

Portland State University

**PDXScholar**

---

Civil and Environmental Engineering Master's  
Project Reports

Civil and Environmental Engineering

---

Winter 2018

# Exploring High-Resolution Data Obtained from ASC/3 Controller of Intersection SE Division St & SE 122nd Ave, Portland, Oregon

Naowar Al-Abbas  
*Portland State University*

Follow this and additional works at: [https://pdxscholar.library.pdx.edu/cengin\\_gradprojects](https://pdxscholar.library.pdx.edu/cengin_gradprojects)



Part of the [Transportation Engineering Commons](#)

**Let us know how access to this document benefits you.**

---

## Recommended Citation

Al-Abbas, Naowar, "Exploring High-Resolution Data Obtained from ASC/3 Controller of Intersection SE Division St & SE 122nd Ave, Portland, Oregon" (2018). *Civil and Environmental Engineering Master's Project Reports*. 39.

<https://doi.org/10.15760/CCEMP.38>

This Project is brought to you for free and open access. It has been accepted for inclusion in Civil and Environmental Engineering Master's Project Reports by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: [pdxscholar@pdx.edu](mailto:pdxscholar@pdx.edu).

Exploring High-Resolution Data Obtained from ASC/3 Controller of Intersection SE Division St  
& SE 122nd Ave, Portland, Oregon.

BY

Naowar Al-Abbas

A research project report submitted in partial fulfillment  
of the requirement for the degree of

MASTER OF SCIENCE  
IN  
CIVIL AND ENVIRONMENTAL ENGINEERING

Project Advisor:  
Avinash Unnikrishnan

Portland State University  
©2017

## **ACKNOWLEDGMENTS**

I would like to express my deepest appreciation to my advisor Dr. Avinash Unnikrishnan for his guidance and support. I would also like to give special thanks to Dr. Kelly Clifton, Dr. Miguel Figliozzi, and Dr. Christopher Monsere for supporting me throughout my master program. I would also like to thank my family and friends for their support and encouragement during the whole journey.

## ABSTRACT

High-resolution data obtained from ASC/3 controllers can be used to create a variety of performance measures, helping engineers and professionals in monitoring intersection performance, identifying issues, and making proper improvements. These measures concern the intersection quality of the progression, capacity, and non-vehicular operations (e.g. pedestrian crossings). In this report, we explored high-resolution data obtained from the intersection of SE Division Street and 122<sup>nd</sup> Avenue. This intersection had some safety concerns as it is considered one of most dangerous intersections in Portland, Oregon. Thus, the first variable we examined from the data was pedestrian wait time when a cross request is made until pass indication is shown. It was found around 60% of pedestrians who crossed in an east-west direction experienced a delay of 30 seconds or more. Although there was more pedestrian demand from the south to west direction, 70% of these pedestrians received a 30 second delay or longer. Delay values determined from controller data in this report only refer to pedestrians who made a request/call to cross at a particular direction in each cycle. Additionally, an alternate means of calculating actual pedestrian delay was also addressed.

The other part of the analysis regarded vehicle operations and signal indications. Green time, cycle length, green-to-cycle ratio, and vehicle count per each cycle were all explored. Furthermore, the quality of the progression of coordinated phases of the through vehicular movement on SE division was quantitatively examined using the proportion of vehicles that had arrived during green indication and Arrival Type (AT), and visually, using the Purdue Coordination Diagram (PCD). The results showed that quality of progression was in good standard. Finally, errors found in the data and the accuracy of the results were also addressed.

## TABLE OF CONTENTS

<b>1.0</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	OBJECTIVES .....	2
1.2	STUDY INTERSECTION .....	2
<b>2.0</b>	<b>LITERATURE REVIEW .....</b>	<b>4</b>
<b>3.0</b>	<b>DATA 6</b>	
<b>4.0</b>	<b>APPROACH METHOD .....</b>	<b>8</b>
4.1	4.1 PEDESTRIAN-RELATED APPROACH METHOD .....	8
4.1.1	<i>Actual Pedestrian Delay Based on 1<sup>st</sup> Call</i> .....	8
4.1.1.1	Determination of 1 <sup>st</sup> Call-Based Pedestrian Delay from High-Resolution Data .....	10
4.1.1.2	Errors Associated with Determining 1 <sup>st</sup> -Call Based Delay .....	13
4.1.2	<i>Pedestrian Delay Using Highway Capacity Manual Equation</i> .....	14
4.1.3	<i>Time Intervals of Pedestrian Signal Indication</i> .....	15
4.1.3.1	Determination Time Intervals of Pedestrian Signal Indication from High-Resolution Data .....	15
4.2	VEHICULAR-RELATED APPROACH METHOD .....	17
4.2.1	<i>Background Cycle Length &amp; Effective Cycle length</i> .....	17
4.2.1.1	Determination of Background Cycle Length from High-Resolution Data .....	18
4.2.1.2	Determination Effective Cycle Length from High-Resolution Data .....	19
4.2.1.3	Data Error Associated with Determining Effective Cycle Length .....	20
4.2.2	<i>Green Time</i> .....	24
4.2.2.1	Effective Green Time Determination from High Resolution Data .....	25
4.2.2.2	Data Errors Associated with Determining Effective Green Time .....	26
4.2.3	<i>Vehicle Count</i> .....	28
4.2.3.1	Determination of Vehicle Count per Cycle from High Resolution Data .....	30
4.2.3.2	Data Errors Associated with Determination of Vehicle Count per Cycle .....	34
4.2.4	<i>Arrivals on Green &amp; Arrival Type</i> .....	35
4.2.4.1	Determination of Arrivals on Green & Arrival Type from High-Resolution Data .....	37
4.2.5	<i>Purdue Coordination Diagram (PCD)</i> .....	39
4.2.5.1	Determination of Purdue Coordination Diagram (PCD) From High Resolution Data .....	40
4.2.5.2	Data Errors Associated with Determination of Purdue Coordination Diagram (PCD) .....	42
<b>5.0</b>	<b>ANALYSIS AND DISCUSSION .....</b>	<b>43</b>
5.1	PEDESTRIANS DELAY .....	43
5.1.1	<i>Pedestrians Delay by 1st Called</i> .....	43
5.1.1.1	Pedestrian delay by 1st call compared to reality .....	53
5.1.2	<i>Number of Pedestrian Calls</i> .....	54
5.1.3	<i>Level of Service Based on Pedestrian Delay by 1<sup>st</sup> Call</i> .....	59
5.1.4	<i>Highway Capacity Manual (HCM) based Delay and Level of Service</i> .....	62
5.1.5	<i>Pedestrian Delay by Actuation</i> .....	67
5.2	VEHICULAR PERFORMANCE MEASURES .....	71
5.2.1	<i>Cycle Length and Plans of Day</i> .....	71
5.2.2	<i>Green Time &amp; Green/Cycle (g/C) Ratio</i> .....	74
5.2.3	<i>Vehicle Count</i> .....	77
5.2.4	<i>Proportion of Vehicles Arrived During Green Indication &amp; Arrival Type (AT)</i> .....	80
5.2.5	<i>Purdue Coordination Diagram</i> .....	83
5.3	ACCURACY OF RESULTS .....	85
<b>6.0</b>	<b>CONCLUSIONS .....</b>	<b>87</b>
<b>7.0</b>	<b>REFERENCES .....</b>	<b>89</b>
<b>8.0</b>	<b>APPENDIX .....</b>	<b>91</b>

## LIST OF TABLES

Table 3-1 : A partial high resolution event data .....	6
Table 4-1: An example of partial subset data for determining pedestrian delay based on 1st call .....	11
Table 4-2: An example of partial subset data after the selection of Pedestrian Calls (code 45) and beginning of pedestrian green time following calls (code 21 only when following code 45) .....	11
Table 4-3: An example of calculating actual pedestrian delay based on 1st call, by subtracting the timestamp of pedestrian calls from the timestamp of beginning pedestrian green time .....	12
Table 4-4: Highway Capacity Manual ranges of delay with associated Level of Service (Exhibit (18-9 HCM) (TRB, 2010).....	12
Table 4-5: An example of assigning Highway Capacity Manual Level of Service criteria to pedestrian delay values calculated based on 1st call .....	12
Table 4-6: An example of errors associated with determining delay based on 1st call .....	13
Table 4-7: An example of calculating average pedestrian delay based on HCM method .....	15
Table 4-8: An example of partial subset data for determining periods of time of pedestrian signal indication.....	16
Table 4-9: An example of partial data for determining Walking symbol period of time .....	17
Table 4-10: An example of partial data for determining Flash Don't Walk period of time.....	17
Table 4-11: An example of partial data for determining Solid Don't Walk period of time .....	17
Table 4-12: An example of partial data with a subset 'event' column for determining background cycle length.....	18
Table 4-13: An example of partial data with a subset 'event' and 'parameter' column for determining background cycle length.....	18
Table 4-14: An example of partial data for calculating background cycle length .....	19
Table 4-15: An example of partial data for determining effective cycle length, in a typical way (when data contains no errors) .....	20
Table 4-16: An example of partial data with proper sequence of indication of phases 2, 6, 4, and 8 (no errors).....	21
Table 4-17: An example of partial data with missing rows related to the sequence of signal indication of phases 2, 6, 4, and 8 .....	21
Table 4-18: An example of partial data of identifying beginning and end of cycles .....	23
Table 4-19: An example of partial data of calculating cycle length after identifying beginning and end of cycles, and assigning a unique number .....	23
Table 4-20: Conditions to locate potential cycles (errors in hindering calculating proper cycle length).....	23
Table 4-21: An example of partial data of final columns after running R scripts to calculate cycle length .....	24
Table 4-22: An example of subset partial data for determining green time for phase 2 .....	25
Table 4-23: An example of partial data for sorting columns and calculating green time (phase 2).....	26
Table 4-24: An example of an error faced when determining green time of phase 2 .....	27
Table 4-25: An example of assigning number of cycles to each row of partial subset data for calculating green time .....	27
Table 4-26: An example of partial data of final columns after running R scripts to effective green time and green to cycle ratio. ....	28
Table 4-27:- Number of loop detector numbers and associated channel number (PBOT) .....	32
Table 4-28: Shows how data needs to be subset when vehicles count each cycle, of a particular phase, is required at SE Division St. and 122nd Ave. ....	33
Table 4-29: An example of partial subset data after adjusting timestamp to determine vehicle count per each cycle of phase 2.....	34
Table 4-30: An example of partial data after counting number of vehicles for each cycle of phase 2 .....	34
Table 4-31: Platoon ratio with associated quality of progression criteria (TRB, 2010) .....	36
Table 4-32: An example of partial subset data after adjusting timestamp to determine vehicles arrived during green indication of phase 2.....	38
Table 4-33: An example of partial data after counting number of vehicles arrived during green indication of phase 2 .....	38
Table 4-34: An example of partial data of determining number of vehicles arrived during green, platoon ratio, arrival type and quality of progression .....	39
Table 4-35: An example of partial data of determining the three components of PCD. ....	42
Table 5-1: Average pedestrian delay based on 1st call, with its descriptive statistics for phase 6 .....	47
Table 5-2: Average pedestrian delay based on 1st call, with its descriptive statistics for phase 8 .....	47

Table 5-3: Average pedestrian delay based on 1st call, with its descriptive statistics of a weekend day vs 5 weekend days, from 3:00 to 23:59, for phase 6 .....	48
Table 5-4: Average pedestrian delay based on 1st call, with its descriptive statistics of a weekend day vs 5 weekend days, from 3:00 to 23:59, for phase 8 .....	48
Table 5-5: Average pedestrian delay based on 1st call, with its descriptive statistics of a weekend day vs 5 weekend days, from 3:00 to 23:59, for phase 8 .....	49
Table 5-6: Average pedestrian delay based on 1st call of five days of available data, phase 6 .....	52
Table 5-7: Average pedestrian delay based on 1st call of five days of available data, phase 8 .....	53
Table 5-8: Number of pedestrian calls of the 5 days of available data (phases 6 & 8) .....	54
Table 5-9: Number of pedestrian actuations of the 5 days of available data (phases 6 & 8) .....	55
Table 5-10: Number of pedestrian's calls and actuations of a weekend day vs 5 weekend days, from 3:00 to 23:59, for phase 6 .....	55
Table 5-11: number of pedestrian's calls and actuations of a weekend day vs 5 weekend days, from 3:00 to 23:59, for phase 8 .....	55
Table 5-12: Number of pedestrian calls per each hour of the day of five days of available data, phase 6 .....	58
Table 5-13: Number of pedestrian calls per each hour of the day of five days of available data, phase 6 .....	59

## LIST OF FIGURES

Figure 4-1: Sequence of phases at SE Division St. and 122nd Ave. (FHWA, 2008) .....	8
Figure 4-2: Phase numbers operating at SE Division St. and 122nd Ave. (only through and left turn movement) (PBOT).....	9
Figure 4-3: Pedestrian signal indications (Walking person, Flash Don't Walk with count down display, and Solid Don't Walk) (FHWA, 2009) .....	16
Figure 4-4: Effective green time related to actual (Day & Bullock, 2010).....	24
Figure 4-5: Loop detectors locations and numbers, and vehicular movement on each lane. ....	31
Figure 4-6: an example of PCD of phase 2.....	39
Figure 5-1: Actual pedestrian delay based on 1st call for phase 6.....	44
Figure 5-2: Actual pedestrian delay based on 1st call for phase 8.....	45
Figure 5-3: Average pedestrian delay based on 1st sorted ascendingly for phase 6 .....	46
Figure 5-4: Average pedestrian delay based on 1st call, sorted ascendingly, for phase 8 .....	46
Figure 5-5: Average pedestrian delay based on 1st call for each hour of the day (phase 6) .....	50
Figure 5-6: Average pedestrian delay based on 1st call for each hour of the day (phase 6) .....	50
Figure 5-7: Number of pedestrian calls per each hour of the day (phase 6) .....	57
Figure 5-8: Number of pedestrian calls per each hour of the day (phase 8) .....	57
Figure 5-9: Level of Service based on pedestrians 1st calls (phase 6).....	61
Figure 5-10: Level of Service based on pedestrians 1st calls of phase 6 vs. phase 8 .....	61
Figure 5-11: Level of Service based on based on estimated delay determined by HCM approach (phase 2) .....	63
Figure 5-12: Level of Service based on based on estimated delay determined by HCM approach (phase 6) .....	63
Figure 5-13: Level of Service based on based on estimated delay determined by HCM approach (phase 4) .....	64
Figure 5-14: Level of Service based on based on estimated delay determined by HCM approach (phase 8) .....	64
Figure 5-15: Time of pedestrian signal indication for phase 6 .....	66
Figure 5-16: Time of pedestrian signal indication for phase 8 .....	67
Figure 5-17: Level of Service of pedestrian delay by actuation vs. delay by 1st call (phase 2) .....	68
Figure 5-18: Level of Service of pedestrian delay by actuation vs. delay by 1st call (phase 4) .....	68
Figure 5-19: Level of Service of pedestrian delay by 1st call vs. HCM method based delay (phase2).....	70
Figure 5-20: Level of Service of pedestrian delay by actuation vs. HCM method based delay (phase 2).....	70
Figure 5-21: Effective cycle length .....	71
Figure 5-22: Background cycle length.....	73
Figure 5-23: Effective green time of all phases operating at SE Division and 122nd .....	75
Figure 5-24: Effective green time of phases 2 and 6 SE Division and 122nd .....	75
Figure 5-25: Effective green time of phases 1 and 5 SE Division and 122nd .....	76
Figure 5-26: Green to Cycle (g/C) ratio of all phases operating SE Division and 122nd .....	77
Figure 5-27: vehicle count for left turn phases and through movement of phases 2, 4, 6, and 8 at SE Division and 122nd.....	78
Figure 5-28: vehicle count for through lane movements of each hour of day. ....	79
Figure 5-29: vehicle count for left turn movements of each hour of day.....	79
Figure 5-30: Proportion of vehicles arrived during green indication of Phase 2 (eastbound) .....	81
Figure 5-31: Proportion of vehicles arrived during green indication of Phase 6 (westbound) .....	82
Figure 5-32: Arrival Type (AT) of phase 2 (eastbound).....	82
Figure 5-33: Arrival Type (AT) of phase 6 (westbound).....	83
Figure 5-34: Purdue Coordination Diagram (PCD) of through movement of phase 2 .....	84
Figure 5-35: Purdue Coordination Diagram (PCD) of through movement of phase 6 .....	85



## 1.0 INTRODUCTION

With the limited budgets assigned for new transportation facilities, along with a continuous increase in the number of vehicles flowing in a transportation network, it becomes very important to use novel ideas to optimize and make current transportation systems efficient. One of these approaches involves mining traffic data that more accurately reflects the actual situation, helping engineers and professionals to identify and tackle encountered problems (Wu & Lui, 2014).

A type of data that has helped transportation engineering is high-resolution data, obtained from ASC/3 controllers. The cost of acquiring this data is relatively low because it does not require deployment of extra hardware. This kind of data has been used extensively by Purdue University for a variety of purposes related to capacity, progression quality, and pedestrian operations in a single level intersection as well as along arterials.

In this research, high-resolution data will be examined at SE Division and 122<sup>nd</sup> regarding pedestrian and vehicular operations, with more focus on the former item. According to the Portland Bureau of Transportation, SE Division Street and 122<sup>nd</sup> Avenue is considered one of the most dangerous intersections in Portland. There were five fatalities at SE Division Street in 2016. Furthermore, research shows that pedestrians may take a risky action if the waiting time to cross an intersection exceeds 30 seconds (Mahmassani & Peeta, 1993). Therefore, in this research, pedestrian waiting time for pass indication was determined based on pedestrian calls/requests registered in the controller.

However, in reality, not all the pedestrians crossing SE Division and 122<sup>nd</sup> Avenue experienced the same amount of delay as the delay calculated based on the pedestrian calls. The reason for this is pedestrian calls were only registered once for every cycle at a particular direction, when the first person arrives and presses the pushbutton. To account for this, findings from this approach were compared to the Highway Capacity Manual method, and delay values were

calculated based on pedestrian actuation. Part of the pedestrian operation (pedestrian demand at intersection) was also examined.

In addition, the data provided was used to explore how the intersection operates during a 24-hour period. For this purpose, background and effective cycle length, green time parameter, and green-to-cycle ratio, and vehicle counts have been addressed along with the quality of the progression.

## **1.1 Objectives**

The objective of this research was to explore high-resolution data drawn from intersection of SE Division Street and 122<sup>nd</sup> Avenue and:

- Determine the delay experienced by pedestrians crossing the intersection in both east-west and north-south directions.
- Explore the variation in pedestrian delay in relation to the hour of the day.
- Determine the pedestrian demand at intersections.
- Explore the variation in pedestrian demand in relation to the hour of the day.
- Study the background and effective cycle length of the intersection during the day.
- Study the green time allocations among the phases and the green-to-cycle ratio.
- Evaluate the quality of coordinated phases quantitatively using the proportion of vehicles arriving during the green time and visually assess these moments using Purdue Coordination Diagram (PCD)

## **1.2 Study Intersection**

This study has been conducted on data obtained from the intersection of SE Division Street and 122<sup>nd</sup> located in Portland, Oregon. The intersection is about 7 miles away from downtown and

around 5 miles away from the city of Gresham's downtown (TriMet). It is a four-leg intersection where east-west vehicle movements occur on Division Street, and north-south movement on 122<sup>nd</sup>. All four approaches consist of four lanes: a protected left-turn, a two lane through, and a right turn movement. The location is supplied with loop detectors, specifically a stop bar and two sets of advanced detectors. However, in this report, only setback detectors will be used for the analysis.

## **2.0 LITERATURE REVIEW**

High-resolution data can be used to assess the signal operation at intersections. Furthermore, the cost of collecting high-resolution data is low compared to other techniques because data can be pulled out from the existing cabinets and does not require the deployment of special hardware. High-resolution data has been utilized for a variety of purposes at isolated intersections and along arterials containing multiple intersections. Concerning corridor performance measures, high-resolution data was used to assess the quality of the progression along Arterials using Arrival Type (AT) (Smaglik et al., 2007). Arrival Type (AT) is difficult to accurately measure using aggregated data, because it requires both vehicle counts during green along with timing of the phases. Furthermore, (Smaglik et al., 2007) explained some of the issues might negatively affect the accuracy of Arrival Type results when high-resolution data is used, such as occupying advanced detectors by queued vehicles during red indication or connecting detectors of two adjacent lanes into one channel instead of each having its own channel.

Related to Arrival Type and quality of progression, a Purdue Coordination Diagram (PCD) was used to visually evaluate the poor progression along arterials (Day et al., 2010; Day et al., 2012). In fact, PCD is not only a powerful tool to identify poor progression; it also helps professionals to infer the reasons behind poor progression. Moreover, event data helped develop optimization functions used to adjust the offset of coordinated arterials. As a result, the travel time along that corridor was reduced (Day et al., 2011).

High-resolution data was used along arterials to identify intersections with deficiencies in their capacity. Moreover, within these intersections, data helped finding unused capacities and solved by reallocating the green time or cycle length of that particular intersection (Day, Sturdevant, & Bullock, 2010).

With regard to measuring a single intersection's effectiveness, (Day et al., 2008) the following measures were used: equivalent hourly volume, green time periods, and volume over capacity ratio ( $v/c$ ) along with Arrival Type to quantitatively compare the coordinated phases of fully actuated versus non-actuated coordinated phases.

In addition to the aforementioned measures, high-resolution data was used to determine the delay with queue length of approaching vehicles on a cycle-by-cycle basis. The delay was calculated based on the departure and arrival rates of vehicles (Smaglik et al., 2007) (Sharma, Bullock, & Bonneson, 2007).

Furthermore, at railroad-preempted intersections, the event-based data was used to ensure that there was enough time (enough track clearance green) for the queued vehicles to clear out after the railroad gate was closed (Brennan, Day, Sturdevant, Raamot, & Bullock, 2010).

In terms of pedestrian operation, high-resolution data was helpful in determining the actual pedestrian delay (by actuation). Furthermore, pedestrians' safety was negatively affected by right turn vehicular movement and permissive left turn movement. High resolution data was used to determine traffic volume, in terms of vehicles per hour that conflicted with pedestrian movement, and helped in assessment (Hubbard, Bullock, & Day, 2008).

### 3.0 DATA

Data available for this report was acquired from an ASC/3 Controller at SE Division Street and 122<sup>nd</sup> Avenue. A controller is one of the components of a signalized intersection that manages the signal display based on the vehicular or pedestrian activity at the intersection. Often, controllers work in relation with other intersections (NCHRP, 2012). Controller data consists of three columns: “Timestamp”, “Event Type”, and “Parameter.” The following table is an example of this data:

Timestamp	Event Type	Parameter
5/19/2016 10:30:17	1	2
5/19/2016 10:30:17	9	1
5/19/2016 10:30:17	12	1
5/19/2016 10:30:17	21	2
5/19/2016 10:30:17	82	24
5/19/2016 10:30:17	82	29
5/19/2016 10:30:17	82	18
5/19/2016 10:30:18	81	24
5/19/2016 10:30:18	81	18
5/19/2016 10:30:18	81	29
5/19/2016 10:30:19	82	15

*Table 3-1 : A partial high resolution event data*

“Timestamp” contains the date and time of activities taking place at the intersection. These activities are related to vehicle detection or signal indications from pedestrians, etc. The timestamp of this data is accurate to 0.1 of a sec, thus it is called High-resolution data (Day et al., 2014).

The “Event Type” column includes activates happening at the intersection, which are encoded as numbers. For example, the beginning of the green phase, is coded as “1,” or, when vehicle detection occurs, this is encoded as 82. The Indiana Traffic Signal High-Resolution Data Logger Enumerations provides all code description in the “Event Type” column.

Finally, “Parameter” includes the phase number if the event in the second column was related to the phase indication, or channel number if it was related to vehicle detection. For example, reading the three rows at once, the first row in the table above demonstrates that phase number 2 began its green phase at “10:30:17” on May 19<sup>th</sup>, 2016.

Data provided in this report includes 15-minute CSV files. In each file, the three columns begin from row 9. Therefore, above rows had to be removed and 15-minute files were merged to construct 24 hour’s worth of data. R studio was used in this task. A complete set of the data is available for the weekdays between May 23–26 and 1st of June. Partial data for the weekend is available on May 22, 2016, from 03:00 to 23:59.

## 4.0 APPROACH METHOD

### 4.1 Pedestrian-Related Approach Method

#### 4.1.1 Actual Pedestrian Delay Based on 1<sup>st</sup> Call

High-resolution event based data (output of the controller data) is comprised of three columns: Timestamp, parameter (phase), and event. Each row of data tells us that a particular event (for example, the beginning of the green time) of a particular phase at a certain timestamp has taken place. The high-resolution event based data obtained from the intersection controller can be utilized to calculate the pedestrian actual delay based on the first call registered in the controller. The pedestrian delay is defined as the waiting time when a pedestrian registers a call/request (by pressing the pushbutton) to cross the intersection at particular direction prior to the pedestrian pass indication being on. In the study intersection, SE Division & 122nd, there are four pedestrian phases (2, 6, 4, and 8). The movement of the pedestrian phases happen with the adjacent vehicular through movement, see figures 4-1 and 4-2.

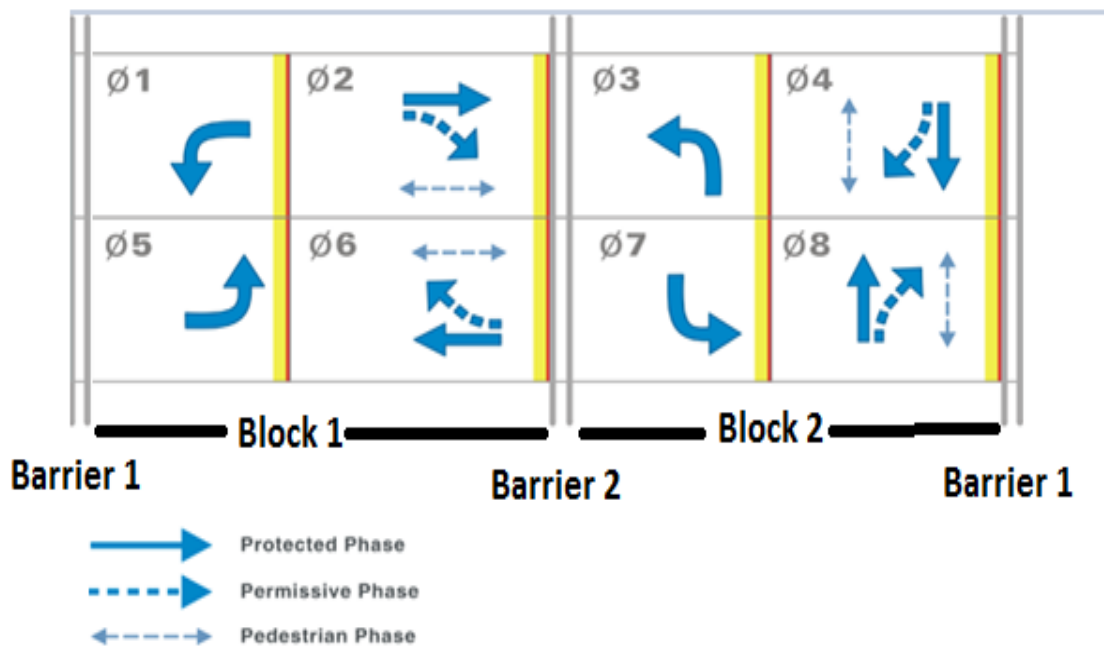
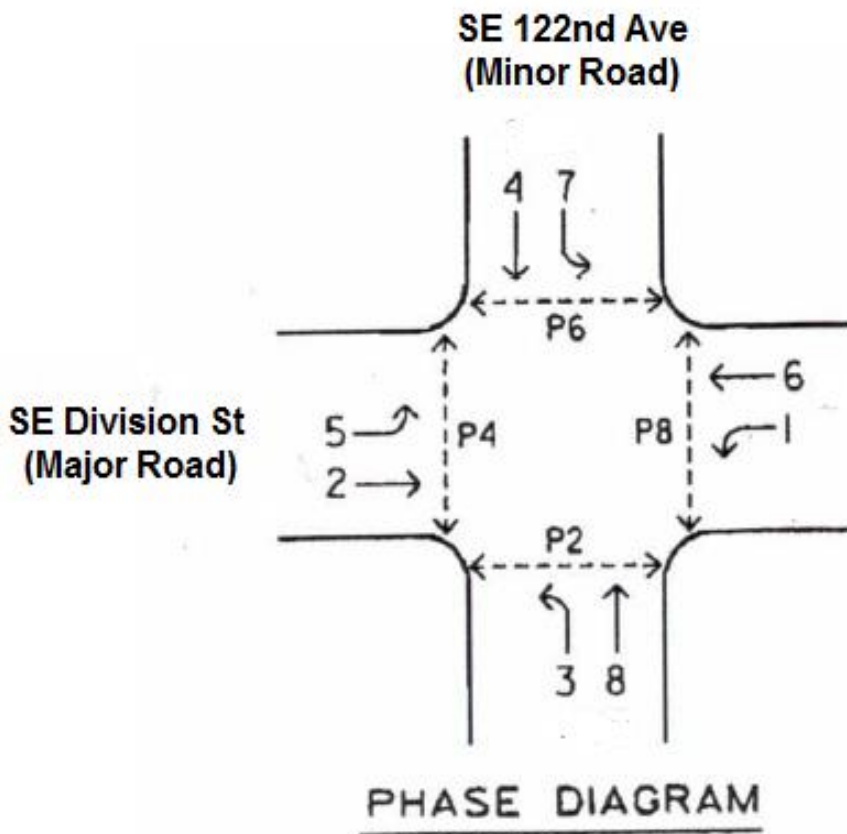


Figure 4-1: Sequence of phases at SE Division St. and 122nd Ave. (FHWA, 2008)



At the study intersection, the phases operate as shown in figure 4-1, from left to right. First, the east/west pedestrian movements, represented by phase 2 and phase 6, take place on SE Division Street. Once this is over, the north-south pedestrian movement occurs on 122<sup>nd</sup>. In other words, pedestrian movement in phases 2 and 6 for both directions (east and west) occur at the same time, and pedestrian movement in phases 4 and 8 for both directions (north and south) occur at the same time. The pedestrian movements of all the phases are conflicted by the adjacent permissive right turn vehicular movement.



*Figure 4-2: Phase numbers operating at SE Division St. and 122nd Ave. (only through and left turn movement) (PBOT)*

#### ***4.1.1.1 Determination of 1<sup>st</sup> Call-Based Pedestrian Delay from High-Resolution Data***

To calculate actual pedestrian delay based on the first call of a particular phase (2 or 6, or 4 or 8) and at particular time of the day, using the controller data, the following steps should be taken:-

1. All three columns of data should be subset in the following manner:
  - The “Event Type” column should be subset when it is equal to 45 and 21. According to the traffic signal high-resolution event based data logger, number 45 in the “Event Type” column of the data represents “Pedestrian Call Registered,” Every time a pedestrian registers a call to cross the intersection, it will be coded as number 45 in the controller. Code 21 represents “Pedestrian Begin Walk” (Sturdevant et al., 2012). So, code 21 sets the controller whenever the pedestrian “Walking Person” indication is active.
  - The “Parameter” column should be subset to the particular phase (2 or 4 or 6 or 8)
  - The “Timestamp” column should be subset to include the time of the day when the actual pedestrian delay requires calculation. See table 4-1
2. From the subset data in the previous step, code 45 was selected and only code 21 was preceded by code 45 was identified. To translate this, the selection should only include pedestrian call events registered followed directly (the very next row of pedestrian call data) by an event when the pedestrian green time begins. It is obvious from the table above that there are more triggering’s of event 21 than 45. This means that the pedestrian green phase (number 2) is activated without the need of pressing the pedestrian push button (i.e., without the need of registering a call). Therefore, 21s that are not preceded by 45s should be eliminated because they cannot be used to calculate delay. Thus, table 4-2 shows the data after the selection of code 45s and only 21s preceded by 45s:

Timestamp	Event Type	Parameter
14:48:13	45	2
14:48:51	21	2
14:50:38	21	2
14:52:31	21	2
14:54:13	45	2
14:54:21	21	2
14:54:52	45	2
14:56:11	21	2
14:57:43	45	2
14:58:01	21	2
14:58:57	45	2
14:59:46	21	2
15:00:58	45	2

**Table 4-1: An example of partial subset data for determining pedestrian delay based on 1st call**

Timestamp	Event Type	Parameter	Timestamp 1	Event Type 1	Parameter 1
14:48:13	45	2	14:48:51	21	2
14:54:13	45	2	14:54:21	21	2
14:54:52	45	2	14:56:11	21	2
14:57:43	45	2	14:58:01	21	2
14:58:57	45	2	14:59:46	21	2

**Table 4-2: An example of partial subset data after the selection of Pedestrian Calls (code 45) and beginning of pedestrian green time following calls (code 21 only when following code 45)**

3) Calculating the actual pedestrian delay of a particular phase can be achieved by subtracting the timestamp when pedestrian indication of a particular phase is active from the timestamps when the pedestrian call was registered. In other words, subtracting the timestamp when code 21 occurs from the timestamp when code 45 occurs for a particular phase will determine the pedestrian delay based on the first call. See the table 4-3 below:

Timestamp	Event Type	Parameter	Timestamp 1	Event Type1	Parameter 1	Delay (secs)
14:48:13	45	2	14:48:51	21	2	38
14:54:13	45	2	14:54:21	21	2	8
14:54:52	45	2	14:56:11	21	2	79
14:57:43	45	2	14:58:01	21	2	18
14:58:57	45	2	14:59:46	21	2	49

**Table 4-3: An example of calculating actual pedestrian delay based on 1st call, by subtracting the timestamp of pedestrian calls from the timestamp of beginning pedestrian green time**

A level of service (LOS) criteria can be assigned to actual pedestrian delay calculated in the table above (Hubbard, Bullock, & Day, 2008). According to the Highway Capacity Manual (HCM), if a pedestrian experienced a delay at a signalized intersection within a particular range of time, a certain LOS can be given to it. See table 4-4

LOS	Pedestrian Delay (sec/Pedestrian)
A	< 10
B	≥ 10 – 20
C	> 20 - 30
D	> 30 - 40
E	> 40 - 60
F	> 60

**Table 4-4: Highway Capacity Manual ranges of delay with associated Level of Service (Exhibit (18-9 HCM) (TRB, 2010)**

If the Highway capacity manual LOS criteria is adopted on the pedestrian delay calculated above, LOS will be assigned as follows:

Delay (secs)	LOS
38	LOS (D)
8	LOS (A)
79	LOS (F)
18	LOS (B)
49	LOS (E)

**Table 4-5: An example of assigning Highway Capacity Manual Level of Service criteria to pedestrian delay values calculated based on 1st call**

#### 4.1.1.2 Errors Associated with Determining 1<sup>st</sup>-Call Based Delay

After subsetting the data for the purpose of determining pedestrian delay based on the first call, as earlier illustrated, errors can be found in rows where an “Event Type” column is equal to 45 followed directly by a row where its event column is equal to 45 as well. The following table 4-6 demonstrates an error after subsetting data associated with phase 2. The error is clearly demonstrated in the last row and tagged with cross mark. Such rows have been removed from data for more accurate results. The same procedure was adopted with other pedestrian performance measures in this report, as errors associated with them are very similar to those with pedestrian delay by the first call.

Timestamp	Event Type	Parameter	Timestamp 1	Event Type 1	Parameter 1	Tag
5:40:41	45	2	5:41:58	21	2	✓
5:45:39	45	2	5:46:33	21	2	✓
5:48:10	45	2	5:48:59	21	2	✓
5:50:48	45	2	5:51:21	21	2	✓
5:53:11	45	2	5:54:08	21	2	✓
5:59:57	45	2	5:59:58	45	2	✗

*Table 4-6: An example of errors associated with determining delay based on 1st call*

Moreover, regarding pedestrian delay by first call or by actuation, an R script was placed to remove any value exceeding 140 seconds, although such cases were rare when determining pedestrian delay at the intersection of SE Division and 122<sup>nd</sup> avenue. Values beyond 140 seconds appear unreasonable, as the maximum cycle length values never reached a value above 140 seconds. Thus, such values were considered as errors, and have been tagged and removed accordingly.

#### 4.1.2 *Pedestrian Delay Using Highway Capacity Manual Equation*

The Highway Capacity Manual provides two formulas to calculate the average pedestrian delay at a signalized intersection. One formula is dependent on two variables: cycle length and pedestrian effective green time. The other is dependent on the number of pedestrians crossing the crosswalk, the number of conflicting vehicles. In this study, the controller data can be used to calculate pedestrian delay from the first formula (the one that is dependent on cycle length and pedestrian “green” time). On the other hand, controller data cannot be used to determine pedestrian delay using the second HCM formula because it requires a camera to count the number of pedestrians crossing the intersection and the number of vehicles turning right and thus conflicting with pedestrian crossing is unknown. This is due to lack of a right turn detector in our intersection. However, HCM pedestrian delay can be calculated as below (equation 18-5 in HCM):

$$dp = \frac{0.5 (C - g)^2}{C}$$

dp = Average Pedestrian Delay (s/p)

g=Effective Pedestrian Green Time (Walking indication interval + 4 seconds of pedestrian clearance time)

C = Cycle Length

The table below is a sample of our data for calculating pedestrian delay using the above formula. The table also shows Level of Service (LOS) based on HCM pedestrian delay criteria, as illustrated in table 4-5.

Walking Interval (sec)	4 sec of Clearance Time	Cycle Length (sec)	Pedestrian Delay (sec/person)	LOS
7	4	80.5	30.00	LOS (D)
7	4	70.3	25.01	LOS (C)
7	4	56.8	18.47	LOS (B)

*Table 4-7: An example of calculating average pedestrian delay based on HCM method*

#### 4.1.3 *Time Intervals of Pedestrian Signal Indication*

The pedestrian phase consists of two intervals: the Walking interval and the Pedestrian Clearance interval. Regarding the Walking interval, which is indicated by appearance of the “Walking person” symbol (see figure 4-3), time must be sufficient enough to allow pedestrians to observe the beginning of the phase while also entering the crosswalk (FHWA, 2009). On the other hand, the pedestrian clearance interval itself contains two periods of time: Flash Don’t Walk and Solid Don’t Walk. The pedestrian clearance time must be sufficient to at least allow pedestrians to complete crossing the intersection if they already had entered it (FHWA, 2009).

##### 4.1.3.1 *Determination Time Intervals of Pedestrian Signal Indication from High-Resolution Data*

The controller data can be used to observe and visualize period of time for pedestrian phase components (Walking person symbol, Flash Don’t Walk, and Solid Don’t Walk) along the day. To approach that, the following steps should be followed:

1. The three controller data columns should be subset as follows:
  - The “Event Type” should be subset when it is equal to codes 21, 22, 23, and 10. According to the Indiana Traffic Signal High-Resolution Data Logger, code 21 refers to “Pedestrian Begin Walk” (beginning Walking interval); code 22 refers to “Pedestrian Begin Clearance” (beginning of Flash Don’t Walk interval); code 23 refers to “Pedestrian Begin Solid Don’t

Walk” (beginning of the Solid Don’t Walk interval), and code 10 refers to beginning of the red of through and right turn phases adjacent to the pedestrian Sturdevant et al., 2012).

- The “Parameter” or phase column should subset to include the required phase (2 or 4, or 6 or 8 as in our data).
- The “Timestamp” column is subset to include the required period of time when the pedestrian phase component needs to be observed/visualized.



**Figure 4-3: Pedestrian signal indications (Walking person, Flash Don’t Walk with count down display, and Solid Don’t Walk) (FHWA, 2009)**

The table below shows the data after the subset:

Timestamp	Event Type	Parameter
13:09:51	21	2
13:10:10	22	2
13:10:27	23	2
13:10:31	10	2
13:11:41	21	2
13:12:00	22	2
13:12:17	23	2
13:12:21	10	2

**Table 4-8: An example of partial subset data for determining periods of time of pedestrian signal indication**



2. To calculate intervals of Walk, Flash Don't Walk, and Solid Don't Walk indications, we must subtract the timestamps of code 22 from code 21, code 23 from code 22, and code 10 from code 23 respectively. See three tables below:

Timestamp	Event Type	Timestamp	Event Type	Phase	Walk (min:sec)
13:09:51	21	13:10:10	22	2	00:19
13:11:41	21	13:12:00	22	2	00:19

*Table 4-9: An example of partial data for determining Walking symbol period of time*

Timestamp	Event Type	Timestamp	Event Type	Phase	FDW (min:sec)
13:10:10	22	13:10:27	23	2	00:17
13:12:00	22	13:12:17	23	2	00:17

*Table 4-10: An example of partial data for determining Flash Don't Walk period of time*

Timestamp	Event Type	Timestamp	Event Type	Phase	SDW (min:sec)
13:10:27	23	13:10:31	10	2	00:04
13:12:17	23	13:12:21	10	2	00:04

*Table 4-11: An example of partial data for determining Solid Don't Walk period of time*

## 4.2 Vehicular-Related Approach Method

### 4.2.1 Background Cycle Length & Effective Cycle length

In general, the cycle length can be defined as the period of time required to serve all the phases at the intersection (FHWA, 2008). In other words, it is the period of time when a phase completes a set of indications (green, yellow and red interval).

Background cycle length is the cycle length previously programmed into the controller. So, when the controller data is visualized, the coordinated phases will be identified and the programmed cycle length will be determined as well throughout the day. In order to visualize the

background cycle length using the high-resolution event based data, the timestamp of the yield point is subtracted. The yield point can be defined as the earliest point at which the coordinated phase ends (FHWA, 2008).

**4.2.1.1 Determination of Background Cycle Length from High-Resolution Data**

To illustrate the calculation of the background cycle length the following steps should have taken place:

- The event column should be subset to the number 151. According to the Indiana Traffic Signal High-Resolution Data Logger, code 151 is “Coordinated phase yield point.”
- From table 4-12, it is known that the coordinated phases are 2 and 6. A further subset should take the place of column “Parameter” by selecting either phase 2 or phase 6 (either one will yield the same results). If phase 2 was selected, the results will appear as in table 4-13:

Timestamp	Event Type	Parameter
10:30:40	151	2
10:30:40	151	6
10:32:30	151	2
10:32:30	151	6
10:34:20	151	2
10:34:20	151	6
10:36:10	151	2

*Table 4-12: An example of partial data with a subset ‘event’ column for determining background cycle length*

Timestamp	Event Type	Parameter
10:30:40	151	2
10:32:30	151	2
10:34:20	151	2
10:36:10	151	2

*Table 4-13: An example of partial data with a subset ‘event’ and ‘parameter’ column for determining background cycle length*

- Column “Timestamp” should be subset to include the required period of time, normally 24 hours.
- The difference in timestamps between the current and the following row will result in background cycle length.

Timestamp	Event Type	Parameter (phase)	Background Cycle Length (sec)
10:30:40	151	2	
10:32:30	151	2	110
10:34:20	151	2	110
10:36:10	151	2	110

*Table 4-14: An example of partial data for calculating background cycle length*

#### **4.2.1.2 Determination Effective Cycle Length from High-Resolution Data**

Effective cycle length will be slightly different than background cycle length due to the phase actuation. Further, obtaining the effective cycle length using the controller data is different than the way of determining the background cycle length.

Determining the effective cycle, and subsequently identifying the beginning of the cycle is a very important aspect for the further analysis since the analysis and calculations take place on the basis of cycle-by-cycle performance measures. Moreover, assigning a unique number to each cycle is necessary when filtering data in following analysis.

In order to calculate the effective cycle length from the controller data, the first step is to find the barrier between block one (phases 1, 2, 5 and 6) and block two (phases 3, 4, 7, and 8) in the data, see figure 4-1. In the Standard ring-and-barrier structure, which the intersection of SE Division St. & 122<sup>nd</sup> Ave follows, the barrier can be either: 1<sup>st</sup> end of phases 2 and 6, or the beginning of phases 3 and/or 7, or the beginning of phases 4 and/or 8 in case any of the phases 3

or 7 was/were omitted (barrier2) see figure 4-1. 2<sup>nd</sup>, end of phases 4 and 8, or the beginning of phases 1 and/or 3, or beginning of phases 2 and/or 6 in case any of the phases 1 or 3 were emitted (barrier 1). Because the intersection operates in coordination at certain times of the day, and phases 2 and 6 are the coordinated phases, using barrier 2 as the basis to calculate will be more accurate. If the first point above was picked, and if the data was clear (i.e. data with no errors), the cycle length can be calculated as follows:

- The “Event Type” column should be subset to the number 10. According to the Indiana Traffic Signal High-Resolution Data Logger, 1 is “Phase Begin Red Clearance”.
- The “Parameter” column should be subset to 2 or 6 (the coordinated phases during certain times of the day)
- The required period should be selected from the “Timestamp” column, normally 24 hours.

Timestamp	Event Type	Parameter	Effective Cycle Length (sec)
14:52:31	10	2	
14:54:21	10	2	110
14:56:11	10	2	110
14:58:01	10	2	110
14:59:46	10	2	105
15:01:41	10	2	115
15:03:31	10	2	110

*Table 4-15: An example of partial data for determining effective cycle length, in a typical way (when data contains no errors)*

#### **4.2.1.3 Data Error Associated with Determining Effective Cycle Length**

As mentioned earlier, the available data for this report contains some errors. There are rows that should exist in the data, yet they are not present for unknown reasons. An example of these errors is that it is not uncommon to find a particular phase started its green, which is represented

by code 1. However, the data does not show the row containing the end of that particular phase, which is represented by code 8 (beginning of yellow clearance) and/or by code 10 (beginning of red clearance). See tables 4-16 and 4-17 below:

Block number	Timestamp	Event Type	Parameter	Description
Block #1	2:13:24	1	2	Phase #2 starts green
	2:13:24	1	6	Phase #6 starts green
	2:13:48	8	2	Phase #2 starts yellow
	2:13:48	8	6	Phase #6 starts yellow
	2:13:52	10	2	Phase #2 starts red
	2:13:52	10	6	Phase #6 starts red
Block #2	2:13:53	1	4	Phase #4 starts green
	2:13:53	1	8	Phase #8 starts green
	2:14:12	8	4	Phase #4 starts yellow
	2:14:12	8	8	Phase #8 starts yellow
	2:14:16	10	4	Phase #4 starts red
	2:14:16	10	8	Phase #8 starts red

*Table 4-16: An example of partial data with proper sequence of indication of phases 2, 6, 4, and 8 (no errors)*

Block number	Timestamp	Event Type	Parameter	Description
Block #2	4:14:48	1	4	Phase #4 starts green
	4:14:48	1	7	Phase #7 starts green
	4:14:53	8	7	Phase #7 starts yellow
	4:14:56	10	7	Phase #7 starts red
	4:14:56	1	8	Phase #8 starts green
Block #1				Missing data
				Missing data
	4:15:35	8	2	Phase #2 starts yellow
	4:15:35	8	6	Phase #6 starts yellow
	4:15:39	10	2	Phase #8 starts red

*Table 4-17: An example of partial data with missing rows related to the sequence of signal indication of phases 2, 6, 4, and 8*

The first table shows the proper sequence of signal indications, which begins with green, then yellow, then red, of the through phases. On the other hand, the second table shows that there

are missing rows, which are the end of phases 4 and 8 and the beginning of phases 2 and 6. Therefore, the effect of these errors on our results should be minimized, avoided, and/or removed.

Because such errors are present in the available data for this report, a special procedure was performed to determine the beginning end of cycles, assigning a unique number to each cycle, and finally, calculating cycle length. The special procedure attempts to find the last phase events between block 1 and block 2. An R code was used for this task to find a row when the “Parameter” column is equal to 10 and “Event Type” equals to 2 or 6 – end of block 1, followed directly by a row when the “Parameter” is equal to 1 and the “Event Type” is equal to 3 or 7, or 4 or 8 – beginning of block 2.

The R code worked to find rows whenever the “Parameter” column equaled 2 and 6, and the “Event Type” column equaled to 10, followed by a row when the “Parameter” column equaled 3 or 7, or 4 or 8, and the “Event Type” column equaled to 1. Thus, whenever this condition is met, the beginning (boc) and end of cycle (eoc) will be identified, and a unique number will be assigned, see tables 4-18 and 4-19.

However, if other event activities related to the phase status took place at the barrier number 2 between block 1 and block 2, these will be tagged as a potential cycle, numbered, and then removed. For example, if a row where the “Parameter” equals to 2 and 6 and the “Event Type” equals 10 is followed by a row where the “Parameter” equals 3 or 7, or 4 or 8, but an “Event Type” equals 8 (instead of 1), it will be tagged as an error and removed. Table 4-20, shows the conditions that whenever found, will be labeled as a potential cycle length, numbered, and removed.

Timestamp	Event Type	Parameter	Timestamp	Event Type	Parameter	boc & eoc
0:00:24	8	2	0:00:24	8	6	
0:00:24	8	6	0:00:28	10	2	
0:00:28	10	2	0:00:28	10	6	
0:00:28	10	6	0:00:29	1	4	*
0:00:29	1	4	0:00:29	1	8	
0:00:29	1	8	0:00:44	8	4	
0:00:44	8	4	0:00:44	8	8	
0:00:44	8	8	0:00:48	10	4	
0:00:48	10	4	0:00:48	10	8	
0:00:48	10	8	0:00:49	1	2	
0:00:49	1	2	0:00:49	1	5	
0:00:49	1	5	0:00:57	8	5	
0:00:57	8	5	0:01:00	10	5	
0:01:00	10	5	0:01:00	1	6	
0:01:00	1	6	0:01:24	8	2	
0:01:24	8	2	0:01:24	8	6	
0:01:24	8	6	0:01:27	10	2	
0:01:27	10	2	0:01:27	10	6	
0:01:27	10	6	0:01:28	1	3	*
0:01:28	1	3	0:01:28	1	8	
0:01:28	1	8	0:01:35	8	3	

**Table 4-18: An example of partial data of identifying beginning and end of cycles**

Beginning of cycle (boc)	End of cycle	Cycle length (sec)	Number of cycle
0:00:28	0:01:28	60	1

**Table 4-19: An example of partial data of calculating cycle length after identifying beginning and end of cycles, and assigning a unique number**

IF			Followed by		
Phase =	Event Type =	Row #	Phase =	Event Type =	Row #
1 or 2 or 5 or 6	1	x	3 or 4 or 7 or 8	1	x+1
1 or 2 or 5 or 6	1	x	3 or 4 or 7 or 8	8	x+1
1 or 2 or 5 or 6	1	x	3 or 4 or 7 or 8	10	x+1
1 or 2 or 5 or 6	8	x	3 or 4 or 7 or 8	1	x+1
1 or 2 or 5 or 6	8	x	3 or 4 or 7 or 8	8	x+1
1 or 2 or 5 or 6	8	x	3 or 4 or 7 or 8	10	x+1
1 or 2 or 5 or 6	10	x	3 or 4 or 7 or 8	8	x+1
1 or 2 or 5 or 6	10	x	3 or 4 or 7 or 8	10	x+1

**Table 4-20: Conditions to locate potential cycles (errors in hindering calculating proper cycle length)**

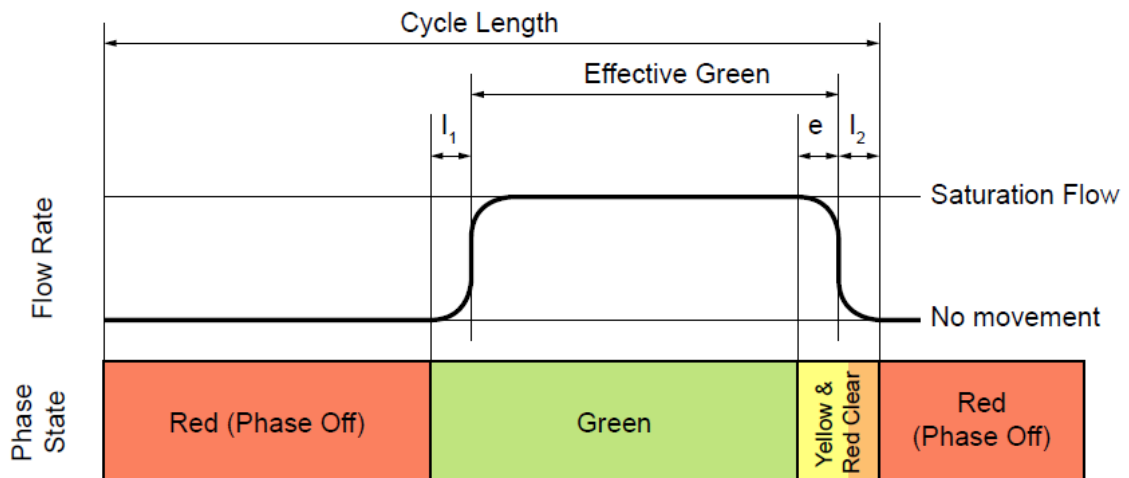
After running the R scripts, the final cycle length data will be as follows:-

Beginning of Cycle	Cycle Number	Cycle Length (sec)
7:22:50	473	110
7:26:41	474	110
7:28:31	475	110
7:30:21	476	109.8
7:32:11	477	108.8

*Table 4-21: An example of partial data of final columns after running R scripts to calculate cycle length*

#### 4.2.2 Green Time

The actual green time is the period of time between the beginning and the end of the green indication of a particular phase. However, when the green phase starts, vehicles do not enter the intersection immediately; they experience a few seconds of delay until they move. This lag in vehicle movement at the beginning of the green phase is called *start up lost time* ( $l_1$ ), (TRB, 2010). Furthermore, vehicles normally do not stop as soon as the green phase ends; they tend to use some of the change and clearance interval. The time used by the vehicles to cross the intersection, when the green phase ends and change interval starts, is called the *extension of green time* ( $e$ ), (TRB, 2010). See the following figure (4-4):



*Figure 4-4: Effective green time related to actual (Day & Bullock, 2010).*



The effective green time is used in a further analysis and is calculated using the following HCM equation:

$$g_i = G_i + I_1 + e$$

Where,

$g_i$  = effective green time (sec)

$G_i$  = Actual green time

$I_1$  = Startup lost time

$e$  = Extension of the green time

#### 4.2.2.1 *Effective Green Time Determination from High Resolution Data*

According to the HCM (startup lost time and extension of the green time can be taken as 2 seconds). The effective green time can be obtained from the controller data as follows:

- The “Event Type” column is subset to code 1 (Phase Begin Green) and to code 8 (Sturdevant et al., 2012).
- The “Parameter” column is subset to the required phase, the phase required to calculate its green time
- The required period of time is selected from the “Timestamp” column, normally one day

Timestamp	Event Type	Parameter
0:00:49	1	2
0:01:24	8	2
0:01:53	1	2
0:02:20	8	2
0:02:52	1	2
0:03:18	8	2

*Table 4-22: An example of subset partial data for determining green time for phase 2*

- Create a data of event with code 1 only and its timestamp and merge it with the data of event 8 only and its timestamp. Then, add startup lost time (2 sec) to the timestamp of the

actual beginning of the green time and add extension of the green time (2 sec) to timestamp of the actual end of the green time.

$$tg = tG + l_1$$

$$ty = tY + e$$

tg = Time stamp of effective green time

tG = Time stamp of actual green time

ty = Timestamp of effective yellow clearance (effective end of the green phase)

tY = Timestamp of effective yellow clearance (effective end of the green phase)

tG	l <sub>1</sub> (sec)	tg	Event Type	tY	e (sec)	ty	Event Type	Phase	Green Time (sec)
0:00:49	2	0:00:51	1	0:01:24	2	0:01:26	8	2	0:00:35
0:01:53	2	0:01:55	1	0:02:20	2	0:02:22	8	2	0:00:27
0:02:52	2	0:02:54	1	0:03:18	2	0:03:20	8	2	0:00:26

*Table 4-23: An example of partial data for sorting columns and calculating green time (phase 2)*

#### **4.2.2.2 Data Errors Associated with Determining Effective Green Time**

The above method is a simple calculation of the green time using controller data. However, as mentioned, the data provided contains some errors. Within the framework of calculating the green time, an example of errors can be seen in table 4-24. The error was tagged table 4-24 because the beginning of the green event (Event Type = 1) must be followed by its termination (Event Type = 8), yet the latter row was missing.

Timestamp	Event Type	phase	Description	
3:42:57	1	2	Phase #2 starts green	
3:43:21	8	2	Phase #2 ends green	
3:43:40	1	2	Phase #2 starts green	
3:44:04	8	2	Phase #2 ends green	
3:44:40	1	2	Phase #2 starts green	
3:45:31	1	2	Phase #2 starts green	Error
3:45:55	8	2	Phase #2 ends green	

*Table 4-24: An example of an error faced when determining green time of phase 2*

An R script was used to tag these errors and to exclude them from calculating the green time. The R script performs the following:

- Because the “Timestamp” at the beginning and the end of the each cycle has been previously determined, the R script would allocate each row of the subset data (selected to calculate the green time) to its cycle. Thus, each beginning and end of the green phase belongs to a unique cycle, see the following table 4-25.

Timestamp	Event Type	Parameter	Number of cycle
0:01:53	1	2	2
0:02:20	8	2	2
0:02:52	1	2	3
0:03:18	8	2	3
0:03:52	1	2	4
0:04:28	8	2	4

*Table 4-25: An example of assigning number of cycles to each row of partial subset data for calculating green time*

- The syntax set two conditions for selecting data: First, select two rows if the first “Event Type” column equals the first and second row “Event Type” equals 8. Secondly, the two selected rows must belong to the same cycle (i.e. number of the cycle is equal). Rows that

do not meet this condition will be tagged and excluded. Finally, after running the R script, the final table can be obtained:

Beginning of Cycle	Number of Cycle	Cycle Length	Phase number	Beginning of Effective	Effective Green time	Green/Cycle ratio
0:01:53	2	59	2	0:01:55	28	0.47
0:02:52	3	61	2	0:02:54	26	0.43
0:03:52	4	56	2	0:03:54	36	0.65
0:04:48	5	57	2	0:05:00	25	0.44
0:05:45	6	53	2	0:05:56	24	0.45

*Table 4-26: An example of partial data of final columns after running R scripts to effective green time and green to cycle ratio.*

Furthermore, by dividing the effective green time over the cycle length, the green to cycle ratio (g/C) is obtained. This value indicates how much green time out of the entire cycle length is shared by a particular phase. Also, this value is important in further analysis of the intersection.

#### 4.2.3 *Vehicle Count*

Vehicle count at the intersection is a very important aspect of the intersection's performance measures. When the information regarding the vehicle count is combined with the timing data of the phases, utilization of each phase can be determined (Sturdevant et al., 2012). Moreover, knowing the number of vehicles arriving on the green or red indication is important to find out how well the coordination is functioning.

However, vehicle count is implemented by detecting the presence of vehicles passing over set back detectors provided at each lane of the intersection, except for the right turn lane movement. Thus, this report lack information regarding right turn vehicle count.

Conceptually, the number of vehicles should be counted and then allocated to the cycle when these vehicles have been served. For a particular phase, some of the vehicles arrive on the

red and others arrive on the green, but both will be served when the phase is green. Therefore, these vehicles should be counted (both on red and green) and then mapped to the cycle when they are served. In other words, the counting period for each cycle of a particular phase occurs from the end of the previous effective green to the end of the current one.

The intersection of SE Division St. and 122nd avenue contains two sets of advanced detectors provided for through and left turn movements (no right turn detectors). Regarding the through lanes, the first set is 86 ft. away from the stop bar and the second set is 185 ft. away. On the other hand, the first set of the left turn detectors is 36 ft. away from the stop bar and the second set is 100 ft. away see figure 4-5.

Since vehicle detection occurs when a vehicle passes over the advanced detectors, the detection timestamp of these vehicles should be adjusted to the stop bar, as if the detection happens at there. The reason for this adjustment is to relate the vehicle count with the beginning and end of the effective green or/and effective red indication.

At the time when the data was obtained, the posted speed on both Division and 122nd street was 35 miles per hour (51.33 ft. per second) (Portland Bureau of Transportation). Although, the relationship between operating speed limit varies from one roadway to another, research has shown that operating speed can be 5-15 miles per hour higher than posted speed (Fitzpatrick et al, 2003). Thus, for the study intersection, the operating speed was assumed to 45 miles per hour (66 ft. per second). Using the operating speed along with the detector's distance to the stop bar, the adjustment of the first and second set of through detectors will be 1.3 and 2.8 seconds respectively. Likewise, the adjustment of the first and second set of the left turn detectors was 0.6 and 1.5 second respectively.

#### ***4.2.3.1 Determination of Vehicle Count per Cycle from High Resolution Data***

The simple way to calculate the vehicle counts using the high-resolution event based data is as follows:

- Subset the data when the “Event Type” column is equal to 8 (Phase Begin Yellow Clearance) and the “Parameter” column is equal to the number of the particular phase (e.g. phase #2) *and* when the “Event Type” column is equal to 82 (Detector On) and the “Parameter” column is equal to the channel number representing the group of the detectors working in accordance with the specified phase. Information on figure 4-5 and table 4-27 below are provided by the Portland Bureau of Transportation. Figure 4-5 shows the distribution of phases operating at the intersection and the location of detectors along with their designated numbers. Moreover, the first column in table 4-27 shows the designated number of detectors and the second shows the channel number, which is the output in data when an associated group of detectors are actuated. For example, if a vehicle was heading east on SE Division St, using the through lane, and actuating either detector number 5 or 6, the output of the controller in data is 82 (Detector On) in the “Event Type” column and 9 in the “Parameter” column.

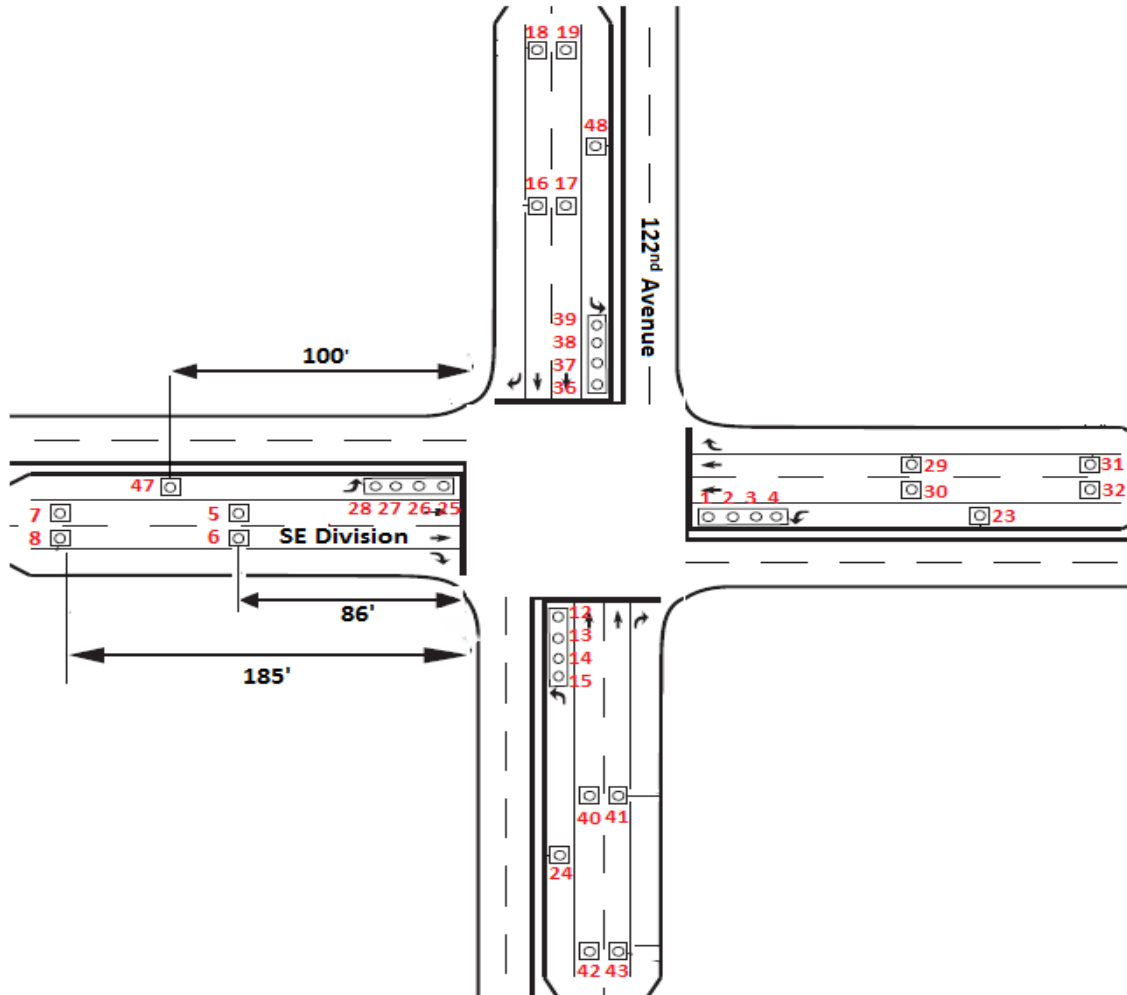


Figure 4-5: Loop detectors locations and numbers, and vehicular movement on each lane.

Loop(s)	VD
1-4	1
23	2
12-15	3
24	4
25-28	5
47	6
36-39	7
48	8
5-6	9
7-8	10
9	11
	12
10-11	13
16-17	14
18-19	15
20	16
	17
21-22	18
29-30	19
31-32	20
	21
33	22
34-35	23
40-41	24
42-43	25
	26
44	27
45-46	28

Red extend loops not connected

**Table 4-27:- Number of loop detector numbers and associated channel number (PBOT)**

The table 4-28 summarizes how the data should be subset if vehicle count is required for any phase at SE Division Street and 122<sup>nd</sup>, using either the front or back set of detectors. For example, if the number of vehicles using the through lanes of phase 2 needs to be determined based on the front sets of detectors, then data will be subset as the “Event Type” column is equal to 8 and 82, and Parameter column is equal to 2 and 9 (highlighted in red).



Event Type =	Parameter (Phase #) =	Event Type =	Parameter =	Detector Location
8	1	82	1	Front
			2	Back
8	2	82	9	Front
			10	Back
8	3	82	3	Front
			4	Back
8	4	82	14	Front
			15	Back
8	5	82	5	Front
			6	Back
8	6	82	19	Front
			20	Back
8	7	82	7	Front
			8	Back
8	8	82	24	Front
			25	Back

**Table 4-28: Shows how data needs to be subset when vehicles count each cycle, of a particular phase, is required at SE Division St. and 122nd Ave.**

- The “Timestamp” column should be subset to include the required period of time (normally the 24 hour day).
- The time of the detection occurrence at advance detectors should be adjusted, as if the detection happened at the stop bar, by adding projected travel time from the advance detector to the stop bar.
- Timestamp of the end of the green time should be adjusted to the effective end of the green time (add 2 seconds)
- The subset data above should be merged with the unique number of the cycles determined as in the previous section. The output will be represented as in the following table:

Timestamp	Adjusted Timestamp	Event Type	Parameter	Number of Cycle	Description
5:52:41	5:52:43	8	2	404	End of Green Phase
5:53:08	5:53:09	82	9	404	Detector On
5:53:12	5:53:13	82	9	404	Detector On
5:53:14	5:53:15	82	9	404	Detector On
5:53:27	5:53:27	8	2	405	End of Green Phase
5:53:46	5:53:47	82	9	405	Detector On
5:54:12	5:54:13	82	9	405	Detector On
5:54:33	5:54:34	82	9	405	Detector On
5:54:46	5:54:47	82	9	405	Detector On
5:54:53	5:54:55	8	2	406	End of Green Phase
5:55:12	5:55:13	82	9	406	Detector On
5:55:56	5:55:57	82	9	406	Detector On
5:55:58	5:55:59	82	9	406	Detector On
5:56:00	5:56:01	82	9	406	Detector On
5:56:11	5:56:12	82	9	406	Detector On
5:56:16	5:56:18	8	2	407	End of Green Phase

**Table 4-29: An example of partial subset data after adjusting timestamp to determine vehicle count per each cycle of phase 2**

Number of Cycle	Phase	Vehicles Count
405	2	3
406	2	4
407	2	5

**Table 4-30: An example of partial data after counting number of vehicles for each cycle of phase 2**

- Vehicle detection (when Event Type equals 82) should be counted between the previous and the current effective end of the green. The output will be as in table 4-30.

#### **4.2.3.2 Data Errors Associated with Determination of Vehicle Count per Cycle**

The above procedure is helpful to obtain the accurate vehicle count, if the data did not contain any errors. As mentioned earlier, the Counting Period is the period of time beginning from the end of the effective green time of the previous *phase instance* to the end of the effective green

time of the current *phase instance*. To reduce errors resulting from the missing rows of data, and consequently, to obtain more accurate results, the definition of the Counting Period is modified, **ONLY** with through movement phases (2, 4, 6, and 8). The new Counting Interval is defined as the period of time between the end of the green times of the previous and the current cycle. The reasons that this definition can only apply to the through movements is because they are constantly on demand; thus, they are rarely omitted. As a result, the cycle numbers and phase instance numbers of through phases are most likely the same.

To avoid errors in the data regarding vehicle count of through movement in cycle, an R script was written. It counts the number of vehicles, for a particular phase, **ONLY** between cycle number  $x$  and  $x+1$ . As a result, if this part of the data does not meet this condition, then the number of vehicles will not be counted, and associated rows will be tagged as an error. Regarding the left turn phases, because these phases are omitted more than through phases due to less demand, the associated counting interval remained the same, i.e. between the end of the green indication of the current and following phase instance.

#### 4.2.4 *Arrivals on Green & Arrival Type*

Arrivals on green is the proportion of vehicles arriving on the effective green phase to the total number of vehicles that have arrived during a particular cycle. This performance measure is important to provide information in regard with the quality of vehicle progression of the intersection. According to the HCM, this is calculated as follows:-

$$P_{\Phi, a} = \frac{V_{g, \Phi, a}}{V_{\Phi, a}}$$

$P_{\Phi}$ : - The percentage of vehicles arrived on the effective green for a particular phase

$V_{g,\Phi,a}$  :- The number of vehicles arrived on effective green for a particular phase at a particular cycle

$V_{\Phi,a}$  :- The total number of vehicles arrived for a particular phase at a particular cycle

Concerning Arrival Type, the Highway Capacity Manual (HCM) placed six qualitative categories to evaluate the quality of the progression. These categories are determined based a quantitative value called the Platoon Ratio ( $R_p$ ). Based on the HCM, the platoon ratio can be calculated as follows:-

$$R_{p, \Phi, a} = \frac{C, a}{g, \Phi, a} P_{\Phi, a}$$

$R_{p,\Phi,a}$  = The Platoon Ratio of particular phase at a particular cycle

$C_{,a}$  = cycle length (sec)

$g_{,\Phi,a}$  = effective green time (sec)

$P_{\Phi,a}$  = The percentage of vehicles arrived on the effective green for a particular phase

Arrival type can be determined, based on Platoon Ratio, from the HCM (Exhibit 15-4). See table 4-31 below

Arrival Type	Platoon Ratio ( $R_p$ )	Default Value	Progression Quality
1	$\leq 0.5$	0.333	Very Poor
2	$> 0.5 - 0.85$	0.667	Unfavorable
3	$> 0.85 - 1.15$	1	Random Arrivals
4	$> 1.15 - 1.5$	1.333	Favorable
5	$> 1.5 - 2$	1.667	Highly Favorable
6	$> 2$	2	Exceptional

**Table 4-31: Platoon ratio with associated quality of progression criteria (TRB, 2010)**

As we can see, the platoon ratio is the same as Arrival on Green value, but it takes into consideration the share of the green phase compared to the cycle ( $g/C$ ). In other words, the arrival on the green is factored by the  $g/C$ . Thus, when  $g/C$  is high, the platoon ratio decreases, resulting in a lower Progression Quality compared to the lower value of  $g/C$ .

#### ***4.2.4.1 Determination of Arrivals on Green & Arrival Type from High-Resolution Data***

In order to obtain the arrival on the green using the controller data (assuming that the data does not have any errors), the following steps should be taken:

- Sort the data when the “Event Type” column is equal to 1, 8 and 82, these codes are: “Phase Begin Green,” “Phase Begin Yellow Clearance,” “Detector On” respectively (Sturdevant et al., 2012).
- Sort the “Parameter” to include the particular phase (e.g. phase #2), and when it is equal to the number representing the group of the detectors that work in accordance with that specified phase (e.g. if phase #2 is to be picked, then the associated detectors group number will be 10, if “back” sets of detectors to be used see figure 4-25, table 4-27 , and table 4-28
- The “Timestamp” column should include the required analysis period (normally 24 hours)
- Adjust *actual* beginning and end of the green phase to obtain *effective* values. This can be achieved by adding start up lost time and the extension of the green time (2 seconds is used in this report – the default value in the HCM)
- Adjust the timestamp of vehicles detection as if the detection happened in the stop bar by adding the estimated travel time from the advance detector to the stop bar.
- The number of vehicles arriving during green indication can be determined by counting the number of rows when it is equal to 82 (vehicles detection) located between 1 and following 8 or, in other words, between the start and end of the green phase. See the rows highlighted in

red in table 4-32. Moreover, it is worth mentioning that the number of rows equaling 82 and lying between row 8 and following row 1 equal the number of vehicles arriving on red (between the end and the start of the green phase). See table 4-32

Adjusted Timestamp	Event Type	Parameter	Description
6:46:16	1	2	Effective green starts
6:46:30	82	9	Vehicle detection
6:46:46	82	9	Vehicle detection
6:46:50	8	2	Effective green ends
6:46:56	82	9	Vehicle detection
6:47:00	82	9	Vehicle detection
6:47:13	82	9	Vehicle detection
6:47:59	1	2	Effective green starts
6:48:06	82	9	Vehicle detection
6:48:19	82	9	Vehicle detection
6:48:21	82	9	Vehicle detection
6:48:22	82	9	Vehicle detection
6:48:23	82	9	Vehicle detection
6:48:24	82	9	Vehicle detection
6:48:25	82	9	Vehicle detection
6:48:27	82	9	Vehicle detection
6:48:33	8	2	Effective green starts

**Table 4-32: An example of partial subset data after adjusting timestamp to determine vehicles arrived during green indication of phase 2**

Timestamp	Event Type	Timestamp	Event Type	Arrivals on Green ( $V_g, \Phi$ )
00:17:12	1	00:17:36	8	2
00:18:10	1	00:18:45	8	8

**Table 4-33: An example of partial data after counting number of vehicles arrived during green indication of phase 2**

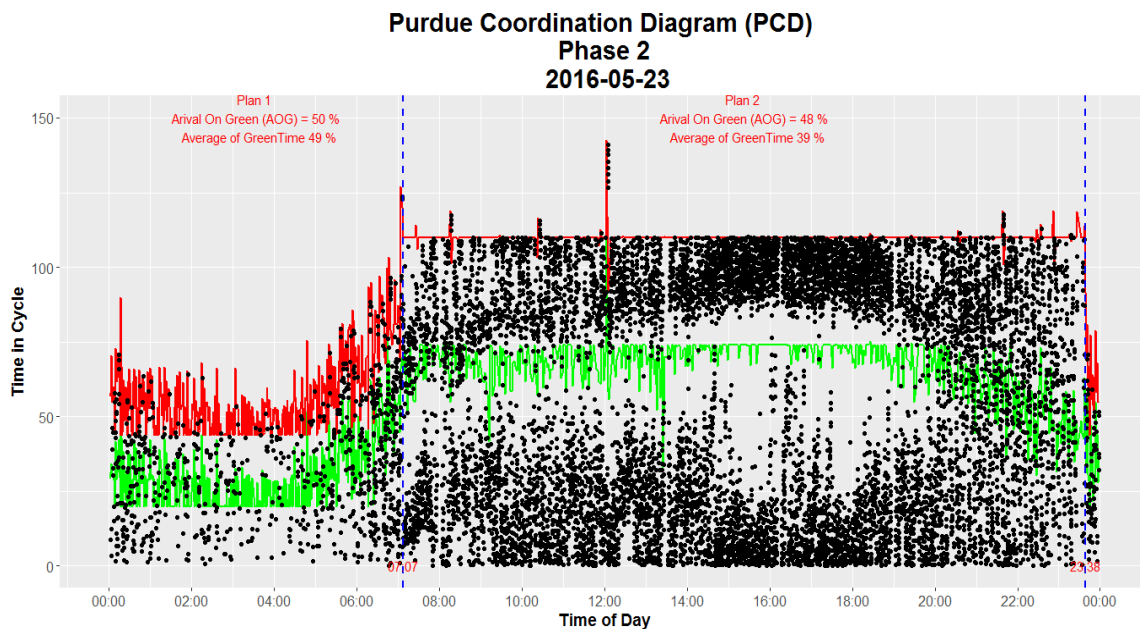
- Map the rows of the subset data with the associated Number of Cycles (NOC), green time, and green time over cycle ratio, and previously determined vehicle count. Then arrival type can be determined in accordance. See table below:

NOC	V <sub>g,2</sub>	g <sub>2</sub>	C	g/C	V <sub>2</sub>	P <sub>2,a</sub>	R <sub>p,2,a</sub>	AT	Progression Quality
449	2	34.2	95.8	0.36	5	0.4	1.11	3	Random arrivals
450	8	34.4	86.1	0.4	11	0.73	1.83	5	Highly Favorable

**Table 4-34: An example of partial data of determining number of vehicles arrived during green, platoon ratio, arrival type and quality of progression**

#### 4.2.5 *Purdue Coordination Diagram (PCD)*

PCD is a diagram which helps visually show the quality of the progression along an arterial or an intersection (Day et al, 2014). PCD visualizes vehicle arrival, therefore, showing if the platoons arrive on red or green during the cycle. The x-axis is the time of the day, and the y-axis is the time in the cycle. Figure 4-6 shows the Purdue Coordination Diagram (PCD) for phase number 2, of the SE Division and 122<sup>nd</sup> intersection.



**Figure 4-6: an example of PCD of phase 2**

The green and red lines on the graph represent the beginning and end of the effective green time respectively. On the other hand, the black dots on the graph denote the arrival of vehicles at the

intersection within the cycle. The dots below the red line illustrate vehicles arriving during the effective green time.

The Purdue Coordination Diagram (PCD) consists of three parts: 1) the beginning of the effective green (BOG), or the green line, 2) the end of the effective green (EOG), or the red line, and 3) vehicle arrival or detection, or the black dots.

In order to construct the plot, the three parts mentioned above should be obtained. To simplify, location of the beginning of the effective green (BOG) is the difference between the timestamp of the end of the effective green (EOG) of the *previous cycle* and the timestamp of the beginning of the effective green of the *current cycle*. On the other hand, the location of the end of the effective green (EOG) on the graph is the difference in the timestamp between the end of the effective green (EOG) of the *previous cycle* and the end of the effective green (EOG) of the *current cycle*. Lastly, the location of vehicle arrival (black dots) on the graphs represents the difference between the adjusted timestamp to the stop bar of the vehicle's detection on advanced detectors for the *current cycle*, and the timestamp of the end of the effective green (EOG) of the *previous cycle*.

#### ***4.2.5.1 Determination of Purdue Coordination Diagram (PCD) From High Resolution Data***

To plot the Purdue Coordination Diagram (PCD) using the controller data, the following steps should be followed (the first three of the following five points are the same as when data is sorted to determine arrivals on green):-

- Data should be subset when the “Event Type” column is equal to 1, 8, and 82. According to the traffic signal high-resolution event-based data logger, these codes are: “Phase Begin Green”, “Phase Begin Yellow Clearance”, “Detector On” respectively.



- Sort the “Parameter” data to include the particular phase (e.g. phase #2), when it is equal to the number representing the group of the detectors that work in accordance with a specified phase (e.g. if phase #2 is to be picked, then the associated detectors group number will be 10, if “back” sets of detectors are used). See figure 4-5 ,table 4-27, and table 4-28
- The “Timestamp” column should include a required period of analysis (normally 24 hours)
- Adjust *actual* beginning and end of the green phase to obtain the *effective* values. This can be achieved by adding the startup lost time and extension of the green time (2 seconds is used in this report – the default value in the HCM)
- Adjust the timestamp of vehicle detection as if the detection happened in the stop bar by adding the estimated travel time from the advance detector to the stop bar.
- Map every event row which subsetted as in the above points to its associated Number of Cycle (NOC).
- The beginning of the effective green within the cycle (or its location on the diagram) (BOG) is determined by subtracting the timestamp when the event equals 1 and the number of cycles equals (X) from the timestamp when the Event Type equals 8 and the number of the cycle is equal to (X-1).
- The end of the green effective (EOG) can be determined by subtracting the timestamp when the event equals 8 and the number of cycles equals (X) from the timestamp when the Event Type equals 8 and the number of cycle is equal to (X-1)
- The location of the vehicles arrived within the cycle (black dots on the graph) can be determined by subtracting the timestamp when the Event Type equals 82 and the NOC is equal to X-1 or X, from the timestamp when the Event Type equals 8 and the number of the cycle is equal to (X-1).

Timestamp	Event Type	Parameter	NOC	Time in Cycle	Description
6:46:50	8	2	449		
6:46:56	82	9	449	0:00:06	Vehicle Arrival
6:47:00	82	9	449	0:00:10	Vehicle Arrival
6:47:13	82	9	449	0:00:23	Vehicle Arrival
6:47:59	1	2	450	0:01:09	BOG
6:48:06	82	9	450	0:01:16	Vehicle Arrival
6:48:19	82	9	450	0:01:29	Vehicle Arrival
6:48:21	82	9	450	0:01:31	Vehicle Arrival
6:48:22	82	9	450	0:01:32	Vehicle Arrival
6:48:23	82	9	450	0:01:33	Vehicle Arrival
6:48:24	82	9	450	0:01:34	Vehicle Arrival
6:48:25	82	9	450	0:01:35	Vehicle Arrival
6:48:27	82	9	450	0:01:37	Vehicle Arrival
6:48:33	8	2	450	0:01:43	EOG

*Table 4-35: An example of partial data of determining the three components of PCD.*

#### **4.2.5.2 Data Errors Associated with Determination of Purdue Coordination Diagram (PCD)**

The procedure above can be implemented when there is no error in the data. As mentioned earlier, the data available in this report, unknowingly, contains missing rows. Therefore, an R script was written to avoid the consequences of these mistakes. After the filtering took place in the numbering cycles, the R script calls for rows between the “Event Type” equals 8 of the x index, the NOC is equal to y, and when the “Event Type” equals 8 of the index x+1 and the NOC is equal to y+1. The data set will be excluded if it does not meet this condition.

## **5.0 ANALYSIS AND DISCUSSION**

### **5.1 Pedestrians Delay**

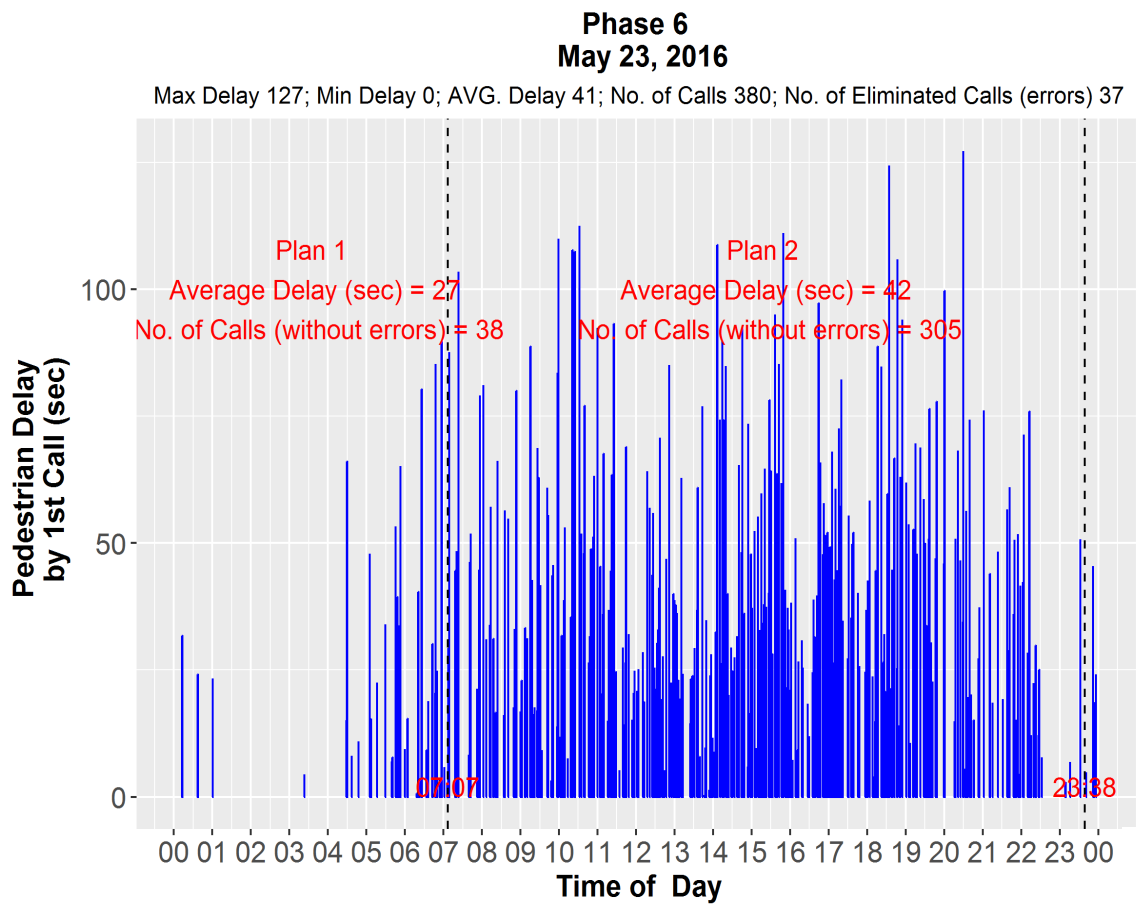
A complete data set is available for the weekdays of May 23 - 26, and the 1st of June. Partial data for the weekend is available on May 22, 2016, from 03:00 to 23:59 (Sunday). Thus, the following data analysis will be conducted on days with complete data. If necessary, these days will be compared to partially available data from the weekend, only for the hours available. For the most part, the following performance measures, explanatory graphs, and results are shown only for May 23, 2016 if they also represent identical measures for the remaining examined days.

The number of pedestrian calls for phases in the same direction (phases 2 and 6, and phases 4 and 8) should be exactly the same. Therefore, in the following analysis, one phase will be chosen to represent the number of pedestrian calls of the particular pedestrian movement (east-west or north-south). There is also a slight difference between pedestrian delays experienced when crossing for phases in the same direction. Hence, for simplicity, only one out of the two phases of parallel movements will be chosen to demonstrate the results of pedestrian delay (either phase 2 or 6 for north-south Pedestrian movement, or phase 4 or 8 for east-west pedestrian movement).

#### **5.1.1 *Pedestrians Delay by 1st Called***

The first factor to explore is the actual pedestrian delay based on the first call: The time when a pedestrian call is registered to pass the intersection in a particular direction until the pedestrian pass indication (walking symbol) is shown for that direction (Figures 5-1 and 5-2). The x-axis is the time of the day and the y-axis is the actual pedestrian delay based on the first call. This figure is taken from data on May 23, 2016, and is a phase 6 pedestrian movement (east-west movement).

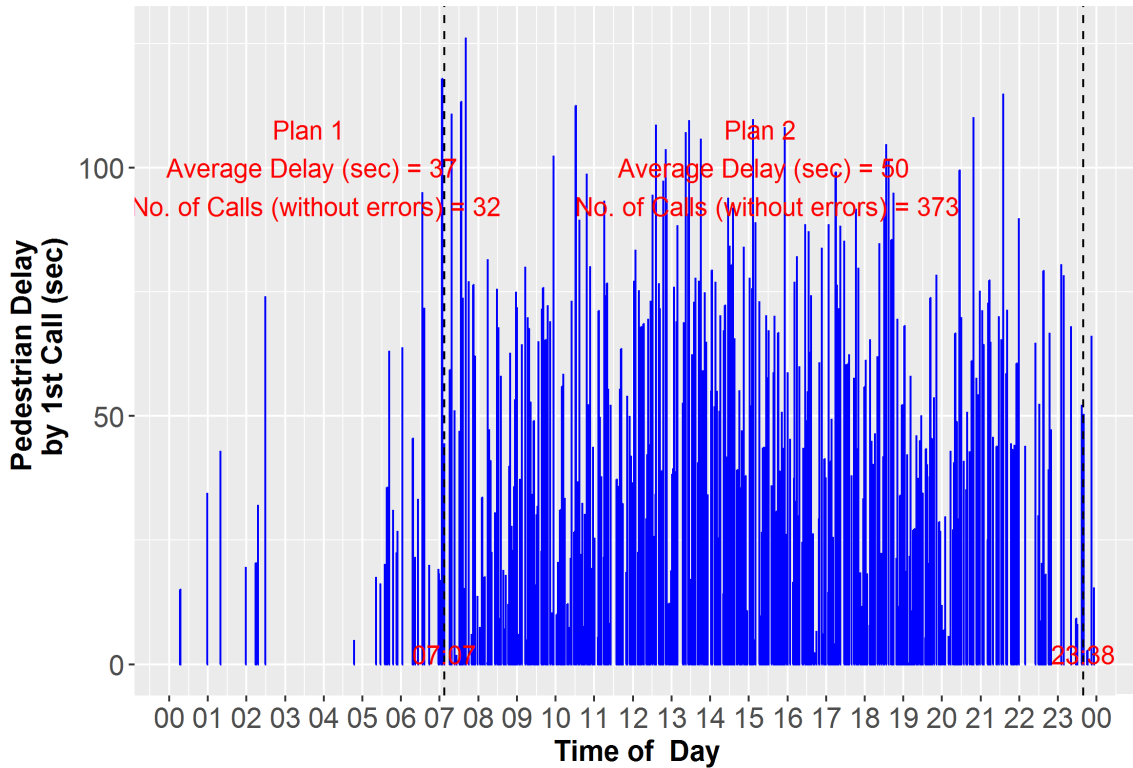
Both graphs display a random pattern regarding the actual pedestrian delay for the day. Moreover, results show there are fewer calls during the morning and late at night, particularly between 22:00 and 5:00. Conversely, the highest number of calls is found around peak hours – between 12:00 to 18:00. In graphs (5-1) and (5-2), a high number of calls are identified by large clusters of bolded blue lines.



*Figure 5-1: Actual pedestrian delay based on 1st call for phase 6*

**Phase 8  
May 23, 2016**

Max Delay 126; Min Delay 0; AVG. Delay 49; No. of Calls 423; No. of Eliminated Calls (errors) 18



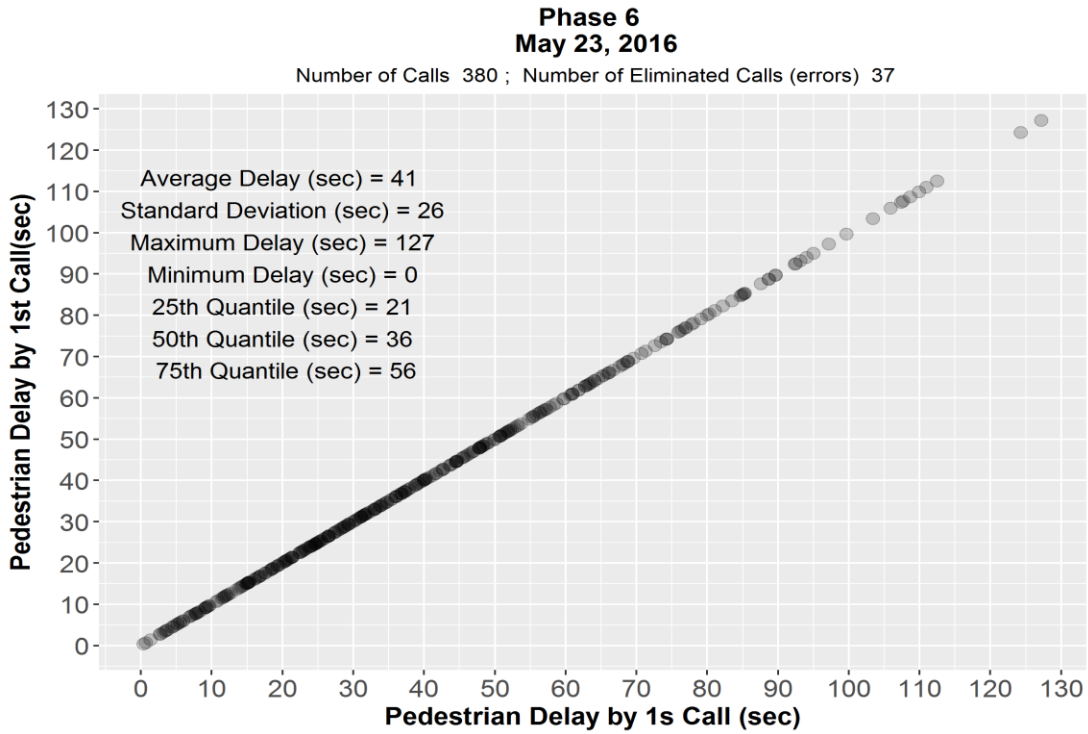
*Figure 5-2: Actual pedestrian delay based on 1st call for phase 8*

Continuing on May 23, the maximum, minimum, and average delay experienced by a registered pedestrian call waiting to cross the intersection, on phase 6, east-west (on Division street) were 127, 0, and 41 seconds, respectively. On the other hand, the same values on the same date for phase 8, north-south (on 122nd street) were 126, 0, and 49 seconds respectively.

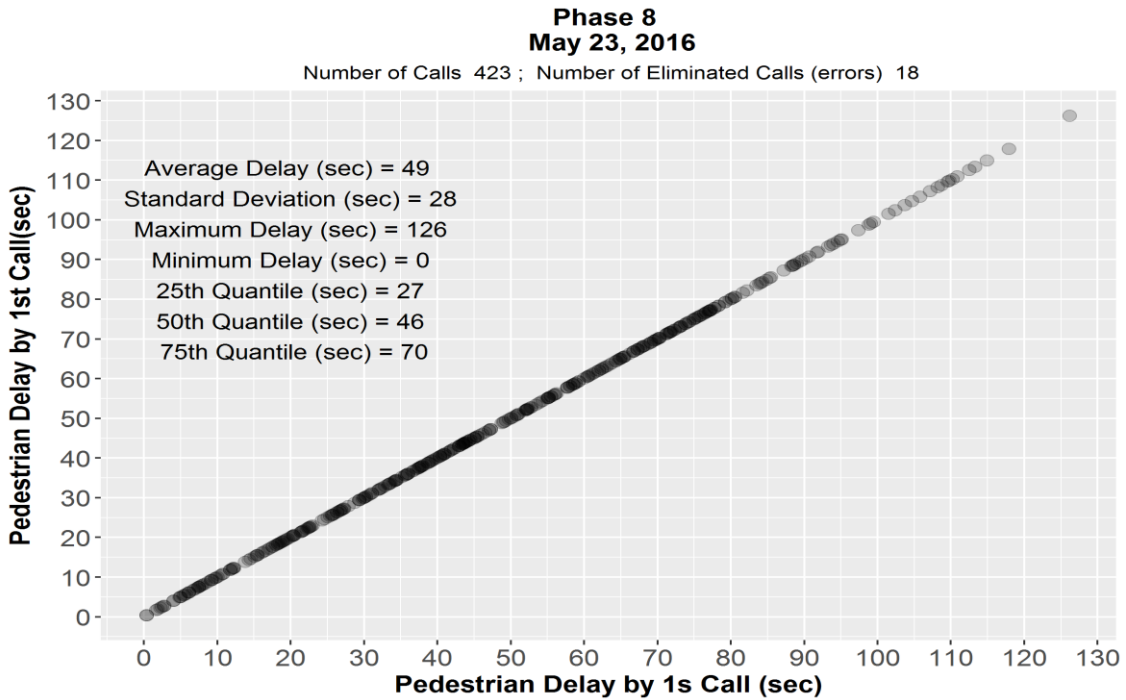
Graphs 5-1 and 5-2 can be altered for a different frame of view. The new graph displays the “actual delay based on 1<sup>st</sup> call” values sorted ascendingly, with lower values on the left and higher values on the right. Pedestrian delay values deviate from the minimum and maximum pedestrian delay, particularly the higher values (Graphs 5-3 and 5-4).

Additionally, the quantile of the pedestrian delay has been calculated. During phase 6, east-west (on Division street) on May 23, about 25% of pedestrians that registered a call

experienced 21 seconds or less of delay, and 50% experienced 36 seconds or less. Roughly 75% of pedestrians experienced a delay of 56 seconds or less.



*Figure 5-3: Average pedestrian delay based on 1st sorted ascendingly for phase 6*



*Figure 5-4: Average pedestrian delay based on 1st call, sorted ascendingly, for phase 8*

During phase 8, north-south (122nd street) for the same date, around 25% of pedestrians that registered a call experienced 27 seconds or less of a delay, and 46% experienced 37 seconds or less. Finally, 75% experienced a delay of 70 seconds or less.

For a more comprehensive view, tables 5-1 and 5-2 show the average pedestrian delay for a first call with some related descriptive statistics (standard deviation, minimum and maximum value, the quantiles of 25%, 50%, and 75%, and number of pedestrian calls registered in the controller) of the available data of the 5 weekdays for both phases 6 and 8 (both directions: east-west and north-south

Day	Date	Avg. Delay (sec)	St	Min (sec)	Max (sec)	25th	50th	75th
Mo	23/05/2016	41	26	0	127	21	36	56
Tu	24/05/2016	40	26	0	122	21	34	56
We	25/05/2016	40	26	0	127	20	35	57
Th	26/05/2016	38	24	0	115	21	33	56
Th	1/6/2016	41	26	0	125	21	36	58

**Table 5-1: Average pedestrian delay based on 1st call, with its descriptive statistics for phase 6**

Day	Date	Avg. Delay (sec)	St	Min (sec)	Max (sec)	25%	50%	75%
Mo	23/05/2016	49	28	0	126	27	46	70
Tu	24/05/2016	49	28	0	119	28	46	70
We	25/05/2016	49	29	0	123	26	46	70
Th	26/05/2016	47	29	1	129	23	44	67
Th	1/6/2016	51	28	0	120	30	48	72

**Table 5-2: Average pedestrian delay based on 1st call, with its descriptive statistics for phase 8**

For both directions, it can be seen that the average pedestrian delays are consistent for the five weekdays. The same trend can also be noticed among the standard deviation, minimum and maximum values for pedestrian delay. However, a pedestrian crossing the intersection in the north-south direction may experience slightly more delay than crossing the other direction, east-west (on SE Division). In general, a pedestrian registering a call to cross the intersection on 122nd avenue may experience an additional 9-second delay until the PASS indication is shown than crossing on

SE Division. In more detail, 25% of the pedestrians registering for a call to cross the intersection in the north-south direction faced a delay of approximately 6 seconds over pedestrians registering a call to cross in the east-west direction for the same intersection. Fifty percent experienced an 11-second delay, and 75% faced a delay of 13 seconds.

As mentioned earlier, partial data from May 22, 2016 (Sunday) is available. The data ranges from 03:00 to 23:59. Thus, a comparison was performed between these hours of available data for the weekend day, and the same hours of the available data for the weekdays. The comparison shown in tables 5-3 and 5-4 regards the average pedestrian delay with descriptive statistics. It is clear that the values for the average pedestrian delay by the first call are close to each other during weekdays and weekends for both directions.

Day	Date	Avg. Delay (sec)	St	Min (sec)	Max (sec)	25th	50th	75th
Su	22/05/2016	37	26	0	131	18	31	54
Mo	23/05/2016	41	26	0	127	21	36	56
Tu	24/05/2016	40	26	0	122	21	35	57
We	25/05/2016	40	26	1	127	20	36	58
Th	26/05/2016	39	24	0	115	21	33	56
Th	1/6/2016	41	26	0	125	22	37	59

**Table 5-3: Average pedestrian delay based on 1st call, with its descriptive stats of a weekend day vs 5 weekend days, from 3:00 to 23:59, for phase 6**

Day	Date	Avg. Delay (sec)	St	Min (sec)	Max (sec)	25th	50th	75th
Su	22/05/2016	46	29	1	121	21	43	69
Mo	23/05/2016	49	28	0	126	27	47	70
Tu	24/05/2016	50	28	0	119	30	48	70
We	25/05/2016	49	29	0	123	26	46	70
Th	26/05/2016	47	19	1	129	24	45	68
Th	1/6/2016	51	28	0	120	32	48	73

**Table 5-4: Average pedestrian delay based on 1st call, with its descriptive stats of a weekend day vs 5 weekend days, from 3:00 to 23:59, for phase 8**

To compare the two programmed plans in the controller with regard to pedestrian delay, Table 5-5 shows the average pedestrian delay for intersection plan 1 versus plan 2 for both



directions during the days from the available data. The average delay during plan 1 is less than the average delay during plan 2 for both pedestrian movements. In plan 2, for the east-west direction, the average pedestrian delay exceeded that of plan 1 by approximately 13 seconds. On the other hand, in the north-south pedestrian movement, the average delay of plan one is 21 seconds less than plan 2 delays. The reason that pedestrian delay by the first call is less during plan 1 than plan 2 is due to the fact that less vehicles approach the intersection during the early hours of the day, and consequently the green time for all the phases was less during plan 1, meaning pedestrians had less of a wait time.

Day	Date	Phase 6		Phase 8	
		Plan 1	Plan 2	Plan 1	Plan 2
Mo	23/05/2016	29	42	37	50
Tu	24/05/2016	26	42	28	52
We	25/05/2016	27	41	29	51
Th	26/05/2016	32	39	27	49
Th	1/6/2016	29	42	28	53

**Table 5-5: Average pedestrian delay based on 1st call, with its descriptive statistics of a weekend day vs 5 weekend days, from 3:00 to 23:59, for phase 8**

Moreover, graphs 5-1 and 5-2 can be smoothed out into a graph that shows average pedestrian delay (based on the 1st call) per each hour of the day. See figure 5-5 and 5-6. From the two figures below, in both directions, the average pedestrian delay per hour varies among the day. However, it can be seen that there is less average delay during the beginning of the day, from 00:00 to 6:00, and also at the end of the day at around 23:00. On the same date for phase 6 (east-west direction), the top 5 hours containing the highest average pedestrian delay were 10:00, 14:00, 15:00, 18:00, and 19:00 and the average delay was 52, 47, 49, 47, and 49 seconds respectively. The average

pedestrian delay in the other direction, phase 8 (north-south), occurred at 7:00, 12:00, 13:00, 14:00, and 21:00 and the average pedestrian delay was 59, 57, 64, 57, and 62 seconds, respectively.

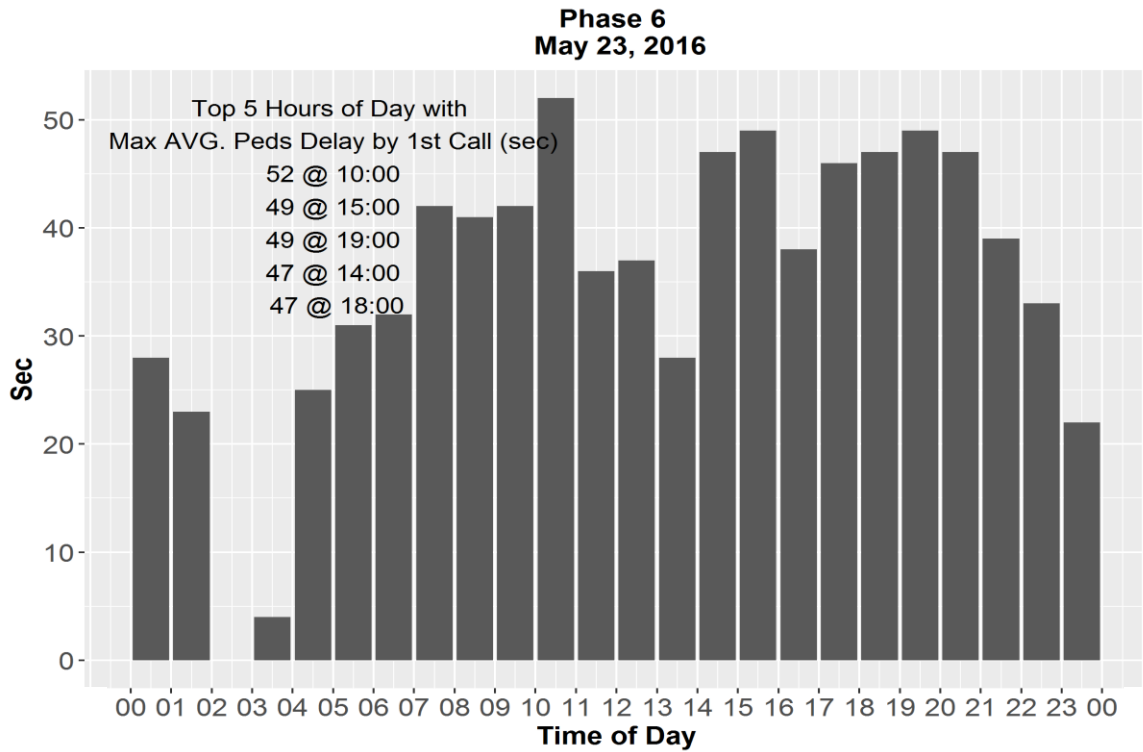


Figure 5-5: Average pedestrian delay based on 1st call for each hour of the day (phase 6)

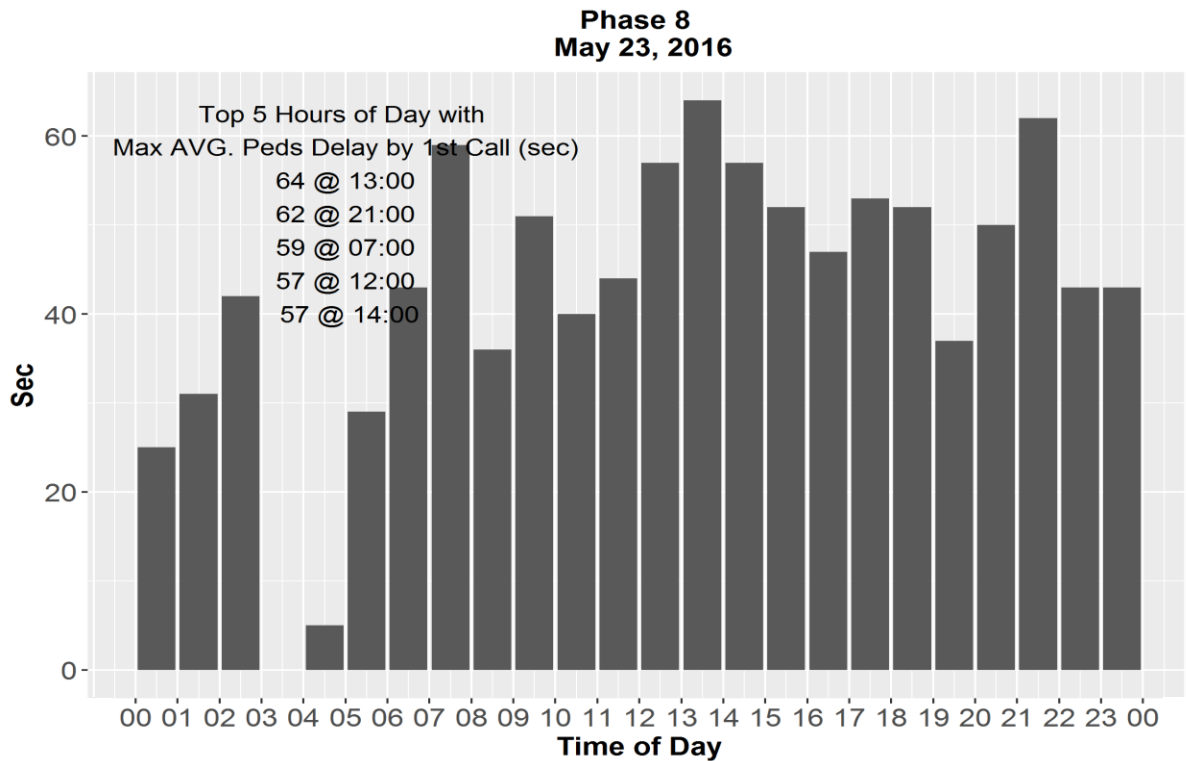


Figure 5-6: Average pedestrian delay based on 1st call for each hour of the day (phase 6)

In order to explore the 5 day available data regarding the average pedestrian delay, tables 5-6 and 5-7 show the available data for average pedestrian delay per hour for 5 weekdays in both directions. For both directions, there is no specific trend on particular hours of the day that contain the largest pedestrian delay by the first call. However, in the east-west pedestrian movement, it was found that at 2:00 and 17:00 there is always an average pedestrian delay of 40 seconds or more, consistent along the five days of available data. Furthermore, for the same direction, it was noticed that hours 8:00, 10:00, and 18:00 contained an average pedestrian delay of 40 seconds or more among 4 out of 5 days available in the data. In the other direction, for the north-south pedestrian movement, it was found that hours 13:00 and 15:00 compromise an average pedestrian delay of 50 seconds and are more consistent along the five days. In addition to the aforementioned two hours, pedestrians who registered for a call experienced an average delay of 50 seconds or more at 14:00, 17:00, and 18:00 for 4 out 5 days in the available data. The average pedestrian delay by the first call was 40 seconds or more during 9:00 – 18:00, and also for 20:00 and 23:00. To conclude, pedestrians who register for a call to cross the intersection may experience more delays during these hours of the day.

Hours of day	Avg. Delay	Hours of day	Avg. Delay	Hours of day	Avg. Delay	Hours of day	Avg. Delay	Hours of day	Avg. Delay
23 May, 2016		24 May, 2016		25 May, 2016		26 May, 2016		01 June, 2016	
10:00	52	12:00	59	17:00	47	13:00	47	13:00	49
15:00	49	18:00	48	8:00	46	12:00	46	22:00	48
19:00	49	8:00	47	12:00	46	23:00	45	8:00	47
14:00	47	16:00	47	18:00	46	17:00	44	16:00	46
18:00	47	11:00	46	22:00	46	19:00	44	17:00	46
20:00	47	9:00	44	11:00	43	7:00	42	10:00	45
17:00	46	10:00	43	14:00	42	14:00	42	14:00	45
7:00	42	17:00	42	16:00	42	11:00	40	12:00	43
9:00	42	14:00	40	10:00	41	16:00	40	6:00	42
8:00	41	5:00	38	15:00	40	8:00	39	7:00	42
21:00	39	13:00	38	13:00	38	18:00	39	15:00	42
16:00	38	21:00	38	20:00	38	6:00	35	18:00	42
12:00	37	15:00	37	9:00	37	9:00	35	19:00	41
11:00	36	6:00	36	3:00	36	15:00	35	9:00	40
22:00	33	22:00	35	21:00	36	20:00	34	21:00	35
6:00	32	20:00	34	6:00	34	0:00	33	11:00	33
5:00	31	19:00	33	0:00	33	10:00	32	20:00	32
0:00	28	23:00	23	19:00	31	21:00	32	0:00	31
13:00	28	1:00	22	23:00	29	22:00	31	2:00	29
4:00	25	7:00	20	4:00	27	3:00	30	3:00	29
1:00	23	0:00	14	7:00	26	4:00	23	4:00	28
23:00	22	4:00	11	5:00	22	5:00	22	5:00	23
3:00	4	3:00	8	2:00	13	1:00	16	23:00	22
2:00	NA	2:00	NA	1:00	8	2:00	NA	1:00	17

*Table 5-6: Average pedestrian delay based on 1st call of five days of available data, phase 6*

Hours of day	Avg. Delay	Hours of day	Avg. Delay	Hours of day	Avg. Delay	Hours of day	Avg. Delay	Hours of day	Avg. Delay
23 May, 2016		24 May, 2016		25 May, 2016		26 May, 2016		01 June, 2016	
13:00	64	23:00	71	18:00	59	18:00	63	21:00	61
21:00	62	17:00	62	14:00	58	15:00	58	11:00	60
7:00	59	13:00	61	16:00	57	14:00	56	13:00	60
12:00	57	15:00	57	10:00	55	23:00	55	9:00	59
14:00	57	12:00	55	15:00	55	10:00	52	15:00	59
17:00	53	16:00	55	17:00	52	13:00	52	20:00	58
15:00	52	8:00	54	20:00	52	9:00	51	23:00	58
18:00	52	19:00	54	7:00	51	20:00	49	22:00	57
9:00	51	7:00	52	13:00	50	7:00	48	8:00	54
20:00	50	11:00	52	9:00	49	17:00	48	10:00	54
16:00	47	18:00	52	11:00	49	11:00	47	14:00	51
11:00	44	22:00	49	19:00	49	12:00	47	17:00	50
6:00	43	10:00	48	23:00	47	16:00	47	16:00	49
22:00	43	14:00	48	8:00	43	19:00	47	12:00	48
23:00	43	1:00	45	12:00	43	8:00	41	19:00	45
2:00	42	20:00	44	6:00	41	21:00	38	18:00	43
10:00	40	9:00	41	21:00	40	6:00	34	1:00	42
19:00	37	21:00	40	22:00	32	22:00	26	5:00	39
8:00	36	5:00	35	1:00	28	5:00	24	7:00	34
1:00	31	0:00	25	5:00	28	1:00	23	6:00	31
5:00	29	6:00	25	2:00	26	3:00	12	2:00	28
0:00	25	2:00	21	0:00	24	4:00	9	0:00	24
4:00	5	4:00	10	4:00	20	0:00	4	4:00	22
3:00	NA	3:00	8	3:00	11	2:00	NA	3:00	8

*Table 5-7: Average pedestrian delay based on 1st call of five days of available data, phase 8*

### **5.1.1.1 Pedestrian delay by 1st call compared to reality**

The pedestrian delay based on the first call should represent the delay experienced by the first person/s registering a request by pressing the pushbutton to pass the intersection. This implies that this delay does not reflect delays faced by others who arrive at the intersection after the call was registered. Furthermore, although delay based on the first call was calculated and analyzed, there is no available way at the moment to confirm whether the first person requested to pass the intersection stayed at their position after their call was registered. Thus, there is a level of

uncertainty for all delays by the first call if these times were really experienced by the pedestrian who registered the first call or not.

### 5.1.2 *Number of Pedestrian Calls*

Table 5-8 displays the number of calls for the 5 weekdays based on available data for both phases 6 and 8. There is a small variation of the number of calls in these days for each phase. To compare the number of pedestrian calls in both directions, there are more calls in the north-south pedestrian movement than the east-west direction. On average, the number of calls on 122nd exceeded the number of calls on SE Division Street by nearly 30.

Day	Date	# Ped Calls (phase 6)	# Ped Calls (phase 8)
Mo	23/05/2016	380	423
Tu	24/05/2016	396	454
We	25/05/2016	424	431
Th	26/05/2016	397	434
Th	1/6/2016	448	455

*Table 5-8: Number of pedestrian calls of the 5 days of available data (phases 6 & 8)*

Furthermore, the number of pedestrian actuations, for both directions, was determined, see table 5-9. The intersection controller registers a pedestrian actuation for a particular direction whenever a pedestrian hits the push button to cross the street. These values are not accurate because some people tend to press the cross button more than once, and every time it is pressed, it is registered in the controller as an action, encoded with the number 90. However, although inaccurate, the number of pedestrian actuations is consistently more on the north-south direction than the east-west direction, which may indicate a larger number of pedestrians crossing the intersection of 122nd street than on SE Division.

Day	Date	Number of Pedestrian Actuations (Phase 6)	Number of Pedestrian Actuations (Phase 8)
Mo	23/05/2016	1406	1699
Tu	24/05/2016	1390	1854
We	25/05/2016	1634	1705
Th	26/05/2016	1223	1789
Th	1/6/2016	1759	2046

**Table 5-9: Number of pedestrian actuations of the 5 days of available data (phases 6 & 8)**

To continue, the number of pedestrian calls and pedestrian actuations of the five weekdays are compared to those drawn from the available weekend data between 3:00 to 23:59 (Tables 5-10 and 5-11).

Day	Date	Number of Pedestrian calls (Phase 6)	Number of Pedestrian Actuations (Phase 6)
Su	22/05/2016	270	784
Mo	23/05/2016	377	1402
Tu	24/05/2016	391	1380
We	25/05/2016	418	1627
Th	26/05/2016	392	1206
Th	1/6/2016	434	1721

**Table 5-10: Number of pedestrian's calls and actuations of a weekend day vs 5 weekend days, from 3:00 to 23:59, for phase 6**

Day	Date	Number of Pedestrian calls (Phase 8)	Number of Pedestrian Actuations (Phase 8)
Su	22/05/2016	315	1074
Mo	23/05/2016	416	1685
Tu	24/05/2016	441	1823
We	25/05/2016	421	1684
Th	26/05/2016	427	1768
Th	1/6/2016	476	2028

**Table 5-11: number of pedestrian's calls and actuations of a weekend day vs 5 weekend days, from 3:00 to 23:59, for phase 8**

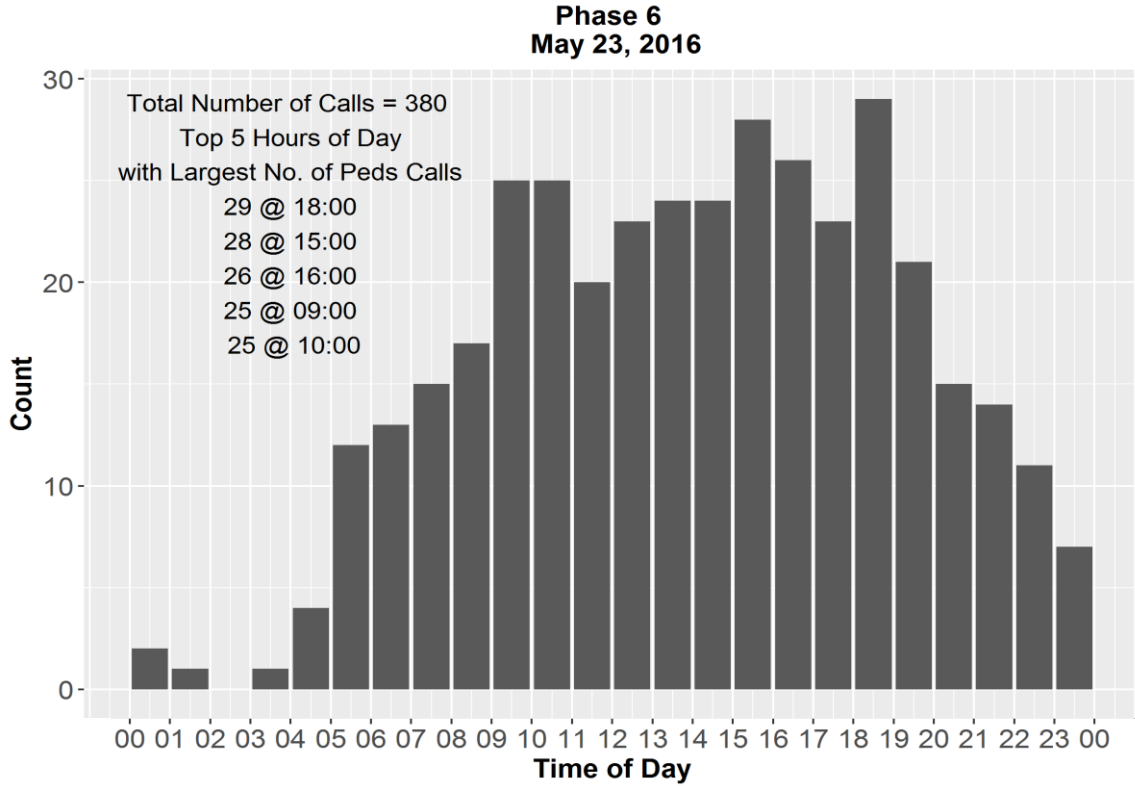
The number of pedestrian calls and actuations are less on the weekend than the weekdays for both directions. This likely indicates that the number of pedestrians crossing the intersections during the weekdays are higher than the weekends.

As mentioned earlier, figure 5-1 and 5-2 can be transformed to produce a graph with the number of pedestrian calls for each hour of the day (Figures 5-7 and 5-8). To illustrate, during May 23, 2016 on phase 6, east-west (Division street), and on phase 8, north-south (122nd street), the number of pedestrian calls are lower at the beginning of the day but gradually increased until 08:00 - 9:00. After that time, the number of pedestrian calls varied slightly until around 20:00, when the number of calls gradually decreased until the end of the day.

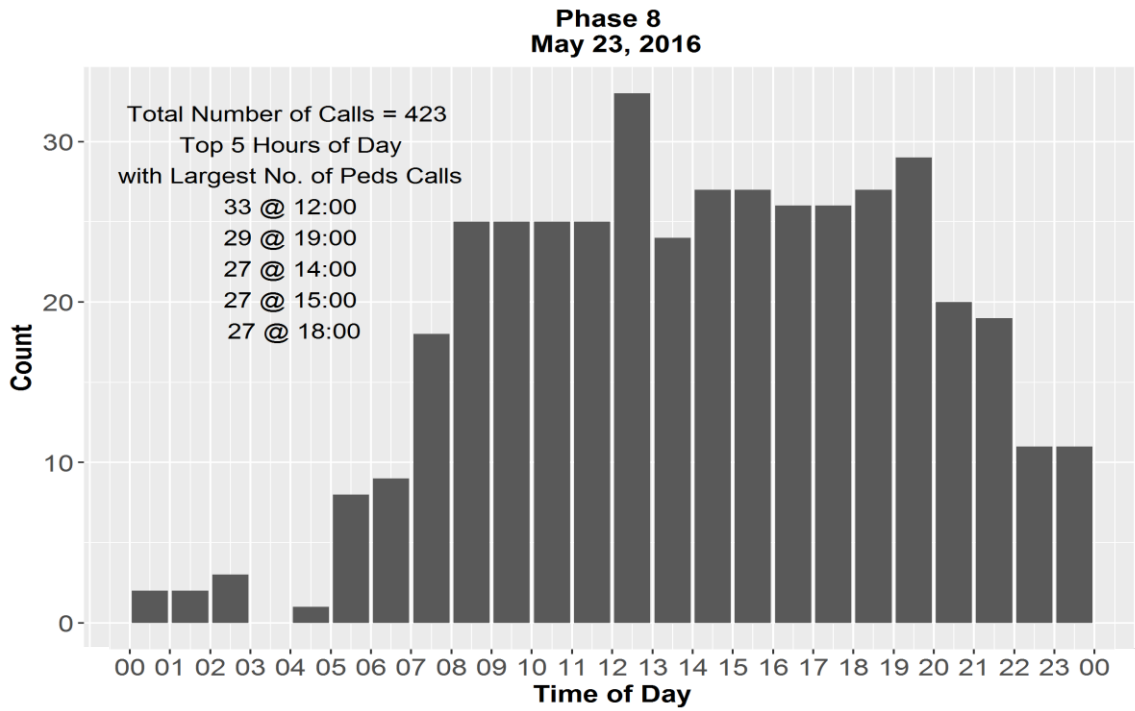
The 5 top hours with the highest number of pedestrian calls in both directions was also identified. In the east-west direction (phase 6), hours, 9:00, 10:00, 15:00, 16:00, and 18:00, the number of calls placed were 25, 25, 28, 26, and 29 respectively. For the same date in the north-south direction, the highest number of pedestrian calls were placed at 12:00, 14:00, 15:00, 18:00 and 19:00, at 33, 27, 27, 27, and 29 respectively. To expand the above analysis pertaining to the number of calls sorted in descending order, per each hour for the five days of available data, for both phase 6 and Phase 8 respectively, see Table 5-12 and 5-13. It is evident that there are no specific hours where the highest number of calls were placed in either pedestrian direction. The top 10 hours containing the highest number of pedestrian calls along the five available days were determined. For the east-west (SE Division) pedestrian movement, it was found that the hours of 1:00 – 17:00 were consistent among the five days.

In the north-south pedestrian movement (122<sup>nd</sup> Street), 2:00 - 18:00 include the highest number of calls. In contrast, the ten hours with the least number of calls occurred between 00:00 - 7:00 and 22:00 - 23:00. To summarize, the hours between 1:00 - 18:00 were always found in the top ten hours with highest number of calls.





*Figure 5-7: Number of pedestrian calls per each hour of the day (phase 6)*



*Figure 5-8: Number of pedestrian calls per each hour of the day (phase 8)*

Hours of Day	# of Calls	Hours of Day	# of Calls	Hours of Day	# of Calls	Hours of Day	# of Calls	Hours of Day	# of Calls
23 May, 2016		24 May, 2016		25 May, 2016		26 May, 2016		01 June, 2016	
15:00	159	17:00	178	15:00	222	17:00	104	13:00	194
18:00	131	11:00	143	14:00	163	12:00	103	14:00	155
16:00	123	16:00	124	20:00	154	14:00	101	16:00	139
19:00	99	20:00	110	16:00	127	16:00	98	19:00	133
14:00	97	10:00	103	17:00	127	15:00	94	12:00	128
17:00	96	15:00	102	18:00	125	18:00	86	20:00	124
8:00	86	14:00	95	13:00	119	13:00	68	15:00	114
13:00	84	13:00	86	11:00	98	19:00	68	17:00	113
21:00	77	18:00	73	12:00	87	21:00	62	9:00	95
10:00	76	7:00	67	21:00	68	20:00	58	18:00	88
12:00	69	8:00	60	8:00	60	11:00	55	11:00	83
20:00	59	9:00	60	10:00	60	9:00	52	8:00	72
11:00	58	12:00	56	19:00	60	10:00	50	10:00	61
9:00	45	19:00	48	9:00	46	6:00	46	5:00	55
5:00	43	6:00	27	6:00	29	7:00	39	21:00	41
7:00	36	21:00	24	7:00	28	8:00	36	7:00	36
6:00	27	22:00	11	22:00	20	23:00	28	22:00	33
22:00	16	0:00	9	5:00	12	4:00	20	6:00	25
23:00	10	5:00	5	23:00	11	5:00	15	23:00	25
4:00	7	4:00	4	4:00	10	22:00	15	1:00	23
3:00	4	23:00	3	1:00	3	1:00	12	0:00	10
0:00	3	1:00	1	0:00	2	3:00	8	2:00	5
1:00	1	3:00	1	2:00	2	0:00	5	4:00	4
2:00	0	2:00	0	3:00	1	2:00	0	3:00	3

**Table 5-12: Number of pedestrian calls per each hour of the day of five days of available data, phase 6**

Hours of Day	# of Calls	Hours of Day	# of Calls	Hours of Day	# of Calls	Hours of Day	# of Calls	Hours of Day	# of Calls
23 May, 2016		24 May, 2016		25 May, 2016		26 May, 2016		01 June, 2016	
15:00	170	15:00	227	16:00	157	15:00	222	15:00	230
12:00	157	18:00	174	10:00	144	17:00	207	11:00	202
18:00	124	14:00	158	13:00	144	14:00	176	12:00	167
11:00	123	10:00	139	9:00	139	16:00	132	17:00	166
16:00	123	12:00	135	14:00	136	21:00	125	13:00	155
17:00	106	19:00	135	17:00	136	18:00	116	20:00	152
14:00	103	11:00	123	20:00	121	12:00	114	10:00	130
19:00	92	17:00	111	18:00	117	19:00	91	16:00	106
9:00	90	8:00	107	15:00	99	13:00	84	8:00	105
8:00	89	13:00	94	11:00	94	9:00	77	14:00	100
7:00	88	16:00	90	12:00	77	8:00	74	18:00	96
13:00	75	22:00	66	19:00	76	10:00	68	9:00	95
21:00	72	21:00	62	8:00	60	11:00	63	19:00	85
10:00	65	9:00	56	7:00	48	20:00	63	21:00	57
20:00	65	20:00	41	21:00	35	7:00	46	22:00	57
22:00	50	7:00	32	23:00	33	23:00	36	7:00	52
23:00	41	6:00	27	5:00	25	22:00	30	6:00	28
6:00	29	5:00	18	6:00	20	6:00	24	23:00	23
5:00	21	23:00	16	0:00	14	1:00	17	0:00	12
0:00	5	2:00	15	22:00	14	5:00	12	4:00	11
2:00	5	0:00	11	4:00	6	3:00	5	5:00	8
1:00	4	4:00	7	2:00	5	0:00	4	2:00	5
4:00	2	1:00	5	3:00	3	4:00	2	3:00	3
3:00	0	3:00	5	1:00	2	2:00	0	1:00	1

**Table 5-13: Number of pedestrian calls per each hour of the day of five days of available data, phase 6**

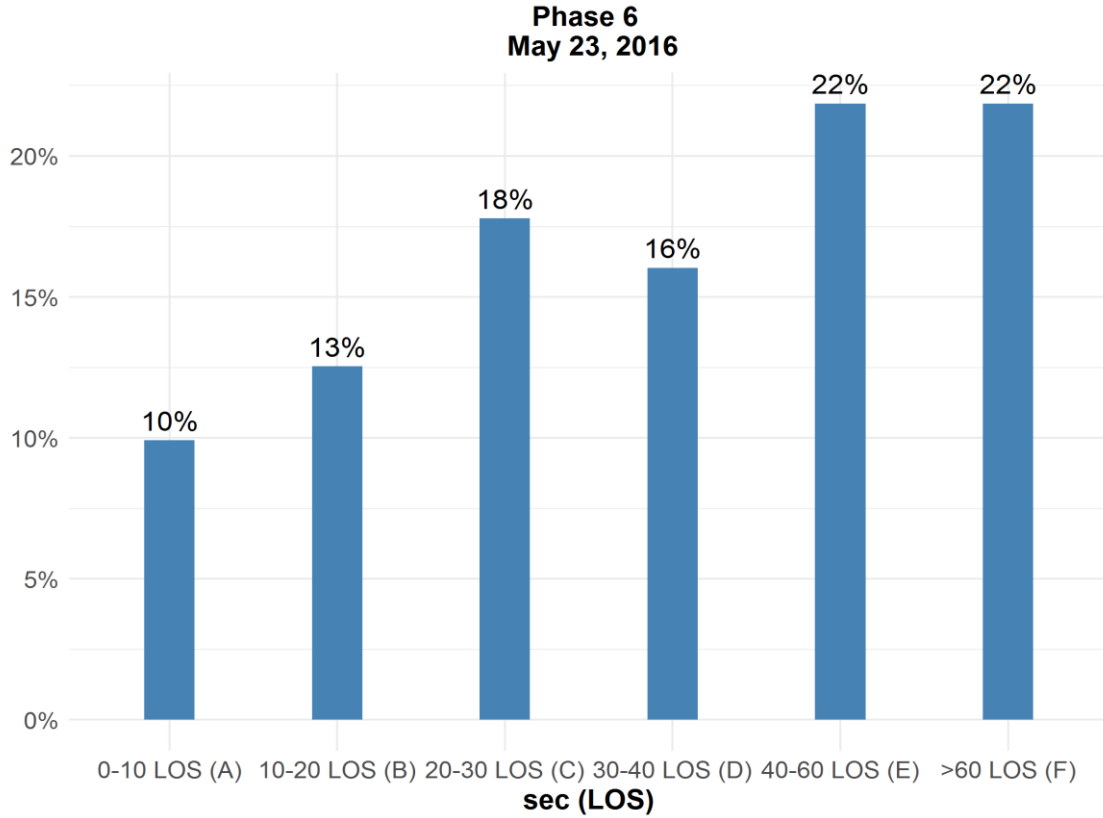
### 5.1.3 Level of Service Based on Pedestrian Delay by 1<sup>st</sup> Call

Another method for determining pedestrian actual delay in the studied intersection is by creating bins of delays and allocating the observed delays, determined from controller data, to

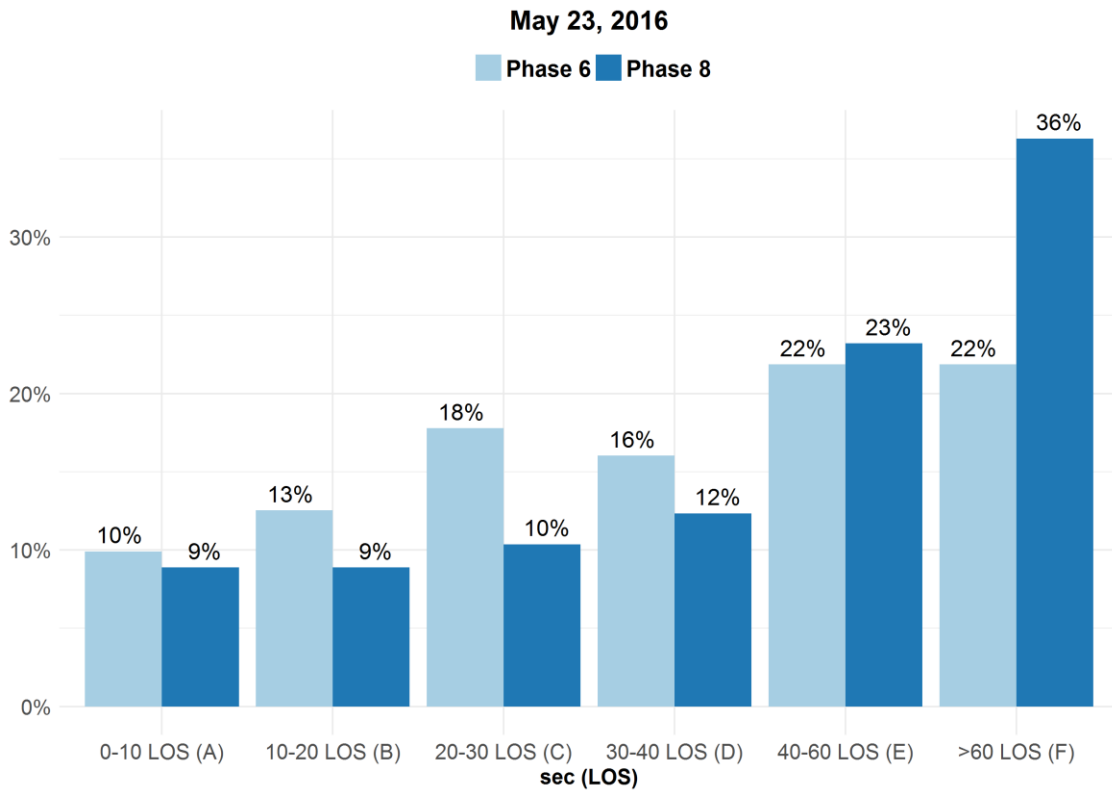
associated bins. The Highway Capacity Manual (HCM) assigns Level of Service to intersection based on ranges of delays, see table 4-4. In the HCM, the average pedestrian delay is calculated based on the cycle length and effective green time for the pedestrian. Pedestrian delay with associated level of service is illustrated in table 4-4. The same HCM level of service criteria for pedestrians can be used with pedestrian delay by actuation and assessed. Research has shown that pedestrians may engage in risky behavior if they experience a delay of 30 seconds or more (Mahmassani & Peeta, 1993).

Figure 5-19 shows the pedestrian delay versus the ranges of delay with its associated level of service on May 23, 2016, for east-west pedestrian movements (SE Division Street). Around 60% of pedestrians who first registered a call to cross experienced Levels of Service (LOS) F, E, and C, while around 40% faced delays which were under LOS A, B, and C. For the other direction (122<sup>nd</sup> Ave), the delay range was larger. Around 72% of the Level of Service based on pedestrian delays was LOS F, E, and C, while only 28 % were under LOS A, B, and C.

Furthermore, Figure 5-10, displays a comparison between east - west and north - south Level of Service based on the delay experienced by a pedestrian who registered a call. The level of service in the east - west direction is slightly better than that of north-south. There were 14 % less people in the east to west direction who experienced a delay under LOS F. In regard to LOS A and E, and associated with the two directions of pedestrian movement, they were quite similar with a slight difference of 1 - 2%. To summarize, the parentage of pedestrians crossing in the east-west direction of the intersection experienced a range of delays falling within LOS A, B, C, and E are more than the percentage of those crossing in the north-south direction, and less with regard to LOS F.



*Figure 5-9: Level of Service based on pedestrians 1st calls (phase 6)*



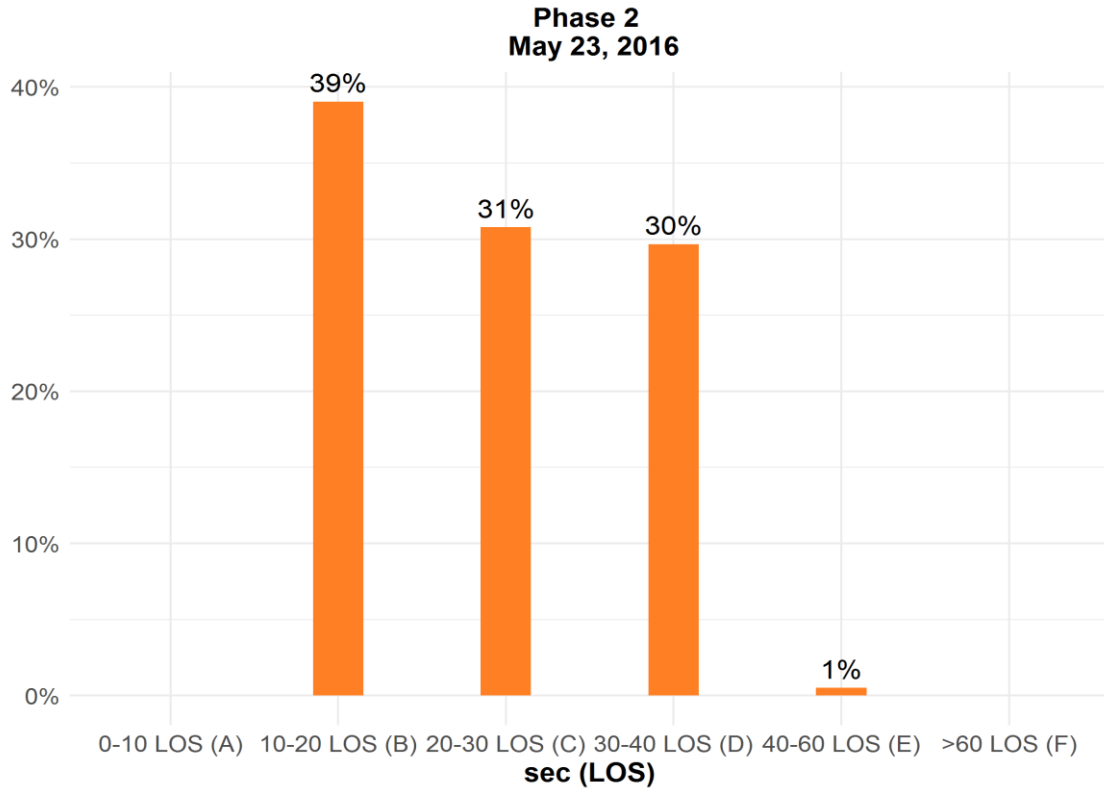
*Figure 5-10: Level of Service based on pedestrians 1st calls of phase 6 vs. phase 8*

#### 5.1.4 *Highway Capacity Manual (HCM) based Delay and Level of Service*

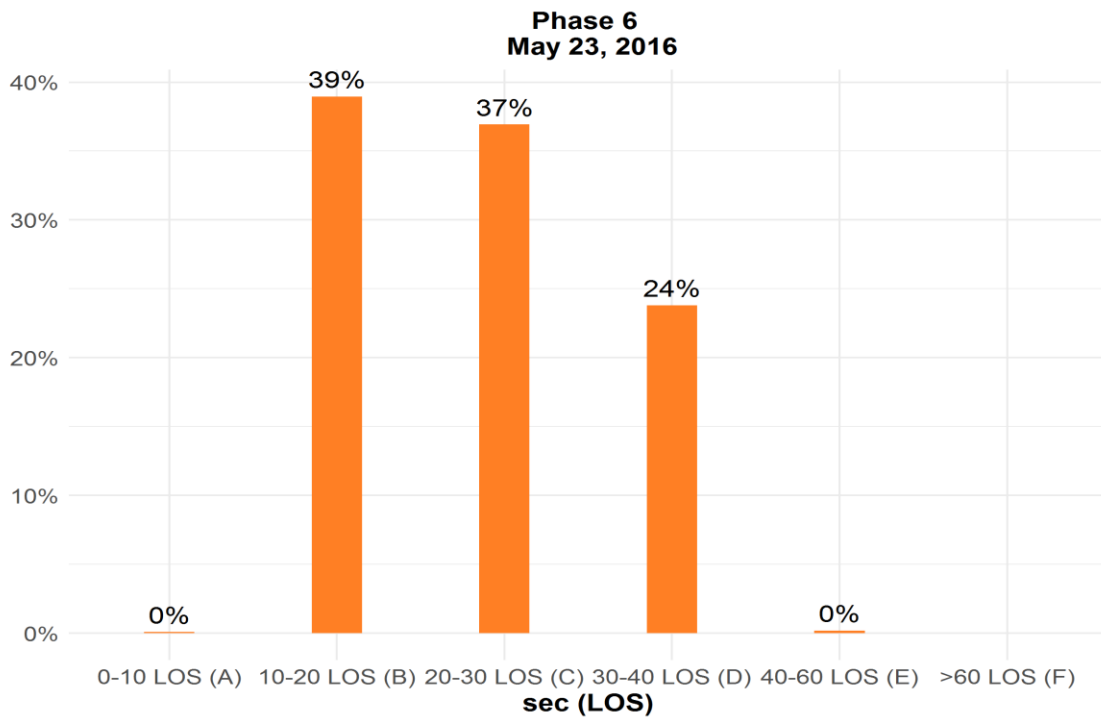
As mentioned earlier, one way to calculate the estimated average delay per pedestrian at a signalized intersection is by using the Highway Capacity Manual equation, found in chapter 18 (HCM). This equation is dependent on two variables: cycle length and effective green time for pedestrians that is equal to the walking indication period plus 4 seconds of a pedestrian clearance interval. These two variables can be obtained from the controller data, used for calculating the average delay per pedestrian and assigning a Level of Service from the HCM to these averaged delays accordingly.

Figures 5-11 and 5-12 demonstrate the Level of Service of Phases 6 and 2 (pedestrian movement on SE Division Street) based on average pedestrian delay. This was calculated using the 18-5 HCM equation, on May 23, 2016. Nearly all average delays per person were within LOS B, C, and D, indicating a period of delay from 10 - 40 seconds. In particular, Level of Service B had the highest percentage for both phase 2 and phase 6 at 39%. This means that 39% of the time on May 23, the estimated average delay per pedestrian is between 10 to 20 seconds. For phases 2 and 6, LOS C had percentages of 31% and 37%, and the percentage of LOS D were 30% and 24%.

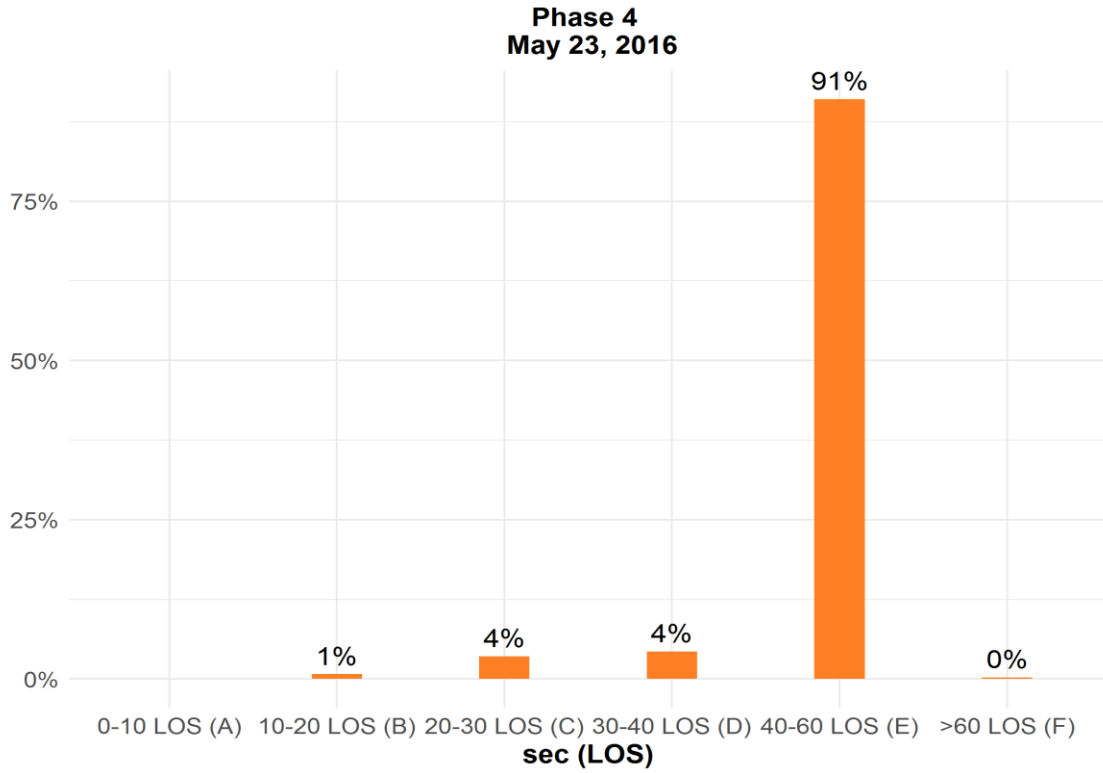
On the other hand, for the same day, the average pedestrian delay, as determined by the HCM approach for phases 4 and 6, was within the range of 40-60 seconds, which is represented by LOS E. The parentage was 91% for phase 4 and 92% for phase 8.



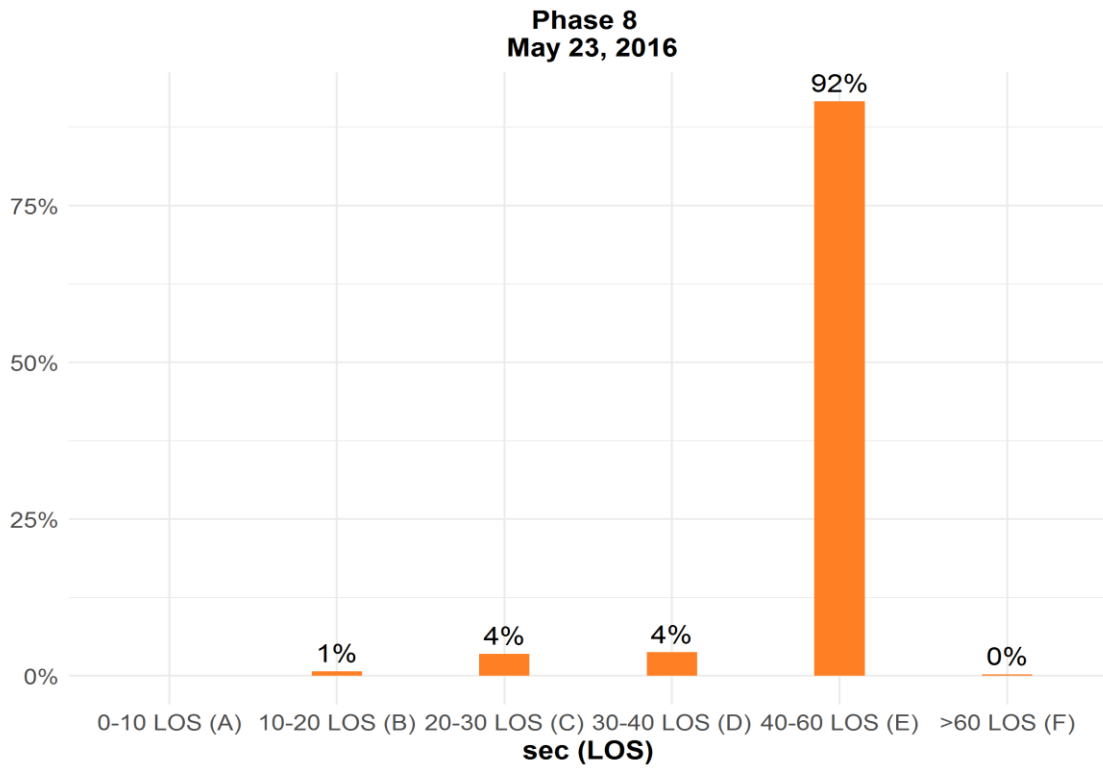
*Figure 5-11: Level of Service based on based on estimated delay determined by HCM approach (phase 2)*



*Figure 5-12: Level of Service based on based on estimated delay determined by HCM approach (phase 6)*



*Figure 5-13: Level of Service based on based on estimated delay determined by HCM approach (phase 4)*



*Figure 5-14: Level of Service based on based on estimated delay determined by HCM approach (phase 8)*



The estimated average delay per pedestrian was higher for phases 4 and 8 compared to 2 and 6 because the effective green time for phases 4 and 8 are less than that of phases 2 and 6. As mentioned earlier, calculating the average delay using the HCM equation is dependent on two variables: cycle length and effective green time. Since the cycle length is a fixed value for these phases, effective green time for a particular walking indication period was the dominant variable for the difference in results seen above.

Figures 5-15 and 5-16 show the timing for pedestrian signal indications (steady “WALKING”, “Flash Don’t Walk (FWD)”, and solid “DON’T WALK” indications) for all pedestrian movement directions (phases 2, 6, 4 and 8) on May 23, 2016. Time of “WALKING” indication for phases 2 and 6 was higher than the time in phases 4 and 8. In plan 2 for that day, the average walking indication time in phases 2 and 6 exceeded that of phases 4 and 8 by around 20 seconds. Therefore, the LOS based-on-average pedestrian delay calculated using the HCM method was higher during phases 2 and 6.

In plan 1 for this day, in all phases of pedestrian movement, the time of indications “WALKING” and “Flash Don’t Walk” are fixed values at 7 and 17 seconds, respectively. Also, the time of solid “DON’T WALK” indications vary for all pedestrians, as it depends on the vehicle actuations of adjacent vehicle phases. Timing changes in plan 2 for phases 2 and 6 is consistent with phases 2 and 4. For phases 2 and 6, the time of “Flash Don’t Walk” and solid “DON’T WALK” are fixed values at 17 and 4 seconds, while the time for the “WALKING” indication varies.

It is worth mentioning that pedestrian phases 2 and 6 operate whenever their adjacent vehicular phases operates, whether there is a call to pedestrian call to cross the intersection or not,

while phases 4 and 8 only work when someone taps the push button to cross (a call). Therefore, we can see more dots on the graphs for phases 2 and 6 than those of phases 4 and 8.

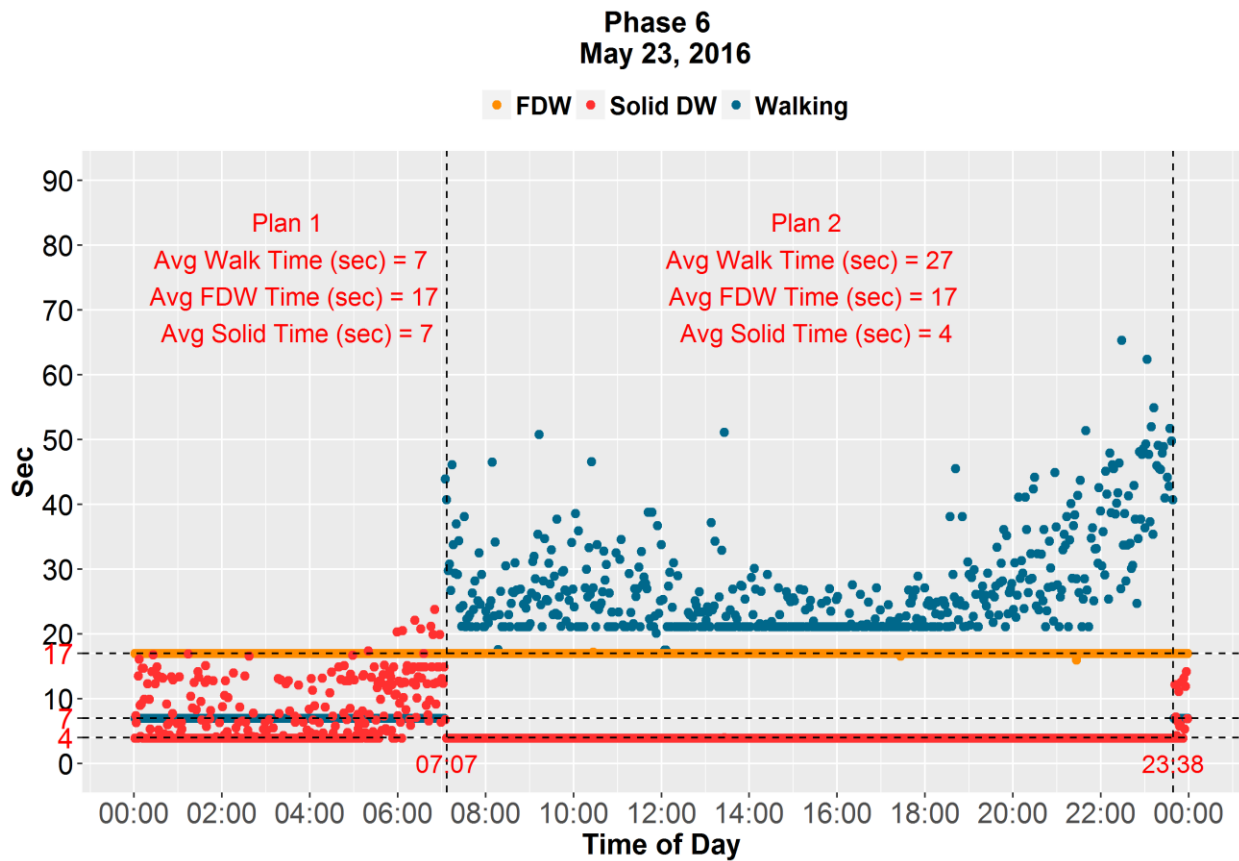


Figure 5-15: Time of pedestrian signal indication for phase 6

Phase 8  
May 23, 2016

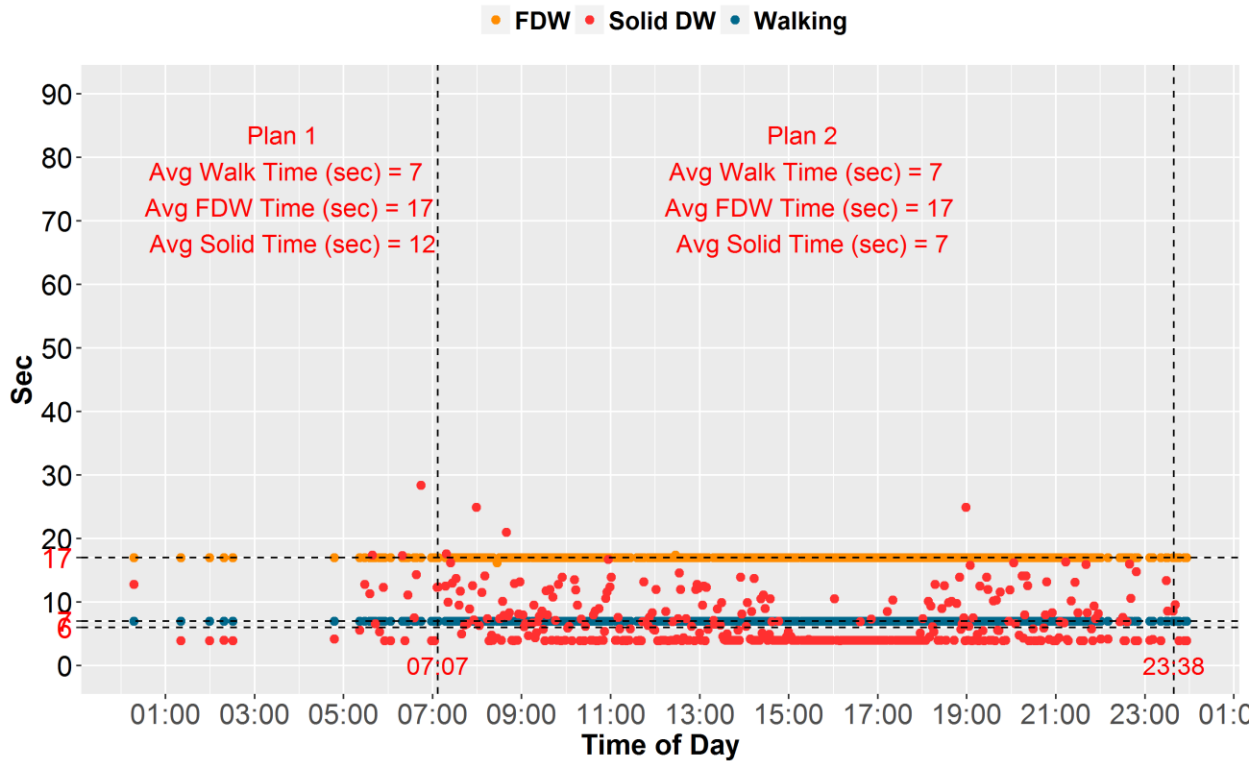
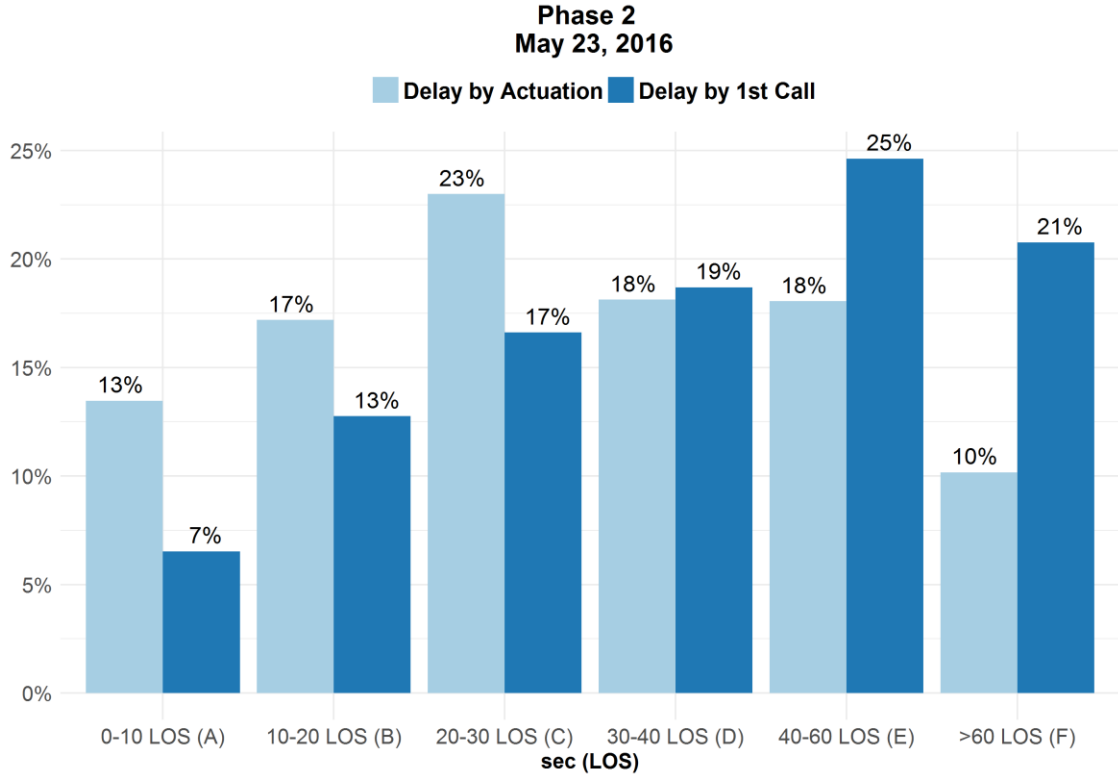


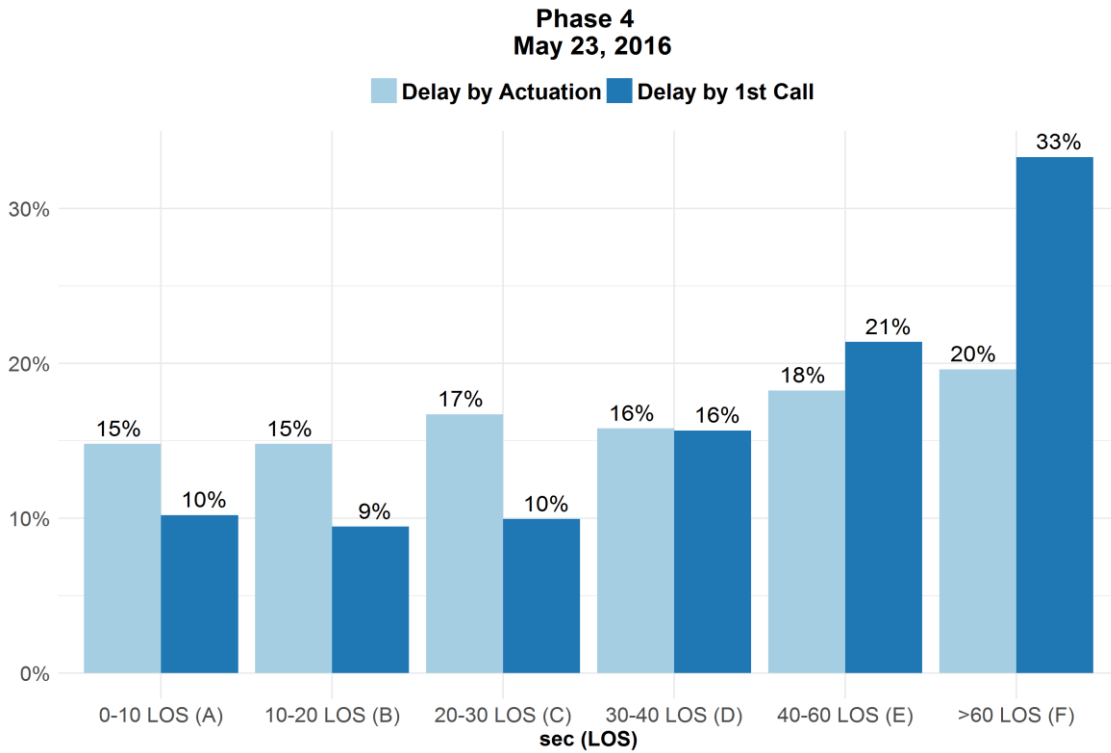
Figure 5-16: Time of pedestrian signal indication for phase 8

5.1.5 *Pedestrian Delay by Actuation*

Every time a pedestrian hits the push button to cross the intersection, a pedestrian actuation is registered in a controller, coded with the number 90. Based on the timestamp of every pedestrian actuation, the delay is calculated accordingly. This delay adds one point of non-accuracy, which occurs when a pedestrian taps the pushbutton more than once when attempting to cross the intersection. Thus, the delay calculation based on pedestrian actuation does not represent the real delay experienced by every pedestrian.



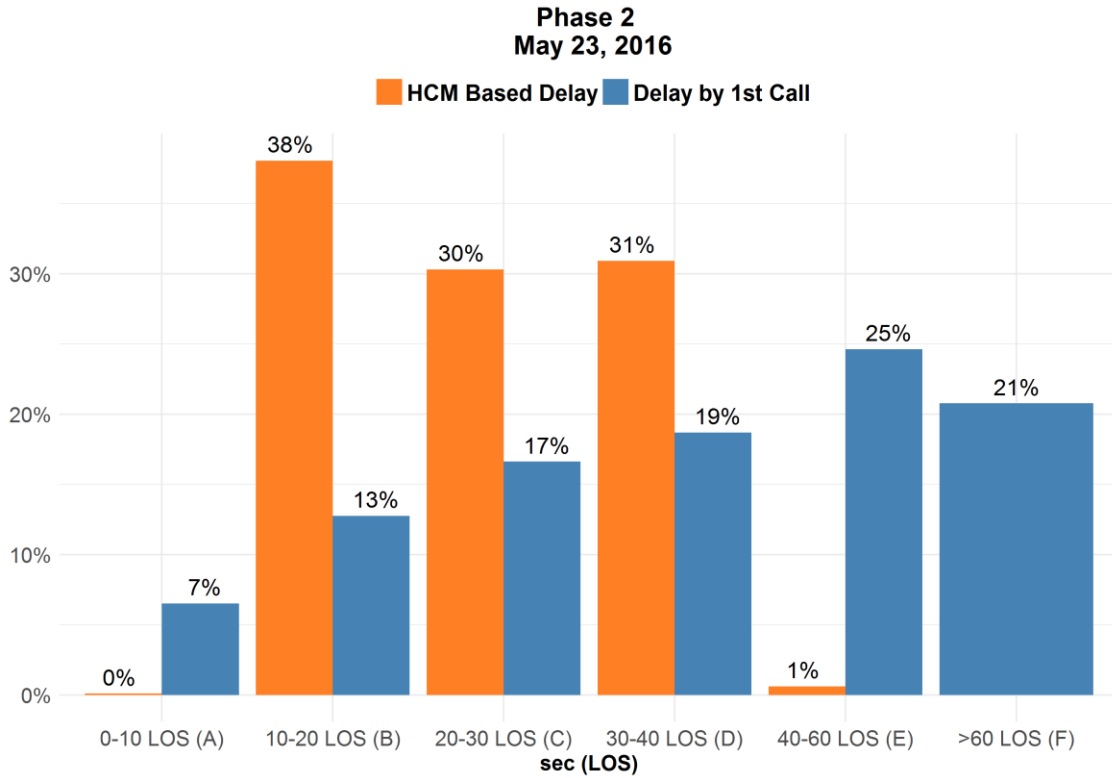
*Figure 5-17: Level of Service of pedestrian delay by actuation vs. delay by 1st call (phase 2)*



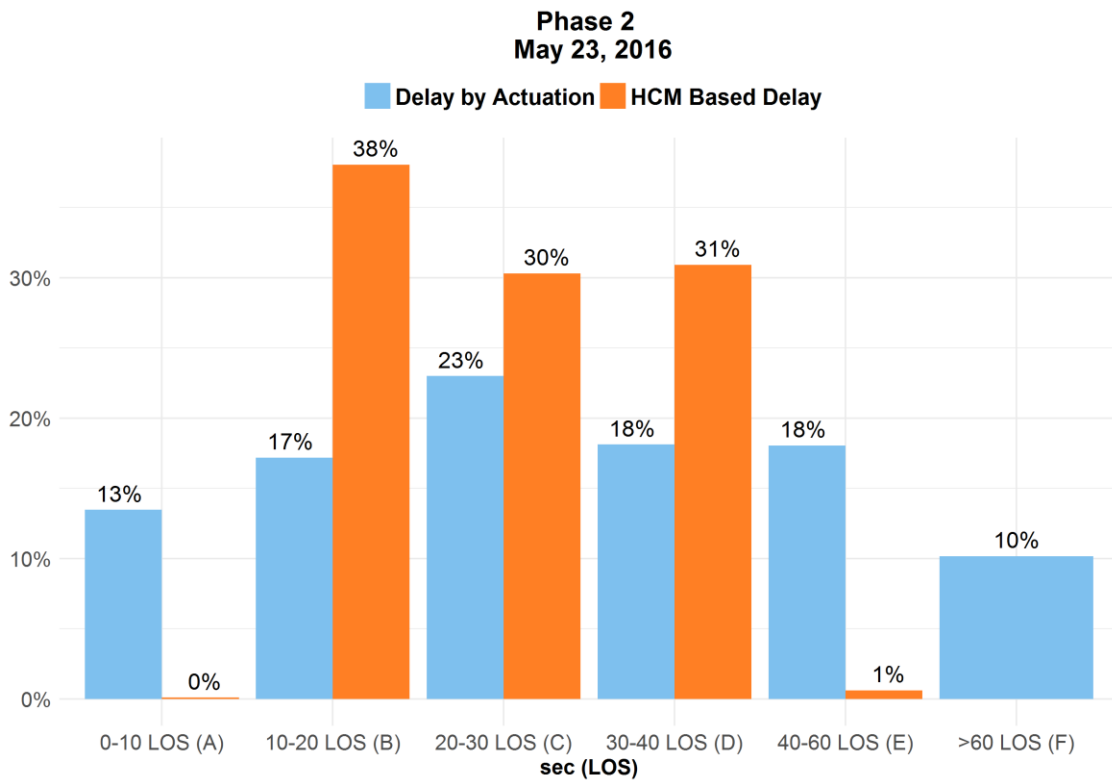
*Figure 5-18: Level of Service of pedestrian delay by actuation vs. delay by 1st call (phase 4)*

Figures 5-17 and 5-18 show a comparison between the ranges of delay with associated LOS. These are determined by comparing the first call registered by pedestrians to cross the intersection phase versus the delay calculation based on pedestrian actuation in phases 2 and 4. It is clear that the delay is less when calculated based on pedestrian actuation. For example, the range of delays within LOS F and LOS E have been reduced by about 12% and 7% for phase 2, and 13% and 3% for phase 4, when the delay is determined based on pedestrian actuation and the first pedestrian call. Additionally, delays that meet the criteria of LOS A, B, and C seem to be reduced for both phases by an average of 5%. The probable reasons that calculating delays based on pedestrian actuations express lower values are likely due to: 1) Additional actuations by the same person after the pushbutton was originally pressed when they arrived to the intersection and 2) other actuations registered by pedestrians arriving at the intersection after the first person's arrival and call registration to cross the intersection. These values represent a real delay experienced by pedestrians arriving after the first call was registered. Also, these lower values could be a combination of the previous two points.

Delay determined by the HCM method is compared to delay by actuation and delay based on first call, see figure 5-20 and 5-21. Both results from delay by actuation and delay from first call with associated percentage of LOS are different than that of the estimated delay by the HCM method. However, the percentages of delays by actuation that are within LOS B, C, D are closer to the percentages of HCM-based delay than the percentages of delay from the first call to the HCM based delay.



**Figure 5-19: Level of Service of pedestrian delay by 1st call vs. HCM method based delay (phase2)**

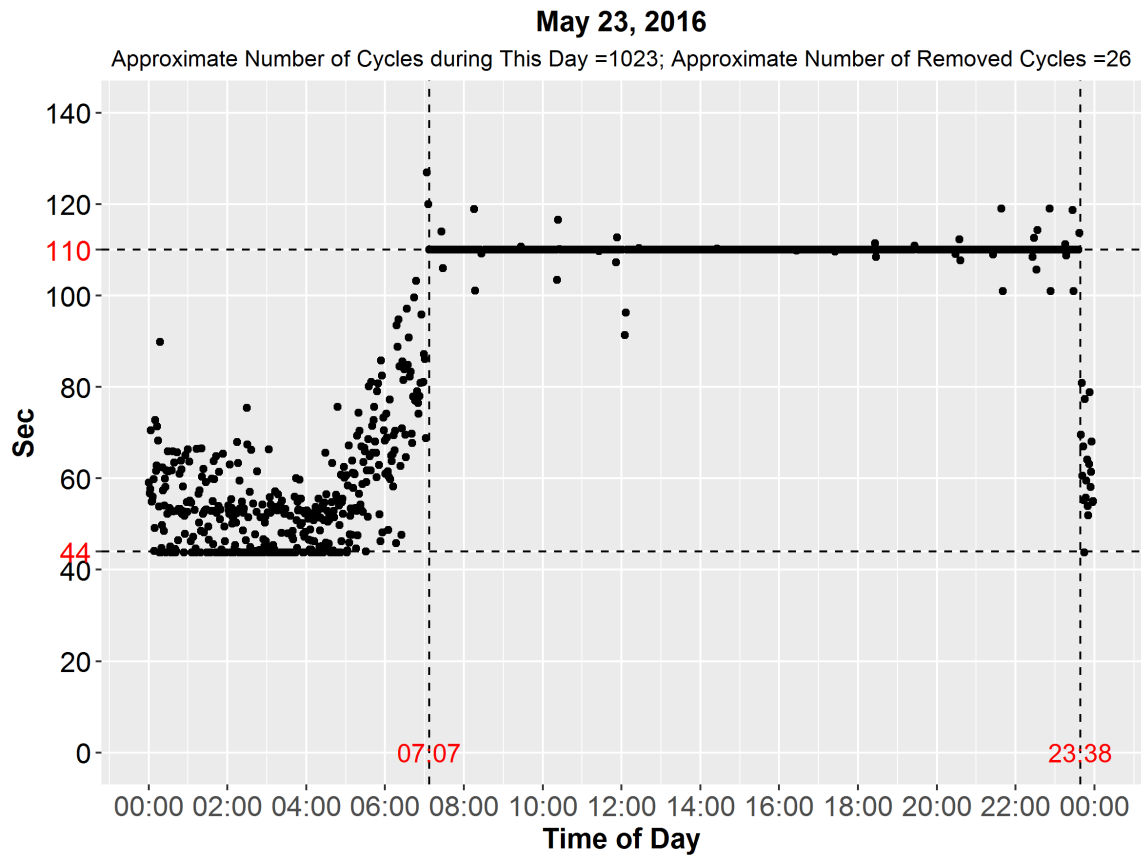


**Figure 5-20: Level of Service of pedestrian delay by actuation vs. HCM method based delay (phase 2)**

## 5.2 Vehicular Performance Measures

### 5.2.1 Cycle Length and Plans of Day

The study intersection operates with two plans during the day. The first runs free or uncoordinated; signal time assignments operate separately from other signals (FHWA, 2005). Plan 1 starts from the beginning of the day until 7:00 am and then restarts again around 23:30 (see figure 5-21).



*Figure 5-21: Effective cycle length*

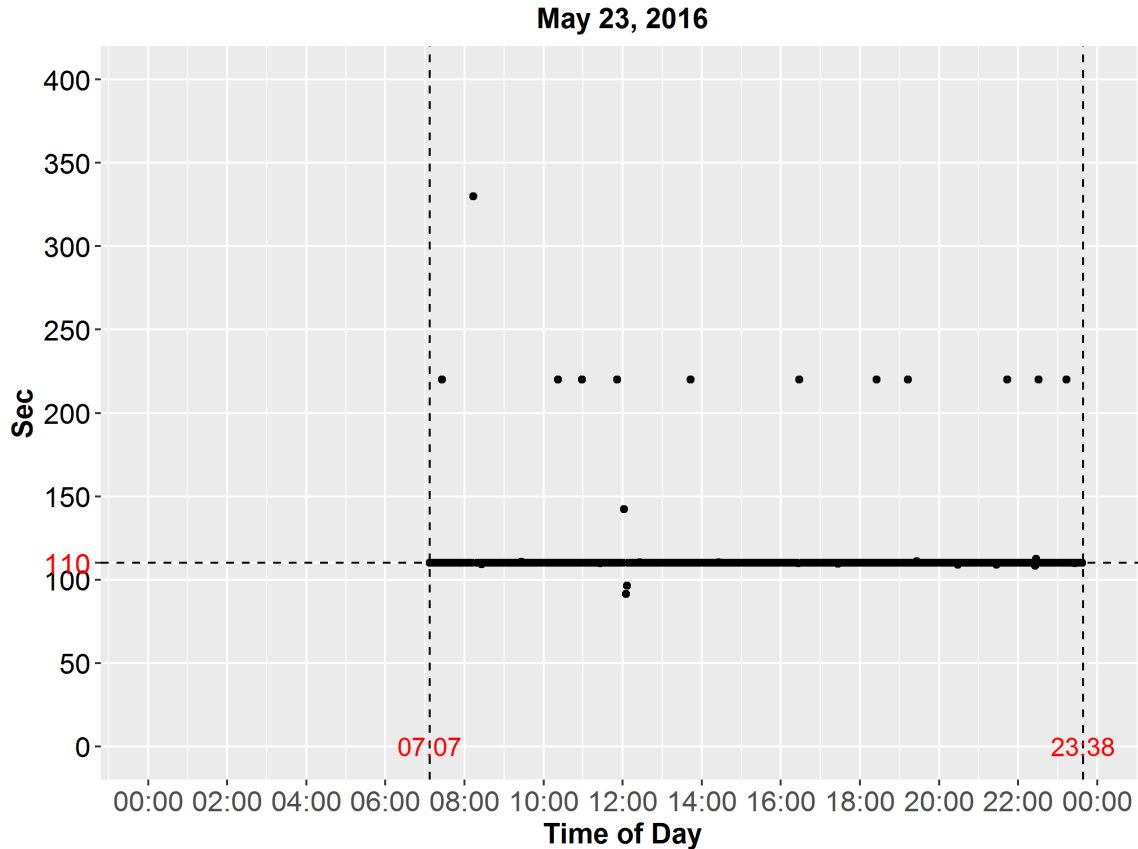
It is clear that the cycle length varies during plan one, when the controller runs free (uncoordinated) because the green time obtained by each phase varies based on vehicle actuations. However, the green time that each phase operates with is governed by the values of the minimum

and maximum green time and vehicle extension time, assigned to each phase. Furthermore, it is obvious that the minimum cycle length during plan 1 is 44 seconds, which occurs when phases (2, 6, 4, and 8) use their minimum green values and when left-turn movements are omitted. In addition, it is clear from figure 5-20 that the cycle length is relatively low between 00:00 and 5:00, which resulted from the low number of vehicle actuations on all phases, and consequently, a lower green time. However, as the number of vehicles increases, it can be seen that the cycle length values gradually increases from around 5:00pm until the beginning of the second plan.

On the other hand, the coordination plan (plan 2) runs from around 7:00pm up to 23:30. During this plan, the intersection operated on full actuation, when the actuations are featured on all the phases of the intersection (FHWA, 2010). Hence, all the phases can register calls and be terminated in accordance based on the controller activity. It is clear that there is a nearly consistent cycle length of 110 seconds along that plan. The effective cycle length for plan 2 is very similar to the background cycle length on figure 5-22, which is calculated based on yield point.

There are a few points that deviated from the straight line of a 110 second cycle length, in both the background and plan 2 effective cycle length. Regarding the background cycle length, the points that deviate outside the 110 seconds are due to the transitioning of signal patterns taking place in the controller (Day et al., 2014).





*Figure 5-22: Background cycle length*

The deviated measures are shown as a result of the actuations in the coordinated phase. This implies that the controller might be operated in the fully actuated mode during plan 2. These normally show when there are less vehicles actuations (off peak hours). Likewise, the patterns mentioned above regarding plans, operation, etc., are found to be the same during the days for which there are available data.

It is worth noting that (26) cycles are not shown in the figure 5-20 on May 23, 2016. The reasons these cycles have been removed is due to missing rows, which hindered the cycle length determination. For example, there are a few spots in the data where the beginning or end of the cycle is missing, and the cycle length cannot be calculated.

### 5.2.2 *Green Time & Green/Cycle (g/C) Ratio*

Figure 5-23 shows the effective green time of the phases operating at the intersection, sorted based on the sequence of operation from right to left, also see figure 4-1. The blue dots are the green time values and the black line is the 10-point moving average of the values.

It is clear that coordinated phases (phase 2 and 6, on SE Division street) obtained the highest share of green time, followed by roads (phases 4 and 8, on SE 122<sup>nd</sup> street). The left turn phases acquired the least amount of green time. Phases 2 and 6 always receive a minimum particular value of green time or more. The minimum values of the two phases can be seen as straight lines. To explain in more detail, on May 23, 2016, the minimum effective green time received by phases 2 and 6 was 24 seconds for plan one. The values received on plan 2 were 36 seconds during phase 2, and 38 seconds for phase 6, see figure 5-24. The 2-second difference between the minimum values of phases 2 and 6 was compensated with the maximum values of the effective green time of phases 1 and 5.

The maximum effective green time for phases 1 and 5 were 16 seconds and 14 seconds, see figure 5-25. As in phases 2 and 6, the maximum effective green values appear as a straight line on the graph. The reason that phases 1 and 5 received more green time than their maximum values is because they obtained additional unused green time from phases 4 and 8 when they are in low demand. The same concept is applied for all phases of the same ring (Ring 1 contain phases 1, 2, 3, and 4; Ring 2 contain phases 5, 6, 7, and 8) when phases received unused time or time resulted from omitting a phase. The only phases that do not receive green time are phases that follow coordinated phases 3 and 7. The highest demand is around 16:00 for all the phases at the intersection, where most phases did not receive extra time over their force-off points.

May 23, 2016

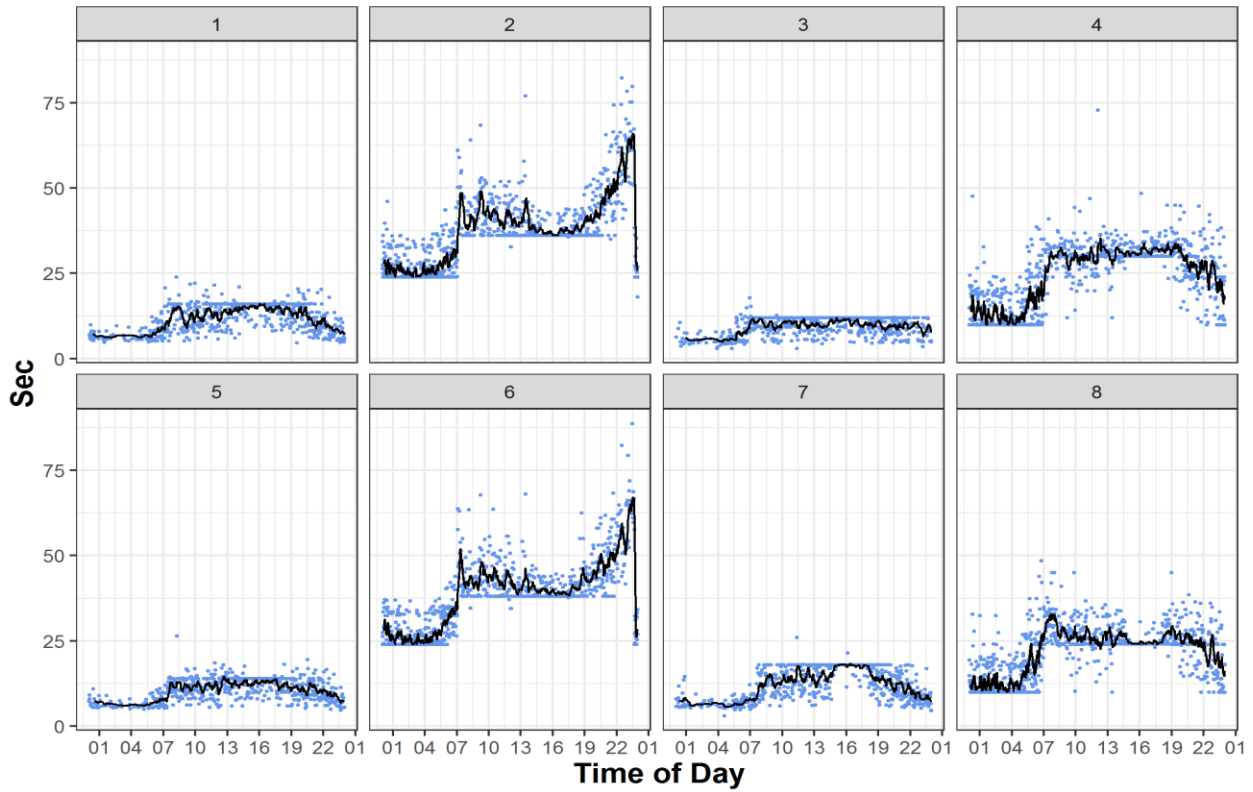


Figure 5-23: Effective green time of all phases operating at SE Division and 122nd

May 23, 2016

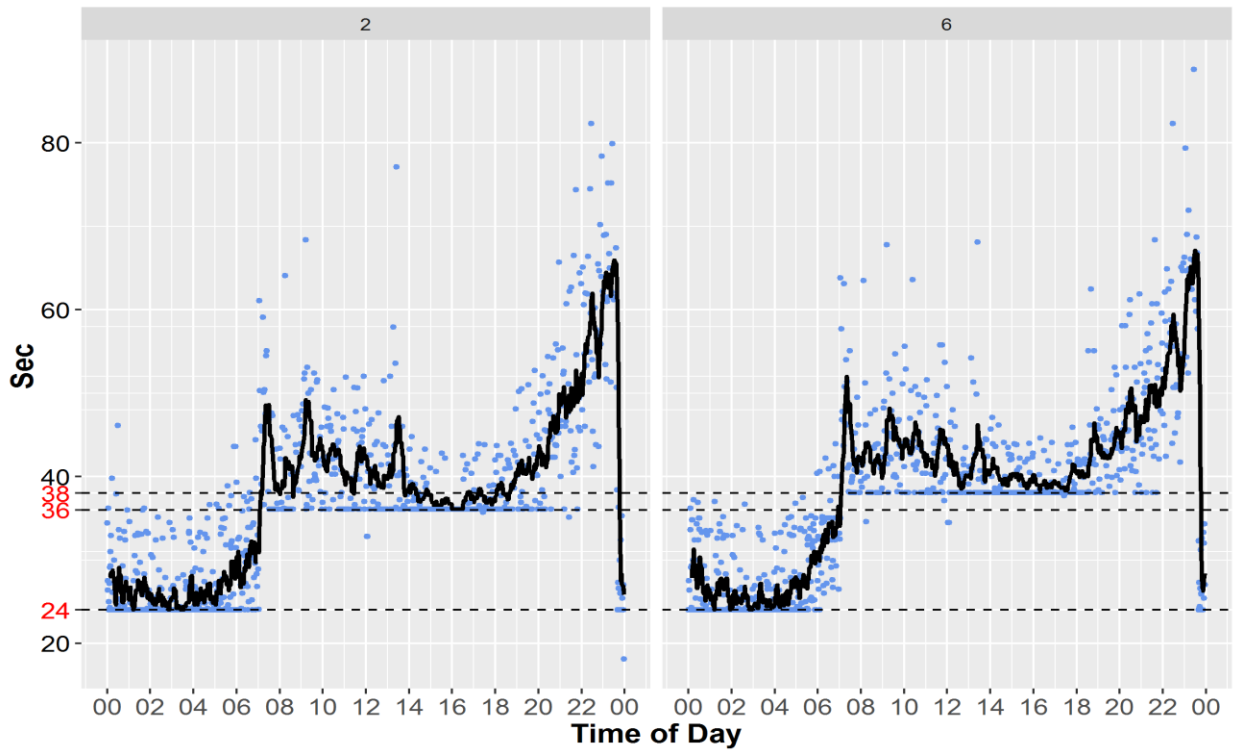
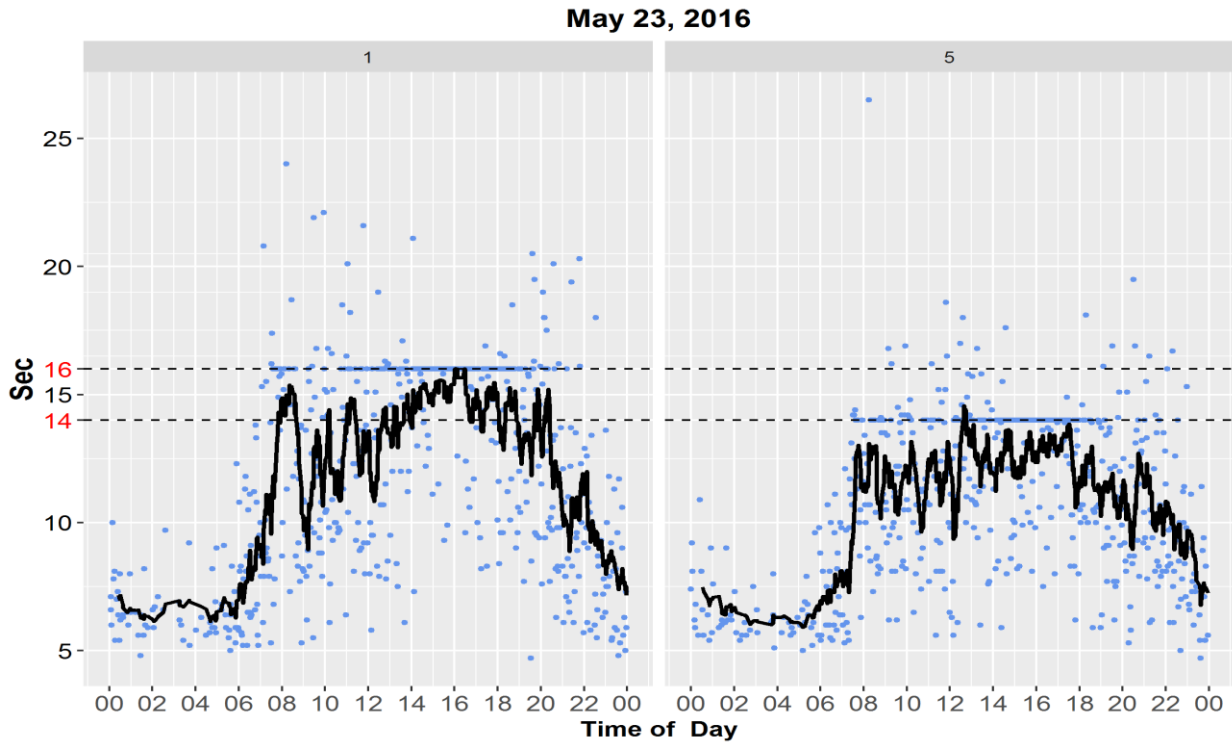


Figure 5-24: Effective green time of phases 2 and 6 SE Division and 122nd



*Figure 5-25: Effective green time of phases 1 and 5 SE Division and 122nd*

Figure 5-26 shows the green over cycle ratio (g/C) for all the phases. As mentioned earlier, the coordinated phases (2 and 6) share the largest proportion of green time related to cycle length.

While the green time values of the coordinated phases during plan 2 are noticeably larger than those for plan 1, the proportion of the green time to cycle length are greater in plan 1 than 2. To illustrate, during plan 1, the most frequent g/C ratio for phases 2 and 6 were 0.55, and 0.23 for phases 4 and 8. The most common g/c ratio during plan 2 was 0.33 for phase 2, 0.35 for phase 6, 0.27 for phase 4, and 0.22 for phase 8. Finally, the most frequent values for left turns during phase 1, phase 5, phase 3, and phase 7 were 0.15, 0.13, 0.11, and 0.16, respectively.

However, the summation of the most frequent the g/C mentioned in one of the two rings will not equal 1 because the cycle length includes the clearance interval (yellow change and all red) for every phase. In the study intersection, values for the yellow change and all red for

through' and 'right turn' movements are 4 seconds and 1 second respectively, while the yellow change for a left turn is 3 seconds and zero seconds for all red.

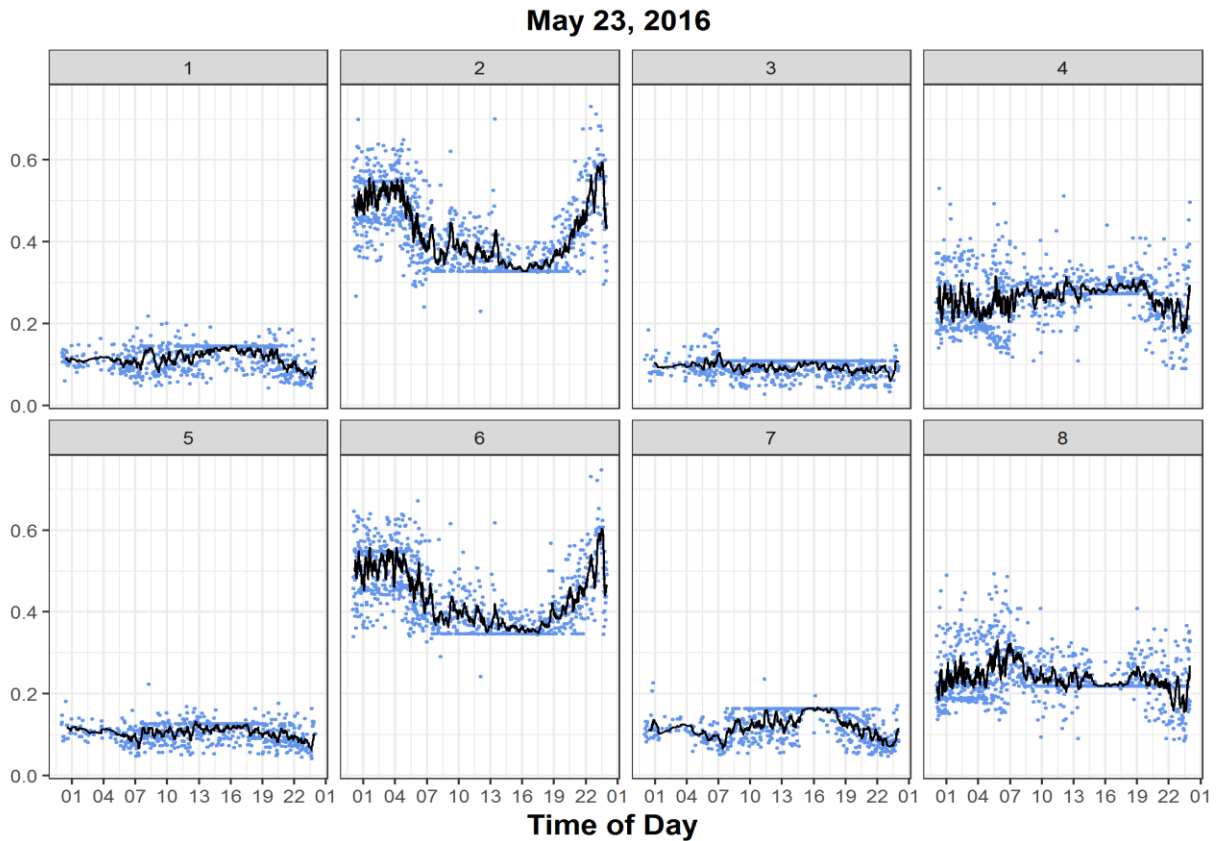


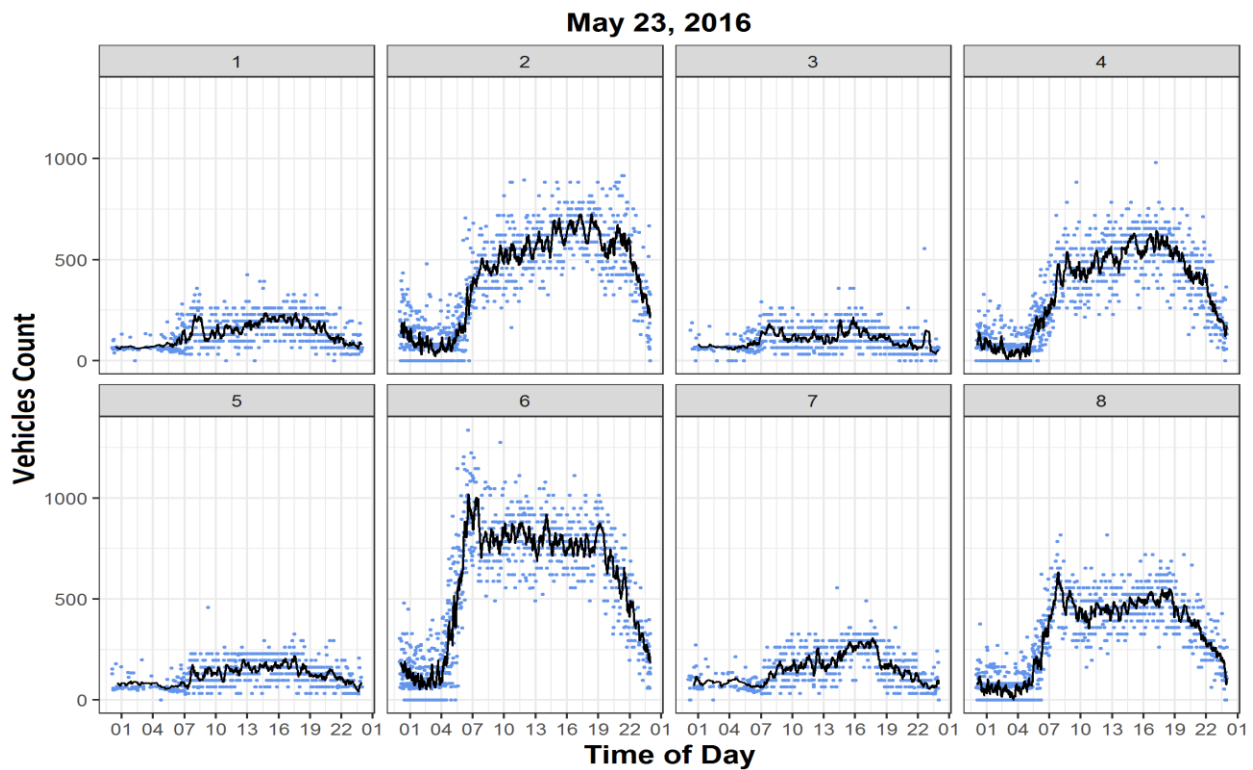
Figure 5-26: Green to Cycle (g/C) ratio of all phases operating SE Division and 122nd

### 5.2.3 Vehicle Count

Figure 5-27 shows the number of vehicles serviced in every counting period of the cycle for May 23, 2016. The counting period, as defined earlier in section 4, is the period from the end of the green time of a particular phase for a previous service instance, to the end of the green time of the following service instance.

The number of vehicles was determined based on the number of detected actuations, which happens when a vehicle passes over a detector. With regard to the number of actuations during phases 2, 4, 6, and 8, they only represent the through movement vehicle count, because the intersection is not provided with a right turn detector.

However, what can be seen from the figure above is that the number of vehicles approaching the intersection started very slow at the beginning of the day and had a gradual increase from 4pm-7pm. Following this, the number of vehicles fluctuated until 19:00 when the vehicles gradually decreased. In figure (5-27), the peak hours are not prominently seen in all phases. However, it can be seen that a larger number of vehicles passed the intersection around 7:00 during phases 6 and 8. Likewise, there were slightly more vehicles between 16:00 and 19:00 on the through lane movement during phases 4 and 8.



*Figure 5-27: vehicle count for left turn phases and through movement of phases 2, 4, 6, and 8 at SE Division and 122nd*

Figures 5-28 and 5-29 show a comparison between the number of vehicle actuation phases, including the through movement and left turn phases on May 23, 2016. It is clear that there are more vehicles traveling via the through lanes of phase 6, westbound (SE Division Street), followed by phase 2 (eastbound) coordinated phases, and finally phases 4 and 8. From 6:00 to 9:00, the

highest number of vehicles traveled via through lanes of phase 6, while 4:00 to 6:00 was the highest for phases 2 and 4.

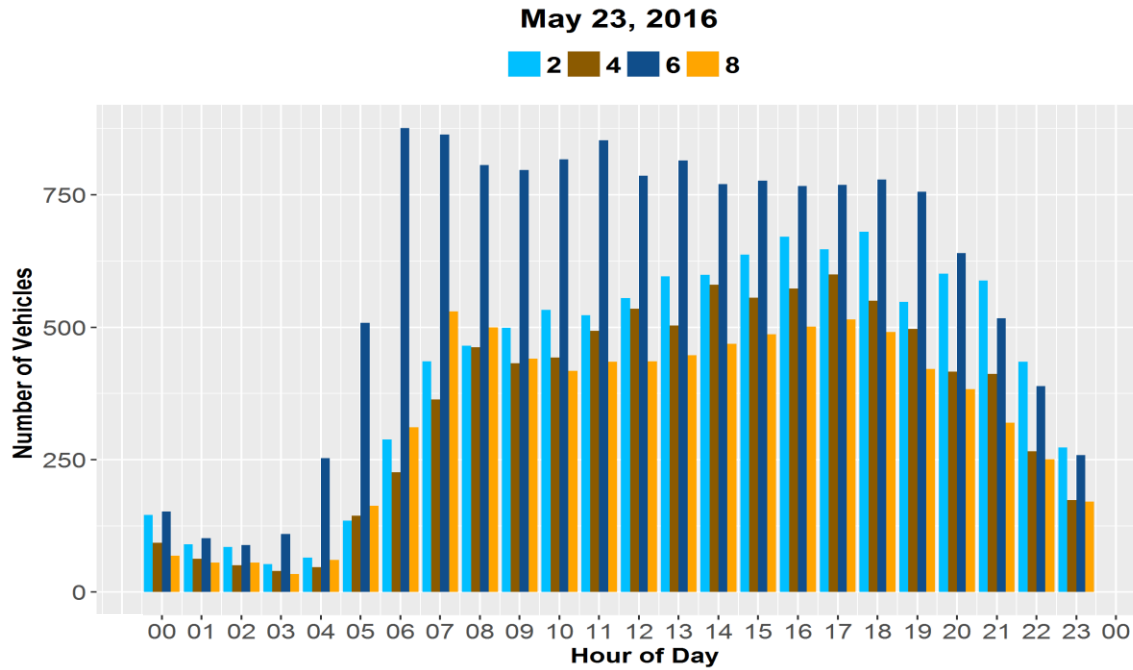


Figure 5-28: vehicle count for through lane movements of each hour of day.

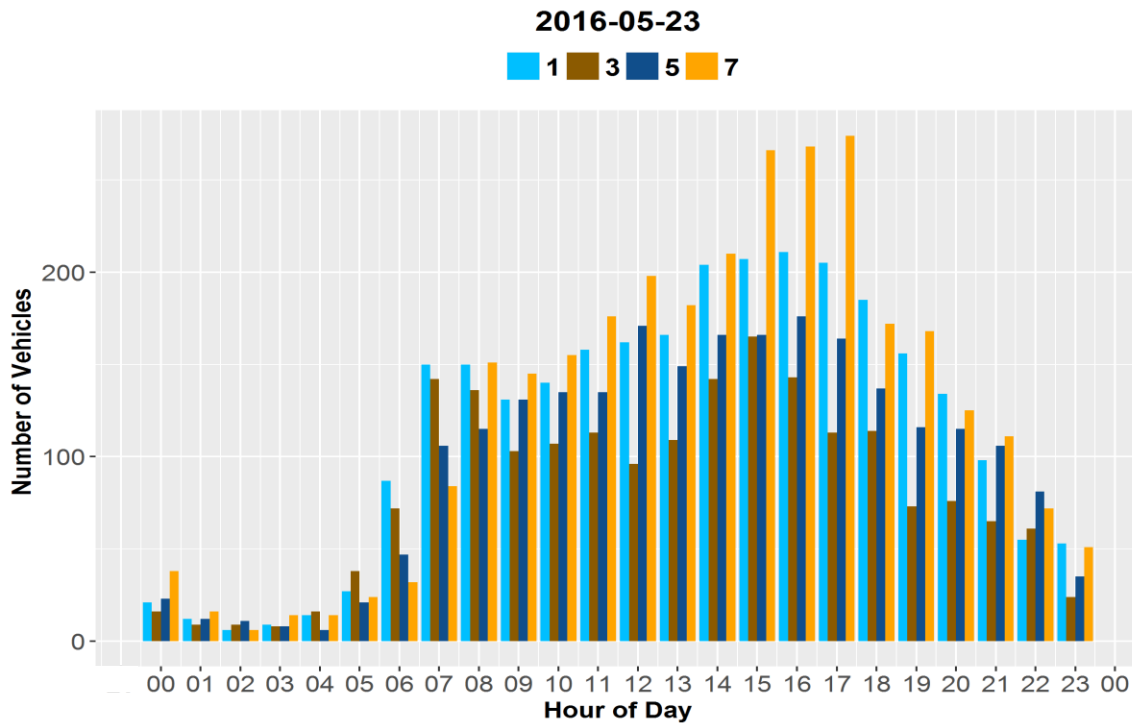


Figure 5-29: vehicle count for left turn movements of each hour of day.

Regarding left turn phases, it seems that phase 1 (westbound), and phase 7 (southbound) had more demand than phases 3 (northbound) and 5 (eastbound). Concerning the highest number of left turn vehicles, a greater number used the westbound left turn lane from 15:00 to 16:00 (phase 1), while the highest number of vehicles actuations was from 17:00 to 18:00 for phase 7.

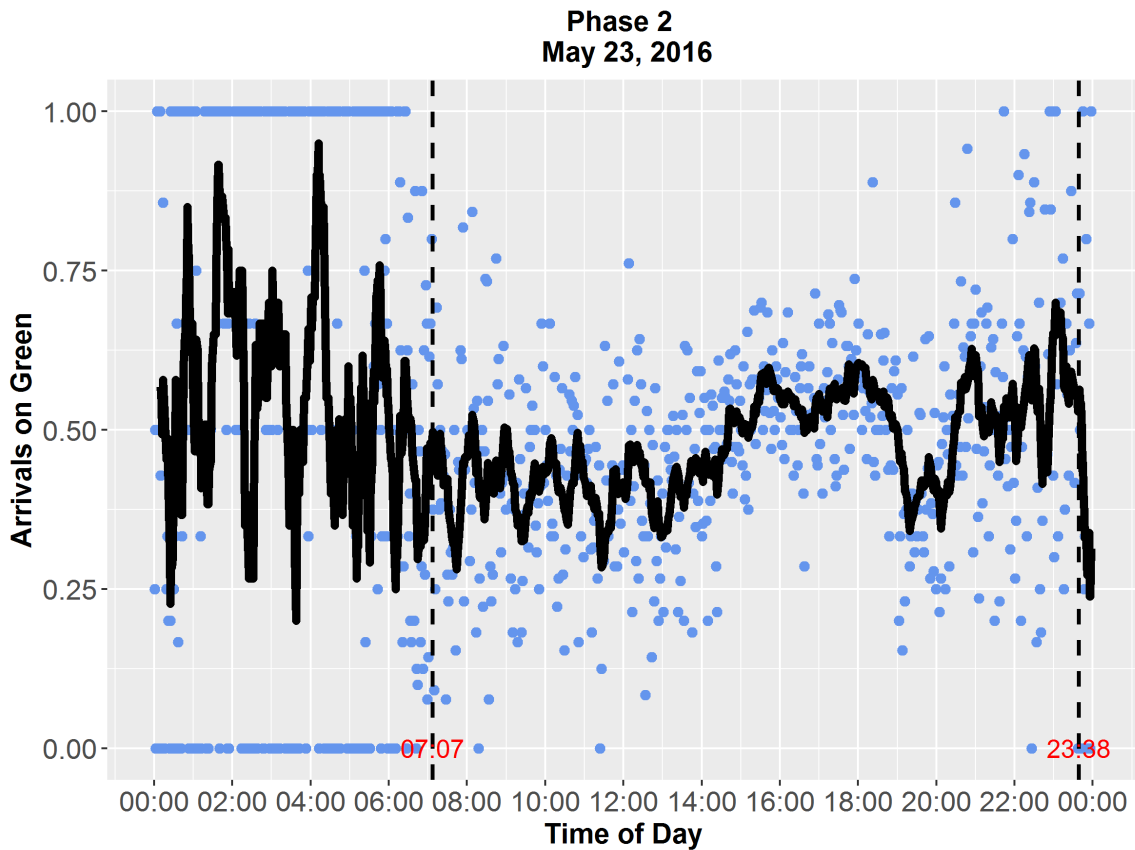
#### 5.2.4 *Proportion of Vehicles Arrived During Green Indication & Arrival Type (AT)*

Figures 5-30 and 5-31 show the proportion of the vehicles that arrived to the intersection during the green signal indication, for the through movements of coordinated phases 2 and 6 during May 23, 2016. In these figures, the arrivals on green are represented by the blue dots, and the black line is a 10-point moving average, which expresses the trend of the actual values. In both figures for plan 1, platoons arrived on green (arrivals on green value=1) or arrived on red (value=zero). This is attributed to the low number of vehicles approaching the intersection during that period of time. During plan 2, for both the coordinated phases, the proportion of the arrivals on green were fluctuating around 0.5. For example, during phase 2, the percentage of arrivals on green was slightly higher between 16:00 and 18:00 than the preceding hours in plan 2. Likewise, in phase 6, the proportion was higher around 7:00.

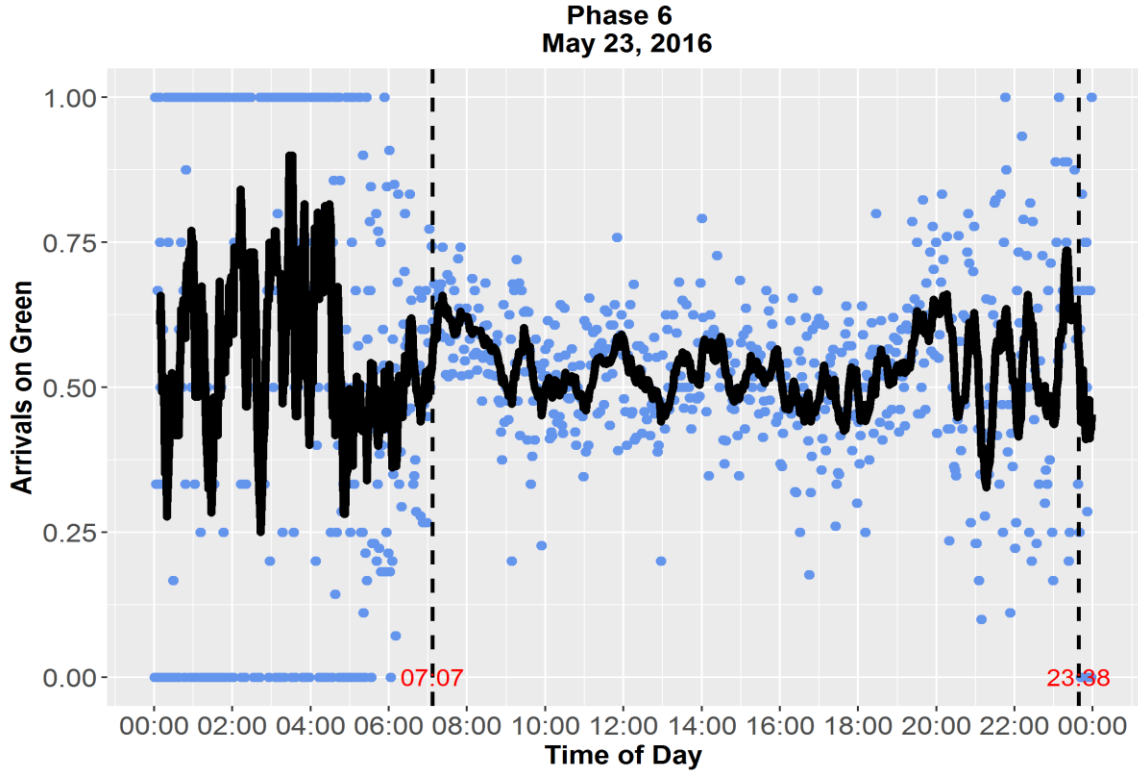
In order to put the above results in a qualitative measure, Arrival Type (AT) can be put into place. Arrival Type considers the amount of the green time relative to the cycle length (g/C ratio). Figures 5-32 and figure 5-33 show the AT of the coordinated phases on May 23, 2016. For phase 2, plan 2, from 7:00 to 15:00, the moving average line was fluctuating around 3 and 4, which are considered random arrivals and more favorable according to the HCM (Exhibit 15-4), see table 4-31. For the same phase, AT is the highest between 16:00 to 18:00, which is around 5 (Highly favorable). Continuing with phase 2, although the moving average of proportion of vehicles arrived on green between 20:00 to 23:00 looks larger than that in hours between 7:00 to



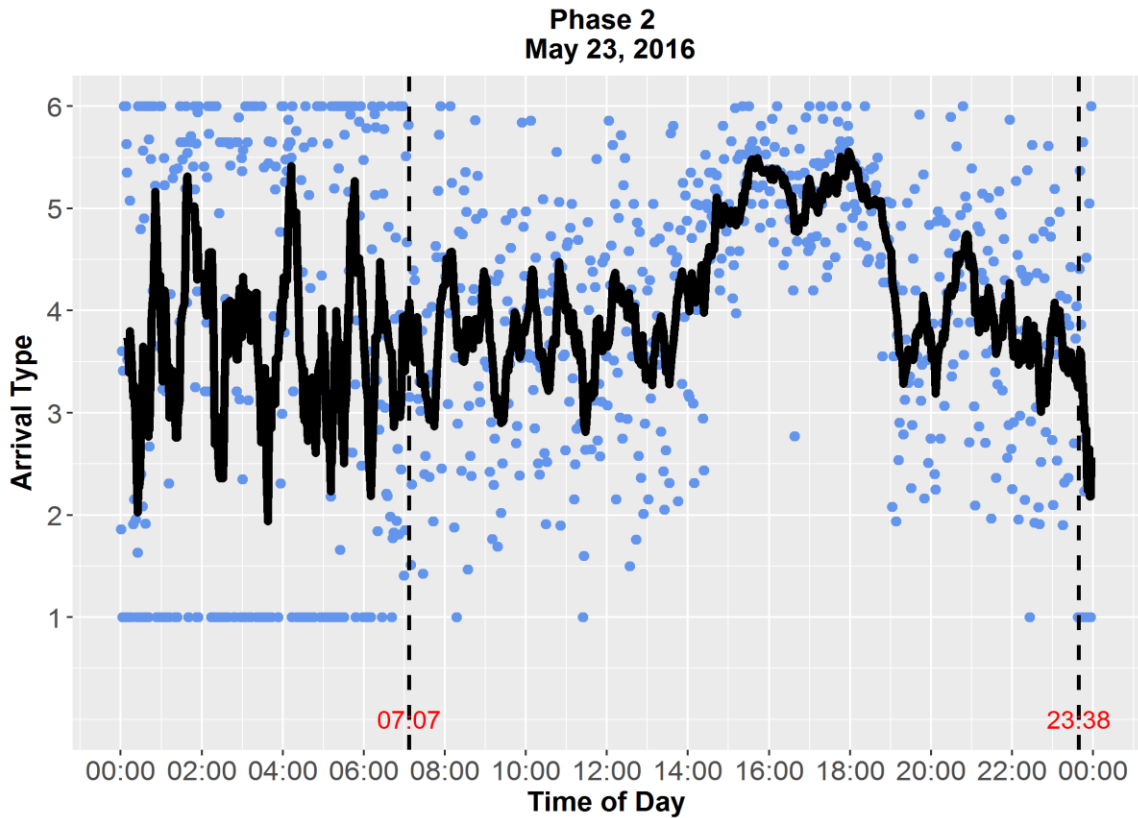
16:00, the moving average of AT fluctuates between 3 and 4 for both periods due to the effect of g/C. During phase 6, plan 2, from 7:00 and 20:00, the moving average of AT ranged between 4 and 5, meaning that based on the HCM manual, that quality of the progression is either “favorable” or “highly favorable”. After these hours, AT slightly dropped. To conclude, phase 6 had a slightly larger progression than phase 2.



*Figure 5-30: Proportion of vehicles arrived during green indication of Phase 2 (eastbound)*



*Figure 5-31: Proportion of vehicles arrived during green indication of Phase 6 (westbound)*



*Figure 5-32: Arrival Type (AT) of phase 2 (eastbound)*

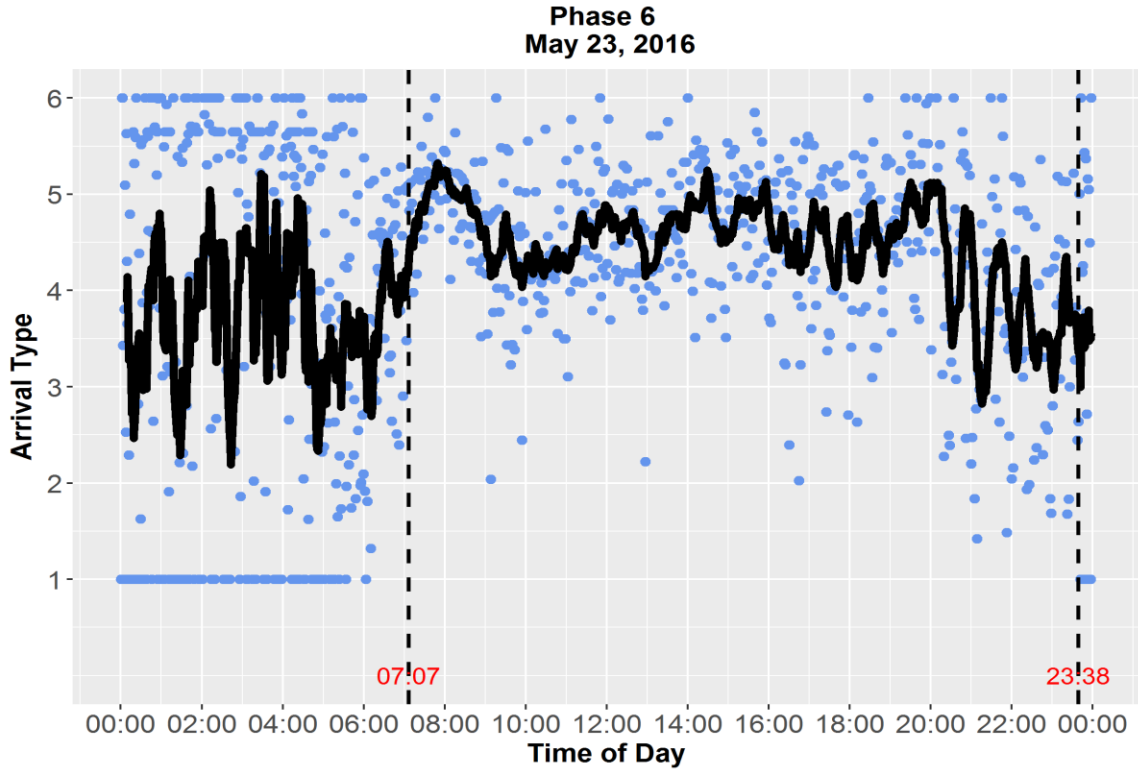


Figure 5-33: Arrival Type (AT) of phase 6 (westbound)

### 5.2.5 *Purdue Coordination Diagram*

Figure 5-34 and 5-35 show a Purdue Coordination Diagram (PCD) of the coordinated phases 2 and 6. The green line is the beginning of the green indication and the red line is the end of the green. Thus, black dots above the green line represent the vehicles that arrived during the green time, while dots below the green line show the vehicles arrived during red time. Although, PCD is normally used to assess the coordination and offset of an arterial, PCD can be helpful to visualize arrivals on green and arrivals on red. Moreover, it is applicable to determining the hours of day when it contains the highest demand at the intersection.

Therefore, based on the PCD figures of the coordinated phases (2 & 6) below, during plan 2, it is obvious that there are many more vehicles arriving during the green time than those arriving during red indication. That was inferred from the associated figures by noticing macroscopically the high number of black dots above the green line compared to those below.

Moreover, proportion peaks for vehicles arriving during the green indication on figures 5-29 and 5-30 of phases 2 and 6, can be clearly seen on PCD figures 5-34 and 5-35

Further, as was noticed before, it is clear that the highest number of vehicles using the through lanes of phase 2 occurred between the hours of 16:00 to 18:00, and around 7:00 regarding phase 6.

To conclude, because there were more number of vehicles arrived during green than red indication, the coordination plan at intersection SE Division Street and 122<sup>nd</sup> Avenue is considered effective.

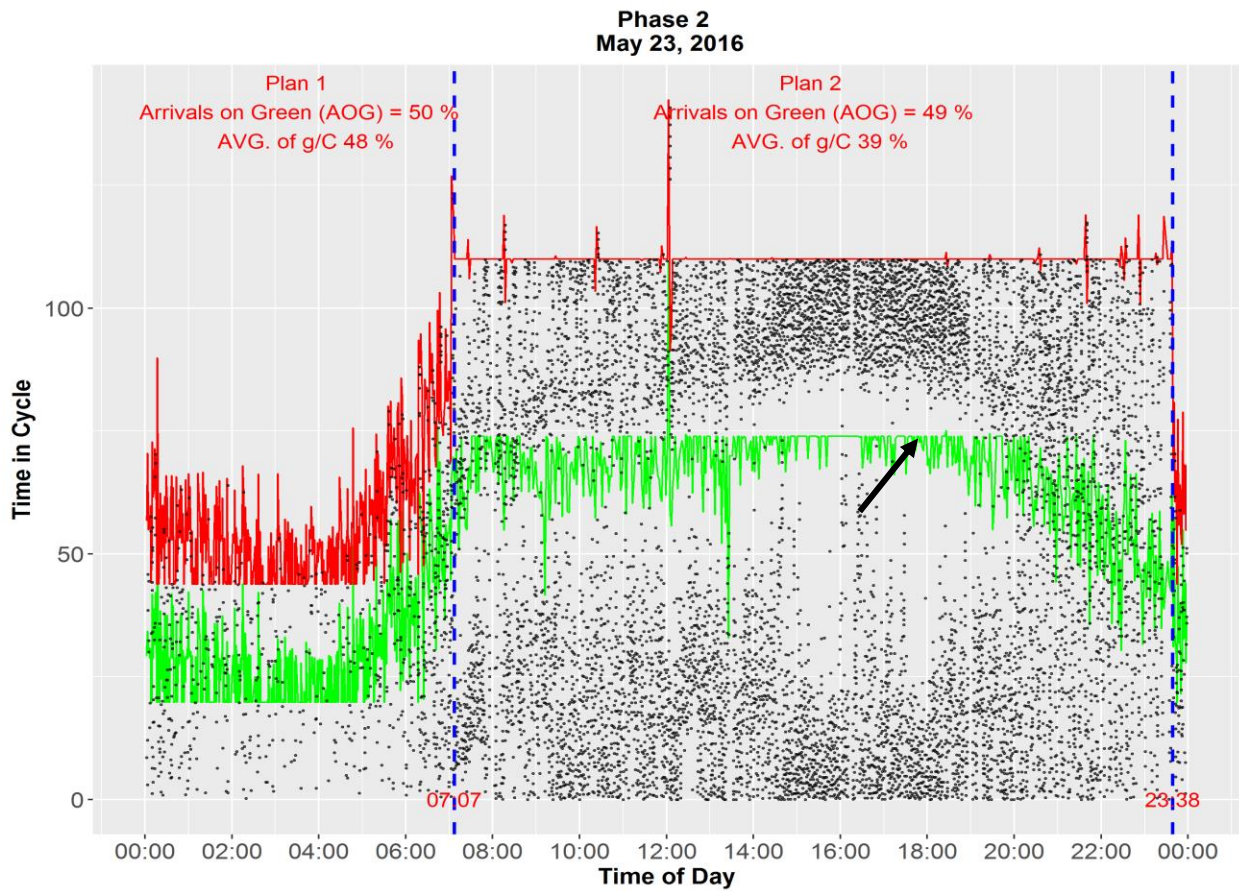
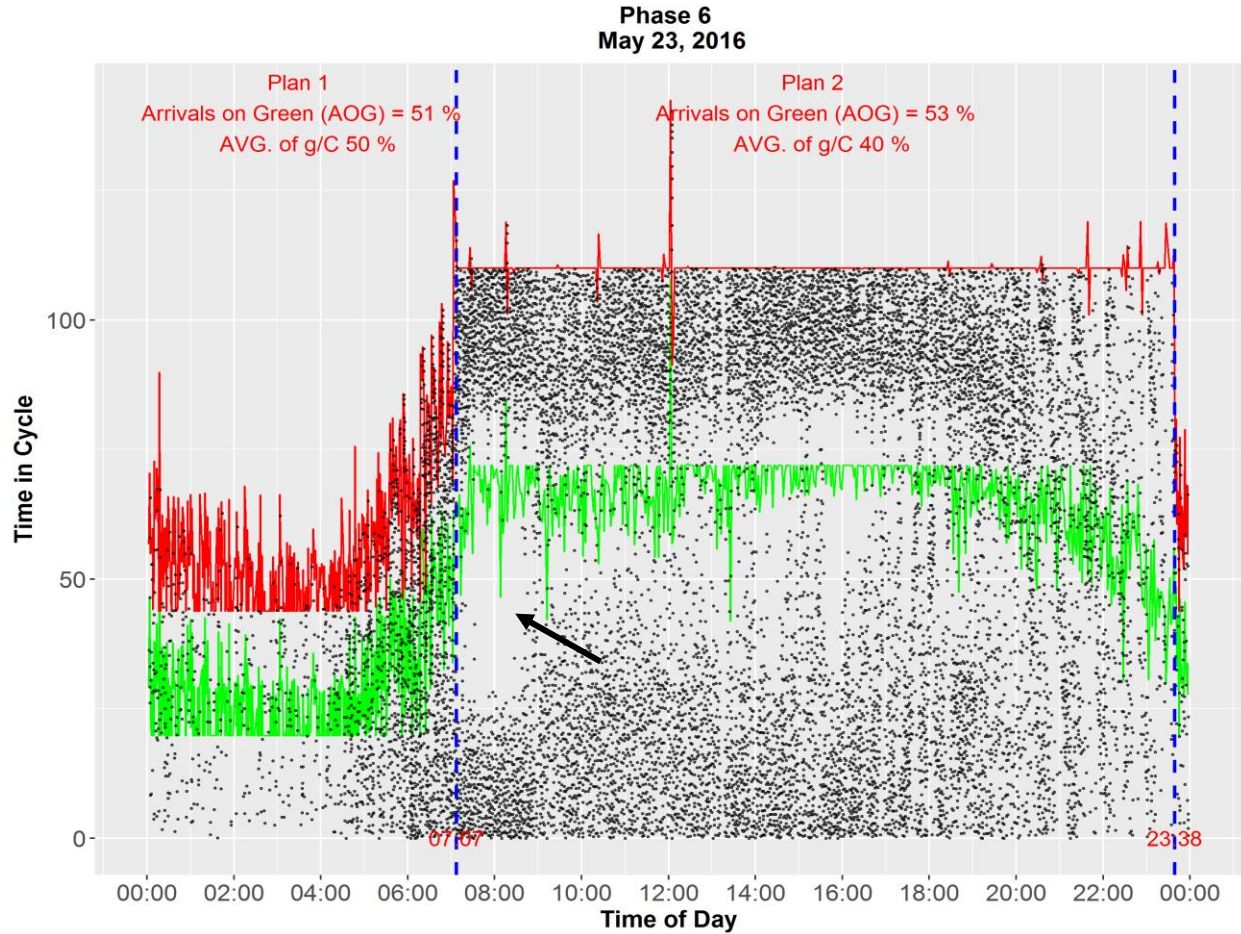


Figure 5-34: Purdue Coordination Diagram (PCD) of through movement of phase 2



*Figure 5-35: Purdue Coordination Diagram (PCD) of through movement of phase 6*

### 5.3 Accuracy of results

All the performance measures and explanatory variables determined in this report were subject to a certain level of inaccuracy due to rows missing data that directly contributes to the calculations of these performance measures. However, the number of missing rows are minimal compared to the overall number of rows that took part in the final results. Moreover, the number of errors can accumulate if the calculation approach of a particular measure is dependent on two or more variables containing their own errors. For example, the g/C ratio is dependent on green time values and cycle length values. Thus, the number of values removed from g/C results are directly related to the number of errors removed in green time and cycle length.

On the other hand, in addition to the level of inaccuracy mentioned above, performance measures including, Vehicle Count, Proportion of Vehicles Arrived on Green, and PCD have their own issues. Any imprecision relating to these measures concerns the location of detectors and their connection to the channels (output). Issues related to the Vehicle Counts can be summarized in the following points:

- The through vehicular movements at SE Division and 122<sup>nd</sup> avenue consist of two adjacent lanes. The detectors on two of the lanes are connected to one channel. This type of connection gives less accurate, but still acceptable results (Day et al., 2014).
- There are no detectors on the right turn lane at the intersection. Thus, vehicle counts at the intersection have only been performed on through and left-turn lanes.
- Results could potentially be affected by situations when vehicles pass over a detector in a certain movement and then deviate to another movement.

In addition to losing some accuracy due to the first two points mentioned above, Proportion of Vehicles arrived During Green, Arrival Type, and PCD results are affected by the location of the setback detectors. During hours of high demand, and when signal indication is red, vehicles might build up in queue and thus back detectors could be triggered for a certain period of time. During this time, vehicles could continue to approach the intersection, but no detection would occur. As a result, the proportion of vehicles arriving during green and red times is affected (Smaglik, Bullock, & Sharma, 2007). This situation can be clearly noticed on figures 5-33 and 5-44 where the arrows on graphs are pointed.

## 6.0 CONCLUSIONS

After exploring controller data at the intersection of SE Division and 122<sup>nd</sup> avenue, the following conclusions were drawn:

- Pedestrian's registering a first call to cross SE Division and 122<sup>nd</sup> in the north-south direction may experience an average of 9 seconds or more delay than those crossing in the east-west direction.
- In the east-west direction, around 60% of pedestrians registering a first call faced a waiting time of 30 seconds or more, whilst in the other direction, the percentage was around 72%. This implies that the aforementioned percentage fell within LOS D, E, and F based on HCM criteria.
- Pedestrian delays based on first calls were quite similar during weekdays and the weekend.
- In plan 2, the pedestrian delay exceeded that of plan 1 by an average of 13 seconds in the east-west direction. On the other hand, for north-south pedestrian movement, the average delay in plan 1 was 25 seconds less than plan 2.
- For east-west pedestrian movement, it was found that hours 14:00 and 17:00 always contain an average delay of 40 seconds or more, consistent among the five days of available data. On the other hand, in the north-south direction, it was found that hours 8:00, 10:00, and 18:00 always contain an average pedestrian delay of 50 seconds or more.
- More pedestrians crossed in the north-south direction than on the east-west. In addition, pedestrian demand at the intersection is less during weekend than weekdays.
- The hours between 13:00 and 18:00 exhibit the highest pedestrian demand.
- Estimated pedestrian delay as determined by the HCM approach showed different values than that calculated based on pedestrian calls.

- When pedestrian waiting time was calculated based on pedestrian actuation, the values were relatively smaller than those determined based on first calls. That is, delay values that lie within LOS D, E, and F dropped by an average of 18% when they were determined based on pedestrian actuation.
- This intersection operates on two plans: first, free-actuation, which starts from 00:00 and lasts to around 7:00, with a minimum cycle length of 44 seconds and second, fully coordinated actuation, which starts from 7:00 and lasts until 23:30, with a cycle length of 110 seconds.
- Phases 2 and 6 obtain the largest portion of the green time at the intersection. In fact, the phases' average green-to-cycle ratio was 0.55 in plan one and 0.34 in plan two, while the average g/C ratio of phases 4 and 8 was 0.23 in plan one and around 0.25 in plan two.
- All lanes of through and left turn movements are in high demand around 16:00, where most of the phases do not receive extra time over their force off points.
- The largest number of vehicles passing on lanes of through movement were due westbound, followed by eastbound movement on SE Division Street (both coordinated phases). Further, the peak period regarding phase 6 is around 7:00, whereas the peak period for phase 2 occurs around 16:00.
- Concerning the highest number of left turning vehicles, a higher number used the westbound left turn lane from 15:00 to 16:00 (phase 1), while the highest number of vehicles actuations was from 17:00 to 18:00 for phase 7.

Quality of the progression at the intersection is considered good. Based on HCM criteria, the coordination appraisal ranges from “favorable” to “highly favorable.”



## 7.0 REFERENCES

Brennan, T. M., C. M. Day, J. R. Sturdevant, E. Raamot, and D. M. Bullock. Track Clearance Performance Measures for Railroad Preempted Intersections. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 2192, Transportation Research Board of the National Academies, Washington, D.C., 2010, pp. 64–76.

City of Portland, Portland Bureau of Transportation. 2017, April 4. Retrieved from <https://www.portlandoregon.gov/transportation/article/634736>

Day, C. M. and D. Bullock. Arterial Performance Measures, Volume 1: Performance Based Management of Arterial Traffic Signal Systems. Final Report, NCHRP 3-79A (Vol. 1), 2010.

Day, C. M., E. J. Smaglik, D. M. Bullock, and J. R. Sturdevant. Quantitative Evaluation of Fully Actuated Versus Non-Actuated Coordinated Phases. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 2080, Transportation Research Board of the National Academies, Washington, D.C., 008, pp. 8–21.

Day, C. M., R. J. Haseman, H. Premachandra, T. M. Brennan, J. S. Wasson, J. R. Sturdevant, and D. M. Bullock. Evaluation of Arterial Signal Coordination: Methodologies for Visualizing High-Resolution Event Data and Measuring Travel Time. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 2192, Transportation Research Board of the National Academies, Washington, D.C., 2010, pp. 37–49.

Day, C. M., Sturdevant, J. R. and Bullock, D. M. 2010. Outcome-oriented performance measures for management of signalized arterial capacity. *Transportation Research Record: Journal of the Transportation Research Board*, 2192: 24–36.

Day, C.M., Ernst, J.M., Brennan, T.M., Chou Jr., C.S., Hainen, A.M., Remias, S.M., Nichols, A., Griggs, B.D., Bullock, D.M., 2012. Performance measures for adaptive signal control: case study of system-in-the-loop simulation. *Transport. Res. Rec.* 2311, 1–15.

Day, C.M., T.M.Brennan, A.M. Hainen, S.M. Remias, H. Premachandra, J. R. Sturdevant, G. Richards, J. S. Wasson, and D. M. Bullock. Reliability, Flexibility, and Environmental Impact of Alternative Objective Functions for Arterial Offset Optimization. In *Transportation Research Record: Journal of the Transportation Research Board*, No. 2259, Transportation Research Board of the National Academies, Washington, D.C., 2011, pp. 8–22.

Federal Highway Administration. *Manual of Uniform Traffic Control Devices*, 2009.

Fitzpatrick, K., Paul, C., Marcus, A.B., Mark, D. W., Shaw- Pin, M., 2003. Design Speed, Operating Speed, and Posted Speed Practices. TRB, National Research Council, Washington, D.C, Report for NCHRP Project 15- 18.

Highway Capacity Manual 2010, Transportation Research Board of the National Academies, Washington D.C.

Hubbard, S. M. L., D. M. Bullock, and C. M. Day. Integration of Real-Time Pedestrian Performance Measures into Existing Traffic Signal System Infrastructure. In Transportation Research Record: Journal of the Transportation Research Board, No. 2080, Transportation Research Board of the National Academies, Washington, D.C., 2008, pp. 37–47.

Koonce, P., L. Rodegerdts, K. Lee, S. Quayle, S. Beard, C. Braud, J. Bonneson, P. Tarnoff, and T. Urbanik. Traffic Signal Timing Manual. Report FHWA-HOP-08-024. Federal Highway Administration, Washington, D.C., 2008.

Koonce, P., Rodegerdts, L., Lee, K., Quayle, S., Beard, S., Braud, C., Bonneson, K., Tarnoff, P. and T. Urbanik. Traffic Signal Timing Manual. Accessed <http://ops.fhwa.dot.gov/publications/fhwahop08024/index.htm>

Mahmassani, H.S., Peeta, S., 1993. Network Performance under System Optimal and User Equilibrium Dynamic Assignments: Implications for ATIS. Transportation Research Record: Journal of the Transportation Research Board, No. 1408, 83-93.

Sharma, A., D. M. Bullock, and J. A. Bonneson. Input–Output and Hybrid Techniques for Real-Time Prediction of Delay and Maximum Queue Length at Signalized Intersections. In Transportation Research Record: Journal of the Transportation Research Board, No. 2035, Transportation Research Board of the National Academies, Washington, D.C., 2007, pp. 69–80.

Smaglik, E. J., A. Sharma, D. M. Bullock, J. R. Sturdevant, and G. Duncan, Event-Based Data Collection for Generating Actuated Controller Performance Measures.’’. In Transportation Research Record No. 2035, Transportation Research Board of the National Academies, Washington, D.C., 2007, pp. 97–106.

Smaglik, E.J., Bullock, D.M., Sharma, A., 2007a. Pilot study on real-time calculation of arrival type for assessment of arterial performance. ASCE J. Transport. Eng. 133 (7), 415–422.

Sturdevant, J. R., T. Overman, E. Raamot, R. Deer, D. Miller, D. M. Bullock, C. M. Day, T. M. Brennan, H. Li, A. Hainen, and S. M. Remias. Indiana Traffic Signal Hi Resolution Data Logger Enumerations. , Indiana Department of Transportation and Purdue University, West Lafayette, Indiana, 2012. <http://dx.doi.org/10.4231/K4RN35SH>

Traffic control system handbook. FHWA, U.S. Department of Transportation, 2005

Utah Department of Transportation. Automated Traffic Signal Performance Measures. Retrieved from <http://udottraffic.utah.gov/atspm/>

Wu, X., Liu, H., 2014. Using high-resolution event-based data for traffic modeling and control: an overview. Transp. Res. Part C: Emerg. Technol. 42, 28–43.

## 8.0 APPENDIX

The followings are R codes of the performance measures: call-based pedestrian delay and its Level of Service (LOS), Highway Capacity Manual based pedestrian delay, intervals of pedestrian signal indication, LOS of pedestrian delay by actuation, background cycle length, effective cycle length, green time, green-to-cycle ratio, vehicle counts in each cycle, proportion of vehicles arrived during green, Arrival Type, and Purdue Coordination Diagram.

### #### Pedestrian Delay by First Call & LOS ####

```
rm(list=ls(all=TRUE))#Run if all data needs to be cleared
#1#set the path bellow to where the data is located
datapath<-"C:/Users/Nawar/Documents"

#2# if needed, set the path bellow to where the graphs will be saved
graphpath<-"C:/Users/Nawar/Desktop/check/ped"

#3# uploads the whole data and subset the required one by running the
script below until #4#.
setwd(datapath)
library(plyr)
library(lubridate)
library(ggplot2)
library(TTR)
temp = list.files(path = datapath, pattern = "*.csv" )
data = do.call(rbind, lapply(temp, function(x) read.csv(x,
stringsAsFactors = FALSE, skip= 8, sep = ",", header = FALSE, col.names =
c("timestamp", "event", "phase") )))
data$timestamp<-strptime(data$timestamp, "%m/%d/%Y %H:%M:%OS")

#4# run the below script until #5#
pedsub<-subset(data, (phase == 2 | phase == 4 | phase == 6 | phase == 8) &
(event == 21 | event == 45 | event == 151))
f<-function(d1,d2,ph){
  coor<-subset(pedsub, phase == 2 & event == 151 & (timestamp >= d1 &
timestamp < d2))
  pedall<-subset(pedsub, (phase == 2 | phase == 4 | phase == 6 | phase ==
8) &
                (event == 21 | event == 45) & timestamp >= d1 &
timestamp < d2)
  ped<-subset(pedall, phase == ph &
              (event == 21 | event == 45))
  a<-ped[which(ped$event==45),]
  b<-ped[which(ped$event==45)+1,]
  colnames(b)<-c("timestamp2", "event2", "phase")
  c<-cbind(a,b)
  d<-c[!(c$event==45 & c$event2 ==45),]
```

```

d$delay<-difftime(d$timestamp2,d$timestamp, units = "secs")
#delay values larger than 140 sec have been removed
d$check<-ifelse(d$delay >= 140,F,T)
d<-d[which(d$check == T),]
maxdelay<-round(max(d$delay),digits = 0)
mindelay<-round(min(d$delay), digits = 0)
avedealy<-round(mean(d$delay),digits = 0)
call<-length(which(ped$event == 45))
nocalls<-length(which(ped$event == 45))-length(d$delay)
w2<-coor[which(coor$event == 151)[1],1] ## timestamp when Plan 2 begins
w3<-coor[tail(which(coor$event == 151))[6],1]
w4<-d$timestamp[1]## timestamp when Plan 2 ends
w5<-w4+(w2-w4)/2
w6<-w2+(w3-w2)/2
avgplan1<-round(mean(d[d$timestamp < w2 | d$timestamp > w3 ,7]), digits
= 0)
avgplan2<-round(mean(d[d$timestamp >= w2 & d$timestamp <= w3 ,7]),
digits = 0)
nocalls1<-length(d[d$timestamp < w2 | d$timestamp > w3 ,4])
nocalls2<-length(d[d$timestamp >= w2 & d$timestamp <= w3 ,4])

ggplot(d)+geom_bar(stat = "identity", color = "blue",aes(x =
d$timestamp,y = as.numeric(d$delay)))+
  scale_x_datetime(date_labels = "%H",date_breaks = ("1 hour"))+
  labs(x="Time of Day",
       y= paste("Pedestrian Delay","\n", "by 1st Call (sec)",
               title=paste("Phase",ph,"\n",format(as.POSIXct(d1),format='%B %d,
%Y')),
       subtitle= paste("Max Delay ",maxdelay,";",sep = "", " Min Delay
",mindelay,";", " AVG. Delay ",avedealy,";", " No. of Calls ",call,";", " No.
of Eliminated Calls (errors) ",nocalls))+
  theme(axis.text =element_text(size=14),
        axis.title=element_text(size=14,face="bold"),
        plot.title = element_text(size=14, face = "bold",hjust = 0.5),
        plot.subtitle=element_text(hjust=0.5, size = 11))+
  geom_vline(xintercept = as.numeric(w2), linetype = 2)+
  geom_vline(xintercept = as.numeric(w3), linetype = 2)+
  annotate("text",x = w2, y = 2, label = format(w2,"%H:%M"), colour =
"red", size = 4.5)+
  annotate("text",x = w3, y = 2, label = format(w3,"%H:%M"),colour =
"red", size = 4.5)+
  annotate("text",x = w5, y = 100, label = paste("Plan 1","\n", "Average
Delay (sec) =",avgplan1,"\n", "No. of Calls (without errors)
=",nocalls1),colour = "red",size = 4.5)+
  annotate("text",x = w6, y = 100, label = paste("Plan 2","\n", "Average
Delay (sec) =",avgplan2,"\n", "No. of Calls (without errors)
=",nocalls2),colour = "red",size = 4.5)
ggsave(filename = paste("Phase",ph,d1,".png"), path = graphpath,width =
20, height = 15, units = "cm")

d$los<-ifelse(d$delay <= 10,"0-10 LOS (A)",
             ifelse(d$delay > 10 & d$delay <= 20,"10-20 LOS (B)",
             ifelse (d$delay > 20 & d$delay <= 30,"20-30 LOS
(C)",

```

```

                                ifelse (d$delay > 30 & d$delay <= 40,"30-40
LOS (D)",
                                ifelse (d$delay > 40 & d$delay <=
60,"40-60 LOS (E)", ">60 LOS (F)"))))

    order<-c("0-10 LOS (A)", "10-20 LOS (B)","20-30 LOS (C)","30-40 LOS
(D)","40-60 LOS (E)", ">60 LOS (F)")

    print(ggplot(d, aes(d$los,y = (..count..)/sum(..count..)))+
          geom_bar(width=0.3, fill="steelblue")+
          scale_x_discrete(limits= order)+
          theme_minimal()+

geom_text(stat='count',aes(label=scales::percent(round(..count..)/sum(..c
ount..), digits = 2)),vjust=-0.5, size=5)+
          theme(axis.text
=element_text(size=12),axis.title=element_text(size=14,face="bold"),plot.t
itle = element_text(size=14, face = "bold",hjust = 0.5))+
          ggtitle(paste("Phase",ph,"\n",format(as.POSIXct(d1),format='%B
%d, %Y')))+xlab("sec (LOS)")+ ylab(""))+
          scale_y_continuous(labels=scales::percent))
    ggsave(filename = paste("Level of Service Based on Peds delay 1st
call",ph,d1,".png"), path = graphpath,width = 20, height = 15, units =
"cm")

    delaybycall<-d[,c(1:3,7,9)]
    print(delaybycall)
}

```

```
#5# set the designated date and phase
```

```

##Bellow, substitute the beginning and the end of the day along with the
designated phase in the form ("yyyy-mm-dd","yyyy-mm-dd", phase number).
##For example, if phase 2 needs to be determined on date May 23 2016, then
the begging of the day will be "2016-05-23" and the end of the day will be
"2016-05-24".
##So the final syntax will be f("2016-05-23","2016-05-24").
##DON'T FORGET THE TWO QUOTATION MARKS.
## "delaybycall" is final file

```

```
delaybycall<-f("2016-05-23","2016-05-24",2)
```

```
#####
```

### ####HCM Delay####

```
rm(list=ls(all=TRUE))#Run if all data needs to be cleared

#1#set the path bellow to where the data is located
datapath<-"C:/Users/Nawar/Documents"

#2# if needed, set the path bellow to where the graphs will be saved
graphpath<-"C:/Users/Nawar/Desktop/check/ped"
#3# uploads the whole data and subset the required one by running the
script below until #4#.
setwd(datapath)
library(plyr)
library(lubridate)
library(ggplot2)
library(TTR)
temp = list.files(path = datapath, pattern = "*.csv" )
data = do.call(rbind, lapply(temp, function(x) read.csv(x,
stringsAsFactors = FALSE, skip= 8, sep = ",", header = FALSE, col.names =
c("timestamp", "event", "phase" ) ))
data$timestamp<-strptime(data$timestamp, "%m/%d/%Y %H:%M:%OS")

#4# run the below script until #5#
pedsub<-subset(data, (phase == 2 | phase == 4 | phase == 6 | phase == 8) &
(event == 21 | event == 22 | event == 23))
bocall<-subset(data, (event == 1 | event == 8 | event == 10))
f<-function(d1,d2,ph) {
  pedall<-subset(pedsub, (phase == 2 | phase == 4 | phase == 6 | phase ==
8) &
                    (event == 21 | event == 22 | event == 23) & timestamp
>= d1 & timestamp < d2)
  ped<-subset(pedall, phase == ph &
              (event == 21 | event == 22 | event == 23))
  a<-ped[which(ped$event==21),]
  b<-ped[which(ped$event==21)+1,]
  c<-ped[which(ped$event==21)+2,]
  d<-cbind(a,b,c)
  d[,c(3,6,9)]<-NULL
  colnames(d)<-
c("timestamp", "event", "timestamp1", "event1", "timestamp2", "event2")
  d<-d[which(d$event == 21 & d$event1 == 22 & d$event2 == 23),]
  d$walk<-difftime(d$timestamp1,d$timestamp, units = "secs")
  d$fdw<-difftime(d$timestamp2,d$timestamp1,units = "secs")
  d$pedcrosstime<-d$walk+d$fdw
  boc<-subset(bocall, (event == 1 | event == 8 | event == 10) & timestamp
>= d1 & timestamp < d2)
  b<-boc[-1,]
  a<-head(boc,-1)
  colnames(b)<-c("timestamp2", "event2", "phase2")
```

```

c<-cbind(a,b)

c$boc<-ifelse((c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase == 5
| c$phase == 6) &
                c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)),
                "boc", "")

h<-c[which(c$boc=="boc"),]
e<-h[,1:3]
colnames(e)<-c("eoc", "event2", "phase2")
e<-e[-1,]
f<-h[,4:6]
colnames(f)<-c("boc", "event", "phase")

```

```

f<-head(f,-1)
g<-cbind(f,e)
g$NOC<-(1:length(g$boc))
g$cyclelength<- difftime(g$eoc,g$boc, units = "secs")
new<-g[which(g$event == 1 & g$event2 == 10),]
f<-function(x) {
  as.numeric(new[which(new$boc <= d$timestamp1[x] & new$eoc >=
d$timestamp1[x]),7])
}

d$NOV<-lapply(1:length(d$timestamp1),f)
d$cyclelength<-g[match(d$NOV,g$NOC),8]

new1<-d[!(is.na(d$cyclelength)),]

new1$delay<- (0.5 * (as.numeric(new1$cyclelength) -
(as.numeric(new1$walk) + 4))^2)/as.numeric(new1$cyclelength)

new1$los<-ifelse(new1$delay <= 10,"0-10 LOS (A)",
  ifelse(new1$delay > 10 & new1$delay <= 20,"10-20 LOS
(B)",
    ifelse (new1$delay > 20 & new1$delay <= 30,"20-
30 LOS (C)",
      ifelse (new1$delay > 30 & new1$delay <=
40,"30-40 LOS (D)",
        ifelse (new1$delay > 40 &
new1$delay <= 60,"40-60 LOS (E)", ">60 LOS (F)"))))
  order<-c("0-10 LOS (A)", "10-20 LOS (B)","20-30 LOS (C)","30-40 LOS
(D)","40-60 LOS (E)", ">60 LOS (F)")
  print(ggplot(new1, aes(new1$los,y = (..count..)/sum(..count..)))+
  geom_bar(width=0.3, fill="chocolate1")+
  scale_x_discrete(limits= order)+
  theme_minimal()+

geom_text(stat='count',aes(label=scales::percent(round((..count..)/sum(..c
ount..), digits = 2))),vjust=-0.5, size=5)+
  theme(axis.text
=element_text(size=12),axis.title=element_text(size=14,face="bold"),plot.t
itle = element_text(size=14, face = "bold",hjust = 0.5))+
  ggtitle(paste("Phase",ph,"\n",format(as.POSIXct(d1),format='%B
%d, %Y')))+xlab("sec (LOS)")+ ylab("")+
  scale_y_continuous(labels=scales::percent))
  ggsave(filename = paste("LOS HCM delay",ph,d1,".png"), path =
graphpath,width = 20, height = 15, units = "cm")

HCMdelay<-new1
print(HCMdelay)
}

```



```
#5# set the designated date and phase

##Bellow, substitute the beginning and the end of the day along with the
designated phase in the form ("yyyy-mm-dd","yyyy-mm-dd", phase number).
##For example, if phase 2 needs to be determined on date May 23 2016, then
the begging of the day will be "2016-05-23" and the end of the day will be
"2016-05-24".
##So the final syntax will be f("2016-05-23","2016-05-24").
##DON'T FORGET THE TWO QUOTATION MARKS.
## "HCMdelay" is final file
```

```
HCMdelay<-f("2016-05-23","2016-05-24",2)
```

```
#####
```

#### ####Pedestrian Signal Indication Intervals####

```
rm(list=ls(all=TRUE))#Run if all data needs to be cleared

#1#set the path bellow to where the data is located
datapath<-"C:/Users/Nawar/Documents"

#2# if needed, set the path bellow to where the graphs will be saved
graphpath<-"C:/Users/Nawar/Desktop/check/ped"

#3# uploads the whole data and subset the required one by running the
script below until #4#.
setwd(datapath)
library(plyr)
library(lubridate)
library(ggplot2)
library(TTR)
temp = list.files(path = datapath, pattern = "*.csv" )
data = do.call(rbind, lapply(temp, function(x) read.csv(x,
stringsAsFactors = FALSE, skip= 8, sep = ",", header = FALSE, col.names =
c("timestamp","event","phase" ) ))
data$timestamp<-strptime(data$timestamp, "%m/%d/%Y %H:%M:%OS")

#4# run the below script until #5#
pedsub<-subset(data, (phase == 2 | phase == 4 | phase == 6 | phase == 8 )
& (event == 21 | event == 22 | event == 23 | event == 10 | event == 151))
```

```

f<-function(d1,d2,ph) {
  coor<-subset(pedsub, phase == 2 & event == 151 & (timestamp >= d1 &
timestamp < d2))
  pedall<-subset(pedsub, (phase == 2 | phase == 4 | phase == 6 | phase ==
8) &
                    (event == 21 | event == 22 | event == 23 | event == 10)
& timestamp >= d1 & timestamp < d2)

  ped<-subset(pedall, phase == ph &
              (event == 21 | event == 22 | event == 23 | event == 10))
  a<-ped[which(ped$event==21),]
  b<-ped[which(ped$event==21)+1,]
  c<-ped[which(ped$event==21)+2,]
  cc<-ped[which(ped$event==21)+3,]
  d<-cbind(a,b,c,cc)
  d[,c(3,6,9,12)]<-NULL
  colnames(d)<-
c("timestamp", "event", "timestamp1", "event1", "timestamp2", "event2", "timesta
mp3", "event3")
  d<-d[which(d$event == 21 & d$event1 == 22 & d$event2 == 23 & d$event3 ==
10),]
  d$walk<-difftime(d$timestamp1,d$timestamp, units = "secs")
  d$fdw<-difftime(d$timestamp2,d$timestamp1,units = "secs")
  d$solid<-difftime(d$timestamp3,d$timestamp2,units = "secs")
  d$pedcrosstime<-d$walk+d$fdw+d$solid
  col<-
c("Black", "Black", "Black", "Black", "Black", "Black", "Black", "Black", "Black",
"Black", "red", "red", "red")

  w1<- round(as.numeric(d$walk[1]), digits = 0) ## walking time in plan 1
  w2<-coor[which(coor$event == 151)[1],1] ## timestamp when plan 2 begins
  w3<-coor[tail(which(coor$event == 151))[6],1] ## timestamp when plan 2
ends
  w4<-d$timestamp[1]
  w5<-w4+(w2-w4)/2
  w6<-w2+(w3-w2)/2
  w7<-round(mean(d[d$timestamp < w2 | d$timestamp > w3 ,9]), digits = 0)
## the average walking time in plan 1
  w8<-round(mean(d[d$timestamp > w2 & d$timestamp < w3 ,9]), digits = 0)
## the average walking time in plan 2

  fdw1<-round(as.numeric(d$fdw[1]),digits = 0) ## the average walking time
in plan 2
  fdw7<-round(mean(d[d$timestamp < w2 | d$timestamp > w3 ,10]), digits =
0)## the FDW time in plan 1
  fdw8<-round(mean(d[d$timestamp > w2 & d$timestamp < w3 ,10]), digits =
0) ## the FDW time in plan 2

  solid1<-round(as.numeric(d[d$timestamp > w2,11][11]),digits = 0) ## the
average walking time in plan 2
  solid7<-round(mean(d[d$timestamp < w2 | d$timestamp > w3 ,11]), digits =
0)## the FDW time in plan 1

```

```

    solid8<-round(mean(d[d$timestamp > w2 & d$timestamp < w3 ,11]), digits =
0) ## the FDW time in plan 1

    ggplot(d, aes(x = timestamp))+geom_point(aes(y = walk, color =
"skyblue1"),size = 2)+
    geom_point(aes(y = fdw,color = "darkorange"), size = 2)+
    geom_point(aes(y = solid, color = "firebrick1"), size = 2)+
    ggtitle(paste("Phase",ph,"\n",format(as.POSIXct(d1),format='%B %d,
%Y')))+ xlab("Time of Day")+ ylab("Sec")+
    theme(axis.text.y = element_text(colour = col),plot.title =
element_text(hjust = 0.5))+
    scale_x_datetime(date_labels = "%H:%M",date_breaks = ("2 hour"))+
    theme(axis.text
=element_text(size=16),axis.title=element_text(size=16,face="bold"),plot.t
itle = element_text(size=16, face = "bold"),legend.text = element_text(
size=14,face="bold"),legend.title =
element_blank(),legend.position="top")+
    scale_y_continuous(breaks = c(seq(0,90,10),w1,fdw1,solid1),
minor_breaks = seq(0,90,10), limits = c(0,90), labels=
(c(seq(0,90,10),w1,fdw1,solid1)))+
    geom_hline(yintercept = w1, linetype = 2)+geom_hline(yintercept =
fdw1, linetype = 2)+geom_hline(yintercept = solid1, linetype = 2)+
    geom_vline(xintercept = as.numeric(w2), linetype = 2)+
    geom_vline(xintercept = as.numeric(w3), linetype = 2)+
    annotate("text",x = w2, y = 0, label = format(w2,"%H:%M"), colour =
"red")+
    annotate("text",x = w3, y = 0, label = format(w3,"%H:%M"),colour =
"red")+
    annotate("text",x = w5, y = 75, label = paste("Plan 1","\n", "Avg Walk
Time (sec) =",w7,"\n", " Avg FDW Time (sec) =",fdw7,"\n", " Avg Solid Time
(sec) =",solid7),colour = "red",size =5)+
    annotate("text",x = w6, y = 75, label = paste("Plan 2","\n", "Avg Walk
Time (sec) =",w8,"\n", " Avg FDW Time (sec) =",fdw8,"\n", " Avg Solid Time
(sec) =",solid8),colour = "red",size = 5)+
    scale_colour_manual(labels = c("FDW","Solid
DW","Walking"),values=c("darkorange","firebrick1","deepskyblue4"))+
    labs(fill="")
    ggsave(filename = paste("Pedestrian Signal Time",ph,d1,".png"), path =
graphpath,width = 25, height = 17, units = "cm")

    signalind<-d
    print(signalind)

}

#5# set the designated date and phase

##Bellow, substitute the beginning and the end of the day along with the
designated phase in the form ("yyyy-mm-dd","yyyy-mm-dd", phase number).
##For example, if phase 2 needs to be determined on date May 23 2016, then
the begging of the day will be "2016-05-23" and the end of the day will be
"2016-05-24".
##So the final syntax will be f("2016-05-23","2016-05-24").

```

```
##DON'T FORGET THE TWO QUOTATION MARKS.
## "signalind" is final file

signalind<-f("2016-05-23","2016-05-24",2)
```

```
#####
```

### ####LOS of Pedestrian Delay by Actuation####

```
rm(list=ls(all=TRUE))#Run if all data needs to be cleared

#1#set the path bellow to where the data is located
datapath<-"C:/Users/Nawar/Documents"

#2# if needed, set the path bellow to where the graphs will be saved
graphpath<-"C:/Users/Nawar/Desktop/check/ped"

#3# uploads the whole data and subset the required one by running the
script below until #4#.
setwd(datapath)
library(plyr)
library(lubridate)
library(ggplot2)
library(TTR)
temp = list.files(path = datapath, pattern = "*.csv" )
data = do.call(rbind, lapply(temp, function(x) read.csv(x,
stringsAsFactors = FALSE, skip= 8, sep = ",", header = FALSE, col.names =
c("timestamp","event","phase" ) ))
data$timestamp<-strptime(data$timestamp, "%m/%d/%Y %H:%M:%OS")

#4#Bellow, substitute the beginning and the end of the day in form "yyyy-
mm-dd"
#Then, run the below script until #5#
d1<-"2016-05-23"      #### beginning of the day
d2<-"2016-05-24"      #### end of the day

all<-subset(data,timestamp >= d1 & timestamp < d2 )
coor<-subset(all, phase == 2 & event == 151)
```

```

boc<-subset(all, (event == 1 | event == 8 | event == 10))
b<-boc[-1,]
a<-head(boc,-1)
colnames(b)<-c("timestamp2","event2","phase2")
c<-cbind(a,b)
c$boc<-ifelse((c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase == 5 |
c$phase == 6) &
               c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4| c$phase2
== 7 | c$phase2 == 8)) |
               (c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase == 5
| c$phase == 6) &
               c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
               (c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase == 5
| c$phase == 6) &
               c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
               (c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase == 5
| c$phase == 6) &
               c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
               (c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase == 5
| c$phase == 6) &
               c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
               (c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase == 5
| c$phase == 6) &
               c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
               (c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase == 5
| c$phase == 6) &
               c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
               (c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase == 5
| c$phase == 6) &
               c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
               (c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase == 5
| c$phase == 6) &
               c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)),
        "boc", "")

d<-c[which(c$boc=="boc"),]
e<-d[,1:3]

```

```

colnames(e)<-c("eoc","event2","phase2")
e<-e[-1,]
f<-d[,4:6]
colnames(f)<-c("boc","event","phase")
f<-head(f,-1)
g<-cbind(f,e)
g$NOC<-(1:length(g$boc))
# one sec is added for the correction
g$cyclelength<-difftime(g$eoc,g$boc, units = "secs")+as.numeric(1, units =
'secs')
length(g[which(g$event != 1 | g$event2 != 10 ),7]) ## number of the cycles
that has been removed
g[which(g$event != 1 | g$event2 != 10),7] ## designated cycle numbers that
have been removed

# Final file of the cycle length
final<-g[which(g$event == 1 & g$event2 == 10),c(1,7,8)]

vol<- merge(all,g, by.x = "timestamp", by.y = "boc", all.x = T)
vol[,c(4:8,10:11)]<-NULL
colnames(vol)<-c("timestamp","event","phase","NOC")
vol<-tail(vol,-which(vol$NOC == 1)[1]+1)
f4 <- function(x, blank = is.na) {
  # Find the values
  if (is.function(blank)) {
    isnotblank <- !blank(x)
  } else {
    isnotblank <- x != blank
  }
  # Fill down
  x[which(isnotblank)][cumsum(isnotblank)]
}

vol$NOC<-f4(vol$NOC)#every event has been assigned to its cycle length

# the script below is to determine pedestrian actual delay based on
actuation

f<-function(ph) {
  mm<-data.frame(a = as.POSIXct("2016-05-23 00:00:00", origin="1970-01-
01"), b = 0, c = 0 , d = 0, e=0)
  a<-subset(vol, (phase == ph & event == 21) | (phase == ph & event ==
90))

  s=1
  for (i in 1:(length(which(a$event == 21))-1)) {
    if ((which(a$event == 21)[i+1] - which(a$event == 21)[i] >=2)) {
      x<-a[(which(a$event == 21)[i+1]):(which(a$event == 21)[i+1]),]
      for(j in 1:length(which(x$event == 90))){

```

```

        if(x$NOC[j] == a[which(a$event == 21)[i+1],4] | x$NOC[j]+1 ==
a[which(a$event == 21)[i+1],4]) {
            mm[s,3]<-difftime(a[(which(a$event == 21)[i+1]),1],x[j,1], units
= "secs" )
            mm[s,1]<-as.POSIXct(x$timestamp[j])
            mm[s,2]<-x$event[j]
            mm[s,4]<-x$phase[j]
            mm[s,5]<-x$NOC[j]
            s=s+1
        }

    }
}

colnames(mm)<-c("timestamp","event","delay","phase","NOC" )
d<-mm
# Remove delay values which are higher than 140 sec. they are most
probably errors
# they cannot be higher than max cycle length
d$check<-ifelse(d$delay >= 140,F,T)
d<-d[which(d$check == T),]
####

d$los<-ifelse(d$delay <= 10,"0-10 LOS (A)",
             ifelse(d$delay > 10 & d$delay <= 20,"10-20 LOS (B)",
             ifelse (d$delay > 20 & d$delay <= 30,"20-30 LOS
(C)",
                    ifelse (d$delay > 30 & d$delay <= 40,"30-40
LOS (D)",
                            ifelse (d$delay > 40 & d$delay <=
60,"40-60 LOS (E)", ">60 LOS (F)")))))

order<-c("0-10 LOS (A)", "10-20 LOS (B)","20-30 LOS (C)","30-40 LOS
(D)","40-60 LOS (E)", ">60 LOS (F)")

maxdelay<-round(max(d$delay), digits = 0)
mindelay<-round(min(d$delay), digits = 0)
q1<-round(quantile(d$delay,0.25), digits = 0)## 25th quantile
q2<-round(quantile(d$delay,0.5), digits = 0)## 50th quantile
q3<-round(quantile(d$delay,0.75), digits = 0)## 75th quantile
er<-length(which(a$event == 90))-length(d$delay)
tot<-length(which(a$event == 90))

print(ggplot(d, aes(d$los,y = (..count..)/sum(..count..)))+
      geom_bar(width=0.3, fill="steelblue")+
      scale_x_discrete(limits= order)+
      theme_minimal()+

```

```

geom_text(stat='count',aes(label=scales::percent(..count../sum(..count..
)),vjust=-0.5, size=5)+
  theme(axis.text
=element_text(size=12),axis.title=element_text(size=14,face="bold"),plot.t
itle = element_text(size=22, face = "bold",hjust =
0.5),plot.subtitle=element_text(hjust=0.5))+
  labs(x="sec (LOS)",
       y= paste(""),
       title=paste("Pedestrian Delay By Actuation
(sec)", "\n", "Phase", ph, "\n", format(as.POSIXct("2016-05-23"), format='%B %d,
%Y')),
       subtitle= paste("Max Delay ",maxdelay,";", " Min Delay
",mindelay,";", " 25th Quantile (sec) =",q1," 50th Quantile (sec) =",q2,"
75th Quantile (sec) =",q3, "No of removed rows =",er, "calls=",tot))+
  scale_y_continuous(labels=scales::percent)
  ggsave(filename = paste(" Pedestrian Delay By Actuation
(90)",ph,d1,".png"), path = graphpath,width = 25, height = 17, units =
"cm")

delaybyactuation<-d[,c(1:5,7)]
print(delaybyactuation)

}

#5# set the designated phase parentheses
# for example if LOS of pedestrian delay by actuation is for phase 2
required, run as f(2)
## "delaybyactuation" is final file

delaybyactuation<-f(2)

#####

##### Background Cycle Length #####

rm(list=ls(all=TRUE))#Run if all data needs to be cleared

#1#set the path bellow to where the data is located
datapath<-"C:/Users/Nawar/Documents"

#2# if needed, set the path bellow to where the graphs will be saved
graphpath<-"C:/Users/Nawar/Desktop/check/ped"

```



```

#3# uploads the whole data and subset the required one by running the
script below until #4#.
setwd(datapath)
library(plyr)
library(lubridate)
library(ggplot2)
library(TTR)
temp = list.files(path = datapath, pattern = "*.csv" )
data = do.call(rbind, lapply(temp, function(x) read.csv(x,
stringsAsFactors = FALSE, skip= 8, sep = ",", header = FALSE, col.names =
c("timestamp","event","phase") )))
data$timestamp<-strptime(data$timestamp, "%m/%d/%Y %H:%M:%OS")

#4# run the below script until #5#
##In our intersection, Division & 122nd, the coordinated phases are 2 and
6.
##Therefore, "phase" in the syntax bellow can be substituted as either 2
or 6.
a<-subset(data,event==151 & phase==2 )

f<-function(d1,d2) {
  z<-subset(a, timestamp >= d1 & timestamp < d2)
  z1<-z$timestamp[-1]
  z2<-head(z,-1)
  z3<-cbind(z1,z2)
  z3$bg.cycle.length<-(z3$z1-z3$timestamp)*60
  l3<-
as.numeric(names(which(table(z3$bg.cycle.length)==max(table(z3$bg.cycle.le
ngth))))))
  l2<-z[1,1]
  l1<-tail(z)[6,1]
  col<-
c("Black","Black","Black","Black","Black","Black","Black","Black",
"red")

  ggplot(z3,aes(timestamp, bg.cycle.length))+geom_point(size = 1.5)+
  ggtitle(paste(format(as.POSIXct(d1),format='%B %d, %Y')))+ xlab("Time
of Day")+ ylab("sec")+
  theme(plot.title = element_text(hjust = 0.5),axis.text.y =
element_text(colour = col))+
  scale_x_datetime(date_labels = "%H:%M",date_breaks = ("2 hour"),
limits = c(as.POSIXct(d1),as.POSIXct(d2)))+
  theme(axis.text
=element_text(size=14),axis.title=element_text(size=14,face="bold"),plot.t
itle = element_text(size=14, face = "bold"))+
  scale_y_continuous(breaks = c(seq(0,400,50),l3), minor_breaks =
seq(0,400,50), limits = c(0,400) )+
  geom_vline(xintercept = as.numeric(l1), linetype =
2)+geom_vline(xintercept = as.numeric(l2), linetype =
2)+geom_hline(yintercept = l3, linetype = 2)+
  annotate("text",x = l1, y = 0, label = format(l1,"%H:%M"), colour =
"red",size = 4.5)+

```

```

    annotate("text",x = 12, y = 0, label = format(12,"%H:%M"),colour =
"red",size = 4.5)
    ggsave(filename = paste("Background cycle length",d1,".png"), path =
graphpath,width = 20, height = 15, units = "cm")
    BGCL<-z3[,c(2:5)]
    print(BGCL)
}

```

```

#5# set the designated date and phase

```

```

##Bellow, substitute the beginning and the end of the day ("yyyy-mm-
dd,"yyyy-mm-dd").
##DON'T FORGET THE TWO QUOTATION MARKS.
## "BGCL" is final file

```

```

BGCL<-f("2016-05-23","2016-05-24")

```

```

#####

```

#### ####Effective Cycle Length + Effective Green Time + Vehicle Count + POG + Arrival Type####

```

#note# All the above will be visualized when running the below code

```

```

rm(list=ls(all=TRUE))#Run, if all data needs to be cleared

```

```

#1#set the path bellow to where the data is located
datapath<-"C:/Users/Nawar/Documents"

```

```

#2# if needed, set the path bellow to where the graphs will be saved
graphpath<-"C:/Users/Nawar/Desktop/check/ped"

```

```

#3# uploads the whole data and subset the required one by running the
script below until #4#.

```

```

setwd(datapath)
library(plyr)
library(lubridate)
library(ggplot2)
library(TTR)
temp = list.files(path = datapath, pattern = "*.csv" )
data = do.call(rbind, lapply(temp, function(x) read.csv(x,
stringsAsFactors = FALSE, skip= 8, sep = ",", header = FALSE, col.names =
c("timestamp","event","phase") )))
data$timestamp<-strptime(data$timestamp, "%m/%d/%Y %H:%M:%OS")

```

```

#4# Follow points; "a" and "b" below, then run the code until #5#
#a#Set below which advanced detectors is going to be used for the results.
#If it is the front, place "front", or if it is back set "back"
#Estimated operating speed is 45 mph (66) ft/sec

```

```

detcocation <- c("back")
operatingspeed = 66

```

```

#b#bellow, substitute the beginning and end of required day in form "yyyy-
mm-dd" and run code until #5#
#1st date
d1<-"2016-05-23"
#2nd date
d2<-"2016-05-24"

```

```

all<-subset(data,timestamp >= d1 & timestamp < d2 )
f<-function(ph){
  coor<-subset(all, (phase == 2 & event == 151 & timestamp >= d1 &
timestamp < d2))
  boc<-subset(all, (event == 1 | event == 8 | event == 10))
  b<-boc[-1,]
  a<-head(boc,-1)
  colnames(b)<-c("timestamp2","event2","phase2")
  c<-cbind(a,b)

  c$boc<-ifelse((c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase == 5
| c$phase == 6) &
                c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |

```

```

(c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
      c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4 |
c$phase2 == 7 | c$phase2 == 8)) |

(c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
      c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4 |
c$phase2 == 7 | c$phase2 == 8)) |

(c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
      c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4 |
c$phase2 == 7 | c$phase2 == 8)) |

(c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
      c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4 |
c$phase2 == 7 | c$phase2 == 8)),
      "boc", "")
d<-c[which(c$boc=="boc"),]
e<-d[,1:3]
colnames(e)<-c("eoc", "event2", "phase2")
e<-e[-1,]
f<-d[,4:6]
colnames(f)<-c("boc", "event", "phase")
f<-head(f,-1)
g<-cbind(f,e)
g$NOC<-(1:length(g$boc))
# one sec is added for the correction
g$cyclelength<-difftime(g$eoc,g$boc, units = "secs")+as.numeric(1, units
= 'secs')
# #####
length(g[which(g$event != 1 | g$event2 != 10),7]) # number of the cycle
length that has been removed
g[which(g$event != 1 | g$event2 != 10),7] #that have been removed

# Final file of the cycle length
final<-g[which(g$event == 1 & g$event2 == 10),c(1,7,8)]

##### -----below is green time code ----- #####

dd<-merge(boc,g, by.x = "timestamp", by.y = "boc", all.x = T)
dd<-tail(dd,-which(dd$NOC == 1)[1]+1)
f4 <- function(x, blank = is.na) {
  # Find the values
  if (is.function(blank)) {
    isnotblank <- !blank(x)
  } else {
    isnotblank <- x != blank
  }
  # Fill down

```

```

    x[which(isnotblank)][cumsum(isnotblank)]
  }

  dd$NOC<-f4(dd$NOC)
  new<-subset(dd,phase.x == ph & (event.x == 1 | event.x == 8))
  for(i in 1:length(new$event.x)){
    new$check[i]<-ifelse((new$event.x[i] == 1 & new$event.x[i+1] == 8 &
new$NOC[i]==new$NOC[i+1]),"T","F")

  }
  new$check2[1]<-"T"
  ## below is only to determine the number of cycles and the cycles
numbers that will be removed
  s=2
  for(i in 1:(length(new$event.x)-1)){
    new$check1[i]<-ifelse((new$event.x[i] == 1 & new$event.x[i+1] == 8 &
new$NOC[i]==new$NOC[i+1]),"T","F")
    new$check2[s]<-ifelse((new$event.x[i] == 1 & new$event.x[i+1] == 8 &
new$NOC[i]==new$NOC[i+1]),"T","F")
    if (s<=(length(new$event.x)-1)) {s=s+1}
  }
  new$check3<-ifelse(new$check == "T" | new$check2 == "T", "T","F")

  # below, number of cycles that we couldn't calculate their GT
(regardless of the cycle length errors in the previous section)
  length(new[which(new$check3 == "F"),9])
  #which cycles having the errors
  new[which(new$check3 == "F"),9]
  new1<-cbind(new[which(new$event.x == 1 & new$check == "T"),c(1,2)],
             new[which(new$event.x == 1 & new$check == "T")+1,c(1:3,9)])
  new1$starLT<-as.numeric(2,units = "secs")
  new1$clearADD<-as.numeric(2,units = "secs")
  new1$BOGeffective<-new1[,1]+new1$starLT
  new1$EOGeffective<-new1[,3]+new1$clearADD
  new1$greentime<-difftime(new1$EOGeffective,new1$BOGeffective, units =
"secs")

  ###Final File###

  final1<-merge(final,new1, by.x = "NOC", by.y = "NOC", all.x =
T)[,c("boc","NOC","cyclelength","phase.x","BOGeffective","greentime")]
  final1<-na.omit(final1)
  final1$gC <- as.numeric(final1$greentime)/as.numeric(final1$cyclelength)

  ##### ----- below is vehicles count code-----#####

  vol<- merge(all,g, by.x = "timestamp", by.y = "boc", all.x = T)
  vol[,c(4:8,10:11)]<-NULL
  colnames(vol)<-c("timestamp","event","phase","NOC")
  vol<-tail(vol,-which(vol$NOC == 1)[1]+1)
  f4 <- function(x, blank = is.na) {
    # Find the values

```

```

    if (is.function(blank)) {
      isnotblank <- !blank(x)
    } else {
      isnotblank <- x != blank
    }
    # Fill down
    x[which(isnotblank)][cumsum(isnotblank)]
  }

vol$NOC<-f4(vol$NOC)
#phase info#
vol<-vol[which(vol$event == 8 & vol$phase == ph),]
vol$timestamp<-vol$timestamp + as.numeric(2, units = "secs")

# detector info#

v<-if (vol$phase[1] == 1 & detcocation == "front") {subset(all,event ==
82 & phase == 1)
} else if (vol$phase[1] == 1 & detcocation == "back") {subset(all,event
== 82 & phase == 2)
} else if (vol$phase[1] == 2 & detcocation == "front") {subset(all,event
== 82 & phase == 9)
} else if (vol$phase[1] == 2 & detcocation == "back") {subset(all,event
== 82 & phase == 10)
} else if (vol$phase[1] == 3 & detcocation == "front") {subset(all,event
== 82 & phase == 3)
} else if (vol$phase[1] == 3 & detcocation == "back") {subset(all,event
== 82 & phase == 4)
} else if (vol$phase[1] == 4 & detcocation == "front") {subset(all,event
== 82 & phase == 14)
} else if (vol$phase[1] == 4 & detcocation == "back") {subset(all,event
== 82 & phase == 15)
} else if (vol$phase[1] == 5 & detcocation == "front") {subset(all,event
== 82 & phase == 5)
} else if (vol$phase[1] == 5 & detcocation == "back") {subset(all,event
== 82 & phase == 6)
} else if (vol$phase[1] == 6 & detcocation == "front") {subset(all,event
== 82 & phase == 19)
} else if (vol$phase[1] == 6 & detcocation == "back") {subset(all,event
== 82 & phase == 20)
} else if (vol$phase[1] == 7 & detcocation == "front") {subset(all,event
== 82 & phase == 7)
} else if (vol$phase[1] == 7 & detcocation == "back") {subset(all,event
== 82 & phase == 8)
} else if (vol$phase[1] == 8 & detcocation == "front") {subset(all,event
== 82 & phase == 24)
} else if (vol$phase[1] == 8 & detcocation == "back") {subset(all,event
== 82 & phase == 25)
}else ("wrong")

```

```

    if (v$phase[1] == 1 | v$phase[1] == 3 | v$phase[1] == 5 | v$phase[1] ==
7 ) {v$timestamp<- v$timestamp + as.numeric(36/operatingspeed , units =
"secs")
    }else if (v$phase[1] == 2 | v$phase[1] == 4 | v$phase[1] == 6 |
v$phase[1] == 8 ) {v$timestamp<- v$timestamp +
as.numeric(100/operatingspeed , units = "secs")
    }else if (v$phase[1] == 9 | v$phase[1] == 14 | v$phase[1] == 19 |
v$phase[1] == 24 ) {v$timestamp<- v$timestamp +
as.numeric(86/operatingspeed , units = "secs")
    }else if (v$phase[1] == 10 | v$phase[1] == 15 | v$phase[1] == 20 |
v$phase[1] == 25 ) {v$timestamp<- v$timestamp +
as.numeric(185/operatingspeed , units = "secs")}

a<-rbind.fill(v,vol)
b<-a[order(a$timestamp),]
if (vol$phase[1] == 2 | vol$phase[1] == 4 | vol$phase[1] == 6 |
vol$phase[1] == 8 ) {
    test<-vol
    #dupliactions
    test<-vol[which(duplicated(test$NOC) != TRUE),] # The duplication has
been removed
    length(which(duplicated(vol$NOC) == TRUE)) # the number of
duplications

    test$vol<-list(NA)
    ff<-function(x){
        as.integer(which(b$event == 8 & b$NOC == x+1)- which(b$event == 8 &
b$NOC == x) - 1)
    }
    test$vol<-(lapply(unique(test$NOC),ff))
    test$vol<-c(which(b$event == 8) [1]-1,head(test$vol,-1))
    test$check<-"NA"
    for(i in 2:length(test$NOC)-1){
        test$check[i+1]<-ifelse(test$NOC[i+1]-test$NOC[i]==1,"T","F")
    }
    length(test[which(test$check == "F"),][,4])
    test[which(test$check == "F"),][,4]
    test$vol<-as.numeric(as.character(test$vol))
    last22<-merge(finall1,test,by = "NOC", all.x = T)
    last22[,c(8:10,12)]<-NULL
    final2<-na.omit(last22)
}else if(vol$phase[1] == 1 | vol$phase[1] == 3 | vol$phase[1] == 5 |
vol$phase[1] == 7 ) {
    abc<-data.frame(a=0,b=0)
    for(i in 1:length(which(b$event == 8)-1)) {
        abc[i,1]<-which(b$event == 8) [i+1]-which(b$event == 8) [i]-1
        abc[i,2]<-b[which(b$event == 8) [i+1],"NOC"]
    }
    colnames(abc)<-c("vol","NOC")
    last22<-merge(finall1,abc,by = "NOC", all.x = T)
    final2<-na.omit(last22)
}

```

```

#####--below is Arrival Type and Percentage on Green (POG) code--#####

vol<- merge(all,g, by.x = "timestamp", by.y = "boc", all.x = T)
vol[,c(4:8,10:11)]<-NULL
colnames(vol)<-c("timestamp","event","phase","NOC")
vol<-tail(vol,-which(vol$NOC == 1)[1]+1)
f4 <- function(x, blank = is.na) {
  # Find the values
  if (is.function(blank)) {
    isnotblank <- !blank(x)
  } else {
    isnotblank <- x != blank
  }
  # Fill down
  x[which(isnotblank)][cumsum(isnotblank)]
}

vol$NOC<-f4(vol$NOC)

vol<-vol[which((vol$event == 1 & vol$phase == ph) | (vol$event == 8 &
vol$phase == ph)),]
vol$timestamp<-vol$timestamp + as.numeric(2, units = "secs")
vol<-vol[which(vol$event == 1) & which(vol$event == 1)+1,]
vol$check<-NA

for ( i in 1:length(vol$NOC)){
  vol$check[i]<-ifelse((vol[i,2] == 1 & vol[i+1,2] == 8 & vol[i,4] ==
vol[i+1,4]) , "T","" )

}
a<-vol[which(vol$check == "T"),]
b<-vol[which(vol$check == "T")+1,]
c<-rbind(a,b)
c<-c[order(c$timestamp),]
vol<-cbind(a,b)
a<-rbind.fill(v,c)
b<-a[order(a$timestamp),]
test<-vol
test$volONGREEN<-NA
for(i in 1:length(test$NOC)){
  test$volONGREEN[i]<-(which(b$event == 8)[i]-which(b$event == 1)[i]-1)
}
final4<-
merge(test[,c(1,3,4,6,11)],final2[,c("NOC","boc","cyclelength","gC",
"vol","greentime")], by.x = "NOC", by.y = "NOC", all.x = T)
colnames(final4)
final4$POG<-final4$volONGREEN/final4$vol
final4$volONRED<-final4$vol-final4$volONGREEN
final4<-final4[,c( "timestamp","timestamp.1","boc","NOC","phase"
,"greentime","cyclelength", "vol","volONGREEN","volONRED", "gC" ,"POG" )]
final4$Rp<-final4$POG/final4$gC

```



```

final4$AT<-ifelse(final4$Rp<=0.5,2*final4$Rp +1,
                 ifelse(final4$Rp > 0.5 & final4$Rp
<=0.85,(final4$Rp/0.35)+(3-(0.85/0.35)),
                 ifelse(final4$Rp > 0.85 & final4$Rp <=
1.15,(final4$Rp/0.3)+(4-(1.15/0.3)),
                 ifelse(final4$Rp > 1.15 & final4$Rp <=
1.5,(final4$Rp/0.35)+(5-(1.5/0.35)),
                 ifelse(final4$Rp > 1.5 &
final4$Rp <= 2,2 * final4$Rp +2,
                 ifelse(final4$Rp >
2,6,"wrong")))))))

final4$Quality<-ifelse(final4$Rp<=0.5,"Very poor",
                      ifelse(final4$Rp > 0.5 & final4$Rp
<=0.85,"Unfavorable",
                      ifelse(final4$Rp > 0.85 & final4$Rp <=
1.15,"Random arrivals",
                      ifelse(final4$Rp > 1.15 & final4$Rp
<= 1.5,"Favorable",
                      ifelse(final4$Rp > 1.5 &
final4$Rp <= 2,"Highly favorable",
                      ifelse(final4$Rp >
2,"Exceptional","wrong")))))))

### Info on the graph
if (vol$phase[1] == 2 | vol$phase[1] == 4 | vol$phase[1] == 6 |
vol$phase[1] == 8 ) {
  e<-length(which(is.na(last22$vol)==T))+length(g[which(g$event != 1 |
g$event2 != 10),7]) # the number of cycles have been removed (cycles and
vehicle count calculations)
} else {e<-0}
final2$ma<-SMA(na.omit(final2$vol), n =10)# 10 point moving average for
vehicles count
final2$vol_v_h<-(final2$vol * 3600)/as.numeric(final2$cyclelength) ###
Equivalent Hourly volume (vph)
final2$ma_v_h<-SMA(na.omit(final2$vol_v_h), n =10)# 10 point moving
average for Equivalent Hourly volume (vph)
final2$ma_gc<-SMA(na.omit(final2$gC), n =10)

### numbers on the graph
l1<-round(as.numeric(tail(names(sort(table(final$cyclelength))),2)),
digits = 0)[1] #2nd most frequent cycle length (rounded to digits 0)
l2<-round(as.numeric(tail(names(sort(table(final$cyclelength))),2)),
digits = 0)[2] #1st most frequent cycle length (rounded to digits 0)

```

```

col<-
c("Black","Black","Black","Black","Black","Black","Black","Black","Black","red","r
ed") # colors of numbers on the y-axis

w2<-coor[which(coor$event == 151)[1],1] ## Timestamp when Plan 2 begins
w3<-coor[tail(which(coor$event == 151))[6],1] ## Timestamp when Plan 2
ends
w4<-d$timestamp[1] ## the first timestamp of the date
w5<-w4+(w2-w4)/2 ## the spot where the 1st test is written in the graph

n<-tail(final$NOC,1) # Apro number of cycles during the day
r<-length(g[which(g$event != 1 | g$event2 != 10),7]) #Apro number of
cycles that have been removed

##### Visualizing Cycle Length
print(ggplot(final, aes(boc,cyclelength))+geom_point(size = 3)+
  labs(x="Time of the Day",
        y= "Sec",
        title=paste("Effective Cycle Length","\n",d1),
        subtitle= paste("Approximate Number of Cycles during This
Day =",n,"; ",sep ="","Approximate Number of Removed Cycles =",r))+
  scale_x_datetime(date_labels = "%H:%M",date_breaks = ("2 hour"),
limits = c(as.POSIXct(d1),as.POSIXct(d2)))+
  scale_y_continuous(breaks = c(seq(0,400,20),11,12), minor_breaks
= seq(0,400,20), limits = c(0,140) )+
  theme(plot.title = element_text(hjust = 0.5), axis.text.y =
element_text(colour = col), plot.subtitle=element_text( hjust=0.5))+
  theme(axis.text
=element_text(size=12),axis.title=element_text(size=14,face="bold"),plot.t
itle = element_text(size=22, face = "bold"))+
  geom_hline(yintercept = 11, linetype = 2)+geom_hline(yintercept
= 12, linetype = 2)+
  geom_vline(xintercept = as.numeric(w2), linetype = 2)+
  geom_vline(xintercept = as.numeric(w3), linetype = 2)+
  annotate("text",x = w2, y = 0, label = format(w2,"%H:%M"),
colour = "red")+
  annotate("text",x = w3, y = 0, label = format(w3,"%H:%M"),colour
= "red"))

#####

### Information on the graph
final1$ma<-SMA(na.omit(final1$greentime), n =10)
e<-length(g$NOC)- length(final1$NOC) # cycles have been removed
# numbers on the graph
l1<-round(as.numeric(tail(names(sort(table(final1$greentime))),2)),
digits = 0)[1] #2nd most frequent greentime (rounded to digits 0)
l2<-round(as.numeric(tail(names(sort(table(final1$greentime))),2)),
digits = 0)[2] #1st most frequent greentime (rounded to digits 0)

w2<-coor[which(coor$event == 151)[1],1] ## Timestamp when Plan 2 begins
w3<-coor[tail(which(coor$event == 151))[6],1] ## Timestamp when Plan 2
ends

```

```

# y-axis limit
if (final1$phase.x[1] == 2 | final1$phase.x[1] == 4 | final1$phase.x[1]
== 6 | final1$phase.x[1] == 8) {
  lim<-c(0,80)
} else { lim<-c(0,40)}

# the color of y-axis numbers
if (final1$phase.x[1] == 2 | final1$phase.x[1] == 4 | final1$phase.x[1]
== 6 | final1$phase.x[1] == 8) {
  col<-c("Black","Black","Black","Black","Black","red","red")
} else { col<-c("Black","Black","Black","red","red") }
# colors of numbers on the y-axis

##### Visualizing Green Time
print(ggplot(final1,aes(boc,greentime))+geom_point(size = 2, color =
"cornflowerblue")+geom_line(aes(y = ma), size = 2, color = "Black")+
  labs(x="Time of Day",
        y= "sec",
        title=paste("Green Time","\n","Phase ",ph,"\n",d1),
        subtitle= paste("Approximate Number of Removed Cycles
=",e))+
  theme(plot.title = element_text(hjust = 0.5),
plot.subtitle=element_text( hjust=0.5))+
  theme(axis.text =element_text(size=12),axis.text.y =
element_text(colour =
col),axis.title=element_text(size=14,face="bold"),plot.title =
element_text(size=22, face = "bold"))+
  scale_x_datetime(date_labels = "%H:%M",date_breaks = ("2 hour"),
limits = c(as.POSIXct("2016-05-23"),as.POSIXct("2016-05-24")))+
  scale_y_continuous(breaks = c(seq(0,400,20),l1,l2), minor_breaks
= seq(0,400,20), limits = lim)+
  geom_hline(yintercept = l1, linetype = 2)+geom_hline(yintercept
= l2, linetype = 2)+
  geom_vline(xintercept = as.numeric(w2), linetype = 2)+
geom_vline(xintercept = as.numeric(w3), linetype = 2)+
  annotate("text",x = w2, y = 0, label = format(w2,"%H:%M"),
colour = "red")+
  annotate("text",x = w3, y = 0, label = format(w3,"%H:%M"),colour
= "red"))
  ggsave(filename = paste("Average Pedestrian Delay By Actuation in Each
Hour of the Day",ph,d1,".png"), path = graphpath,width = 20, height = 15,
units = "cm")

##### Visualizing vehicle count
print(ggplot(final2,aes(x = boc))+geom_point(aes(y = vol), color =
"cornflowerblue")+
  geom_line(aes(y = ma), size = 2, color = "Black")+
  labs(x="Time of Day",
        y= "No. of Vehicles ",

```

```

        title=paste("Approach Volume","\n","Phase
",ph,"\n",format(as.POSIXct(d1),format='%B %d, %Y'))+
        theme(plot.title = element_text(hjust = 0.5),
plot.subtitle=element_text( hjust=0.5))+
        theme(axis.text
=element_text(size=12),axis.title=element_text(size=14,face="bold"),plot.t
itle = element_text(size=22, face = "bold"))+
        scale_x_datetime(date_labels = "%H:%M",date_breaks = ("2 hour"),
limits = c(as.POSIXct(d1),as.POSIXct(d2)))+
        scale_y_continuous(minor_breaks = 5, breaks = seq(0,400,5))+
        geom_vline(xintercept = as.numeric(w2), linetype = 2)+
geom_vline(xintercept = as.numeric(w3), linetype = 2)+
        annotate("text",x = w2, y = 0, label = format(w2,"%H:%M"),
colour = "red")+
        annotate("text",x = w3, y = 0, label = format(w3,"%H:%M"),colour
= "red"))
    ggsave(filename = paste("vol",d1,ph,".png"), path = graphpath,width =
20, height = 15, units = "cm")

```

```

final4<-na.omit(final4)
final4$map<-SMA(na.omit(final4$POG), n =10)

```

```

# Visualizing POG
print(ggplot(final4,aes(x = boc))+geom_point(aes(y = POG), color =
"cornflowerblue")+
        geom_line(aes(y = map), size = 2, color = "Black")+
        labs(x="Time of Day",
            y= "POG",
            title=paste("", "\n", "Phase
",ph,"\n",format(as.POSIXct(d1),format='%B %d, %Y'))+
            theme(plot.title = element_text(hjust = 0.5),
plot.subtitle=element_text( hjust=0.5))+
            theme(axis.text
=element_text(size=12),axis.title=element_text(size=14,face="bold"),plot.t
itle = element_text(size=22, face = "bold"))+
            scale_x_datetime(date_labels = "%H:%M",date_breaks = ("2 hour"),
limits = c(as.POSIXct(d1),as.POSIXct(d2)))+
            scale_y_continuous(breaks = seq(0,1,0.25))+
            geom_vline(xintercept = as.numeric(w2), linetype = 2)+
geom_vline(xintercept = as.numeric(w3), linetype = 2)+
            annotate("text",x = w2, y = 0, label = format(w2,"%H:%M"),
colour = "red")+
            annotate("text",x = w3, y = 0, label = format(w3,"%H:%M"),colour
= "red"))
    ggsave(filename = paste("POG",d1,ph,".png"), path = graphpath,width =
20, height = 15, units = "cm")

```

```

##### Visualizing Arrival Type (AT)
final4$maat<-SMA(na.omit(as.numeric(final4$AT)), n =10)
print(ggplot(final4,aes(x = boc))+geom_point(aes(y =
as.numeric(final4$AT)), color = "cornflowerblue")+
        geom_line(aes(y = maat), size = 2, color = "Black")+

```

```

      labs(x="Time of Day",
           y= "Arrival Type",
           title=paste("", "\n", "Phase
", ph, "\n", format(as.POSIXct(d1), format='%B %d, %Y')))+
      theme(plot.title = element_text(hjust = 0.5),
plot.subtitle=element_text( hjust=0.5))+
      theme(axis.text
=element_text(size=12), axis.title=element_text(size=14, face="bold"), plot.t
itle = element_text(size=22, face = "bold"))+
      scale_x_datetime(date_labels = "%H:%M", date_breaks = ("2 hour"),
limits = c(as.POSIXct(d1), as.POSIXct(d2)))+
      scale_y_continuous(breaks = seq(0, 6, 1))+
      geom_vline(xintercept = as.numeric(w2), linetype = 2)+
geom_vline(xintercept = as.numeric(w3), linetype = 2)+
      annotate("text", x = w2, y = 0, label = format(w2, "%H:%M"),
colour = "red")+
      annotate("text", x = w3, y = 0, label = format(w3, "%H:%M"), colour
= "red"))
      ggsave(filename = paste("At", d1, ph, ".png"), path = graphpath, width = 20,
height = 15, units = "cm")

All<-final4
print(All)

}

```

```

#5##below, place the number of the phase in which above variables will be
visualized between parentheses
#note# file named "ALL" contained all the values of the variables

```

```

All<-f(2)

```

```

#####

```

#### #### Purdue Coordination Diagram (PCD) ####

```

rm(list=ls(all=TRUE)) #Run, if all data needs to be cleared

```

```

#1#set the path bellow to where the data is located
datapath<-"C:/Users/Nawar/Documents"

```

```

#2# if needed, set the path bellow to where the graphs will be saved

```

```

graphpath<-"C:/Users/Nawar/Desktop/check/ped"

#3# uploads the whole data and subset the required one by running the
script below until #4#.
setwd(datapath)
library(plyr)
library(lubridate)
library(ggplot2)
library(TTR)
temp = list.files(path = datapath, pattern = "*.csv" )
data = do.call(rbind, lapply(temp, function(x) read.csv(x,
stringsAsFactors = FALSE, skip= 8, sep = ",", header = FALSE, col.names =
c("timestamp","event","phase") )))
data$timestamp<-strptime(data$timestamp, "%m/%d/%Y %H:%M:%OS")

#4# Follow points; "a" and "b" below, then run the code until #5#
#a#Set below which set of advanced detectors is going to be used for the
analysis.
#If it is the front, place "front", or if it is back set "back"
#Estimated operating speed is 45 mph (66) ft/sec

detcocation <- c("back")
operatingspeed = 66 #(ft/sec)

#b#Bellow, substitute the beginning and the end of required day in form
"yyyy-mm-dd" and run code until #5#
#1st date
d1<-"2016-05-23"
#2nd date
d2<-"2016-05-24"

all<-subset(data,timestamp >= d1 & timestamp < d2 )

f<-function(ph){
  coor<-subset(all, phase == 2 & event == 151)
  boc<-subset(all, (event == 1 | event == 8 | event == 10))
  b<-boc[-1,]
  a<-head(boc,-1)
  colnames(b)<-c("timestamp2","event2","phase2")
  c<-cbind(a,b)

  c$boc<-ifelse((c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase == 5
| c$phase == 6) &
                c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |
                (c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
                c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4|
c$phase2 == 7 | c$phase2 == 8)) |

```

```

(c$event==1 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
      c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4 |
c$phase2 == 7 | c$phase2 == 8)) |

(c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
      c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4 |
c$phase2 == 7 | c$phase2 == 8)) |

(c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
      c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4 |
c$phase2 == 7 | c$phase2 == 8)) |

(c$event==8 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
      c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4 |
c$phase2 == 7 | c$phase2 == 8)) |

(c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
      c$event2 == 1 & (c$phase2 == 3 | c$phase2 == 4 |
c$phase2 == 7 | c$phase2 == 8)) |

(c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
      c$event2 == 8 & (c$phase2 == 3 | c$phase2 == 4 |
c$phase2 == 7 | c$phase2 == 8)) |

(c$event==10 & (c$phase == 1 | c$phase == 2 | c$phase ==
5 | c$phase == 6) &
      c$event2 == 10 & (c$phase2 == 3 | c$phase2 == 4 |
c$phase2 == 7 | c$phase2 == 8)),
      "boc", "")

```

```

d<-c[which(c$boc=="boc"),]
e<-d[,1:3]
colnames(e)<-c("eoc", "event2", "phase2")
e<-e[-1,]
f<-d[,4:6]
colnames(f)<-c("boc", "event", "phase")
f<-head(f,-1)
g<-cbind(f,e)
g$NOC<-(1:length(g$boc))
# one sec is added for the correction
g$cyclelength<-difftime(g$eoc,g$boc, units = "secs")+as.numeric(1, units
= 'secs')
# #####
length(g[which(g$event != 1 | g$event2 != 10),7]) # number of the cycle
length that has been removed

```

```

g[which(g$event != 1 | g$event2 != 10),7] ## number of the cycles that
have been removed

# Final file of the cycle length
final<-g[which(g$event == 1 & g$event2 == 10),c(1,7,8)]

##### -----below is green time code ----- #####

dd<-merge(boc,g, by.x = "timestamp", by.y = "boc", all.x = T)
dd<-tail(dd,-which(dd$NOC == 1)[1]+1)
f4 <- function(x, blank = is.na) {
  # Find the values
  if (is.function(blank)) {
    isnotblank <- !blank(x)
  } else {
    isnotblank <- x != blank
  }
  # Fill down
  x[which(isnotblank)][cumsum(isnotblank)]
}

dd$NOC<-f4(dd$NOC)
new<-subset(dd,phase.x == ph & (event.x == 1 | event.x == 8))
for(i in 1:length(new$event.x)) {
  new$check[i]<-ifelse((new$event.x[i] == 1 & new$event.x[i+1] == 8 &
new$NOC[i]==new$NOC[i+1]),"T","F")
}

# The number of cycles that we couldn't calculate the GT (regardless of
the cycle length errors)
length(which(new$event.x == 1 & new$check == "F"))
#which cycles having the errors
new$NOC[which(new$event.x == 1 & new$check == "F")]

new1<-cbind(new[which(new$event.x == 1 & new$check == "T"),c(1,2)],
            new[which(new$event.x == 1 & new$check == "T")+1,c(1:3,9)])

new1$starLT<-as.numeric(2,units = "secs")
new1$clearADD<-as.numeric(2,units = "secs")
new1$BOGeffective<-new1[,1]+new1$starLT
new1$EOGeffective<-new1[,3]+new1$clearADD

new1$greentime<-difftime(new1$EOGeffective,new1$BOGeffective, units =
"secs")

###Final File##

```



```

final1<-merge(final,new1, by.x = "NOC", by.y = "NOC", all.x =
T) [,c("boc", "NOC", "cyclelength", "phase.x", "BOGeffective", "greentime")]
final1$gC <- as.numeric(final1$greentime)/as.numeric(final1$cyclelength)

```

```

##### -----below is PCD code -----#####
dd<- merge(all,g, by.x = "timestamp", by.y = "boc", all.x = T)
dd[,c(4:8,10:11)]<-NULL
colnames(dd)<-c("timestamp", "event", "phase", "NOC")
dd<-tail(dd,-which(dd$NOC == 1)[1]+1)
f4 <- function(x, blank = is.na) {
  # Find the values
  if (is.function(blank)) {
    isnotblank <- !blank(x)
  } else {
    isnotblank <- x != blank
  }
  # Fill down
  x[which(isnotblank)][cumsum(isnotblank)]
}

dd$NOC<-f4(dd$NOC)

pp<-subset(dd, (event == 8 & phase == ph) | (event == 1 & phase == ph))
p1<-as.data.frame(pp[which(pp$event == 8),])
p2<-as.data.frame(pp[which(pp$event == 8)+1,])
p3<-as.data.frame(pp[which(pp$event == 8)+2,])

p4<-cbind(p1,p2,p3)
p4$check<-p4[,4]==p4[,8]-1 & p4[,4]==p4[,12]-1 # check 1st NOC col is
equal to x & 2nd and 3rd NOC col are equal to x+1
p4$check2<-p4[,2]== 8 & p4[,6] == 1 & p4[,10] == 8 ## check 1st phase
col is equal to 8 &, 2nd = 8, 3rd = 1
p4$checkall<- ifelse(p4$check == T & p4$check2 == T,T,F)
# remove rows that don't match the above condition
p5<-p4[which(p4$checkall == T),]

# adjusting the actual timestamp of beginning and end of green with 2
sec

p5[,1]<-p5[,1]+as.numeric(2, units ="secs")
p5[,5]<-p5[,5]+as.numeric(2, units ="secs")
p5[,9]<-p5[,9]+as.numeric(2, units ="secs")

# creating relative locations
p5$RL1<-p5[,1]-p5[,1]
p5$RL2<-p5[,5]-p5[,1]
p5$RL3<-p5[,9]-p5[,1]

```

```

p6<-rbind(p5[,c(1:4)],p5[,c(9:12)])
p6<-p6[order(p6$timestamp),]
vol<-p6[duplicated(p6$NOC),]

```

```

v<-if (vol$phase[1] == 1 & detcocation == "front") {subset(dd,event ==
82 & phase == 1)
} else if (vol$phase[1] == 1 & detcocation == "back") {subset(dd,event
== 82 & phase == 2)
} else if (vol$phase[1] == 2 & detcocation == "front") {subset(dd,event
== 82 & phase == 9)
} else if (vol$phase[1] == 2 & detcocation == "back") {subset(dd,event
== 82 & phase == 10)
} else if (vol$phase[1] == 3 & detcocation == "front") {subset(dd,event
== 82 & phase == 3)
} else if (vol$phase[1] == 3 & detcocation == "back") {subset(dd,event
== 82 & phase == 4)
} else if (vol$phase[1] == 4 & detcocation == "front") {subset(dd,event
== 82 & phase == 14)
} else if (vol$phase[1] == 4 & detcocation == "back") {subset(dd,event
== 82 & phase == 15)
} else if (vol$phase[1] == 5 & detcocation == "front") {subset(dd,event
== 82 & phase == 5)
} else if (vol$phase[1] == 5 & detcocation == "back") {subset(dd,event
== 82 & phase == 6)
} else if (vol$phase[1] == 6 & detcocation == "front") {subset(dd,event
== 82 & phase == 19)
} else if (vol$phase[1] == 6 & detcocation == "back") {subset(dd,event
== 82 & phase == 20)
} else if (vol$phase[1] == 7 & detcocation == "front") {subset(dd,event
== 82 & phase == 7)
} else if (vol$phase[1] == 7 & detcocation == "back") {subset(dd,event
== 82 & phase == 8)
} else if (vol$phase[1] == 8 & detcocation == "front") {subset(dd,event
== 82 & phase == 24)
} else if (vol$phase[1] == 8 & detcocation == "back") {subset(dd,event
== 82 & phase == 25)
}else ("wrong")

```

```

if (v$phase[1] == 1 | v$phase[1] == 3 | v$phase[1] == 5 | v$phase[1] ==
7 ) {v$timestamp<- v$timestamp + as.numeric(36/operatingspeed , units =
"secs")
}else if (v$phase[1] == 2 | v$phase[1] == 4 | v$phase[1] == 6 |
v$phase[1] == 8 ) {v$timestamp<- v$timestamp +
as.numeric(100/operatingspeed , units = "secs")
}else if (v$phase[1] == 9 | v$phase[1] == 14 | v$phase[1] == 19 |
v$phase[1] == 24 ) {v$timestamp<- v$timestamp +
as.numeric(86/operatingspeed , units = "secs")
}else if (v$phase[1] == 10 | v$phase[1] == 15 | v$phase[1] == 20 |
v$phase[1] == 25 ) {v$timestamp<- v$timestamp +
as.numeric(185/operatingspeed , units = "secs")}

```

```

a<-rbind.fill(v,vol)
b<-a[order(a$timestamp),]

m<-data.frame(a = 0 , b =0, c=0,d=0,e=0,f=0)
m<-m[FALSE,]
wrong1<-data.frame(st = 0, nd=0)
wrong2<-data.frame(st = 0, nd=0)
l=1
n=1
s=1
k=0
for(i in head(unique(vol$NOC),-1)) {
  if (((b[which(b$event == 8),4][s])+1 == (b[which(b$event ==
8),4][s+1])) & (which(b$event == 8)[s+1]-which(b$event == 8)[s]>1)) {
    len<-length((b[(which(b$event == 8 & b$NOC ==i)):which(b$event ==
8 & b$NOC ==i+1)-1))[-1,])[1])
    for (j in 1:len){
      m[k+1,]<-(b[(which(b$event == 8 & b$NOC ==i)):which(b$event == 8
& b$NOC ==i+1)-1))[-1,])[j,]
      m[k+1,5]<-(b[which(b$event == 8 & b$NOC ==i),1])
      m[k+1,6]<-difftime(((b[(which(b$event == 8 & b$NOC
==i)):which(b$event == 8 & b$NOC ==i+1)-1))[-1,])[j,1]) ,
(b[which(b$event == 8 & b$NOC ==i),1]), units = "secs")

      k=k+1
    }
  } else if ((b[which(b$event == 8),4][s])+1 != (b[which(b$event ==
8),4][s+1])) {
    wrong1[n,1]<-(b[which(b$event == 8),4][s])
    wrong1[n,2]<-(b[which(b$event == 8),4][s+1])
    n=n+1
  } else if (which(b$event == 8)[s+1]-which(b$event == 8)[s]<=1) {
    wrong2[l,1]<-(b[which(b$event == 8)[s],4])
    wrong2[l,2]<-(b[which(b$event == 8)[s+1],4])
    l=l+1
  }

  s=s+1
}
m$a<-as.POSIXct(m$a,origin="1970-01-01")
m$e<-as.POSIXct(m$e,origin="1970-01-01")

m$g<-difftime(m$a,m$e, units = "secs")

#### calculating Arrivals on Green ###
x1<-p5[which(p5$checkall == T),]
a1<-rbind.fill(x1[,1:4],x1[,5:8])
a1<-a1[order(a1$timestamp),]

b1<-m[,c(1,2,3,4)]

```

```

colnames(b1)<-c("timestamp","event","phase","NOC" )
c2<-rbind.fill(a1,b1)
c2<-c2[order(c2$timestamp),]

mmm<-data.frame(a = 0, b = 0, c = 0, d= 0, OnRed= 0, OnGreen=0)

for(i in 1:(length(which(c2$event == 1))-1)){
  NOC1<-c2[c2$event == 8,4][i]
  NOC2<-c2[c2$event == 8,4][i+1]
  if (NOC1+1 == NOC2) {
    group<- c2[which(c2$event == 8)[i]:which(c2$event == 8)[i+1],]
    mmm[i,1:2]<-head(group,1)[c(1,4)]
    mmm[i,3:4]<-tail(group,1)[c(1,4)]
    mmm[i,5]<- which(group$event == 1)- which(group$event == 8)[1] - 1
#On red
    mmm[i,6]<- which(group$event == 8)[2] - which(group$event == 1) - 1
#On green
  }
}

mmm$a<-as.POSIXct(mmm$a, origin="1970-01-01")
mmm$c<-as.POSIXct(mmm$c, origin="1970-01-01")
colnames(mmm)<-
c("eventtime8_1","NOC1","eventtime8_2","NOC2","OnRed","OnGreen")

mmm$Vtotal<-mmm$OnRed+mmm$OnGreen

Arival<-merge(finall,mmm, by.x = "NOC", by.y = "NOC2", all.x = T)
Arival<-na.omit(Arival[,c("NOC","boc"
,"cyclelength","greentime","gC","OnRed","OnGreen","Vtotal" )])
Arival$PlatoonRatio<-Arival$OnGreen*Arival$gC
Arival<-Arival[Arival$Vtotal !=0,]
Arival$POG<-Arival$OnGreen/Arival$Vtotal

####Information on the Graph

w2<-coor[which(coor$event == 151)[1],1] ## Timestamp when Plan 2 begins
w3<-coor[tail(which(coor$event == 151))[6],1] ## Timestamp when Plan 2
ends

AoG_p1 = round(sum( Arival[Arival$boc < w2 | Arival$boc > w3 ,
"OnGreen"])/sum( Arival[Arival$boc < w2 | Arival$boc > w3 , "Vtotal"]),
digits = 2) *100# % Arivals on Green Plan 1
AoG_p2 = round(sum( Arival[Arival$boc >= w2 & Arival$boc <= w3 ,
"OnGreen"])/sum( Arival[Arival$boc >= w2 & Arival$boc <= w3 , "Vtotal"]),
digits = 2) *100# % Arivals on Green Plan 2

GT_p1<-round(mean(Arival[Arival$boc < w2 | Arival$boc >= w3 ,5])* 100,
digits = 0) ## Avaerage Of the Green time of plan 2

```

```

GT_p2<-round(mean(Arival[Arival$boc > w2 & Arival$boc <= w3 ,5])* 100,
digits = 0) ## Avaerage Of the Green time of plan 2

w4<-d$timestamp[1]
w5<-w4+(w2-w4)/2
w6<-w2+(w3-w2)/2

###Visualizing PCD###

ggplot(p5,aes(x = timestamp))+geom_line(aes (y = as.numeric(RL2)), color
= "green", size = 0.5)+
  geom_line(aes (y = as.numeric(RL3), ) , color = "red",size = 0.5)+
  geom_point(data = m,aes(a,f),alpha = 3/5,size=0.5)+
  labs(x="Time of Day",
       y= "Time in Cycle",
       title=paste("Phase",ph,"\n",format(as.POSIXct(d1),format='%B %d,
%Y')))+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(axis.text
=element_text(size=14),axis.title=element_text(size=14,face="bold"),plot.t
itle = element_text(size=14, face = "bold"))+
  scale_x_datetime(date_labels = "%H:%M",date_breaks = ("2 hour"),
limits = c(as.POSIXct(d1),as.POSIXct(d2)))+
  geom_vline(xintercept = as.numeric(w2), linetype = 2, size = 1 , color
= "blue")+ geom_vline(xintercept = as.numeric(w3), linetype = 2, size = 1
, color = "blue")+
  annotate("text",x = w2, y = 0, label = format(w2,"%H:%M"), colour =
"red",size = 4.5)+annotate("text",x = w3, y = 0, label =
format(w3,"%H:%M"),colour = "red",size = 4.5)+
  annotate("text",x = w5, y = 140, label = paste("Plan 1","\n",
"Arrivals on Green (AOG) =",AoG_p1,"%","\n", " AVG. of
g/C",GT_p1,"%"),colour = "red",size = 4.5)+
  annotate("text",x = w6, y = 140, label = paste("Plan 2","\n",
"Arrivals on Green (AOG) =",AoG_p2,"%","\n", " AVG. of
g/C",GT_p2,"%"),colour = "red",size = 4.5)
  ggsave(filename = paste("PCD",d1,"phase",ph,".png"), path =
graphpath,width =25, height = 20, units = "cm")

PCD<-p5
print(p5)

}
#5##below, place the number of the phase which above variables will be
visualized between parentheses
#note# file named "PCD" is final one

PCD<-f(2)

```