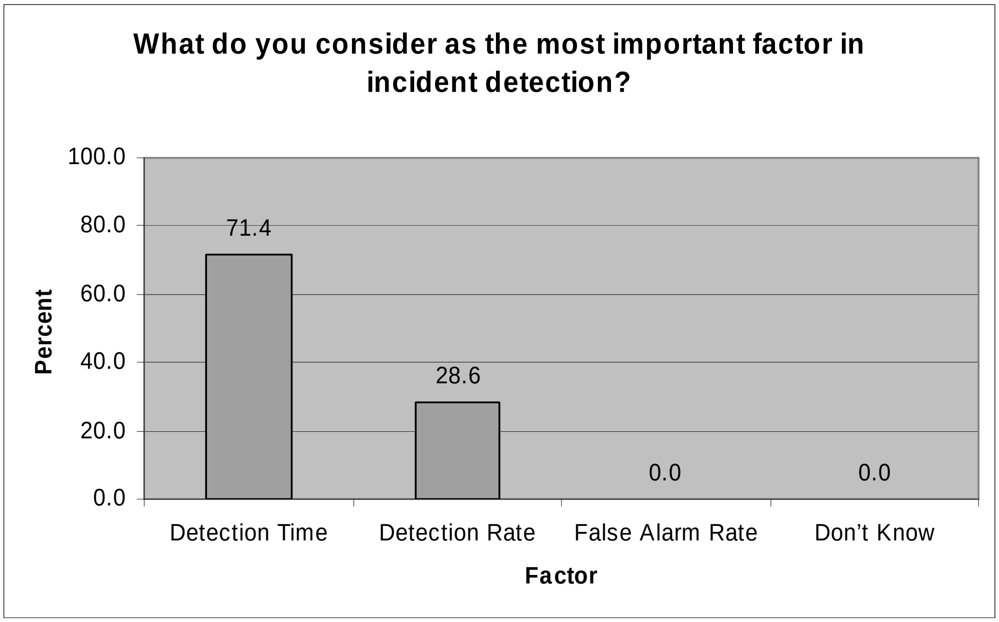


Figure 44. Operator Survey – Critical Factor in Incident Detection.

Figure 45 shows the responses for the operators' desire for incident detection speed. It is interesting to note that 60 percent of the respondents would like to see an incident detected in less than 90 seconds. However, referring back to Figure 43, nearly 80 percent of the incident detection platforms in use throughout Texas utilize a detection time period of 2-minutes or longer. In this response, the operators' expectations are clearly inconsistent with the software capabilities.

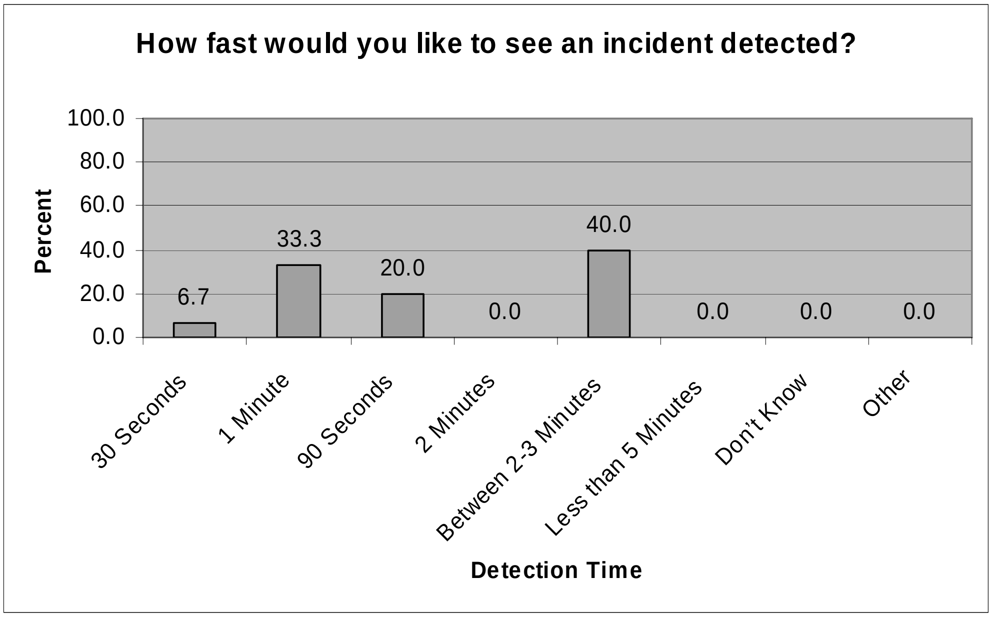


Figure 45. Operator Survey – Desired Speed for Incident Detection.

Figure 46 illustrates the responses that operators gave when posed the question about their maximum waiting time to achieve a 100% detection capability. More than 70 percent indicated a willingness to wait less than two minutes. Nearly 50 percent indicated a willingness to wait 1-minute or less. This is again contrasted with Figure 43, which shows that 80 percent of the respondents have a detection time period exceeding two minutes. It is highly unlikely that these detection time expectations can be met with the current software implementations.

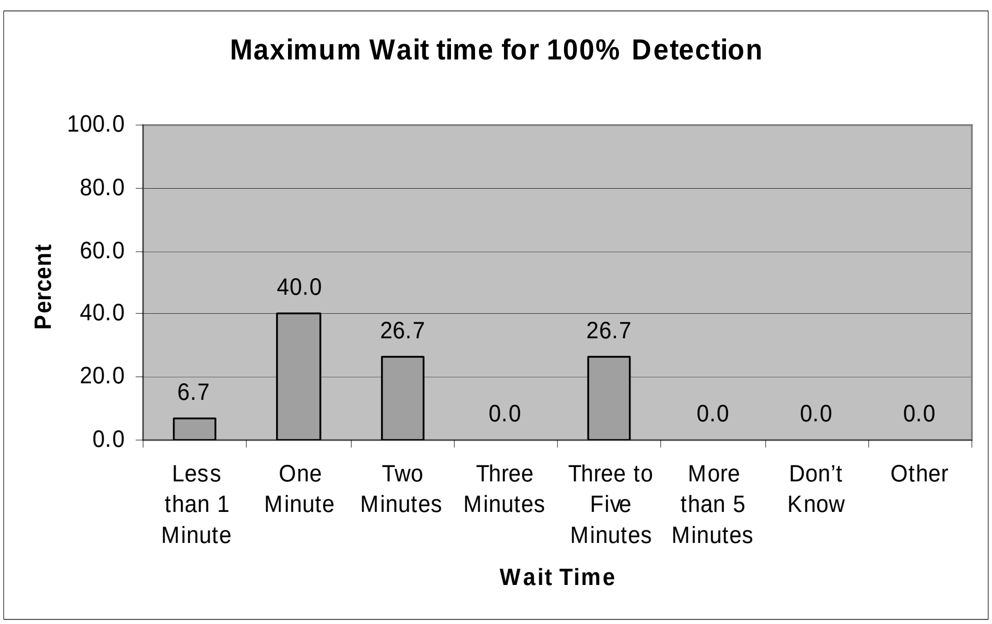


Figure 46. Operator Survey – Maximum Wait Time for 100% Detection.

Figure 47 details the acceptable numbers of false alarms that operators are willing to accept, per day, per station. A significant percentage of the respondents indicated a range of two to five false alarms per day, per station. While that value sounds minimal, it can quickly add up.

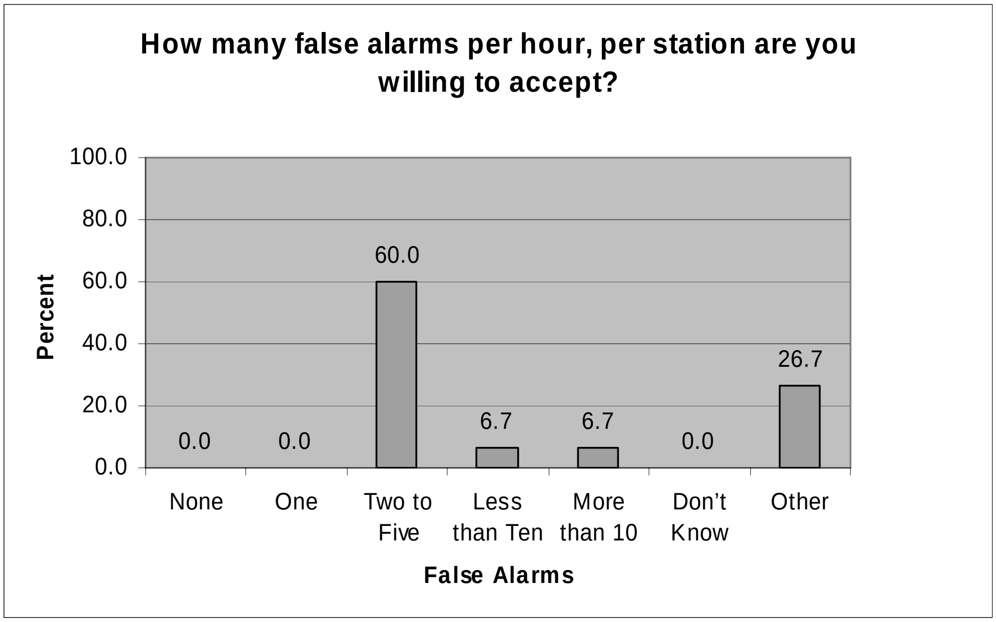


Figure 47. Operator Survey – Acceptable False Alarms.

An ATMS implementation such as Austin, with more than 100 detector stations would therefore receive 200 to 500 false alarms per day. In addition, since Austin stations their detectors by lane, a typical 3-lane cross section would increase the daily false alarms to between 600 and 1,500. This would be a significant operator workload, resulting in one to three false alarms per minute over the operating hours of the facility. In comparison, Figure 48 summarizes the number of simultaneous false alarms that responding operators felt they could effectively handle. The predominant answer of two to five is well within the range of the one to three false alarms per minute from the previous question. However, it is anticipated that the constant display of false alarms would substantially reduce the effectiveness of the operators.

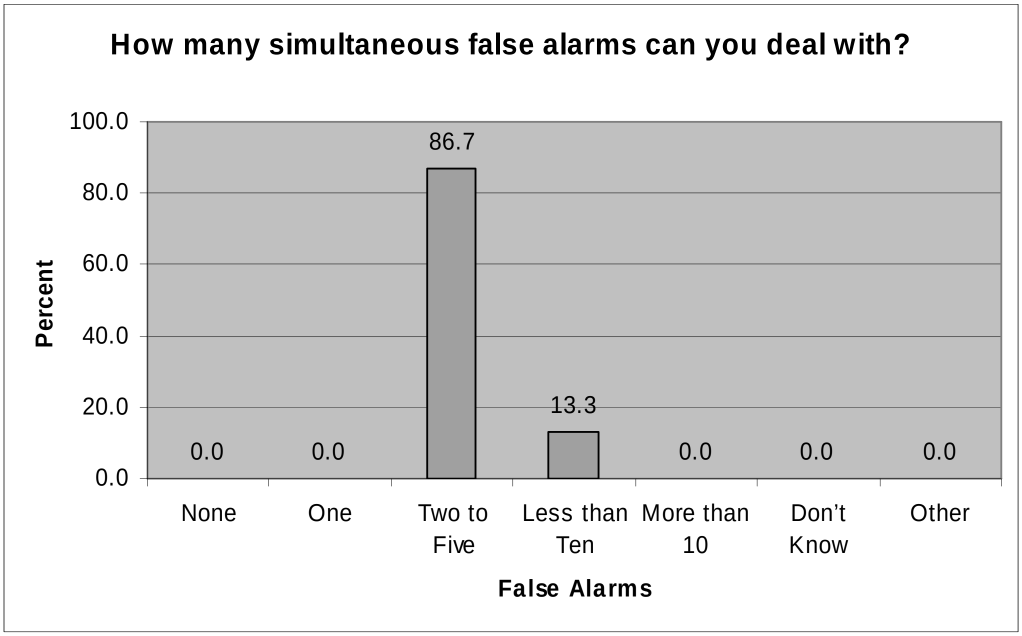


Figure 48. Operator Survey – Simultaneous False Alarms.

Figure 49 highlights the responses from the last 3 questions in the incident detection portion of the survey. Each respondent was asked to determine the acceptable number of false alarms, per station, per day, that they would accept in order to achieve a specified detection rate of 100, 75, or 50% percent. In Figure 49, the first bar in each series represents the response for 100 percent detection, the second bar for a 75 percent detection, and the third bar for a 50 percent detection.

The predominant answer, at any rate of detection, was two to five false alarms per station, per day. While this corresponds with the responses detailed in Figure 47 and Figure 48, the impact of this level of false alarms in an implementation with a significant number of stations must not be underestaimated.

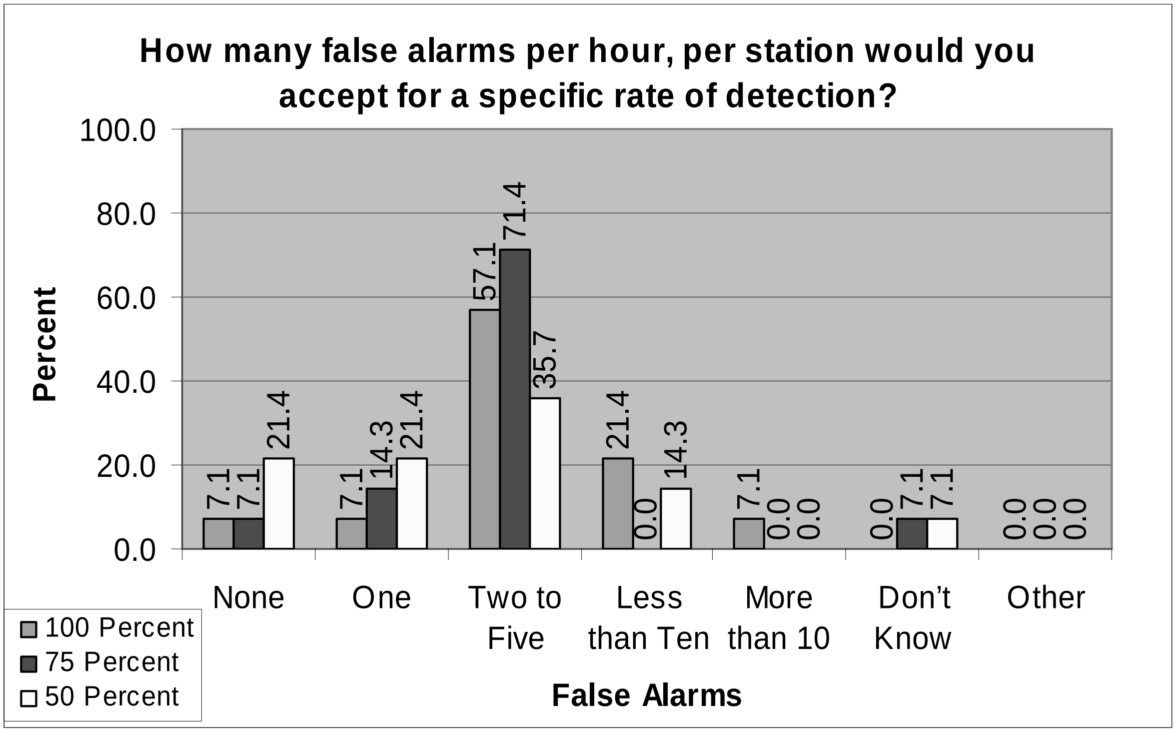


Figure 49. Operator Survey – False Alarms Per Detection Rate.

CHAPTER 8 – RECOMMENDATIONS FOR IMPROVEMENTS

The following recommendations are made in support of improving not only the incident detection algorithm in use by TxDOT's ATMS, but also to improve the data manipulation process critical to establishing accurate thresholds. These recommendations are Deliverable P2 of the project.

- ***<u>SCU File Naming Convention</u>*** **–** The file naming convention for the SCU hourly data files poses significant limitations for efficient handling of large amounts of data. The filename does not currently contain a date portion of the file, resulting in files being overwritten on a weekly basis. To avoid that situation and preserve the data for future use, operators must manually access and save the data files on a weekly basis, a time consuming step that is prone to errors. The inclusion of a date stamp in the filename that provide for Year, Month, and Day information would remove this impediment.
- ***<u>SCU Detector File Format</u>*** **–** The current format of the hourly SCU data files utilizes a time stamp at the beginning of each row followed by a sequential listing of all detector records. This results in files with extremely long record lengths which increases the parsing difficulty and time associated with any data manipulation. A revision of the file format to support a single detector and associated data per row would greatly reduce the data parsing difficulties and reduce the time necessary for data manipulation.
- ***<u>SCU Detector File Length</u>*** **–** The timeframe covered by a single SCU detector data file is one hour. This division of data results in a large number of files and requires significant data manipulation to combine multiple hours and says into a single comprehensive data archive. The use of a longer timeframe for recording detector information, such as a time period of 24-hours would greatly reduce the number of data files in use and significantly reduce the time required for data manipulation.
- ***<u>SCU Detector Data File Logging Structure</u>*** **–**The current data logging structure utilizes a single file containing all detector data for a given timeframe (one hour). Future improvement should consider the creation of multiple data files, one per detector, reporting to the SCU. As with previous recommendations, this would decrease the time and effort required to create comprehensive detector data archives.

This would be of significant benefit to the research community since most research activities look at data on a per-detector or station basis.

- ***SCU Detector Data Cleaning*** **–** The SCU currently utilizes no logic or rules for data cleaning or analysis. However, it is a known fact that loop detectors go off-line, misreport data, suffer from a loss of communication, or experience other situations which result in bad data. After the fact, this bad data is difficult to detect and can negatively influence data analysis procedures. A comprehensive set of rules for logic and consistency checks as well as a marking or replacement mechanism for bad data would improve the overall quality of the data archives and potentially result in additional improvements to those processes that utilize the data, such as incident detection.
- ***Database Detector Data*** **–** The current implementation of the SCU data archiving is a flat file repository. This mechanism has significant limitations, most notably for real-time analyses such as performance measurement. A critical improvement path for the ATMS product line is the design and incorporation of a 'live data' database. This database could provide a number of the improvement needs noted in the previous recommendations, as well as help to transform ATMS with a new set of real-time capabilities for improving transportation operations. This database structure would also be a significant benefit to the research community and greatly increase the capability of constructing and analyzing large data sets.
- ***Implement the CLC Algorithm*** **–** The CLC algorithm, as shown by the performance assessment under this research, can provide incremental but significant improvements to the incident detection capabilities that TxDOT experiences today. Specifically, the CLC algorithm shows a substantial reduction in false alarms, while providing an incremental improvement in incident detection rate.
- ***Utilize the CLC Threshold Procedure*** **–** The key to a successful implementation of the CLC algorithm is establishing effective thresholds. The procedure created in this research and documented in this report provides for a consistent and scientific process for establishing time periods and thresholds for the CLC algorithm. In addition, the specification of a logic flow for an automated tool provides TxDOT with the ability to incorporate this capability into future versions of ATMS.

CHAPTER 9 – CONCLUSIONS

INTRODUCTION

The trade-off faced by any incident detection algorithm is to maximize the efficient and timely detection of incidents while minimizing the number of false alarms. While this exchange is difficult even with detailed data-driven algorithms, it is challenging with simple comparative algorithms. A key component of maximizing the effectiveness of any algorithm is having an effective parameter-setting procedure in place.

CROSS-LANE COMPARISON ALGORITHM

Through the research on this project and summarized in this report, effective improvements can be made to the existing TxDOT algorithm. Whiles these improvements would require slight modifications to the design and software code of the TxDOT ATMS, the potential improvements in incident detection capabilities should be sufficient to prove worthwhile.

ASSESSMENT OF CAPABILITIES

A comprehensive assessment of the capabilities of the Cross-Lane Comparison algorithm was performed in this research. The results show substantial promise for decreasing false alarms and improving the detection rate over the performance characteristics of the existing algorithm. The caveat to this increase is that the thresholds must be refreshed on a yearly basis. While this is a time-consuming, data-intensive process, the results of this research convey a standardized, scientific process for creating thresholds. Portions of the process can be automated in desktop computer software, simplifying at least one component of the threshold selection process.

PROJECT DELIVERABLES

The four project deliverables within this research project show significant promise for providing incremental improvements to the ATMS incident detection capabilities and performance. By supplying a threshold creation process, ATMS implementations across the state now have a standardized method of creating thresholds. In addition, the assessment of capabilities should form the basis for a real-world understanding of not only the capabilities of the algorithm, but also the expectations of system operators. Finally, recommendations pertaining to data structure and file manipulation, as well as the detailed logic flow for an automation tool, should provide TxDOT with a roadmap for future improvements to the ATMS product line.

REFERENCES

1. Schrank, D., and Lomax, T. *The 2002 Urban Mobility Report*. Texas Transportation Institute., Texas A&M University, College Station, TX. June 2002. Access on March 19, 2003 at http://mobility.tamu.edu/ums/study/final_report.pdf.

2. Balke, K.N. *An Evaluation of Existing Incident Detection Algorithms*. Report No. FHWA/TX-93/1232-20. Texas Transportation Institute, College Station, TX., November 1993.

3. PB Farradyne, Inc. *Traffic Incident Management Handbook*. Federal Highway Administration, Office of Travel Management, Washington, DC, 20590, USA.

4. Payne, H.J., Helfenbein, E.D., and Knobel, H.C. *Development and Testing of Incident Detection Algorithms, Volume 2: Research Methodology and Detailed Results*. Report No. FHWA-RD-76-20. U.S. Department of Transportation, FWHA, Washington, DC, April 1976.

5. Tignor, S.C., and Payne, H.J. Improved Freeway Incident Detection Algorithms. In *Public Roads*. U.S. Department of Transportation, FWHA, Washington, DC, June 1977.

6. Collins, J.F., Hopkins, C.M., and Martin, J.A. *Automatic Incident Detection – TRRL Algorithms HIOCC and PATREG*. TRRL Supplementary Report 526, Transport and Road Research Laboratory, Crowthorne, Berkshire, 1979.

7. Masters, P.H., Lam, J.K., and Wong, J. Incident Detection Algorithms of COMPASS – An Advanced Traffic Management System. In *Vehicle Navigation and Information System Conference Proceedings*. Society of Automotive Engineers, Inc. Warrendale, PA., October 1991.

8. Dudek, C.L., Messer, C.J., and Nuckles, N.B. Incident Detection on Urban Freeways. In *Transportation Research Record 495*. Transportation Research Board, National Research Council, Washington, D.C., 1974.

9. Levin, M.L., and Krause, G.M. Incident Detection: A Bayesian Approach. In *Transportation Research Record 682*. Transportation Research Board, National Research Council, Washington, D.C., 1978.

10. Ahmed, S.A.., and Cook, A.R. Application of Time-Series Analysis Techniques to Freeway Incident Detection. In *Transportation Research Record 841*. Transportation Research Board, National Research Council, Washington, D.C., 1982.

11. Cook, A.R., and Cleveland, D.E. Detection of Freeway Capacity-Reducing Incidents by Traffic-Stream Measurements. In *Transportation Research Record 495.* Transportation Research Board, National Research Council, Washington, D.C., 1974.

12. Stephanedes, Y.J. and Chassiakos, A.P. Applications of Filtering Techniques for Incident Detection. In *Journal of Transportation Engineering*. Volume 119, No. 1. American Society of Civil Engineers, New York, NY, 1993.

13. Persuad, B.N., Hall, F.L., and Hall, L.M. Congestion Identification Aspects of the McMaster Incident Detection Algorithm. In *Transportation Research Record 1287.* Transportation Research Board, National Research Council, Washington, D.C., 1990.

14. Willsky, A.S., Chow, E.Y., Gershwin, S.B., Greene, C.S., Houpt, P.K., and Kurkjian, A.L. Dynamic Model-Based Techniques for the Detection of Incidents on Freeways. In *IEEE Transactions on Automatic Control.* Volume AC-25, No. 3. Institute of Electrical and Electronic Engineers, Piscataway, NJ, 1980.

15. Karim, A. and Adeli, H. Incident Detection Algorithm Using Wavelet Energy Representation of Traffic Patterns. In *Journal of Transportation Engineering*. Volume 128, Issue 3. American Society of Civil Engineers, Reston, VA, May 2002.

16. Chang, E.C.P. and Wang, S. Improved Freeway Incident Detection Using Fuzzy Set Theory. In *Transportation Research Record 1453*. Transportation Research Board, National Research Council, Washington, DC, 1994.

17. Stephanedes, Y.J. and Liu, X. Artificial Neural Networks for Freeway Incident Detection. In *Transportation Research Record 1494.* Transportation Research Board, National Research Council, Washington, DC, 1995.

18. Hellinga, B. and Knapp G. Automatic Vehicle Identification Technology-Based Freeway Incident Detection. In *Transportation Research Record 1727*. Transportation Research Board, National Research Council, Washington, DC, 2000.

19. Carvell,J.D., Balke, K., Ullman, J., Fitzpatrick, K., Nowlin, L., and Brehmer, C. *Freeway Management Handbook*. Report No. FHWA-SA-97-064. U.S. Department of Transportation, FWHA, Washington, DC, August 1997.

20. Gordon, R.L., Reiss, R.A., Haenel, H., Case, E.R., French, R.L., Mohaddes, A., and Wolcott, R. *Traffic Control Systems Handbook.* Report No. FWHA-SA-95-032. U.S. Department of Transportation, FWHA, Washington, DC, February 1996.

21. Arceneaux, J., Smith, J., Dunnet, A., and Payne, H. Calibration of Incident Detection Algorithms for Operational Use. In *Traffic Control Methods*. Proceeding of the Fifth Engineering Foundation Conference. Engineering Foundation. New York, NY, 1990.

22. Park, E. S., S. Turner, and C.H. Spiegelman, “Empirical Approaches to Outlier Detection in ITS Data,” accepted for publication in Transportation Research Record, 2003.

23. Turner, S. M., R. E. Brydia, Jyh-Charn. S. Liu, and W. L. Eisele. "ITS Data Management System: Year One Activities," Final Report to Texas Department of Transportation. TX-97/72350-1. August 1997.

24. Stout, Thomas L., Jyh-Charn. S. Liu., R. E. Brydia, and C. M. Poe. "TransLink Information Database Requirements," Interim Report to Texas Department of Transportation. TX-97/2988-2. October 1996.

25. Mardia, K.V., Kent, J.T., and Bibby, J. M. Multivariate Analysis, Academic Press, New York, 1979.

APPENDIX

AUTOMATIC INCIDENT DETECTION SURVEY

Questionnaire on Desired Performance for Automatic Incident Detection

Project 0-4770 – Evaluation and Optimization of Automatic Incident Detection Algorithm Used in TxDOT Traffic Management Systems

Agency:

Contact:

Address:

Phone: **Fax:** **E-Mail:**

This survey is being conducted by the Texas Transportation Institute in support of a research project with the Texas Department of Transportation. The Principle Investigator of the research project is Robert Brydia, who can be reached at (979) 845-8140 or via email at r-brydia@tamu.edu.

The purpose of this task is to determine the desired performance level for automatic incident detection algorithms. This information will be useful to the project even if your TMC does not use the TxDOT ATMS software.

Participation is this survey is entirely voluntary.

This survey will take approximately 10-15 minutes to complete.

Definitions:

Incident – A non-recurring event on the roadway that has a negative effect on traffic flow, capacity, or normal operating conditions.

Incident Detection – A method of detecting incidents automatically, using software and data analysis.

Detection Time – The elapsed time from when an incident occurred on the roadway to when it is detected by the incident detection algorithm.

Detection Rate – The number of incidents that have been detected as a percentage of the total number of capacity-reducing incidents that occur on the system.

False Alarm Rate – How many times an algorithm incorrectly identifies an incident and issues an alarm when, in fact, no incident is present.

Contact Record

Full Interview:

Date: TTI Representative:

Time Started: Time ended:

TMC Background Information

1. Does your TMC / software HAVE the capabilities for Automatic Incident Detection?

Yes	No	Don't know

2. Does your TMC / software USE the capabilities for Automatic Incident Detection?

Yes	No	Don't know

3. How many stations are on your system?

   ______________________ Don't know

4. How many miles of roadway do you monitor?

   ______________________ Don't know

5. How many operators are typically on duty during NON-PEAK conditions?

   ______________________ Don't know

6. How many operators are typically on duty during PEAK conditions?

   ______________________ Don't know

7. What are the hours of operation for your center?

Weekday	______________________	Don't know
Weekend	______________________	Don't know

Automatic Incident Detection (AID)

8. If you do not use AID, please explain why you do not?

Not needed
Too complicated to set up
Too inaccurate
Don't know
Other (please provide explanation below)

9. What parameter does your incident detection algorithm use?

Speed
Volume
Occupancy
A combination of the above
Don't know
Other (please provide explanation below)

10. What is the time period that you use to collect data from the roadway?

Every 15 seconds
Every 20 seconds
Every 30 seconds
Every minute
Don’t know
Other (please provide explanation below)

11. What is the time period that your software uses in looking for incidents?

Every minute
2 minute rolling average
3 minute rolling average
Don’t know
Other (please provide explanation below)

12. Automatic Incident Detection algorithms are generally evaluated by looking at:

- Detection Time,
- Detection Rate, and
- False Alarm Rate.

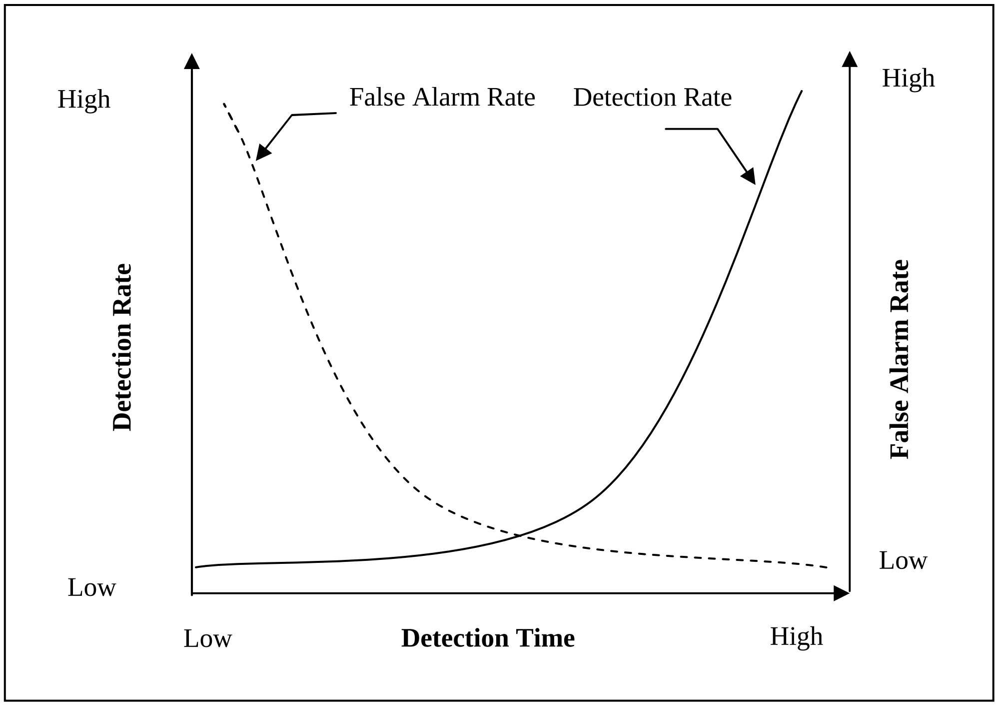
Which do you consider to be most important?

Detection Time
Detection Rate
False Alarm Rate
Don’t know

13. Do you understand the trade-off between these three parameters?

The graphic below illustrates this trade-off. Typically, algorithms that are set to detect a large percentage of incidents also create a large number of false alarms. In a similar manner, if the algorithm is set to minimize false alarms, a number of incidents are typically missed.

You can increase accuracy and decrease false alarms by taking longer to detect an incident, but that tends to have negative impacts on the traffic flow.



I understand these trade-offs
I do not understand these trade-offs
Don't know

14. How fast would you like to see an incident detected on the roadway?

30 seconds	60 seconds	90 seconds	120 seconds
Between 2 – 3 minutes			
Less than 5 minutes			
Don't know			
Other (specify) ______________________________			

15. If it meant that you could detect EVERY incident, what is the MAXIMUM amount of time you would be willing to wait for the algorithm to detect an incident?

Less than 1 minute
One minute
Two minutes
Three minutes
Three to five minutes
More than 5 minutes
Don’t know
Other (specify) ____

16. How many false alarms *per hour*, *per station* are you willing to put up with?

None
One
Two to Five
Less than Ten
More than Ten
Don’t know
Other (specify) ____

17. How many false alarms can you deal with simultaneously? (each one at a different station)

None
One
Two to Five
Less than Ten
More than Ten
Don’t know
Other (specify) ____

18. If it meant that you could **detect 100%** of the incidents on the roadway, what is the maximum number of false alarms *per hour*, *per station*, that you would be willing to deal with?

None
One
Two to Five
Less than Ten
More than Ten
Don't know
Other (specify) ______________________________

19. If it meant that you could **detect 75%** of the incidents on the roadway, what is the maximum number of false alarms *per hour*, *per station*, that you would be willing to deal with?

None
One
Two to Five
Less than Ten
More than Ten
Don't know
Other (specify) ______________________________

20. If it meant that you could **detect 50%** of the incidents on the roadway, what is the maximum number of false alarms *per hour*, *per station*, that you would be willing to deal with?

None
One
Two to Five
Less than Ten
More than Ten
Don't know
Other (specify) ______________________________

General Comments:

Please feel free to provide any other comments you may have concerning this survey or the use of automatic incident detection.