Detailed Analysis of Roadway Users Interactions at Intersections with Flashing Yellow Arrows: A Streamlined Approach to Data Collection

SAFETY RESEARCH USING SIMULATION

UNIVERSITY TRANSPORTATION CENTER

Kelvin R. Santiago-Chaparro, Ph.D.

Assistant Researcher

Department of Civil and Environmental Engineering

University of Wisconsin-Madison

Detailed Analysis of Roadway Users Interactions at Intersections with Flashing Yellow Arrows:
A Streamlined Approach to Data Collection

Kelvin R. Santiago-Chaparro
Assistant Researcher
Department of Civil and Environmental Engineering
University of Wisconsin-Madison
https://orcid.org/0000-0001-6897-0351

Hiba Nassereddine
Graduate Research Assistant
Department of Civil and Environmental Engineering
https://orcid.org/0000-0001-5277-9464

David A. Noyce, Ph.D.
Professor
Department of Civil and Environmental Engineering
University of Wisconsin-Madison
https://orcid.org/0000-0001-5887-8391

A Report on Research Sponsored by

SAFER-SIM University Transportation Center

Federal Grant No: 69A3551747131

August 2021

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
UW-1-Y3		

4. Title and Subtitle Detailed Analysis of Roadway Users Interactions at Intersections with Flashing Yellow Arrows: A Streamlined Approach to Data Collection	**5. Report Date** August 2021 **6. Performing Organization Code**
7. Author(s) Kelvin R. Santiago-Chaparro https://orcid.org/0000-0001-6897-0351 Hiba Nassereddine https://orcid.org/0000-0001-5277-9464 David A. Noyce, Ph.D. https://orcid.org/0000-0001-5887-8391	**8. Performing Organization Report No.**
9. Performing Organization Name and Address University of Wisconsin-Madison Department of Civil & Environmental Engineering 1415 Engineering Drive Madison, WI 53706	**10. Work Unit No.** **11. Contract or Grant No.** Safety Research Using Simulation (SAFER-SIM) University Transportation Center (Federal Grant #: 69A3551747131)
12. Sponsoring Agency Name and Address Safety Research Using Simulation University Transportation Center Office of the Secretary of Transportation (OST) U.S. Department of Transportation (US DOT)	**13. Type of Report and Period Covered** Final Research Report (August 2019 – August 2021) **14. Sponsoring Agency Code**

15. Supplementary Notes

This project was funded by Safety Research Using Simulation (SAFER-SIM) University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program.

16. Abstract

Quantifying vehicle-pedestrian interactions in simple terms is achievable with the availability of the correct dataset. The feasibility of such approach has been demonstrated by previous work. However, the procedures required for obtaining the required datasets are labor-intensive and difficult to implement on a large scale. This challenge in acquisition of the necessary datasets makes having a detailed understanding of vehicle-pedestrian interactions difficult as well as using that detailed understanding to evaluate the effectiveness of countermeasures such as the flashing yellow arrow on the safety of the interactions prior to the availability of crash data. In other words, streamlined data acquisition procedures are key to the use of vehicle-pedestrian interactions metrics to conduct proactive safety evaluations. The research effort documented ahead introduces a set of tools that can streamline the process of collecting the required data to study vehicle-pedestrian interactions both in the field as well as through driving simulation experiments.

17. Key Words	18. Distribution Statement

<table>
<tr><td colspan="2">Highways; Operations and Traffic Management; Safety and Human Factors</td><td colspan="3">No restrictions. This document is available through the SAFER-SIM website, as well as the National Transportation Library</td></tr>
<tr><td>19. Security Classif. (of this report)
Unclassified</td><td colspan="2">20. Security Classif. (of this page)
Unclassified</td><td>21. No. of Pages
37</td><td>22. Price</td></tr>
<tr><td>Form DOT F 1700.7 (8-72)</td><td colspan="4">Reproduction of completed page authorized</td></tr>
</table>

Table of Contents

List of Figures

Abstract

Quantifying vehicle-pedestrian interactions in simple terms is achievable with the availability of the correct dataset. The feasibility of such approach has been demonstrated by previous work. However, the procedures required for obtaining the required datasets are labor-intensive and difficult to implement on a large scale. This challenge in acquisition of the necessary datasets makes having a detailed understanding of vehicle-pedestrian interactions difficult as well as using that detailed understanding to evaluate the effectiveness of countermeasures such as the flashing yellow arrow on the safety of the interactions prior to the availability of crash data. In other words, streamlined data acquisition procedures are key to the use of vehicle-pedestrian interactions metrics to conduct proactive safety evaluations. The research effort documented ahead introduces a set of tools that can streamline the process of collecting the required data to study vehicle-pedestrian interactions both in the field as well as through driving simulation experiments.

1 Introduction

A permissive left turn is a complex driving maneuver. When only vehicles are involved, a left turning driver must examine the opposing traffic stream and estimate the available gaps using their best judgement about the speed of the opposing vehicle and distance between the vehicles. When the permissive maneuver is complicated even further by the presence of a conflicting pedestrian the mental workload required for safely completing the maneuver is increased because a driver must judge the opposing vehicle stream as well as the corresponding conflicting pedestrian stream. These types of maneuvers can become more dangerous if there is not a clear understanding of the permitted nature of the movement and the need for the driver to yield to conflicting road users, a scenario that far too often results in crashes which can have unfortunate results including, in the most extreme situations, the loss of life.

Therefore, transportation agencies need to find ways to communicate effectively with drivers the type of left turn movement they are completing, i.e., permissive or permitted. The flashing yellow arrow (FYA) indication has been used over the years by agencies to improve the safety and efficiency of signalized left turns during permissive operation whether entirely permissive or part of a signal sequence that involves both a permissive and permitted maneuver. A core principle behind the use of FYA indications is the reinforcement of the message communicated to left turning drivers regarding the need to wait for a gap in the opposing traffic stream before crossing. Numerous research efforts have studied and continue to study the impact of FYA on the safety and operation of left turns at signalized intersections [1]–[3]. Furthermore, the use of FYA is no longer limited to left turns; in fact, transportation agencies have started to use FYA indications on right turn maneuvers to improve the safety of vehicle-pedestrian interactions.

Recently conducted research [4] demonstrated the feasibility of modeling vehicle pedestrian interactions by focusing on two key predictor variables: the presence of a right turn FYA indication and the relative position of the pedestrian within the crosswalk. The response variable was the deviation from expected behavior which was reported in terms of time for the vehicle to complete a turning maneuver. Arguably, the deviation from expected behavior could be used as an indicator of safety benefits of a specific treatment such as a FYA because the additional time that drivers take to complete a maneuver when a pedestrian is present (compared to when no pedestrian is present) could be treated as an indicator that they recognize the potential danger of the interaction. Therefore, the modeling approach can be used to compare the behaviors at similar sites with and without a treatment to study the effectiveness of a treatment.

A limitation of the recently conducted research is that the procedures used to obtain the necessary datasets were largely manual and labor-intensive in nature. As a result, scaling the procedures to support a detailed analysis is not feasible and there is a need for more streamlined procedures that take advantage of advanced vehicle detection technologies or driving simulation platforms to support similar evaluations on existing infrastructure or in a laboratory environment in the case of non-existent infrastructure.

1.1 Objectives and Contribution

The key objective of the research effort documented in this report was establishing procedures and tools that can be used to streamline the process of studying vehicle-pedestrian interactions on the field or using a driving simulation environment. The foundation for the field procedure is a streamlined version of a trajectory classification algorithm that enables extracting the detailed trajectory of turning vehicles using dynamically generated machine learning models for a specific site. On the other hand, the foundation behind conducting evaluation using driving simulation is a collection of tools that can streamline the process of creating driving simulations that involve realistic signal operations. The collection of tools include signal heads, ambient traffic controls, and a data streaming control used to facilitate data collection.

For both vehicle-pedestrian interaction evaluation approaches, i.e. field and driving simulation-based, the nature of the data that can be obtained is similar to that used in the previously-mentioned and recently conducted research [4] that demonstrated the feasibility of modeling vehicle-pedestrian interactions as well as the potential for using the resulting model parameters as an indicator of the effectiveness of a treatment on the safety of the interactions.

1.2 Structure of the Report

Each chapter of this report summarizes information key to the previously outlined objectives. Chapter 2 provides background information that is pertinent to surrogate safety measures which are key to discussions that involve proactive safety evaluations as well as a more detailed discussion of the previously completed research efforts on which this effort expanded. Chapter 3 introduces a methodology to classify vehicle trajectory data by movement by taking advantage of machine learning algorithms, which represents an improved approach from current practice. The purpose of the improved classification approach is to eliminate the need for manually documenting the timestamp associated with key vehicle positions, a crucial component of the previously defined modeling approach. Chapter 4 describes a collection of objects created with the specific goal of streamlining the creation of driving simulation scenarios that include complex signal sequences such as the FYA. Finally, Chapter 5 contains a summary of findings, conclusions, and future work.

2 Background Information

A key objective of the research effort outlined in this report involves the creation of tools that can streamline the process of obtaining the data required to model vehicle-pedestrian interactions using a previously defined [4] modeling approach. For purposes of supporting the discussions ahead, a summary of the mentioned modeling approach is presented in the sections ahead.

2.1 Definition of Key Positions in Modeling Vehicle-Pedestrian Interactions

A vehicle making a right turn must cross the locations shown in Figure 2.1 as lines A, B, and C. Location A is important for vehicles that come to a stop prior to initiating a right turn. The timestamps when right turning vehicles reach locations B and C can be used to determine the time it takes a vehicle to complete the right turning maneuver. That time is affected by the presence of a pedestrian walking along the line associated with Location C. Along the line associated with Location C there is a point where the path of the right turning vehicle and the path of the pedestrian meet.

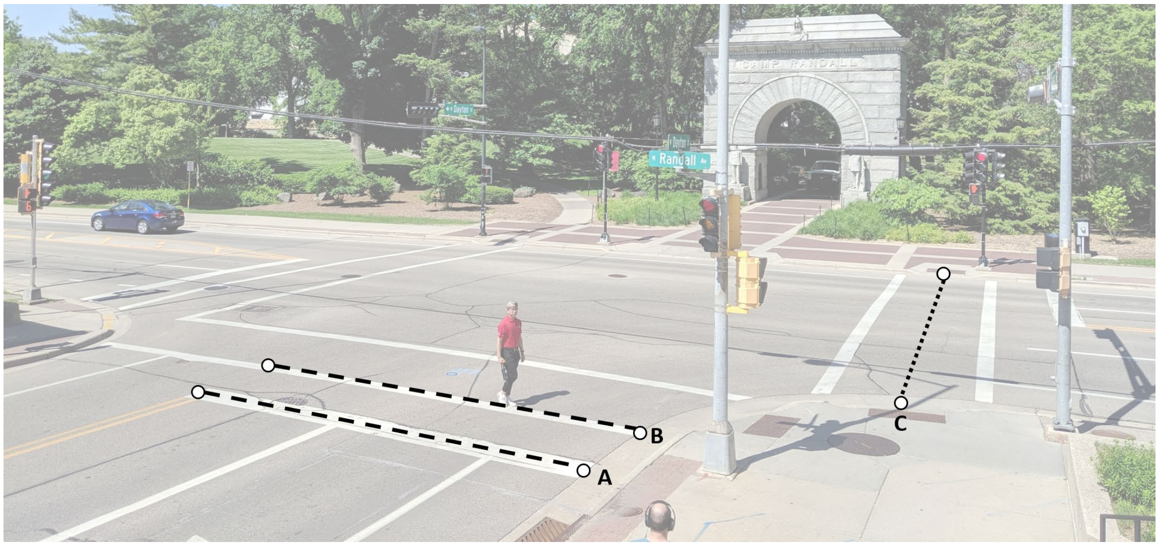


Figure 2.1 Key Vehicle Positions to Model Vehicle-Pedestrian Interactions

At a given intersection, an average time for completing a right turn can be measured using field observations. For right turns completed when there is a pedestrian traveling along the Location C the difference in time to complete the right turn against the average time to complete the right turn when no pedestrian is present can be computed. Previously completed research [4] found that the difference in time can be modeled as a function of the time it would take a pedestrian to reach the point where the theoretical path of the vehicle and the pedestrian meet, i.e., the theoretical conflict point.

2.2 Modeling Time to Complete Turning Maneuver

An example of previous results demonstrating the use of a segmented regression model is shown in Figure 2.2. The predictor variable of the model is the time (in seconds) for a pedestrian to reach the theoretical conflict point and the response variable is the log-transformed value of the difference in time between the completion of a right turn when a pedestrian is present and the average time at the site to complete a right turn when no pedestrian is present.

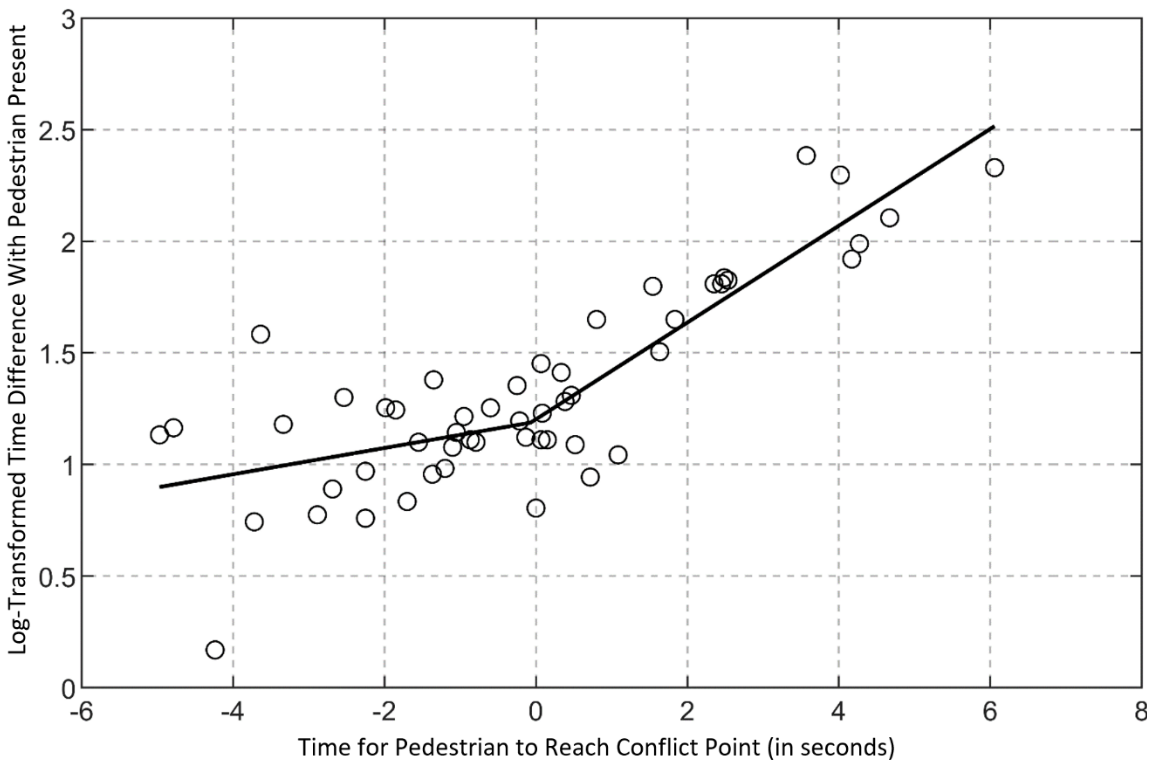


Figure 2.2 Segmented Regression Model

The segmented nature of the model is based on the observed behaviors that drivers react in different ways depending on whether a pedestrian is approaching a conflict point or moving away from the conflict point. With sufficient observations, similar models could be developed for individual sites.

2.3 Using Model Results as Indicators for Safety Evaluations

The time it takes a vehicle to complete a right turn maneuver when a pedestrian is using the crosswalk can be treated as an indicator of safety. If data like the one described by the model shown in Figure 2.2 is available on a site before and after the implementation of a countermeasure, the separation between the model lines can be used as an indicator of the effectiveness of the countermeasure. If the model lines for the site after implementation of the countermeasure is above the model lines for the site prior to the implementation of the countermeasure then an argument can be made that the countermeasure was successful. This simple concept, as is the case with other more direct surrogate safety measures, arguably makes it possible to evaluate the safety of treatments without having to wait for the availability of crash data.

2.4 Improvements Needed to Process

One key roadblock to the use of proposed methodology to evaluate the impact that a safety treatment has on the behavior of road users without having to wait for the availability of crash data is the need for datasets that require labor-intensive procedures. Furthermore, if a safety countermeasure is not existing on the field, creating the necessary driving simulations is also a labor-intensive process that can benefit from streamlining. Based on these needs, a method of identifying vehicle trajectories that can be used to quantify the time for a vehicle to complete a right turn is presented along with a set of driving simulation tools that can be used to streamline the process of creating scenarios to conduct experiments that can produce similar datasets.

3 Trajectory Data Collection

A streamlined approach to quantifying vehicle-pedestrian interactions requires the availability of vehicle trajectory data and pedestrian position. Numerous tools can be used to obtain pedestrian positions thus the focus of the sections ahead and describing a process to classify vehicle trajectory data from an intersection approach into the corresponding movement thus allowing the calculation of timestamps associated with the key positions that are required to quantify the vehicle-pedestrian interaction.

The focus of this chapter is on explaining the use of a machine-learning approach to improve or re-define an existing classification approach for trajectory data obtained from a commercially available radar-based vehicle detection system. Therefore, an important aspect of the procedures proposed ahead is the reliance on existing technology thus going beyond an academic exercise and proposing an approach that makes the process of bringing theory to practice feasible. The sections ahead describe the collection of vehicle volume information from video, then a discussion of trajectory data collected during the same period is presented, an explanation then follows on how a trajectory classification model was developed, and finally the performance of the classification model and the feasibility of using the results for quantifying vehicle-pedestrian interactions is presented.

3.1 Video Dataset Collection and Characteristics

Video from the East Northland Avenue and North Meade Street intersection in Appleton, WI (Latitude = 44.287226, Longitude = -88.395792) was manually reviewed using a frame-by-frame approach. A total of three hours of video were analyzed. Through the manual review process, timestamps associated with vehicles crossing the Y_{Bar} position shown in Figure 3.1 were documented. Y_{Bar} is a horizontal asymptote on the approach shown that can be considered the point on the approach that a vehicle can move to without interrupting the path of vehicles from another approach. Of particular interest were the vehicles in Lane A, a shared lane for thru and right movements. For each timestamp documented, a corresponding movement (right or thru) was assigned.

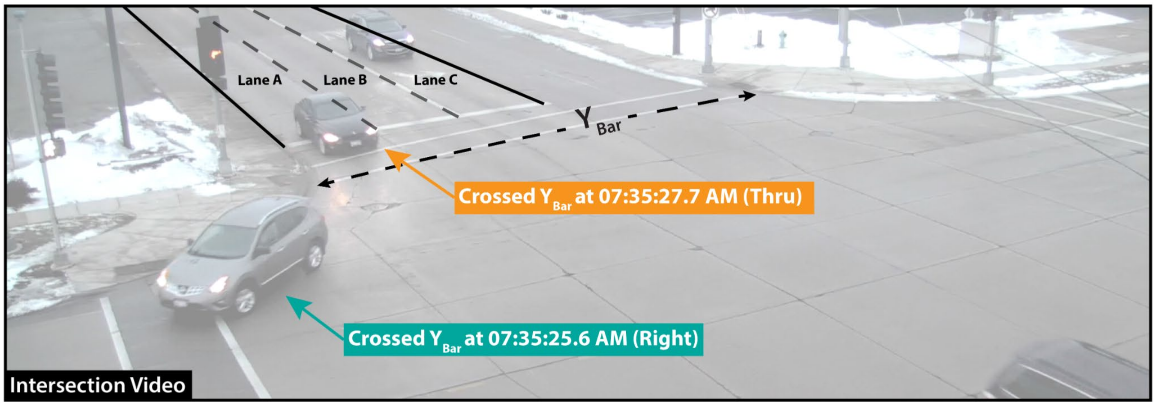


Figure 3.1 Intersection Characteristics and Information Logged for Vehicles from Video

The result of the manual review process is a dataset containing the timestamp in the video when a vehicle crosses the Y_{Bar} as well as the corresponding movement. The information collected was key to understanding the performance of the classification methodology described in the sections ahead.

3.2 Vehicle Trajectory Data Characteristics and Applications

Vehicle trajectory data was collected during the same period covered by the manually reviewed video of the intersection. The data was collected using a previously documented [5] data collection approach and noise removal procedures. The data collection procedure creates a dataset containing a vehicle identifier, position (x and y coordinates), speed, and vehicle length every 0.5 seconds. A visualization of the dataset collected is shown in Figure 3.2. The figure is not meant to contain all the data included in the dataset, i.e., only sufficient data to describe the characteristics of the dataset is shown.

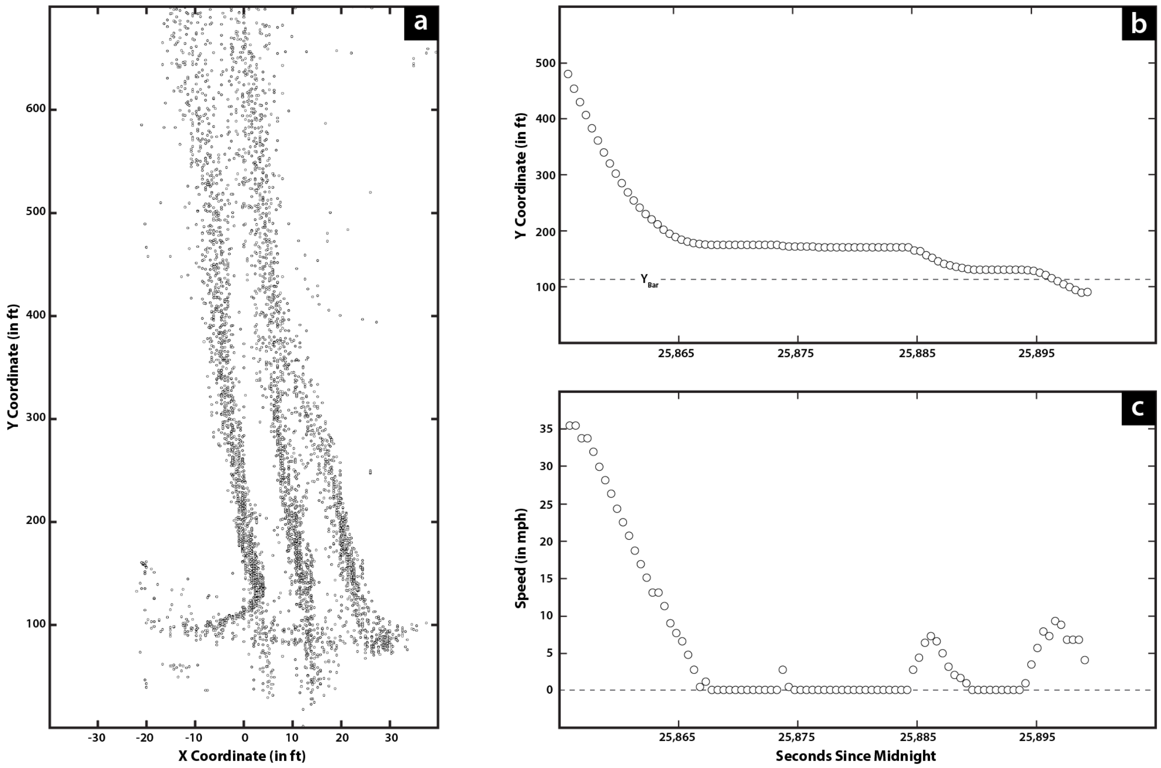


Figure 3.2 Visualization of Trajectory Dataset

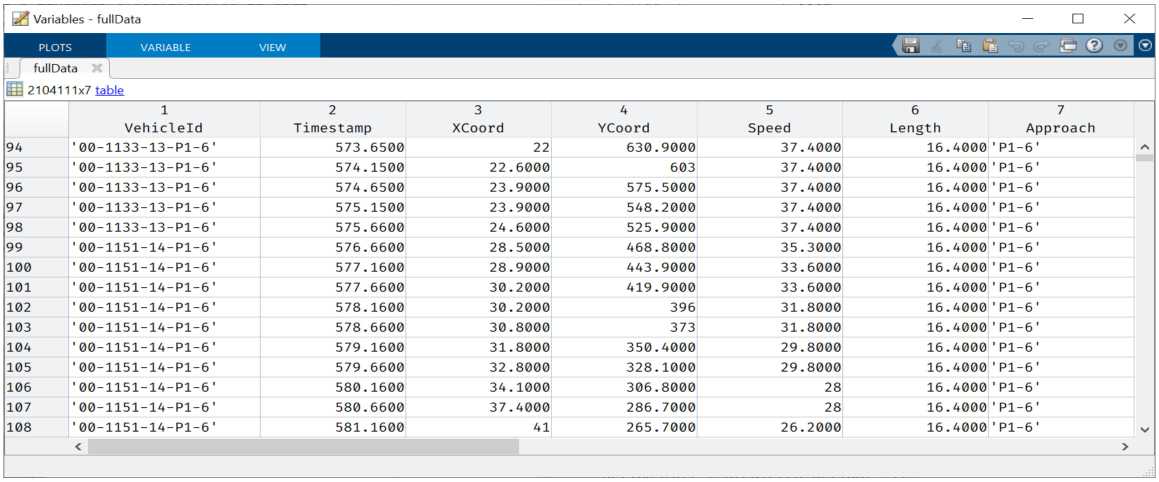
Figure 3.2a shows the x-y coordinate for each trajectory point and thus resembles the top view of an intersection. It should be noted that when the line of sight of the radar sensor responsible for collecting the data is aligned with the centerline of the approach monitored, the Y coordinate in the dataset can be used as an indicator of the position of the vehicle with respect to the stop bar of the approach, which is located at a fixed Y coordinate. Therefore trajectory-derived profiles like the ones shown in Figure 3.2b and Figure 3.2c can be obtained, and more importantly, the behavior described in the profile can be extracted from the dataset. This means that if the trajectory points associated with a specific vehicle are available for analysis, the timestamps associated with key vehicle positions used to model vehicle-pedestrian interactions can be obtained using a systematic approach.

The challenge with the data available is the need to isolate trajectories associated with a specific movement, e.g., right-turning identifying the trajectory of right turning vehicles. Previous efforts proposed a solution to this problem that relies on a collection of rules [6]. The approach presented in the section ahead relies on machine learning principles. Specifically, the proposed classification approach relies on the creation of a machine learning model created specifically for each data collection site thus providing a tailored classification approach.

3.3 Classification Model

Different types of classifiers were tested for accuracy and after various iterations, a support vector machine (SVM) classifier was selected as the best option. As a starting point, the development of the classifier starts with raw (no removal of noise) trajectory data. The raw data was converted from a SQLite format into a MAT format to simplify the analysis process using MATLAB. The screenshot in Figure 3.3 shows the structure of the trajectory data.

A key aspect of the dataset shown is that the data points associated with a specific vehicle can be isolated for analysis using the *VehicleId* column. The raw trajectory data then undergoes noise removal procedures based on previously established procedures [5]. Removal of the noise does not affect the structure of the data shown in Figure 3.3.



	1 VehicleId	2 Timestamp	3 XCoord	4 YCoord	5 Speed	6 Length	7 Approach
94	'00-1133-13-P1-6'	573.6500	22	630.9000	37.4000	16.4000	'P1-6'
95	'00-1133-13-P1-6'	574.1500	22.6000	603	37.4000	16.4000	'P1-6'
96	'00-1133-13-P1-6'	574.6500	23.9000	575.5000	37.4000	16.4000	'P1-6'
97	'00-1133-13-P1-6'	575.1500	23.9000	548.2000	37.4000	16.4000	'P1-6'
98	'00-1133-13-P1-6'	575.6600	24.6000	525.9000	37.4000	16.4000	'P1-6'
99	'00-1151-14-P1-6'	576.6600	28.5000	468.8000	35.3000	16.4000	'P1-6'
100	'00-1151-14-P1-6'	577.1600	28.9000	443.9000	33.6000	16.4000	'P1-6'
101	'00-1151-14-P1-6'	577.6600	30.2000	419.9000	33.6000	16.4000	'P1-6'
102	'00-1151-14-P1-6'	578.1600	30.2000	396	31.8000	16.4000	'P1-6'
103	'00-1151-14-P1-6'	578.6600	30.8000	373	31.8000	16.4000	'P1-6'
104	'00-1151-14-P1-6'	579.1600	31.8000	350.4000	29.8000	16.4000	'P1-6'
105	'00-1151-14-P1-6'	579.6600	32.8000	328.1000	29.8000	16.4000	'P1-6'
106	'00-1151-14-P1-6'	580.1600	34.1000	306.8000	28	16.4000	'P1-6'
107	'00-1151-14-P1-6'	580.6600	37.4000	286.7000	28	16.4000	'P1-6'
108	'00-1151-14-P1-6'	581.1600	41	265.7000	26.2000	16.4000	'P1-6'

Figure 3.3 Sample Raw Trajectory Data

A de-noised version of the trajectory data is then summarized. The summarized dataset describes information about each unique vehicle and the sample structure of the dataset is shown in Figure 3.4. Of particular interest in the summarized dataset is the information that describes the timestamps when a vehicle crosses the Y_{Bar} location as well as information about key coordinates in the trajectory such as the last position recorded for a vehicle, an indicator of the number of 0.5 second intervals the vehicle spent stopped, direction of travel, and acceleration downstream of the Y_{Bar} location.

Variables - summaryData

PLOTS VARIABLE VIEW

summaryData

1402x11 table

	1 VehicleId	2 Timestamp	3 FirstX	4 LastX	5 FirstY	6 LastY	7 DeltaX	8 DeltaY	9 Acceleration	10 StoppedCount	11 ExitAzimuth
1	'06-6412-61-P3-8'	2.4843e+04	23	24.9000	108.6000	67.6000	1.9000	-41	2.1143	0	177.3467
2	'06-6477-11-P3-8'	2.4847e+04	22.3000	24.9000	109.6000	76.1000	2.6000	-33.5000	-0.7200	0	175.5621
3	'06-6517-17-P3-8'	2.4888e+04	1.6000	1	108.9000	98.4000	-0.6000	-10.5000	-2.6667	0	183.2705
4	'06-6534-23-P3-8'	2.4920e+04	25.6000	26.2000	104.3000	77.4000	0.6000	-26.9000	-0.9000	1	178.7222
5	'06-6538-24-P3-8'	2.4918e+04	27.6000	33.5000	107.9000	75.8000	5.9000	-32.1000	0	1	169.5852
6	'06-6556-30-P3-8'	2.4915e+04	-2.3000	-4.6000	108.3000	91.5000	-2.3000	-16.8000	-0.1000	1	187.7956
7	'06-6562-31-P3-8'	2.4889e+04	-6.2000	-9.8000	107.3000	92.5000	-3.6000	-14.8000	-0.5500	0	193.6713
8	'06-6576-32-P3-8'	2.4929e+04	3	5.9000	100.1000	31.8000	2.9000	-68.3000	0	0	177.5687
9	'06-6616-36-P3-8'	2.4915e+04	19.7000	20.7000	106.6000	99.4000	1	-7.2000	-3.6000	0	172.0928
10	'06-6650-39-P3-8'	2.4947e+04	27.2000	30.5000	107	73.5000	3.3000	-33.5000	-1.9000	0	174.3741
11	'06-6707-40-P3-8'	2.5004e+04	-2.3000	-4.3000	105.3000	87.9000	-2	-17.4000	-0.8609	1	186.5569
12	'06-6771-54-P3-8'	2.5037e+04	14.4000	17.7000	108.9000	54.5000	3.3000	-54.4000	0.5980	1	176.5286
13	'06-6798-62-P3-8'	2.5030e+04	3.9000	5.2000	98.4000	34.1000	1.3000	-64.3000	0.7171	0	178.8418
14	'06-6821-0-P3-8'	2.5037e+04	28.5000	28.9000	106	81	0.4000	-25	-1.4428	0	179.0833
15	'06-6829-1-P3-8'	2.5040e+04	-2.3000	-8.9000	107.3000	86	-6.6000	-21.3000	-1.9000	0	197.2161
16	'06-6835-2-P3-8'	2.5027e+04	2.6000	4.9000	107.3000	29.5000	2.3000	-77.8000	1.8667	0	178.3067
17	'06-6873-8-P3-8'	2.5115e+04	13.1000	13.1000	104.3000	86.9000	0	-17.4000	-1.9802	1	180
18	'06-6890-12-P3-8'	2.5115e+04	3.9000	4.9000	103	40	1.0000	-63	2.2400	1	179.0906
19	'06-6891-13-P3-8'	2.5118e+04	2.6000	3.9000	101.4000	43	1.3000	-58.4000	1	1	178.7248
20	'06-6895-16-P3-8'	2.5117e+04	12.8000	14.4000	107.3000	47.2000	1.6000	-60.1000	1.9000	1	178.4750
21	'06-6898-17-P3-8'	2.5119e+04	13.5000	15.1000	101	45.9000	1.6000	-55.1000	1.9000	1	178.3367

Figure 3.4 Example of Summarized Trajectory Data and Available Predictor Variables

Based on the summary data shown in Figure 3.4, vehicle trajectories considered to clearly represent left, thru, or right traveling vehicles were identified (1,060 in the 3-hour period analyzed). The identification of these trajectories was based on the last coordinate logged by the data collection system for each vehicle. The subset of trajectories was used for training a classification model as input, i.e., as the ground truth data. The approach used can be considered one in which the ground truth data can be automatically identified for the model based on known information about the geometry of the intersection.

The geometric information can be obtained directly from the radar-based vehicle detection system or entered by the user, but it should be noted that it is not ground truth data in the traditional sense of undergoing manual verification. The ability of using ground truth data systematically identified is one of the benefits of the classification procedure proposed because it eliminates a time-consuming stage of the typical process used to generate machine learning models. The subset of the dataset was then passed as input to a classification function created using the *Classification Learner* application in MATLAB. Predictor variables passed as input were *FirstX, LastX, FirstY, LastY, DeltaX, DeltaY, Acceleration,* and *StoppedCount*.

A linear support vector machine (Linear SVM) was trained with the input data (1,060 observations) using 5-fold cross-validation. As expected, when the performance of the classifier is analyzed based on the input data, performance metrics such as the true-positive rate achieved 100% as shown in Figure 3.5. This is expected because the data passed as input were observations for which there was no question about the movement at the intersection.

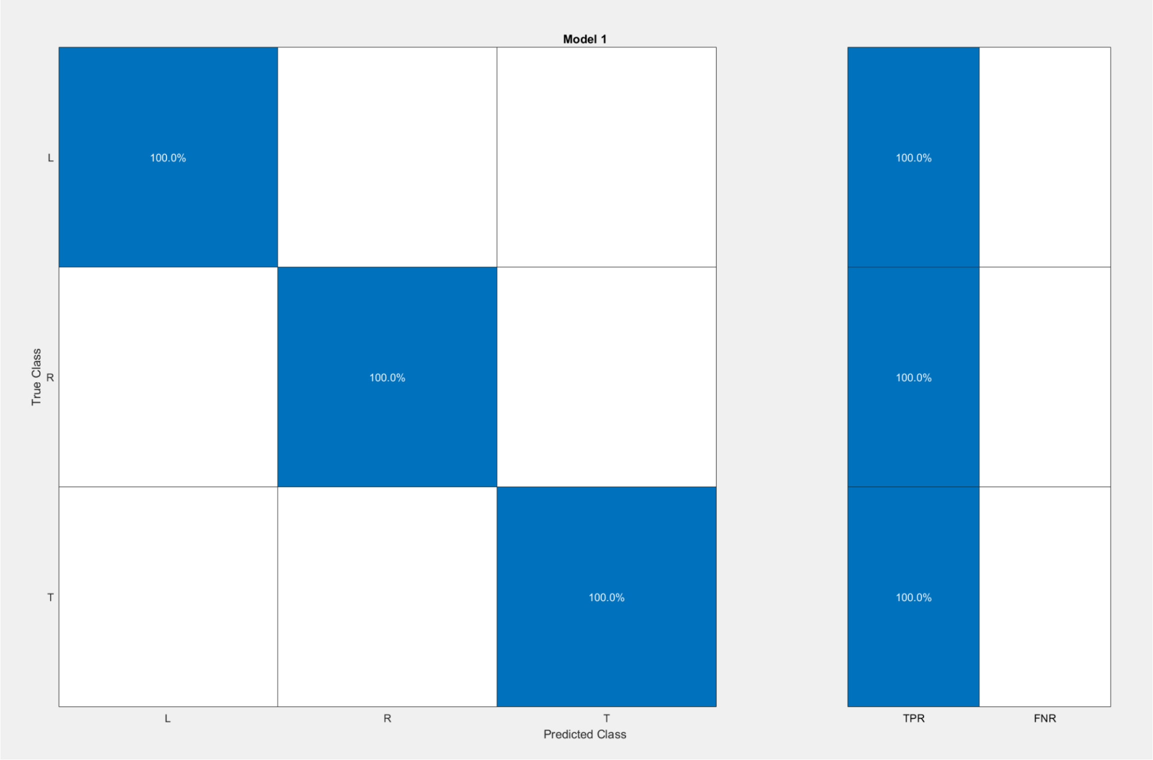


Figure 3.5 Performance Metric Based Solely on Input Data

The resulting classifier (Linear SVM) was then passed the initial summarized dataset as input that contains the same type of input variables used to train the model. In other words, the classifier developed solely on clearly defined input data was used to assign a predicted movement to observations in the summary dataset without a clearly defined movement. The result of the process is a summarized dataset containing a fully populated new column, shown as *Predicted* in Figure 3.6, that contains a travel direction indicator such as *L, T, or R* which indicates left, thru, or right.

Variables - classified

PLOTS VARIABLE VIEW

classified

1402x12 table

	1 VehicleId	2 Timestamp	3 FirstX	4 LastX	5 FirstY	6 LastY	7 DeltaX	8 DeltaY	9 Acceleration	10 StoppedCount	11 ExitAzimuth	12 Predicted
49	'07-1063-47-P3-8'	25745	12.8000	14.1000	103.7000	37.1000	1.3000	-66.6000	0	0	178.8818	'T'
50	'07-1065-49-P3-8'	2.5737e+04	12.1000	12.8000	103.7000	58.1000	0.7000	-45.6000	1.1921	0	179.1205	'T'
51	'07-1066-51-P3-8'	2.5737e+04	-2.6000	-6.2000	107.6000	95.5000	-3.6000	-12.1000	-4.5033	0	196.5688	'R'
52	'07-1075-31-P3-8'	2.5742e+04	13.8000	15.7000	105.6000	43	1.9000	-62.6000	1	0	178.2615	'T'
53	'07-1079-53-P3-8'	2.5765e+04	-1	-5.6000	107	96.5000	-4.6000	-10.5000	-4.3046	0	203.6580	'R'
54	'07-108-59-P3-8'	2.5315e+04	25.3000	25.3000	105	93.2000	0	-11.8000	0	1	180	'L'
55	'07-1102-61-P3-8'	2.5771e+04	-1.3000	-9.5000	108.3000	96.8000	-8.2000	-11.5000	-3.4000	0	215.4905	'R'
56	'07-1103-62-P3-8'	2.5786e+04	-2.3000	-1.6000	107.9000	89.2000	0.7000	-18.7000	-2.6866	1	177.8562	'R'
57	'07-1109-63-P3-8'	2.5837e+04	24.9000	29.5000	109.6000	81.4000	4.6000	-28.2000	-1	1	170.7355	'L'
58	'07-1246-15-P3-8'	2.5839e+04	12.1000	12.5000	104.3000	59.1000	0.4000	-45.2000	0	0	179.4930	'T'
59	'07-1259-19-P3-8'	2.5842e+04	27.6000	27.9000	106.3000	96.5000	0.3000	-9.8000	0	0	178.2466	'L'
60	'07-1276-20-P3-8'	2.5851e+04	-1	-6.2000	108.3000	94.5000	-5.2000	-13.8000	-5.6667	0	200.6470	'R'
61	'07-1280-22-P3-8'	2.5854e+04	-3.6000	-9.8000	107.6000	92.5000	-6.2000	-15.1000	-2.0000	0	202.3229	'R'
62	'07-1282-24-P3-8'	2.5856e+04	13.1000	14.4000	92.5000	24.9000	1.3000	-67.6000	-1.2000	0	178.8983	'T'
63	'07-1286-26-P3-8'	2.5897e+04	-4.9000	-7.9000	104.7000	90.9000	-3	-13.8000	-2.3500	1	192.2648	'R'
64	'07-1287-27-P3-8'	2.5886e+04	0	-7.5000	109.3000	100.7000	-7.5000	-8.6000	-4.3333	1	221.0915	'R'
65	'07-1303-33-P3-8'	2.5857e+04	13.1000	14.1000	92.5000	28.2000	1	-64.3000	-1.1333	0	179.1090	'T'
66	'07-1306-36-P3-8'	2.5918e+04	14.1000	12.5000	105.6000	79.4000	-1.6000	-26.2000	1.8000	1	183.4946	'T'
67	'07-1310-38-P3-8'	2.5902e+04	0	-4.9000	106	99.1000	-4.9000	-6.9000	-2.0000	1	215.3803	'R'
68	'07-1317-39-P3-8'	2.5915e+04	29.9000	29.9000	103	82.7000	0	-20.3000	0	1	180	'L'

Figure 3.6 Classified Vehicle Trajectories

Using the presented classification approach, the trajectory of, for example, all right turning vehicles can be obtained and analyzed for each individual vehicle to determine the moment when each vehicle arrives at the stop bar and the moment when that vehicle crosses the expected path of a pedestrian on the crosswalk. This information can then be used as input for previously developed models [4] used to study vehicle-pedestrian interactions. The performance of the presented classification approach on the entire trajectory dataset is discussed in the section ahead.

3.4 <u>Performance Analysis</u>

An indicator of classification algorithm performance commonly used by transportation engineers is the difference between predicted volume and observed volumes over 5-min periods. Predicted volumes are the vehicle volumes calculated by a prediction algorithm such as the Linear SVM discussed in the previous sections over 5-min periods. Observed volume is the volume over the corresponding collection of 5-min periods based on ground truth observations such as the ones made using a frame-by-frame analysis of the video discussed in an earlier section. The performance of the classifier can then be measured in terms of how 5-min volumes by movement in the observed volumes compare to the 5-min volumes by movement in the predicted volumes. Figure 3.7 shows a visualization of the absolute differences in observed-predicted volumes over 5-minute periods over a 3-hour window for which video and trajectory data were available.

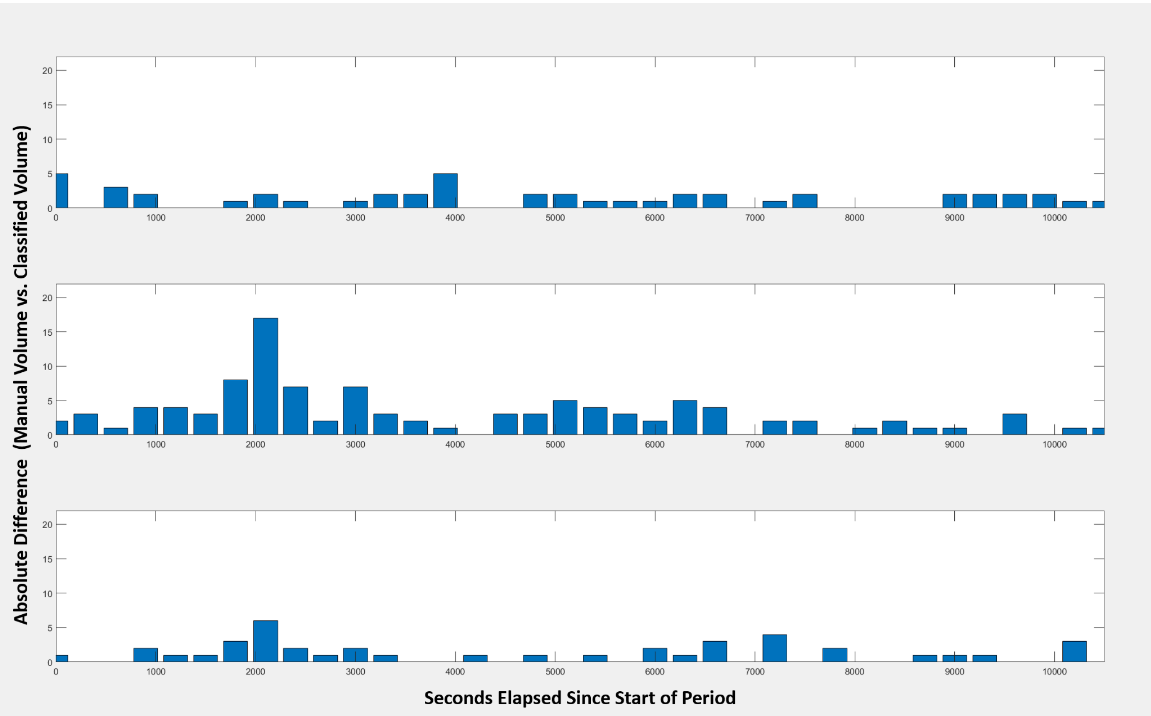


Figure 3.7 Classification Procedure Performance Based on Vehicle Volume

Statistics derived from Figure 3.7 indicate that on average the observed-predicted absolute difference in volume for a left turn movement is 1.3 vehicles, 3.0 vehicles for the thru movement, and 1.1 vehicles for right turn movements. Based on past experiences and observations, peaks observed in the difference (e.g. around the

2,000 second mark shown in Figure 3.7) can be explained by the effect of occlusion. For alternative detection systems such as those relying on line-of-sight technology (e.g. radar and optical systems), occlusion means that no vehicle is detected because the corresponding sensor is unable to see the vehicle. Other reasons such as those caused by software glitches during the data collection can also explain some of the higher absolute differences observed in Figure 3.7.

3.4.1 Implications of Performance Observed and Guidance

From the perspective of efficiently supporting vehicle-pedestrian interaction evaluations, the type of classified trajectory data is deemed appropriately accurate. The reason for deeming the performance sufficiently accurate is based on knowledge of the research team of the steps involved in the process and the operation of the radar-based vehicle detection system. For example, in the analysis of the trajectory data of individual vehicles to quantify vehicle-pedestrian interactions, the leading vehicles in the right turn movement is likely to be the key to the analysis. Therefore, the likelihood of the uncertainty in the classification algorithm materializing in a negative way and affecting the model generation is considered negligible because as an additional step in the process a successful analysis of the interaction requires sufficient data to quantify the moment when the vehicle crosses the pedestrian path and at this point an additional check for quality can be implemented.

4 Driving Simulation Tools

Driving simulation experiments provide the flexibility to study driving behaviors in a controlled environment. This means that scenarios in which a driver interacts with a pedestrian on a crosswalk while presented with a flashing yellow arrow can be studied in a driving simulation and behavioral metrics obtained. Experiments that involve a flashing yellow arrow have been conducted using driving simulators in the past [1]–[3]. However, one challenge with the use of existing driving simulator platforms is that the signal creation tools are limited or non-existent thus forcing researchers to find workaround to the limitations of the platform to make simulator experiments display a flashing yellow arrow indication. Workarounds used often break key functionality related to the control of ambient vehicle traffic and are meant for a specific experiment.

The sections ahead describe the characteristics of a collection of objects created using the Unity Game Engine that are reusable and simplify the development of driving simulation scenarios that involve a flashing yellow arrow. These objects (pre-fabs) were created with flexibility as a goal and were designed to operate in a way that resembles traffic engineering principles thus making configurations of objects like signals more realistic to allow bringing existing field conditions to a driving simulation environment. The collection of objects is available for download from a code repository [7] to simplify access to the resources by others and to provide access to future updates as the tools developed continue to evolve.

4.1 Signal Heads

Three types of signal head models were added functionality to enable operation in a way that resembles phasing schemes that can be visualized using the ring diagram commonly understood by traffic engineers. Each of the signal was converted into a pre-fab in Unity to allow portability and support use in different experiments. The same script can be used to control every signal head (regardless of the configuration) and the indication displayed by the signal depends on a concept of state. State 0 means the signal is off (no signal indication is on). On the other hand, State 1 means that the top-most lens of the signal is on while State 2 means that the second lens from the top is active. The lens and lights controlled by a state are based on a corresponding texture that reflects a desired color. This reliance on textures makes it possible to have multiple lenses associated with a signal state, a feature that is beneficial for the 5-section head shown in Figure 4.1 as well as with a doghouse style signal head.

Figure 4.1 Signal Heads on a State 0

The controlling script created exposes control over the signal functionality and operation via a graphical user interface as shown in Figure 4.2 which shows how controls are exposed in the Unity environment for the 4-section signal head shown in Figure 4.1. Specifically, for each signal state the user can define aspects such as a specific signal state allowing movements and control whether the signal is in flashing mode thus allowing any signal state to flash, a key functionality to support research involving the use of flashing yellow arrows. Similarly, aspects such as the flashing interval (in seconds) can also be defined via the user interface shown in Figure 4.2. The signal state can also be modified at any time as well as a value indicating the NEMA phase number associated with the signal at any point in time.

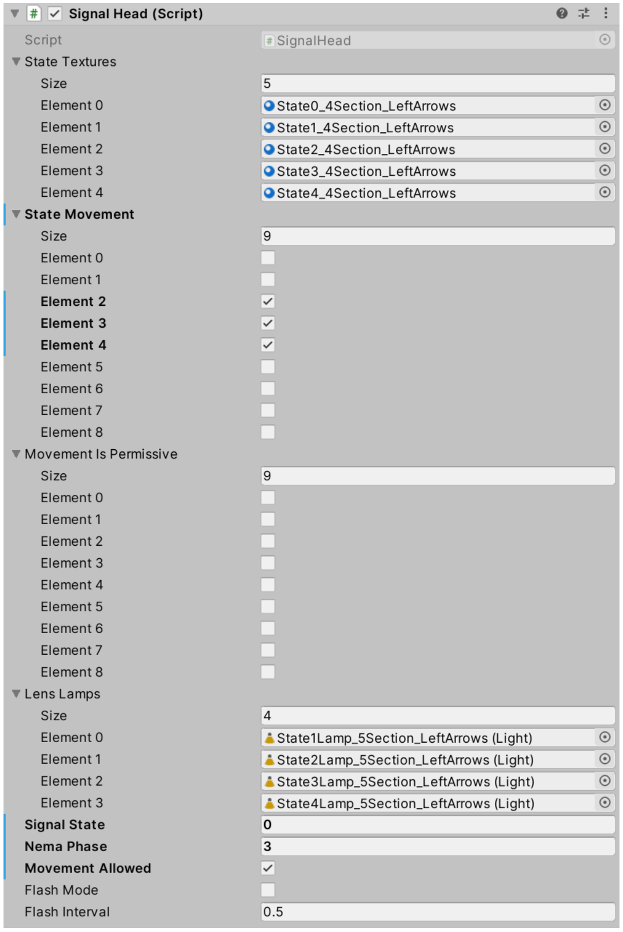


Figure 4.2 Example of User-Configured Functionality Enabled Through Scripting

The functionality enabled the control scrips assigned to a signal head go beyond displaying a specific phase. Specifically, the ability to link a specific lens to a movement is key to the operation of ambient traffic in a simulation. Each of the signal heads described can be used to control the movement of vehicles by making the behavior of ambient traffic sensors dependent on the “movement allowed” state of the signal. This type of granular control makes it possible to create complex control scenarios such as those in which signals in the same approach have conflicting indications, e.g., the approach described ca be used to operate an approach in such a way that the left turns are allowed to move while the thru movements are not and vice-versa.

4.2 Ambient Traffic Controllers

A collection of traffic controllers (objects with a collider attached) were created that are useful to control a simulation. Figure 4.3 shows a screenshot of a collection of sensors in a signalized intersection model. Sensors with a specific color have similar functions and a total of 5 types of controllers are shown, each with a corresponding number assigned to facilitate discussions. Together with the signal head, the controllers operate to replicate typical behaviors observed at a signalized intersection.

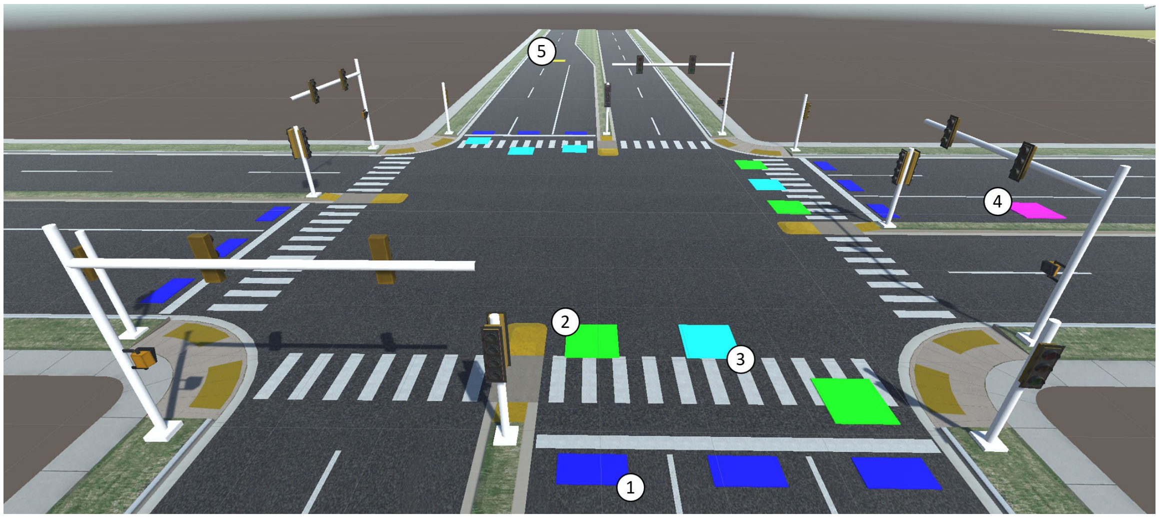


Figure 4.3 Sensors to Control Ambient Traffic

The goal of controller type 1 is to determine if a vehicle that is detected by the collider assigned to the controller can continue moving or if it must come to a stop. For optimal operation, controllers or type 1 can be linked to a signal head and a decision on stopping a vehicle is made based on the “movement allowed” parameter of the signal head. Information about movement state once a vehicle enters the collider of the controller can be retrieved from the controller object directly or the controller object can modify a parameter on the vehicle to indicate the need to come to a stop. Because of the way in which colliders operate, a vehicle collider can interact with the controller prior to being directly on top, assuming the size of the vehicle collider is configured properly.

The goal of controller type 2 is to initiate a turning maneuver on a vehicle. Parameters for the turning maneuver are specified in the form of a target azimuth and a turning radius. A vehicle going over the sensor receives the turn parameter and can follow the instructions until the turn is completed. Logic in the controlling script of the vehicle must be capable of handling the task. Parameters in the controller can be configured to determine the proportion of vehicles that follow the turn instructions.

The goal of controller type 3 is minimal, it is designed to act as a sensing tool and can be used to log events. If enabled, an event signal can be generated once the collider associated with controller type 3 is activated by another object. It should be mentioned that all controllers described here have the same capability of controller type 3. In other words, all controllers shown are an expanded version (in terms of capabilities) of controller type 3.

Controller type 4 is meant to generate ambient traffic at a specified interval and up to a given limit. Ambient traffic generated by this controller type can be assigned a control script designed to operate with the collection of controllers shown in Figure 4.3. Ambient traffic generated can be assigned parameters such as initial speed and target speed along with the acceleration used to reach the goal.

Controller type 5 is designed to force a portion of ambient traffic that triggers the corresponding collider to change lanes, or specifically, move left or right a specified distance. The goal of this controller type is enabling behavior such as last-minute lane at changes at specified location as well as to support populating turning bays in driving simulation experiments based on ambient traffic generated using a controller type 4 and that was generated further upstream of the location of the dedicated turn lane.

4.3 Data Streaming

One of the challenges associated with driving simulation experiments is the collection of data and the identification of the portions of the data that are of interest to the experiment. To simplify the process of data collection a data streaming module for unity was created. This data streaming module can be attached to a simulation environment and entities in the simulation can communicate their position. The module listens for input from the entities and then produces a data stream in UDP format that is sent to a specify end point and port. In addition to position information, the data streaming module can assign an entity name and event code to the data stream produced. The structure of the data is meant to facilitate analyses of specific events and areas of the simulation environment. Streaming of individual information can be started and stopped as desired throughout the simulation to, if desired, simplify the complexity of the resulting dataset.

The decision to stream that data over UDP instead of logging the stream locally was made to enable future applications such as the direct integration of vehicle position data with custom dashboard systems (e.g. for navigation purposes) as well as to allow end users to select their own data collection tool. For example, the nature of the data stream makes it possible to log the data using popular data analysis tools based in Python and MATLAB thus streamlining some of the steps in a typical data analysis tool chain in that involves transferring data from the simulator environment to a new location, processing, and then running specific analyses. As with the other components of the driving simulation tools created, a data streaming pre-fab has been uploaded to a code repository [7].

5 Conclusions

This report documented two approaches that can be used as the foundation for streamlining the process of quantifying and understanding vehicle-pedestrian interactions without the need for labor-intensive data collection efforts. The two approaches presented have the potential for meaningful time savings when conducting proactive before-after evaluations regarding the impact of a treatment such as a FYA on the safety of vehicle-pedestrian interactions. The safety evaluations possible are of the proactive nature because the data that can be acquired through the procedures can then be used to create models that describe the behavior of a driver when said driver interacts with a pedestrian. The driver behavior that can be modeled as documented in previously completed research [4]; specifically, by how much the presence of a countermeasure such as a FYA reduces the driver turning maneuver completion time when compared to a control condition, could then be used as a safety indicator.

5.1 Isolation of Trajectories Using Dynamically Generated Machine Learning Models

Understanding vehicle-pedestrian interactions requires information about the position of interaction vehicles and pedestrians over a period of time. Technology available makes the monitoring of pedestrian positions using video possible and the monitoring of vehicle position using radar-based vehicle detection also possible. One challenge for trajectory data obtained from radar-based vehicle detection systems is that each trajectory is not necessarily associated with a specific maneuver. For example, on lanes that have a shared movement such as a lane that allows right and thru movements, identifying the trajectory associated with right turning vehicles to then identify the corresponding pedestrians is not always a process based on clear cut measurements.

Previous work has developed classifications methodologies for turning movements on shared lanes that can help with the process [6]. The effort described in this report focused on the development of an improved approach to classifying vehicle trajectories based on the use of machine learning. The approach presented is able to assign a turning movement to trajectories monitored and logged by radar-based vehicle detections systems. These trajectories, when combined with pedestrian position information, can be used to streamline the modeling procedures documented in previously completed research without the need for labor-intensive procedures. One unique and important differentiator of the proposed trajectory classification procedure, and other approaches, is the use of dynamically generated machine learning models used for vehicle classification.

By having a methodology that creates a machine learning model for each analyzed site, by relying on an initial set of generic parameters, the quality of the classification process can improve thus reducing the amount of effort required for acquiring the type of data required for modeling vehicle-pedestrian interactions and studying the impact that treatments such as a FYA have on the safety of those interactions.

5.1.1 Applications Beyond Safety Evaluations

While the goal of the trajectory classification approach developed is to isolate turning maneuvers to quantify their behavior, applications of the classification approach expand beyond safety evaluations. Specifically, there is potential for the trajectory classification approach to be used for conducting turning movement counts and the approach represents an improvement on previously developed methodologies [6]. The dynamic and streamlined nature of the machine learning models that are created to classify vehicle trajectories makes the deployment of these models on embedded system (if additional steps are taken) a possibility and as a result could be used to improve the state of the practice for conducting detailed turning movement counts on signalized intersections instrumented with compatible radar-based vehicle detection systems.

5.2 Objects to Streamline the Creation of Driving Simulation Scenarios

A collection of objects ("pre-fabs") designed and programmed with the specific goal of supporting driving simulation evaluations that involve FYA signal indications were created. Three of the objects that are a key output of the described work are: signal head objects along with the corresponding controllers, data collection streaming system, and ambient vehicle controllers. When these objects are combined with existing tools and resources such as pre-configured pedestrian movements and study subject controls, the process of creating driving simulations using modern game engines that involve FYA is streamlined.

Specifically, one challenge of some of the popular platforms used by research teams for driving simulations is the inability of customizing signal sequences to reflect real life conditions that involve complex phasing operations such as those that involve FYA without having to rely on heavily customized code that often breaks native functionality in the simulation platform such as the proper control of ambient traffic. The signal sequence control procedures developed as part of this research effort make it possible for a potential user of the collection of objects created to specify the sequences of signals using an approach that closely resembles that of a real traffic signal controller and have traffic respond to the movement instructions indicated by the signal sequences.

By providing a signal control procedure that resembles the operation of real traffic signal controllers, transferability of real field conditions to driving simulation scenarios can be simplified thus making before-after evaluations of countermeasures for improving vehicle-pedestrian interactions simpler and feasible to implement in a driving simulation environment.

5.2.1 Limitations and Considerations

The collection of objects and tools created are meant to support the creation of driving simulation scenarios to study vehicle-pedestrian interactions under the control of a FYA. It should be noted that while numerous experiments have been conducted in the past using driving simulators that involve FYA scenarios, there should be no assumption made that the type of experiment that can be created using the tools developed is valid until the experiment is conducted and the results are evaluated.

5.3 Future Work and Additional Applications

Due to the Covid-19 pandemic, physical experiments and in-person data collection were impacted across the board. Therefore, from the driving simulation perspective the focus of the research effort described was the development and the expansion of driving simulation tools to facilitate running similar experiments in the future. Similarly, from the perspective of field-based data collection, the focus of the research was on streamlining the methods available to obtain the necessary data to model vehicle-pedestrian interactions.

As a result, a key aspect of future work related to the research effort described is conducting experiments using the tools developed and modeling the vehicle-pedestrian interactions. Additionally, future work should look at the use of the tools not only for understanding vehicle-pedestrian interactions from a purely scientific point of view but also for the use in the design process to evaluate different geometric design alternatives.

5.3.1 Applications Beyond the Scope of Research Effort

Besides the potential future use of the driving simulation tools and procedures developed as part of the described research effort, there are immediate uses outside the scope of the project that are already taking place. For example, the signal objects created are being used by another project studying the way in which remote drivers accept gaps when control from an automated vehicle is transferred to a local operator.

Similarly, the data collection system established for the subject vehicle is also being used in a project studying mechanism for alerting drivers about imminent crossings by pedestrians detected through monitoring of cell phone signals.

References

[1] C. L. Brehmer, K. C. Kacir, D. A. Noyce, and M. P. Manser, “NCHRP 493: Evaluation of traffic signal displays for protected/permissive left-turn control,” Transportation Research Board, 493, 2003. [Online]. Available: http://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_rpt_493.pdf

[2] D. A. Noyce, A. R. Bill, and M. A. Knodler Jr, “Evaluation of the Flashing Yellow Arrow (FYA) Permissive Left-Turn in Shared Yellow Signal Sections,” 2014.

[3] M. A. Knodler, D. A. Noyce, K. C. Kacir, and C. L. Brehmer, “An Evaluation of Driver Comprehension of Solid Yellow Indications Resulting from Implementation of Flashing Yellow Arrow,” presented at the 86th Annual Meeting of the Transportation Research Board, Washington, D.C., Jan. 2007. Accessed: Mar. 05, 2018. [Online]. Available: https://trid.trb.org/view/802137

[4] H. Nassereddine, K. R. Santiago-Chaparro, and D. A. Noyce, “Modeling Vehicle–Pedestrian Interactions using a Nonprobabilistic Regression Approach,” *Transportation Research Record*, vol. 2675, no. 1, pp. 356–364, 2021, doi: 10.1177/0361198120962799.

[5] K. R. Santiago-Chaparro and D. A. Noyce, “Expanding the Capabilities of Radar-Based Vehicle Detection Systems: Noise Characterization and Removal Procedures,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 11, pp. 150–160, Nov. 2019, doi: 10.1177/0361198119852607.

[6] K. R. Santiago-Chaparro, M. Chitturi, A. Bill, and D. A. Noyce, “Automated Turning Movement Counts for Shared Lanes: Leveraging Vehicle Detection Data,” *Transportation Research Record*, no. 2558, pp. 30–40, 2016.

[7] K. Santiago-Chaparro, “Unity Signal Control Tools.” https://github.com/kelvinsantiago/unitysignalcontroltools