Technical Report Documentation Page

1. Report No.
FHWA/TX-06/0-4729-2

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
DISTRIBUTED ARCHITECTURE AND ALGORITHM FOR ROBUST REAL-TIME PROGRESSION EVALUATION AND IMPROVEMENT

5. Report Date
October 2005
Resub: Mar 2006/Pub:Apr. 2006

6. Performing Organization Code

7. Author(s)
Montasir Abbas, Hassan Charara, Nadeem Chaudhary, and Youn su Jung

8. Performing Organization Report No.
Report 0-4729-2

9. Performing Organization Name and Address
Texas Transportation Institute
The Texas A&M University System
College Station, Texas 77843-3135

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
Project No. 0-4729

12. Sponsoring Agency Name and Address
Texas Department of Transportation
Research and Technology Implementation Office
P. O. Box 5080
Austin, Texas 78763-5080

13. Type of Report and Period Covered
Technical Report:
September 2003-August 2005

14. Sponsoring Agency Code

15. Supplementary Notes
Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration.
Project Title: Reducing Traffic Congestion and Fuel Emission through a Minimal Cost, Self-Tuning Closed-Loop Progression System
URL: http://tti.tamu.edu/documents/0-4729-2.pdf

16. Abstract

Closed-loop systems are widely implemented in Texas arterials to provide efficient operation of arterial intersections while still providing signal progression. Nevertheless, poor progression can be observed along most arterials due to outdated offsets, short-term variations in traffic patterns (early-return-to-green), or changes in arterial's speed and changes in traffic volumes.

The limited abilities of closed-loop systems to adapt to traffic variations have stimulated interest into incorporating the technologies of adaptive control software (ACS) into closed-loop systems in order to address such issues. These integrated-type systems require lower cost and minimal staff training, in comparison to fully adaptive systems, since traffic engineers and technicians managing the traffic signal operating systems are already familiar with the closed-loop logic.

The objective of this research is to develop, implement, and test an algorithm that will address the limitations of previous efforts in the area of real-time offset-tuning. This final report describes a flexible experimental framework for theoretical development and experimentation of adaptive control algorithms, in the context of closed-loop systems operation. The developed system includes: 1) robust classification algorithm of progression quality and remedies, and 2) vendor-independent distributed implementation architecture for the proposed algorithm.

17. Key Words
Traffic Control, Closed-Loop Signal Systems, Traffic Signal Timing, Offset, Adaptive Control

18. Distribution Statement
No restrictions. This document is available to the public through NTIS:
National Technical Information Service
Springfield, Virginia 22161
http://www.ntis.gov

19. Security Classif.(of this report) Unclassified	20. Security Classif.(of this page) Unclassified	21. No. of Pages 62	22. Price

DISTRIBUTED ARCHITECTURE AND ALGORITHM FOR ROBUST REAL-TIME PROGRESSION EVALUATION AND IMPROVEMENT

by

Montasir Abbas, Ph.D., P.E.
Assistant Research Engineer
Texas Transportation Institute

Hassan Charara
Associate Research Scientist
Texas Transportation Institute

Nadeem Chaudhary, Ph.D., P.E.
Senior Research Engineer
Texas Transportation Institute

and

Youn su Jung
Graduate Research Assistant
Texas Transportation Institute

Report 0-4729-2
Project Number 0-4729
Project Title: Reducing Traffic Congestion and Fuel Emission through a Minimal Cost, Self-Tuning Closed-Loop Progression System

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

October 2005
Resubmitted: March 2006
Published: Arpil 2006

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

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official view or policies of the Federal Highway Administration (FHWA) or the Texas Department of Transportation (TxDOT). This report does not constitute a standard, specification, or regulation. Its contents are not intended for construction, bidding, or permit purposes. The use and names of specific products or manufacturers listed herein does not imply endorsement of those products or manufacturers. The principal investigator and engineer in charge of this research was Dr. Montasir M. Abbas, P.E. # 92989.

ACKNOWLEDGMENTS

This project was conducted in cooperation with TxDOT and FHWA. The researchers acknowledge the support and guidance provided by the Project Director, Mr. Henry Wickes, as well as the members of the Project Monitoring Committee, including: Program Coordinator Ms. Carol Rawson, Project Advisors Mr. Don Baker, Mr. Kirk Barnes, and Mr. Roy Parikh, and RTI Research Engineer Mr. Wade Odell.

We also acknowledge the guidance and assistance provided to us by Mr. Robert Deroche from Texas Transportation Institute (TTI).

TABLE OF CONTENTS

Page

List of Figures ix
List of Tables x
Chapter 1: Introduction 1
Overview 1
Research Objectives 2
Research Approach 2
Chapter 2: Literature Review 5
Adaptive Control Software and Algorithms 5
History of Adaptive Control Software 5
Adaptive Control Algorithms 6
Dynamic Arterial Responsive Traffic System (DARTS) 11
Closed-Loop Systems 11
Bridging the Gap: PRO-TRACTS and ACS Lite 13
Purdue Real-Time Offset Transitioning Algorithm for Coordinating Traffic Signals (PRO-TRACTS) 13
ACS Lite 14
Summary 14
Chapter 3: Robust Offset Classification 17
Introduction 17
Traffic Parameters Affecting Progression 17
General Findings 18
System Delay 18
Arterial Stops 18
Comparison between One-Way and Two-Way Offsets 18
Experimental Design for Robust Offset Classification 18
Offset Classification 20
Performance Evaluation 21
Two-Way Offset Tuning Algorithm 22
Chapter 4: Algorithm’s Prototype and Field Communication Structure 27
Introduction 27
Algorithm Prototype 27
Field Communication Module 28
Master Subsystem 28
Slave Subsystem 30
Communication Subsystem 31
PILOT05 Laboratory Setup 31
Chapter 5: Conclusion 35
Overview 35
Research Approach 35
Future Work 36
References 37

Appendix: PILOT05 System Specifications 39
Introduction... 41
NTCIP Object Definitions for PILOT05 Application Support... 41

LIST OF FIGURES

Page

Figure 1. Research Approach. 3
Figure 2. Experimental Simulation Network. 19
Figure 3. Detector Profile Data. 19
Figure 4. Median Count Profile Recognition. 20
Figure 5. Skewness Count Profile Recognition. 21
Figure 6. Effect of Offset Value on Vehicular Stops. 21
Figure 7. Offset Group Classification Logic for Each Direction of the Arterial. 23
Figure 8. Offset Movement Determination Logic. 24
Figure 9. Algorithm's Shared Memory Structure. 28
Figure 10. PILOT05 Prototype System. 29
Figure 11. Industrial PC. 30
Figure 12. Microcontroller. 31
Figure 13. Wireless Radio. 31
Figure 14. TransLink Lab PILOT05 Master Subsystem. 32
Figure 15. TransLink Installed Master and Slave Subsystems. 32
Figure 16. Message Structure in Communication Subsystem. 33
Figure 17. PILOT05 Detectors Assignment 45
Figure 18. Offset Group Classification Logic for Each Direction of the Arterial. 46
Figure 19. Offset Movement Determination Logic. 47

LIST OF TABLES

Page

Table 1. Adaptive Control Literature Review Summary. 9
Table 1. Adaptive Control Literature Review Summary (continued).......... 10
Table 2. Classification Accuracy Matrix 22
Table 3. Thresholds Structure for Two-Way Offset Tuning with Default Values. 23
Table 4. TS 3.5 Object Definition Hierarchy.......... 42
Table 5. Table of Tables for New Object's ASN Definitions and Example Data 42
Table 6. PILOT05 Detectors Object 43
Table 7. Example of PILOT05 Detectors. 43
Table 8. PILOT05 Physical Channel Table 44
Table 9. PILOT05 Detectors Assignment.......... 45
Table 10. PILOT05 Parameters (table 1 of 2).......... 49
Table 11. PILOT05 Parameters (table 2 of 2).......... 50
Table 12. Example of PILOT05 Parameters.......... 51
Table 13. PILOT05 Count Table 52

CHAPTER 1: INTRODUCTION

OVERVIEW

Researchers in the traffic control field have invested extensive efforts in the development of Adaptive Control Software (ACS) over the last few decades. The most common and recognized ACS systems developed in the United States are the Optimized Policies for Adaptive Control (OPAC) and Real-Time, Hierarchical, Optimized, Distributed, and Effective System (RHODES) *(1, 2)*. From overseas, Split, Cycle, Offset Optimization Technique (SCOOT) *(3)* and Sydney Coordinated Adaptive Traffic System (SCATS) were developed in the United Kingdom and Australia, respectively *(4)*. These Adaptive Control Software systems are based on applying specific proprietary software to a signal control system.

ACS systems have several advantages over traditional control systems. ACS can model and track individual vehicles on a second-by-second basis. An ACS system is not bound by traditional control parameters such as cycles, splits, and offsets. Rather, ACS optimizes the green duration and phase sequencing in real time, providing optimal control of traffic signals. ACS, however, typically requires an extensive input of system parameters for favoring individual movements plus a large number of vehicle detectors to collect movement-specific traffic data. A major drawback of these systems is the extensive effort required for training personnel on the new proprietary architecture.

On a parallel track, the private sector has developed closed-loop systems that are operated by coordinated-actuated controllers. A closed-loop system consists of a master traffic signal controller connected to a series of traffic signal controllers using hardwire connections, fiber-optic cables, or spread spectrum radio. The on-street master supervises the individual intersection controllers and issues commands to implement timing plans stored at the local controllers. The master controller can also report detailed information to a traffic management center using dial-up telephone or other similar communications channel for monitoring purposes.

Closed-loop systems provide actuated control capabilities through their ability to respond to cycle-by-cycle variation in traffic demand while still being able to provide progression for the arterial movement. These systems are widely implemented in Texas arterials to provide efficient operation of arterial intersections while still providing signal progression. Nevertheless, poor progression can be observed along most arterials due to outdated offsets, short-term variations in traffic patterns (early-return-to-green), or changes in the arterial's speed and changes in traffic volumes and waiting queues.

RESEARCH OBJECTIVES

The objectives of this research are to develop, implement, and test an algorithm that automatically fine-tunes offsets in real time in response to changes in traffic patterns measured at an upstream detector. This algorithm will address the limitations of previous efforts in this area to improve progression in both directions of the arterial when feasible. Such an algorithm will reduce traffic congestion and fuel emissions by minimizing vehicle stops and delays at arterial intersections.

Expanding the control logic for modern coordinated-actuated systems to account for problems such as outdated offsets, early-return-to-green, and waiting queues in an adaptive fashion would address many of the day-to-day problems associated with closed-loop systems. Additional training would be minimized, in comparison to fully adaptive systems, since traffic engineers and technicians managing traffic signal operating systems are already familiar with coordinated-actuated logic. Fundamental concepts and communication systems for coordinated-actuated systems would also remain the same, and extra cost would be kept to a minimum.

RESEARCH APPROACH

ACS systems have the greatest potential to provide optimal control of traffic signals. However, ACS systems come with a high price both in terms of initial system cost and operation and maintenance costs. Closed-loop systems operated with a Traffic Responsive Plan Selection (TRPS) mode rank second to ACS and far exceed the performance of closed-loop systems operated with outdated timing plans with time-of-day (TOD) mode *(5)*. Both Adaptive Control Software and closed-loop systems have inherent limitations; assumptions about travel time and platoon dispersion characteristics. Previous research has introduced an innovative "at-the-source" (ATS) method to adaptively fine-tune offsets in real time *(6)*. The ATS method does not suffer from the limitation associated with assumptions about travel time and platoon dispersion characteristics. Augmenting the control strategy with an ATS algorithm can greatly increase the "net benefit" of the control strategy. This approach is illustrated in Figure 1. The developed algorithm addresses the most critical limitation of previous efforts in this area.

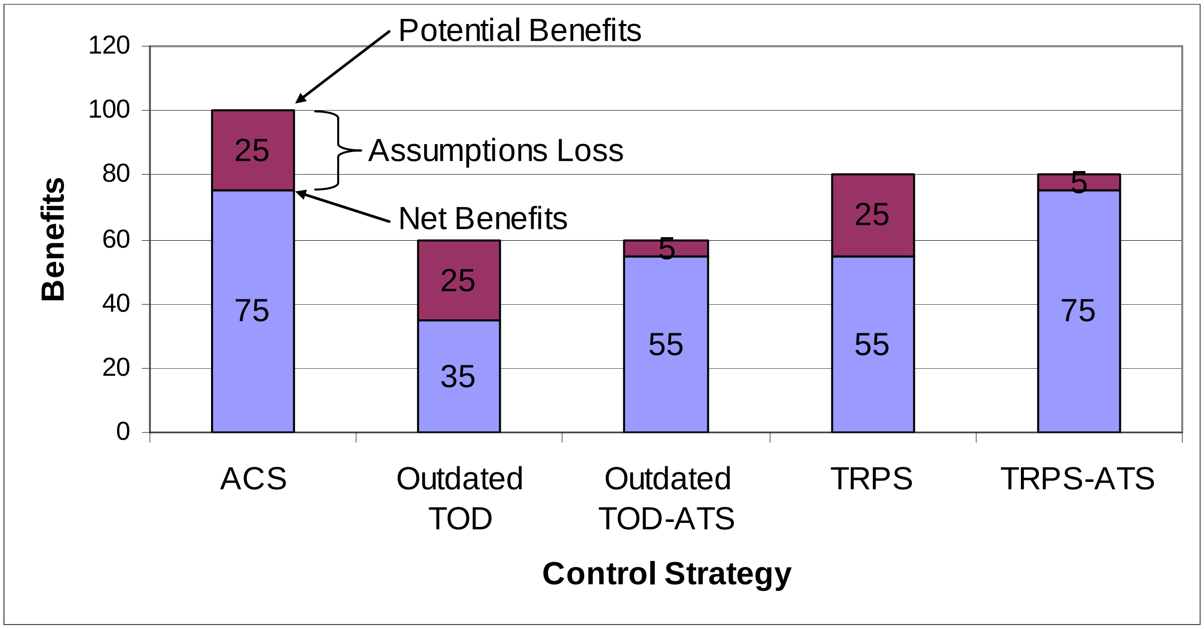


Figure 1. Research Approach.

CHAPTER 2: LITERATURE REVIEW

ADAPTIVE CONTROL SOFTWARE AND ALGORITHMS

There are two primary motivations for development of adaptive control algorithms. The first motivation is the need for the controllers to react to unexpected deviations from historical traffic patterns, either as a result of incidents or day-to-day random variations of the magnitude and temporal distribution of the demand peaks. The second motivation is that even for predicted traffic conditions there are a finite number of time-of-day plans that can be handled by current controllers. Also, pre-selected plans do not typically perform well during periods of temporal transitions in traffic patterns. Adaptive strategies usually respond to changes in traffic patterns in real time, either reactively or proactively. Reactive adaptive strategies “follow” changes in traffic patterns and therefore always lag, whereas proactive strategies try to “predict” changes in traffic patterns, aiming at a better performance.

History of Adaptive Control Software

The Federal Highway Administration (FHWA) started developing a structured approach to centralized traffic signal control, called urban traffic control software (UTCS), in the 1970s. UTCS defined and tested various levels of traffic control, ranging from time-of-day plan selection to real-time adaptive signal timing. Although UTCS did not achieve many of its objectives, it resulted in the development of many concepts and system displays that are currently used in traffic operation centers.

The first generation control (1-GC) of UTCS used a library of pre-stored signal timing plans calculated off-line, based on historical traffic data, in the same way as the pretimed control strategies. The original 1-GC selected a particular timing plan by either time-of-day or pattern matching every 15 minutes. The second generation control (2-GC) used surveillance data and predicted values to compute and implement timing plans in real time. Timing plans were updated no more than once per 10-minute period to avoid transition disturbances from one implemented plan to the next *(7)*. The third generation control (3-GC) differed from the 2-GC in its shorter periods after which the timing plans are revised. The cycle length in 3-GC was also allowed to vary among the signals, as well as the same signal, during the control period *(7)*.

Adaptive Control Algorithms

Experience with the 3-GC in UTCS experiments of the 1970s revealed that new strategies for adaptive control needed to be developed. Adaptive control attempts to achieve real-time optimization of signal operations by using current short-term vehicle information obtained from advanced detectors. However, the performance of the adaptive control system response is entirely dependent on the quality of the prediction model *(7)*. The implementation of adaptive control logic is not always superior to pretimed and actuated control, especially when traffic is highly peaked.

Significant advances in adaptive traffic control were achieved with the introduction of four control strategies, namely SCOOT, SCATS, OPAC, and RHODES. Researchers in the United Kingdom developed SCOOT. It is considered a UTCS-3-GC, although some authors put it into the 2-GC category. SCATS was developed in Australia and is considered to be a variant of UTCS 2-GC. OPAC was introduced by Gartner in the United States and involved the determination of when to switch between successive phases based on actual arrival data at the intersection. RHODES consists of a distributed hierarchical framework that operates in real time to respond to the natural stochastic variation in traffic flow.

SCATS

SCATS calculates degree of saturation (DS) and uses it to make cycle-by-cycle split adjustments based on equal DS. Outside of Australia, SCATS has seen limited deployment. Each SCATS controller has a microcontroller that uses stop line detector information. The philosophy of SCATS is that it has no comprehensive plans; rather, it selects from a library of offsets and phase splits to optimize timing plans in real time. SCATS divides the network into sub-regions with homogeneous flow characteristics. In each region, the intersection with the highest saturation determines the cycle length of the region.

Information from microcontrollers is passed to a central computer, which calculates target timing plans (cycle, split, and offset) in real time to minimize stops in the whole subsystem. Each subsystem can have one or more signals, but only one of them must be critical. As traffic patterns change, the computer at a higher level “marries” and “divorces” intersections by reassigning them to regions with similar flow characteristics. Each subsystem makes independent decisions regarding timing parameters involving cycle, offset, and phase lengths.

The timing plans are incrementally adjusted to varying traffic conditions *(2)*. SCATS has limited deployment in the United States, with the largest deployment in Oakland County, Michigan (575 intersections) *(7)*. However, system operators have not been particularly comfortable with the system because of the significant differences between SCATS conventions and National Electrical Manufacturers Association (NEMA) standards.

SCOOT

SCOOT is based on the Traffic Network Study Tool (TRANSYT) optimization model that runs in a background called the SCOOT Kernel *(8)*. SCOOT uses a central computer and immediate downstream detectors to measure flow profiles at detectors and predicts queues using phase data and estimated vehicle arrivals with dispersion. SCOOT uses cyclic flow profiles (CFP) the same way TRANSYT does. Since SCOOT has evolved from an on-line model of TRANSYT, it can be used to optimize performance indices such as the number of stops, delays, or a mix of both. SCOOT smoothes detected profiles using previous data, detects stationary vehicles at detectors, and takes appropriate action to prevent spillback. The algorithm minimizes average sum of delay due to queues with stop penalties and maintains a degree of saturation under a specified value (i.e., 90 percent). SCOOT also calibrates using specified travel time from detector to stopbar.

SCOOT timing parameters are communicated to the controller immediately. The controller makes incremental adjustments to the cycle lengths, phase lengths, and offsets for the current and next cycles. SCOOT has been installed at 56 intersections in Minneapolis, Minnesota *(7)*. The system showed 19 percent reduction in delay during special events. The main criticism of SCOOT is its inability to handle closely spaced signals due to its particular detection configuration requirements. Another common complaint is that SCOOT’s interface is difficult to handle and its traffic terminologies are different from those used in the United States.

OPAC

OPAC uses detectors placed far upstream of the intersections to predict vehicle arrivals at the intersection and to proactively determine the phase timings. OPAC is a dynamic programming-based heuristic algorithm with rolling horizon, using actual arrival plus projected volumes. The first version of OPAC (OPAC-I) used dynamic programming to minimize delay at the intersection. The main limitation of OPAC-I was its need for elaborate, and most likely very costly, surveillance detectors since it needs the arrival data for the entire planning horizon. The algorithm has gone through several development efforts ranging from OPAC-I through OPAC-RT *(9)*. The last versions of OPAC had several enhancements and added features over OPAC-I, including the ability to optimize all eight phases, skip phases, and an algorithm to coordinate adjacent signals. OPAC is targeted toward oversaturated conditions and demand conditions change for the arterial *(10)*. Simulation-based research showed that OPAC performs better in under-saturated traffic conditions, but limited field implementation revealed good performance for congested traffic conditions. OPAC was implemented on the Reston Parkway in Northern Virginia, and it showed an improvement of 5 to 6 percent over highly fine-tuned timing plans *(10)*.There is no fixed cycle length in OPAC. If smoothed occupancies are larger than a certain threshold, the algorithm allows phases to max-out.

RHODES

Head et al. introduced an adaptive control strategy entitled RHODES in 1992. RHODES is reported to be better than SCOOT and SCATS in the way it proactively responds to the natural stochastic variation in traffic flow *(2)*. RHODES is entirely based on dynamic programming, and it formulates a strategy that makes phase-switching decisions based on vehicle arrival data. The REALBAND algorithm in RHODES minimizes delay to platoons using simulation and a decision tree method. The Controlled Optimization of Phases (COP) algorithm included in RHODES determines phase lengths based on delays and stops using predicted data. Data used include phase and detector data from upstream links and detector data at subject link. RHODES uses predefined turn percentages and estimated travel time between two detector locations.

Like SCOOT, RHODES has the ability to use a variety of performance measures, including delays, queues, and stops. In addition, it allows for phase sequencing to be optimized along within addition to the various timing parameters. Table 1 shows a summary of the four major adaptive control algorithms' characteristics, along with those of other algorithms found in the literature.

Table 1. Adaptive Control Literature Review Summary.

Name	Installations/ Use		Control/ Frequency		Advance Detection		Stopbar Detector Data	Optimization Objective		Distributed System	Peer-to-Peer
	U.S.	Other	Center	Local	Location	Data		System	Local		
SCOOT	Some	Lots	Before phase change for splits and every cycle for offset		Near upstream (u/s) signal	Flow Profile and Occupancy	No	Performance Index (PI) = Delay + Stops	No	No	No
SCATS	Limited use	Many	Strategic	Critical Intersection Control (CIC)			15 ft Gaps	Coordinate for critical cycle and Minimize stops	Equalize DS on all critical phases	Constrained	No
OPAC-RT Version 2.0	Some Tests	No		Yes	400-600 ft from stopbar using 6X6 loops	Volume (V) & Occupancy (O)	No	No	Delay + Stops	Yes	No
UTOPIA/ SPOT	Omaha	Europe Italy, Sweden, Norway	5 minutes	3 seconds	From u/s signals	V & O	At or near stopbar V & O	Level of interaction between signals	Cost Function: Queue-Length, Stops, Wait time and stop to buses, maximum queue, & excess capacity	Yes	Yes. Data exchange every 3 seconds
RHODES/ COP	Some Tests			30-40 second prediction over a rolling horizon	100-130 ft from stopbar	From current and upstream signals	No	Platoon Delay	Delay + Stops	Yes	No

Table 1. Adaptive Control Literature Review Summary (continued).

Name	Installations/ Use		Control/ Frequency		Advance Detection		Stopbar Detector Data	Optimization Objective		Distributed System	Peer-to-Peer
	US	Other	Center	Local	Location	Data		System	Local		
GASCAP	Proto-type		When congest. occurs, use last 15-min data. Update every other fixed cycle.	Real-time split adjustment	600-700 ft	V & O. Keeps a 30-min record of data for use when congested	No	Minimize Q and progress via cycle-offset adjustment	Volume-based phase priority to minimize queue	Yes	No
PRODYN		Limited use		75 second planning horizon	150-160 ft for Q and advance detection at 600 ft from the stopbar or the distance to the upstream signal, whichever is smaller	Queue and Volume			Delay		Yes

DYNAMIC ARTERIAL RESPONSIVE TRAFFIC SYSTEM (DARTS)

DARTS is an open-loop system that was originally developed in the 1970s by Harvey Beierle of the San Antonio District of TxDOT *(11)*. The objective of DARTS was to provide platoon progression by dynamically linking adjacent signalized intersections. DARTS shares some aspects of ACS in the sense that it does not use cycle, splits, and offset parameters. DARTS devices communicate messages about approaching platoons. Commands are executed external to the signal controller in the form of electrical signals applied to the cabinet back panel. DARTS has 14 timing mechanisms and parameters (platoon timer, detector disabled timer, etc.). Each of these timers needs to be set and calibrated to achieve good results. In addition to the cumbersome calibration needs, DARTS still suffers from ACS's drawbacks such as assumptions about platoon arrival time, platoon characteristics and identifications, etc. DARTS is not compatible with traditional coordination parameters such as cycle, splits, and offsets; therefore, it cannot be easily incorporated with closed-loop systems.

CLOSED-LOOP SYSTEMS

Closed-loop systems were mainly developed by the private sector in order to "synchronize" individual intersections to provide arterial progression. A closed-loop system consists of a master traffic signal controller connected to a series of secondary traffic signal controllers using hardwire connections, fiber-optic cables, or spread spectrum radio. The master controller can report detailed information back to a traffic management center using dial-up telephone or other similar communications channel for monitoring purposes.

There are four modes under which closed-loop systems can be operated:

- "Free" mode. In this mode, each intersection runs individually, usually under a fully actuated isolated signal control. This mode can only be efficient if no coordination is needed. It is therefore not recommended for intersections included in a closed-loop system unless under late night light traffic conditions.
- Manual mode. Under this mode, the closed-loop system is operated under a constant plan, unless changed by the system operator. This mode is typically not optimal.

- Time-of-day mode. In this mode, all intersections are coordinated under a common background cycle length. The timing plans are selected at specific times based on historical traffic conditions. TOD is a common mode of operation and can provide a stable and good performance when traffic patterns are predictable. However, in networks where traffic patterns are not predictable or where demands shift with time, TOD can cause the signal system to implement plans that are totally inappropriate for the actual traffic patterns.
- Traffic Responsive Plan Selection. TRPS mode provides a mechanism by which the traffic signal system is able to change timing plans in real time in response to changes in traffic demands. The objective is to enable the signal controller to implement timing plans that are optimal for the traffic conditions that currently exist, rather than for some set of average conditions.

TRPS mode can provide optimal and snappiest operation over all the other closed-loop system operation modes. TRPS mode switches the closed-loop system's current plan to a better plan when unexpected events, incidents, or temporal changes in traffic volume occur. Most importantly, TRPS mode reduces the need for frequent redesign/update of the signal timing plans for new traffic patterns as required if running TOD mode. This later statement stems from the fact that TRPS mode automatically switches plans in response to changes in traffic patterns.

Nevertheless, closed-loop systems are still limited in comparison to Adaptive Control Software. One of the limiting factors is the small number of stored timing plans that the closed-loop system can choose from. Another major limiting factor is the inability of closed-loop systems to react quickly to changes in traffic demand. Even in its optimal mode of operation, TRPS mode, there is always a trade-off between setting the closed-loop system to be very responsive and setting it to be reasonably stable (not bouncing off from one timing plan to another). Besides their own limitations, closed-loop systems share with ACS systems the limitation that they both are dependent on travel time estimations and assumptions. The only difference is that in case of ACS, travel time estimation is used in real time (based on posted speed limits) to estimate arrival time of platoons at the downstream intersection, and in TOD and TRPS, travel time is used to calculate offsets in the design stage and store them in the controller's database.

BRIDGING THE GAP: PRO-TRACTS AND ACS LITE

Purdue Real-Time Offset Transitioning Algorithm for Coordinating Traffic Signals (PRO-TRACTS)

In 2001, researchers at Purdue University developed PRO-TRACTS to fine-tune offsets in real time *(12)*. PRO-TRACTS mitigates the effect of the early-return-to-green problem experienced with coordinated-actuated controllers and accounts for downstream vehicle queues that may impede vehicle progression. The algorithm can be viewed as an integrated optimization approach that is designed to work with traditional coordinated-actuated systems. The objective of the algorithm is to add to the actuated controllers the ability to adaptively change their offsets in response to changes in an arterial's traffic demand, providing an intermediate solution between traditional coordinated-actuated control systems and Adaptive Control Software.

PRO-TRACTS used a novel approach by evaluating the quality of progression near the traffic signal itself *(6)*. This approach eliminated the need to estimate travel time between adjacent signals or any other assumptions about platoons' dispersion characteristics. PRO-TRACTS uses a real-time methodology for estimating the degree of shockwave effect on the coordinated traffic using one detector located 200 to 250 feet upstream of the signal. PRO-TRACTS uses a unique cycle-based tabulation of occupancy and count profiles at the upstream detector to test for the significance of the presence of shockwaves. The philosophy of PRO-TRACTS is that an inappropriate offset causes parts of the platoon to face the red interval, causing several shockwaves to pass through the detector location. These different shockwaves cause high variation in occupancy at the upstream detector. On the other hand, a well-designed offset aligns the green window with the arrival of platoon, minimizing the proportion of traffic arriving at the signal during the red interval and therefore producing a minimal shockwave that does not reach the detector. PRO-TRACTS uses this philosophy to evaluate the existing offset's performance and adjusts it accordingly.

Although PRO-TRACTS uses an innovative "at-the-source" evaluation of progression, the algorithm improves progression in only one direction of the arterial. The impact on the other direction is typically an increase in travel time. PRO-TRACTS also exhibits unstable performance when presented with a high frequency of phase skips and oscillatory traffic patterns caused by spillbacks or lane blockages, due to its reactive nature.

ACS Lite

While the original ACS systems showed promise in field tests, they still lag in deployment efforts. The major reason for this lag is the significant investment needed to switch to completely adaptive systems. Operating agencies are faced with extensive efforts required for training personnel on the new proprietary architecture. In addition to the price of the hardware, implementing adaptive systems requires a large number of vehicle detectors to collect movement-specific traffic data.

Faced with these facts, FHWA started a program that is intended to develop a less complex version of ACS. This version, named ACS Lite, is intended to assist upgrading existing closed-loop control systems to support adaptive control at a moderate initial cost *(13)*. ACS Lite adheres to traditional closed-loop systems signal control parameters (cycle, splits, and offsets). The software creates new signal timing plans in real time, like ACS. However, ACS Lite recomputes timing plans less often, every 10 to 15 minutes. The algorithm is currently under development by Siemens-Gardner Systems and will likely have features similar to TRPS control.

SUMMARY

ACS can provide optimal control of traffic signals due to their ability to optimize the green duration and phase sequencing in real time. Although there are several adaptive control strategies that attempt to adapt to traffic patterns either reactively or proactively, each of these strategies performs differently under different types of conditions. The performance of ACS response is entirely dependent on the quality of the prediction model. The major drawbacks of these systems are:

- Procurement, operation, and maintenance of ACS can be very costly.
- ACS systems require extensive detection infrastructure.
- ACS systems require extensive efforts for training personnel on the new proprietary architecture.

On a parallel track, coordinated-actuated systems continue to be deployed in arterial systems to provide efficient operation with their ability to respond to cycle-by-cycle variations in traffic demand, while still being able to provide progression for the arterial movement. In most cases, coordinated-actuated control saves a significant amount of delay in arterial systems when compared to fixed-time systems. However, closed-loop systems are less optimal in comparison to adaptive systems due to their limited number of timing plans and their limited ability to quickly respond to changes in traffic demand.

The limited ability of closed-loop systems to adapt to traffic variations has stimulated interest in incorporating the technologies of Adaptive Control Software into closed-loop systems. PRO-TRACTS and ACS Lite are two examples of such efforts. Both of these systems have their pros and cons. The objective of this research is to develop an algorithm that improves on previous efforts.

CHAPTER 3: ROBUST OFFSET CLASSIFICATION

INTRODUCTION

Vehicular progression through a closed-loop system can be affected by several traffic and signal parameters. One traffic parameter is the platoon ratio in the traffic stream. Platoon ratio is the percentile of arterial traffic that travels from the first intersection through the last intersection in the system. Another traffic parameter is traffic volume, which itself plays a major role in signal progression. As traffic volume increases, traffic speed decreases and the platoon becomes denser. Signal parameters that might affect the offset are cycle length, green/cycle ratio, and phase sequence. It is also important to note that signal performance also depends on the amount of traffic on minor movements in two folds: (1) the traffic volume on cross streets affects the percentage of traffic turning into the main street, and therefore affects the platoon ratio; and (2) low volume on minor movements results in an "early-return-to-green" situation, where extra green is given back to the coordinated movement. The first year report of this project studied the effect of these parameters on signal progression *(14)*. This report summarizes these findings and introduces robust measures for offset classification.

TRAFFIC PARAMETERS AFFECTING PROGRESSION

Researchers designed and conducted an experiment to study the effect of each of the above-mentioned factors on traffic coordination *(14)*. The experiment was conducted such that:

1) The effect of major arterial movement volume, cross-street volume, cycle length, green/cycle ratio at the first intersection (g1/c), green/cycle ratio at the second intersection (g2/c), phase sequence, and offset value on the system delay and stops can be determined.
2) Combinations of each of the above parameters were simulated using the Corridor Simulation (CORSIM) package *(15)*.
3) Optimal offsets for one-way and two-way progression were calculated from the simulation output. This step was performed by finding the least number of stops and delay in one direction for a given combination of parameters and the associated offset

value with that minimum delay and stops. The step was repeated again, but this time considering the overall delay and stops in both directions of the arterial.

GENERAL FINDINGS

System Delay

Researchers performed a statistical analysis on the simulation output with Statistical Analysis Software (SAS) *(16)*. They found that the longer the cycle, the larger the delay could be. They also found that as the green/cycle ratio at the upstream intersection (g1/c) increases, the effect of other parameters (g2/c, cycle, volumes, etc.) diminishes very quickly. As the g2/c ratio increases, there is still a possibility of high delay in the system.

Arterial Stops

A statistical analysis was also conducted to examine the effect of experiment parameters on the arterial total number of vehicular stops. It was found that the cycle length has little effect on vehicular stops (as long as the optimal offset is in effect). It was also found that the higher the g2/c ratio, the fewer the stops and the higher their variability.

Comparison between One-Way and Two-Way Offsets

A study of the optimized offset values for one-way and two-way progression revealed that there is not much change (or room for optimization) in offset to favor two-way progression versus one-way progression in shorter cycle lengths. Longer cycle lengths provide more room for optimization. This could be attributed to the fact that longer cycle lengths have longer green windows to work with, which is especially important when optimizing phase sequences. These findings were further explored to determine best surrogate measures for robust offset classification, as will be discussed in the next section.

EXPERIMENTAL DESIGN FOR ROBUST OFFSET CLASSIFICATION

In order to increase the stability and robustness of offset classification algorithm, researchers conducted a simulation experiment to obtain representative data points for detector actuations for each of the offset cases examined. CORSIM simulation was used to represent a network of two signals (shown in Figure 2 below) with 75 seconds cycle length. The green duration of the main street green was set to 30 seconds. Fifteen offsets were simulated (5 second increments over the cycle length). The simulation output was analyzed to determine the optimum offset with the least number of vehicle stops. All other offsets were then expressed as their deviation value from the optimum offset. Different statistical parameters were calculated based on the count profile (count values every 5 seconds of the cycle, plotted versus time in cycle) obtained from an advance detector located 150 feet upstream of the signal. The objective was to find the best parameters that can be used to identify optimum offsets from suboptimum ones.

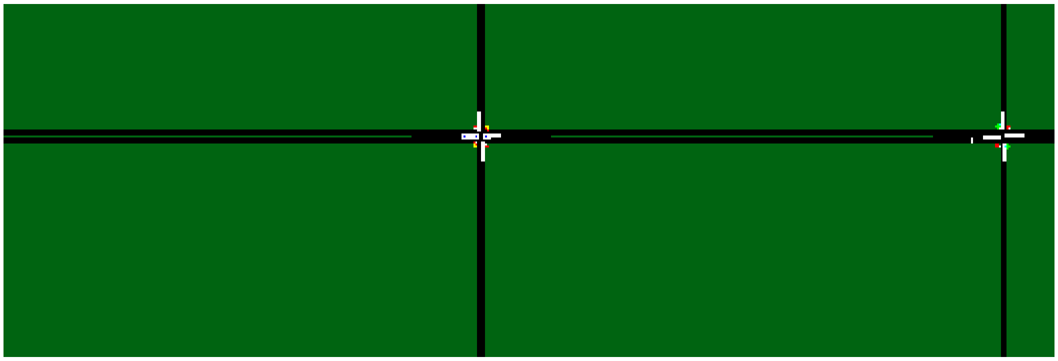
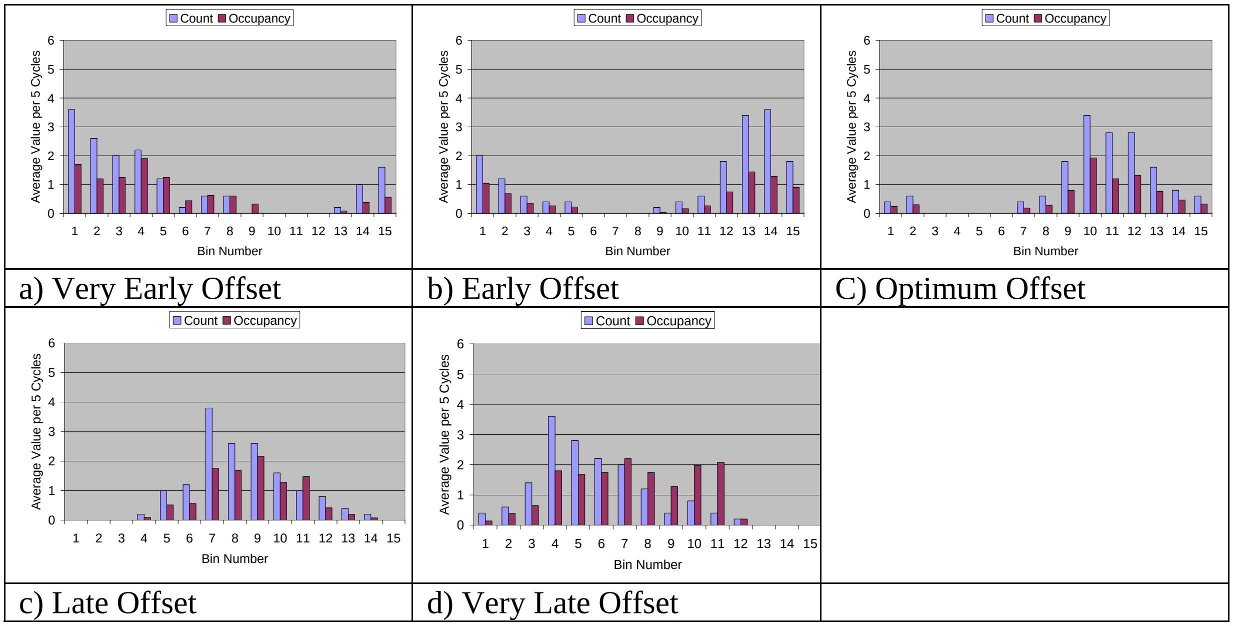


Figure 2. Experimental Simulation Network.

The detector data profiles are shown in Figure 3 below. Notice that the offset values are expressed as five major groups: very early (VE) offset, early (E) offset, optimum (O) offset, late (L) offset, and very late (VL) offset.



a) Very Early Offset

b) Early Offset

C) Optimum Offset

c) Late Offset

d) Very Late Offset

Figure 3. Detector Profile Data.

Offset Classification

The objective of this study was to be able to classify optimum offsets from suboptimum ones. However, stability of the classification algorithm was a major requirement. In order to avoid excessive controller transitioning, researchers decided to make offset improvements in small steps. Rather than being able to tell exactly how deviant the current offset is from the optimum offset, it was deemed enough to classify the offset into the five major groups mentioned above. Although this classification scheme would require only five offset groups, the parameters identified for offset classification were plotted for all 15 offset groups to study the stability of the classification mechanism.

Figure 4 shows a plot of the count profile median versus the offset group. Each group was represented with eight cycle profiles. It was clear from the figure that groups 3, 4, and 5 could be classified from each other. Plus, the groups were distinguished from group 1 and 2. However, group 1 and 2 are not easily distinguished from each other. It was therefore necessary to supplement this parameter with additional parameters for further classification.

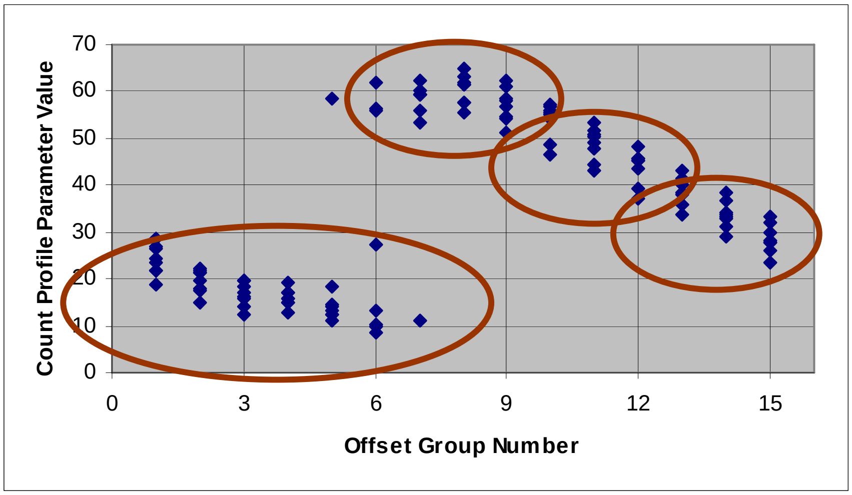


Figure 4. Median Count Profile Recognition.

The second parameter used was the count profile skewness. Figure 5 shows a plot of the profile skewness versus the offset group. Note that with this parameter groups 1 and 2 are easily distinguished from each other. Although groups 4 and 5 are not well separated using this parameter, supplementing the information from the first parameter is sufficient for complete classification of the five groups.

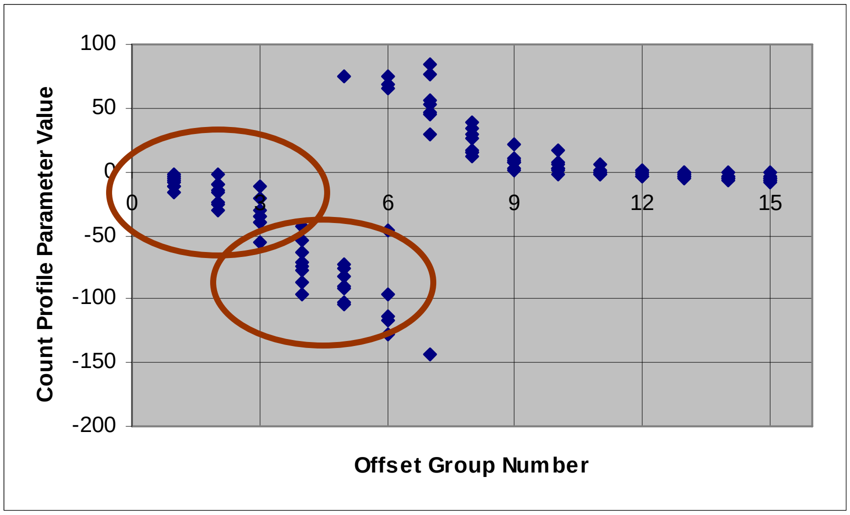


Figure 5. Skewness Count Profile Recognition.

Performance Evaluation

After distinguishing the five offset groups, the next step was to quantify the benefits obtained from this classification. Before and after studies were not conducted for quantification purposes since the total benefit depends on how bad the before case is. Figure 6, for example, shows that vehicular stops can be increased by up to 77 percent if the worst case was chosen as the before case. On the other hand, no benefits are obtained if the before case happens to be the optimum case. For this purpose, only a matrix of accuracy expectation analysis was performed (Table 2). The thresholds between different offset groups were selected such that the optimum offset is always recognized as such, to assure algorithm stability.

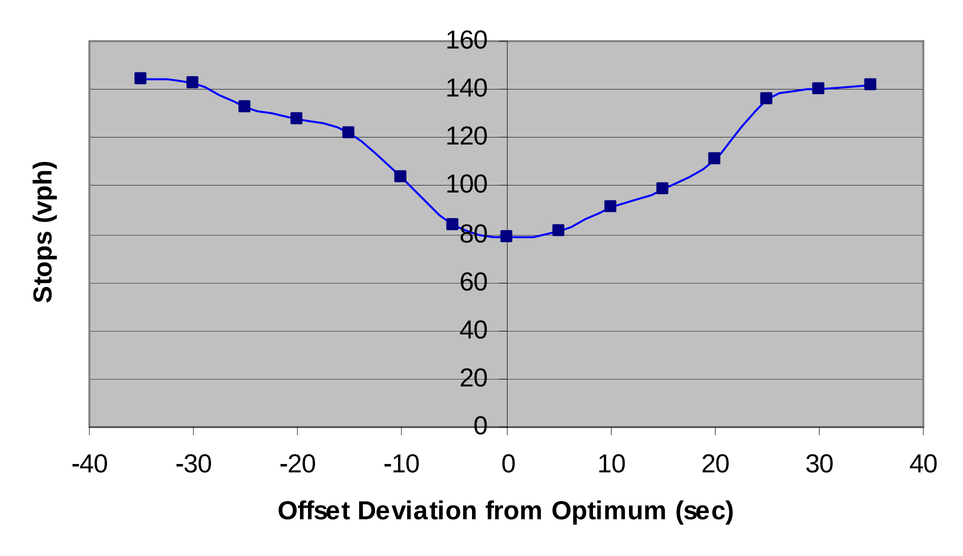


Figure 6. Effect of Offset Value on Vehicular Stops.

Table 2. Classification Accuracy Matrix

Actual Offset Group	Number of Samples	Offset Classified As Group					Accuracy (%)
		1	2	3	4	5	
1	24	20	0	0	0	4	83
2	24	2	22	0	0	0	92
3	24	0	0	24	0	0	100
4	24	0	0	6	18	0	75
5	24	0	0	0	4	20	83

Careful investigation of Table 2 shows that even for lower classification accuracy (e.g., group 4), consequences are not very bad. For example, group 4 (late offset) is misclassified as group 3 (optimum offset) 25 percent of the time. This means that no action will be taken to correct a late offset 25 percent of the time. Figure 6 reveals that a late offset (not a very late offset) does not significantly degrade the performance if left uncorrected.

TWO-WAY OFFSET TUNING ALGORITHM

In order to conduct two-way offset tuning, the algorithm expands the concepts shown in the previous section to evaluate the progression quality on both directions of the arterial. The need for offset tuning, or the lack thereof, in each direction of the arterial is compared to reach a final decision on the direction of offset movement. For example, if the eastbound direction offset is evaluated as a “very early” offset (which means the offset needs to be increased) and the westbound offset is evaluated as optimal, the offset will be increased by one step size. This decision is based on the fact that increasing the offset by one step size will benefit the eastbound traffic tremendously, while not hurting the westbound traffic much. This concept can be made clearer when looking at Figure 6, where it can be seen that small deviations from the optimal offset do not have severe impacts on vehicular stops; it is only when the deviation from optimal offsets are large that severe impacts occur. It should be noted that evaluation of the progression quality occurs only when the system is in sync (i.e., not in transitioning state). National Transportation Communication for ITS Protocol (NTCIP) controllers track the system status and can easily provide this information.

In order to implement this logic, it was necessary to define two sets of thresholds—one for each of the developed system parameters—namely: count median and count skewness. One set of thresholds was used for the count median and the other for the count skewness (Table 3).

Table 3. Thresholds Structure for Two-Way Offset Tuning with Default Values.

Threshold	Threshold Level		
	1	2	3
CountMed	24	40	50
CountSkew	-40	-10	

The algorithm uses the detector data and the above thresholds to determine the offset group classification in each direction. The logic used to determine the offset group is shown in Figure 7 below. This logic is based on the concept illustrated earlier in Figure 4 and Figure 5. The offset group numbers for each direction are then stored in algorithm parameters Dir1Grp and Dir2Grp. Note that only one parameter (DirXGrp) is shown in the figure to avoid redundancy.

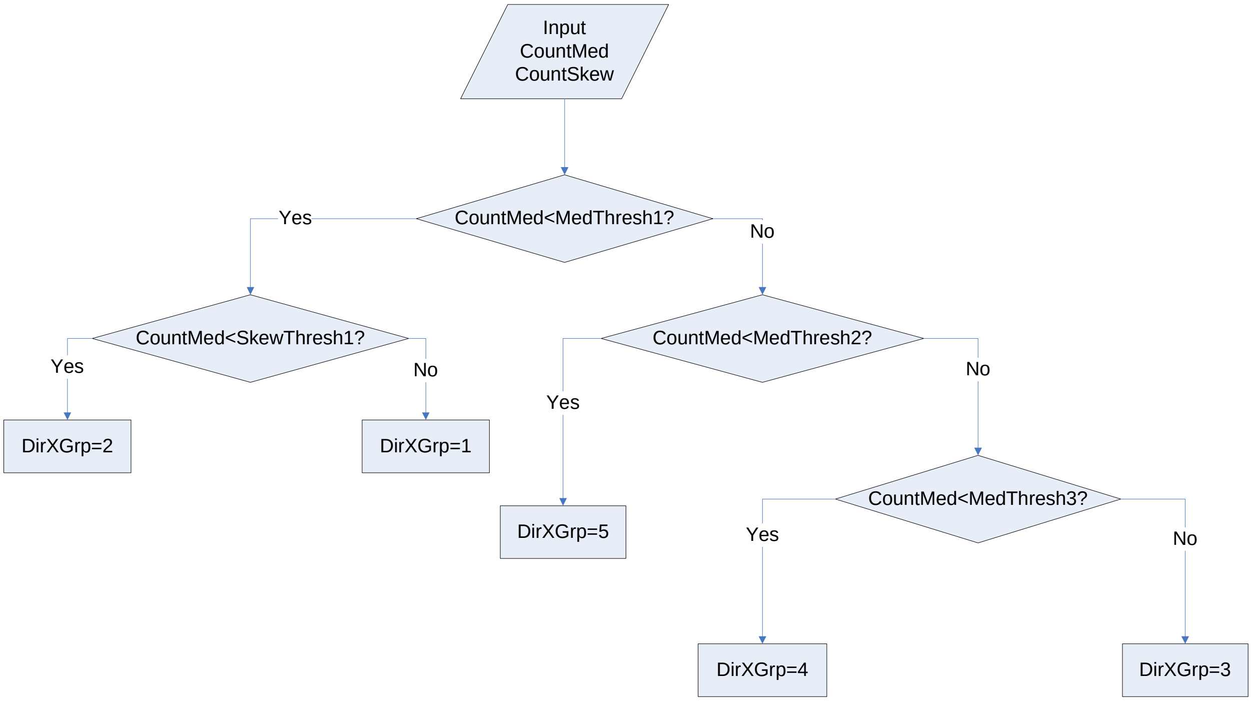


Figure 7. Offset Group Classification Logic for Each Direction of the Arterial.

Once the offset group number in each direction is determined, the direction of offset movement is then decided according to the logic presented in

Figure 8. The logic is based on an implicit preference to direction 1 (if both directions have equal and opposite needs for offset movement). However, direction 1 is not optimized if that action would cause severe consequences to direction 2.

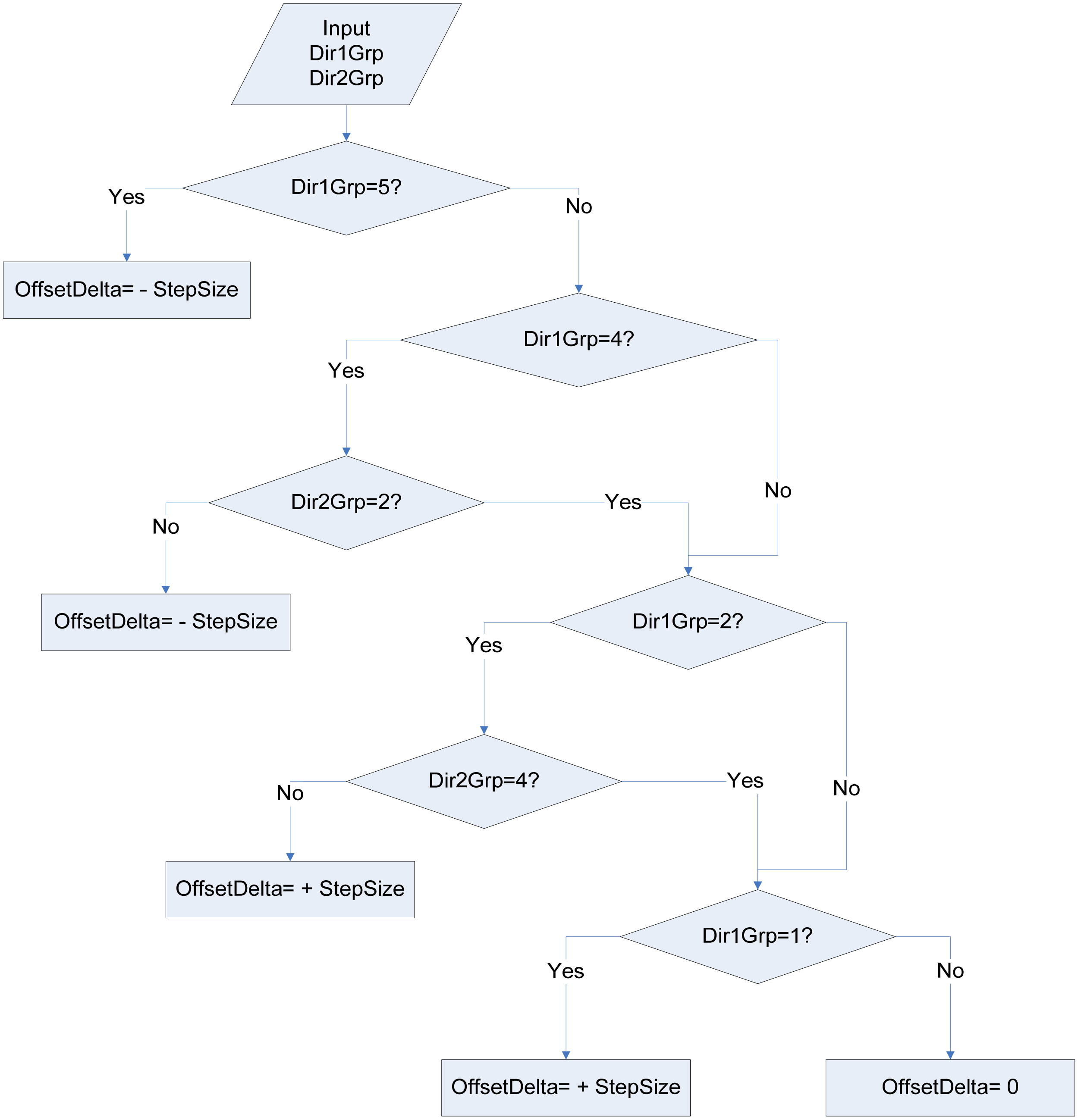


Figure 8. Offset Movement Determination Logic.

Finally, the amount of offset movement is determined by the value stored in algorithm parameter StepSize. Researchers recommend the use of 5 seconds for this value unless traffic conditions at a specific site suggests otherwise.

CHAPTER 4: ALGORITHM’S PROTOTYPE AND FIELD COMMUNICATION STRUCTURE

INTRODUCTION

The objective of the algorithm developed in this research is to evaluate the current offsets, decide whether a new offset is needed, and download any new offset in real time to the traffic controllers in the field. To achieve its objective, the developed system uses various hardware components and a custom software algorithm to monitor, in real time, data elements like phase indications, reason for phase terminations, current plan’s cycle, splits, offset, status of stopbar detectors, and other detectors required by the algorithm at each intersection in the selected arterial test site. The system uses the monitored data to (1) calculate, in real time, the occupancy and count profile over the cycle length of each monitored detector; (2) calculate phase lengths during the cycle; and (3) determine the proper offsets or timing plan to download to the traffic controllers in the field.

ALGORITHM PROTOTYPE

The algorithm developed in this research is designed such that it can be easily evaluated with simulation. The algorithm was named PILOT05. PILOT is an acronym for Pattern Identification Logic for Offset Tuning; 05 is the year of the algorithm development. The module itself is written in C programming language with a shared memory component as shown in Figure 9. The shared memory component facilitates communication between the algorithm and the simulation program. This component is especially important during the evaluation phase of the algorithm, as it allows the use of hardware-in-the-loop simulation before field deployment. The current algorithm prototype evaluates, determines, and downloads offsets to CORSIM—and can therefore be demonstrated in a hardware-in-the-loop simulation environment. The algorithm was developed to address areas of improvement in published algorithms. These areas of improvements include enhancements to algorithm stability to accommodate phase skips and cyclic platoon patterns, transitioning mechanisms, and activation mechanisms. Development related to field implementation includes interface with the field communication module, which is described in the next section.

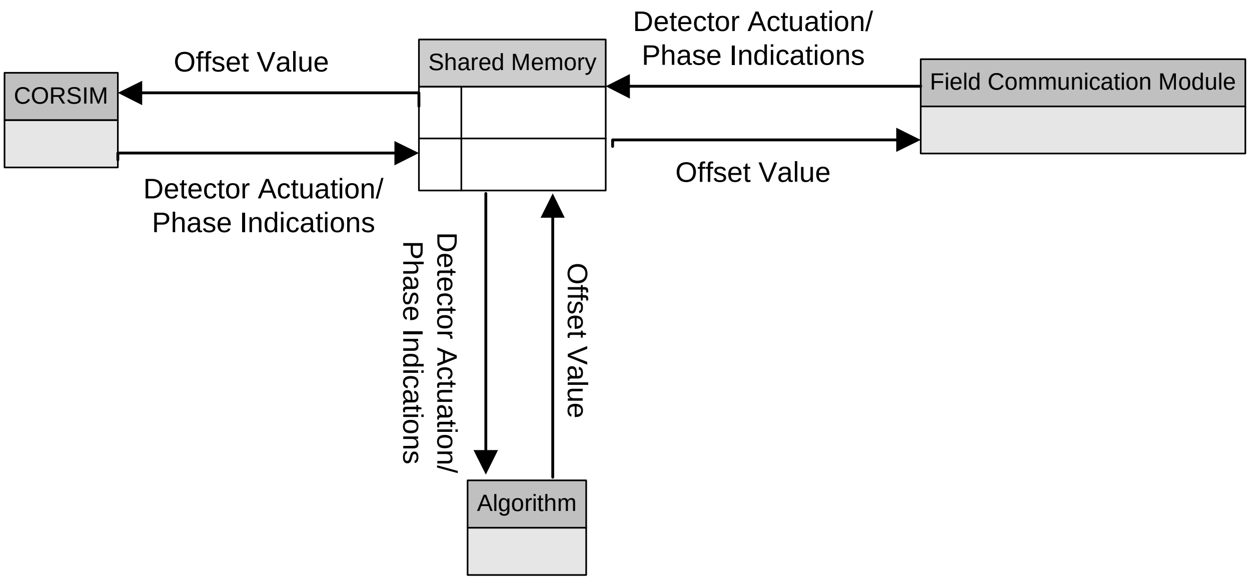


Figure 9. Algorithm’s Shared Memory Structure.

FIELD COMMUNICATION MODULE

As illustrated in Figure 10, the field communication module consists of two components or subsystems: the master subsystem and a number of slave subsystems. The master subsystem consists of an industrial PC that runs the algorithm itself and issues commands to the slave subsystems. Each slave subsystem consists of a microcontroller that houses and runs the slave subsystem’s custom software algorithm. The master subsystem is connected to the slave subsystem with wireless radios. The following sections describe in more detail the master and slave subsystem’s software algorithms, specification of the hardware components used by each subsystem, and communications between the subsystems.

Master Subsystem

The master subsystem (Figure 10) consists of an industrial PC that resides in the cabinet where the master controller resides at the selected arterial test site or in a central office depending on:

- the communication infrastructure available at the selected test site,
- the master controller used by the locality in charge of the arterial site, and
- the PILOT05 system final design.

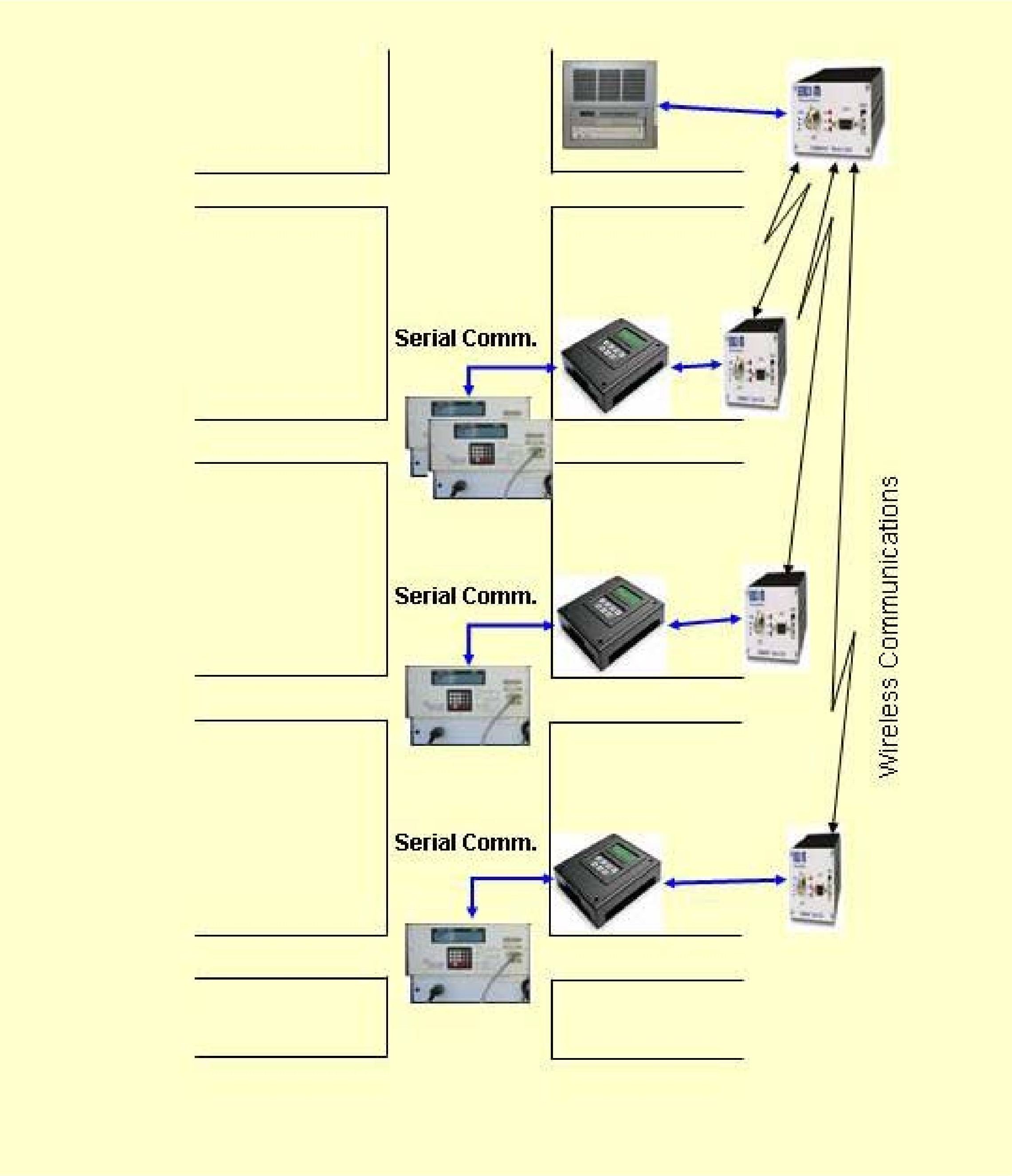


Figure 10. PILOT05 Prototype System.

The master subsystem software algorithm resides and runs on the industrial PC (Figure 11). The master subsystem communicates with slave subsystems either over an ENCOMM 5200 wireless transceiver or through the communication infrastructure available at the selected arterial test site. The master subsystem software algorithm receives data collected and calculated by slave subsystems including occupancy and count profiles over the cycle length of stopbar and algorithm detectors, phase durations during each cycle, and reason for phase termination. The master software algorithm determines the optimal offset or timing plan in real time based on the data received from slave subsystems and transmits the new offset or timing plan to slave subsystems to download to the field controllers in real time.

Figure 11. Industrial PC.

Slave Subsystem

As shown in Figure 12, the slave subsystem consists of a ZWORLD BL2100 microcontroller running the Rabbit 2000 microprocessor at 22.1 MHz and an ENCOMM 5200 transceiver. The slave subsystem hardware components reside in the same cabinet as the traffic controller at each intersection in the selected arterial test site. The slave subsystem software algorithm resides and runs on the BL2100 microcontroller. The slave subsystem software algorithm communicates regularly in real time with the traffic controller in the cabinet and monitors the phase indications, reason for phase termination, stopbar and algorithm detectors' occupancy and count profiles, current plan's cycle, splits, and offset, and calculates phase durations. For cabinets equipped with an NTCIP-compliant traffic controller, the microcontroller collects the required data by exchanging NTCIP standard messages with the controller over an RS-232 serial port.

Figure 12. Microcontroller.

Communication Subsystem

The information collected by the BL2100 microcontroller is transmitted on a regular basis to the master subsystem via the ENCOMM 5200 wireless transceiver (shown in Figure 13). The slave subsystem also receives through the ENCOMM 5200 wireless transceiver the offset or timing plan to be downloaded to the local controller from the master subsystem.

Figure 13. Wireless Radio.

PILOT05 Laboratory Setup

The prototype laboratory installation of PILOT05 at Texas Transportation Institute (TTI) utilizes a digital input/output (I/O) module of the microcontroller to interface with the cabinet's back panel and monitor in real time the phase indications, stopbar, and algorithm detectors' occupancy and count profiles, current plan's cycle, splits, and offset, and phase durations. For cabinets equipped with an NTCIP-compliant traffic controller, the microcontroller collects the required data by exchanging NTCIP standard messages with the controller over an RS-232 serial port. The slave subsystem collects the counts and occupancy of the algorithm detectors in periods of X seconds, where X is specified by the user. At the end of each cycle, the slave algorithm calculates the occupancy and count profiles for the algorithm detectors over the cycle length and sends a message with the information to the master system. Figure 14 shows a master subsystem installed in the TransLink lab for algorithm evaluation and testing. Figure 15 shows the TransLink lab installed master and slave subsystems.

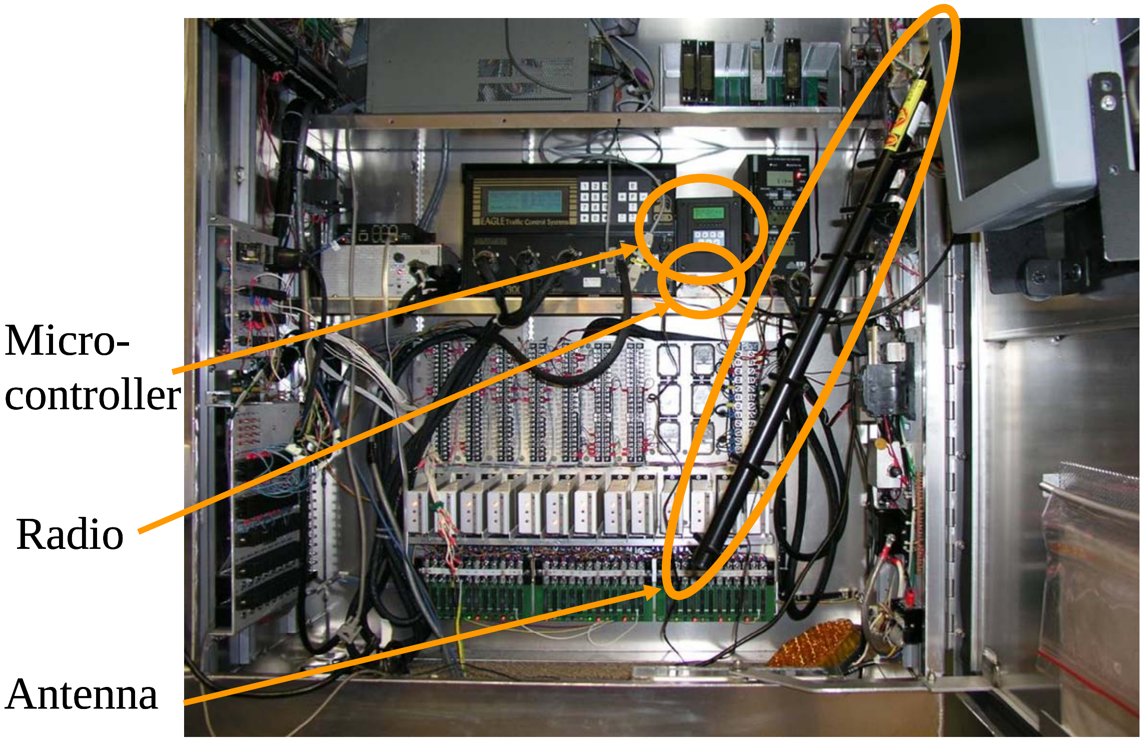


Figure 14. TransLink Lab PILOT05 Master Subsystem.

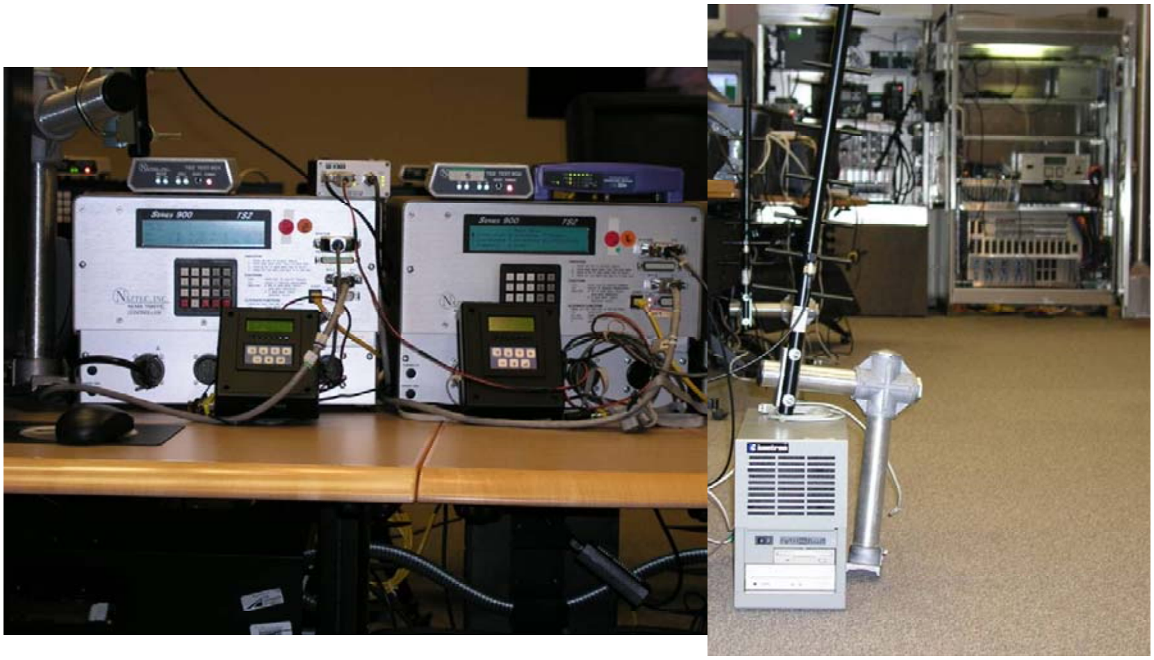


Figure 15. TransLink Installed Master and Slave Subsystems.

The information collected by the BL2100 microcontroller slave subsystems is transmitted at the end of each cycle to the master system via the ENCOMM 5200 wireless transceiver. The slave subsystem also receives through the ENCOMM 5200 wireless transceiver the offset or timing plan to download to the local controller from the master subsystem. The ENCOMM 5200 radio connected to the master system is configured as the master and the ENCOMM 5200 radios connected to the BL2100 microcontrollers in the traffic cabinets are configured as slaves on the same network. The ENCOMM radio network was tested extensively in the lab, and no interference was detected in the different messages received by the master from the slave subsystems even if the slaves transmitted simultaneously. The protocol used for exchanging messages between the master and the slave microcontrollers was designed to be very simple due to the negligible interference that was detected by the master from messages sent by the slave subsystems. Figure 16 illustrates the contents of the message sent by the slave subsystem to the master system at the end of each cycle. The various fields in the messages exchanged between the master system and the slave subsystems are comma delimited. The controller's cycle end is determined by the end of green of one of the phases that is specified by the user.

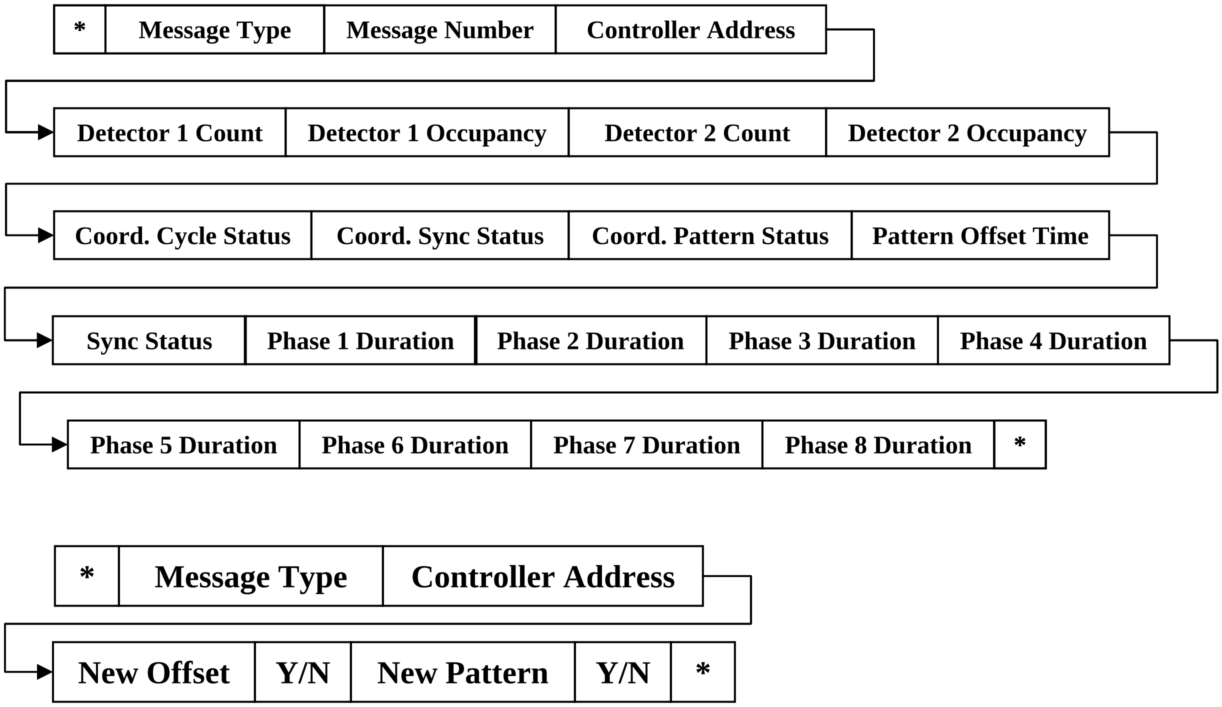


Figure 16. Message Structure in Communication Subsystem.

CHAPTER 5: CONCLUSION

OVERVIEW

ACS has several advantages over traditional control systems. An ACS system is not bound by traditional control parameters such as cycles, splits, and offsets. Rather, ACS systems optimize the green duration and phase sequencing in real time, providing optimal control of traffic signals. ACS, however, typically requires an extensive input of system parameters for favoring individual movements plus a large number of vehicle detectors to collect movement-specific traffic data. A major drawback of these systems is the extensive effort required for training personnel on the new proprietary architecture.

On the other hand, closed-loop systems provide actuated control capabilities through their ability to respond to cycle-by-cycle variation in traffic demand while still being able to provide progression for the arterial movement. These systems are widely implemented in Texas arterials to provide efficient operation of arterial intersections while still providing signal progression. Nevertheless, poor progression can be observed along most arterials due to outdated offsets, short-term variations in traffic patterns (early-return-to-green), or changes in arterial's speed and changes in traffic volumes and waiting queues.

This research aims at augmenting commonly installed closed-loop systems with the abilities of an adaptive control system. This work addresses some of the major limitations of previous research in this area. Specifications are included in the appendix of this report to facilitate vendor implementation of the system.

RESEARCH APPROACH

ACS has the greatest potential to provide optimal control of traffic signals. However, ACS comes with a high price both in terms of initial system cost and in operation and maintenance costs. Closed-loop systems operated with a TRPS rank second to ACS and far exceed the performance of closed-loop systems operated with outdated timing plans with TOD mode. Both Adaptive Control Software and closed-loop systems have inherent limitations: assumptions about travel time and platoon dispersion characteristics. Previous research introduced an innovative "at-the-source" (ATS) method to adaptively fine-tune offsets in real time. This research further explores the ATS expansion and application to two-way progression with robust offset classification structure.

FUTURE WORK

The PILOT05 system contributes to the research of real-time progression evaluation and improvement by focusing efforts on (1) a robust classification algorithm of progression quality and remedies and (2) vendor-independent distributed implementation architecture for the proposed algorithm. This research and development effort establishes a flexible development framework for theoretical expansion and experimentation of both theoretical and implementation elements of adaptive control algorithms.

Further investigation of NTCIP communication protocol compliance and integration with TRPS features is recommended. It is also recommended to coordinate the development and improvement of ATS technology with ACS Lite efforts by seeking active involvement of the FHWA.

REFERENCES

1. Gartner, Nathan H. OPAC: A Demand-Responsive Strategy for Traffic Signal Control. *Transportation Research Record 906.* Transportation Research Board, National Research Council, Washington DC, 1983.

2. Head, K. Larry, Pitu B. Mirchandani, and Dennis Sheppard. Hierarchical Framework for Real-Time Traffic Control. *Transportation Research Record 1360,* Transportation Research Board, National Research Council, Washington DC, 1992.

3. Hunt, P. B., D. I. Robertson, R. D. Bretherton, and R. I. Winton. *SCOOT- A Traffic Method of Coordinating Signals*. Laboratory Report No. LR 1014. Transportation and Road Research, Crowthorne, Bershire, England, 1981.

4. Lowrie, P. R. *SCATS – Sydney Coordinated Adaptive Traffic System – A Traffic Responsive Method of Controlling Urban Traffic*. Roads and Traffic Authority, Sydney Australia, 1992.

5. Abbas, M. M., N. A. Chaudhary, A. Sharma, S. P. Venglar, and R. J. Engelbrecht. *Methodology for Determination of Optimal Traffic Responsive Plan Selection Control Parameters.* Research Report 4421-1. Texas Transportation Institute, College Station, TX, September 2003.

6. Abbas, M., and D. Bullock. On-Line Measure of Shockwaves for ITS Applications. *ASCE Journal of Transportation Engineering*, Vol. 129, No. 1, 2003. pp. 1-6.

7. Hicks, B., and M. Carter. *What Have We Learned About Intelligent Transportation Systems?* Chapter 3 – A. What Have We Learned About ITS? Arterial Management. U.S. Department of Transportation, Federal Highway Administration, Washington DC, December 2000.

8. Robertson, D. I. *TRANSYT: A Traffic Network Study Tool.* Report No. LR 253. Transport and Road Research Laboratory, United Kingdom, 1969.

9. Gartner, Nathan H., Chronis Stamatiadis, and Phillip J. Tarnoff. Development of Advanced Traffic Signal Control Strategies for Intelligent Transportation Systems: Multilevel Design*.* *Transportation Research Record 1494,* Transportation Research Board, National Research Council, Washington DC, 1995.

10. Andrews, C. M., and S. M. Elahi. *Evaluation of New Jersey Route 18 OPAC/MIST Traffic Control System.* Presented at the 76th Annual Meeting of the Transportation Research Board, Washington, DC, January 1997.

11. Lee, Clyde E. *Evaluation of the Dynamic Arterial-Responsive Traffic System (DARTS).* Research Report No. FHWA/TX-82/25+243-1F, Center for Transportation Research, Austin, TX, August 1982.

12. Abbas, Montasir M., Darcy Bullock, and Larry Head. Real Time Offset Transitioning Algorithm for Coordinating Traffic Signals. *Transportation Research Record 1748*, Transportation Research Board, National Research Council, Washington DC, 2001.

13. *FHWA Awards Contract to Develop ACS Lite.* Newsletter of the ITS Cooperative Deployment Network. http://www.nawgits.com/fhwa/acslite.html. Accessed 2003. [Author: remove link from access date.]

14. Abbas, M., N. Chaudhary, H. Charara, and Y. Jung. *Foundation of Self-Tuning Closed-Loop Progression System.* Research Report 4729-1. Texas Transportation Institute, College Station, TX, September 2004.

15. *CORSIM User's Manual*. ITT Systems & Sciences Corporation. FHWA, U.S. Department of Transportation, Washington DC, 2003.

16. SAS Institute, Inc., *SAS/STAT User's Guide,* Version 6., SAS Institute , Cary, North Carolina, 1989.

APPENDIX:
PILOT05 SYSTEM SPECIFICATIONS

INTRODUCTION

The NTCIP standard supporting traffic signal control has been evolving at a relatively slow rate. Research conducted in this project revealed that many NTCIP controller features are yet to be fully implemented in traffic controllers. Nevertheless, researchers believe that it is very important to develop and present the PILOT05 system specification in NTCIP format. This belief stems from the fact that NTCIP assures interchangeability and interoperability of signal control equipment. The following section provides the Abstract Syntax Notation (ASN) for the PILOT05 objects. The ASN for PILOT05 builds on and enhances features of PRO-TRACTS to provide more stable signal operation.

NTCIP OBJECT DEFINITIONS FOR PILOT05 APPLICATION SUPPORT

PILOT05 stores and processes detector actuation at the controller level rather than sending all count profile values to the central master. This approach enables a significantly lower bandwidth demand while using NTCIP. Tabulating the detector information at the controller level also allows the controllers to work independently if necessary. For example, controllers at the coordination system boundaries can "fine-tune" themselves to the arrival pattern if vehicle platoons at the boundaries existed.

This section defines the proposed NTCIP object for capturing detector actuation at the controller level, patterned on the NTCIP TS 3.5 standards. The attributes of the new object are described and the ASN for each attribute is provided. The X label at the beginning of the defined ASN is the node to which the object is attached. Most likely, X will be 8 in section 2.3 of TS 3.5 object definition hierarchy shown in Table 4. In such a case, all "X" labels in the following section would be replaced by the notation "2.3.8." Table 5 shows a table of tables that provides the reader with a link between this chapter's ASN definitions and example data for the new object's attributes. Note that supporting PILOT05 implementation is optional and therefore the PILOT05 detectors object is optional. However, if the controller supports PILOT05 implementation, it must support all of its object attributes, and therefore all of the attributes are mandatory.

Table 4. TS 3.5 Object Definition Hierarchy.

TS 3.5 Section	Object Definition	
2.1	MIB Header	
2.2	Phase Parameters	
2.3	Detector Parameters	
	2.3.1	Maximum Vehicle Detectors
	2.3.2	Vehicle Detector Parameter Table
	2.3.3	Maximum Vehicle Detector Status Group
	2.3.4	Vehicle Detector Status Group Table
	2.3.5	Volume/Occupancy Report
	2.3.6	Maximum Pedestrian Detectors
	2.3.7	Pedestrian Detector Parameter Table
2.4	Unit Parameters	
2.5	Coordination Parameters	
2.6	Time Base Parameters	
2.7	Preempt Parameters	
2.8	Ring Parameters	
2.9	Channel Parameters	
2.10	Overlap Parameters	
2.11	TS2 Port 1 Parameters	

Table 5. Table of Tables for New Object's ASN Definitions and Example Data.

ASN Definition	Example Data
Table 6	Table 7
Table 8	Table 9
Table 10 and Table 11	Table 12

NTCIP TS 3.5 standards define the detectors' objects for NTCIP-compliant controllers. From those detectors, only a limited number need to support PILOT05 implementation. Although current implementation of PILOT05 requires only one detector per device (controller), future work might need more than one detector. For example, to account for two-way coordination, a PILOT05 detector will be required on at least two different approaches.

The maximum number of PILOT05 detectors, along with the number of PILOT05 detectors currently activated in the controller, is determined by the max PILOT05Detectors and active PILOT05Detectors parameters, respectively, as described in Table 6. Example values of max PILOT05Detectors and active PILOT05Detectors are shown in Table 7.

Table 6. PILOT05 Detectors Object.

X PILOT05detectors

PILOT05Detector OBJECT IDENTIFIER

::={detector X}

-- This node contains the object necessary to support PILOT05application

X.1 Maximum PILOT05detectors

max PILOT05Detectors OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

“The maximum number of PILOT05detectors supported in this device. This value indicates how many rows are in the physicalChannelsTable object.”

X.2 Active PILOT05detectors

active PILOT05Detectors OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS mandatory

DESCRIPTION

“The number of PILOT05detectors in this device. This object value multiplied by the value of cyclesSaved object indicates how many rows are in the countProfileTable objects.”

Table 7. Example of PILOT05 Detectors.

Max PILOT05Detectors	20
Active PILOT05Detectors	6

Since PILOT05 detectors are basically advanced detectors, they possess all of the attributes of the ordinary detectors plus some special attributes. The relationship between PILOT05 detectors and the advanced detectors is specified in the physical channels table (Table 8). This table maps PILOT05 detectors to the controller detectors object. Table 9 shows an example of a PILOT05 detectors assignment to the controller’s detectors of the intersection shown in Figure 17. In this figure, controller detector number 2 is declared as the first PILOT05 detector, while controller detector number 10 is declared as the second PILOT05 detector. The third detector is shown to include both detectors 2 and 10.

Table 8. PILOT05 Physical Channels Table.

X.3 Physical Channels Table

```
physicalChannelsTable OBJECT-TYPE
SYNTAX SEQUENCE OF physicalChannelsEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
        “A table containing physical channel definitions for PILOT05detectors.”
physicalChannelsEntry OBJECT-TYPE
SYNTAX physicalChannelsEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
        “The physical channel address for one of PILOT05detectors.”
INDEX {PILOT05DetectorNumber}
physicalChannelsEntry::= SEQUENCE{
PILOT05DetectorNumber        INTEGER,
PILOT05DetectorChannel INTEGER}
```

X.3.1 PILOT05Detector Number

```
PILOT05DetectorNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
Status mandatory
DESCRIPTION
        “PILOT05Detector Number for this entry. This value shall not exceed active
PILOT05Detectors object value.”
```

X.3.2 PILOT05Detector Channel

```
PILOT05DetectorChannel OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
Status mandatory
DESCRIPTION
        “PILOT05detector channel for this entry. This value shall not exceed
vehicleDetectorNumber* object value.”
```

* Defined in TS 3.5 section 2.3.2.

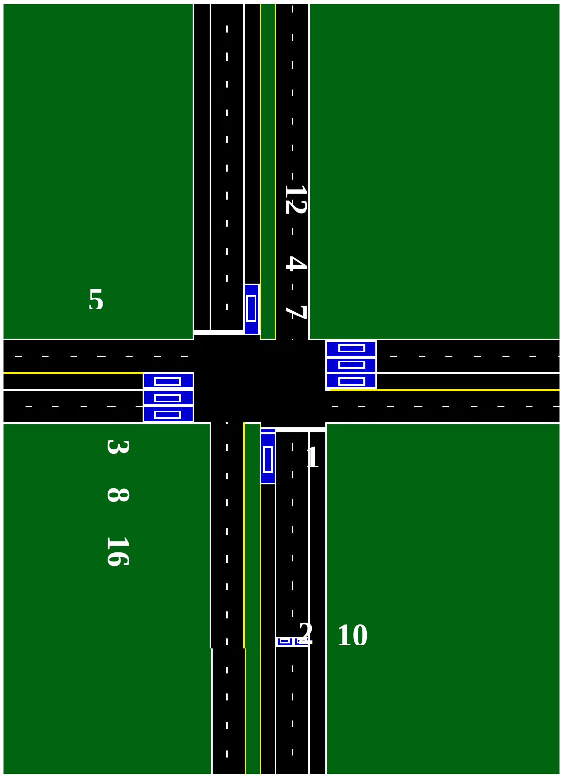


Figure 17. PILOT05 Detectors Assignment.

Table 9. PILOT05 Detectors Assignment.

PILOT05Detectors (X)	Intersection Detectors (2.3)															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1		X														
2										X						

The PILOT05 parameters table lists the thresholds needed for activation of PILOT05. In order to conduct two-way offset tuning, the algorithm expands the concepts shown in the previous section to evaluate the progression quality on both directions of the arterial. The need for offset tuning, or the lack thereof, in each direction of the arterial is compared to reach a final decision on the direction of offset movement. For example, if the eastbound direction offset is evaluated as a "very early" offset (which means offset needs to be increased) and the westbound offset was evaluated as optimal, the offset will be increased by one step size. This decision is based on the fact that increasing the offset by one step size will benefit the eastbound traffic tremendously, while not hurting the westbound traffic much. This concept stems from the fact that small deviations from the optimal offset do not have severe impacts on vehicular stops; it is only when the deviation from optimal offsets are large that severe impacts occur. It should be noted that evaluation of the progression quality occurs only when the system is in sync (i.e., not in transitioning state). NTCIP controllers track the system status and can easily provide this information.

In order to implement this logic, it was necessary to define two sets of thresholds—one for each PILOT05 parameter—namely: count median and count skewness. One set of thresholds was used for the count median and the other for the count skewness. The algorithm uses the detector data and the above thresholds to determine the offset group classification in each direction. The logic used to determine the offset group is shown in Figure 18 below. The offset group numbers for each direction are then stored in PILOT05 parameters Dir1Grp and Dir2Grp. Note that only one parameter (DirXGrp) is shown in the figure to avoid redundancy.

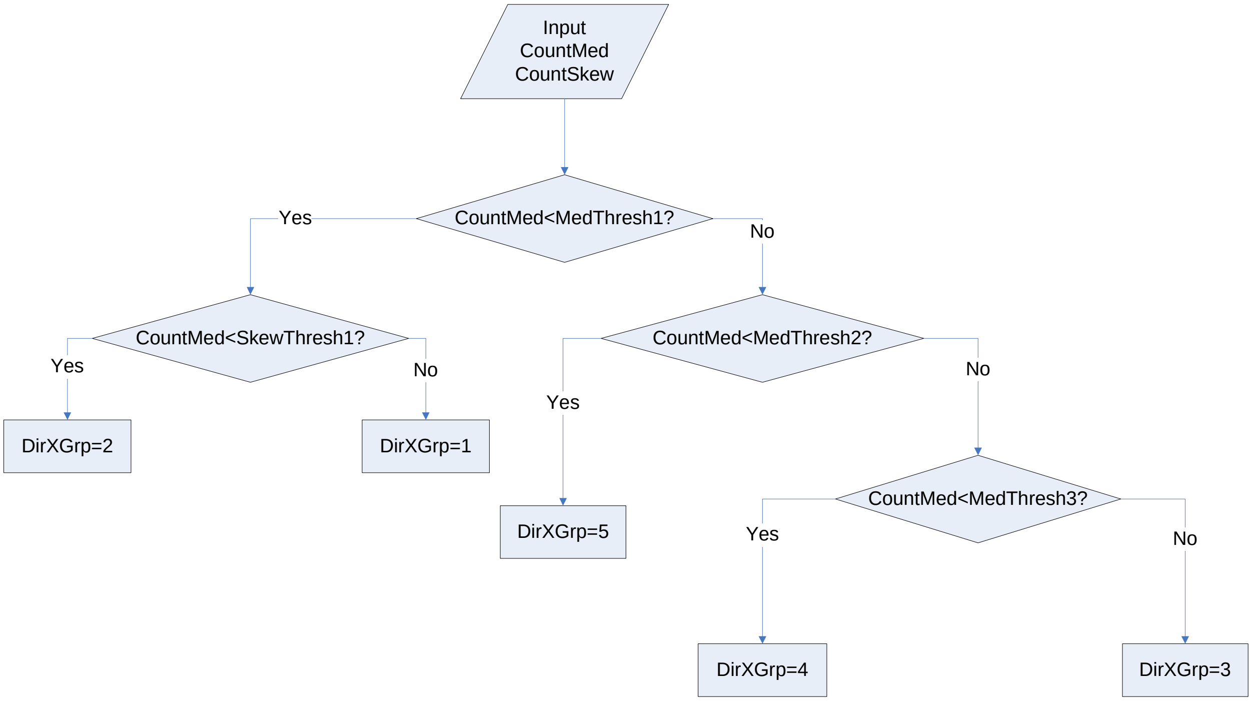


Figure 18. Offset Group Classification Logic for Each Direction of the Arterial.

Once the offset group number in each direction is determined, the direction of offset movement is then decided according to the logic presented in Figure 19. The logic is based on an implicit preference to direction 1 (if both directions have equal and opposite needs for offset movement). However, direction 1 is not optimized if that action would cause severe consequences to direction 2.

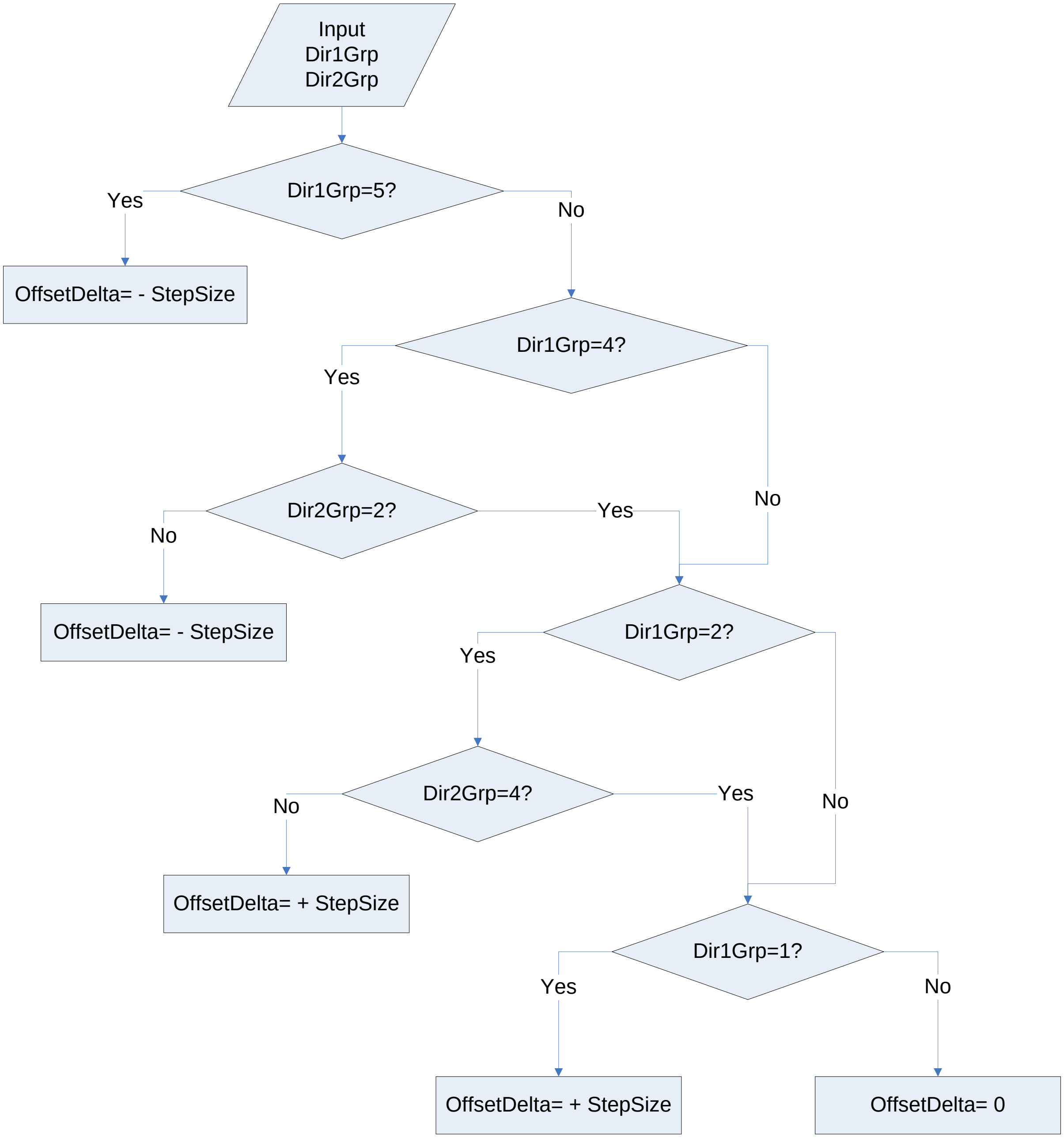


Figure 19. Offset Movement Determination Logic.

Finally, the amount of offset movement will be determined by the value stored in PILOT05 parameter StepSize. Researchers recommend the use of 5 seconds for this value, unless the traffic condition at a specific site suggests otherwise. All PILOT05 parameters necessary for the algorithm to function properly are specified in the PILOT05 parameter table (Table 12). The ASN definitions for those parameters are defined in Table 10 and Table 11 and example data are shown in Table 12.

Table 10. PILOT05 Parameters (table 1 of 2)

X.4 PILOT05Parameter Table

```
PILOT05Parameters OBJECT-TYPE
SYNTAX PILOT05Parameters
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
        “PILOT05 control parameters.”
PILOT05Parameters::= SEQUENCE{
CountMedThresh1         INTEGER,
CountMedThresh2         INTEGER,
CountMedThresh3         INTEGER,
CountSkewThresh1        INTEGER,
CountSkewThresh2        INTEGER,
StepSize        INTEGER,
Dir1Grp         INTEGER,
Dir2Grp         INTEGER}
```

X.4.1 CountMedThresh1

```
CountMedThresh1  OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
Status mandatory
DESCRIPTION
        “Count Median threshold + 100 used to determine offset classification group that results
in invoking PILOT05 to apply offset value changes in the controller. The amount of offset change
is determined by StepSize parameter. This value should range between 0 and 255.”
```

X.4.2 CountMedThresh2

```
CountMedThresh2  OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
Status mandatory
DESCRIPTION
        “Count Median threshold + 100 used to determine offset classification group that results
in invoking PILOT05 to apply offset value changes in the controller. The amount of offset change
is determined by StepSize parameter. This value should range between 0 and 255.”
```

X.4.3 CountMedThresh3

```
CountMedThresh3  OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
Status mandatory
DESCRIPTION
        “Count Median threshold + 100 used to determine offset classification group that results
in invoking PILOT05 to apply offset value changes in the controller. The amount of offset change
is determined by StepSize parameter. This value should range between 0 and 255.”
```

Table 11. PILOT05 Parameters (table 2 of 2)

X.4.4 CountSkewThresh1

CountSkewThresh1 OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
Status mandatory
DESCRIPTION

"Count Skewness threshold + 100 used to determine offset classification group that results in invoking PILOT05 to apply offset value changes in the controller. The amount of offset change is determined by StepSize parameter. This value should range between 0 and 255."

X.4.5 CountSkewThresh2

CountSkewThresh2 OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
Status mandatory
DESCRIPTION

"Count Skewness threshold + 100 used to determine offset classification group that results in invoking PILOT05 to apply offset value changes in the controller. The amount of offset change is determined by StepSize parameter. This value should range between 0 and 255."

X.4.6 StepSize

StepSize OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
Status mandatory
DESCRIPTION

"A parameter that specifies by how many seconds PILOT05 increments or decrements the offset once thresholds warrant."

X.4.7 Dir1Grp

Dir1Grp OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
Status mandatory
DESCRIPTION

"A parameter that specifies the group to which the offset in the first direction has been classified."

X.4.8 Dir2Grp

Dir2Grp OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
Status mandatory
DESCRIPTION

"A parameter that specifies the group to which the offset in the second direction has been classified."

Table 12. Example of PILOT05 Parameters

PILOT05 Parameter	Value
CountMedThresh1	124
CountMedThresh2	140
CountMedThresh3	150
CountSkewThresh1	60
CountSkewThresh1	90
StepSize	5

The count table stores the count profiles over all saved cycles and for all PILOT05 detectors. The ASN definition is shown in Table 13.

Table 13. PILOT05 Count Table

X.8 Count Table
countProfileTable OBJECT-TYPE
SYNTAX SEQUENCE OF countProfileEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION

“A table containing count profile collected for all PILOT05 detectors over the cycles saved in this unit. The number of rows in this table is equal to the active PILOT05Detectors object value multiplied by 5 cycles.”

countProfileEntry OBJECT-TYPE
SYNTAX countProfileEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION

“The count profile data over the bins collected for one of the PILOT05 detectors in the device over one of the saved cycles.”

INDEX {PILOT05DetectorNumber, cycleNumber}
countProfileEntry::= SEQUENCE{
cycleNumber INTEGER,
countProfileBins countProfileBins}

X.8.1 Count Profile Cycle Number
countCycleNumber OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
Status mandatory
DESCRIPTION

“Cycle number for objects in this row. The value shall not exceed 5.”

X.8.2 count Profile
countProfileBins OBJECT-TYPE
SYNTAX countProfileBins
ACCESS not-accessible
STATUS mandatory
DESCRIPTION

“The count profile data over the bins collected for one of the PILOT05 detectors in the device over 5 cycles.”

countProfileBins::= SEQUENCE{
countBin INTEGER}

X.8.2.1 Count Profile bin data
countBin OBJECT TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

“Average PILOT05detector count collected over 5 seconds in a scale of 0 to 100.”