IMPROVEMENTS TO A QUEUE AND DELAY ESTIMATION ALGORITHM UTILIZED IN VIDEO IMAGING VEHICLE DETECTION SYSTEMS

A Thesis

by

MARSHALL TYLER CHEEK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2007

Major Subject: Civil Engineering

IMPROVEMENTS TO A QUEUE AND DELAY ESTIMATION ALGORITHM UTILIZED IN VIDEO IMAGING VEHICLE DETECTION SYSTEMS

A Thesis

by

MARSHALL TYLER CHEEK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Gene Hawkins
Committee Members ,	Yunlong Zhang
	Paul Nelson
	James Bonneson
Head of Department,	David V. Rosowsky

May 2007

Major Subject: Civil Engineering

ABSTRACT

Improvements to a Queue and Delay Estimation Algorithm Utilized in Video Imaging Vehicle Detection Systems. (May 2007)

Marshall Tyler Cheek, B.S., Iowa State University

Chair of Advisory Committee: Dr. Gene Hawkins

Video Imaging Vehicle Detection Systems (VIVDS) are steadily becoming the dominant method for the detection of vehicles at a signalized traffic approach. This research is intended to investigate the improvement of a queue and delay estimation algorithm (QDA), specifically the queue detection of vehicles during the red phase of a signal cycle.

A previous version of the QDA used a weighted average technique that weighted previous estimates of queue length along with current measurements of queue length to produce a current estimate of queue length. The implementation of this method required some effort to calibrate, and produced a bias that inherently estimated queue lengths lower than baseline (actual) queue lengths. It was the researcher's goal to produce a method of queue estimation during the red phase that minimized this bias, that required less calibration, yet produced an accurate estimate of queue length. This estimate of queue length was essential as many other calculations used by the QDA were dependent upon queue growth and length trends during red.

The results of this research show that a linear regression method using previous queue measurements to establish a queue growth rate, plus the application of a Kalman Filter for minimizing error and controlling queue growth produced the most accurate queue estimates from the new methods attempted. This method was shown to outperform the weighted average technique used by the previous QDA during the calibration tests.

During the validation tests, the linear regression technique was again shown to outperform the weighted average technique. This conclusion was supported by a statistical analysis of data and utilization of predicted vs. actual queue plots that produced desirable results supporting the accuracy of the linear regression method. A predicted vs. actual queue plot indicated that the linear regression method and Kalman Filter was capable of describing 85 percent of the variance in observed queue length data.

The researcher would recommend the implementation of the linear regression method with a Kalman Filter, because this method requires little calibration, while also producing an adaptive queue estimation method that has proven to be accurate.

DEDICATION

This thesis is dedicated to my mother. Her support and encouragement throughout my academic career provided me with the motivation and determination to succeed. Also, to Brooke and Jerry. I owe both of you more than I could ever repay.

ACKNOWLEDGEMENTS

The author would like to thank Dr. Gene Hawkins and Dr. James Bonneson for their support and guidance throughout this project. Their mentorship was essential to the success of this research. Their guidance and valuable input has provided me with the opportunity to develop my skills as a researcher and expand my abilities as an engineer.

Also, the researcher would like to thank the signal operations program at the Texas Transportation Institute under Dr. James Bonneson for their assistance in data collection and reduction, and for the ability to use data and equipment for this thesis.

Finally, the researcher would like to thank the members of his thesis committee: Dr. Gene Hawkins, Dr. Yunlong Zhang, Dr. Paul Nelson and Dr. James Bonneson, for always taking time out of their schedules to discuss the progress and creation of this research.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

Page

LIST OF TABLES

Page

INTRODUCTION

Video imaging vehicle detection systems (VIVDS) are steadily becoming the preferred method for detecting vehicles at signalized intersections. VIVDS are progressively replacing detectors such as inductive loop detectors at signalized intersections due to the high cost of maintenance and frequency of repair involved with non-VIVDS detection (*1*). There is a need for real-time queue and delay estimation of vehicles at signalized intersections, as often times, modern traffic signal controllers are able to use these real-time data in order to optimize intersection performance. Queue length estimates can provide a valuable indication to the traffic engineer as to roadway conditions, and can allow the engineer to assess the performance of a roadway. Accordingly, a queue and delay estimation algorithm (QDA) has been developed by researchers at the Texas Transportation Institute (TTI) in order to procure reasonable estimates of queue length and delay, while minimizing noise associated with measured queue length estimates collected by VIVDS hardware.

Current mathematical techniques used in the QDA involve a weighted average of previous and current estimates of queue length in order to procure output queue length estimates. However, the initial algorithm that was designed presents a mathematical bias, leading to estimates output from the QDA that are inherently low. This is due to the current logic used by the QDA intended to minimize the effects of errant and dropped detections. Errant data or data containing a high degree of variability offers little justification for the use of this video detection technique over other forms of detection. Therefore, the improvement of the QDA used with VIVDS is necessary before it can be relied upon in order to provide accurate estimates of queue length and delay.

This thesis follows the style and format of the *Transportation Research Record*.

It is the researcher's belief that improvements to the previous QDA using fundamental mathematical techniques will yield an improved estimate of queue length. The researcher's hypothesis is that improvements to the previous version of the QDA and newly implemented mathematical techniques will improve queue length estimation. The alternative hypothesis is that there will be no improvement when comparing previous versions of the QDA to the one created in this research as well as when compared to raw measurements obtained from the VIVDS hardware.

OVERVIEW OF NCHRP 3-79

The National Cooperative Highway Research Program (NCHRP) is the primary sponsor for this research as part of NCHRP 3-79. This project is entitled *Measuring and Predicting the Performance of Automobile Traffic on Urban Streets.*

The research in NCHRP 3-79 has two objectives. The first objective of this research seeks to investigate the feasibility of real-time traffic control detection systems and their ability to provide real-time solutions for applications including: adaptive control, traveler information, incident management, and system performance (*1*). Secondly, this research is intended to expose weaknesses in current *Highway Capacity Manual* (*HCM*) methods for estimating travel speeds (*2*). It is believed that factors such as arterial traffic volume, traffic signal offset, access point density, cross-section design, arterial weaving, and platoon dispersion may all be key factors in the determination of travel speed through an urban arterial roadway (*3*).

Within the two main objectives of NCHRP 3-79, a subtask calls for the identification of viable applications of real-time travel time and queue length measurement for the management of traffic flow (*3*). The research and analysis presented in this thesis intends to investigate the viability of utilizing VIVDS to measure queue length.

PROBLEM STATEMENT

A subtask for NCHRP 3-79 investigates non-intrusive methods for detecting vehicles and estimating performance measurements at signalized intersections using VIVDS (*1*, *3*). This subtask specifically investigates the performance of signalized intersections using queue length and delay as the primary MOEs. Thus, the QDA was developed in response to the objectives proposed in this subtask. The QDA estimates MOEs in real-time using current and previous measurements of queue length. However, this method of queue length estimation using the QDA was hindered by a bias in the mathematical procedure used to estimate MOEs (*1*). Also, researchers believed that there were more ways to estimate queue length that could result in more accurate estimates.

Due to the shortcomings of the initial versions of the QDA, it is desired to modify the QDA such that the estimation of queue length output from the QDA does not bias queue length estimates. The modification of the QDA must include the implementation of a mathematical technique that allows for real-time queue length estimation based on previous and current measurements, while maintaining an unbiased output. Eliminating this bias is essential for accurately estimating queue length, as well as estimating delay at a signalized approach. Additionally, it is believed that a mathematical technique that utilizes elements related to real-time measurement and estimates based on traffic queuing theory should provide adequate models for the creation of an improved QDA.

This thesis research intends to investigate mathematical procedures that satisfy the needs for real-time estimation and produce more accurate estimates of queue length than previous versions of the QDA. Furthermore, it is desired to develop a new version of the QDA that does not underestimate queue length.

OBJECTIVES

The research goal is to identify the best mathematical technique for minimizing the error in queue length estimation output from the QDA using VIVDS. This goal will result in

an improved estimate of delay, which is dependant upon an accurate estimate of queue length. The estimation of delay however, is not the primary focus of this thesis, as logic pertaining to the estimation of delay will not be altered from the previous version of the QDA. The specific objectives of this research are as follows:

- Evaluate various methods for minimizing error in queue length data collected using VIVDS and analyzed using the QDA.
- Determine which mathematical technique minimizes error with respect to queue length estimation using the QDA. Inherently, this approach should also determine which mathematical technique minimizes noise with respect to delay estimation at a subject approach.
- Implement the best mathematical technique for queue length estimation in the QDA.

SCOPE

This research applies to signalized intersections where VIVDS may be used for the detection of vehicles and used for the actuation of traffic signals. This research will focus on the improvement of queue length estimates, and will not focus on estimates of delay. The accurate estimation of queue length, will affect the calculation of delay used by the QDA. Therefore, it can be said that the goal of improving the estimation of queue length, will be accompanied by an improved estimate of delay.

This technique will be designed in a manner such that the functions utilized by the VIVDS hardware for queue length estimation can be implemented using any VIVDS processors or cameras that are deployed at an intersection. Lastly, this research does not intend to investigate aspects of real-time traffic control that may be possible using output from the QDA. However, it is the goal of this research to investigate the accuracy, and subsequent potential use of QDA estimates for future implementation for real-time traffic control application.

LITERATURE REVIEW

This section is intended to give an overview of the fundamental concepts and principles involved in the determination of a new queue and delay estimation algorithm. This section will be composed of the following sections:

- Queuing Theory
- Video Imaging Vehicle Detection Systems (VIVDS)
- Previous Research
- Previous QDA Description
- The Kalman Filter
- Summary

QUEUING THEORY

Often in transportation engineering, the number of vehicles demanding to use a facility is greater than can be serviced during a given interval of time. As a result, vehicles are stored or queued along a roadway until a time when they can be serviced, thus delaying their departure (*4*). Queuing theory for purposes of traffic and transportation engineering is generally classified as stochastic or deterministic. Stochastic queuing is associated with arrival and service rates that are probabilistic. That is, the rates of arrival and service are unknown for a given scenario. When the rates of arrival and service (departure) are known, this is deterministic. This scenario is typical of signalized intersections, and forms the basis for the analysis conducted in this thesis (*4*).

A signalized intersection is often characterized by queuing that when observed at the macroscopic level, vehicles that compose the queue arrive and are serviced at a continuous rate. This type of queuing is consistent with deterministic queuing. In its simplest form, deterministic queuing at signalized intersections occurs during

undersaturated conditions. This means that during a given cycle, all vehicles that are requesting service are serviced. Therefore, there is no overflow condition from one cycle to the next (*4*).

Figure 1 illustrates deterministic queuing in its simplest form. The triangles that appear under the arrival line in Figure 1 represent one cycle length and each triangle can be analyzed to provide an array of performance measures for a signalized approach. Specifically, the time duration of queue, length of queue, individual delay and total delay may be analyzed by examining the triangles beneath the arrival curve (*4*).

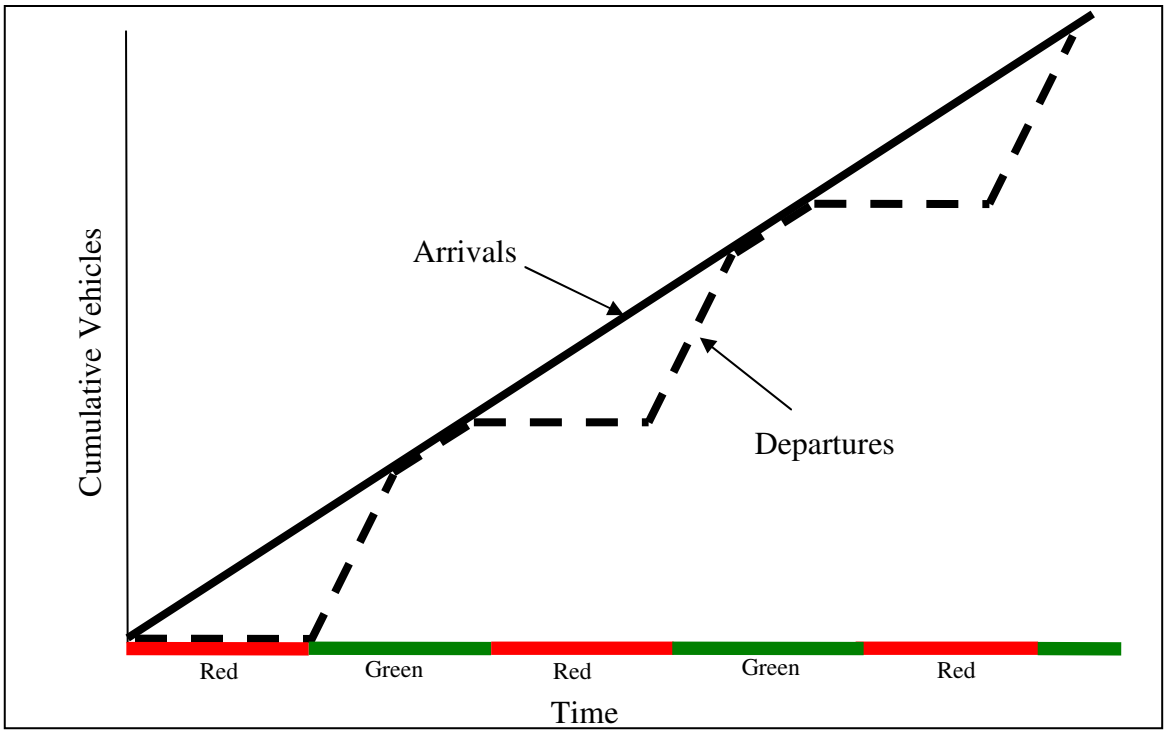


Figure 1 Deterministic Queuing Model for Signalized Intersections (*4*)

In Figure 1, the horizontal projection of the queuing triangle represents the time duration of queue. This period begins at the start of the red period and continues until the queue

is dissipated. This measure of performance is essential for the understanding of storage associated with a queue, and provides a basis for estimates of delay.

The queue length is represented by the vertical component of the queuing triangle. At the beginning of the red period, the queue length is zero. However, as the red period elapses, the queue begins to grow, as does the vertical component of the queuing triangle. Note in Figure 1 that the maximum length of the vertical component occurs at the end of the red period. Once the red period ends, the vertical component decreases. Eventually the queuing triangle is complete and the queue has dissipated.

Individual delay is represented by a horizontal distance slice across the queuing triangle. Notice that when using this model that the first vehicle to arrive during the red interval experiences the largest individual delay, and that vehicles that arrive there after experience a decreasing amount of delay until the queue has dissipated. Using this model, vehicles that arrive during the green interval experience relatively no delay, thus the departure rate (service rate) is equivalent to the arrival rate due to the fact that no queue is present.

Total delay is the aggregate of the individual delay experienced by all vehicles during a given traffic signal cycle. Therefore, total delay is the area of the queuing triangle. The determination of total delay is not only dependent upon the time component associated with queuing but is concerned with the actual length of queue during a cycle. The model presented as the deterministic queuing model in Figure 1 serves as the basis for queue length estimation and serve as the foundation for estimating measures of performance output from the QDA.

VIDEO IMAGING VEHICLE DETECTION SYSTEMS

Early development of video imaging vehicle detection systems (VIVDS) began in the 1970s in the United States and throughout the world (*5*). Today, VIVDS are becoming

an increasingly popular method for detecting vehicles at signalized intersections. VIVDS are primarily used for presence detection near the stop line of a signalized approach. VIVDS cameras are typically placed on mast arms or on mast arm poles. VIVDS technology utilizes a series of virtual video detection zones placed on the roadway through the use of specialized hardware typically consisting of cameras and controller cards.

The primary benefits of these systems reside in their cost efficiency and adaptability compared to alternative detection methods such as inductive loop detectors (*6*). Cost efficiency stems from the fact that VIVDS are designed to be non-intrusive. VIVDS can be implemented without physically disturbing the roadway. During VIVDS implementation, roadway surfaces are unaltered and do not require the physical construction of mediums for detectors to be deployed or embedded in. VIVDS have also shown to be cost effective where alteration of the roadway is imminent or where frequent reconstruction or maintenance of the roadway is necessary (*6*). VIVDS may require recalibration in these circumstances. This is in sharp contrast to inductive loop systems that may require the complete removal and reinstallation of hardware components. Additionally, the non-intrusive nature of VIVDS does not require the disruption of traffic in order for these systems to be implemented, therefore minimizing the delay to motorists and increasing the safety of detector installation at a signalized approach (*6*).

VIVDS typically consists of one or more camera units placed above the roadway that feed information to a unit consisting of a microprocessor. This hardware deciphers vehicle presence or passage and outputs traffic parameters, preferably in real-time (*7*). An illustration of a basic VIVDS hardware setup can be seen in Figure 2.

Figure 2 illustrates the process whereby the image from the camera is transferred to the VIVDS processing unit. Once image data are received by the processing unit, data are

digitized and formatted in a manner such that each point in the image is given a coordinate *(X, Y)*. These coordinates describe the energy, intensity and reflectivity of a scene at a given time, *t,* and the digitized image can be described by the aggregate function *I(X, Y, t)*. The aggregate image function, composed of thousands of points, is stored. Aggregate image functions when stored may be stored for each frame or maybe every *n*th frame. No matter the time step selected, the next frame analyzed, *I(X, Y, t+1)* and subsequent frames, are compared to a threshold value based on statistically calculated differences (related to energy, intensity and reflectivity), in the cumulative data set from previously stored images (*7*). Comparisons to the threshold value are made on a pixel by pixel basis. If the threshold is exceeded, the logic in the VIVDS processing unit interprets this as a detected vehicle.

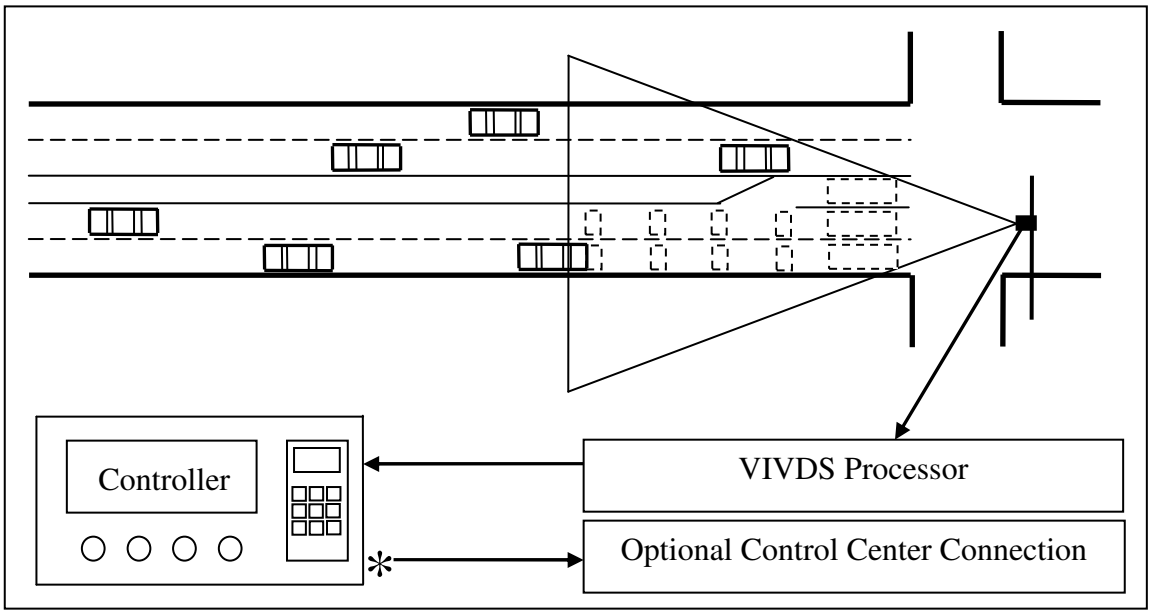


Figure 2 Typical VIVDS Components (*1, 7*)

In order for the VIVDS processing unit to narrow its selection criteria with respect to which fluctuations exceed threshold values, the processing unit must truncate the data set

such that background features are eliminated from analysis. VIVDS processors often have algorithms that distinguish background features from other features within the field of view. Background characteristics that are often targeted by these algorithms include transitions in the roadway surface between different pavement materials, snow packed and bare pavement, and roadway surfaces covered by shadows (*7*). Algorithms for distinguishing background features are often classified in terms of static or dynamic (transitional) phases. Algorithms that compensate for static features located in the background establish a reference signature at the initiation of the algorithm and thus distinguish changes in feature images as areas of detection, while the background should remain constant.

While static background imaging logic references a signature from the beginning of a time period, there also exist durations where background features tend to change or transition with respect to their properties. In instance of background feature transitioning may occur near dusk or dawn where the background lighting may change dramatically over a short period of time. The periods during which there is a transition in background features represent situations where dynamic background imaging logic is used by a VIVDS processing unit. These algorithms monitor background features by continuously updating the reference background signature image each polling interval against the vehicles that must be detected. This transitional period then leads to the static forms of imagery described previously. During this transition process, the feature detection criteria are automatically adapted to compensate for increases or decreases in feature properties. Lastly, these differences are analyzed, and the dynamic background imaging logic terminates and the static form of background logic once again initiates. During transitional stages, algorithm precision is critical, as the static form of the background logic depends upon this precision in order to establish its reference signature to compare imaging functions (*5*, *7*).

PREVIOUS RESEARCH INVOLVING VIVDS

Video imaging vehicle detection systems began to evolve in the 1970s as the United States, Europe, Japan and Australia sought ways to detect vehicles at low costs while maintaining at least the accuracy provided by inductive loop detectors (*7*). Early development of VIVDS was undertaken by the Jet Propulsions Laboratory (JPL) and was originally intended for tracking vehicles individually. While this aspect of the project proved to be challenging, researchers at the JPL were successful in developing algorithms for the detection of vehicles and measuring vehicle speed. The JPL named their system the Wide Area Detection System (WADS). Unfortunately, technological limitations involving video imaging and computer processing in the 1970s hindered the development of WADS and delayed substantial development of this technology until the 1990s (*6*, *7*).

VIVDS Application Research and Development

Video imaging vehicle detection systems utilize technology that has existed since the 1950s. While limited in scope with respect to the applicability of these systems, most early VIVDS systems were developed to provide presence detection on signalized intersection approaches. In the 1990s, research was conducted that investigated the feasibility of using VIVDS for purposes other than presence detection.

Research conducted by Michalopoulos et al. investigated the possibility of using VIVDS for more advanced traffic data measurements (*5*, *7*). This research measured speed, and travel time associated with vehicles traveling along a corridor. The results of this research showed that given the advances in VIVDS technology at the time, VIVDS measurements could be relied upon to make accurate measurements of speed and travel time. Results of this study showed that advanced VIVDS technology used in the study proved to be 95-97 percent accurate for measuring the speed of vehicles through a corridor. Furthermore, the results of this study showed that for simple presence detection, VIVDS performed just as well as loops during experimentation. Research

performed by Michalopoulos et al. also mentions the early realization and possible development of VIVDS technology for the purposes of producing quantitative queue estimates, as well as estimating measures of effectiveness such as delay, number of stops, and energy consumption (*5*). However, no documents could be found that presents results as to the findings of this type of research.

Most recently, VIVDS research has diverted from development of new algorithms for improving measurements and increasing the scope of VIVDS measurement capabilities. Instead, research has focused more on VIVDS camera positioning and calibration techniques. This type of research aims to improve the performance of VIVDS operations by minimizing the chance of error that may occur due to issues such as vehicle occlusion or maintenance functions that might be necessary as a result of poor camera positioning. Furthermore, calibration protocols that have been developed aim at allowing VIVDS cameras to perform at optimal levels, as well as allow for the possibility of automatically adapting camera positioning or field of view to account for prevailing weather and roadway conditions (*6*).

VIVDS Queue Research and Development

Most research involving VIVDS and queue length detection, involves the simple process of identifying when queues are present on a subject approach (*8*). These detection systems offer only a mechanism by which to qualitatively indicate whether a queue has formed. Research conducted by Rourke and Bell investigated the use of fast fourier transforms (FFT) in order to detect the formation of queues. This method was able to detect queue presence by defining an analysis window, then utilizing the frequency and power of the spectrum associated with images produced within this analysis window (*8, 9*). Furthermore, methods developed by Hoose utilized a full frame approach for queue detection (*9*). The full frame method is able to obtain an image no matter the position of the object on the screen. Hence, the full frame is utilized in the analysis, as opposed to the previous method that only analyzes objects within a specified analysis window. The

full frame method is then able to track the obtained image, in this case a vehicle, and is able to track the object through a succession of frames. Both of these methods have been used to establish queue presence detection algorithms. The queue presence information can then be passed to either a traffic signal controller, and an adaptive control feature can be initiated. Additionally, this information can provide a monitoring system for alerting traffic management personnel of roadway conditions (*8 ,9*).

Limited research pertaining to the quantitative measurement of queue length using VIVDS could be found. The researcher was able to identify only one application of VIVDS technology where researchers claim to have successfully implemented VIVDS to estimate the length of a traffic queue. In 1995, the Institution of Electrical Engineering in Great Britain published a paper entitled *Real-time Image Processing Approach to Measure Traffic Queue Parameters* (*10*). The objectives of this research were intended to quantitatively establish measurements in real-time pertaining to traffic queue length.

The algorithm utilized by the authors of this paper consisted of two components, motion detection and vehicle detection. The motion detection algorithm described in this paper is essentially the same process by which standard VIVDS detectors operate. This process involves the comparison of consecutive frames. While applying noise and background filters, the algorithm is capable of distinguishing differences in vehicle location between the two frames. Thus, if imaging properties associated with vehicles surpasses a specified threshold, a detection event is recorded. The second algorithm, vehicle detection, incorporates edge detection. Edge detection utilizes a technique that analyzes the boundaries of objects that appear in each frame of an image. These areas represent areas of substantial structural properties when viewing the full frame image produced by VIVDS. Edges are also known to be less sensitive to variations in ambient lighting. Thus edge detectors were believed by these researchers to be an optimal

method for detecting precisely where vehicles are located on a roadway by placing edge detectors where vehicle outlines are likely to exist (*10*).

The combination of motion and vehicle detection algorithms ultimately produces the estimate of queue length. The motion detection algorithm is used to distinguish areas of relatively little motion, to areas where substantial motion is present. Then, the vehicle detection algorithm serves as a refinement tool, whereby the areas of relatively little motion are analyzed by edge detectors to determine if vehicles are present within this region. If a queue is detected, a queue length is reported based on the calibration input by the engineers (*10*).

The findings of this research state that the queue length estimation technique implemented in this study result in an algorithm that is 95 percent accurate. This researcher questions these results, as the results presented show queue estimates rounded to the nearest 20 meter increment. Furthermore, baseline queue measurements show observations rounded to the nearest meter. From this researcher's experience, it would be very difficult to obtain 95 percent accuracy with respect to estimated queue length under these conditions, as rounding to the nearest 20 meter increment would introduce considerable error. Limited documentation of the actual experimental procedure could be located, nor could other documents that reference this technique. This method implements advanced imaging hardware that is not typical of a standard VIVDS setup. This distinguishes this research from that proposed in this thesis and those objectives prescribed in NCHRP 3-79, whereby a queue and delay estimation algorithm must be implemented in a generic way so that varieties of VIVDS hardware can use the QDA.

PREVIOUS QDA DEVELOPMENT

The queue growth period analyzed by the QDA includes the time period starting at the beginning of a red indication, and continues into the first few seconds of a green indication (*1*). The queue growth period during the green indication includes the time

period when vehicles continue to arrive, but the queue is dissipating at the front of the formation. During the initial formation of the queue, vehicles begin to accumulate and form a queue growing back from the stop-line. The QDA currently estimates the queue length at the end of every 10 second interval during the queue growth period. The queue length is reported based on the furthest activated detector that is occupied by vehicles in a traffic queue. However, due to sensitivity issues involving VIVDS hardware and erroneous detections that occasionally occur, the furthest reporting detector does not always provide the most reliable estimate of queue length as the detector that should be reporting the queue length may be malfunctioning. Therefore, current queue length logic used for queue estimation utilizes a weighted average based on previous and current estimates of queue length to ultimately produce a QDA estimated queue length. The following equation illustrates the weighting procedure currently utilized by the QDA.

$$Q_i = Q_{i-1}(1-f) + q_i f \qquad \textit{(Equation 1)}$$

where

Q_i = best-estimate of queue length during current period "i", ft,

Q_{i-1} = best estimate of queue length from previous period, ft,

q_i = detected queue length estimate from queue detectors during current period "i", ft, and

f = weight given to the current queue length estimate, $(0 < f < 1)$, (Empirically calibrated)

The use of the weighted average technique essentially introduces three estimates of queue length. A previous estimate of queue length is established from the previous QDA output estimate stored within the QDA output file. This estimate represents the best estimate of queue length from the previous period, Q_{i-1}. Next, the current estimate from the queue detectors, q_i, represents the value obtained from the furthest actuated

detector from the VIVDS system. Lastly, these two values are weighted, and the current QDA output estimate, Q_i, is produced. The weighting factor, f, is empirically determined under laboratory conditions, and provides an important step towards properly calibrating this model. This result produces an intermediate estimate (i.e., an estimate that is not a multiple of 50) of queue length. This type of estimate is believed to provide a more realistic, and potentially more accurate estimate of queue length than if q_i were used alone, which only exists in multiples of 50.

In addition to the estimation of queue length, the QDA makes an estimate of delay. Control delay is not estimated by the current version of the QDA, nor is the intent of any future versions of the algorithm to estimate this measure of effectiveness. Control delay cannot be effectively measured due to the fact that knowledge of the percentage of stopped vehicles would have to be measured. Due to the fact that this factor is difficult to measure using VIVDS, estimates of control delay are not produced.

Rather than an estimate of control delay, the QDA aims at making accurate estimates of stopped delay at the end of each signal cycle. As time during a signal cycle progresses, total delay is reported during each reporting period (i.e., each 10 second interval). This process begins during the start of the yellow phase, and terminates at the end of the green phase providing total delay estimates for each interval during this time. The total delay is then summed for the entire interval, then divided by the number of intervals during the cycle (*1*). The following equations illustrate the technique for estimating stopped delay (*1*):

$$d = \frac{1}{N_v} \sum_{1}^{n} D_i \qquad \text{(Equation 2)}$$

where

d = Average delay from previous cycle, sec/veh,

N_v = Count of vehicles discharging during the green interval, veh/cycle,

D_i = Total delay from period i, veh-sec, and

n = Number of reporting periods in the previous cycle.

The method used to compute N_v results from the detector placed closest to the stop line operating as a counter. The counting procedure is embedded within the QDA. The method for estimating delay utilizes estimates of queue length established during each interval. There are two components included in the estimation of total delay. Delay that is incurred during the red and yellow interval is the first component, and delay that is incurred during the queue clearance period during the initial moments of the green interval. Total delay during the red and yellow interval is computed as follows (*1*):

$$D_i = Q_i \frac{t_{rpt}}{L_{qv}} \qquad \textit{(Equation 3)}$$

where

Q_i = Best estimate of queue length during current period i, ft,

t_{rpt} = Queue reporting period (in these tests this value was 10 sec), sec,

L_{qv} = Distance headway between two vehicles in a stopped queue, ft. (Assumed to be 25 ft).

During the initial seconds of the green interval, the front begins to discharge. Due to this circumstance, the queue length is actually smaller than the distance from the stop line to the back of the queue. Therefore, the estimate of queue length is not effectively estimated by the ratio of the best queue estimate to distance headway between vehicles (i.e., the ratio Q_i/L_{qv}). Equation 3 must then be modified to reflect these phenomena. This scenario is corrected in Equation 4 and Equation 5.

$$D_i = Q_i^* \frac{t_{rpt}}{L_{qv}} \qquad \textit{(Equation 4)}$$

$$\frac{Q_i^* = Q_i - \left(t_i - \left[t_{pr1} - t_{pr}\right]\right)L_{qv}}{t_{pr}} \geq L_{qv} \qquad \textit{(Equation 5)}$$

where

Q_i^* = Adjusted best estimate queue length, adjusted to reflect departing vehicles at the front of the queue, ft,

t_i = Time of current reporting period i, (in these tests this value was 10 sec), sec,

t_{pr1} =Perception-reaction time of first queue driver, (Assumed to be 3.0 sec), sec,

t_{pr} =Perception-reaction time of remaining queued drivers (Assumed to be 1.0 sec), sec.

As previously mentioned, this form of queue length estimation utilized by the QDA introduces a bias to the output queue length. By using the term bias, the researcher is indicating that the error that is produced is typically due to estimates being lower than baseline measurements. The QDA output queue lengths are biased low due to the fact that the previous QDA output queue lengths are often smaller than currently detected queue lengths. Moreover, a "dropped" detection will often result in the QDA using the next smallest activated detector. This again results in an estimated queue length that is less than ideal, and would not provide a reliable estimate due to the tendency of queues to grow when comparing previous estimates to current detections obtained from VIVDS.

The researcher believes it is important to stress the fact that accurate estimates of queue length during the red phase are critical due to their use for estimating the dissipation of a traffic queue. The queue length estimates on red are also used to calculate delay, as illustrated by the mathematical procedures in this section. Ultimately, these estimates of

queue length on red and subordinate calculations could be used for real-time adaptive control of traffic signals. Therefore, methods for producing accurate queue length estimates during the red phase are intended to allow for the eventual optimization of traffic signal operations, and maximize the service of traffic on a subject approach.

KALMAN FILTERS

In 1960, the creation of a mathematical filtering procedure for the optimization of discrete-data linear filtering problems was published by Rudolph Kalman. The filter was designed to provide recursive solutions to multiple-input, multiple-output systems intended to find optimal solutions based on noisy outputs (*11*). The Kalman Filter minimizes the mean-squared error. In other words, it minimizes the squared difference between an estimator and the value in which the estimator is approximating. The appeal of the Kalman Filter involves this technique's ability to minimize error in real-time associated with a system's theoretical performance based on measured performance of the system collected at regular intervals. Furthermore, drastic improvements in computer technology around 1960 aided the widespread acceptance of the Kalman Filter for a multitude of applications and made this technique ideally suited for real-time estimation procedures (*12*).

The Kalman filter is designed to minimize the variance of the estimation error experienced during the output of a linear system. Accordingly, in order for a Kalman Filter to be implemented, the process must be described in linear terms (*13*). A linear system is simply the process that can be described by the following two equations involving the state equation (Equation 6), and the observed measurement equation (Equation 7) (*12, 14*):

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \qquad \text{(Equation 6)}$$

$$z_k = Hx_k + v_k \qquad \textit{(Equation 7)}$$

where

x_k = process state vector at time t_k,

A = matrix relating x_{k-1} to x_k,

B = matrix relating optional control input, u_{k-1}, to the state, x_k,

u_k = optional control input,

w_k = assumed to be a white noise sequence with known covariance, Q_k,

z_k = *vector measurement at time* t_k,

H = matrix giving the ideal noiseless connection between the measurement and the state vector at time t_k, and

v_k = measurement error, assumed to be a white noise sequence with known covariance, R_k.

It is important to note that in the previously described mathematical procedure, that the white noise sequences for the state equation and the measurement equation are assumed to be normally distributed with means of zero.

It is easier to think of the Kalman Filter as a predictor-corrector algorithm. In this two-step algorithm, the predictor portion consists of a "time update" function that projects the current state estimate ahead in time. Next, the measurement update (corrector portion), adjusts the predictor estimate by an actual measurement at that time (see Figure 3).

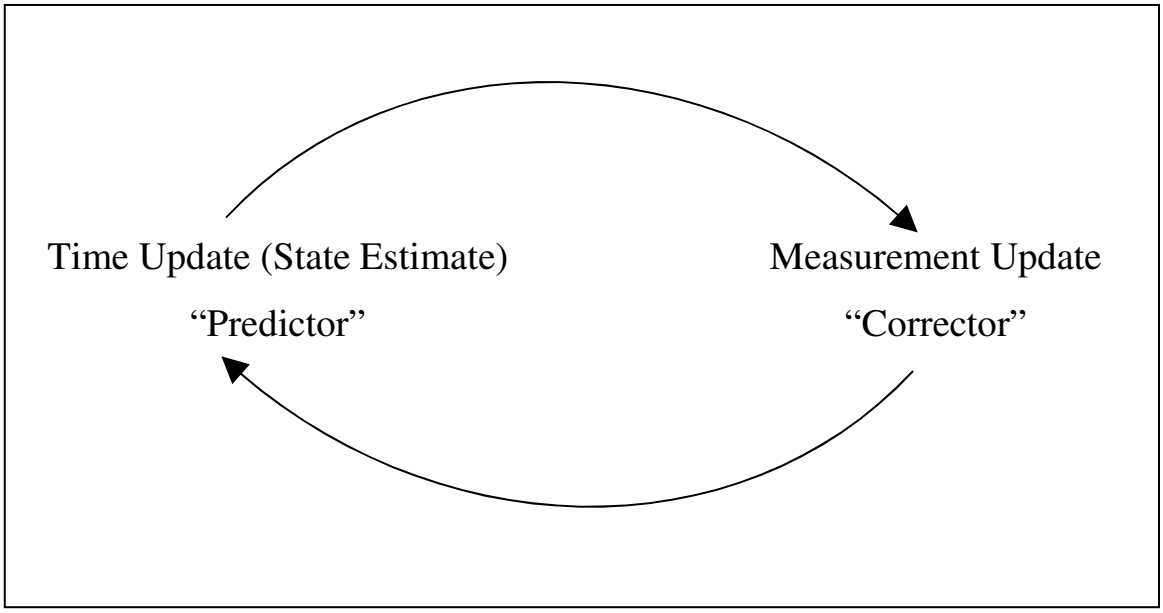


Figure 3 Kalman Filter Cycle (*14*)

To start the iterative process illustrated Figure 3, there must be some must be a set of initial conditions from which to begin. The terms Q_k and R_k, representing process noise covariance and measurement noise covariance respectively, are usually measured during offline calibration before the implementation of the Kalman Filter. The process and measurement covariance error terms can be determined by knowing the error terms w_k and v_k (*12*).

$$E\left[w_k w_i^T\right] = \begin{cases} Q_k & i = k \\ 0 & i \neq k \end{cases} \qquad (\textit{Equation 8})$$

$$E\left[v_k v_i^T\right] = \begin{cases} R_k & i = k \\ 0 & i \neq k \end{cases} \qquad (\textit{Equation 9})$$

where

Q_k = Covariance matrix associated with w_k, and

R_k = Covariance matrix associated with v_k.

While the measurement noise covariance, R_k, is generally easy to determine, the process noise covariance term, Q_k, can often prove difficult to obtain. This is due to the fact that it is often impossible to directly observe the process we are estimating. Therefore, Q_k must often times be estimated at the discretion of the researcher. The proper calibration of Q_k and R_k can lead to superior Kalman Filter performance. As such, care should be applied in determining these values (*12*).

The beginning sequences of the Kalman Filter requires that the process state equation, $\hat{x}_k$ be structured based on knowledge of an *a* priori state estimate, $\hat{x}_k^-$, where the "hat" denotes an estimate, and the super-minus represents the fact that a term is an *a priori* estimate. Additionally, the *a priori* error covariance associated with the *a priori* estimate is given by the term, P_k^-. These terms are determined by evaluating the following equations (*7*, *11*, *14*):

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \qquad \text{(Equation 10)}$$

$$P_k^- = AP_{k-1}A^T + Q_k \qquad \text{(Equation 11)}$$

where

$\hat{x}_k^-$ =A priori estimate of the process state vector,

P_k^- = A priori error covariance matrix associated with $\hat{x}_k^-$, and

Q_k = Process noise covariance.

Now that the time update equations have been established in Equations 10 and 11, the measurement update equations must be established. The first step of this process requires the calculation of the Kalman gain, K_k, also known as the "Blending Factor" (see Equation 12). The next step is to actually measure the process so that z_k can be

obtained, and a *posteriori* state estimate can be calculated (see Equation 13). The final step in the measurement update process is to make a *posteriori* error covariance estimate by evaluating Equation 14 (*12*, *14*).

$$K_k = P_k^- H^T \left(H P_k^- H^T + R_k \right)^{-1} \qquad \textit{(Equation 12)}$$

$$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H\hat{x}_k^- \right) \qquad \textit{(Equation 13)}$$

$$P_k = \left(I - K_k H \right) P_k^- \qquad \textit{(Equation 14)}$$

where

K_k = Kalman gain, "Blending Factor",

$\hat{x}_k$ = Posteriori of the process state vector, and

P_k = Posteriori estimate of the error covariance associate with the process state vector.

Once each phase has been completed (time update and measurement update), the *posteriori* state estimate is recycled to create a new *a priori* estimate of the process state vector. A graphical illustration of the Kalman Filter process can be seen in Figure 4.

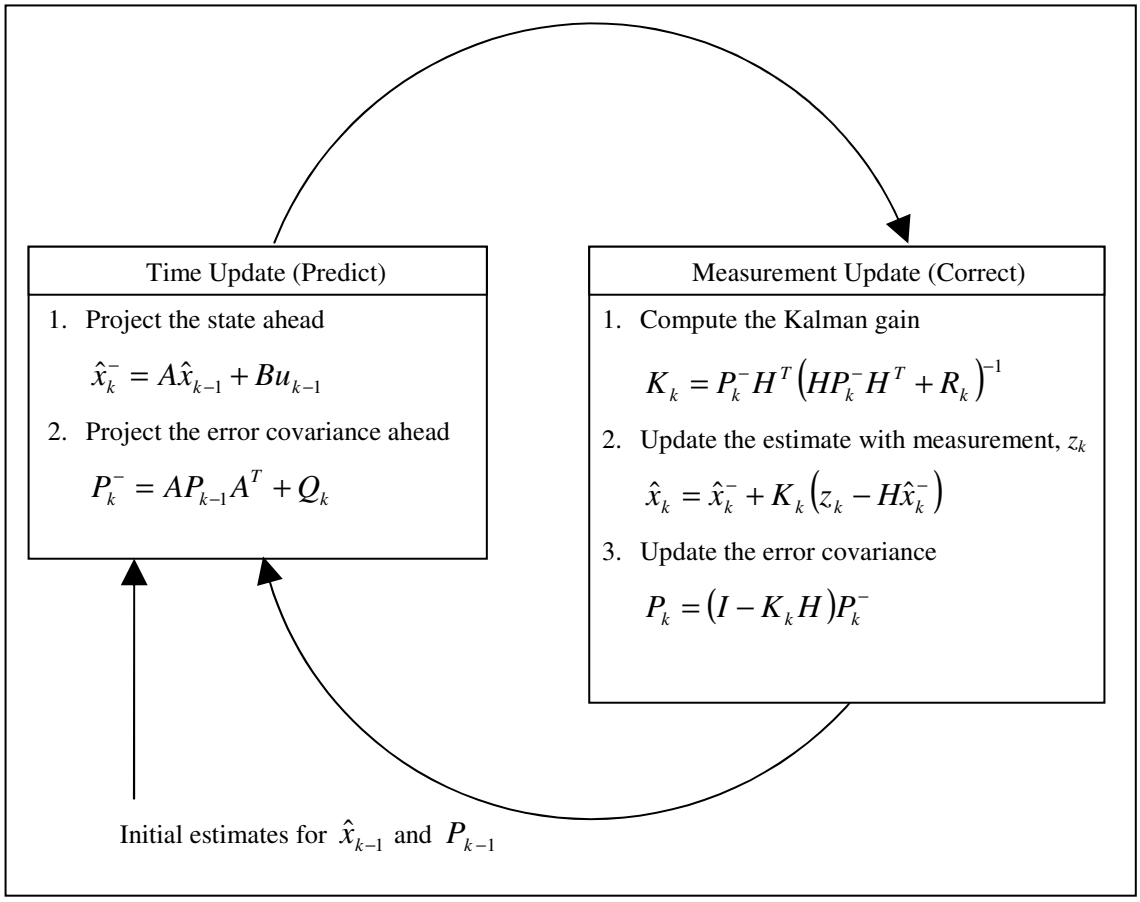


Figure 4 Kalman Filter Illustration (*14*)

SUMMARY

Recent research has not been directed towards improving VIVDS utilization for estimating MOEs. Instead, recent research has been directed more towards optimizing methods of placing VIVDS cameras so that data that are collected using these systems are able to better estimate MOEs. The previous QDA utilized a weighted average of queue estimation based on past and current estimates of queue length during the red phase of a traffic signal cycle. This method had the tendency to estimate queue length lower than what actually existed. However, methods proposed in this thesis intend to correct this low estimate by using advanced mathematical techniques including a Kalman Filter to intelligently combine estimates of queue length with current measurements of queue length, intending to produce an overall estimate of queue length that accurately reflects the current queue length condition at an approach of a signalized intersection.

METHODOLOGY

This section is divided into three parts. It is intended to give a detailed description of the procedures for conducting these experiments. This section describes the setup of field and laboratory experimentation and provides a basic understanding of the methodology for conducting experiments and the procedures used to analyze data. Analysis of the data obtained using these procedures is discussed in the subsequent section.

DATA COLLECTION PROCEDURE

Sites for QDA testing procedures were selected based on a number of criteria. First, the site must have VIVDS currently operating, and the operating agency must have the ability to allow researchers to use the VIVDS video output in order to record video data. Next, the site must also experience queues that occasionally extend beyond 400 ft upstream from the stop line. Lastly, the site must have adequate open space adjacent to the roadway such that video cameras could be placed along the roadway in order to record baseline "ground truth" queue length measurements.

The intersection of George Bush Drive and Wellborn Road in College Station, Texas met these criteria. This site offered ample space for setting up video cameras adjacent to the roadway. During this study, three types of data were recorded. First video data were recorded from the VIVDS camera. Second, the phase status of traffic signals was recorded using an industrial computer. Lastly, video data were recorded for the purposes of establishing baseline measurements involving queue length and vehicle counts on the subject approach. The studied site and experimental setup can be seen in Figure 5. The subject approach can be seen in Figure 6.

Figure 5 Test Site and George Bush Drive and Wellborn Road (*1*)

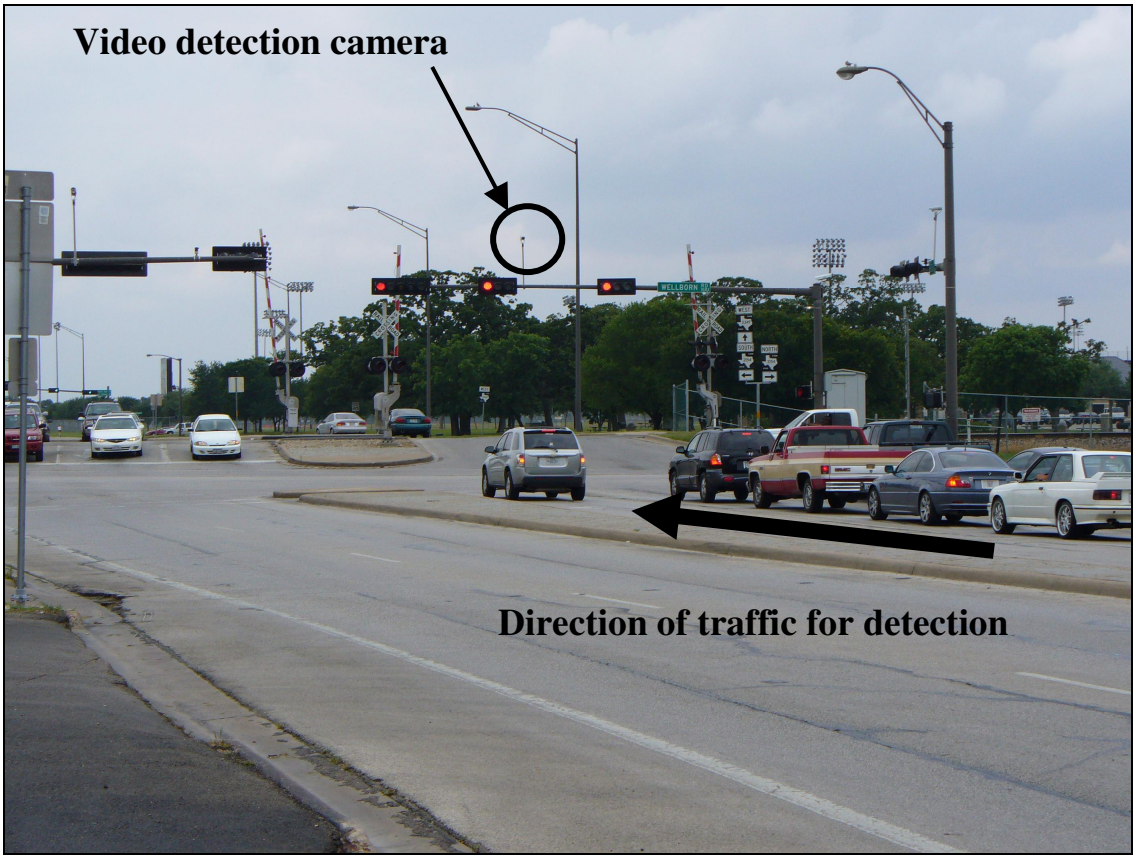


Figure 6 Approach on George Bush Drive and Wellborn Road

VIVDS Data Collection Procedure

The City of College Station allowed a research team to use the VIVDS video feed from the intersection of George Bush Drive and Wellborn Road to record video data. These data would then be reduced and used in the laboratory for the design, calibration and validation of the QDA. Notice in Figure 5 that the VIVDS camera is shown. This camera is mounted on a 5 ft riser arm and is located at an approximate height of 24 ft above the roadway. Video data were recorded for one approach at this intersection. Video data from the VIVDS camera was transformed from an analog signal output from the VIVDS camera and converted to a digital signal where it was then stored to an industrial computer. Later, this digital video data were transferred to DVD, where the data were replayed, data extracted and archived for future analysis.

Phase Status Data

The phase status of the indication displayed by the traffic signal was relayed from the traffic signal controller to the industrial computer. The thru indication reported to the industrial computer relates to the signal indication color displayed to those vehicles on the subject approach. This phase status was then combined with data associated with those data collected from the virtual detectors obtained by the VIVDS camera to estimate queue length at a given interval in the QDA. The data related to the phase status was recorded during experimentation and combined with VIVDS data during laboratory experimentation. Ultimately, it would be necessary for the phase status to be used by the traffic signal controller internally, combined with a software embedded version of the QDA such that queue length estimates could be made automatically by the traffic signal controller.

Baseline Data Collection Procedure

Video cameras placed adjacent to the roadway were able to capture queue formation as far as 400 ft upstream from the stop line on the subject approach. Video cameras were

placed adjacent to the roadway at an approximate distance of 280 ft from the roadway. An illustration of this can be seen in Figure 7. Video cameras recorded video data concurrently with video footage obtained from the VIVDS camera as well the traffic signal phase status data. This was necessary so that researchers could compare VIVDS (QDA predicted) estimates to those baseline estimates determined by the cameras adjacent to the roadway.

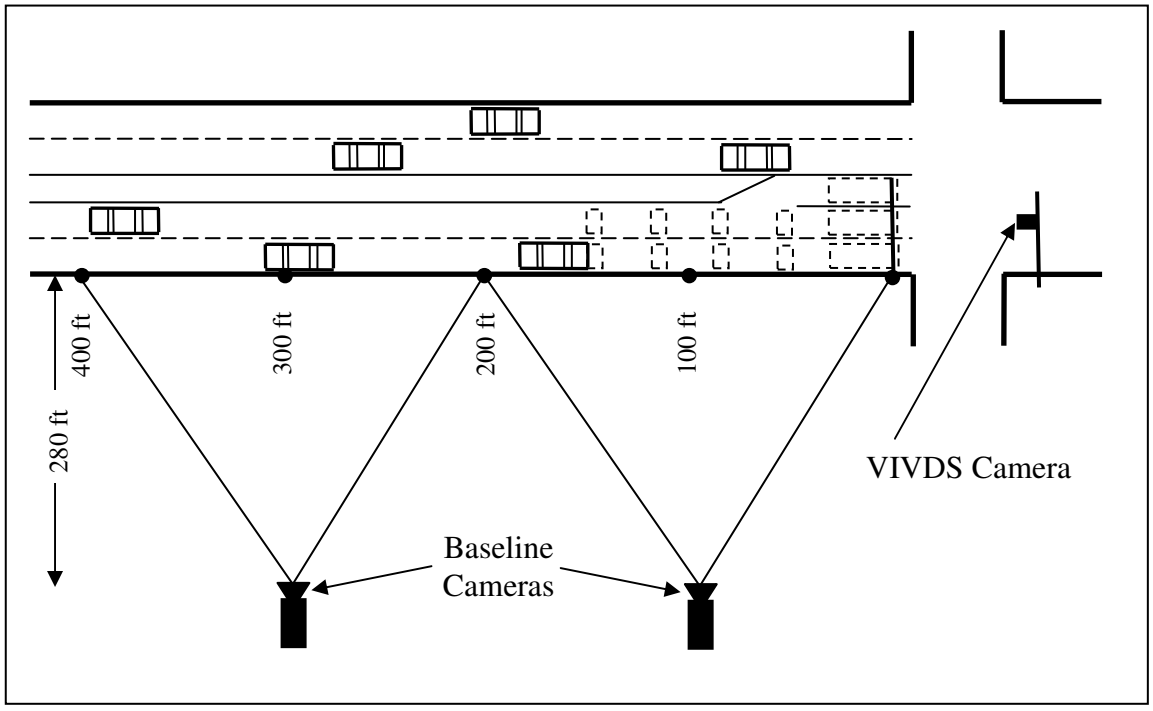


Figure 7 Baseline Data Collection Setup (*1*)

Once data collection concluded, data from the video cameras were then extracted manually. Data pertaining to queue length and vehicle counts were recorded every 10 seconds during video playback. These data then allowed the researcher to obtain baseline measures of effectiveness, including not only queue length, but baseline delay figures.

LABORATORY PROCEDURE

Once data were collected using the data collection procedure describing how data pertaining to VIVDS cameras, phase status and baseline measurements data were analyzed under laboratory conditions. VIVDS camera data were output utilizing the recorded DVD video footage of the subject approach and were fed to an *Autoscope "Rackvision"* VIVDS processing unit. It is believed that this procedure involving the use of recorded DVD video footage offers many advantages over conducting these experiments under field conditions. For instance, using recorded footage allows the researcher to notice the affects of small refinements in queue logic, detector design, or other experimental modifications. Accordingly, this procedure does not experience the type of random traffic fluctuations as would be experienced under field conditions, and offers consistent traffic patterns from which to compare one trial to the next. During the laboratory procedure, four hours of video footage was used to test different queue estimation techniques. Video footage will be used to validate the algorithm determined to produce the best results using the calibration test procedure.

The VIVDS processing unit contains an imaging file that was merged with the output VIVDS camera footage. The imaging file containing sensors designed by the researcher, created virtual detection zones on the VIVDS camera footage. Using these sensors, the QDA was able to procure estimates based on specified assumptions, design guidelines, and traffic engineering principles specified by the researcher.

As can be seen in Figure 8, video imaging data and phase status data are merged when the QDA estimates queue length in 10 second intervals. The phase status data alerts the QDA as to the current phase status, and allows the QDA to initiate or terminate QDA subroutines and algorithms for the estimation of MOEs during a particular phase during a cycle. The current version of the QDA takes measurements of queue length in 50 ft intervals, and modifies these measurements using the weighted average technique to produce an estimate of queue length. The proposed new QDA again takes

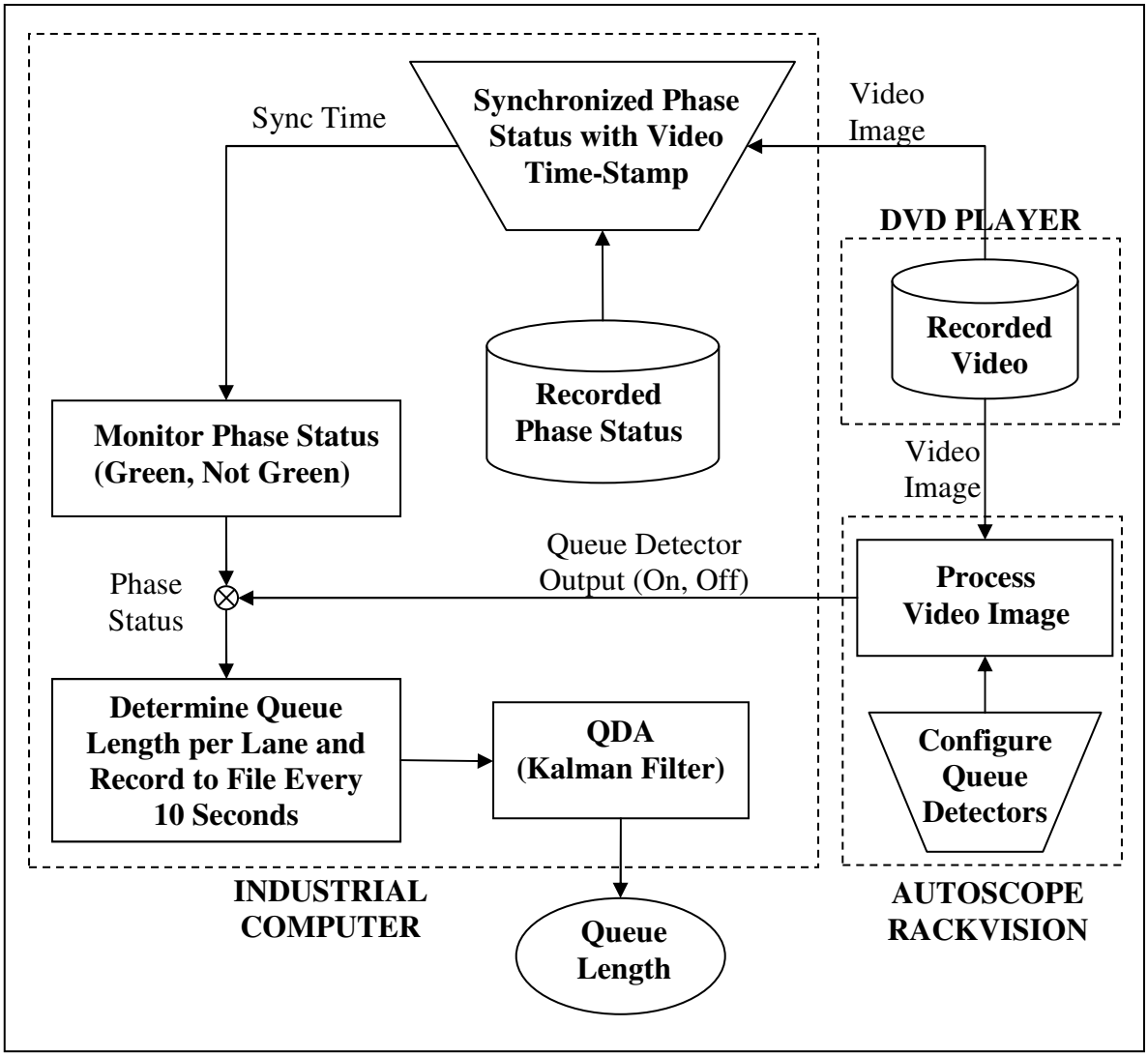


Figure 8 Hardware Setup for QDA Experimentation (*1*)

measurements, then procures an estimate of queue length based on deterministic queuing theory. The new version then modifies this estimate by applying the Kalman Filter before making a final output of queue length.

A typical VIVDS sensor layout for queue detection can be seen in Figure 9. Each horizontal bar in Figure 9 represents a detector placed at a pre-determined distance from the stop line. This setup consists of eight distinct detection zones associated with distances such that queue lengths of 50, 100, 150, 200, 250, 300, 350 and 400 ft from the stop line can be reported (*1*).

Notice in Figure 9 that the two nearest detectors to the stop line (those that report 50 and 100 ft) incorporate two detectors placed in close proximity to one another. The reasoning behind this detector design is that it is believed that this design adds increased reliability due to detector redundancy. A Boolean logic function "OR" joins the two detectors and if either is switched "on," the associated queue length is reported.

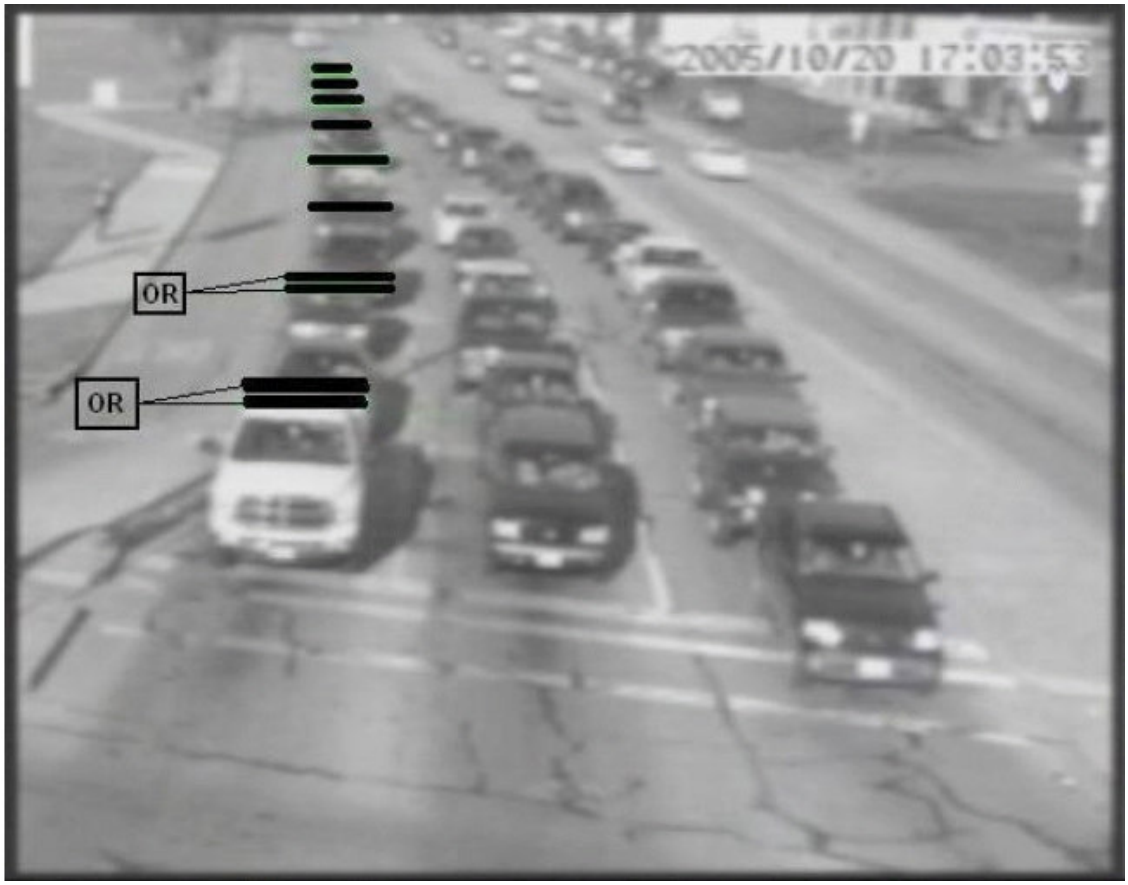


Figure 9 Typical VIVDS Setup for Queue Detection (*1*)

When vehicles begin to accumulate at a signalized intersection, the QDA is allowed to report queue length once a detector has been switched "on" for a certain period of time. Hence, detectors function on a delay and vehicles must be present on a detector for a specified duration of time in order to place a call. Once a detection zone reports that the queue length has reached a certain distance from the stop line, these data are sent to a laboratory computer, and its value is analyzed by the QDA.

Figure 9 shows sensors placed at gradual intervals from the stop line on the subject approach. Detectors in this figure are placed at intervals of 50 ft. However, detectors are not placed beginning at 50 ft from the stop line as one might assume. Detectors are actually placed beginning at a distance of 25 ft from the stop line and continue at intervals of 50 ft upstream (i.e., detectors are placed at 25, 75, 125, 175 ft, etc. from the stop line). Detectors are placed in this manner in order to improve the accuracy of measurements reported by the VIVDS hardware. Detectors are placed at the center of 50 ft zones. Through the researcher's experimentation with VIVDS hardware, this type of setup produces more accurate measurements than if detectors were placed at the ends of a 50 ft detection zone. If detectors had not been shifted, vehicles that extend in a queue that reaches 99 ft from the stop line would not be detected as 100 ft as intuition might suspect. Instead, a queue of 50 ft would be reported. However, when considering that the average of 50 and 99 ft is approximately 75 ft, it was decided to place detectors at this location. Detectors placed at this 75 ft location would then in fact report 100 ft, essentially reducing the magnitude of error in half. Table 1 shows the actual queue length associated with the eight detectors, as well as the reported queue length as used by the QDA.

Table 1 Detector Zone Queue Length Assignments

Detector Zone	Actual Queue Length, ft	Reported Queue Length, ft
1	25	50
2	75	100
3	125	150
4	175	200
5	225	250
6	275	300
7	325	350
8	375	400

ANALYTICAL PROCEDURE

Once data have been obtained, baseline measurements obtained from cameras positioned adjacent to the roadway and estimates of queue length obtained from the new QDA produced in this research were compared. The new QDA will be produced based on experimental results obtained from the analysis of a number of different methods for procuring queue length estimates, as well as the implementation of a Kalman Filter.

Estimates produced from the previous version of the QDA, as well as the new version of the QDA will be compared. This process will be referred to as the calibration experiments. Comparisons will be made using procedures introduced in the following section, *Data Analysis*. Estimates of queue length produced by the QDA will be compared by determining statistics involving the error of estimates, as well as the variability of these error values. Also, figures will be produced that illustrate variability in QDA estimate data. Lastly, illustrations will be produced that exhibit the accuracy of detectors as a function of distance from the stop line. These experiments will be conducted using *Microsoft Excel*™, and using known measurement output from the QDA, a simulation of real-time QDA queue estimation will be conducted using *Microsoft Visual Basic for Applications*™ (VBA) to simulate QDA queue length estimation.

Once the best technique for the estimation of queue length using a Kalman Filter is determined, this technique will be compared against results from queue estimates analyzed by the previous QDA using the weighted average technique. These experiments will be referred to as the validation experiments. The validation experiments will consist of five trial runs, whereby the best new estimation technique and weighted average method are both allowed to analyze each trial run. The results of this analysis will be aggregated, and results will be compared using similar statistical methods as the calibration experiments. Statistical tests and description of the data sets used for validation are included in the next section.

DATA ANALYSIS

The analysis used during this thesis uses four different types of data. The first type involves baseline data. Baseline data or ground-truth data were obtained from video cameras placed adjacent to the roadway. These data represent the true queue length during any one time interval. The next component is measurement data. These data are obtained directly from detectors placed at 50 ft intervals from the stop-line in the VIVDS system. The third component involves estimates that are produced from measurement data. Estimates are procured from a number of modeling techniques aimed at obtaining results that attempt provide estimates that are reasonably close to baseline queue lengths. The last component involves the utilization of the Kalman Filter to correct estimates and minimize error associated with estimated queue lengths. An illustration of the classification of data, and how the Kalman Filter utilizes these data is available in Appendix A.

The four types of data previously mentioned are reduced and either used directly to procure estimates used for analysis, or are used to compare to true values of queue length. The following subsections are included in this section:

- Queue Length Estimation Techniques
- The Kalman Filter Applied to Queue Estimates.
- Example Data
- Statistics Utilized
- Validation of the Improved QDA

QUEUE LENGTH ESTIMATION TECHNIQUES

A series of queue length estimation techniques were investigated by the researcher in order to create mathematical logic that produces the most accurate estimate of queue

length using measurement data from previous intervals. Pseudo-code for each of the queue estimation techniques are described in Appendix B. This section outlines the calibration portion of the data analysis. This analysis consists of three techniques for producing the slope or rate of growth of a traffic queue. The three methods are the incremental slope technique, moving slope technique and linear regression technique. These three methods are designed to produce the rate of growth based on previous measurements of queue length. This rate of growth will then be used to produce a current estimate of queue length through the use of a Kalman Filter.

For the determination of slope, the most current interval used for current queue estimation is the interval "*k-1*". This is due to the fact that it is not proper to use current measurements to produce a current estimate. This is important when considering that these two current values (measurement and estimate at time "*k*") will eventually be combined into one estimate using the Kalman Filter.

Incremental Slope Technique

The incremental slope technique utilized data from the previous two time steps relative to the current time step. This technique calculated the slope, or rate of queue growth, based on the two previous intervals, then output this value of slope for the eventual use in a Kalman Filter. The procedure presented in Equation 15 demonstrates the data requirements to predict the slope at each time step.

$$Slope = \frac{Q_{k-1} - Q_{k-2}}{T_{k-1} - T_{k-2}} \qquad (Equation\ 15)$$

where

Q_{k-1} = Measured queue length from the previous time step, ft

Q_{k-2} = Measure queue length from two time periods before, ft

T_{k-1} = Time at the previous time step, and

T_{k-2} = Time at the time step two periods before.

The inherent flaw with this method is that due to the fact that measurements are made to the nearest 50 ft, if a detector were to "drop" a queue detection (e.g., if a queue length was measured as 150 ft at time T_{k-2}, and at time T_{k-1} the queue length was measured as 100 ft), the slope would be negative. This has the potential of introducing a large quantity of error. However, the researcher believes that this method also holds some potential benefits, as this technique is adaptive. This technique is able to account for periods of slow queue growth (where the slope is moderately positive) and where queue growth is rapid (where the slope is highly positive). An illustration of the incremental slope technique is shown in Figure 10.

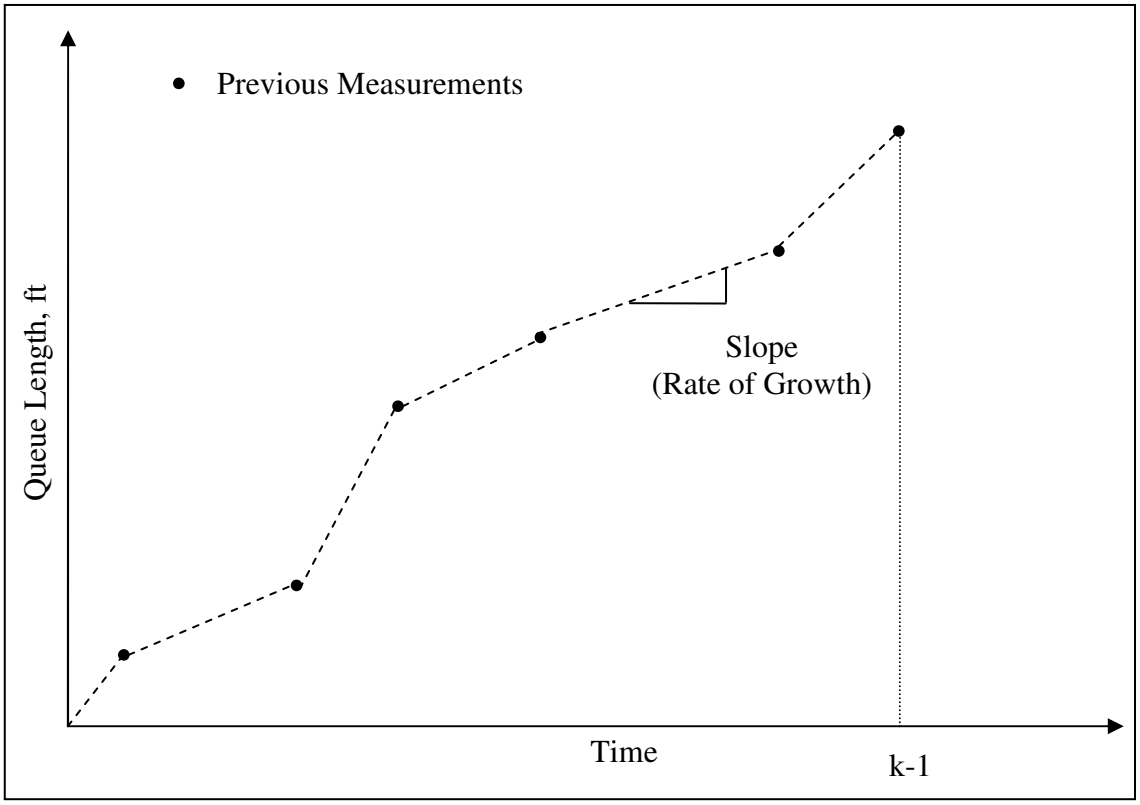


Figure 10 Incremental Slope Technique

Moving Slope Technique

The second method involved the calculation of the slope of the rate of growth of a traffic queue as measured referenced from the beginning of the red phase. This simplistic method required that two components be known. The first component is the time interval. It was important to obtain the time since the first polling interval during the red phase for a given cycle. Next, the queue length corresponding to the first interval after the beginning of the red phase and the interval prior to the current interval had to be determined. Together, these components could be combined by applying Equation 16. Equation 16 calculates the slope respective to the first time interval, t_0 and its corresponding queue length, assumed to be 0 ft. Notice in Equation 16 that the most recent time interval is shown as interval "*k-1*". This is because these values are for the previous interval time step, as it is desired to project estimates based on past measurements to the current time step.

$$Slope = \frac{Q_{k-1} - Q_0}{T_{k-1} - T_0} \qquad (Equation\ 16)$$

where

Q_{k-1} = Measured queue length at time interval *"k-1",* ft

Q_0 = Measured queue length during the initial time interval, ft

T_{k-1} = Time since the initial time interval at time T_0, sec, and

T_0 =Initial time, sec.

An illustration of this technique can be seen in Figure 11. During each time interval, the slope is recomputed with respect to the initial measurements, at time zero at the beginning of the red phase. After the determination of the slope, this value is used within the Kalman Filter as a growth factor included in that calculation for the current interval *"k"*.

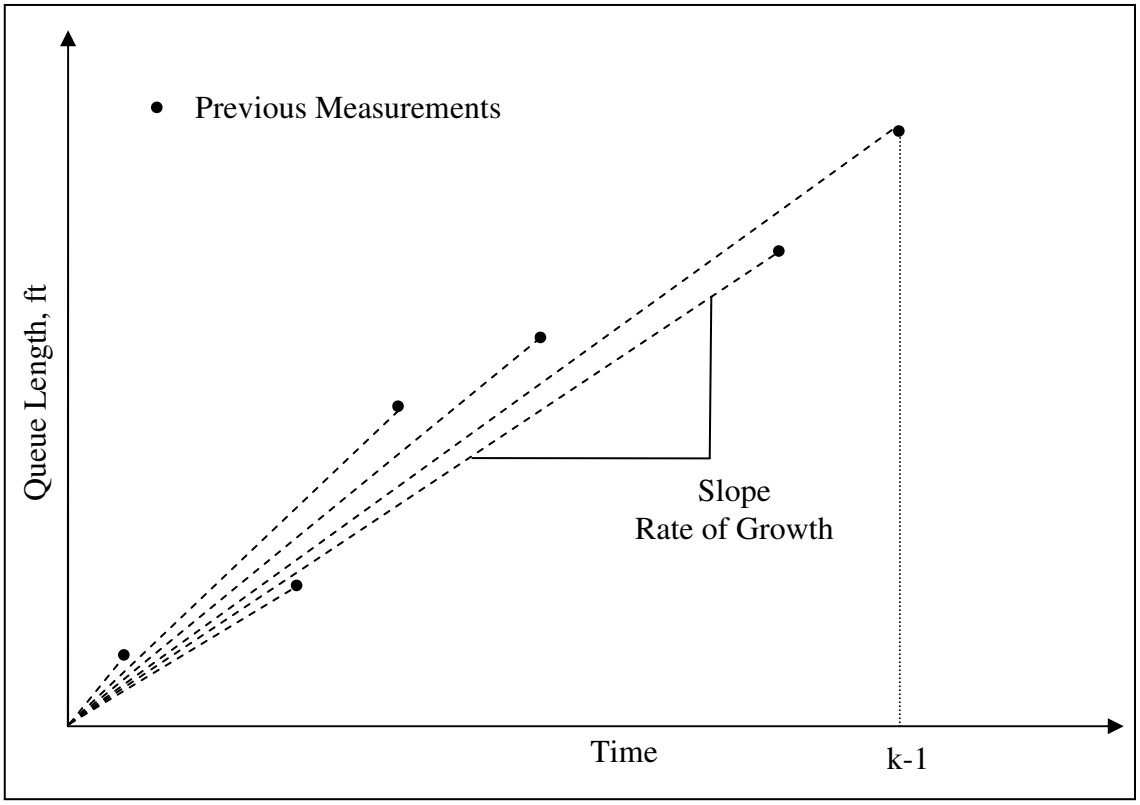


Figure 11 Moving Slope Technique

Linear Regression Technique

The next method for queue estimation attempted by the researcher involved real-time linear regression analysis of measured queue lengths. This method essentially took measured queue lengths recorded from previous intervals, and used these measurements to establish the rate of growth of the traffic queue. Queue lengths were recorded every 10 seconds. If during the red phase, 60 seconds had elapsed since the beginning of the red phase, this would mean that approximately six polling intervals were recorded by the QDA. The QDA would then take these six stored measurements and perform a linear regression analysis and would create a "best-fit" trendline corresponding to these points. The slope of this trendline would then serve as the growth rate to be implemented in the Kalman Filter. The growth rate is recomputed every interval, hence, another linear regression would be carried out at 70 seconds, using 7 measurements of queue length. Figure 12 illustrates the linear regression method.

The calculation of the linear regression best-fit line is recomputed during every polling interval (i.e., every 10 seconds). During this time, an equation is generated using the linear trend of measured queue data that are stored within the QDA. This equation is generated using simplistic statistical theory involving linear regression. Least squares estimation is designed to minimize the sum of the squares of the deviation from the best-fit line. Simple linear regression is often described as follows:

$$Y = \beta_0 + \beta_1 x + \varepsilon \qquad \textit{(Equation 17)}$$

where

β_0 = Intercept of the linear regression equation,

β_1 = Slope of the linear regression equation,

x = Current time analyzed by the QDA,

Y = Estimated linear regression, queue length, ft, and

ε = Describes the error in the fit of the model.

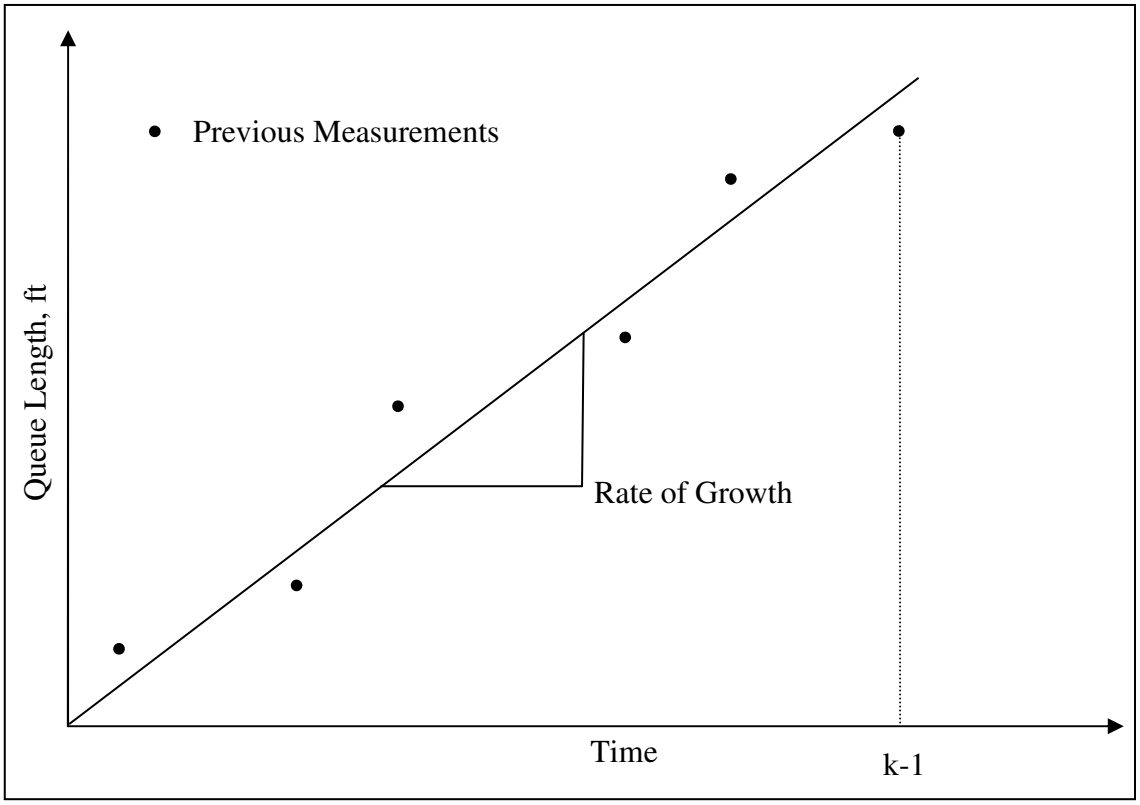


Figure 12 Linear Regression Technique

In Equation 17, the intercept, β_0 and slope β_1, can be determined as follows in Equation 18 and 19.

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \qquad \textit{(Equation 18)}$$

$$Slope = \beta_1 = \frac{\sum_{i=1}^{n} y_k x_k - \left(\sum_{k=1}^{n} y_k\right)\left(\sum_{k=1}^{n} x_k\right)}{\sum_{k=1}^{n} x_k^2 - \frac{\left(\sum_{k=1}^{n} x_k\right)^2}{n}} \qquad \textit{(Equation 19)}$$

where

y_k = Queue length measurement during timing interval *"k"*, ft,

x_k = Time of measurement during time interval *"k", sec,*

n = Number of previous measurements prior to current estimate,

$\bar{y}$ = Mean of previous queue measurements, ft, and

$\bar{x}$ = Mean of the time of measurement of previous queue lengths.

The construction of the linear regression equation is important for the understanding of the trends associated with queue data. However, it is the slope of Equation 17, β_1, that is of particular interest. The β_1 term provides a rate of growth that can be incorporated in the Kalman Filter to project the growth of estimates computed in the iterative calculations performed in that analysis.

One problem does exist with this type of calculation. When the second polling interval commences, or approximately 10-20 seconds after the initiation of the red phase, there only exists one data point from which to perform the linear regression analysis. This results in a β_1 value of zero. No growth rate is produced during this iteration. It is the

researcher's experience however, that during the initial stages of the formation of a queue, queue lengths tend to relatively small. When small queue lengths are present, detectors indicating 50 ft and 100 ft tend to be more accurate than those placed further from the queue. Therefore, the measurements recorded during the initial intervals are the only variable input into the Kalman Filter besides the parameters describing the variance of input models. It is believed that these measurements are accurate enough to be relied upon as the only means of determining queue length during the first interval, as queues are relatively small, and detectors placed at short distances from the stop line are likely reliable.

THE KALMAN FILTER APPLIED TO QUEUE ESTIMATES

The Kalman Filter process demonstrated in the *Literature Review* portion of this thesis illustrates the Kalman Filter in its purest form. The filter is applicable to a number of applications, including applications where multiple equations are necessary to describe the state of a system. The use of multiple equations is accounted for by the Kalman Filter's matrix operations which ultimately lead to a current state estimate of a system.

The process of estimating queue length described in the proceeding sections is accomplished by utilizing a single linear equation, whereby queue estimates are procured from previous queue measurements. Accordingly, the matrix operations accounted for in the theoretical description of the Kalman Filter drastically reduces. The Kalman Filter described in the *Literature Review* reduces from a system of linear equations, best solved through linear algebra processes, to a system described by scalar equations (i.e., a system incorporating 1 x 1 matrices).

The scalar Kalman Filter begins by obtaining values of Q_k and R_k, which represent the covariance of the estimation error and measurement error respectively. These values are obtained by taking offline measurements and estimates and comparing these values to baseline queue measurements. The measurement and estimation error that are produced

are then statistically analyzed, and the standard deviation of each error term obtained. The error covariance with respect to the measurement error and estimation error are determined as follows (*12*):

$$Q_k = (\sigma_{Estimation\ Error})^2 \qquad (Equation\ 20)$$

$$R_k = (\sigma_{Measurement\ Error})^2 \qquad (Equation\ 21)$$

where

$\sigma_{Estimation\ Error}$ = Standard deviation of the estimation error, ft, and

$\sigma_{Measurement\ Error}$ = Standard deviation of the measurement error, ft.

The measurement error is obtained through offline measurements. Offline calculations of the standard deviation of the measurement error were made for each method attempted. Offline standard deviation calculations were made using measurement error readings for an approximate 15 minute period for all red phases within this time. This time period was selected based on the researcher's experience, as well as the fact that of the entire data set, approximately 15-20 minutes was not used for the calibration tests, leaving 15 minutes as the time that could be used to determine offline readings. Equation 22 illustrates how the error of measurements was calculated compared to baseline values and Equation 23 demonstrates how the standard deviation of the error terms was determined.

$$e_k = q_k - q_k' \qquad (Equation\ 22)$$

$$\sigma_{Measurement\ Error} = \sqrt{\frac{\sum_{k=1}^{N}(e_k - \bar{e})^2}{N}} \qquad (Equation\ 23)$$

where

e_k = Measurement error at time *"k"*, ft,

q_k = Baseline queue length at time *"k"*, ft,

q_k' = Measured queue length at time *" k"*, ft,

$\bar{e}$ = Average measurement error, ft, and

N = Number of observations.

There are many methods for obtaining the estimation or process error covariance, denoted "Q_k". Of these techniques, there exists an approximation method proposed by Welch and Bishop. According to Welch and Bishop, acceptable results can be obtained if one "injects" enough uncertainty into the process via the selection of "Q_k"(*14*). Welch and Bishop go onto to state that measurements must be relatively accurate in order for this type of approximation to yield desirable results. Essentially, Welch and Bishop are making their assumptions based on previous knowledge of a process, and are relying upon measurements of the process to correct the estimation error they have selected.

Due to the fact that in this experimental procedure, measurement data are far from ideal, offline estimates during an approximate 15 minute time period will be analyzed in order to produce the estimated error term, Q_k. This term was produced through offline estimation, assuming that the standard deviation of queue estimates is 50 ft. As Welch and Bishop state in their description of the Kalman Filter, it is often common to begin the calibration of the error term Q_k by assuming a reasonable value for this input (*14*). The 50 ft value is an assumed value and is believed to be reasonable as it reflects the distance between detectors and closely resembles the quantities obtained for the standard deviation of the measurement error (see Appendix D). This offline estimation procedure was performed for each of the estimation techniques.

The assumed value of 50 ft for estimation error was utilized in offline testing for the approximate 15 minute duration for each model. The resulting estimates produced during the offline procedure were then compared to baseline queue values corresponding to the same time period of offline analysis. The estimation error was calculated using Equation 24, and the corresponding standard deviation of the error was calculated as shown in Equation 25. The standard deviation of the estimation error in Equation 25 was used in Equation 20 to ultimately produce the estimation error (Q_k) associated with each estimation technique that was used in the remaining four hours of data analysis (see Appendix D).

$$e_k = q_k - \hat{q}_k \qquad \textit{(Equation 24)}$$

$$\sigma_{Estimation\ Error} = \sqrt{\frac{\sum_{k=1}^{N}(e_k - \bar{e})^2}{N}} \qquad \textit{(Equation 25)}$$

where

e_k = Estimation error at time *"k"*, ft,

q_k = Baseline queue length at time *"k"*, ft,

$\hat{q}_k$ = Estimated queue length at time *"k"*, ft,

$\bar{e}$ = Average estimation error, ft, and

N = Number of observations.

When the standard deviation was calculated with respect to estimates for all three linear queue models, all produced standard deviations between 40 ft and 70 ft (see Appendix D). These standard deviation calculations result from the error produced when comparing estimates using different slope calculations (illustrated in the previous subsections) compared to baseline queue values for each time interval.

While "Q_k" is allowed to be a dynamic value that updates every time interval "k", the researcher has decided that "Q_k" be a static value that does not update. This decision was made due to the fact that a simple discrete Kalman Filter was utilized, and tuning "Q_k" requires very detailed data that is impractical given that the use of eight detectors for measurement data already severely limits the accuracy of the Kalman Filter, and a high degree of variance is already experienced.

Note that it is not necessary to compute the standard deviation of measurement error for each queue modeling method as the same measurement data were used in each trial (See Appendix D). However, because different measurement data were used between calibration and validation, another standard deviation value was computed for validation.

Once these values are determined, attention must be turned to the term H_k. The matrix H_k was described in the *Literature Review* as the connection between the measurement and state vector at a specific time, t_k. For the estimation of queue length, the measurement of queue length has a direct relationship to that which is output from the current state equation. Accordingly, this direct one-to-one ratio results in the following reduction associated with Equation 7 describing the observed measurement (see Equations 26 and 27 for reduction).

$$H_k = 1 \qquad (Equation26)$$

$$z_k = (1) \times x_k + v_k \qquad (Equation\ 27)$$

In Figure 4, the Kalman Filter begins with the *a priori* estimate and the error term. The terms $\hat{x}_{k-1}$ and P_{k-1} are both initially zero (see Equations 28 and 29). This simplifies the initial stages of the Kalman Filter process and results in the prediction portion of the filter to yield a perfect estimate, that is, the *a priori* estimate is assumed perfect, with no error associated with the measurement. This seems logical, as during the beginning

seconds of the red interval, it is common for no vehicles to be queued. While this assumption is highly dependant upon the definition of a queued vehicle, the researcher believes that for the purposes of establishing a reference point, the time at the beginning of the red phase where no vehicles are queued serves as a perfect estimate of queue length. During this instance, the time is zero and the queue length is zero. This assumption carries over for every iteration, until there is a non-zero measurement of queue length detected.

$$\hat{x}_0 = 0 \qquad \textit{(Equation 28)}$$

$$P_0 = 0 \qquad \textit{(Equation 29)}$$

Now that all of the parameters have been obtained, the recursive loop utilized by the Kalman Filter can begin. Figure 4 shows that initially, the Kalman Filter begins by inserting the values $\hat{x}_0$ and P_0. This allows for the evaluation of Equations 30 and 31. Recall that in the previous subsections of this section that the slopes of the three different models were calculated. This value is now used. The slope is inserted for the variable u_{k-1}, or for the first iteration this value is u_0. The 1 x 1 matrix, B, is the time step, or in this case 10 seconds. The quantity Bu_{k-1} is an optional term provided in the Kalman Filter. It is an optional control input intended to provide for the adaptation of an estimate from one time step to the next. In this equation there will always be a direct relationship between the Kalman Filter estimate of queue length, and the time update (predicted estimate). Therefore, the A term will be one.

$$\hat{x}_1^- = Ax_0 + Bu_0 \qquad \textit{(Equation 30)}$$

During the first iteration of the algorithm where there is a non-zero measurement of queue length, both the estimate from the previous time step, x_0, and the slope, u_0, are zero. Furthermore, due to the fact that during the first instance where there is a non-zero

measurement of queue length the term P_0 from the previous interval is zero, the error term, P_1, is equal to the error covariance, Q, associated with the estimate error (Equation 31).

$$P_1^- = P_0 + Q = Q \qquad (Equation\ 31)$$

During this time step and subsequent time steps, the equation for determining the Kalman gain in the corrector portion of the Kalman Filter significantly reduces.

$$K_1 = P_1^- H^T \left(H P_1^- H^T + R \right)^{-1} \qquad (Equation32)$$

$$\Rightarrow K_1 = P_1^- (P_1^- + R)^{-1} \qquad (Equation\ 33)$$

Similarly, the updated estimated output from the filter, $\hat{x}_k$ also reduces. In this thesis, the variable z is representative of the measured queue length for the current time step. The equation for determining the Kalman Estimate of queue length reduces similar to that of the Kalman gain, as the H term is equal to one, and can be ignored.

$$\hat{x}_1 = \hat{x}_1^- + K_1 \left(z_1 - H\hat{x}_1^- \right) \qquad (Equation\ 34)$$

$$\Rightarrow \hat{x}_1 = \hat{x}_1^- + K_1 \left(z_1 - \hat{x}_1^- \right) \qquad (Equation\ 35)$$

The last step of this initial iteration of the Kalman Filter concludes with the calculation of a new error covariance term, P_1. Notice that the "I" term reduces to a 1 x 1 identity matrix. This makes this scalar value equal to one.

$$P_1 = \left(I - K_1 H \right) P_1^- \qquad (Equation\ 36)$$

$$\Rightarrow P_1 = (I - K_1)P_1^- \qquad \textit{(Equation 37)}$$

This ends the initial iteration of the Kalman Filter. Calculations can now begin for the second iteration, for time t_2. Once the initial iterations are complete, the recursive nature of the Kalman Filter becomes apparent. Those estimates from the previous time step, t_1 are used in the new prediction portion of the Kalman Filter in the current time-step, t_2 (see Equations 38 and 39). Hence, a new estimate of queue length, given by $\hat{x}_2^-$, is produced based on the Kalman Filter output queue length from the previous time-step.

$$\hat{x}_2^- = A\hat{x}_1 + Bu_1 \qquad \textit{(Equation 38)}$$

$$P_2^- = P_1 + Q \qquad \textit{(Equation 39)}$$

Note that during this iteration, the estimate is not perfect, queue length and the corresponding error will not be zero during this or subsequent time-steps during the present red phase.

The termination of the Kalman Filter occurs when the end of the red phase is acknowledged by the QDA. The improvements to the QDA proposed in this thesis primarily deal with improvements to the estimation of queue length during the red phase, as this phase is mostly likely to accrue queued vehicles at a subject approach during any given time. Accordingly, the researcher believes that the termination of the Kalman Filter at the end of the red phase is appropriate. Any remaining queue during the early seconds of the green interval is accommodated by existing logic incorporated in the QDA, whereby a projection based on queue estimates, and queue growth rates recorded during the previous red phase models the arrival and eventual dissipation of vehicles. An accurate estimate of queue length during the red interval is necessary in order to produce these projections during green. Furthermore, the calculation of delay discussed

in the *Literature Review* shows the dependence upon queue estimates. These dependencies justify this research, and makes accurate estimate of queue length on red of utmost importance.

EXAMPLE DATA

Data that were analyzed during this thesis involved data collected under laboratory conditions. Data output from the QDA in the laboratory setup resemble data presented in Table 2. Two cycles worth of data are shown in this table. Note that these data are merely sample data that illustrate output from the QDA. The columns for *Hours*, *Minutes*, *Seconds* and *Measured* are all collected from the VIVDS hardware. Data contained in the *Estimated* and *KF Estimate* column are determined through the estimation and analytical techniques presented in this section. Data in these two columns are determined internally by the QDA. The column containing *Baseline* queue lengths are used for comparison with estimates produced by the QDA. The complete description of how the QDA processes data is available in Appendix B. Pseudo code for the QDA during the red phase is presented in this appendix.

Table 2 Example Data Processed by the QDA

Time				Queue Length			
Clock	Hours	Minutes	Seconds	Baseline	Measured	Estimated	KF Estimated
14:35:30	14	35	30	0	0	0	0
14:35:40	14	35	40	0	0	0	0
14:35:50	14	35	50	0	0	0	0
14:36:00	14	36	0	0	0	0	0
14:36:10	14	36	10	5	0	0	0
14:36:20	14	36	20	5	0	0	0
14:36:30	14	36	30	5	0	0	0
14:36:40	14	36	40	5	0	0	0
14:36:50	14	36	50	28	100	100	100
14:37:00	14	37	0	58	50	50	50
14:37:10	14	37	10	58	50	0	26
14:37:20	14	37	20	58	50	17	34
14:37:30	14	37	30	138	100	25	65
14:37:40	14	37	40	135	100	70	86
14:37:50	14	37	50	135	100	90	95
14:38:00	14	38	0	135	100	100	100
14:38:10	14	38	10	170	200	105	155
14:38:20	14	38	20	129	0	0	0
14:38:30	14	38	30	0	0	0	0
14:38:40	14	38	40	0	0	0	0
14:38:50	14	38	50	0	0	0	0
14:39:00	14	39	0	0	0	0	0
14:39:10	14	39	10	61	50	50	50
14:39:20	14	39	20	40	50	50	50
14:39:30	14	39	30	62	50	50	50
14:39:40	14	39	40	90	100	50	76
14:39:50	14	39	50	118	150	100	126
14:40:00	14	40	0	118	150	155	152
14:40:10	14	40	10	115	150	177	163
14:40:20	14	40	20	82	100	186	140
14:40:30	14	40	30	40	0	0	0
14:40:40	14	40	40	0	0	0	0
14:40:50	14	40	50	0	0	0	0

STATISTICAL ANALYSIS

A series of statistical analyses were performed on data generated by the QDA. The generated data were compared using a number of mathematical techniques so that the accuracy and precision of QDA generated data compared to baseline data could be determined. The first step towards determining the accuracy of the new QDA using linear models and a Kalman Filter involved determining the error of each observation compared to baseline queue lengths obtained from the cameras placed adjacent to the roadway. A simple method for determining the error is accomplished by calculating the error for each time interval (see Equation 40). The researcher would desire to have as low of a value of error as possible for each time interval.

$$Error_i = q_k - \hat{q}_k \qquad (Equation\ 40)$$

where

q_k = Baseline (ground truth) queue length, ft, and

$\hat{q}_k$ = Estimated queue length from a linear model and Kalman Filter, ft.

Statistical Analysis

The first statistic computed was the average error of estimated queue length predictions. The average error of the estimated queue lengths is one of the most telling statistics due to the ability to see whether data are biased higher or lower than target (baseline) values. It would be desired to obtain average error values that are close to zero, as this would indicate that data are normally distributed about target values. Average error was computed as shown in Equation 41.

$$\bar{e} = \frac{\sum_{k=1}^{n} (q_k - \hat{q}_k)}{n} \qquad (Equation\ 41)$$

where

$\bar{e}$ = Average error, ft, and

n = number of observations in the data set.

The absolute average error of the entire data set was computed using the calculated error from any one given interval, then dividing the sum of all of these absolute error terms by the number of observations. While this statistic is simplistic, it is a valuable benchmark to be used in subsequent statistical analyses. The intent of this calculation is to give a sense of the magnitude of the error expected for each time interval. The mathematical representation for absolute average error is as follows:

$$\bar{e}_{abs} = \frac{\sum_{k=1}^{n} \left| q_k - \hat{q}_k \right|}{n} \qquad \textit{(Equation 42)}$$

where

$\bar{e}_{abs}$ = Absolute average error, ft, and

n = number of observations in the data set.

The third statistic used to compare the different modeling techniques involves the determination of the standard deviation of the error. The standard deviation of the error allowed the researcher to have a better understanding of the spread of the data, as well as the range of accuracy of a particular technique. Equations 43 and 44 show the procedure for determining the standard deviation of the error.

$$s^2 = \frac{\sum_{k=1}^{n} (e_k - \bar{e})}{n-1} \qquad \textit{(Equation 43)}$$

$$s = \sqrt{s^2} \qquad \textit{(Equation 44)}$$

where

s^2 = Variance of the error of the data set, ft^2,

e_k = Error of an interval *"k"*, ft,

$\bar{e}$ = Average error, ft, and

s = Standard deviation of the absolute error of the data set, ft.

The next series of statistics will be comprised of commonly used analysis of variance (ANOVA) statistics. The first analysis of variance statistic will be the sum of the squared error of the prediction error. The second will be the root mean squared error. This ANOVA will be accomplished by using the following statistical procedure demonstrated by Equation 45 and Equation 46. Equation 47 shows the calculation of the degrees of freedom.

$$SS_E = \sum_{i=1}^{n} (q_i - \hat{q}_i)^2 \qquad \textit{(Equation 45)}$$

where

SS_E = Sum of the squared errors, ft^2,

q_i = Baseline value of queue length, ft, and

$\hat{q}_i$ = QDA predicted value of queue length, ft.

$$RMSE = \sqrt{\frac{SS_E}{df}} \qquad \textit{(Equation 46)}$$

$$df = n - 2 \qquad \textit{(Equation 47)}$$

where

$RMSE$ = Root mean squared error, ft,

df = Degrees of freedom, and

n = Number of total observations during analysis.

The sum of the squared error is believed to give an overall picture of the quantity of error that is experienced when using a particular modeling technique. The queue estimation method that produces the smallest quantity for the sum of the squared errors and root mean squared error will provide strong evidence for the implementation of that particular method in the QDA.

Graphical Statistics

This thesis will supplement the statistical analysis with graphical statistics to gauge the performance of the new QDA over the previous version. The primary graphical statistic will be a predicted vs. actual queue plot of the Kalman Filter estimated data compared to the baseline data. The predicted vs. actual queue plot is useful for determining if two data sets are of the same data distribution. The predicted vs. actual queue plot will be setup such that Kalman Filter estimated values will be plotted on the horizontal axis, while baseline values are plotted on the vertical axis. If the data are of the same distribution, a linear trend should be evident in the predicted vs. actual queue plot. The researcher is primarily concerned with two values associated with these plots. The first value is the coefficient of determination. This value will explain how well the model is able to explain variance in actual observed (baseline) data. The second is the resulting best-fit trendline equation. The closer the slope of this equation is to "1", and the closer the intercept value is to "0", the lesser the amount of bias will be with respect to a particular queuing model.

The hypothesis for this graphical statistic would be that estimated queue length output form the new QDA would be of the same distribution as baseline queue length values.

Therefore, a linear trend should be evident. Alternatively, estimated queue length will not be of the same distribution as baseline data, and a non-linear trend would emerge from the predicted vs. actual queue plot.

VALIDATION OF THE IMPROVED QDA

Once the best method for queue estimation was determined, the resulting algorithm was used to reanalyze the video footage. The new method for estimating queue length will be selected from one of the three methods described in the *Queue Length Estimation Techniques* subsection. It was desired to select a method for estimating queue length that provides the lowest average error from baseline queue lengths, and demonstrates the least amount of variability. Additionally, graphical data were used as a secondary consideration for selecting the best linear queue estimation model.

The selected linear queue estimation model and associated Kalman Filter technique was then implemented in *Microsoft Excel* using a *Visual Basic for Applications* macro. This application was executed similar to the application used to determine which linear model was best during calibration testing. The selected queue estimation technique was then used to reanalyze the video footage from George Bush Drive and Wellborn Road. Due to the difficulties of collecting queue data, and the requirements for also collecting baseline queue video footage, the same set of data were reanalyzed. Despite the fact that the same data set was used for validation, there still exists some degree of randomness associated with the VIVDS hardware. Due to the randomness presented by VIVDS hardware, it was decided to allow each algorithm to analyze the video footage five separate times. During each of these five trial runs, each algorithm analyzed the video data from George Bush Drive and Wellborn Road for a four hour duration. Additionally, during each trial run, video data were allocated such that values for estimate and measurement error could be determined.

Once the best linear model application was executed and allowed to analyze the data set, results were produced using similar statistical methods used to produce results obtained during calibration. During the validation proceedings, the best linear model was compared with results output from the previous QDA using the weighted average technique. Results were compared by determining the average error of estimates, absolute average error of estimates, the standard deviation of the estimation error, the sum of squared errors, and the root mean squared error. Graphical analyses consisting of predicted vs. actual queue plots were also used to evaluate queue models and techniques. These results should provide the researcher with enough evidence to support the implementation of one model over the other for estimating queue length during the red phase.

Due to the methods used to calibrate the VIVDS hardware and the ways in which this hardware adapts to different lighting conditions and contrast features, no one experimental run will be identical. The VIVDS hardware is adaptive, and factors too numerous to make any one experimental test identical. Hence the need for algorithms for estimating queue length that are also adaptive and eliminate variability. Therefore, some randomness is introduced into the validation tests that will make the validation tests independent of the calibration tests.

The researcher realizes that the use of the same data set for calibration and validation procedures does introduce a problem. However, due to time and budgetary constraints, collecting a new data set from a different intersection was not feasible. The researcher would eventually like to obtain video data from a different intersection and validate queue estimation algorithms.

RESULTS

The results presented in this section are based on the analyses described in the previous section. Results were based on thorough evaluation of the three linear traffic queue modeling techniques, as well as the inclusion of a Kalman Filter applied to each technique. Queuing models were analyzed and their results are presented in the following sections. Not only will results be presented in this section, but their interpretation and impact to this research will be discussed. Results will culminate in the determination of a definitive model for using VIVDS for quantitative queue length estimation, and should provide the best method for estimating queue length during the red interval of a traffic signal cycle.

QUEUE LENGTH MEASUREMENT RESULTS

The queue length measurement results are based on the raw measurements obtained from the detectors drawn and input to the VIVDS hardware. These measurements are based on the output obtained from detectors oriented such as in Figure 9. These measurements, rounded to the nearest 50 ft increment, were expected to show poor accuracy compared to results produced when a combination of these measurements and advanced mathematical techniques is used. The distribution of the raw queue measurement data can be seen in Appendix C. These data represent QDA measurement output for all phases during all signal cycles during the four hour analysis period. Queue measurements were taken over an approximate four hour period every 10 seconds, resulting in 914 individual measurements of queue length during the red phase of all traffic signal cycles. However, the measurement of queue length during the red phase is of particular interest to the researcher. Figure 13 displays the distribution of queue measurements for red phases during the analysis period. This distribution gives the researcher an idea of the traffic conditions experienced during the analysis period. Also, this distribution should resemble baseline data distributions shown in Appendix C.

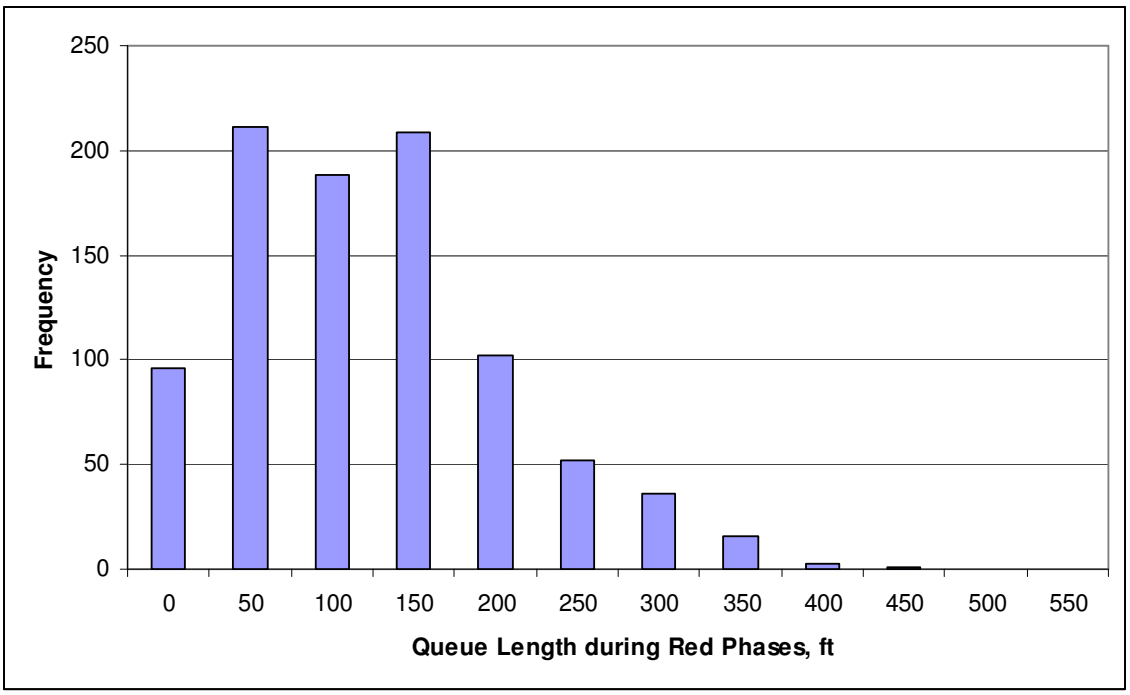


Figure 13 Histogram of Queue Length Measurements During Red Phase

These distributions are intended to give the researcher a graphical representation of the frequency with respect to detector activation, and to be able to qualitatively compare measurement detector activations to baseline queue length distributions during the analysis period.

Notice that nearly 100 of the measurements are accounted for when the VIVDS system detected a queue length of "0 ft". These measurements are most probably during intervals where towards the beginning of the red phase, a queue has not begun to form. During this time period, the QDA will not be executed. Instead, until a queue is sensed by a detector, the QDA outputs a queue length of zero. Also, when a queue was detected at the subject approach, measurements of 50 ft, 100 ft and 150 ft were most common. Measurements longer than 150 ft displayed a decreasing trend.

COMPARISON OF LINEAR QUEUE MODELS

The three linear queue modeling techniques each implementing a Kalman Filter for combining previous estimates of queue length with current measurements of queue length were analyzed. The results of these experiments are described in this subsection. The results are broken in two parts. The first part describes the calibration results, whereby the best linear queue modeling technique was determined. The second part takes the best linear queue modeling technique, validates these results and compares them to the results of the previous QDA that used the weighted average method. A discussion of all results is included at the end of this subsection.

Statistical Analysis Results

The primary statistics used for this analysis consist of the average error, absolute average error, the standard deviation of the error, the sum of the squared error, as well as the root mean squared error.. It was desired to produce an algorithm composed of one of the three modeling techniques plus Kalman Filter that produces results that minimize all

three statistical analyses. While this analysis only examines models and techniques to be used during the red interval of a traffic signal cycle, the algorithm that minimizes these five statistics will provide the most reliable method for projecting traffic trends during the clearance intervals as well.

Table 3 summarizes the results of the statistical analysis. Of the three linear models, results indicate that the linear regression technique produced the lowest values for the average error, absolute average error, the standard deviation of the error, the sum of the squared error, and the root mean squared error. Conversely, the incremental slope method yielded the highest values for these statistics.

Table 3 Statistical Analysis of Results for Calibration of Models

Statistic	Incremental Slope	Moving Slope	Linear Regression
Ave. Error, ft	11.292	9.003	7.666
Abs Ave Error, ft	32.331	29.443	27.578
Stdev. Error, ft	44.856	39.553	36.431
Sum of Squared Error, ft^2	1955689	1483684	1287219
Root Mean Squared Error, ft	44.856	39.553	36.431

Graphical Statistics Results

Graphical statistics for the incremental slope method show results that do not match the accuracy of the other techniques analyzed. A coefficient of determination value of approximately 73 percent is produced for the incremental slope method. This coefficient of determination value is not particularly desirable, and is believed to be rather low compared to other estimation techniques. The researcher will demonstrate in following subsections that in fact, the incremental slope technique performed the worst of the three models analyzed. Hence, the implementation of the incremental slope method is not advised.

Notice in Figure 14 that results show data that are consistently dispersed vertically corresponding to specific queue estimates. This trend is especially noticeable at distances of 50 ft, 100 ft and 150 ft on the horizontal axis (estimated queue length axis). This results due to the way in which this method computes the rate of growth of a queue. The method only utilizes data from the previous two intervals leading up the current measurement. For example, if a detector is first turned on at a time "*k-1*," and the measured queue length is 100 ft, then the estimate of queue length at time *"k"* is based on measurements from "*k-2*," which may be 50 ft, and the measurement of 100 ft at "*k-1*." The resulting slope is 50 ft divided by the 10 seconds separating these measurement intervals. If these previous two measurements are used to calculate the growth of the queue length during the time separating them, equivalent to 10 seconds, this produces an estimate of queue length of 150 ft. Considering that this scenario plays out numerous times in the data set, it is not surprising that only certain values of estimated queue length are possible depending on rate of growth of the queue, as well as the current measurement of queue length.

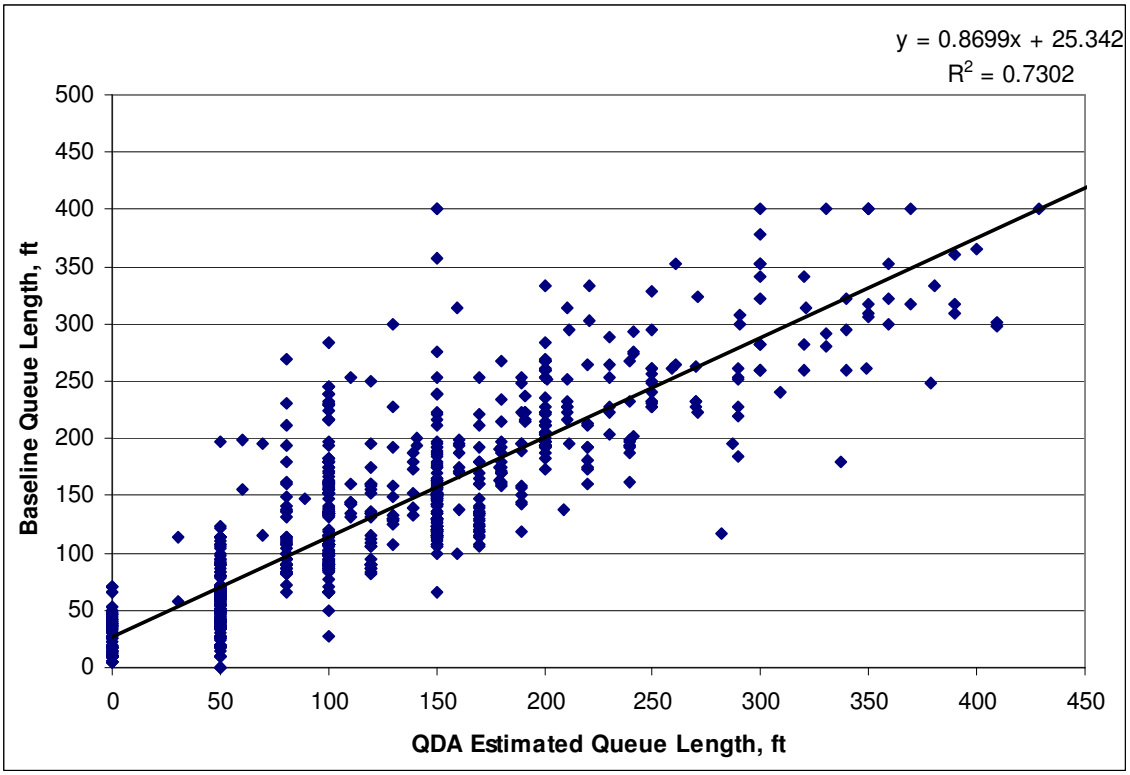


Figure 14 Predicted vs. Actual Queue Plot the Incremental Slope Technique

This method of queue estimation is not recommended because only certain values of estimated queue length are possible using the incremental slope model. The Kalman Filter does not correct the estimates, as the rates of growth calculated, as well as the current measurement, which are rounded to the nearest 50 ft, are not specific. Patterns are formed whereby the same numbers are produced depending on patterns in previous estimates and current measurements. Furthermore, drastic differences in queue measurements from one time step to the next can cause estimated queue lengths to be very large. This contributes to much of the error experienced with this model, and is believed responsible for the rest of the analyzed queue growth models to noticeably outperform the incremental slope method.

Figure 15 shows the predicted vs. actual queue plot of the results for the implementation of a method that uses the moving slope technique. The coefficient of determination, R^2, can be interpreted as 80 percent of the queue estimate data can be explained by the model composed of the moving slope technique and Kalman Filter.

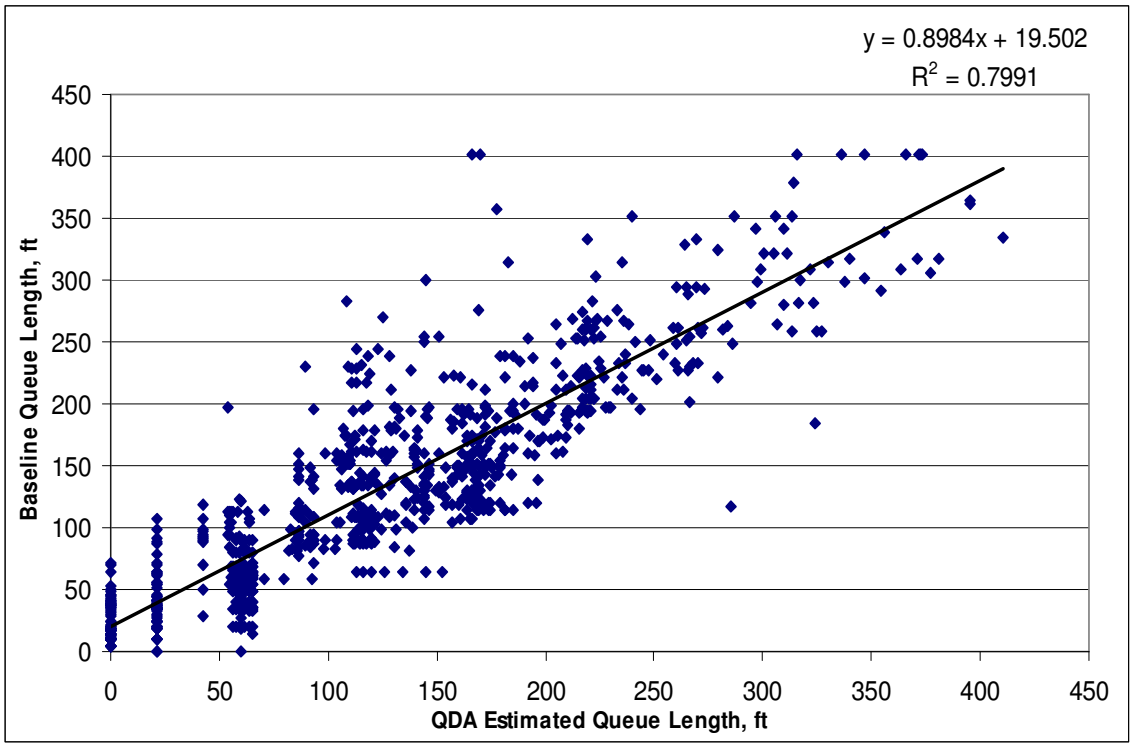


Figure 15 Predicted vs. Actual Queue Plot for the Moving Slope Technique

The results presented in the statistical analysis as well as the predicted vs. actual queue plot shown in Figure 15 pertaining to the moving slope technique are a bit surprising to the researcher. Data points between the initial and previous measurement were essentially not considered when making the estimation of the queue growth. Therefore it was believed that this model would not produce accurate results as the model only considered two data points when producing values for rate of growth (slope). Due to limited intervals considered in this analysis, it was believed that this could introduce

problems if an errant detection from the previous measurement occurred. However, this situation was not evident based on these results. Instead where these phenomena did occur, the Kalman Filter usually corrected these types of errant queue growth projections, and offset them with accurate measurements from the current interval.

Besides noting which methods are most or least successful, it was important to note trends within the data pertaining to each linear queue model. Notice that near the left side of Figure 15 there is a large stack of data running from the point of origin to just above 50 ft on the vertical axis. These values represent values where the technique failed to make any estimate at all, when there in fact was a small queue present on the approach. Some of these occurrences can be explained by the fact that when a short queue was present, vehicles may not have extended to the first detector (placed at the 25 ft mark, but reported 50 ft). This trend is believed to contribute a degree of error to not only this model, but for other models as well. This trend is present in all models analyzed. This contribution to the error is believed to be relatively minor, and is a definite limitation of the VIVDS technology and its capabilities for queue estimation.

The last technique, the linear regression technique plus Kalman Filter, yields a slightly higher coefficient of determination value of 83 percent. While this value may not appear to be significantly higher than the 80 percent value obtained for the moving slope method, it is important to keep in mind that this is a large sample set consisting of more than 900 instances. The results for the linear regression technique can be seen in Figure 16.

A very definite linear trend is present in the data shown in Figure 16. While some baseline values are seen to spike from the trendline, the researcher believes that this model provides the most accurate queue estimation model of the three new models proposed. This is based on the statistical analysis, as well as graphical statistics illustrated through predicted vs. actual queue plots. The estimates produced by this

technique take all previous measurements into consideration, unlike the moving slope and incremental slope techniques. This method produces its rate of growth based on growth during the entire red phase, and produces a "best-fit" for determining the slope of queue growth. The researcher believes this to be better in the sense that all trends are taken into consideration, and it is not as likely that one errant detection will send projected queue length estimates out of control before a Kalman Filter is applied.

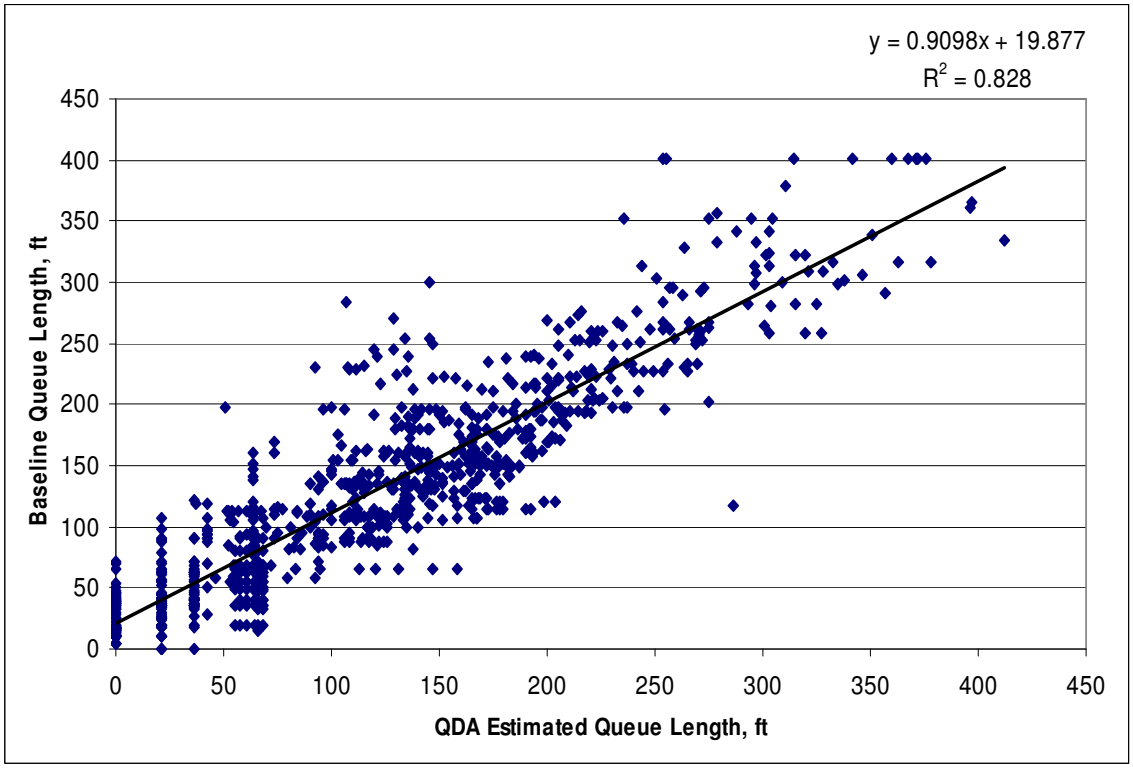


Figure 16 Predicted vs. Actual Queue Plot for the Linear Regression Technique (Calibration)

Therefore, it is the researcher's recommendation that the linear regression technique and accompanying Kalman Filter be used to validate results using a different data set. The results of this analysis are presented in the following subsection.

Validation of Results

The validation procedure was carried out using the same data from George Bush Drive and Wellborn Road. During this procedure, five trial runs were conducted with each trial run consisting of approximately four hours of video data and over 5500 intervals. Results pertaining to the statistics used in the validation process can be seen in Table 4.

Table 4 Statistical Analysis for Validation of Models

Statistic	Weighted Average	Linear Regression
Ave. Error, ft	12.71	6.52
Abs. Ave. Error, ft	22.22	21.86
Stdev. Error, ft	36.63	36.60
Sum of Squared Error, ft^2	8800839	8097047
Root Mean Squared Error, ft	44.69	46.59

The results for the graphical statistics indicated similar results, whereby the linear regression method was observed to outperform the weighted average method. The predicted vs. actual queue plot for QDA validation using the weighted average technique shown in Figure 17 indicates that the coefficient of determination, R^2, is approximately 0.8511. This indicates a relatively good fit, and much of the variance can be described by this model. However, the predicted vs. actual queue plot for the validation of data using the linear regression technique plus the application of a Kalman Filter shows slightly better results for this analysis. This model results in an R^2 value of 0.8574. This indicates that the linear regression model and Kalman Filter provides a rather good system for producing queue estimates. Most of the variance in the actual queue length can be described by the model, as is evident by the linear trend in Figure 18.

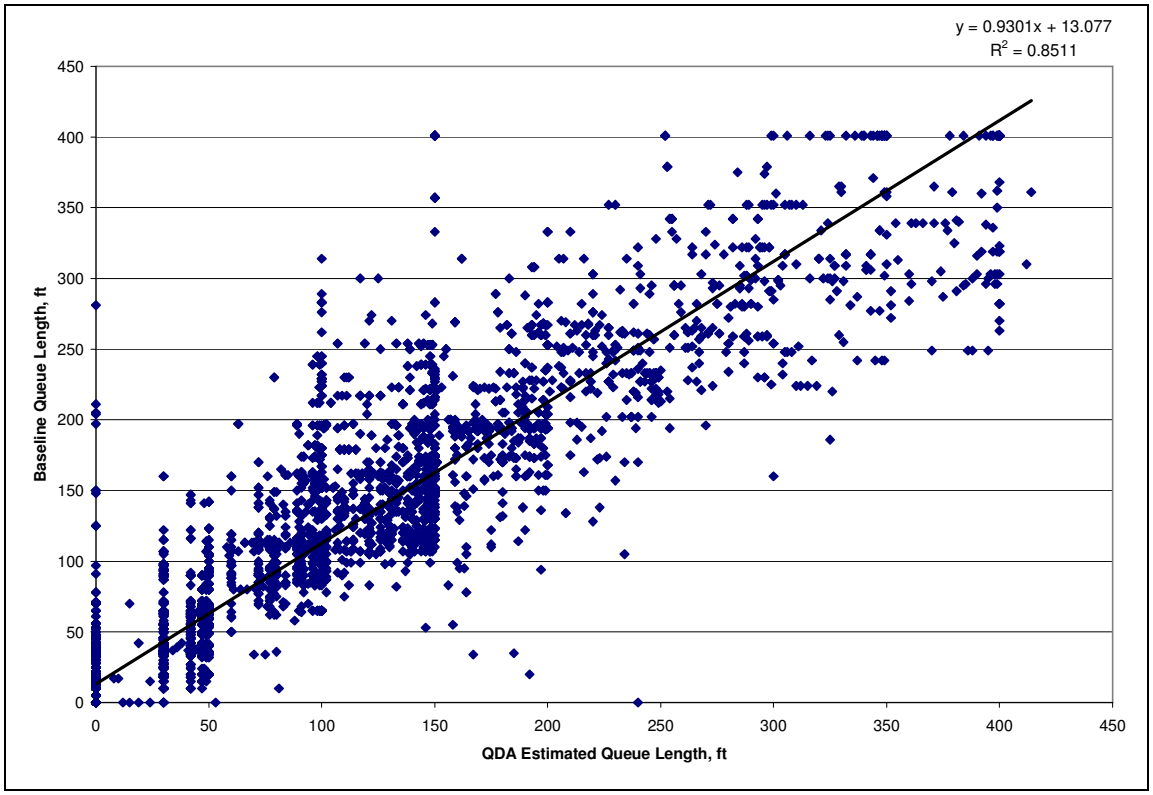


Figure 17 Predicted vs. Actual Queue Plot for the Weighted Average Technique

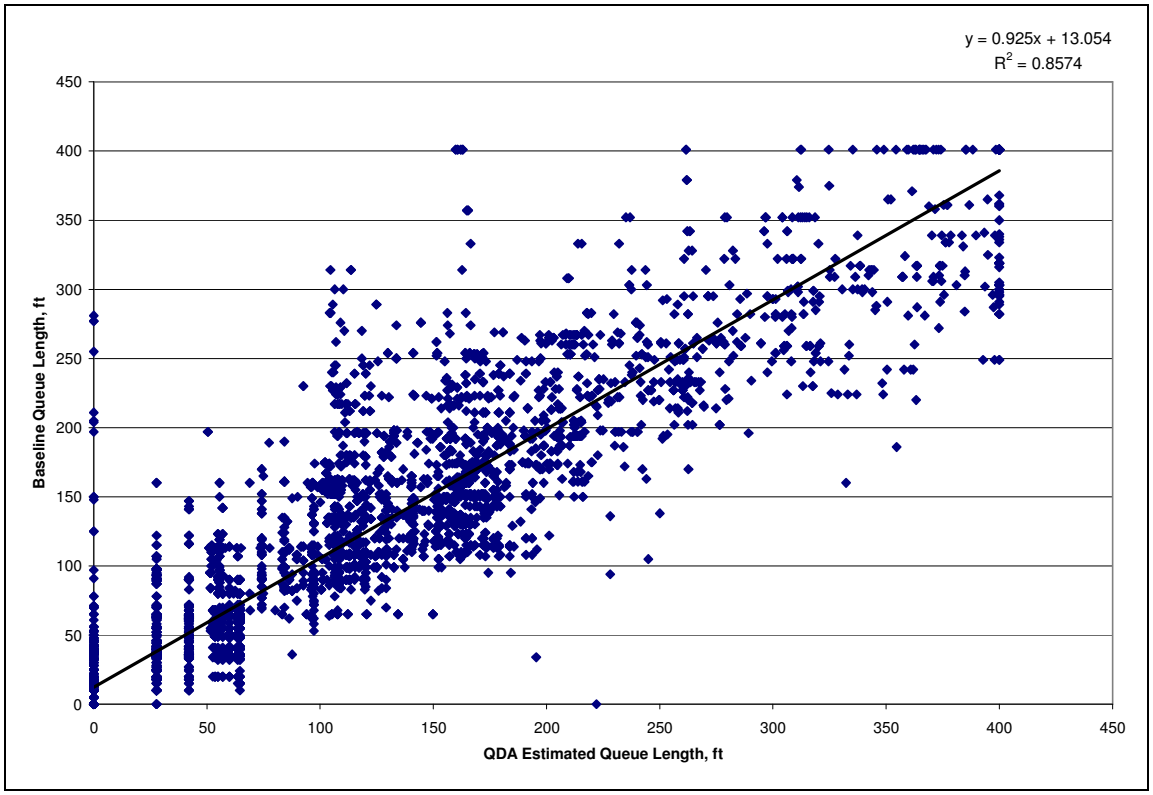


Figure 18 Predicted vs. Actual Queue Plot for the Linear Regression Technique (Validation)

Discussion of Results

The statistical analysis indicates that for any one given queue estimate that the linear regression technique plus Kalman Filter will provide the best estimate of queue lengths of the three queue estimation models investigated. This is demonstrated by the fact that the statistical analysis produced results that were favorable for the linear regression technique compared to the remaining models. Predicted vs. actual queue plots strengthen the conclusion that the linear regression technique was better by producing a larger coefficient of determination, whereby the linear regression model was better able to explain variations in queue length data.

The data indicate that the linear regression method slightly outperforms the weighted average method used in the previous QDA under the calibration tests. This trend was confirmed, as when validation tests were conducted, the linear regression method was shown to outperform the weighted average method with respect to average error, absolute average error, standard deviation of the error, sum of squared error, and root mean squared error. These statistics all demonstrate lower values for the linear regression technique with a Kalman Filter. The average error was determined to be 6.52 for the linear regression technique, as compared to the 12.71 for the weighted average method. It would be desirable to obtain a technique that produces an average error close to zero. This would indicate that a modeling technique would more adequately normalize the error about the actual queue length values. Furthermore, the smaller magnitude of error (demonstrated by the absolute average error) and standard deviation of error terms were smaller for the linear regression technique. This indicates that the linear regression method produces estimates that are less variable, and are able to better approximate actual queue length values. The combination of the ability to construct a normal distribution with estimates that are closer to actual queue length values, and demonstrate less variance, support the relevance of the linear regression technique applied with a Kalman Filter.

Similar trends were observed when comparing predicted vs. actual queue plots for the validation tests. Both tests yielded high R^2 values in the validation tests. The linear regression method produced a value of 0.8574, compared to the 0.8511 demonstrated by the weighted average method. It was the researcher's goal to decipher a queue estimation method using VIVDS hardware and an algorithm that best describes the actual queue length at an approach, while capable of describing variances in actual queue length data. A method capable of describing 85 percent of the variance of a data set provides strong support for the implementation of this method for VIVDS queue estimation. Therefore, the researcher believes the linear regression method with a

Kalman Filter is a viable alternative to the weighted average method, and is capable of adequately estimating queue lengths.

The Kalman Filter is an additional correction mechanism that is applied to queue estimates produced during each polling interval performed by the QDA. It provides a system that controls the growth of the queue estimate. The Kalman Filter is an intelligent means for merging current estimates of queue length and the current measurements provided by VIVDS. While there are many ways to combine these two values using weighting factors and calibration procedures, the Kalman Filter minimizes the mean-squared error with little to no calibration (*11*, *14*, *15*). The Kalman Filter minimizes the squared deviation between an estimator and the true value it is attempting to estimate (*15*). The implementation of the Kalman Filter at the very least provides a process for controlling queue growth trends, and combining queue estimates and measurements intelligently with the intent of minimizing error between estimated and actual baseline queue values.

From examination of the data sets, the researcher believes that the application of the Kalman Filter, while useful for merging measurements and estimates of current queue length, does not directly influence accuracy. This is noticed when seeing the obvious performance differences of the models (see Table 3). The moving slope method is shown to outperform the incremental slope method, while the linear regression method is shown to outperform both of them. Obviously, the Kalman Filter cannot be responsible for these improvements when comparing methods, as the Kalman Filter was applied to all three queue estimation methods. Instead, accurate queue estimation is heavily dependant upon the queue model used. As shown by the statistical analysis and graphical statistics, there is a noticeable difference in performance between the linear queue models.

The three linear queue models are simplistic in their mathematical processes, and it can be argued that a more sophisticated model may provide a more accurate model for describing vehicle queuing. This analysis had to assume linear queue growth, thus, this thesis assumed random arrivals. A more complex queue estimation technique should incorporate more detectors with smaller graduations between detection zones. This should allow for the utilization of a more complex queue growth model, as the smaller detector spacing will not cause as much initial error as detectors spaced at 50 ft introduces to the queue growth models. The fact that there can only be a limited number of detectors applied to an approach severely limits the design of queue models. A simplistic hardware setup requires a simplistic modeling approach supported by correction procedures that ensure that the model is under control. The new QDA, using a linear regression model and a Kalman Filter accomplishes the stated objective, and minimizes noise with respect to queue estimation. The QDA and use of the linear regression method is shown to minimize error to a surprising degree given that only eight detectors over a 400 foot analysis area were used.

CONCLUSION AND RECOMMENDATIONS

Conclusions drawn from the results presented in this thesis are presented in this section as well as responses to stated objectives. Additionally, recommendations are given with respect to considerations for future research and investigation, as well as for potentially application of the QDA.

CONCLUSION

Three linear queue estimation models were analyzed as the base of a new queue and delay estimation algorithm for estimating queue length during the red phase of a traffic signal on an approach. Applied to each one of these linear queue estimation models was a Kalman Filter. The filter was designed to minimize the mean squared error of estimates and to provide an intelligent method of combining current queue measurements that contained some degree of error, with queue estimates based on queue growth trends.

The results of this research show that when the three linear queue models were tested, a model using a linear regression technique whereby a best-fit line of previous queue measurements was used to predict a queue growth rate. The queue rate was then used in the Kalman Filter to optimize current queue estimates.

When these results for the linear regression technique and Kalman Filter were compared to results produce using a weighted average technique applied in the previous version of the QDA, it was found that statistics produced very similar results, with the linear regression method slightly outperforming the weighted average technique. Graphical statistics through the use of predicted vs. actual queue plots reveal the linear regression technique is better able to explain variances in queue length data, as well as demonstrating the ability to better normalize estimates about predicted values.

While the improvements are small when considering that the linear regression technique only slightly outperforms the weighted average technique, it must be considered that the linear regression technique and Kalman Filter requires no weighting factors and very little calibration to implement. The new technique is adaptive in the sense that it seems to automatically correct itself using the Kalman Filter and does not allow estimates to get out of control. It would be the researcher's recommendation that this method be given consideration for future implementation in the QDA due to its adaptive nature, little calibration, and proven accuracy when estimating queue length.

RECOMMENDATIONS

The potential for using existing VIVDS technology and applying an algorithm that estimates queue length introduces an entirely new function for this technology. The potential to use the QDA for estimating MOEs such as queue length and ultimately delay for adaptive control of a signal controller offers the traffic engineer one more tool that may improve the overall operation and safety of an intersection. As such, the researcher has made recommendations that may aid the traffic engineer with implementing the QDA.

Future Research and Development

Many improvements and research should be conducted before the QDA should be relied upon for adaptive control of an intersection. While the research conducted in this thesis is a valuable step towards producing accurate queue estimates, the researcher believes many steps are necessary before optimal estimates can be made. Therefore, the researcher has made the following recommendations for future research and development:

- Multiple VIVDS systems should be tested for compatibility with the QDA.
- A wider variety of intersections should be analyzed under both day and nighttime conditions.
- A variety of traffic conditions should be analyzed, including situations where cycle failures are present.
- Multiple cameras should be tested for situations where the queue length extends beyond the view of only one VIVDS camera.
- The new QDA should be implemented in the previous QDA *Visual Basic 6.0* code. The code for determining the queue length during the red phase using the weighted average method should be removed and the linear regression method and Kalman Filter method inserted.
- The effects of cycle failure and the presence of an initial queue should also be tested using both the linear regression technique and weighted average technique. A video data set where cycle failure occurs would need to be obtained.
- Video data for approaches where non-random arrivals occur should be analyzed. This could potentially nullify the assumption of linear queue growth used in this thesis.

QDA Implementation

The following recommendations should be taken into consideration by the traffic engineer when implementing the QDA, and the modeling techniques analyzed in this thesis for optimal results:

- Due to the limited view of the VIVDS camera, the engineer should consider multiple cameras if queues are believed to consistently extend beyond what the VIVDS camera is able to reasonably detect.
- Adequate lighting should be provided at the approach

- The VIVDS camera should be positioned such that lane occlusion is minimized. Ideally, the camera should be placed at the center of the approach such that multiple lanes of traffic can utilize queue detectors.
- High-quality, high resolution VIVDS cameras should be used for queue detection if possible.
- Investigate the accuracy of the QDA using more detectors spaced at increments less than 50 ft.

REFERENCES

1 Bonneson, J., H. Charara, D. Middleton, and M. Cheek. *Urban Street Performance Measurement: Prototype Non-Intrusive Detection Technique*. National Cooperative Highway Research Program, TRB, National Research Council, Washington D.C., 2006.

2 *Highway Capacity Manual*, TRB, National Research Council, Washington, D.C., 2000

3 *Measuring and Predicting the Performance of Automobile Traffic on Urban Streets*. An Interim Report Prepared for the National Cooperative Highway Research Program, TRB, National Research Council, Washington, D.C., 2005.

4 May, A.D. *Traffic Flow Fundamentals*. Prentice Hall, Englewood Cliffs, N.J., 1990.

5 Michalopoulos, P.G., R. Fitch, and B. Wolf. Development and Evaluation of a Breadboard Video Imaging System for Wide Area Vehicle Detection. In *Transportation Research Record: Journal of Transportation Research Board, No. 1225*, TRB, National Research Council, Washington, D.C., 1989, pp. 140-149.

6 Bonneson, J., and M. Abbas. *Intersection Video Detection Field Handbook,* Research Report 4285-2, Texas Transportation Institute, College, Station, Tex. September 2002.

7 Michalopoulos, P.G. Vehicle Detection Video Through Image Processing: The Autoscope System. *IEEE Transactions on Vehicular Technology*. Vol. 40, No. 1, 1991, pp. 21-29.

8 Rourke, A., and M.G.H. Bell. Queue Detection and Congestion Monitoring Using Image Processing. *Traffic Engineering and Control*. September, 1991, pp. 412-421.

9 Rourke, A., M.G.H. Bell, and N. Hoose. Road Traffic Monitoring Using Image Processing. *Third International Conference on Road Traffic Control, London, United Kingdom,* May, 1990, pp. 163-167.

10 Fathy, M., and M.Y. Siyal. Real-time Image Processing Approach to Measure Traffic Queue Parameters. *IEE Processor-Vision Image Signal Process*. Vol. 142, No. 5, October, 1995, pp. 297-303.

11 Levy, L.J. *The Kalman Filter: Navigation's Integration Workhorse*. The Johns Hopkins University Applied Physics Laboratory, Laurel, Md, 2002.

12 Brown, R.G., and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering,* Third Edition. John Wiley and Sons, Inc., New York, 1997. pp. 214-220.

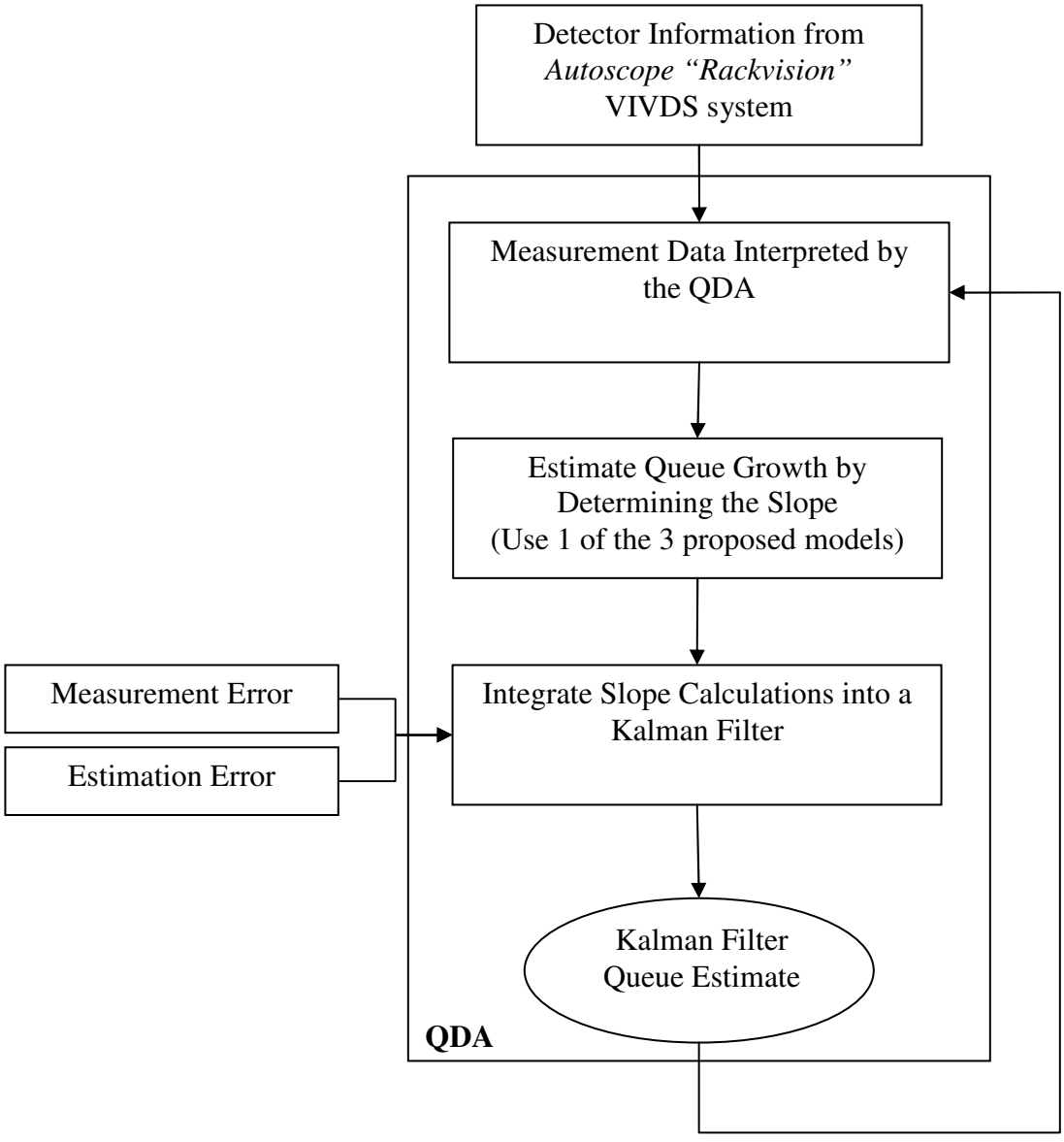
13 Simon, D. Kalman Filtering. *Embedded Systems Programming*. Vol. 14, No. 6, 2001. pp. 72-79.

14 Welch, G., and G. Bishop. *An Introduction to the Kalman Filter*. Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, N.C., 2004.

15 Montgomery, D.C., and G.C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley and Sons, Inc., New York, 2007.

APPENDIX A

Illustration of Queue Estimation Techniques Applied to the Kalman Filter



Baseline data are then compared to the QDA Kalman Filter output of the estimated queue length.

APPENDIX B

Visual Basic Pseudo Code for Queue Estimation including a Kalman Filter

Moving Slope Method

Initialize Variables:

Get offline measurement standard deviation (Meas_SD)
Get estimate standard deviation (Est_SD)
Set queue reporting period to 10 seconds (t_Rpt)
Set previous measurement of queue to 0 (Meas_queue_t_1)
Set Iteration to 1 (Iteration)

Initialize Kalman Filter Variables:

Q = (Est_SD) ^ 2
R = (Meas_SD) ^ 2

Pk = 0
xk = 0

QDA (For Red Phase Only When the First Detector is Activated):

Get Time
Get phaseStatus

If phaseStatus is Red

Do Until phaseStatus is not Red

If Iteration equals 1
Get time_init
Get queue_init

QDA obtains the time and queue length for the interval before red

Get Cur_queue_meas

QDA gets current queue measurement

B=0

```
        uk=0
        zk=Cur_queue_meas

        Set initial Kalman Filter variables

    If Iteration does not equal 1
        Get time_prev
        Get queue_prev
        Get Cur_queue_mes

        Obtain the queue length and time interval of previous interval. Also
        obtain the current queue measurement

        Slope = (queue_prev-queue_init)/(time_prev-time_init)
        uk=Slope
        zk=Cur_queue_meas
    End If

    xk_ = xk + uk
    Pk_ = Pk + Q

    Kalman Filter predictor portion calculation

    Kk = Pk_ / (Pk_ + R)
    xk = xk_ + Kk * (zk - xk_)
    Pk = (1 - Kk) * Pk_

    Kalman Filter corrector portion calculation

    Output current queue estimate (xk)
    Iteration = Iteration +1

  Loop

If phaseStatus is not Red

  Iteration =1
  Pk = 0
  xk = 0

  Reset Variables for next Red phase
```

Incremental Slope Method

Initialize Variables:

Get offline measurement standard deviation (Meas_SD)
Get estimate standard deviation (Est_SD)
Set queue reporting period to 10 seconds (t_Rpt)
Set previous measurement of queue to 0 (Meas_queue_t_1)
Set Iteration to 1 (Iteration)

Initialize Kalman Filter Variables:

Q = (Est_SD) ^ 2
R = (Meas_SD) ^ 2

Pk = 0
xk = 0

QDA (For Red Phase Only When the First Detector is Activated):

Get Time
Get phaseStatus

If phaseStatus is Red

Do Until phaseStatus is not Red

If Iteration equals 1

Get Cur_queue_meas

QDA gets current queue measurement

B=0
uk=0
zk=Cur_queue_meas

Set initial Kalman Filter variables

If Iteration does not equal 1
Get queue_prev
Get queue_prev_2

```
        Get Cur_queue_meas

        Obtain the queue length and for the previous interval and two
        intervals before current interval.  Also obtain the current queue
        measurement

        Slope = (queue_prev-queue_prev_2)/(t_Rpt)
        uk=Slope
        zk=Cur_queue_meas
    End If

    xk_ = xk + uk
    Pk_ = Pk + Q

    Kalman Filter predictor portion calculation

    Kk = Pk_ / (Pk_ + R)
    xk = xk_ + Kk * (zk - xk_)
    Pk = (1 - Kk) * Pk_

    Kalman Filter corrector portion calculation

    Output current queue estimate (xk)
    Iteration = Iteration +1

  Loop

If phaseStatus is not Red

  Iteration =1
  Pk = 0
  xk = 0

  Reset Variables for next Red phase
```

Linear Regression Method

Initialize Variables:

Get offline measurement standard deviation (Meas_SD)
Get estimate standard deviation (Est_SD)
Set queue reporting period to 10 seconds (t_Rpt)
Set previous measurement of queue to 0 (Meas_queue_t_1)
Set Iteration to 1 (Iteration)

Initialize Kalman Filter Variables:

Q = (Est_SD) ^ 2
R = (Meas_SD) ^ 2

Pk = 0
xk = 0

Clear Linear Regression Input Variables:

Xi_sum = 0
Yi_sum = 0
Xi_2_sum = 0
XiYi_sum = 0

QDA (For Red Phase Only When the First Detector is Activated) :

Get Time
Get phaseStatus

If phaseStatus is Red

Do Until phaseStatus is not Red

If Iteration equals 1

Get Cur_queue_meas
Get queue_prev
Get time_prev

QDA gets Previous measurement of queue length, the previous time interval and current queue measurement

```
    B=0
    uk=0
    zk=Cur_queue_meas
```

Set initial Kalman Filter variables

```
    n=1
```

Number of instances of queues and times collected. This value is to be used for the summation during the linear regression calculation.

```
If Iteration does not equal 1
    If n=1 then
        B1=0
```

B1 Term to describe the initial slope of the linear regression calculation.

```
    Else
        B1 = (XiYi_sum - (Yi_sum * Xi_sum) / n) / (Xi_2_sum -
        (Xi_sum) ^ 2 / n)
    End If

    n=n+1

    Get Cur_queue_meas
    Get queue_prev
    Get time_prev
```

QDA gets Previous measurement of queue length, the previous time interval and current queue measurement

```
    Slope = t_Rpt* B1

    uk=Slope
    zk=Cur_queue_meas
```

Set Kalman Filter variables

```
End If

Xi = time_prev
Yi = queue_prev
```

```
        Xi_sum = Xi_sum + Xi
        Yi_sum = Yi_sum + Yi
        Xi_2 = (Xi) ^ 2
        Xi_2_sum = Xi_2_sum + Xi_2
        XiYi = Xi * Yi
        XiYi_sum = XiYi_sum + XiYi
```

Linear regression procedure. Notice for the first iteration, Xi_sum and Yi_sum were cleared at "0" at the beginning of the algorithm.

```
        xk_ = xk + uk
        Pk_ = Pk + Q
```

Kalman Filter predictor portion calculation

```
        Kk = Pk_ / (Pk_ + R)
        xk = xk_ + Kk * (zk - xk_)
        Pk = (1 - Kk) * Pk_
```

Kalman Filter corrector portion calculation

```
        Output current queue estimate (xk)
        Iteration = Iteration +1
    Loop
If phaseStatus is not Red
    Iteration =1
    Pk = 0
    xk = 0
```

Reset Variables for next Red phase

APPENDIX C

Baseline Data Distributions from the Calibration Data Set

Table 5 Frequency Table for Baseline Queue Lengths

Bin	Frequency
0	332
50	217
100	190
150	225
200	163
250	87
300	63
350	26
400	11
450	14
500	0
550	0
More	0
Sum	**1328**

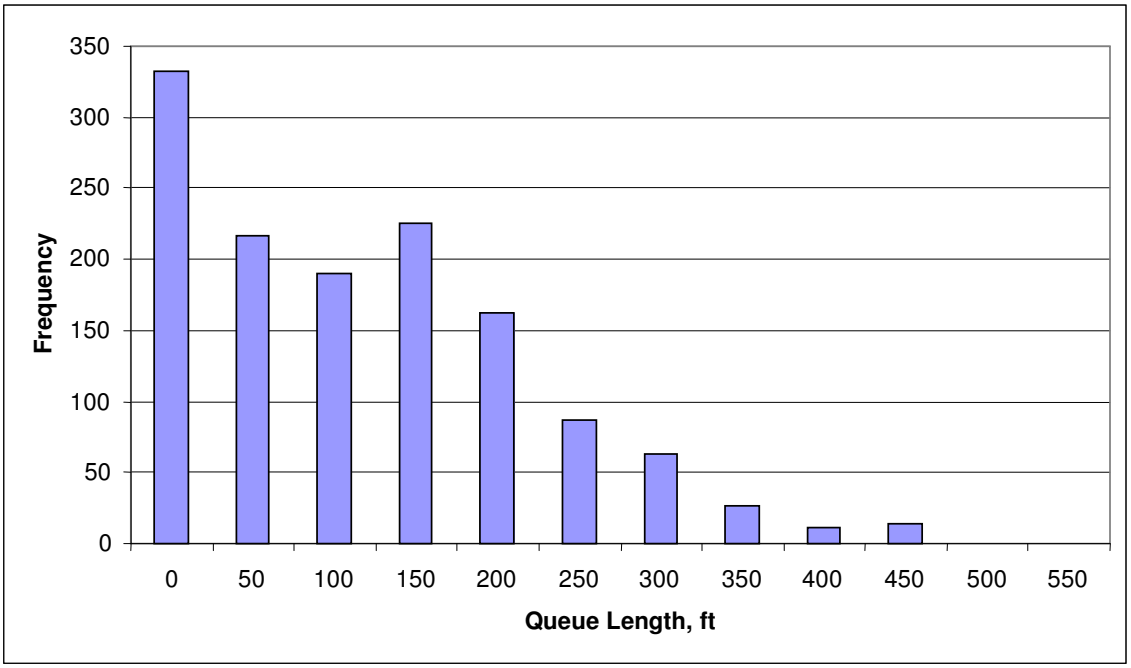


Figure 19 Histogram of Baseline Queue Lengths

Measurement Data Distributions from the Calibration Data Set

Table 6 Frequency Table for Queue Length Measurements

Bin	Frequency
0	512
50	222
100	222
150	185
200	86
250	43
300	30
350	23
400	5
450	0
500	0
550	0
More	0
Sum	**1328**

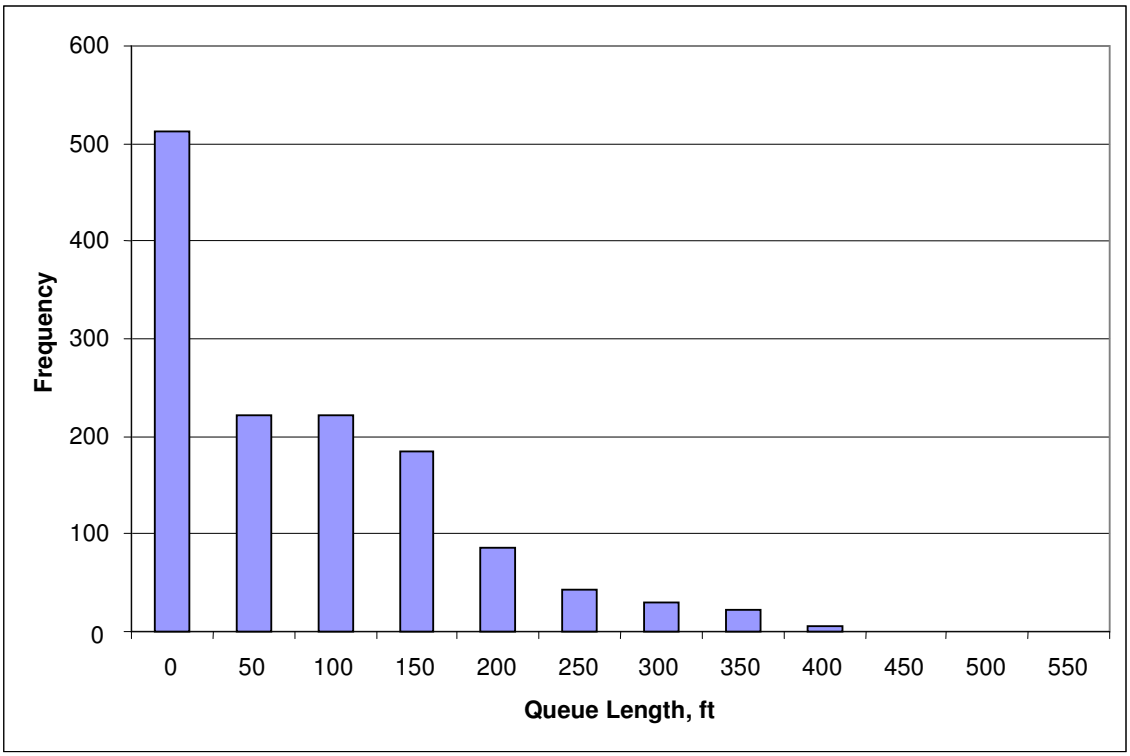


Figure 20 Histogram of Queue Length Measurements

Baseline Data Distributions from the Calibration Data Set During Red Phases

Table 7 Frequency Table for Baseline Data During Red Phases

Bin	Frequency
0	3
50	197
100	176
150	210
200	156
250	77
300	55
350	23
400	8
450	9
500	0
550	0
More	0
Sum	914

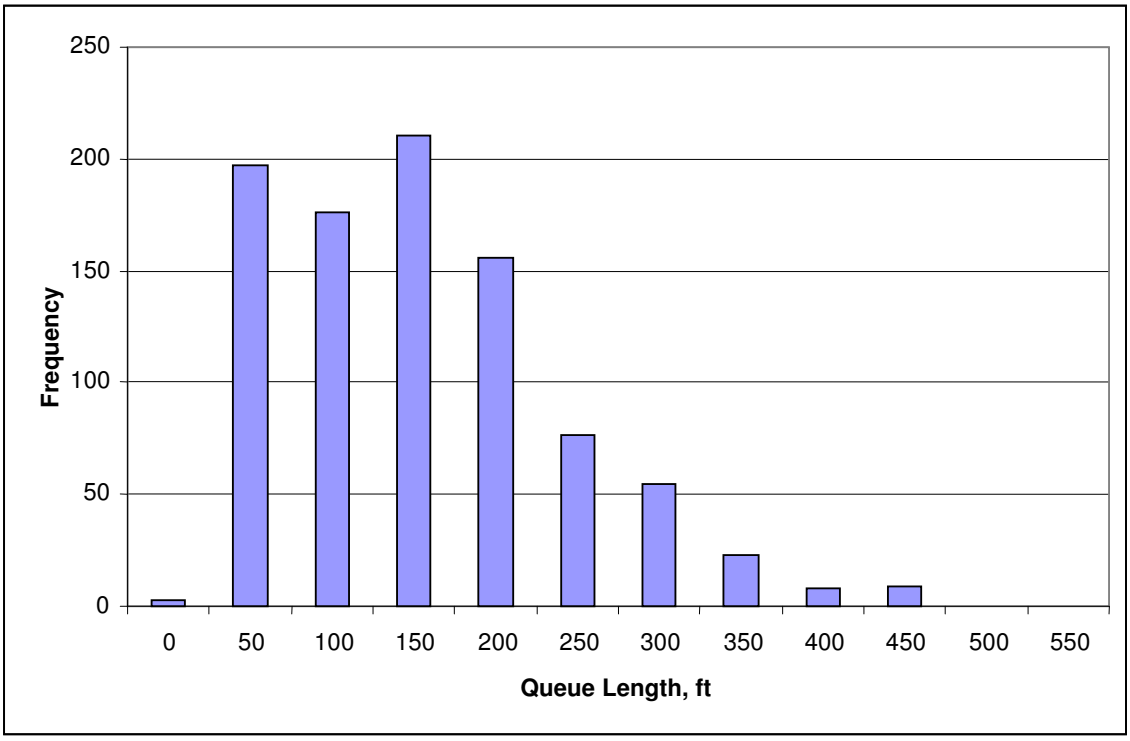


Figure 21 Histogram of Baseline Queue Lengths During Red Phases

Measurement Data Distributions from the Calibration Data Set During Red Phases

Table 8 Frequency Queue Length Measurements During Red Phases

Bin	Frequency
0	96
50	211
100	188
150	209
200	102
250	52
300	36
350	16
400	3
450	1
500	0
550	0
More	0
Sum	**914**

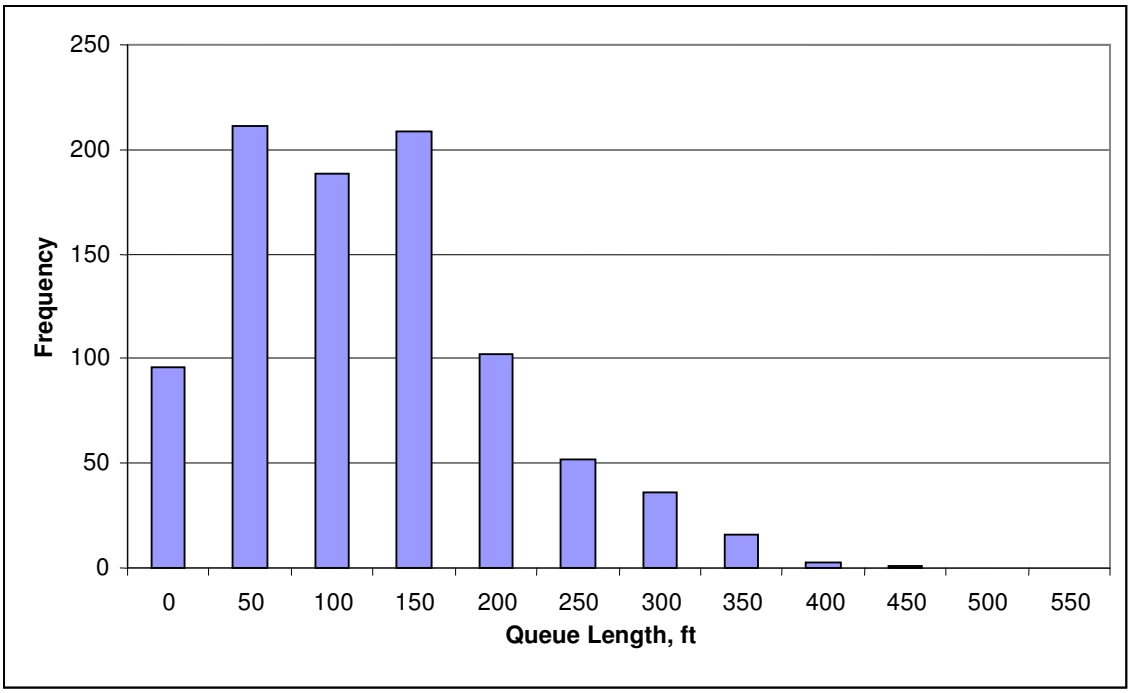


Figure 22 Histogram of Measurement Queue Lengths During Red Phases

Baseline Data Distributions from the Validation Data Set

Table 9 Frequency Table for Baseline Queue Lengths (Validation)

Bin	Frequency
0	321
50	191
100	150
150	149
200	122
250	93
300	71
350	60
400	38
450	22
500	0
550	0
More	0
SUM	**1217**

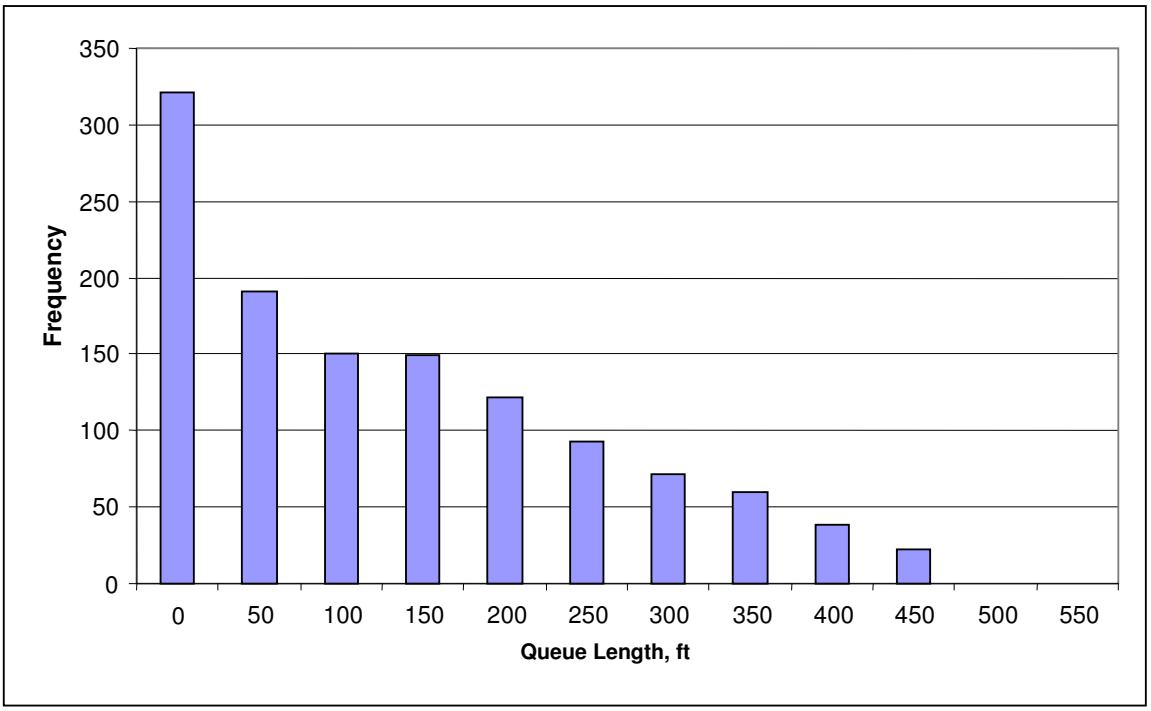


Figure 23 Histogram of Baseline Queue Lengths (Validation)

Measurement Data Distributions from the Validation Data Set

Table 10 Frequency Table for Queue Length Measurements (Validation)

Bin	Frequency
0	351
50	184
100	138
150	177
200	77
250	55
300	52
350	59
400	88
450	36
500	0
550	0
More	0
SUM	**1217**

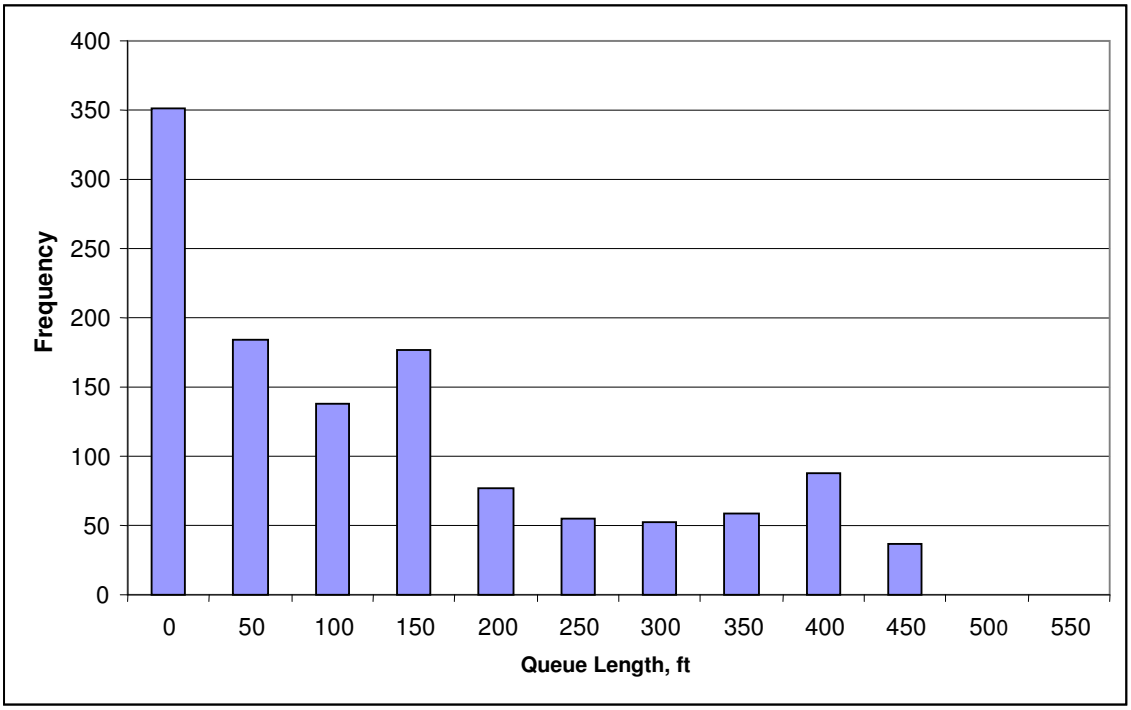


Figure 24 Histogram of Queue Length Measurements (Validation)

Baseline Data Distributions from the Validation Data Set During Red Phases

Table 11 Frequency Table for Baseline Data During Red Phases (Validation)

Bin	Frequency
0	12
50	220
100	146
150	151
200	112
250	83
300	70
350	43
400	21
450	39
500	0
550	0
More	0
Sum	**897**

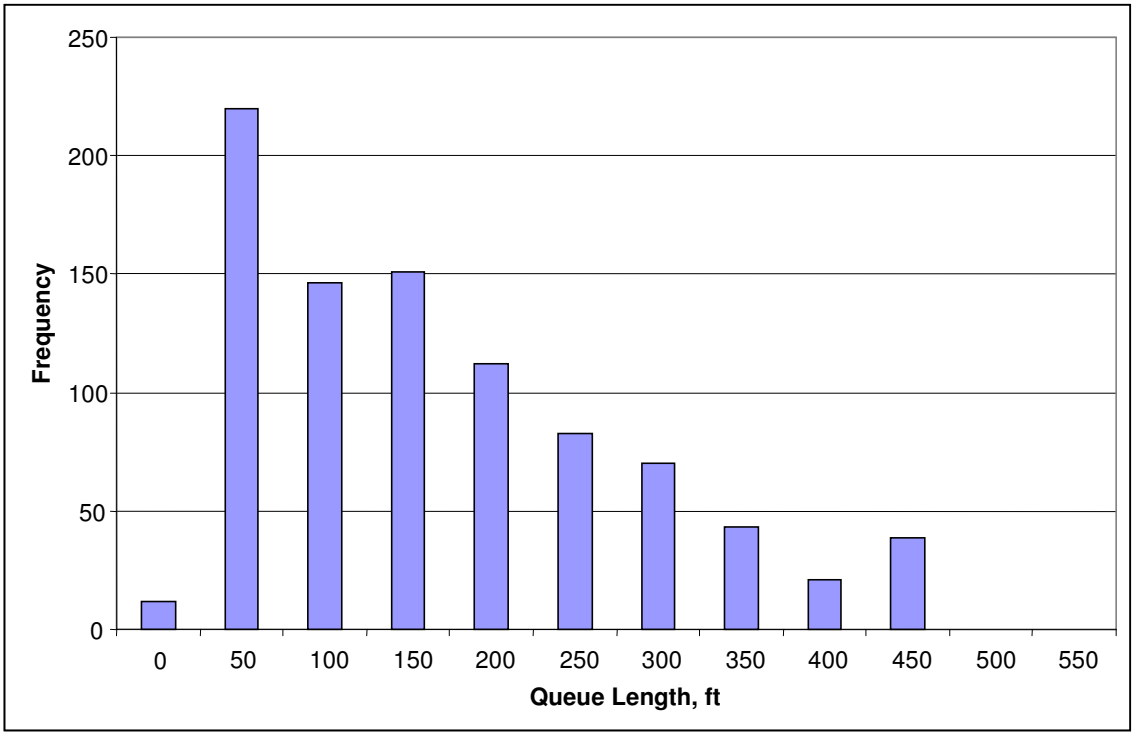


Figure 25 Histogram of Baseline Queue Lengths During Red Phases (Validation)

Measurement Data Distributions from the Validation Data Set During Red Phases

Table 12 Frequency Queue Length Measurements During Red Phases (Validation)

Bin	Frequency
0	69
50	171
100	142
150	169
200	61
250	67
300	49
350	59
400	89
450	21
500	0
550	0
More	0
Sum	**897**

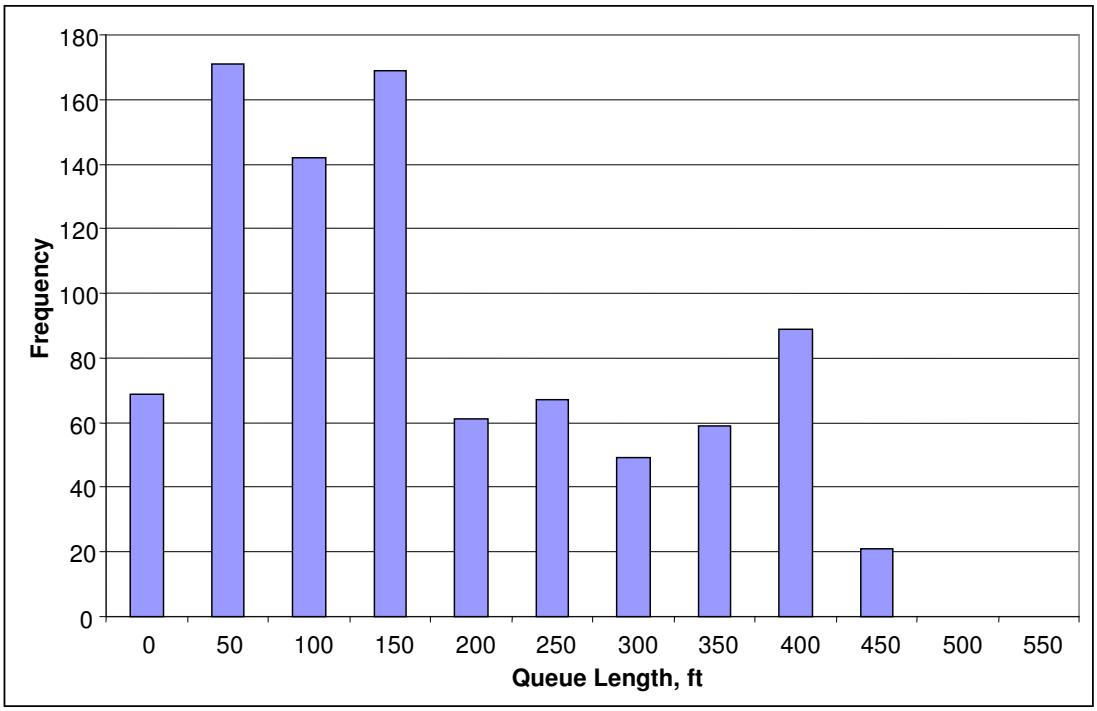


Figure 26 Histogram of Measurement Queue Lengths During Red Phases (Validation)

APPENDIX D

Offline Standard Deviation Results for Measurement Error and Estimate Error

Table 13 Standard Deviation of Error for Measurements (Offline)

Standard Deviation, ft	
Calibration	Validation
85.866	45.667

Table 14 Standard Deviation of Estimates During Calibration (Offline)

Standard Deviation			
Incremental Slope	Moving Slope	Linear Regression (Calibration)	Linear Regression (Validation)
68.056	52.989	48.225	51.013

VITA

Marshall Tyler Cheek
3135 TAMU
College Station, TX 77843-3135

Education

B.S., Civil Engineering, Iowa State University, August 2005.
M.S., Civil Engineering, Texas A&M University, May 2007.

Work Experience

Lockwood Andrews and Newnam – Employer post-graduation
Texas Transportation Institute, September 2005 – December 2006.
Center for Transportation Research and Education, January 2005 – August 2005.
Texas Transportation Institute, June 2004 – August 2004.

Professional Affiliations and Societies

Institute of Transportation Engineers
American Society of Civil Engineers
Tau Beta Pi
Chi Epsilon

Areas of Interest

Traffic operations analysis
Intelligent transportation systems
Traffic signal design
Traffic signal operations and control