Development of a Tool to Calculate Appropriate Advisory Speeds on Horizontal Curves

Julie M. Trumpoldt

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
in
Civil Engineering

Bryan J. Katz, Chair
Antonio A. Trani
Antoine G. Hobeika

December 5, 2014
Blacksburg, Virginia

Key words: advisory speed, horizontal curve, Android Application, CurveAdvisor

Development of a Tool to Calculate Appropriate Advisory Speeds on Horizontal Curves

Julie M. Trumpoldt

(ABSTRACT)

Horizontal curves are a contributing factor for numerous deaths on roadways. The curve characteristics dictate the severity of the curve and require the driver to be more attentive while driving. To address this issue, advisory speeds are posted on horizontal curves to warn drivers to slow down for their safety. There are six main procedures to assign advisory speeds. This paper focuses on two of these methods, finds a connection between the two, and develops an Android Application that can be used to determine an advisory speed for a curve. In this work, a brief summary of the six existing methods for advisory speed assignment are discussed. Pros and cons are included for each for comparison purposes. Next, two of these methods are highlighted by applying them in the field. Data is collected and a relationship between them is determined. Using this relationship, an Android Application is created and the various details of the design process are described. This Application, called CurveAdvisor, allows the user to assign the appropriate advisory speed on a desired horizontal curve. An analysis is then conducted to test the effectiveness of CurveAdvisor. Results indicate that CurveAdvisor is successful in many cases. Finally, contributions and suggestions for future work are included.

Acknowledgements

My sincere appreciation goes to my advisor, Dr. Bryan Katz, for helping me find a topic I absolutely fell in love with and for providing guidance throughout the entire process. I also would like to acknowledge my committee members, Dr. Antonio Trani and Dr. Antoine Hobeika, for their critique, input, and positive feedback.

A special thank you to my friends who helped me collect data for the curves on Harding Avenue and Krzysztof for helping me with my Application coding.

I would also like to recognize my parents for their constant support and love. When times were tough, they always believed in me and encouraged me.

And most importantly, I attribute all my work to the Sovereign Lord. To God be the glory.

Table of Contents

Chapter 1: Introduction **1**

1.1 Literature Review 1

1.2 Problem Statement 6

1.3 Objective 7

1.4 Approach 7

Chapter 2: Data Collection **8**

2.1 Method 8

2.2 Data Organization 10

2.3 Results 12

2.4 Conclusions 21

Chapter 3: Proposed Tool **23**

3.1 Overview of MIT App Inventor 2 23

3.2 Model Description 25

Chapter 4: Evaluation **43**

4.1 Testing the Android Application 43

4.2 Data Collection 43

4.3 Results 45

Chapter 5: Conclusions **52**

5.1 Contributions 52

5.2 Future Work 52

References **54**

Appendix A Supplementary Figures and Tables **55**

List of Figures

2.1 Curves used in the analysis and their corresponding advisory speeds. 8

2.2 BBI used during data collection. 9

2.3 Sample of ACCELEROMETER.txt file after being imported into Excel. 10

2.4 Sample of LOCATION.txt file after being imported into Excel. 11

2.5 Screenshot of kml file. All curves along Harding Avenue are shown in one file. 11

2.6 Updated curves used in the analysis and their corresponding advisory speeds. 12

2.7 Curve 02 driven at 25 mph. Raw data for Run 1 plotted. 13

2.8 Scatterplot of data collected for Curve 02 during Run 1. Relates the x-component versus time. 14

2.9 Final visual representation of all runs for Curve 02 at 20 mph. 15

2.10 Final visual representation of all runs for Curve 02 at 25 mph. 15

2.11 Final visual representation of all runs for Curve 02 at 30 mph. 16

2.12 Final visual representation of all runs for Curve 02 at 35 mph. 16

2.13 Relationship between the x-component values and their corresponding BBI values for Curve 02. 17

2.14 Relationship between maximum averaged x-components and their corresponding BBI values for Curves 01, 02, and 04. 18

2.15 Spot speeds observed on Curve 01. 19

2.16 Spot speeds observed on Curve 02. 19

2.17 Spot speeds observed on Curve 04. 20

3.1 Designer interface [10]. Massachusetts Institute of Technology. App Inventor. 2012; Available from: http://appinventor.mit.edu/explore/designer-blocks.html. Used under fair use, 2014. 24

3.2 Blocks Editor interface [10]. Massachusetts Institute of Technology. App Inventor. 2012; Available from: http://appinventor.mit.edu/explore/designer-blocks.html. Used under fair use, 2014. 24

3.3 A sample of control blocks available in App Inventor [10]. Massachusetts Institute of Technology. App Inventor. 2012; Available from: http://appinventor.mit.edu/explore/ai2/support/blocks/control.html. Used under fair use, 2014. 25

3.4 Initial Viewer layout for CurveAdvisor. 26

3.5 Components used to create display in Figure 3.4. 27
3.6 Incorporating buttons and simple functions to begin Application design. 28
3.7 Key for block colors. 28
3.8 Timer control block. 29
3.9 AccelerationChanged control block. 30
3.10 Distribution of time intervals within ATime file using the Atest. 30
3.11 Distribution of time intervals within TTime file using the Ttest. 31
3.12 Updated interface for CurveAdvisor. Reset button is below “MUTCD: new speed?” 32
3.13 Components used to create display in Figure 3.12. 33
3.14 Updated Start, Stop, and Reset buttons with new variables. 34
3.15 Timer and accelerometer components and their internal calculations. 35
3.16 Criteria that are used within CurveAdvisor to output recommendations. 39
3.17 New addition to the Stop button coding - AASHTO and MUTCD criteria for advisory speeds. 40

4.1 View of Curves A and B. These curves were used to further test CurveAdvisor. 44
4.2 View of Ramps 1 and 2. These curves were used to further test CurveAdvisor. 45
4.3 Screenshot of CurveAdvisor for Curve 01 at 25 mph. 46
4.4 Screenshot of CurveAdvisor for Curve 01 at 30 mph. 47

A.2.1 Final visual representation of all runs for Curve 01 at 20 mph. 55
A.2.2 Final visual representation of all runs for Curve 01 at 25 mph. 55
A.2.3 Final visual representation of all runs for Curve 01 at 30 mph. 56
A.2.4 Final visual representation of all runs for Curve 01 at 35 mph. 56
A.2.5 Final visual representation of all runs for Curve 03 at 35 mph. 57
A.2.6 Final visual representation of all runs for Curve 03 at 40 mph. 57
A.2.7 Final visual representation of all runs for Curve 03 at 45 mph. 58
A.2.8 Final visual representation of all runs for Curve 03 at 50 mph. 58
A.2.9 Final visual representation of all runs for Curve 04 at 15 mph. 59
A.2.10 Final visual representation of all runs for Curve 04 at 20 mph. 59
A.2.11 Final visual representation of all runs for Curve 04 at 25 mph. 60
A.2.12 Final visual representation of all runs for Curve 04 at 30 mph. 60

A.2.13 Relationship between the x-component values and their corresponding BBI values for Curve 01. 62
A.2.14 Relationship between the x-component values and their corresponding BBI values for Curve 03. 63
A.2.15 Relationship between the x-component values and their corresponding BBI values for Curve 04. Values for 30 mph were excluded due to extreme BBI readings. 63
A.3.1 File produced through the Atest to determine the mean time interval. 64
A.3.2 File produced through the Ttest to determine the mean time interval. 65
A.3.3 File produced through the Atest to determine the number of acceleration values collected in each time interval. 66
A.3.4 File produced through the Ttest to determine the number of acceleration values collected in each time interval. 67

List of Tables

1.1 Suggested Ball-Bank Indicator Values for Horizontal Curves 3

2.1 Locations of Each Curve 12
2.2 Averaged X-component Values for Every 0.5 Seconds (Curve 02) 13
2.3 X-components and Corresponding BBI Values for Curve 02 17
2.4 R-squared Values for Each Curve 18
2.5 Spot Speed Statistics for Each Curve 20
2.6 Calculating 85th Percentile for Curve 01 – Step 1 21

3.1 AASHTO and MUTCD Guidelines for Advisory Speed Assignment based on BBI and Calculated X-component Values 38

4.1 Results of CurveAdvisor Testing on Curves 01, 02, and 04 48
4.2 Results of CurveAdvisor Testing on Curves A and B 49
4.3 Results of CurveAdvisor Testing on Ramps 1 and 2 50
4.4 Advisory Speeds for Each Curve According to CurveAdvisor 50

A.2.1 X-components and Corresponding BBI Values for Curve 01 and Curve 03 61
A.2.2 X-components and Corresponding BBI Values for Curve 04 62

CHAPTER 1

INTRODUCTION

Horizontal curves are a contributing factor for numerous deaths on roadways[1]. This is mainly due to the fact that most drivers do not adhere to posted regulatory and advisory speeds and continue at a speed they believe to be safe. Research has proven that many drivers go above the indicated advisory speed by 7 to 10 mph[2]. Most advisory speeds are set much lower than what is actually considered to be a safe speed; often, drivers can traverse the curve with a speed that is higher than the one listed without any consequence. Due to this realization by the public, most drivers go above the posted speed limit and their driving behavior goes unchanged. It's as if the sign was not even posted. In the event of an adequately labeled curve, the driver may end up in a crash based on their previous interactions with other curves. The criteria for setting advisory speeds has essentially remained the same over 50 years [3]. Furthermore, the designation of advisory speeds is not consistent throughout the United States, or even within a state. There are six main procedures to assign advisory speeds. This paper summarizes the six methods for advisory speed assignment, highlights two methods by applying them in the field, determines the relationship between the two methods, and describes how an Application for an Android device can utilize this relationship to calculate an appropriate advisory speed.

1.1 LITERATURE REVIEW

Before determining the appropriate advisory speed on a horizontal curve, many factors need to be considered. Most of these factors are related to curve geometry and the five most significant are listed below[4]:

- Radius
- Superelevation
- Tangent Speed
- Vehicle Type
- Curve Deflection Angle

Given the above influences, there are six different methods for setting an advisory speed. Each has its own advantages and disadvantages with some being used more often than others. The ball-bank indicator method is used by most agencies and is therefore the preferred method [5]. However, new methods have been created and are proving to be more accurate. Listed below are the six procedures:

- Direct Method
- TTI Curve Speed Model – Compass Method

- TTI Curve Speed Model – Global Positioning System (GPS)
- TTI Curve Speed Model – Design Method
- Ball-Bank Indicator Method
- Accelerometer Method

Each method consists of two main steps and sometimes a third step: collecting data, assigning advisory speed, running a test in the field. Below are descriptions of how each method is conducted.

Direct Method

Measures vehicle speed at the curve midpoint through the use of a radar gun and specifies types of cars by using a traffic classifier. Data collection is completed under free-flow conditions. The radar gun is operated by a technician who records speeds on-site over a specific time frame (usually 2 hours) or until the number of cars needed passes by. According to Bonneson et al., the number of vehicles needed for the data collection is assumed to be 125 vehicles [6]. The traffic classifier is stationed overnight to count the different types of cars.

The data collected in the field is used to calculate the advisory speed based on desired criteria. This can be the 85th percentile speed, average speed multiplied by 0.97 (otherwise known as average truck speed), median speed, or any other chosen option. Bonneson et al. states that the advisory speed should equal the average truck speed[1]. When the advisory speed is finally calculated, Bonneson et al. suggests that it should then be rounded[1]. This entails an addition of 1 mph and then rounding down to the nearest 5 mph increment.

In the case of a two lane divided horizontal curve, each direction should be analyzed separately. When the section consists of two different curves, each curve is analyzed separately and is assigned separate speeds. However, if these curves are separated by a tangent section less than 600ft, both curves are assigned the speed that is most conservative.

Pros:

- Reflects the actual speed driven by motorists as they interact with the curve which targets the main problem of drivers choosing their own speed
- Considers all curve geometry that affects the severity of the curve (factors mentioned previously)
- Devices readily available

Cons:

- Cannot cover all types of vehicles – each vehicle has specific characteristics
- The method's biggest downfall is that the advisory speed criteria is undecided - MUTCD 2009 does not mention which curve speed distribution should be used. This creates non-uniformity in advisory speed designation.

- Multitude of resources needed
- May only account for each specific field collection

Ball-Bank Indicator Method

Two instruments are needed for this method: a ball-bank indicator and a speedometer. A device called a ball-bank indicator is installed on the dashboard of the vehicle. This device (digital or vial) displays a value that represents the superelevation, lateral (centripetal) acceleration, and vehicle body roll. With the vial, a small bubble moves from side to side as the vehicle turns. As the bubble moves, it travels along tick marks each representing a value of 1 degree. The limits of the device are -20 degrees and 20 degrees. The amount the bubble travels in one direction or another is based upon the three characteristics previously mentioned. The same concept is utilized for the digital ball bank indicator – the displayed numerical value represents the three characteristics. The threshold of the ball-bank reading is determined to be the speed at which the motorist can comfortably, and safely, drive the curve. This targeted value is predetermined before testing. Below is a table listing some values suggested by AASHTO[7] and MUTCD[2]:

Table 1.1: Suggested Ball-Bank Indicator Values for Horizontal Curves

	AASHTO 2004	**MUTCD 2009**
≤ 20mph	14 degrees	16 degrees
25-30mph	12 degrees	14 degrees
≥ 35mph	10 degrees	12 degrees

The advisory speed is set as the speed the vehicle reaches that is closest to, but does not exceed, the threshold value.

Pros:

- Device readily available
- Visual representation of how fast the car can operate safely
- Easy to administer

Cons:

- Best used only with passenger cars. Characteristics of other vehicles affect the body roll and therefore result in different measurements – readings are subjective
- Varied criteria
- Research has shown that ball-bank indicator readings vary significantly. Even when curves have similar geometry or road surface conditions, the readings can produce associated advisory speeds that differ 5 to 10mph from the other. [8]

Accelerometer Method

Field driving tests are conducted and the vehicle is equipped with an accelerometer which is a device that measures lateral (centripetal) acceleration. A GPS receiver is also utilized.

The advisory speed is set as the speed the vehicle reaches that is closest to, but does not exceed, the threshold value. This threshold value is predetermined before the test is conducted. Generally, a measurement of 0.26 g (8.4 ft/s^2) to 0.30 g (9.7 ft/s^2) is acceptable with 0.28 g (9.0 ft/s^2) being optimal [4].

Pros:

- Only one analyst needed to conduct study
- Easily administered

Cons:

- Sudden movements in the vehicle can produce inaccurate results
- Multiple runs may be needed

TTI Curve Speed Model – Compass Method

This method was created by the Texas Transportation Institute and is newer than the others. It collects the radius and superelevation of a curve and inputs the numbers into a formula. Radius and superelevation can easily be determined either from as-built plan sheets or field measurements. In the case of the latter, radius is found using a compass technique while superelevation is found using a ball-bank indicator.

The compass technique includes a compass (used to determine vehicle heading at two points) and a distance measuring device (used to measure the length between the same two points). The values found from the compass are subtracted from one another to determine the curve deflection angle. The curve length (found by the measuring device) is then divided by the calculated deflection angle to find the radius. All angles are in radians. If analyzing a simple curve, any two points along the curve can be used. However, if the curve is compound or spiral, then points are taken from the "1/3" points which are the points equal to one third of the length of the curve. All curves evaluated must be great than 200ft in length and have a deflection angle greater than 12 degrees[4].

The ball-bank indicator determines the superelevation. In this case, the ball-bank reading is taken when the vehicle is stopped in the middle of the curve. The reading is then multiplied by 1.56 to calculate the superelevation rate[1].

The advisory speed is determined through the use of Curve Advisory Speed software. First, the 85th percentile speed is determined given the regulatory speed limit. This number and all other data collected are entered into the program. The program utilizes a curve-speed prediction model and outputs an unrounded advisory speed. This is then rounded as mentioned previously (add 1mph and round down to nearest 5mph increment).

In the case of a two lane divided horizontal curve, each direction should be analyzed separately. When the section consists of two different curves, each curve is analyzed separately and is assigned separate speeds. However, if these curves are separated by a tangent section less than 600ft, both curves are assigned the speed that is most conservative.

Pros:

- Similar to the Traffic Control Device Handbook (TCDH) nomograph but also includes sensitivity to tangent speed
- Radius used is that driven by motorist NOT the actual radius – this is a flattened version of the radius and is therefore a larger number. It is more accurate for the situation.
- Device readily available
- Only one test run needed

Cons:

- The advisory speed criteria is undecided.
- To collect ball-bank reading, vehicle must be completely stopped. This creates hazards and limits the number of curves that can be evaluated since testing must occur on a very low volume road.

TTI Curve Speed Model – Global Positioning System (GPS)

Collects curve geometry using a GPS (data collected is responsible for calculating radius and deflection angle), an electronic ball-bank indicator (data collected is responsible for superelevation), and a laptop computer. The first two must be mounted on the dashboard while driving.

The analyst must try to drive as close to the center as possible – no cutting corners or drifting to the outside of the curve. The car must be driven below 45mph for accurate superelevation results and generally should be driven at least 10mph below the current advisory speed. In addition, the curve deflection angle must be greater than 6 degrees[4].

The computer is needed to execute the Texas Roadway Analysis and Measurement Software (TRAMS) program. This program calculates the radius, deflection angle, and superelevation from the data collected by the GPS and ball bank indicator. The 85th percentile speed is found by inputting the regulatory speed in the Curve Advisory Speed (CAS) software.

To determine advisory speed, the CAS software is used by importing the data from TRAMS and entering the regulatory speed. This can be done either in the field or in the office. In the office, data is entered manually using the TRAMS produced report files rather than importing the data in the field. Either option is acceptable.

In the case of a two lane divided horizontal curve, each direction should be analyzed separately. When the section consists of two different curves, each curve is

analyzed separately and is assigned separate speeds. However, if these curves are separated by a tangent section less than 600ft, both curves are assigned the speed that is most conservative.

Pros:

- Curve only needs to be driven once – only one test run per direction.
- Produces very accurate measurements

Cons:

- The advisory speed criteria is undecided
- Requires high-tech devices and computer programs
- Curve needs to be driven at low speeds
- Difficult to drive exactly along the center

TTI Curve Speed Model – Design Method

Curve geometry is obtained from files or as built plans. It mainly focuses on the radius, superelevation rate, and deflection angle. Generally this approach would be used for newly constructed or reconstructed curves – curves that have design elements and characteristics that are easily accessible. The advisory speed is calculated through CAS software.

In the case of a two lane divided horizontal curve, each direction should be analyzed separately. When the section consists of two different curves, each curve is analyzed separately and is assigned separate speeds. However, if these curves are separated by a tangent section less than 600ft, both curves are assigned the speed that is most conservative.

Pros:

- Simple procedure – no field testing needed

Cons:

- The advisory speed criteria is undecided
- Very difficult to analyze existing curves
- Impossible to conduct if design elements are not readily available. This is the case for most curves[4]

1.2 PROBLEM STATEMENT

After reviewing the literature on advisory speed assignment, it is very clear that the six methods differ greatly from one another and the advisory speed criteria for each method is inconsistent. With such difference among advisory speeds, potentially hazardous situations will arise when drivers are traversing the curves. In addition, most

of these methods require a substantial amount of time to complete. With available technology, a faster, more efficient way to determine advisory speeds is desirable.

1.3 OBJECTIVE

The goal of this research is to produce a tool that will quickly calculate the advisory speed on a curve that will also be consistent among many curves with various radii and superelevation. The desired outcome is an Android Application that is user friendly and reliable. This Application will relate the superelevation and largest accelerometer x-component of each curve to determine the appropriate advisory speed.

1.4 APPROACH

In order to achieve this goal, several steps must occur. First, data will be collected on existing curves with advisory speeds. The data includes superelevation and x-component values which will be found through a ball-bank indicator and an accelerometer respectively. Next, the data will be analyzed to pinpoint a relationship between the two characteristics. Once the relationship is determined, an Android Application will be created to include the mentioned relationship. Finally, the Application will be tested on curves to determine the reliability of the tool.

CHAPTER 2

DATA COLLECTION

The purpose of data collection is to analyze existing curves with advisory speeds and determine:

1. If there is a relationship between the Ball-Bank Indicator (BBI) readings and the x-component values.
2. How closely drivers follow the advisory speed.

2.1 METHOD

Two sections of roadway were targeted – a long series of curves on Harding Ave and two curves on Happy Hollow Road. These portions of roadway were chosen because they had varying advisory speeds, ranging from 25 mph to 40 mph. This incorporates a wide range of curves in the research. Figure 2.1 below displays the layout of each curve and its associated advisory speed.

Figure 2.1. Curves used in the analysis and their corresponding advisory speeds.

The equipment used during data collection was a Samsung Galaxy Tab 2 with a downloaded Android Application called Data Recording, a BBI, and a video camera. The Data Recording Application includes an accelerometer and a Global Positioning System.

Once the Application has run its course, it produces three files that can be used for analysis. The BBI was installed on the vehicle's dashboard and leveled using a flat driveway and a spirit level. The BBI (vial in this case) displays a value that represents the superelevation, lateral (centripetal) acceleration, and vehicle body roll. A small black ball moves from side to side as the vehicle turns. As the ball moves, it travels along tick marks each representing a value of 1 degree. The amount the ball travels in one direction or another is based upon the three characteristics previously mentioned. The limits of the device are -20 degrees and 20 degrees. Figure 2.2 shows the BBI used.

Figure 2.2. BBI used during data collection.

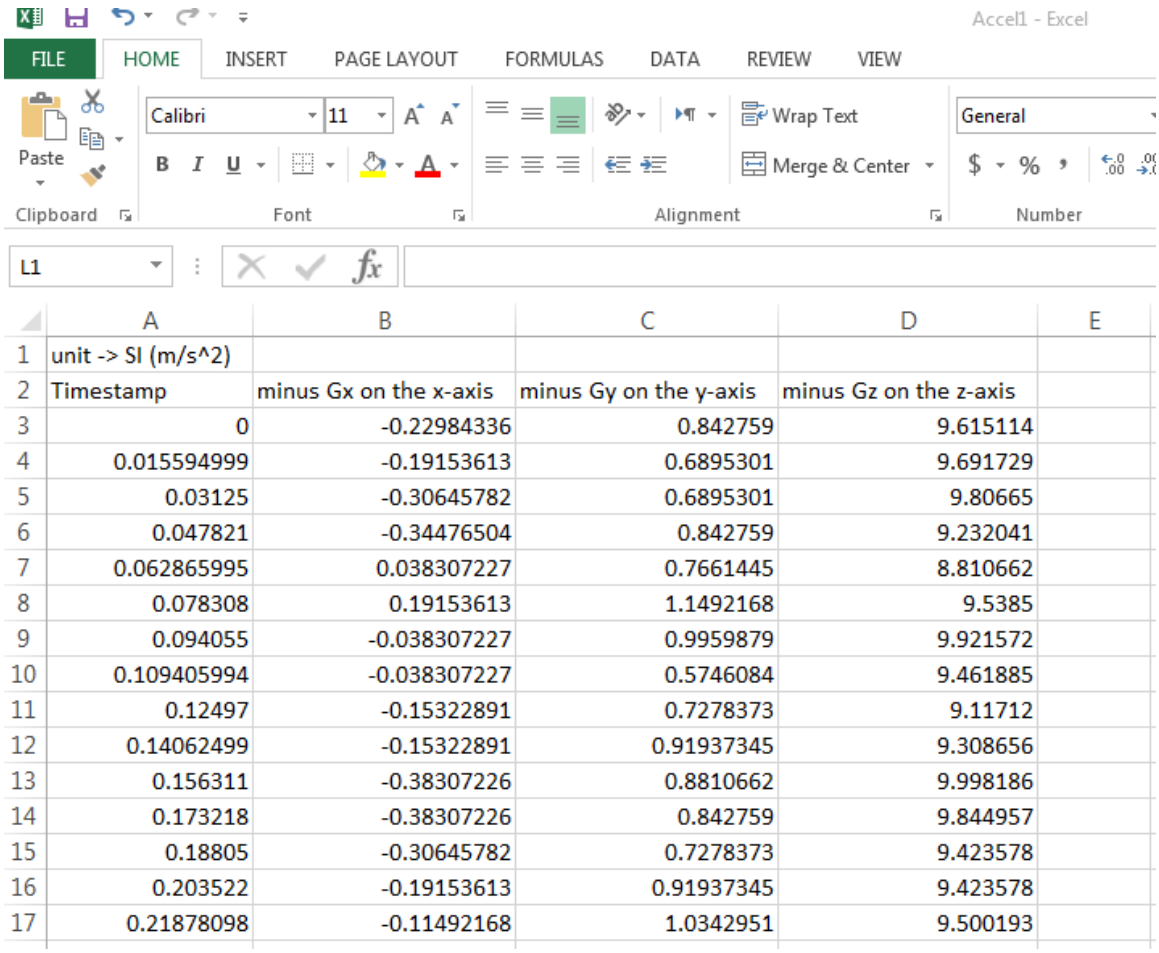
The process involved a driver and two passengers who collected the data. The driver traversed the series of curves in one drive while one person held the Android Application (Data Recording) to record the accelerometer data, and one person video recorded the BBI installed on the vehicle's dashboard. The accelerometer operator was told to start and stop the Application at the same points on the curves for each run. A video camera was used so that the collected BBI values could later be watched repeatedly and slowed down to ensure accurate readings. Each curve was driven at 4 different speeds – the advisory speed, 5 mph below, 5 mph above, and 10 mph above. For each speed, the process of data collection was repeated 4 times to ensure accuracy.

After quickly reviewing the gathered data, the method for data collection was altered. Initially, the data was collected using two passengers – one to control the video camera and one to record accelerometer values. The accelerometer data resulted in error since the device was not held flat and at times the operator started the application after the curve began. It was then decided that the tablet should be strapped flat to the center console and leveled on the same driveway that was used to level the BBI. The accelerometer was operated at the beginning and end of the entire series - eliminating the need for an additional person. This created one long recording for each run instead of four single recordings. Only a driver and one passenger were involved in data collection. The passenger started the accelerometer, filmed the BBI during the entire drive, and then stopped the accelerometer and camera when the run was complete. For this process, the latitude and longitude would have to be known for each curve so that only the necessary values are included in analysis. This is described in the next section.

Each curve was driven at 4 different speeds – the advisory speed, 5 mph below, 5 mph above, and 10 mph above. For each speed, the process of data collection was repeated 4 times to ensure accuracy.

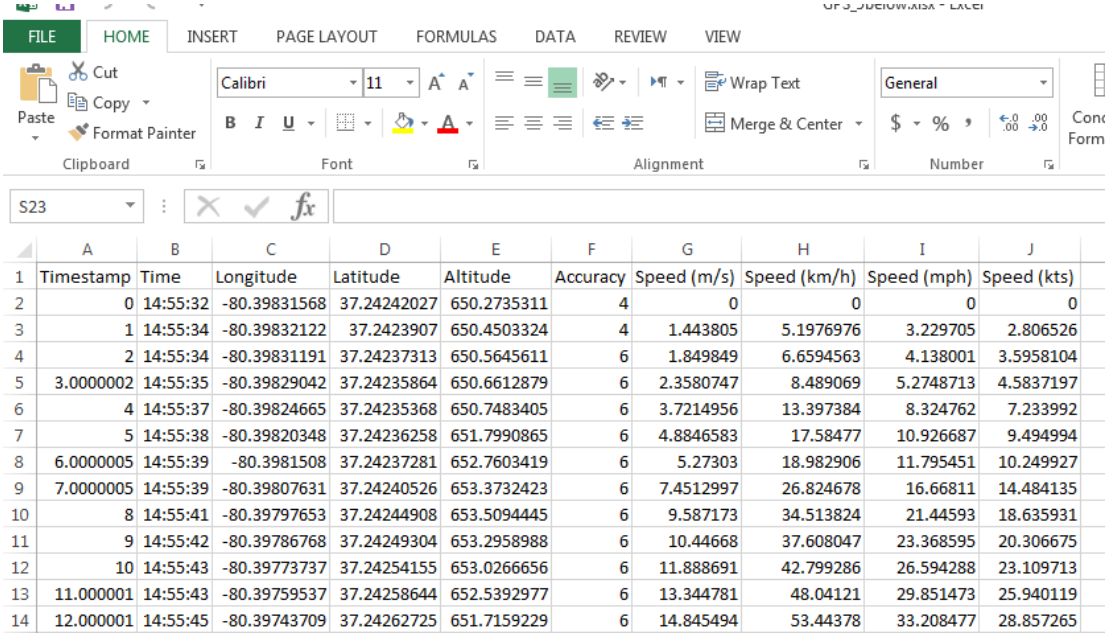
2.2 DATA ORGANIZATION

The accelerometer files were then exported and entered into Excel to create graphs. The Data Recording Application outputs three files for each recording: ACCELEROMETER.txt, LOCATION.txt, LOCATION_KML.kml. Figures 2.3 and 2.4 display a sample of the ACCELEROMETER and LOCATION files after being imported into Excel. The Keyhole Markup Language (KML) file was used to display the route in Google Earth. A screenshot is shown in Figure 2.5.



	A	B	C	D	E
1	unit -> SI (m/s^2)				
2	Timestamp	minus Gx on the x-axis	minus Gy on the y-axis	minus Gz on the z-axis	
3	0	-0.22984336	0.842759	9.615114	
4	0.015594999	-0.19153613	0.6895301	9.691729	
5	0.03125	-0.30645782	0.6895301	9.80665	
6	0.047821	-0.34476504	0.842759	9.232041	
7	0.062865995	0.038307227	0.7661445	8.810662	
8	0.078308	0.19153613	1.1492168	9.5385	
9	0.094055	-0.038307227	0.9959879	9.921572	
10	0.109405994	-0.038307227	0.5746084	9.461885	
11	0.12497	-0.15322891	0.7278373	9.11712	
12	0.14062499	-0.15322891	0.91937345	9.308656	
13	0.156311	-0.38307226	0.8810662	9.998186	
14	0.173218	-0.38307226	0.842759	9.844957	
15	0.18805	-0.30645782	0.7278373	9.423578	
16	0.203522	-0.19153613	0.91937345	9.423578	
17	0.21878098	-0.11492168	1.0342951	9.500193	

Figure 2.3. Sample of ACCELEROMETER.txt file after being imported into Excel.



	A	B	C	D	E	F	G	H	I	J
1	Timestamp	Time	Longitude	Latitude	Altitude	Accuracy	Speed (m/s)	Speed (km/h)	Speed (mph)	Speed (kts)
2	0	14:55:32	-80.39831568	37.24242027	650.2735311	4	0	0	0	0
3	1	14:55:34	-80.39832122	37.2423907	650.4503324	4	1.443805	5.1976976	3.229705	2.806526
4	2	14:55:34	-80.39831191	37.24237313	650.5645611	6	1.849849	6.6594563	4.138001	3.5958104
5	3.0000002	14:55:35	-80.39829042	37.24235864	650.6612879	6	2.3580747	8.489069	5.2748713	4.5837197
6	4	14:55:37	-80.39824665	37.24235368	650.7483405	6	3.7214956	13.397384	8.324762	7.233992
7	5	14:55:38	-80.39820348	37.24236258	651.7990865	6	4.8846583	17.58477	10.926687	9.494994
8	6.0000005	14:55:39	-80.3981508	37.24237281	652.7603419	6	5.27303	18.982906	11.795451	10.249927
9	7.0000005	14:55:39	-80.39807631	37.24240526	653.3732423	6	7.4512997	26.824678	16.66811	14.484135
10	8	14:55:41	-80.39797653	37.24244908	653.5094445	6	9.587173	34.513824	21.44593	18.635931
11	9	14:55:42	-80.39786768	37.24249304	653.2958988	6	10.44668	37.608047	23.368595	20.306675
12	10	14:55:43	-80.39773737	37.24254155	653.0266656	6	11.888691	42.799286	26.594288	23.109713
13	11.000001	14:55:43	-80.39759537	37.24258644	652.5392977	6	13.344781	48.04121	29.851473	25.940119
14	12.000001	14:55:45	-80.39743709	37.24262725	651.7159229	6	14.845494	53.44378	33.208477	28.857265

Figure 2.4. Sample of LOCATION.txt file after being imported into Excel.

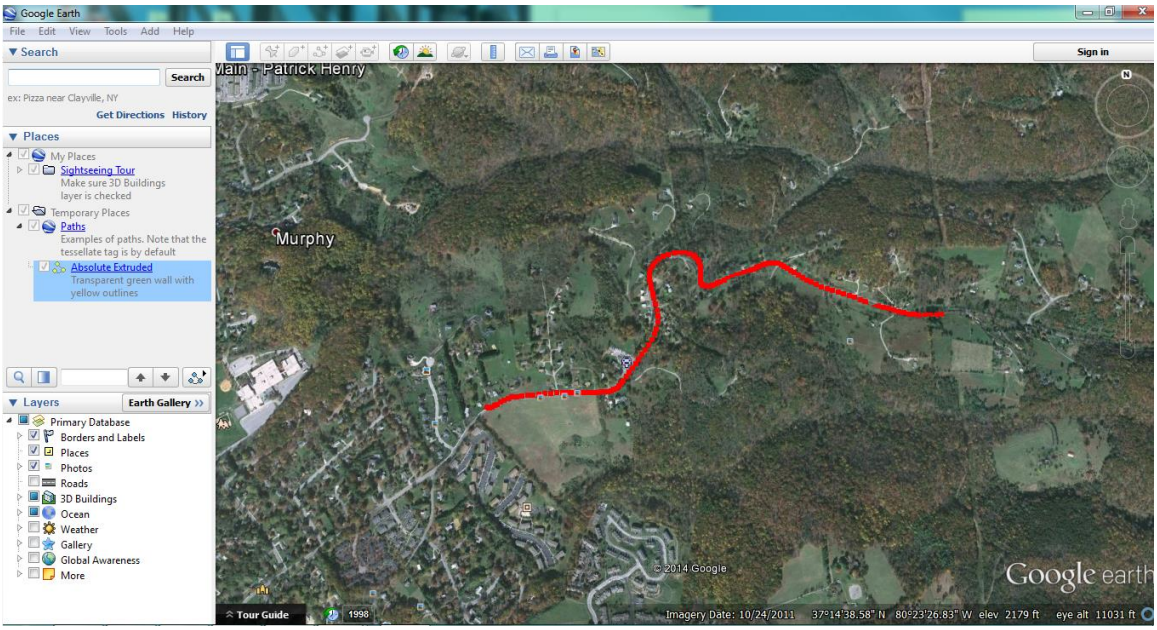


Figure 2.5. Screenshot of kml file. All curves along Harding Avenue are shown in one file.

After exporting the data and reviewing the numbers, Curves 2 and 6 were removed from the research due to their curve geometry. Both curves included a vertical component in addition to the horizontal element. Since the object of this research is to focus on the x-component aspect of the curve, it is superfluous to include any other components and data would be skewed as a result. Once Curve 2 and Curve 6 were removed, the curves were renamed as seen in Figure 2.6.

Figure 2.6. Updated curves used in the analysis and their corresponding advisory speeds.

The latitudes and longitudes were determined for the beginning and end of each curve through Google maps. The following are the approximate latitudes and longitudes of each curve:

Table 2.1: Locations of Each Curve

		Curve 01	Curve 02	Curve 03	Curve 04
Begin	Longitude	-80.3948	-80.3883	-80.3829	-80.3854
	Latitude	37.2427	37.2464	37.2450	37.2489
End	Longitude	-80.3934	-80.3865	-80.3815	-80.3853
	Latitude	37.2430	37.2462	37.2448	37.2499

The latitudes and longitudes of each curve were found within the LOCATION file and the corresponding times were recorded. The times were then matched in the ACCELEROMETER file to distinguish the set of data for each curve.

2.3 RESULTS

The raw data was plotted so that each x-component value was matched with its corresponding time value. This produced graphs with smooth trends overall but with many fluctuations in the data set. Figure 2.7 illustrates the results.

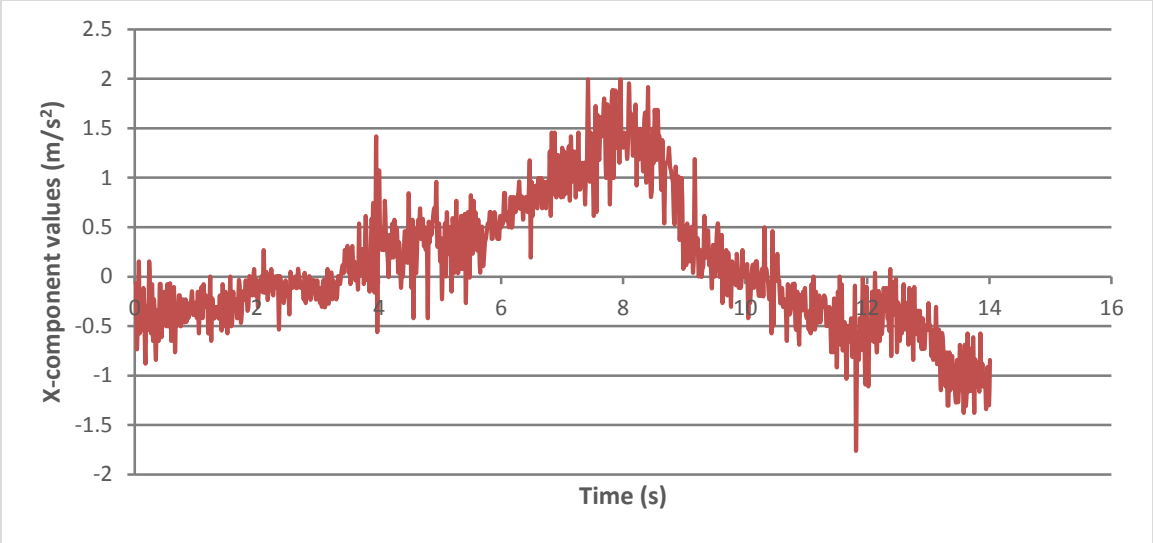


Figure 2.7. Curve 02 driven at 25 mph. Raw data for Run 1 plotted.

Moreover, this created issues when trying to extract the largest x-component value. It is evident that there are some spikes in the data created by the sensitivity of the accelerometer and using these numbers would create an inaccurate relationship with the BBI. A smoothing process needed to be applied for adequate analysis.

The data within the Excel spreadsheets were then manipulated to smooth the graph and eliminate spikes in the plot. To do this, the average of the x-component values was taken every 0.5 seconds. The time interval of 0.5 seconds was chosen because in the results it was seen that the x-component values were so close to one another in time spans less than 0.5 seconds – there was little variation. In addition, as the driver traverses the curve, the change in the curve's characteristics every 0.5 seconds is very minimal so it would be pointless to choose a smaller time interval. Before plotting, the initial 0.5 seconds was set at 0 and the rest of the data was altered accordingly so the plot started at a y-value of zero. Averages were then plotted against their corresponding times. The table below shows a few points used for Curve 02 at 25 mph for Run 1.

Table 2.2: Averaged X-component Values Computed Every 0.5 Seconds (Curve 02)

Time (s)	Averaged x-component (m/s^2)
0.5	0
1.0	0.0150
1.5	0.0402
2.0	0.1395
2.5	0.2664
3.0	0.2616
3.5	0.2987
4.0	0.5789
4.5	0.6782
5.0	0.7417

This produced graphs that had distinct shapes based on the direction of the curve. Curves banking left produced mostly negative values while curves turning right had positive values. For each graph, there was a noticeable extreme point which can be attributed to the sharpest point in the curve. This point is significant because it is where the highest BBI value occurred. Below is a graph produced for Curve 02 using the described procedure. The most readable display, for a single run, employed a scatterplot as the chart type. This is shown in Figure 2.8.

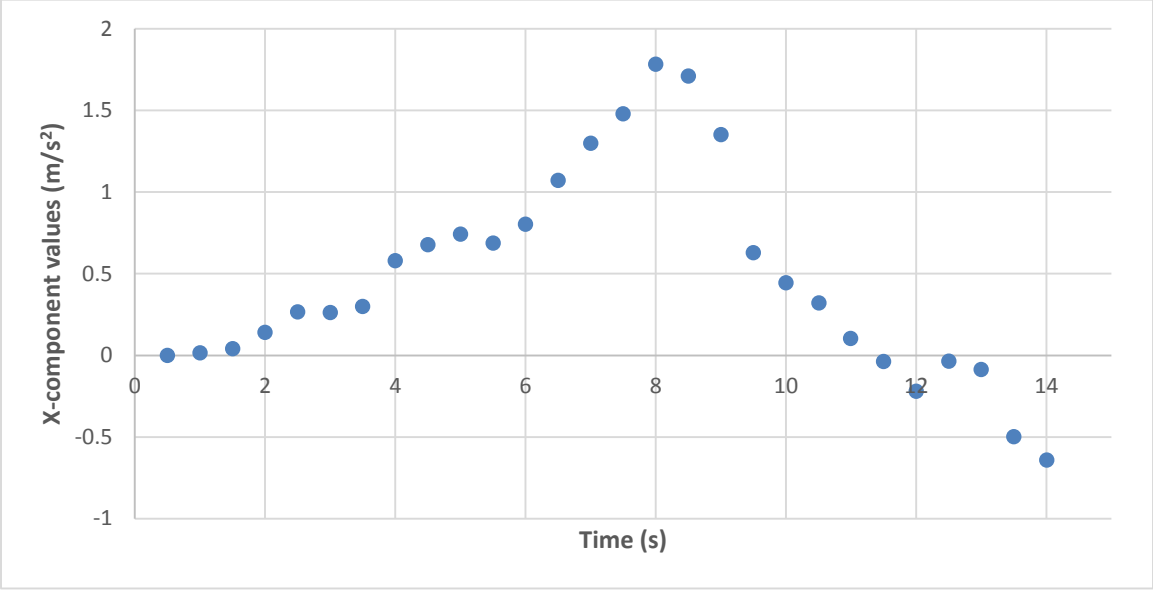


Figure 2.8. Scatterplot of data collected for Curve 02 during Run 1. Relates the x-component versus time.

However, when more than one run was plotted on the same graph, it was hard to discern locations of certain points. It was decided to connect the points on the graphs using colored lines to visually enhance the data set. It is important to note that the lines do not imply a continuous function. They are simply there to reinforce the graph visually. This can be seen in Figures 2.9, 2.10, 2.11, and 2.12. The resulting graphs for the other curves can be found in the Appendix.

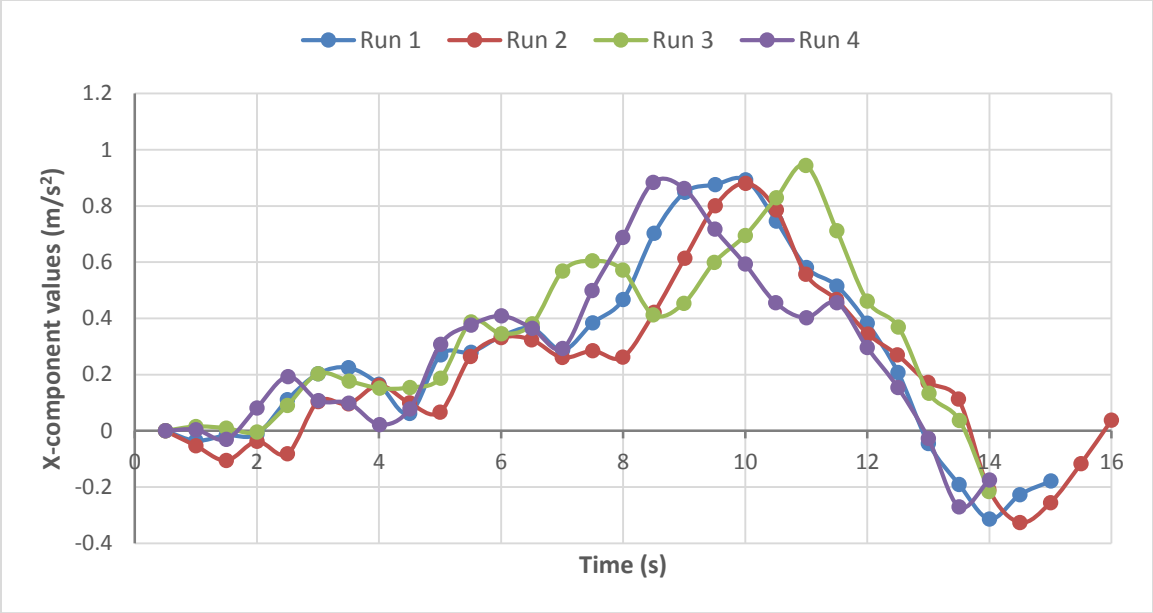


Figure 2.9. Final visual representation of all runs for Curve 02 at 20 mph.

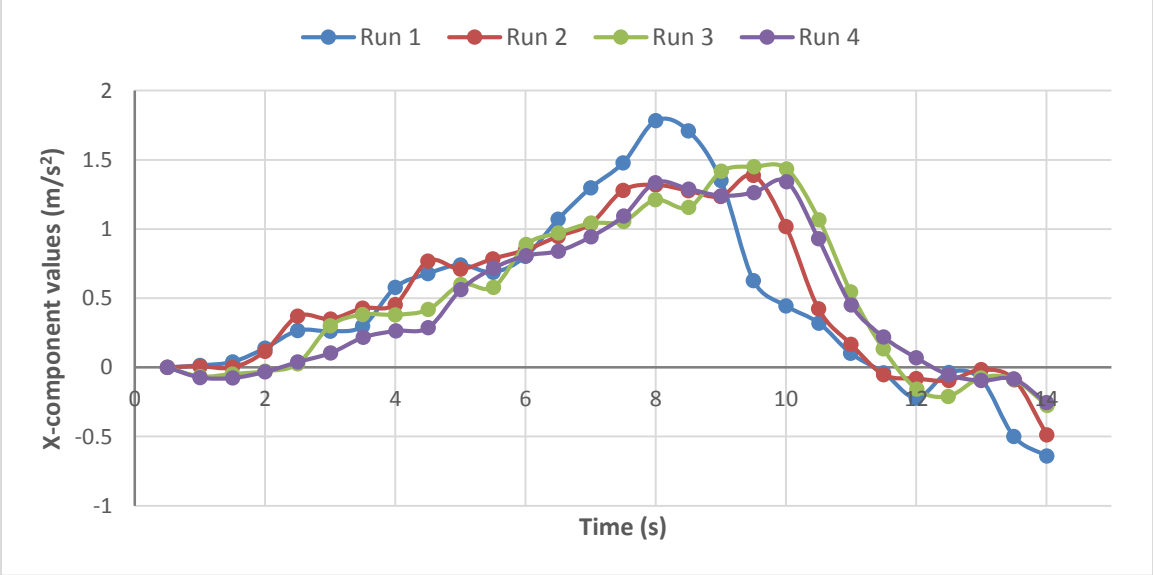


Figure 2.10. Final visual representation of all runs for Curve 02 at 25 mph.

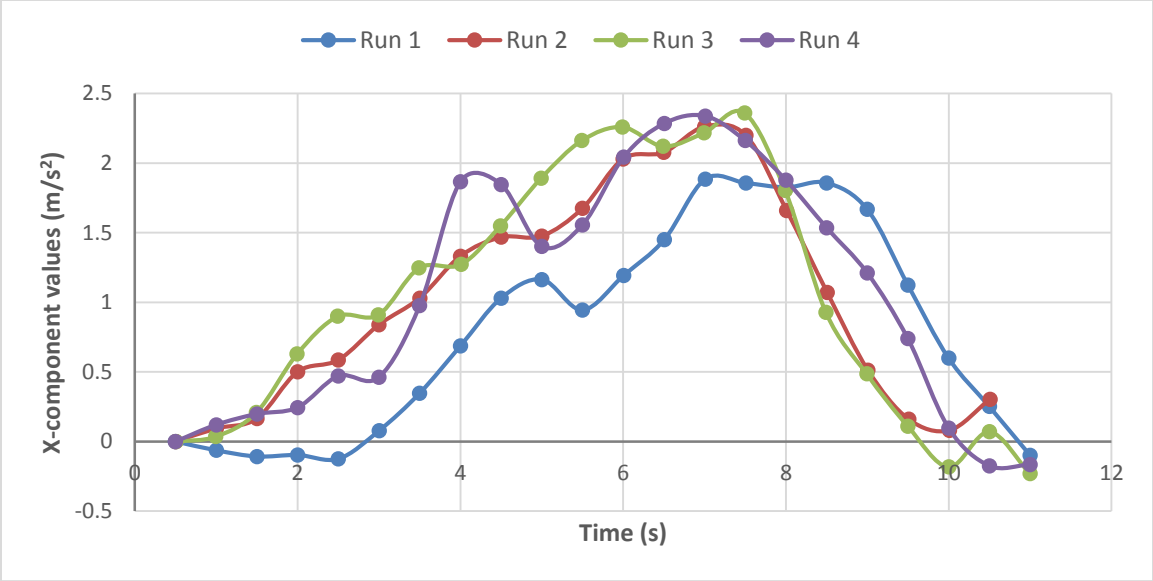


Figure 2.11. Final visual representation of all runs for Curve 02 at 30 mph.

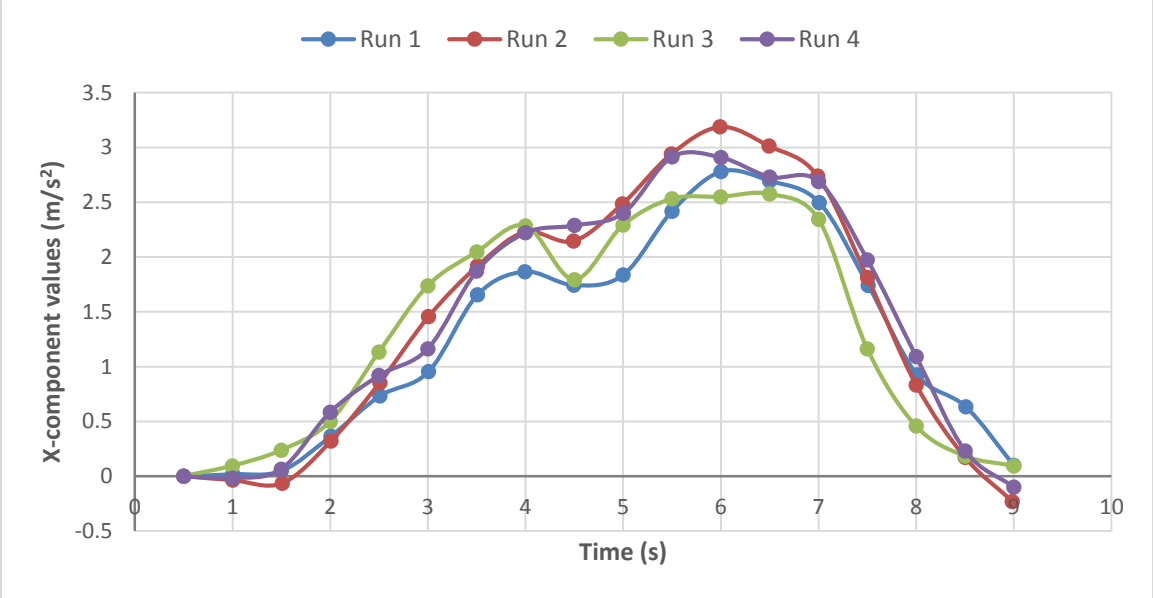


Figure 2.12. Final visual representation of all runs for Curve 02 at 35 mph.

For right turning curves, the maximum x-component value was extracted and for left turning curves the minimum x-component value was extracted. These values were then matched to their corresponding BBI readings observed in the videos. The results for Curve 02 are shown in Table 2.3 and all other curve results are in Tables A.2.1 and A.2.2 in the Appendix.

Table 2.3: X-components and Corresponding BBI Values for Curve 02

Curve Description	Curve Advisory Speed	Driven speed	Max X-component	Ball Bank Indicator
Curve 02 (Harding Avenue)	25	20	0.8927	-4
			0.881	-4
			0.9447	-5
			0.884	-5
		25	1.7831	-9
			1.3876	-7
			1.4508	-7
			1.3426	-7
		30	1.8845	-11
			2.2626	-11
			2.3603	-12
			2.3384	-12
		35	2.7796	-16
			3.1881	-18
			2.5748	-15
			2.9135	-16

Once all runs for all curves (64 total) were analyzed this way, the x-components were plotted against the BBI readings. A trendline was fit to the data points for each curve and a linear relationship was observed. An example is shown in Figure 2.13. All other Curve trendlines are located in the Appendix.

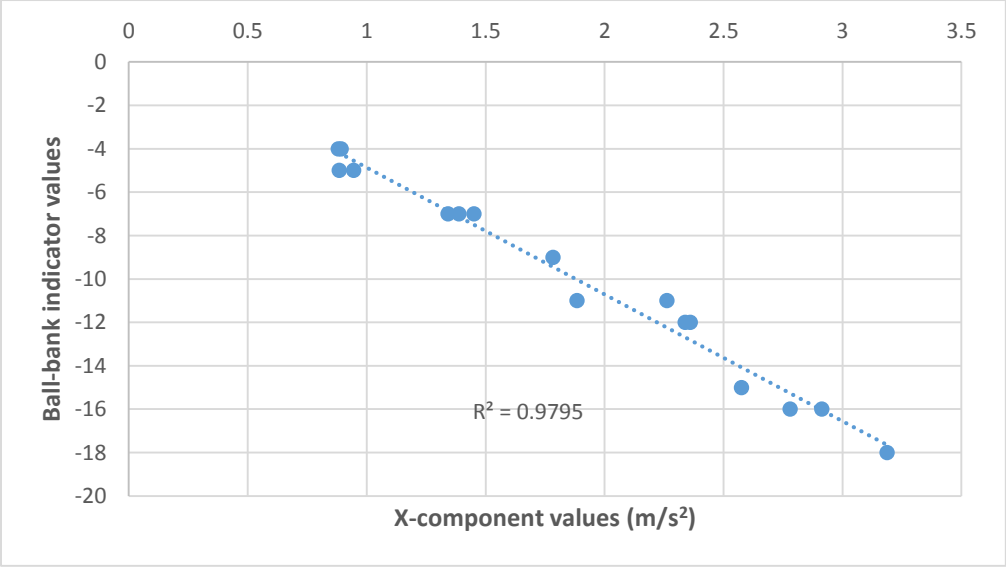


Figure 2.13. Relationship between the x-component values and their corresponding BBI values for Curve 02.

For Curve 04, the data related to 30 mph was omitted since it produced BBI values above 20 (threshold of the device) and therefore the exact BBI values could not be determined. Including this data would undesirably alter the trend for the overall relationship. In addition, it should be noted that the BBI readings could have minor error due to interpretation of the device. The center of the ball within the device was used as the basis of the reading. If the center of the ball fell between two tick marks, the reading was chosen as the larger number to be conservative. The R^2 values for each curve are shown in Table 2.4.

Table 2.4: R-squared Values for Each Curve

	Curve 01	**Curve 02**	**Curve 03**	**Curve 04**
R^2 value	0.9635	0.9795	0.5836	0.9827

The R^2 value for Curve 03 of 0.5836 is much lower in comparison to the other curves and a very poor value in general. It was predicted that the data from this curve may not be useful due to the shallow curve radius and the results proved this supposition to be true. Data taken from this curve will not fall in the line with the purpose of this research. Therefore, this curve was omitted from the rest of the analysis.

The x-component values and BBI values of Curves 01, 02, and 04 were then displayed on one plot. The intercept was set as zero and this produced a trendline that had a very reasonable R^2 value of 0.9711 which can be seen in Figure 2.14. The equation of the line was displayed and recorded for further use in the forthcoming Android Application.

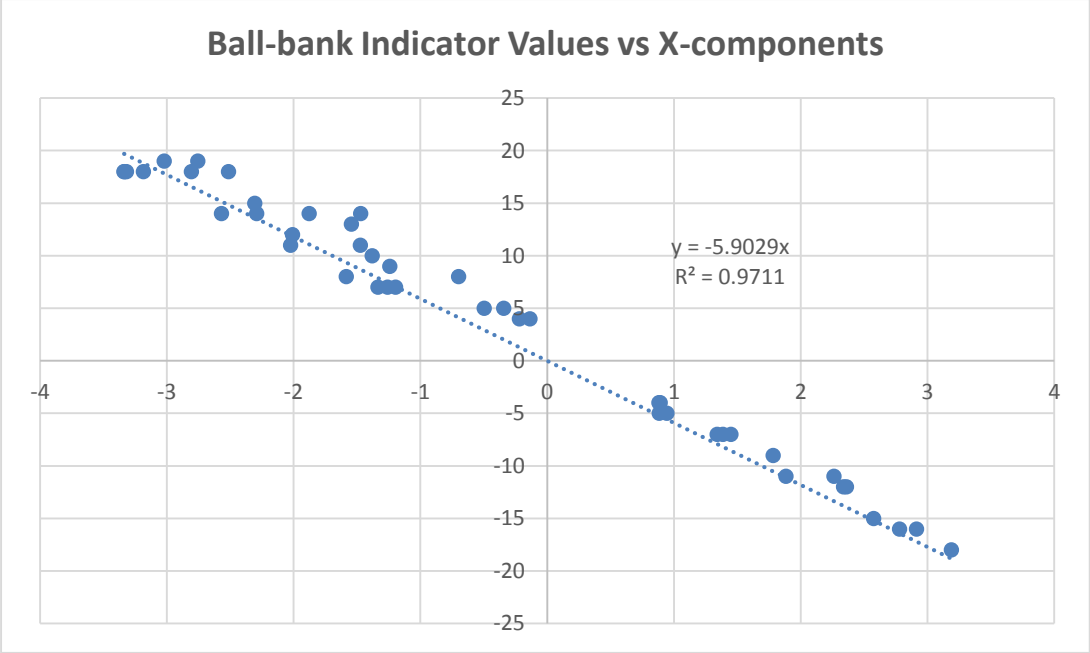


Figure 2.14. Relationship between maximum averaged x-components and their corresponding BBI values for Curves 01, 02, and 04.

After data was collected through the accelerometer and BBI, spot speeds were recorded for each curve using a hand-held radar. Curves 01 and 02 include data for about 50 vehicles while Curve 04 only has 30 vehicles. Curve 04 is an extremely low volume road and so it was acceptable to use a lower data count. The observed speeds can be seen in the distribution graphs below.

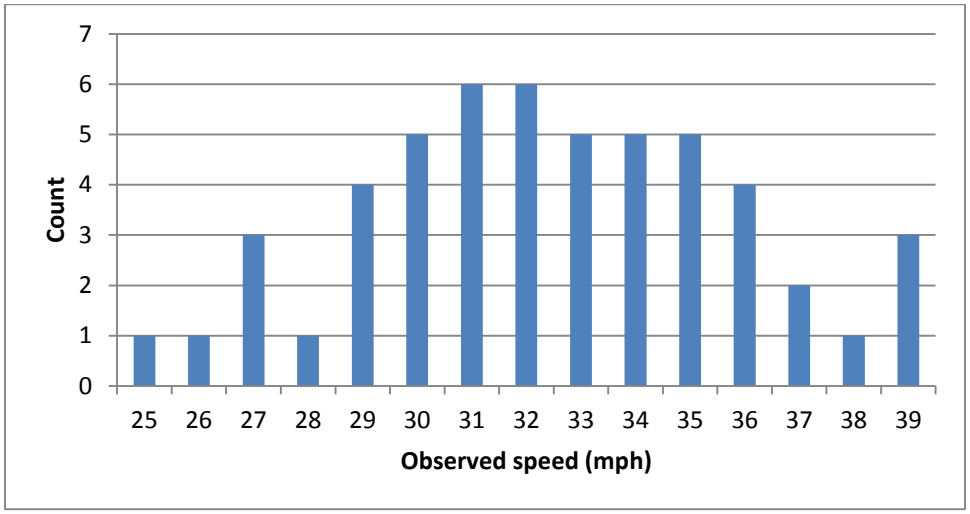


Figure 2.15. Spot speeds observed on Curve 01.

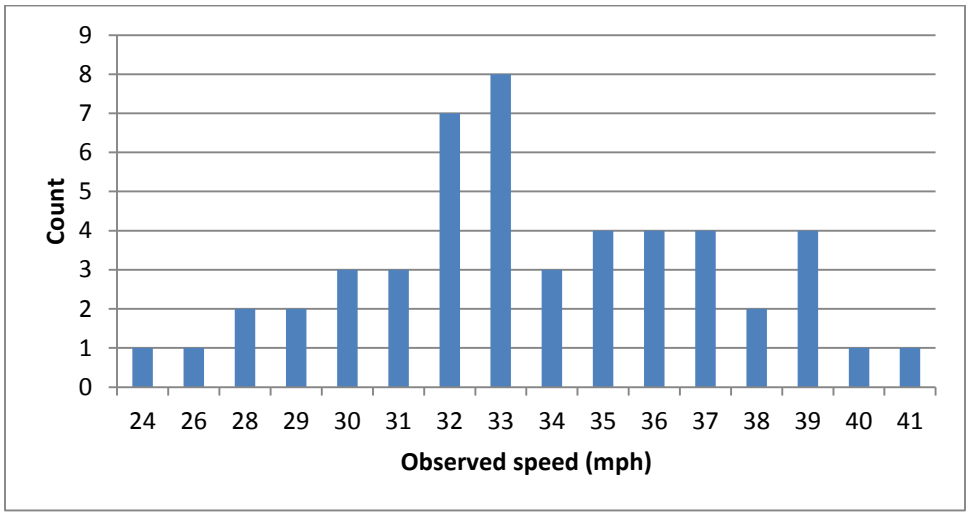


Figure 2.16. Spot speeds observed on Curve 02.

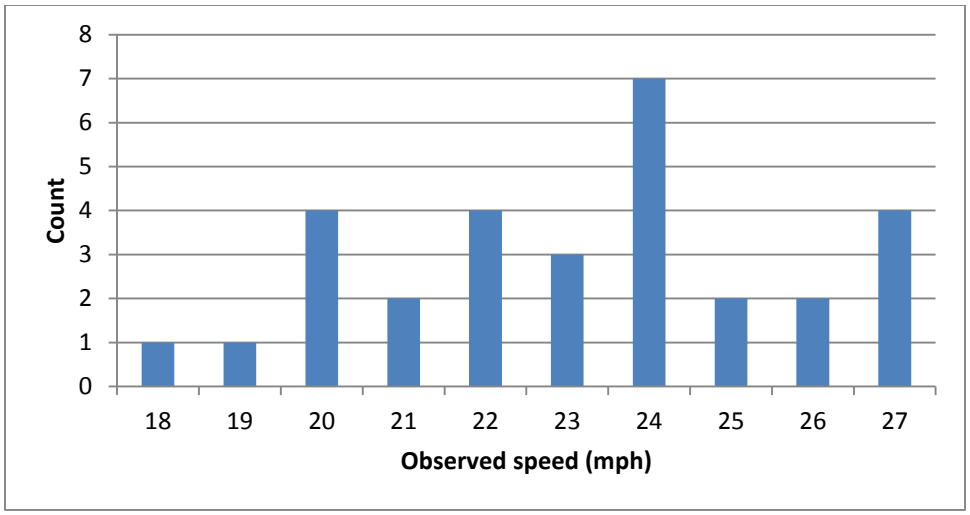


Figure 2.17. Spot speeds observed on Curve 04.

The mean, median, and 85th percentile were calculated from the spot speeds on each curve. Table 2.5 displays the statistics.

Table 2.5: Spot Speed Statistics for Each Curve

	Curve 01	Curve 02	Curve 04
Advisory Speed	25 mph	25 mph	20 mph
Mean	32.4 mph	33.6 mph	23.1 mph
Median	32.0 mph	33.0 mph	23.5 mph
85th %	35.6 mph	37.3 mph	25.8 mph

The 85th percentile was calculated as follows:

1. Divide the count of a specific speed by the total vehicle count to determine the cumulative percent.
2. Locate the cumulative percent of 85. If it lands exactly on one speed, then that is the 85th percentile. If it falls between two speeds, proceed to step 3.
3. Use the following equation[9]:

$$S_D = \frac{P_D - P_{min}}{P_{max} - P_{min}}(S_{max} - S_{min}) + S_{min}$$

where: S_D = speed at P_D
P_D = percentile desired
P_{max} = higher cumulative percent
P_{min} = lower cumulative percent
S_{max} = higher speed
S_{min} = lower speed

A sample calculation for Curve 01 will be performed to demonstrate the process. Step one for Curve 01 is displayed in Table 2.6. The total count is 52 vehicles. For example, 1/52 = 1.9% for the first row. The next row is 1/52 added to the previous row which equals 3.8%. This process is continued for each speed.

Table 2.6: Calculating 85th Percentile for Curve 01 – Step 1

Curve 01		
Count	Speed (mph)	Cumulative %
1	25	1.9
1	26	3.8
3	27	9.6
1	28	11.5
4	29	19.2
5	30	28.8
6	31	40.4
6	32	51.9
5	33	61.5
5	34	71.2
5	35	80.8
4	36	88.5
2	37	92.3
1	38	94.2
3	39	100.0

Since the 85th percent falls between 35mph and 36mph, Step 3 is conducted. Below are the calculations:

$$S_D = \frac{85 - 80.8}{88.5 - 80.8}(36 - 35) + 35 = 35.6\,mph$$

2.4 CONCLUSIONS

By applying a smoothing process to each test run's x-component values (averaging the x-components every 0.5 seconds) it can be seen that the values increase until a certain point which is determined as the sharpest point of the curve. After this point, the values then decrease. The greatest BBI reading also occurs at this point. Due to this, a relationship can be drawn between the two parameters. A linear trend was observed and produced the equation y = -5.9029x, where y is the BBI reading and x is the highest averaged x-component. This equation will be used for advisory speed assignments within the impending Application.

The 85th percentile speeds are approximately 10 mph above the posted advisory speed. This demonstrates that the advisory speed assignment criteria is not providing

reasonable ranges. They are too conservative. Even the mean and median speeds are higher than the advisory speed by 7 mph for Curve 01 and 02 and 3 mph for Curve 04. This portion of the research reveals how drivers react to curves versus the posted advisory speeds which are determined by transportation associations such as AASHTO.

CHAPTER 3

PROPOSED TOOL

The main goal of this research is to create a tool that is easily operated and can calculate the appropriate advisory speed on a horizontal curve. Through the use of an online website called App Inventor[10], the desired Application was created and called CurveAdvisor. This Android Application can be installed on any Android device by downloading the associated .apk file. The following sections describe a step-by-step process of how the Application was designed and how it operates. This chapter also includes Application testing and results.

3.1 OVERVIEW OF MIT APP INVENTOR 2

The Massachusetts Institute of Technology (MIT) created a website where anyone can learn the basics of Android Application coding and even design their own Android Application. It is called MIT App Inventor 2 and can be readily accessed through http://appinventor.mit.edu/explore/. To create an Application, the user must first sign in with a Google account. The layout of the design process is separated into two interfaces – Designer and Blocks Editor. The Designer element shows how the Application will display on an Android device and can be viewed in Figure 3.1. The other interface is the Blocks Editor (Figure 3.2) where the user creates functions, equations, and the overall coding for the Application by piecing colored blocks together. This is a very user-friendly process that simplifies the typical coding algorithms that many users may not understand. A wide variety of blocks are available; they are labeled Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. Examples of some control blocks are shown in Figure 3.3.

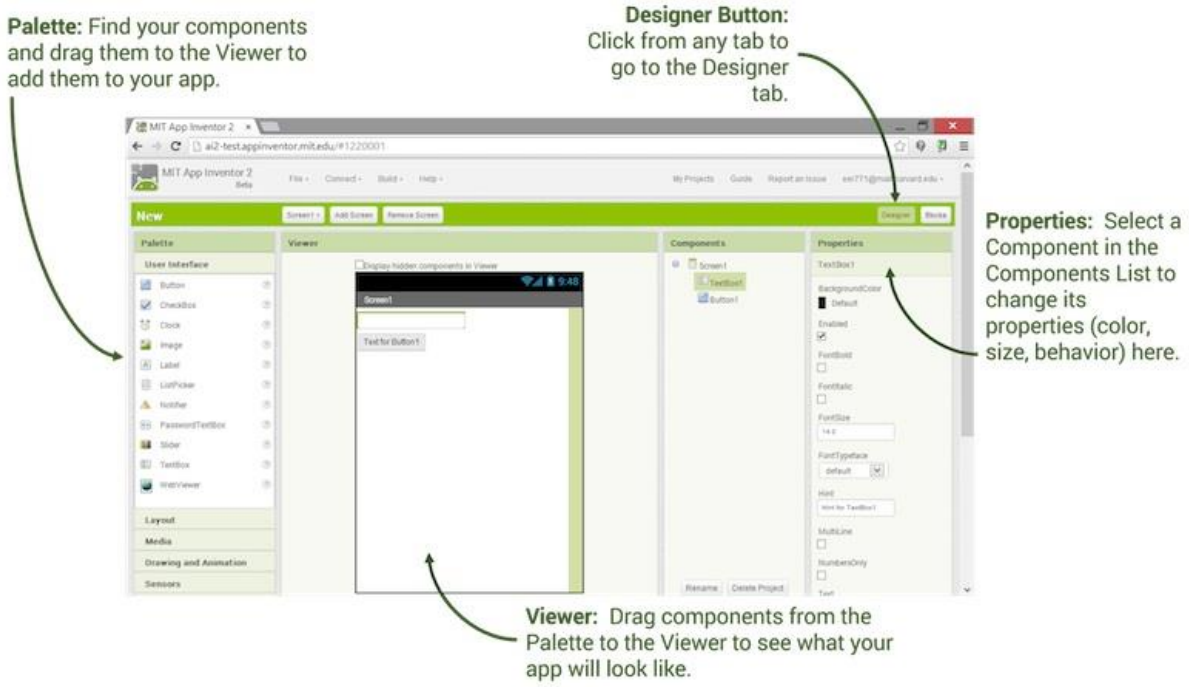


Figure 3.1. Designer interface[10]. Massachusetts Institute of Technology. App Inventor. 2012; Available from: http://appinventor.mit.edu/explore/designer-blocks.html. Used under fair use, 2014.

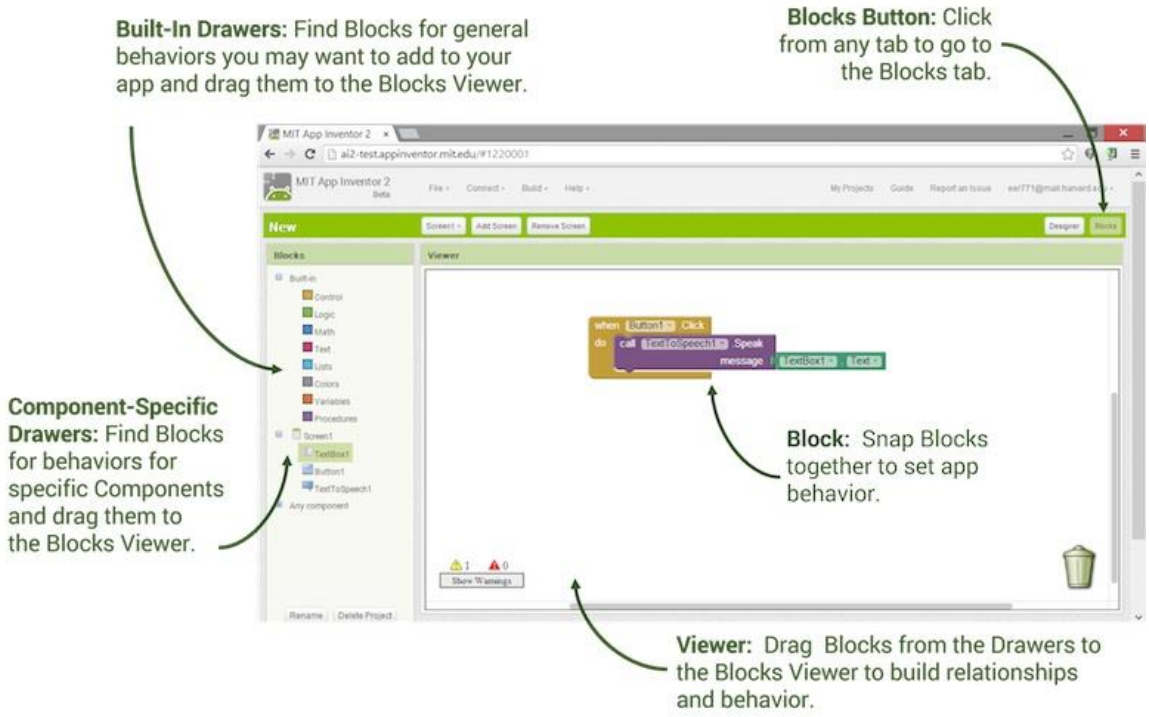


Figure 3.2. Blocks Editor interface[10]. Massachusetts Institute of Technology. App Inventor. 2012; Available from: http://appinventor.mit.edu/explore/designer-blocks.html. Used under fair use, 2014.

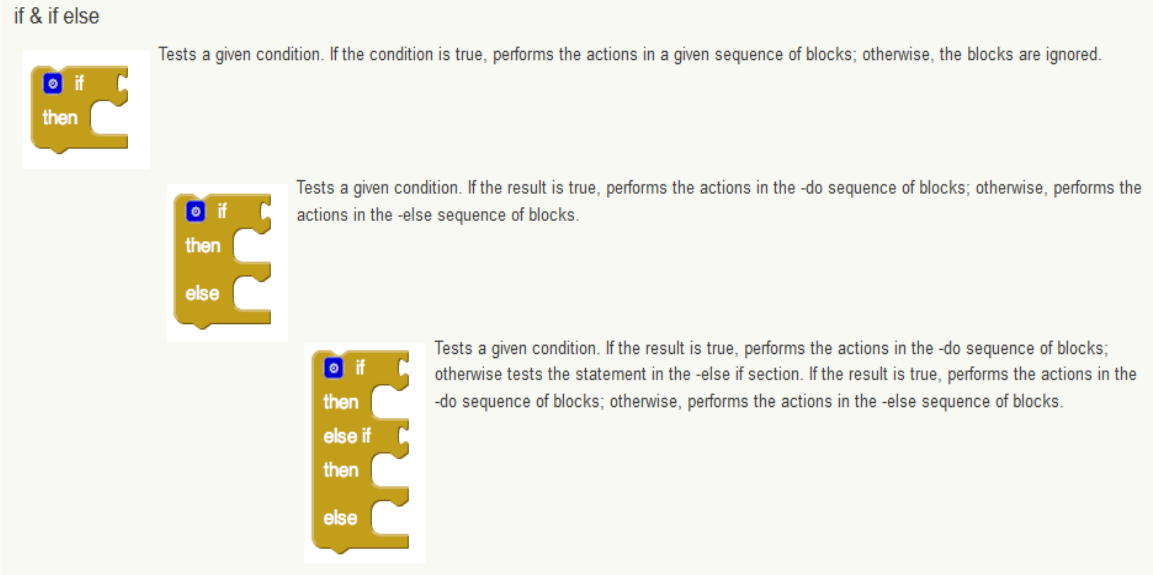


Figure 3.3. A sample of control blocks available in App Inventor [10]. Massachusetts Institute of Technology. App Inventor. 2012; Available from: http://appinventor.mit.edu/explore/ai2/support/blocks/control.html. Used under fair use, 2014.

3.2 MODEL DESCRIPTION

CurveAdvisor began with the general goal of outputting an advisory speed based on the collected parameters and the equation derived through data collection. The first step was to construct an outline of CurveAdvisor:

1. User touches START button on device when curve begins
2. Accelerometer and Timer begin and values are collected
3. CurveAdvisor calculates the average x-component value every 0.5 seconds and only stores the largest value as the process continues
4. User touches STOP button when curve ends
5. Accelerometer and Timer stop
6. CurveAdvisor uses the highest averaged x-component value and the equation found earlier (y = -5.9029x) to output the BBI value
7. CurveAdvisor displays recommendations based on AASHTO and MUTCD criteria

To begin, the display was formed through the Designer interface shown in Figure 3.1. Three Buttons from the Palette were dragged to the interface and named START, STOP, and Reset. Then, AccelerometerSensor (renamed Accelerometer), Clock (renamed Timer), and File were dropped into the interface. Labels and other organizational elements were introduced. Figure 3.4 shows the initial layout in Viewer and Figure 3.5 presents the list of components used to create the display in Figure 3.4.

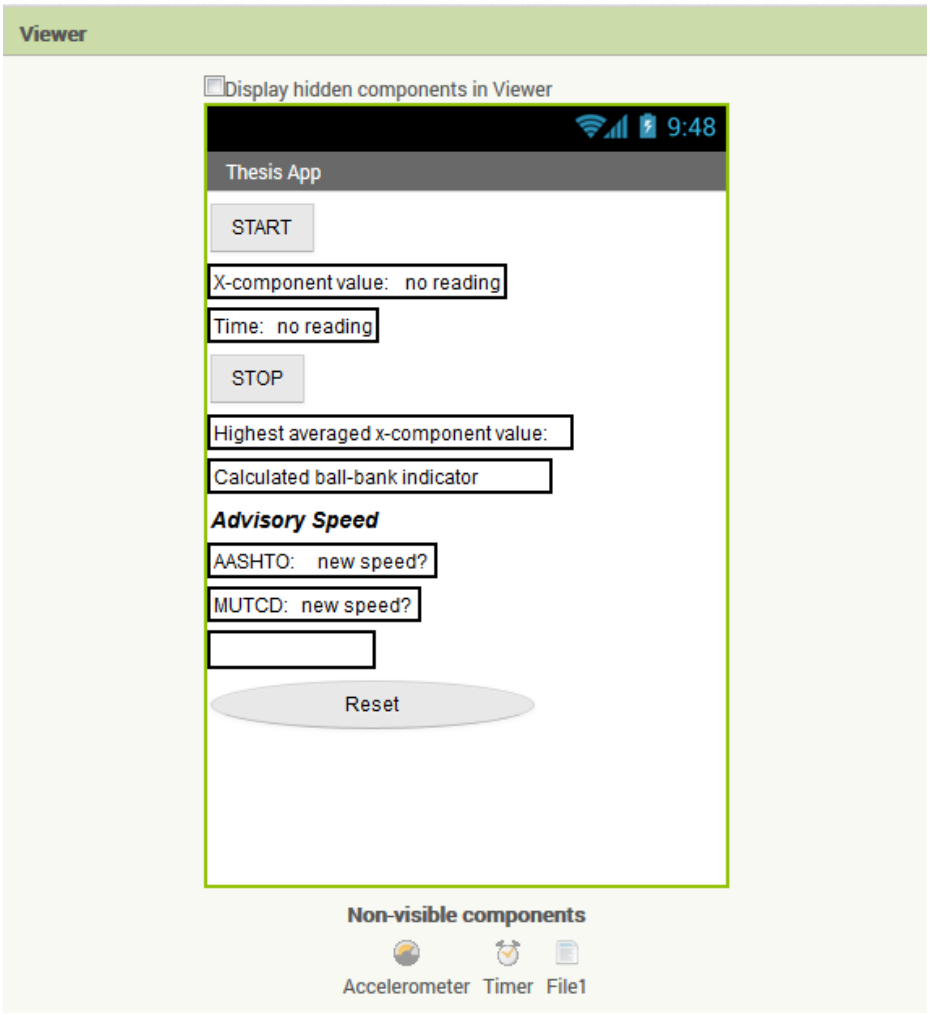


Figure 3.4. Initial Viewer layout for CurveAdvisor.

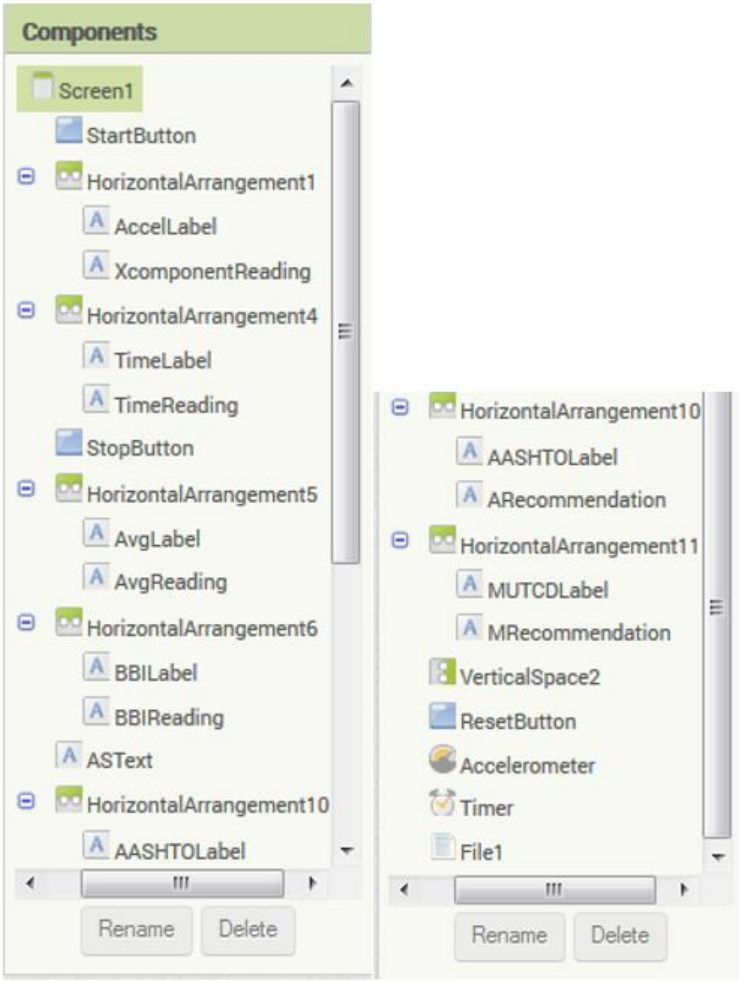


Figure 3.5. Components used to create display in Figure 3.4.

The list of components is in the order that they are shown in the Viewer. For example, the Start button is displayed first and the Reset button is displayed last. The non-visible components are at the very bottom of the list. There is also another column called Properties that appears as the user clicks each component. This enables the user to set the background color, width, height, font color, font size, font style, and other properties for any component.

After the visual setup was complete, coding began within the Blocks Editor. First, the Start and Stop button had to be incorporated. Figure 3.6 displays the sequence of events. The block colors represent different purposes and a summary is seen in Figure 3.7.

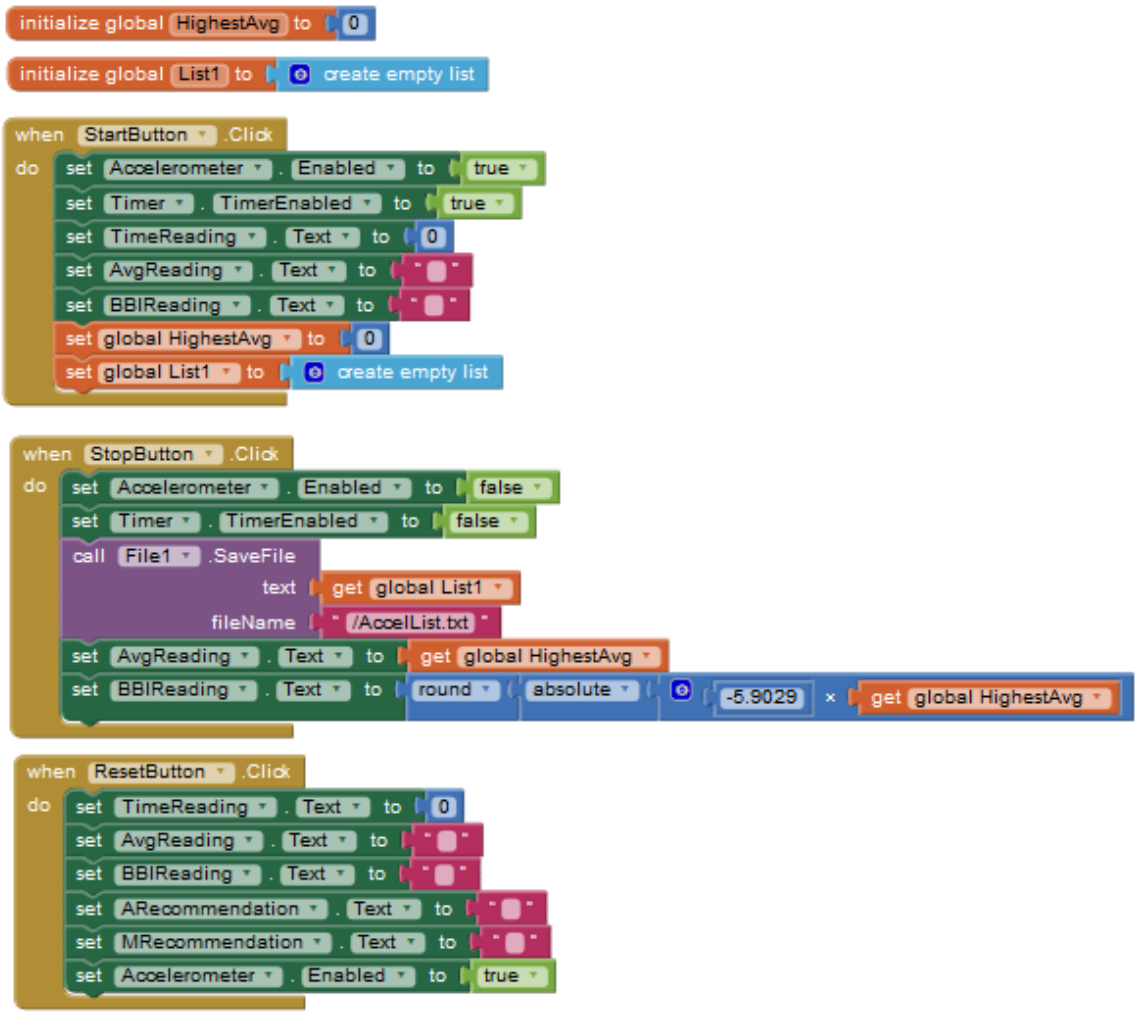


Figure 3.6. Incorporating buttons and simple functions to begin Application design.

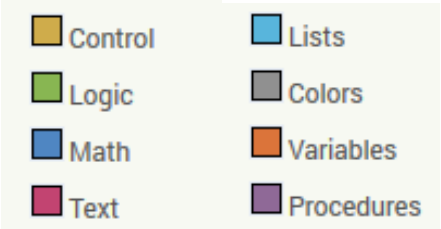


Figure 3.7. Key for block colors.

Starting from the top, the Application creates two global variables that can be used throughout the entire program. The HighestAvg is set to zero and this variable will change as calculations are completed. The calculation portion will be shown in an upcoming figure. The other global variable, List1, is a list that will be output as a .txt file once the Application is stopped. The list is contains each acceleration value collected so the user can check the individual values if need be.

The next event is the Start Button being enabled, or clicked. When this happens, the accelerometer and timer begin (enabled). The TimeReading is set to zero and the

AvgReading and BBIReading are set as empty text. In addition, the two global variables are set to their initial state just in case the Start button is pressed again during the same use. If this was not included, the variables would use the values from the previous run. CurveAdvisor needs to start fresh each time.

When the Stop button is clicked, the accelerometer and timer stop (disabled). A .txt file is created from the List1 variable and can be accessed in the storage files within the Android device. This file will be useful when viewing the collected x-component values. The process of writing this list will be shown in an upcoming figure. The AvgReading outputs the highest averaged x-component value (before taking the absolute value) and the BBIReading displays the corresponding BBI value based on the derived equation from data collection.

The Reset button sets all variables back to their original state. This button should be pressed before each new run. However, in case the user forgets to tap the Reset button, clicking the Start button again will initialize all variables as well. This should only be done when the user is ready to collect data.

Figuring out the sequence of calculations was the next course of action. The tool was designed so that it closely matched how the data was collected earlier with the two most important components being the accelerometer and the timer. During data collection, the x-component was recorded every 0.015 seconds. Therefore, the ultimate goal for CurveAdvisor was to fire a timer every 0.015 seconds and collect the x-component values at each time interval. This would fall in line with the process used for earlier data collection and therefore increase accuracy of CurveAdvisor. It would also collect more values because even if the x-component does not change within that time interval, the number would still be collected. This would produce a different average than if the acceleration was solely collected each time it changed. However, due to the limitation of the clock function within App Inventor, the timer interval does not fire as accurately as it should. It was seen that there was slight error in the first interval and the error amplified as time progressed. A test was conducted where a simple timer was run at an interval of 0.015 seconds with only the process of writing a file. Viewing this file showed that the interval was off by 0.001 seconds to begin then increasingly worsened in its value, sometimes skipping 0.030 seconds. This proved that the problem was not the set interval or amount of activity associated with the timer (such as calculations) - it was an inaccurate component from the beginning. To correct this issue, two methods were composed and compared:

1. Solely use the Timer control block to conduct all calculations while simultaneously collecting acceleration values. The Timer performs actions whenever the preset time interval (in this case 0.015 seconds) is reached.

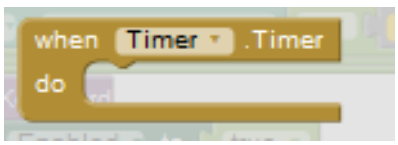


Figure 3.8. Timer control block.

2. Incorporate a trigger within the Timer control block that would initiate calculations within the AccelerationChanged control block every 0.5 seconds. This method uses both control blocks. The AccelerationChanged control block will perform actions whenever the acceleration changes in the X, Y, or Z dimension.

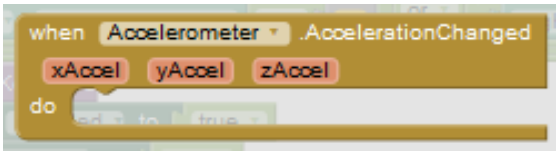


Figure 3.9. AccelerationChanged control block.

For simplification purposes the Timer control block will be referred to as Ttest and the Timer and AccelerationChanged control block combination will be referred to as Atest. Each test was set up so that each produced two files – a file that solely collected the calculated time intervals (named ATime for the Atest and TTime for the Ttest) and a file that recorded the accelerometer values with a break between each new interval (named AAccel for the Atest and TAccel for the Ttest.) The main goal was to see which test produced a time interval closer to 0.015 seconds. The files output from these two tests were imported into Excel and compared. Screenshots of the ATime, AAccel, TTime, and TAccel .txt files can be found in the Appendix.

Based on the ATime and TTime files, it can be seen in Figure 3.10 that the Atest provided time intervals ranging from 0.013-0.151 seconds while the Ttest produced intervals ranging from 0.044-0.111 seconds. The counts of each time interval for each test are show in the figures below.

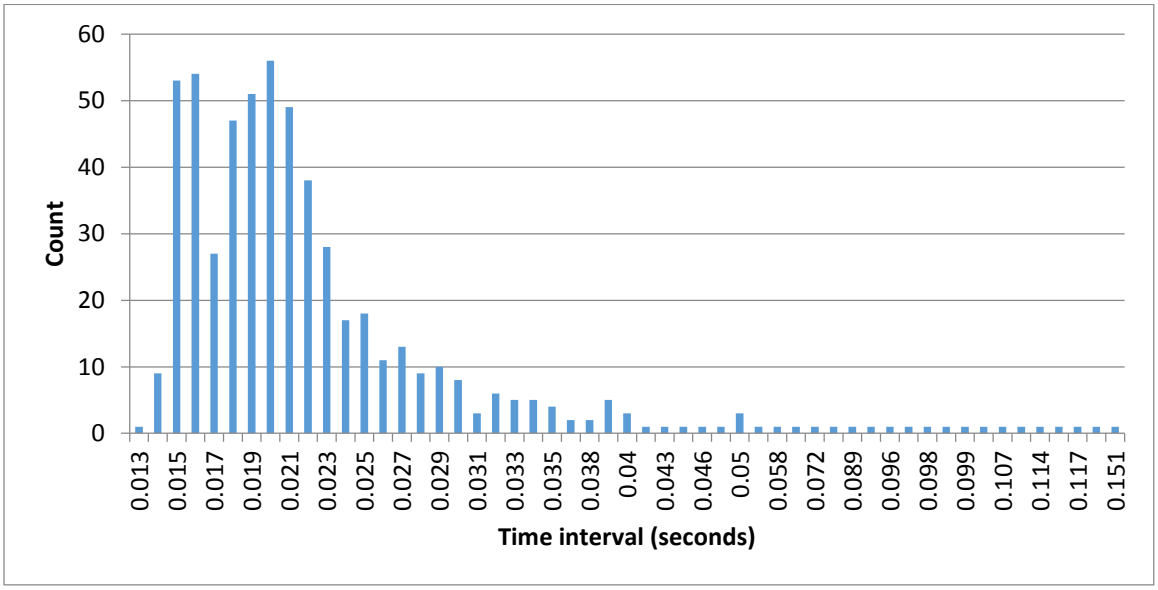


Figure 3.10. Distribution of time intervals within ATime file using the Atest.

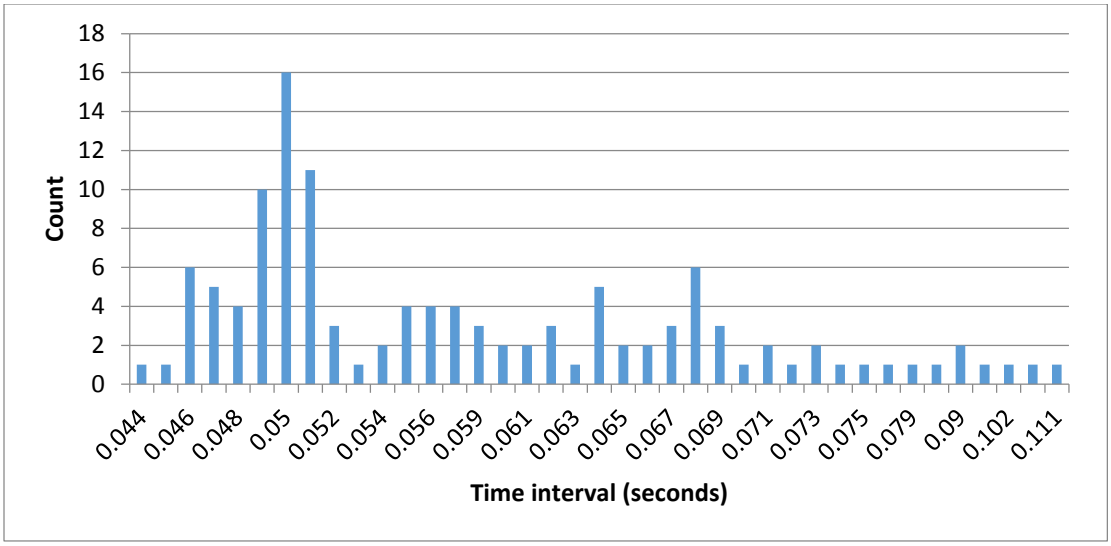


Figure 3.11. Distribution of time intervals within TTime file using the Ttest.

The mode for the Atest was 0.02 seconds and the mode for the Ttest was 0.05 seconds. In addition, the mean values were 0.024 seconds for the Atest and 0.059 seconds for the Ttest. These statistics prove that the Atest produced time intervals closer to the desired 0.015 seconds.

To further support this result, another assessment between the Atest and Ttest was conducted based on the collected accelerometer values (AAccel and TAccel files). These two files demonstrated that the Atest produced more acceleration values within each interval. For each time interval, the average number of collected values was 23 for AAccel and only 9 in TAccel. In addition, a series of zero values in the AAccel file proved that even when the acceleration stayed constant CurveAdvisor collected the values. This was the biggest concern of using the Atest method.

These numerous tests proved that the Timer and AccelerationChanged control block combination was more reasonable for data collection and thus was used when building CurveAdvisor into a final product. The Timer triggered calculations in the Accelerometer and from there, the coding was pieced together and new variables and inputs were introduced along the way. The new Viewer display is shown in Figure 3.12. The Reset button is just below the “MUTCD: new speed?” but could not fit within the screen capture.

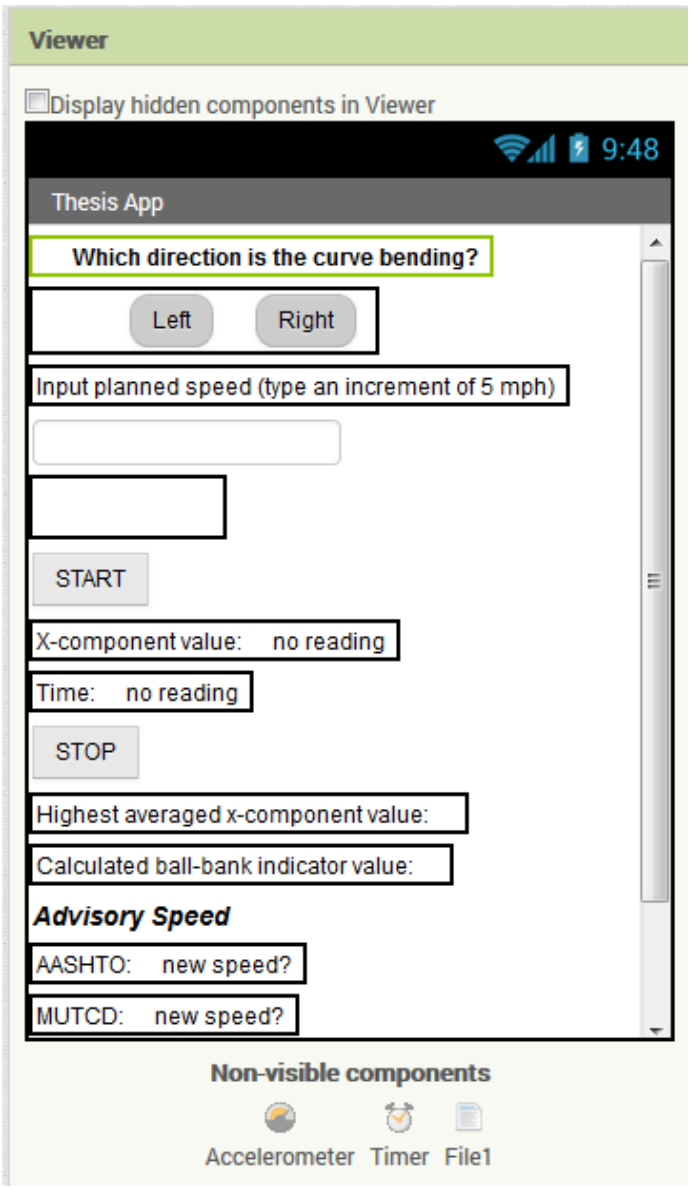


Figure 3.12. Updated interface for CurveAdvisor. Reset button is below “MUTCD: new speed?”

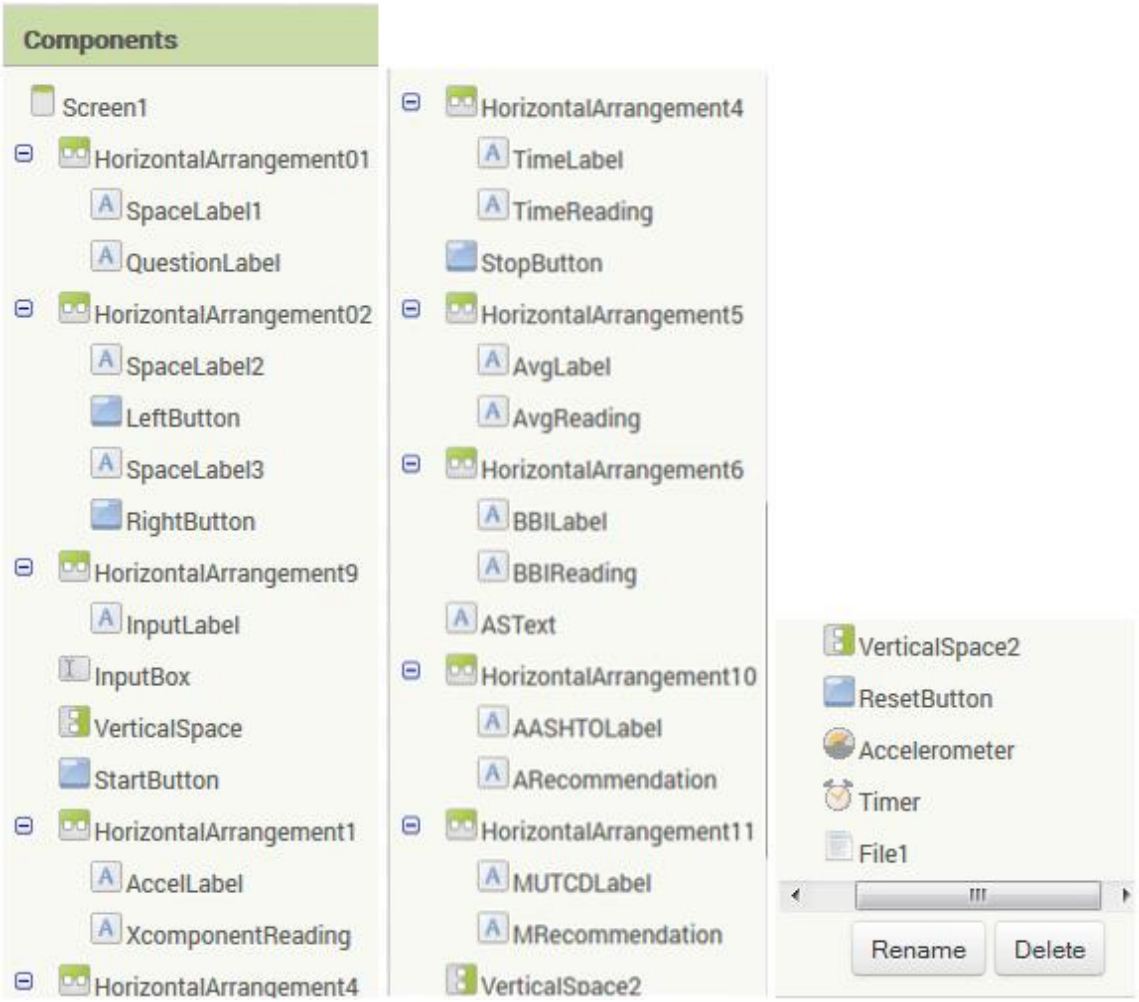


Figure 3.13. Components used to create display in Figure 3.12.

The complete Android code can be seen in Figures 3.14 and 3.15. Figure 3.14 contains the updated Start, Stop, and Reset control blocks with newly introduced global variables. To ensure only positive values were collected for the right turn and only negative values were collected for the left, an input on the display was added (InputBox) and incorporated in the coding. Only input values of 10 mph or higher were accepted because lower speeds would be unreasonable and futile. Traveling at speeds less than 10 mph has not been observed in the field and would consequently inhibit traffic flow. Figure 3.15 shows the timer and accelerometer components and the calculations within each.

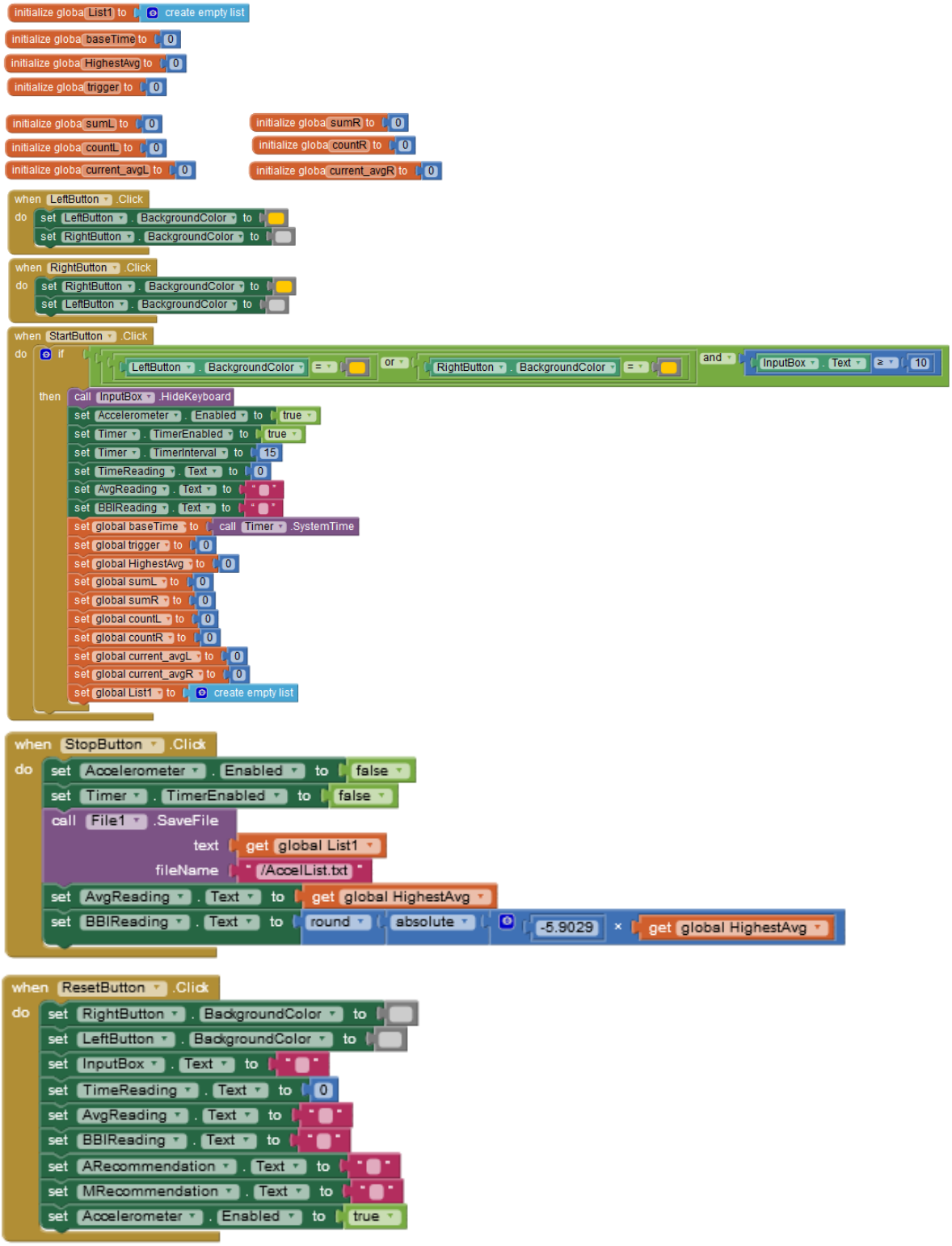


Figure 3.14. Updated Start, Stop, and Reset buttons with new variables.

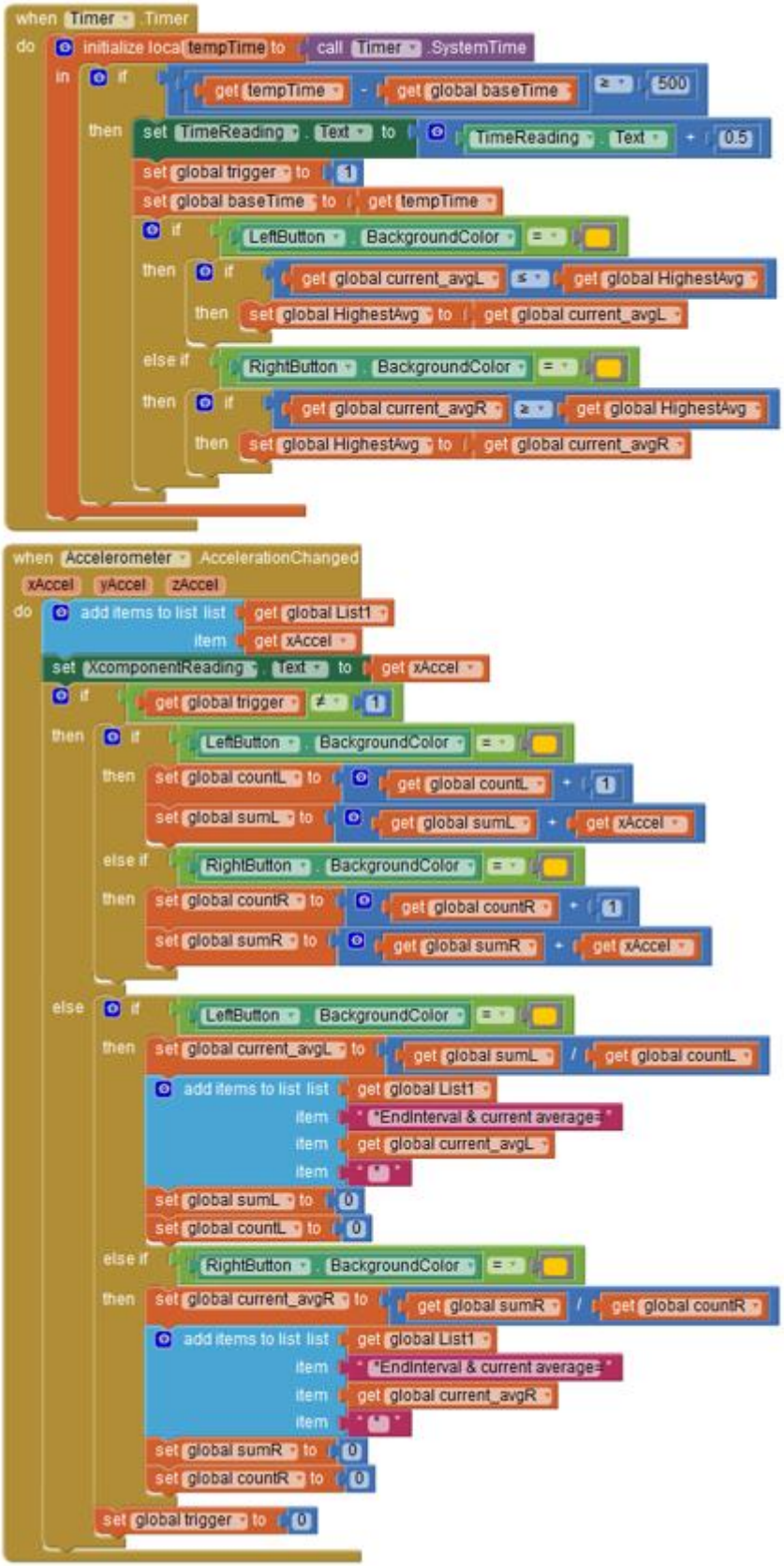


Figure 3.15. Timer and accelerometer components and their internal calculations.

A step-by-step description of the coding is displayed below for clarification purposes.

1. Initialize all global variables:
    a. List1 = empty list, used to output a file when Stop button is pressed
    b. baseTime = 0, used to compare time that has passed in order to fire trigger
    c. HighestAvg = 0, represents the highest averaged x-component value
    d. trigger = 0, used to trigger calculations within the AccelerometerChanged control block
    e. sumL, countL, current_avgL = 0, used during a left turning curve to sum all x-component values, count all x-component values, and calculate the current average for a specific time interval
    f. sumR, countR, current_avgR = 0, used during a right turning curve to sum all x-component values, count all x-component values, and calculate the current averages for each time interval
2. Input of left or right turn is read and highlights the corresponding button
3. StartButton control block:
    a. The Application will not begin unless one of the two buttons (Right or Left) is highlighted AND the user typed a speed greater than 10 in the InputBox
    b. The keyboard for the InputBox is hidden so that the display can be read during data collection
    c. Accelerometer and Timer are enabled
    d. TimerInterval is set to 15 milliseconds (0.015 seconds)
    e. The text for TimeReading, AvgReading, and BBIReading are set as 0, blank space, and blank space
    f. The variable baseTime is set as the current internal system time (which is a very large number)
    g. All global variables are set to their initial states as mentioned in Step 1 just in case the Application is run twice in one use
4. Timer control block (runs simultaneously with AccelerationChanged control block) fires every 15 milliseconds and:
    a. Initializes local variable tempTime as the current internal system time
    b. Calculates the difference between tempTime and baseTime to determine if the interval of 500 milliseconds (0.5 seconds) has been reached
    c. If false, then no action is taken
    d. If true, then
        i. TimeReading text increases by 0.5
        ii. Global variable trigger is set to 1
        iii. baseTime is reset to tempTime to start a new interval
        iv. If LeftButton is highlighted, global variable current_avgL is read from the AccelerationChanged control block and is

compared to global variable HighestAvg. If it is smaller, then HighestAvg becomes the current_avgL

v. Same process is completed if RightButton is highlighted except the maximum current average is kept instead of the minimum

5. AccelerationChanged control block (runs simultaneously with Timer control block) fires every time the acceleration changes in the X,Y, or Z direction and:
    a. The current XAccel value is added to the global list List1
    b. The XcomponentReading text is set as the current XAccel value
    c. Determines if the trigger is activated (trigger = 1)
    d. If false, then each XAccel is collected, counted, and summed based on the highlighted button
    e. If true, then
        i. The current average is calculated and stored as current_avgL or current_avgR based on which button is highlighted
        ii. "*EndInterval & current average=", the current average value, and "*" are added to List1 to designate a new interval within the list
        iii. The sum and count are reset to 0
        iv. The trigger is deactivated (set to 0)
6. StopButton control block:
    a. Accelerometer and Timer disabled
    b. A file created from List1 is output in the Storage files on the Android device and called AccelList.txt
    c. The AvgReading text is set to the variable HighestAvg which was calculated during Application execution
    d. Calculate and display the BBI by multiplying -5.9029 by the HighestAvg, taking the absolute value, and rounding the value

When this setup was complete, there was still one last piece to add: CurveAdvisor had to include the AASHTO and MUTCD criteria. Table 3.1 shows the criteria based on BBI readings and the associated x-component values found by using the equation generated in Chapter 2, y = -5.9029x. For instance, when the BBI reading is 14, 14 = -5.909x and so x = 2.37172.

Table 3.1: AASHTO and MUTCD Guidelines for Advisory Speed Assignment based on BBI and Calculated X-component Values

		AASHTO		MUTCD	
		BBI	X-component	BBI	X-component
Speed (mph)	**10**	14	2.37172	16	2.71053
	15	14	2.37172	16	2.71053
	20	14	2.37172	16	2.71053
	25	12	2.0329	14	2.37172
	30	12	2.0329	14	2.37172
	35	10	1.69408	12	2.0329
	40	10	1.69408	12	2.0329
	45	10	1.69408	12	2.0329
	50	10	1.69408	12	2.0329
	55	10	1.69408	12	2.0329
	60	10	1.69408	12	2.0329
	65	10	1.69408	12	2.0329

Based on this table, the following if/then statements were included in the Application:

If the highest x-component = the number in current index then keep current speed
If the highest x-component > the number in current index then decrease by 5 mph
If the highest x-component ≤ the number in next index then increase by 5 mph
If the highest x-component < the number in current index & > the number in next index then keep current speed

For example, in terms of AASHTO regulations, if the driver traverses a horizontal curve at 30 mph, then the BBI reading must not go over 12 and therefore the highest x-component can be no larger than 2.0329. If the highest averaged x-component value is 2.0329 exactly, then CurveAdvisor will recommend using 30 mph as the advisory speed. If the highest averaged x-component is above 2.0329, then CurveAdvisor will display a statement saying to try again at a speed 5 mph slower. If the highest averaged x-component value is less than or equal to 1.69408, then CurveAdvisor will suggest increasing the speed by 5 mph. And finally, if the highest averaged x-component value is less than 2.0329 but higher than 1.69408, then CurveAdvisor will say to use the current speed (30 mph). These constraints (portion before the word “then”) and outputs (portion after the word “then”) were included in the Stop button control block. Figure 3.16 shows the lists created based on Table 3.1 and Figure 3.17 shows the constraints and outputs from the if/then statements above.

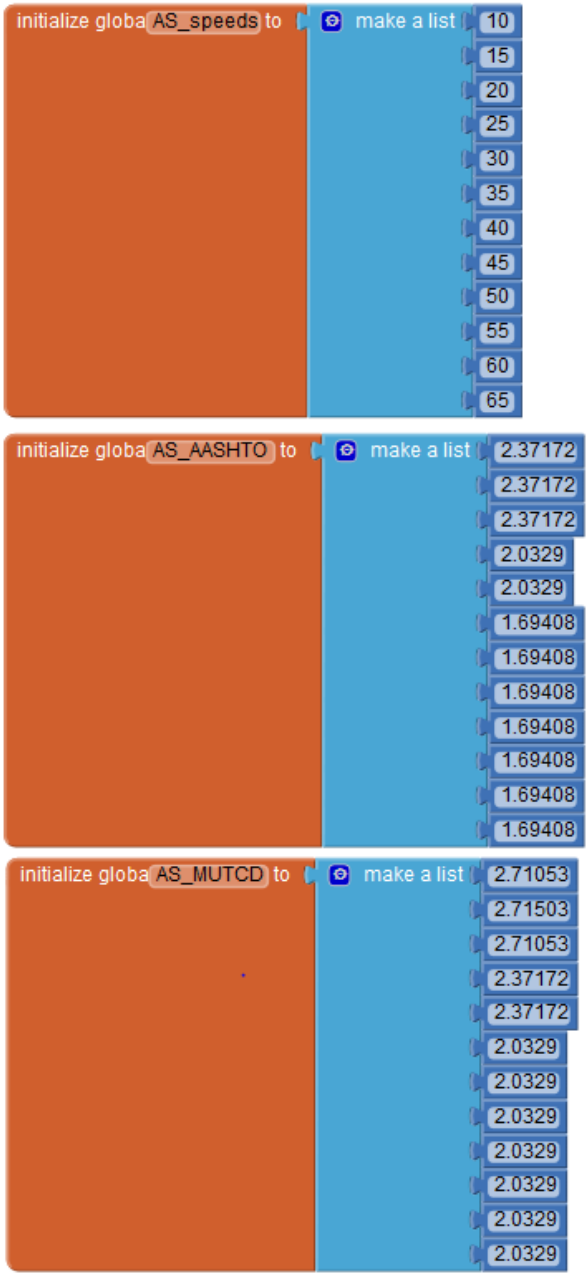


Figure 3.16. Criteria that are used within CurveAdvisor to output recommendations.

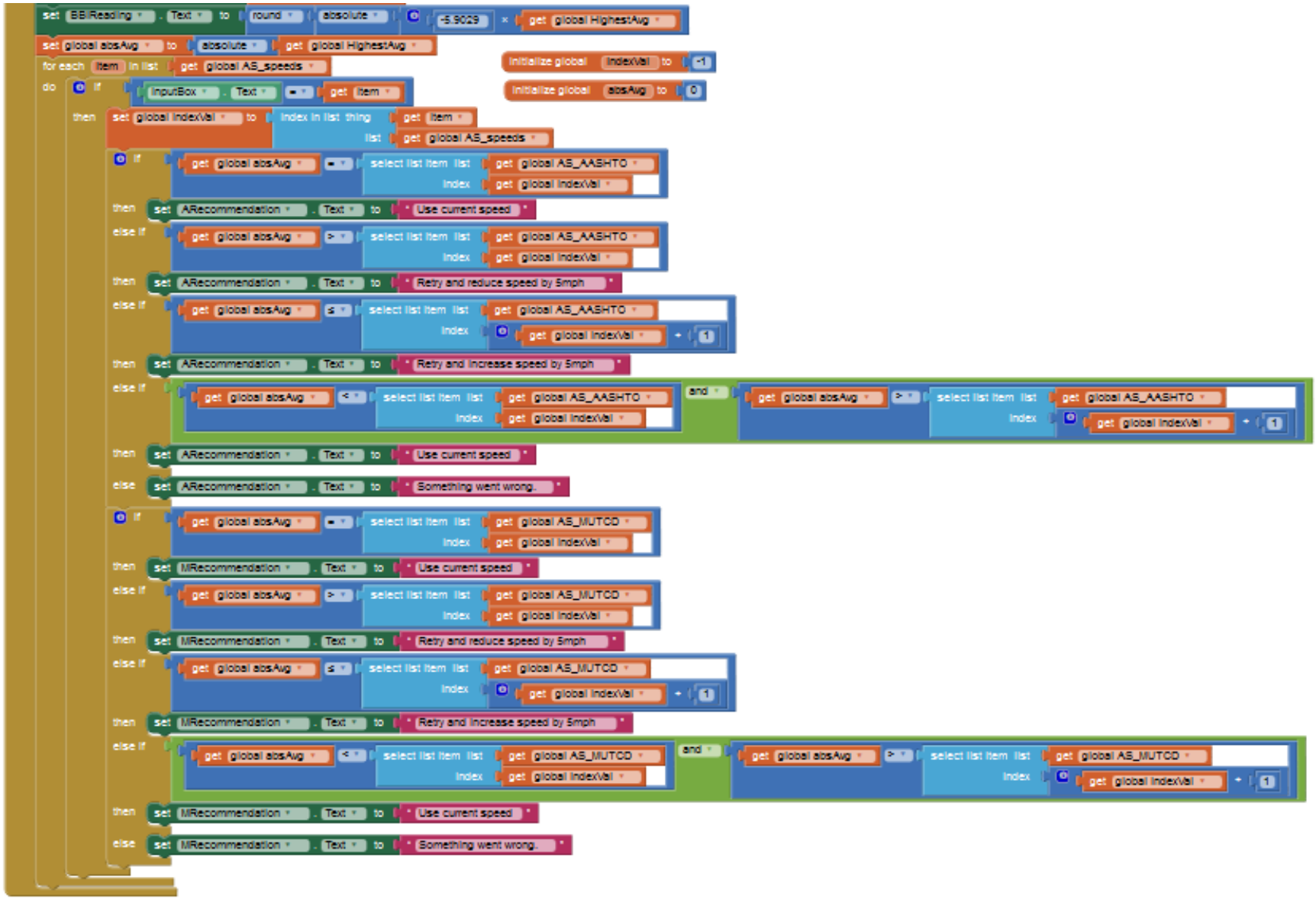


Figure 3.17. New addition to the Stop button coding - AASHTO and MUTCD criteria for advisory speeds.

Based on this new addition, new global variables indexVal and absAvg have to be initialized (added to Step 1) and Step 6 is updated with the following process:

6. StopButton control block:
    a. Accelerometer and Timer disabled
    b. A file created from List1 is output in the Storage files on the Android device and called AccelList.txt
    c. The AvgReading text is set to the variable HighestAvg which was calculated during the Application execution
    d. Calculate and display the BBI by multiplying -5.9029 by the HighestAvg, taking the absolute value, and rounding the value
    e. The variable absAvg takes the absolute value of the highest averaged x-component. Making this value positive allows CurveAdvisor to easily compare it to the AASTHO and MUTCD criteria.
    f. For each item in the AS_speeds list CurveAdvisor determines if the InputBox text matches it. If it does, then
        i. indexVal is set to the corresponding index number in the AS_speeds list

    <u>For AASHTO criteria:</u>

        ii. if absAvg = the number in current index (indexVal) in the AS_AASHTO list then output ARecommendation as “Use current speed”
        iii. if absAvg > the number in current index (indexVal) in the AS_AASHTO list then output ARecommendation as “Retry and reduce speed by 5mph”
        iv. if absAvg ≤ the number in next index (indexVal+1) in the AS_AASHTO list then output ARecommendation as “Retry and increase speed by 5mph”
        v. if absAvg < the number in current index (indexVal) in the AS_AASHTO list & > the number in next index (indexVal+1) in the AS_AASHTO list then output ARecommendation as “Use current speed”
        vi. if absAvg equals anything else then output ARecommendation as “Something went wrong”
        vii. if absAvg = the number in current index (indexVal) in the AS_AASHTO list then output ARecommendation as “Use current speed”

    <u>For MUTCD criteria:</u>

        viii. if absAvg > the number in current index (indexVal) in the AS_MUTCD list then output MRecommendation as “Retry and reduce speed by 5mph”
        ix. if absAvg ≤ the number in next index (indexVal+1) in the AS_MUTCD list then output MRecommendation as “Retry and increase speed by 5mph”

x. if absAvg < the number in current index (indexVal) in the AS_MUTCD list & > the number in next index (indexVal+1) in the AS_MUTCD list then output MRecommendation as “Use current speed”

xi. if absAvg equals anything else then output MRecommendation as “Something went wrong”

CHAPTER 4

EVALUATION

The designed Application, CurveAdvisor, needed to be tested to determine if it produced accurate results. This was first done in the office and then out in the field driving the curves. Curves other than those in the initial data set were included in order to analyze the consistency and successfulness of the Application (App).

4.1 TESTING THE ANDROID APPLICATION

To conduct live testing of CurveAdvisor, the Samsung Tablet was connected to the computer through the use of AI Companion. This allows the user to directly connect to the website where the App is designed and visually see how the App works using the tablet. First, the MIT AI2 Companion App must be downloaded and installed on the device. In addition, the device and the computer must be using the same Wi-Fi network. Once the MIT AI2 Companion App is opened, it asks for a 6 digit code. Through the App Inventor website there is an option to request a code for the project that is being designed. Clicking this option will supply a code that can be manually entered in the tablet. Once this is complete, the tablet was connected to the website.

First, the App was tested simply by holding the device and forcing it to reach a value that would output a desired BBI value. This would ensure that calculations were executing correctly. It was seen that everything was running smoothly. A few minor changes were made for organizational purposes and then the App was tested out in the field.

4.2 DATA COLLECTION

The equipment used during data collection was a Samsung Galaxy Tab 2 installed with CurveAdvisor, a ball-bank indicator (BBI), and a video camera. The process involved a driver and a passenger who operated the video camera and tablet. First, the tablet was strapped to the center console and leveled so that both the BBI and accelerometer were producing a value of zero. The driver traversed Curve 01, Curve 02, and Curve 04 one curve at a time. The passenger touched the Start button on CurveAdvisor's display at the beginning of the curve and video recorded the BBI throughout the run. A video camera was used so that the collected BBI values could be watched repeatedly and slowed down to ensure accurate readings. The passenger then touched the STOP button at the end of the curve. The results would display and CurveAdvisor would provide a suggestion based on the collected data and the internally coded advisory speed regulations. A screenshot of the final display was taken at the end of each run to keep

the output accessible for further analysis. The file produced from the run (AccelList.txt) was renamed within the tablet's storage files so that it would not be overwritten during the next run. The .txt files would later be exported and used for further evaluation.

Each curve was driven first at 5 mph below the posted advisory speed. For instance, Curve 01 was first traversed at 20 mph, Curve 02 at 20 mph, and Curve 04 at 15 mph. Depending on what CurveAdvisor suggested, a new run was conducted with a new speed.

In addition to the curves on Harding Avenue and Happy Hollow Road, two curves on Cedar Run Road were analyzed and named Curve A and Curve B. Figure 4.1 displays a map of the area.

Figure 4.1. View of Curves A and B. These curves were used to further test CurveAdvisor.

The same procedure was performed on Curves A and B except that the first run used the advisory speed instead of 5 mph below. For instance, the first run for Curve A was driven at 30 mph. The runs were conducted on a new day and a different location was used to level the device.

To further test the reliability of CurveAdvisor, two ramps off U.S. Route 460 were driven because they had advisory speeds higher than 30 mph and contained changes in vertical alignment. This would help determine if CurveAdvisor is accurate for a wide variety of curves. The ramps were named as seen in Figure 4.2.

Figure 4.2. View of Ramps 1 and 2. These curves were used to further test CurveAdvisor.

Each ramp was first driven at the advisory speed, 40 mph. The next run was conducted depending on the suggestion given by CurveAdvisor.

4.3 RESULTS

Once all runs were complete, the screenshots and videos for each run were viewed and the values were recorded. Figure 4.3 displays a screenshot for Curve 01 at 25 mph.

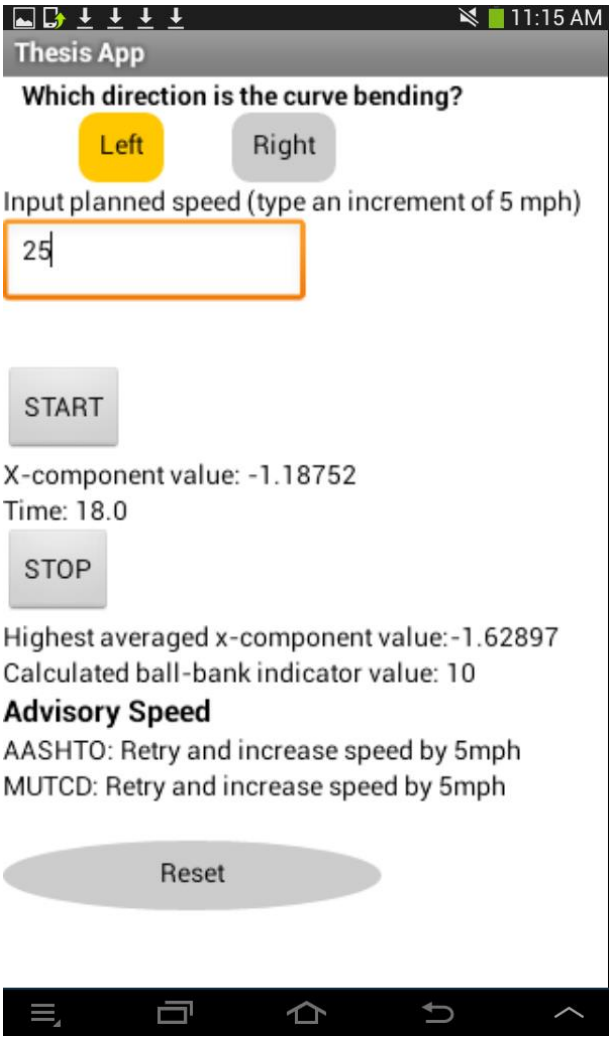


Figure 4.3. Screenshot of CurveAdvisor for Curve 01 at 25 mph.

After driving at 25 mph, CurveAdvisor suggested the next run be driven at a speed of 30 mph. The next run was conducted and then a new screenshot was taken which is shown in Figure 4.4.

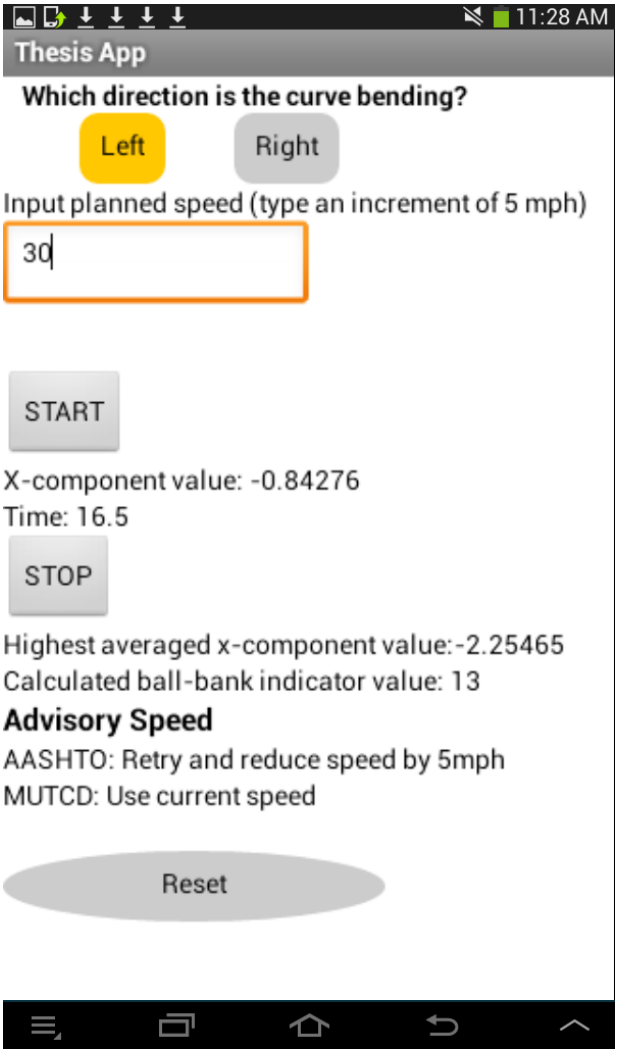


Figure 4.4. Screenshot of CurveAdvisor for Curve 01 at 30 mph.

According to these screenshots, AASHTO would recommend an advisory speed of 25 mph while MUTCD would recommend 30 mph. In some cases, MUTCD advised that the speed be increased by 5 mph but the next run stated “Retry and reduce speed by 5 mph.” This indicated that the previous speed should be used as the advisory speed. This procedure was executed for each curve and Tables 4.1, 4.2, and 4.3 include all the results. The Recorded BBI column is the reading from the videos.

Table 4.1: Results of CurveAdvisor Testing on Curves 01, 02, and 04

Curve Description	Curve Advisory Speed	Driven speed	Recorded BBI	CurveAdvisor Max X-component	CurveAdvisor BBI	AASHTO	MUTCD
Curve 01 (Harding Ave)	25	20	4	-0.64028	4	Increase by 5 mph	Increase by 5 mph
		25	9	-1.62897	10	Increase by 5 mph	Increase by 5 mph
		30	14	-2.25465	13	Reduce by 5 mph	Use current speed
Curve 02 (Harding Ave)	25	20	-6	0.78007	-5	Increase by 5 mph	Increase by 5 mph
		25	-7	1.08172	-6	Increase by 5 mph	Increase by 5 mph
		30	-11	1.74572	-10	Use current speed	Increase by 5 mph
		35	-16	2.57041	-15	Reduce by 5 mph	Reduce by 5 mph
Curve 04 (Happy Hollow)	20	15	8	-1.37384	8	Increase by 5 mph	Increase by 5 mph
		20	11	-1.90806	11	Increase by 5 mph	Increase by 5 mph
		25	17	-3.0402	18	Reduce by 5 mph	Reduce by 5 mph

Table 4.2: Results of CurveAdvisor Testing on Curves A and B

Curve Description	Curve Advisory Speed	Driven speed	Recorded BBI	CurveAdvisor Max X-component	CurveAdvisor BBI	AASHTO	MUTCD
Curve A (Cedar Run Rd)	30	30	-8	1.36407	-8	Increase by 5 mph	Increase by 5 mph
		35	-11	1.78825	-11	Reduce by 5 mph	Increase by 5 mph
		40	-15	2.48815	-15	Reduce by 5 mph	Reduce by 5 mph
Curve B (Cedar Run Rd)	25	25	6	-0.9246	5	Increase by 5 mph	Increase by 5 mph
		30	10	-1.47392	9	Increase by 5 mph	Increase by 5 mph
		35	14	-2.38052	14	Reduce by 5 mph	Reduce by 5 mph

Table 4.3: Results of CurveAdvisor Testing on Ramps 1 and 2

Curve Description	Curve Advisory Speed	Driven speed	Recorded BBI	CurveAdvisor Max X-component	CurveAdvisor BBI	AASHTO	MUTCD
Ramp 1	40	40	-5	0.8683	-5	Increase by 5 mph	Increase by 5 mph
		45	-8	1.24882	-7	Increase by 5 mph	Increase by 5 mph
		50	-10	1.72747	-10	Reduce by 5 mph	Increase by 5 mph
		55	-13	2.00292	-12	Reduce by 5 mph	Increase by 5 mph
		60	-13	2.04369	-12	Reduce by 5 mph	Reduce by 5 mph
Ramp 2	40	40	6	-1.26414	7	Increase by 5 mph	Increase by 5 mph
		45	9	-1.2988	8	Increase by 5 mph	Increase by 5 mph
		50	12	-1.98815	12	Reduce by 5 mph	Increase by 5 mph
		55	14	-2.25283	13	Reduce by 5 mph	Reduce by 5 mph

Based on these results, the corresponding advisory speeds for each curve were determined and are shown in Table 4.4.

Table 4.4: Advisory Speeds for Each Curve According to CurveAdvisor

	AASHTO	MUTCD
Curve 01	25 mph	30 mph
Curve 02	30 mph	30 mph
Curve 04	20 mph	20 mph
Curve A	30 mph	35 mph
Curve B	30 mph	30 mph
Ramp 1	45 mph	55 mph
Ramp 2	45 mph	50 mph

Tables 4.1, 4.2, and 4.3 show that CurveAdvisor is very accurate in determining the BBI values and therefore the appropriate advisory speed. When the CurveAdvisor

BBI did not agree with the recorded BBI, it was only off by 1 unit. Runs on Curve 02 best displayed this slight variance. All runs on Curve 02 produced BBI values one unit below the actual reading. This could be due to many factors – one being the curve's characteristics. Curve 02 is a right bending curve and the most horizontal of all the curves (very little vertical change). However, when analyzing Curve A, another right bending curve, the values matched exactly. When Curve A and Curve B were driven, the test was conducted on a different day and a different location was used to level the device. Due to this change in equipment setup, it is more likely that the Android device was not completely level before data collection on Curves 01, 02, and 04. This would adequately explain why Curve 02 produced values lower than normal. An uneven device will shift the x-component values slightly and consequently alter the BBI calculations. Furthermore, Curve B, a left bending curve, had more error than Curve A. This proves that leveling the device is extremely important. It is slightly challenging to level it exactly at zero since the reading from CurveAdvisor fluctuates so greatly when stationary. This problem can be addressed by using higher quality leveling techniques.

Another explanation for the slight variance in BBI readings could be attributed to the generated equation, y = -5.9029x. Since the y-intercept was set to zero when determining the relationship, this shifted the trendline down from the original data set and therefore may produce lower BBI values than the actual ones observed when driving. It seems that the BBI values are most accurate when they are lower in value and therefore closer to the origin of the graph in Figure 2.14.

CHAPTER 5

CONCLUSIONS

The creation of CurveAdvisor, an Android Application, takes advantage of emerging technology. Through its use, anyone can determine the appropriate advisory speed on a horizontal curve. CurveAdvisor provides a quick and easy method as opposed to those currently in use. It is also readily accessible and can be downloaded onto any Android device. To evaluate the effectiveness of this tool, several field tests were conducted. The evaluation resulted in many successful outcomes, presented a few instances that were not ideal, and revealed that there is room for improvement.

5.1 CONTRIBUTIONS

There are two main contributions of this work. The first is the relationship between an accelerometer's x-component and the BBI reading. Data collection on several curves demonstrated that both the largest x-component value and highest BBI reading occur at the curve's sharpest point. This indicates a correlation between the two. The equation y = -5.9029x represents this relationship. This equation illustrates that the two parameters have a linear relationship.

The second contribution is CurveAdvisor itself. Most of the different methods mentioned in Chapter 1 require certain equipment in order to carry out the process of advisory speed assignment. CurveAdvisor eliminates the need for such equipment and allows the user to readily and easily determine the appropriate advisory speed for a curve. The user interface is simple and outputs immediate feedback once the Application is terminated. Only one person is needed to operate the device. However, there must also be a driver to navigate the curves. CurveAdvisor is a quick and easy tool that can accurately assign the appropriate advisory speed.

5.2 FUTURE WORK

CurveAdvisor is a solid foundation for meeting the goal of delegating advisory speeds on curves. While designing this Application, it was evident that there is potential for enhancement. The following paragraphs highlight suggested improvements to CurveAdvisor.

CurveAdvisor only considers one of the three components of the accelerometer – the horizontal component. To advance this tool, the y-component and z-component should also be evaluated. This can be done by driving a much broader range of curves. Curves with varying characteristics and locations should be targeted. All the collected values for each component should be analyzed separately and then together to see

what effect they may have on BBI results. Through this process, a multi-variable equation can be generated that would replace the y= - 5.9029x equation found in Chapter 2. This will incorporate all angles of the curve which will increase accuracy when outputting the appropriate advisory speed.

CurveAdvisor does not consider vehicle characteristics. A Mazda 3 was used for this research so all results will be based on a compact vehicle. To better represent the correct advisory speed, numerous vehicle characteristics should be incorporated such as weight, engine force, and aerodynamics. The interface should request an input where the user indicates what type of vehicle is being driven for the test runs. Depending on the vehicle type, there will be slight variance in the centrifugal force and vehicle body roll – two key components for the BBI and accelerometer. The accuracy of CurveAdvisor will increase if vehicle characteristics are integrated in the coding.

The curves used for data collection in Chapter 2 were low speed curves, ranging from 20 mph to 25 mph. In order to adequately represent a greater array of curves, data should also be collected from curves with higher advisory speeds. Generally these curves have a large radius and can therefore be traversed at a higher speed. By including curves that have a larger radius, CurveAdvisor can incorporate a greater range of radii. Subsequently, CurveAdvisor will more accurately represent the BBI and x-component relationship and therefore the appropriate advisory speed.

During data analysis in Chapter 2, the first set of averaged x-components was initialized at zero. CurveAdvisor should do the same so that the output BBI reading matches more closely to the BBI readings determined during analysis. This will ensure that the accelerometer is reading a value of zero to begin which essentially self-levels the device at the beginning of the curve.

To make CurveAdvisor even easier to use, a speedometer function should be included in the coding. This would eliminate the need for a user input and therefore decrease the amount of work required from the user. Removing the user input means that the only actions the operator needs to complete are highlighting the direction of the curve and touching the Start button. This would make the Application more user friendly. Additionally, the inclusion of a speedometer would allow CurveAdvisor to read each specific speed throughout the entire drive instead of generalizing it as what the user types into the input box. This would greatly increase the accuracy of the results.

REFERENCES

1. Bonneson, J., et al., *Development of Guidelines for Establishing Effective Curve Advisory Speeds*. 2007. p. 1-1.
2. *Manual on Uniform Traffic Control Devices*. 2009: Federal Highway Administration.
3. Chowdhury, M.A., et al., *Are the criteria for setting advisory speeds on curves still relevant?* Institute of Transportation Engineers. ITE Journal, 1998. **68**(2): p. 32.
4. R. Milstead, X.Q., B. Katz, J. Bonneson, M. Pratt, J. Miles, and P. Carlson, *Procedures for Setting Advisory Speeds on Curves.* 2011: p. 2.
5. Avelar, R.E. and K.K. Dixon, *How Far from Optimal Are Current Advisory Speeds? Analysis Based on Safety Performance.* Transportation Research Record, 2012(2280): p. 183-191.
6. Bonneson, J.A., et al., *Horizontal Curve Signing Handbook*. 2007. p. 56.
7. *A Policy on Geometric Design of Highways and Streets*. 5th ed. 2004, Washington, D.C.: American Association of State Highway and Transportation Officials (AASHTO).
8. Bonneson, J., et al., *Identifying and Testing Effective Advisory Speed Setting Procedures*. 2007.
9. Smith, D. and J. McIntyre, *Handbook of Simplified Practice for Traffic Studies.* 2002: p. 2.2-2.3.
10. Massachusetts Institute of Technology. *App Inventor*. 2012; Available from: http://appinventor.mit.edu/explore/library.html.

APPENDIX A
Supplementary Figures and Tables

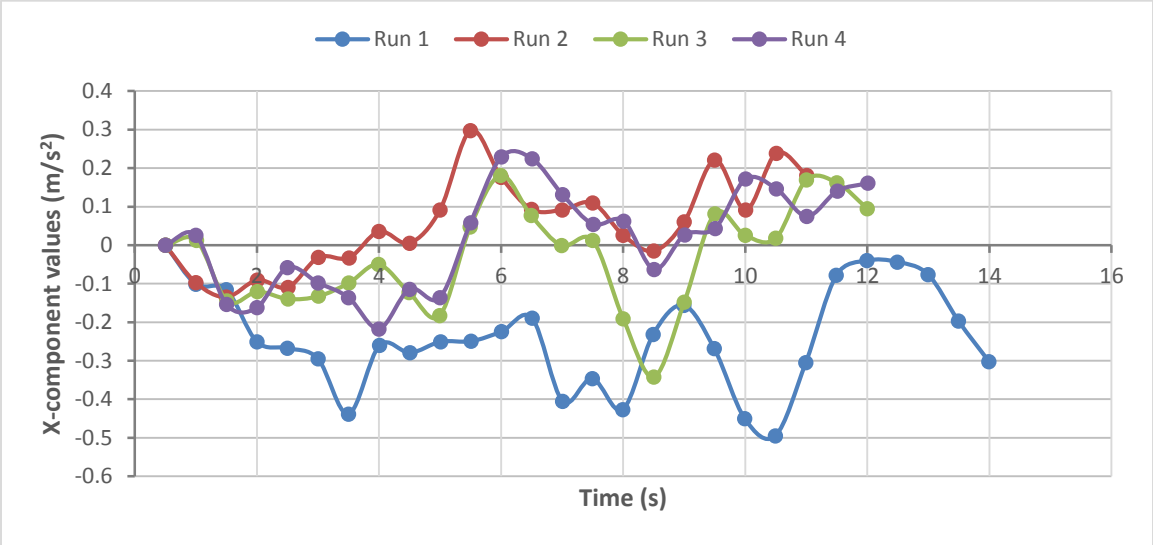


Figure A.2.1. Final visual representation of all runs for Curve 01 at 20 mph.

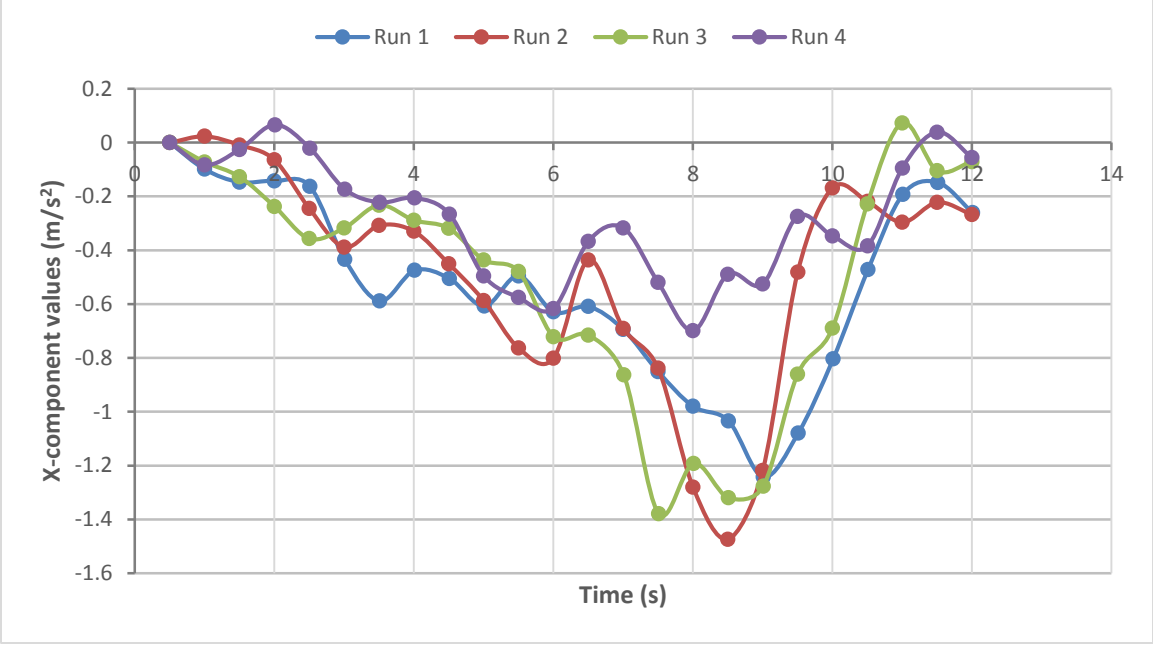


Figure A.2.2. Final visual representation of all runs for Curve 01 at 25 mph.

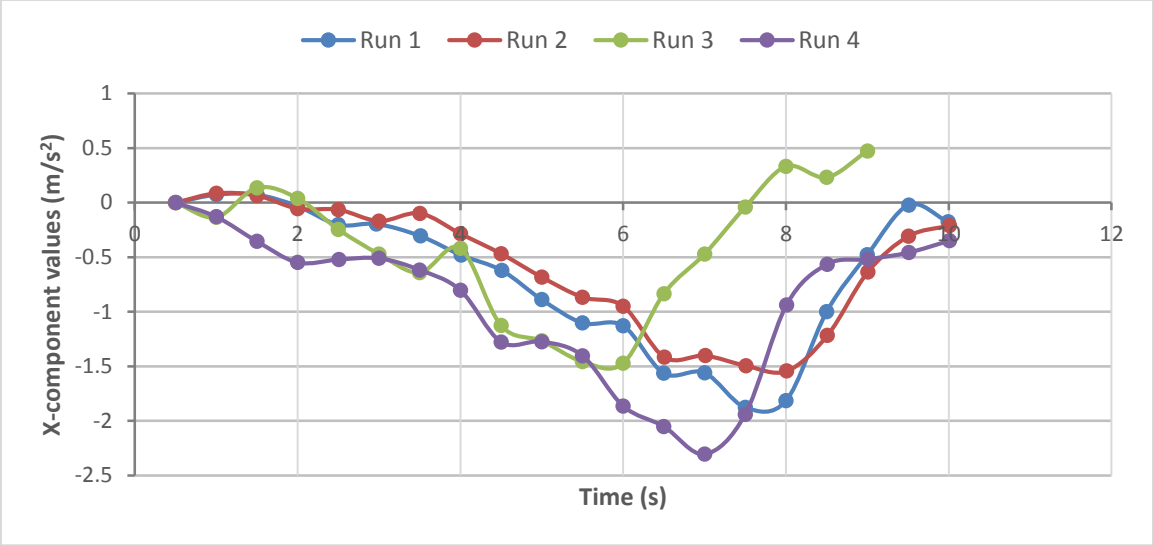


Figure A.2.3. Final visual representation of all runs for Curve 01 at 30 mph.

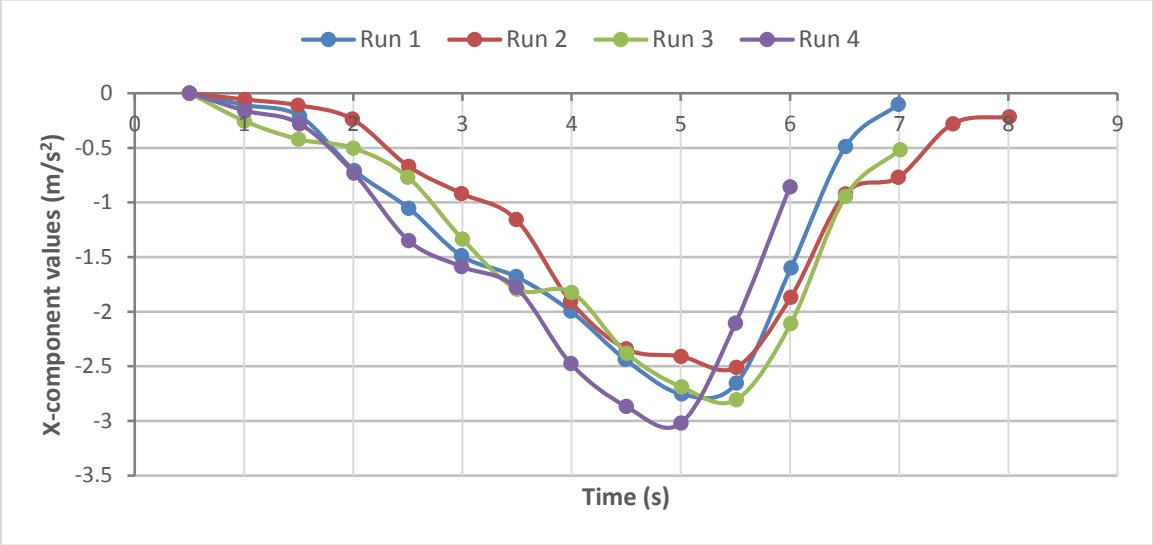


Figure A.2.4. Final visual representation of all runs for Curve 01 at 35 mph.

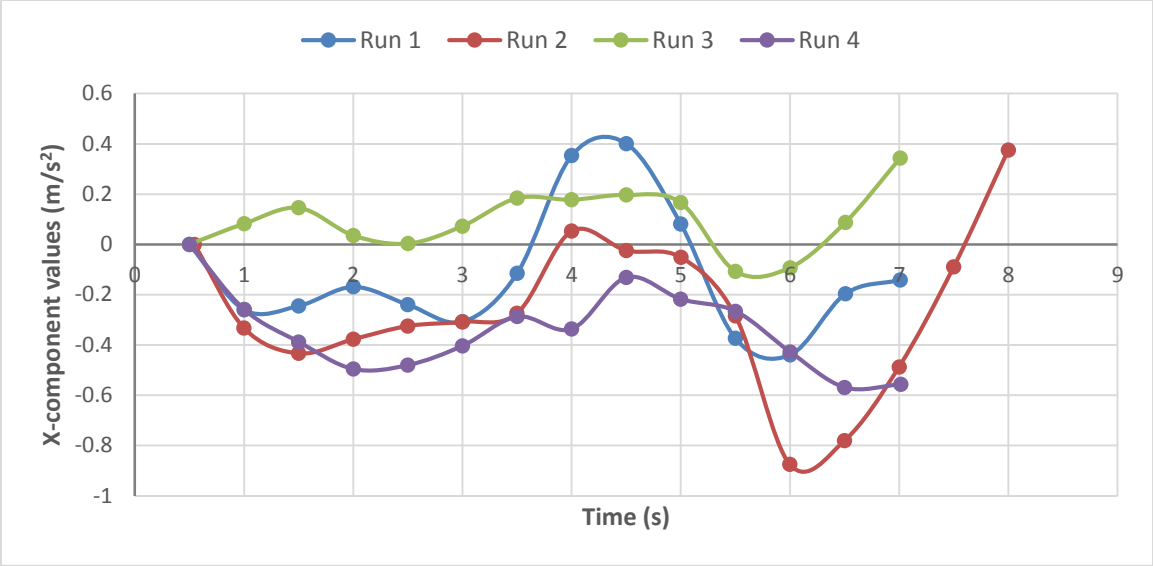


Figure A.2.5. Final visual representation of all runs for Curve 03 at 35 mph.

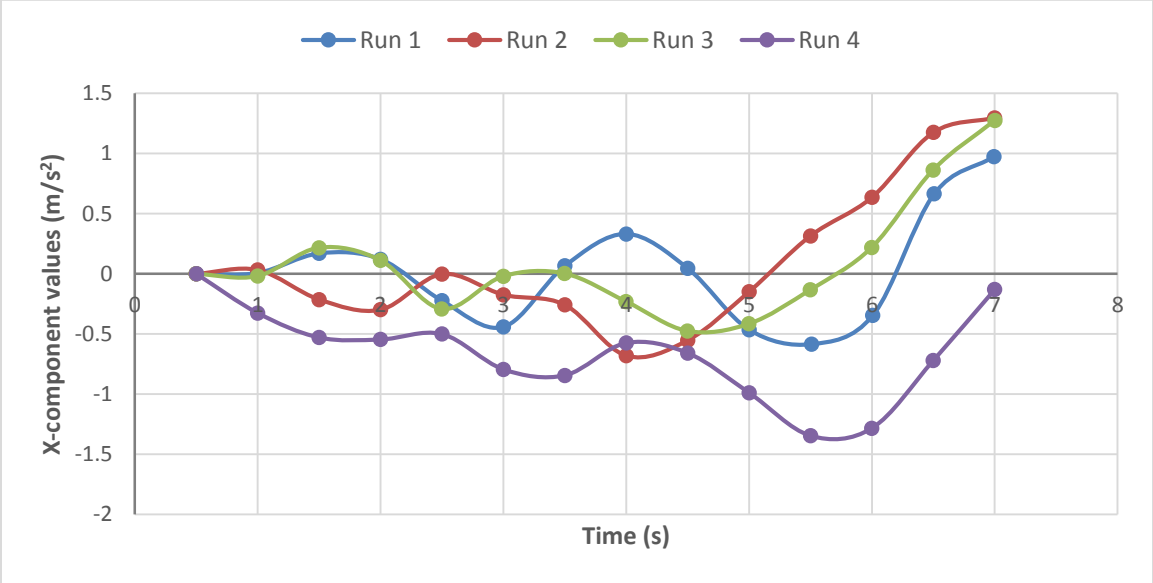


Figure A.2.6. Final visual representation of all runs for Curve 03 at 40 mph.

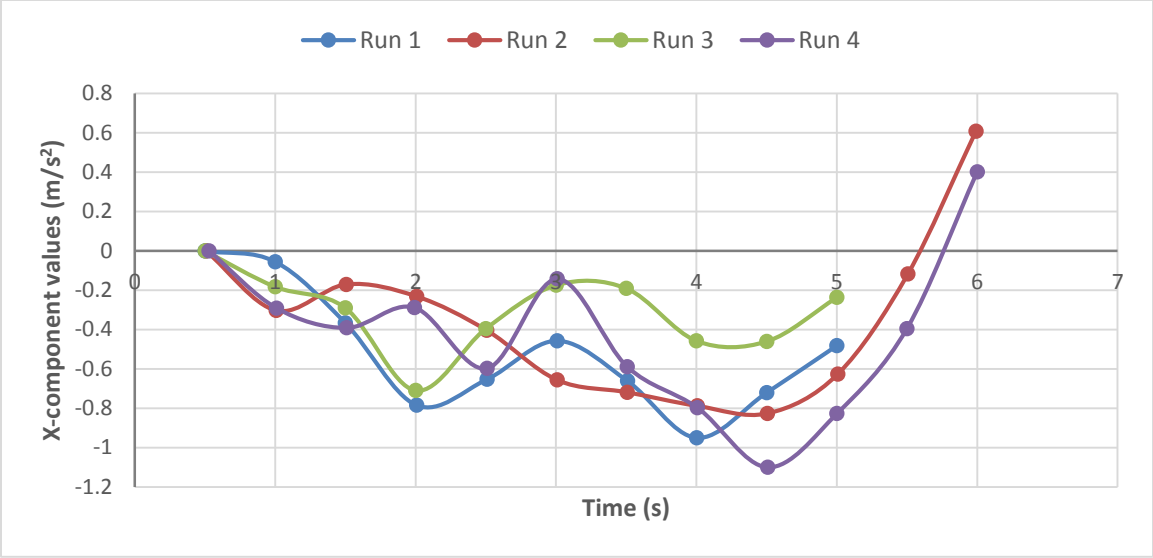


Figure A.2.7. Final visual representation of all runs for Curve 03 at 45 mph.

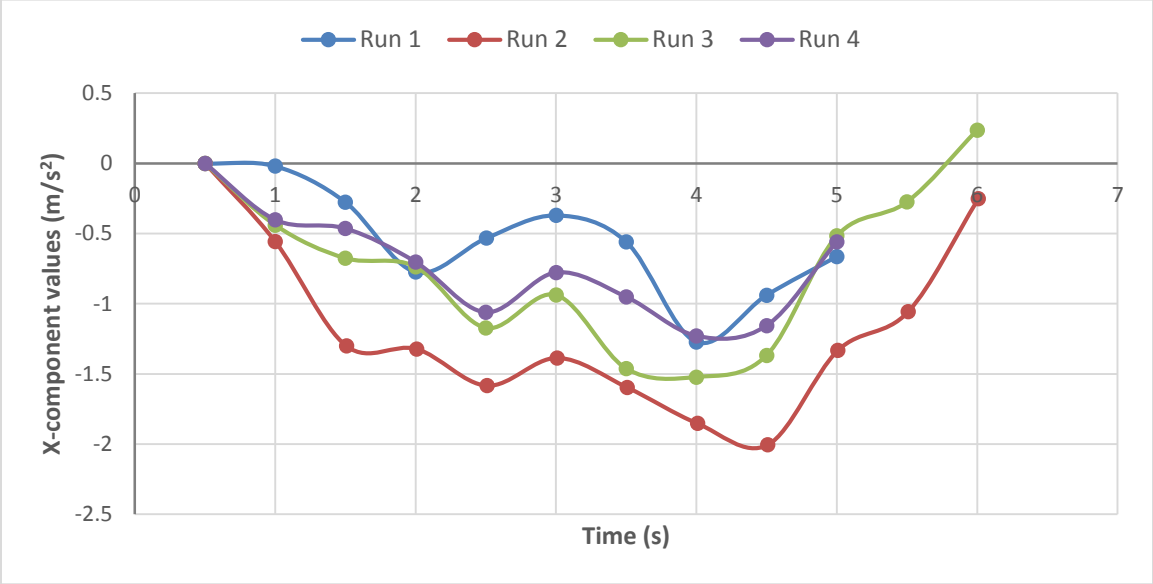


Figure A.2.8. Final visual representation of all runs for Curve 03 at 50 mph.

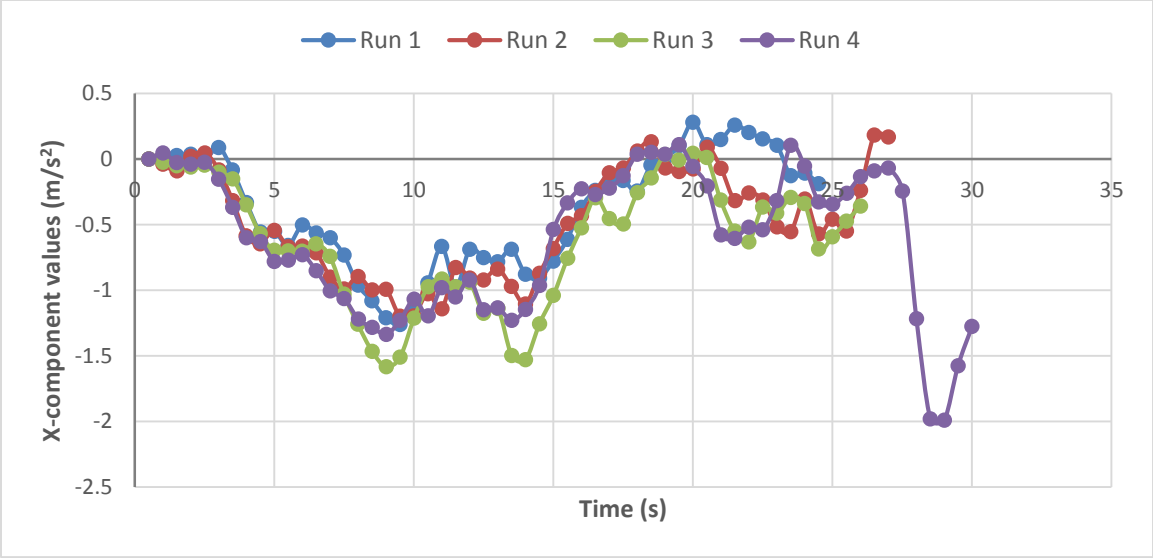


Figure A.2.9. Final visual representation of all runs for Curve 04 at 15 mph.

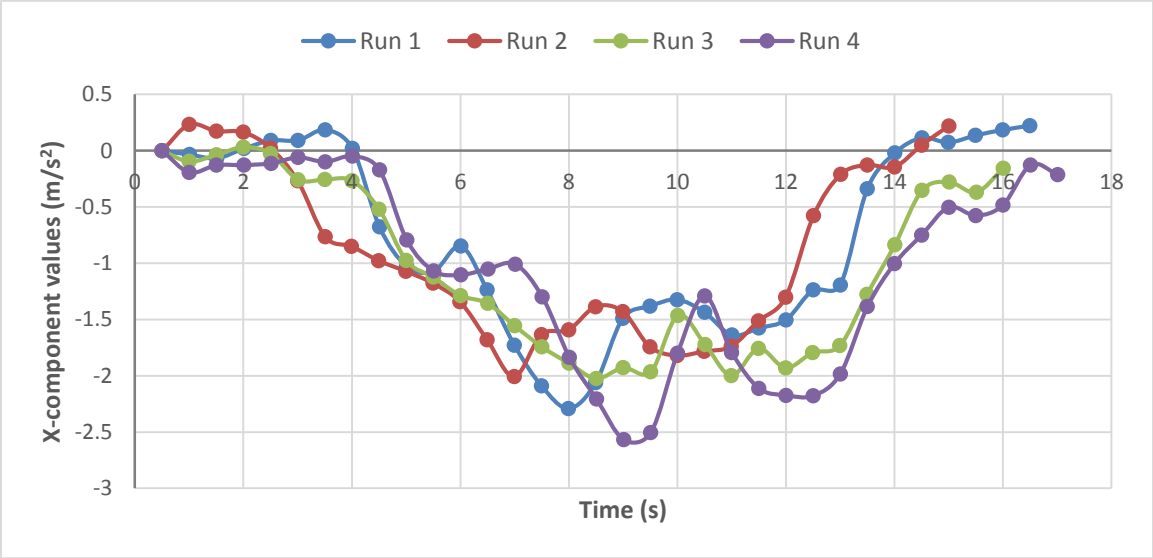


Figure A.2.10. Final visual representation of all runs for Curve 04 at 20 mph.

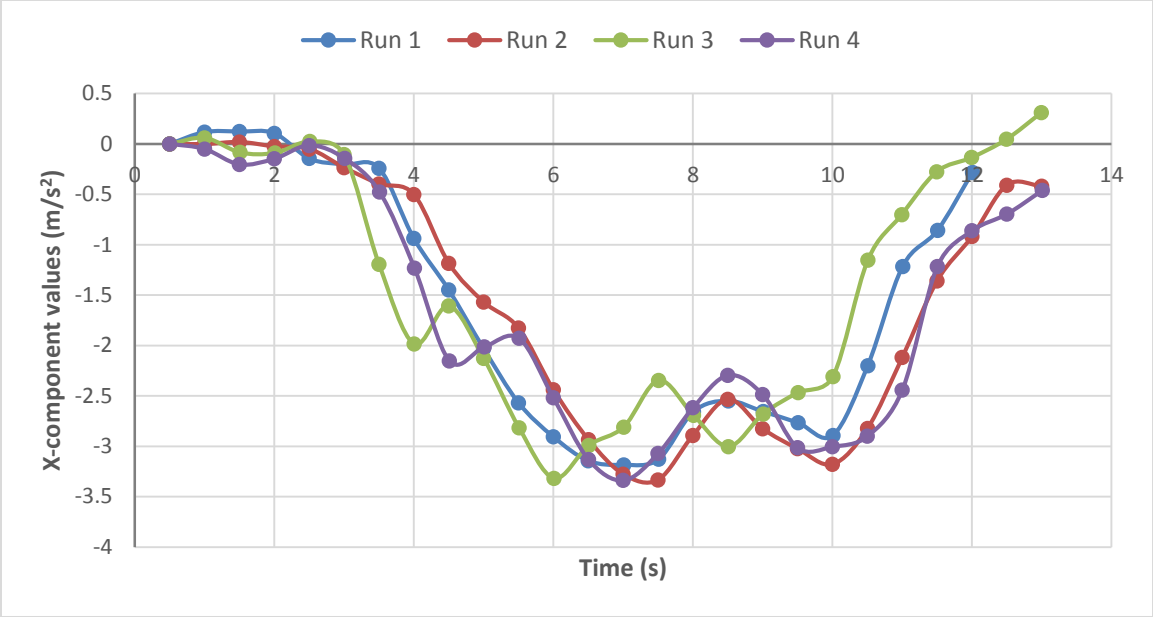


Figure A.2.11. Final visual representation of all runs for Curve 04 at 25 mph.

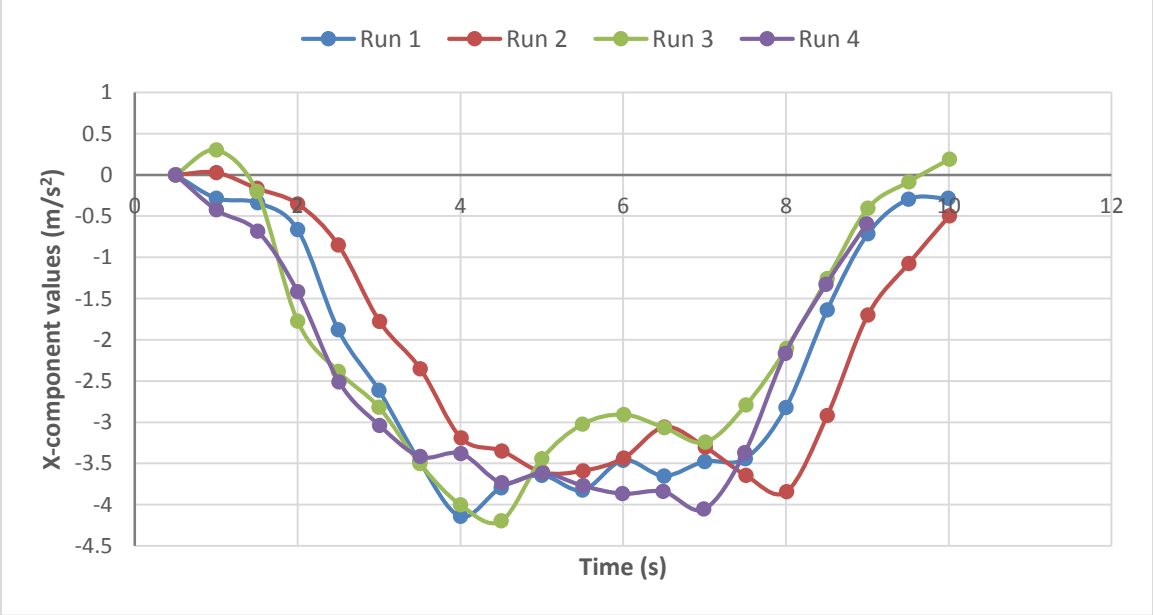


Figure A.2.12. Final visual representation of all runs for Curve 04 at 30 mph.

Table A.2.1: X-components and Corresponding BBI Values for Curve 01 and Curve 03

Curve Description	Curve Advisory Speed	Driven speed	Max X-component	Ball Bank Indicator
Curve 01 (Harding Avenue)	25	20	-0.4951	5
			-0.1348	4
			-0.3421	5
			-0.2175	4
		25	-1.2419	9
			-1.474	11
			-1.3792	10
			-0.6988	8
		30	-1.8765	14
			-1.5428	13
			-1.4706	14
			-2.3054	15
		35	-2.7546	19
			-2.5118	18
			-2.8057	18
			-3.0205	19
Curve 03 (Harding Avenue)	40	35	-0.4391	7
			-0.8750	8
			-0.1070	6
			-0.5565	5
		40	-0.5848	10
			-0.6799	10
			-0.4748	9
			-0.5848	11
		45	-0.9502	12
			-0.8260	11
			-0.7100	12
			-1.0996	12
		50	-1.2719	14
			-2.0047	13
			-1.5223	14
			-1.2292	13

Table A.2.2: X-components and Corresponding BBI Values for Curve 04

Curve Description	Curve Advisory Speed	Driven speed	Max X-component	Ball Bank Indicator
Curve 04 (Happy Hollow)	20	15	-1.2595	7
			-1.1965	7
			-1.5829	8
			-1.3349	7
		20	-2.2904	14
			-2.0079	12
			-2.0237	11
			-2.5681	14
		25	-3.1844	18
			-3.3335	18
			-3.3183	18
			-3.3382	18
		30	-4.1438	20+
			-3.8422	20+
			-4.1986	20+
			-4.0541	20+

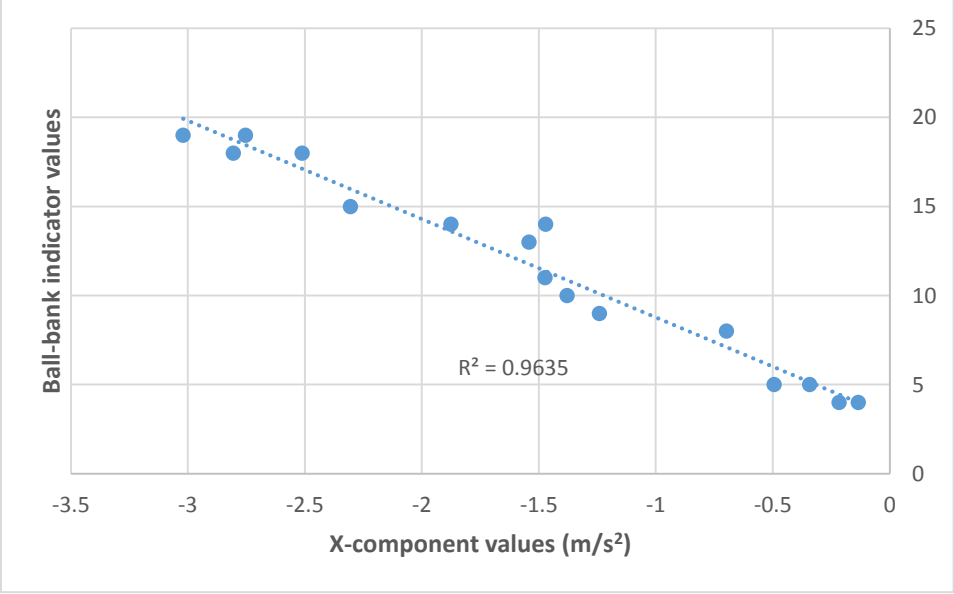


Figure A.2.13. Relationship between the x-component values and their corresponding BBI values for Curve 01.

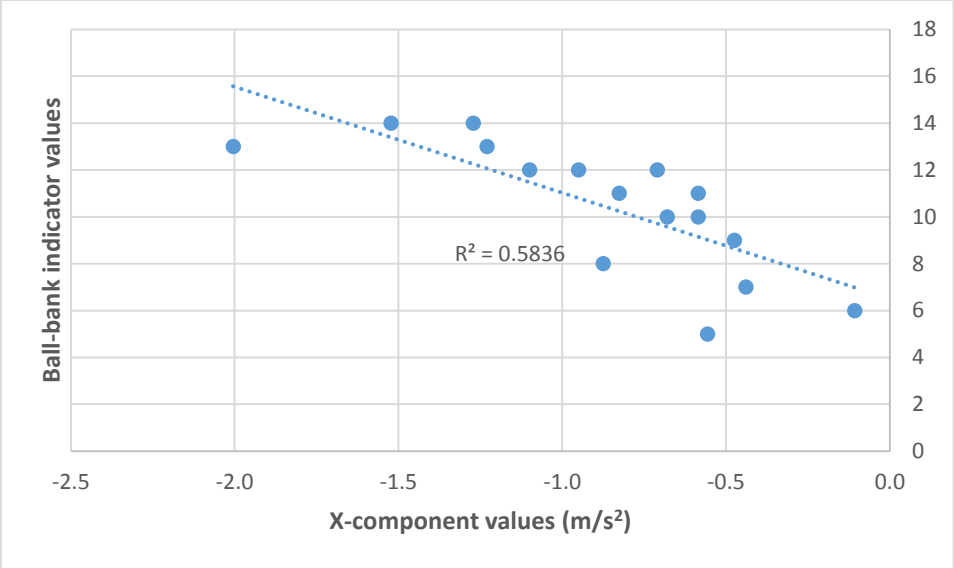


Figure A.2.14. Relationship between the x-component values and their corresponding BBI values for Curve 03.

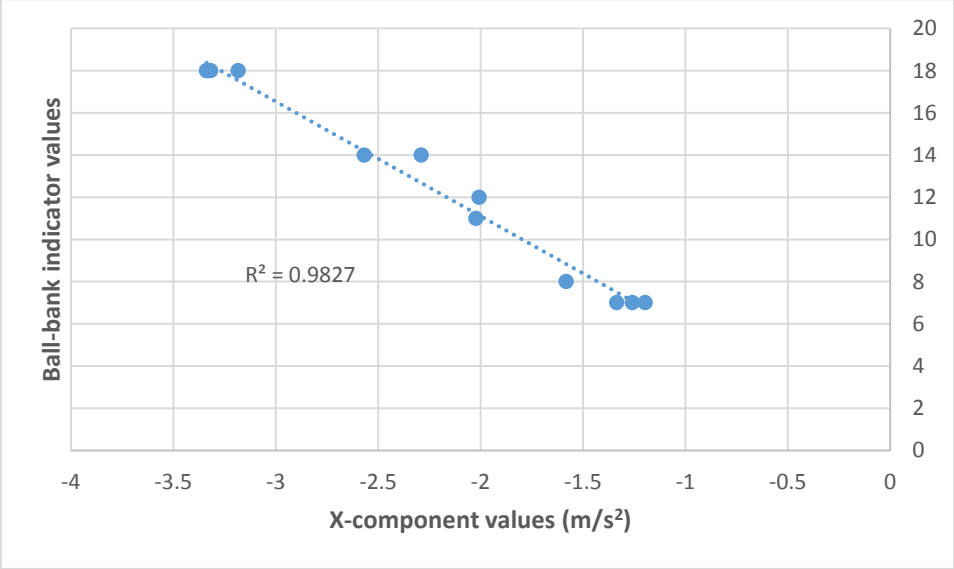


Figure A.2.15. Relationship between the x-component values and their corresponding BBI values for Curve 04. Values for 30 mph were excluded due to extreme BBI readings.

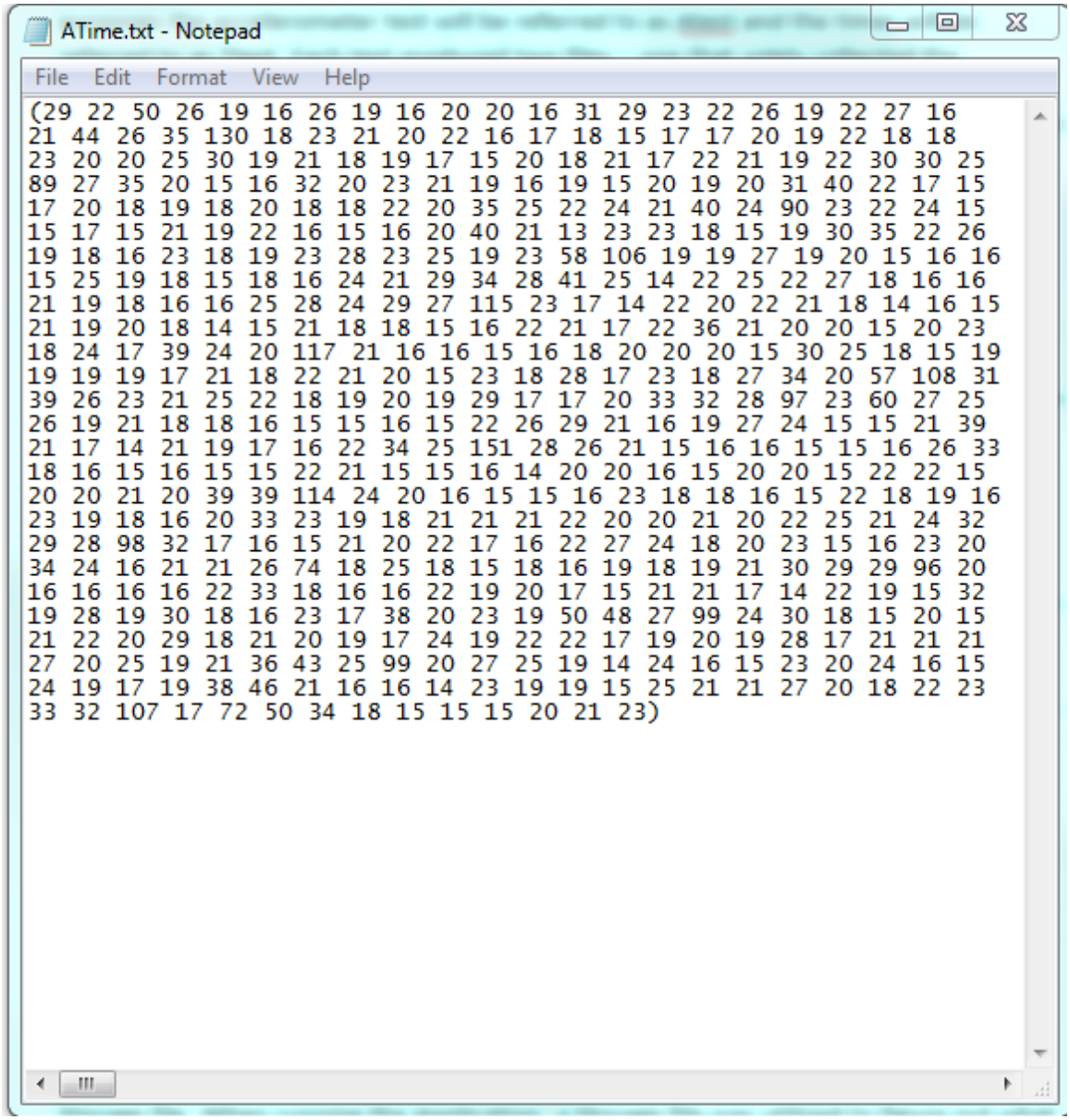


Figure A.3.1. File produced through the Atest to determine the mean time interval.

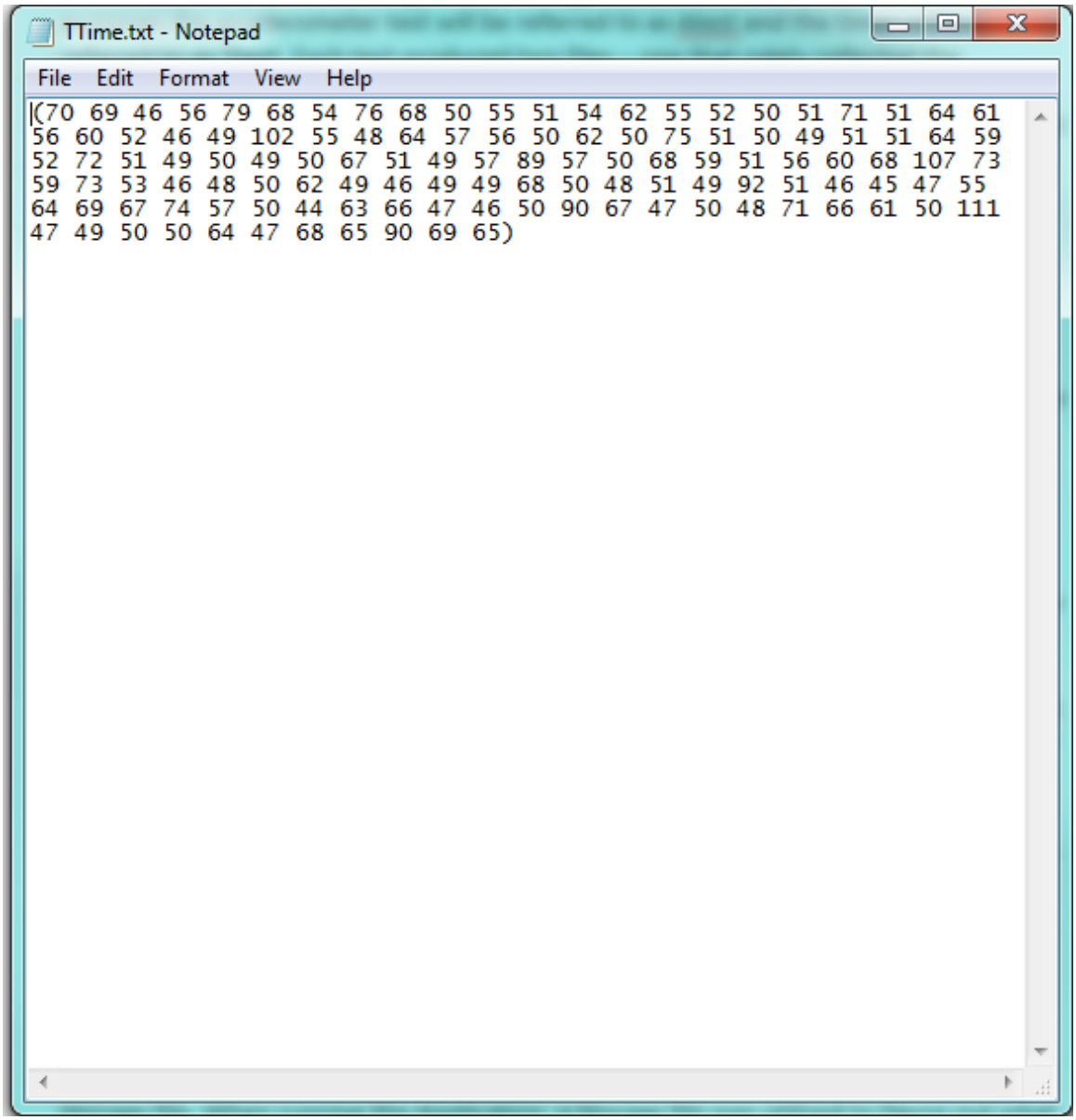


Figure A.3.2. File produced through the Ttest to determine the mean time interval.

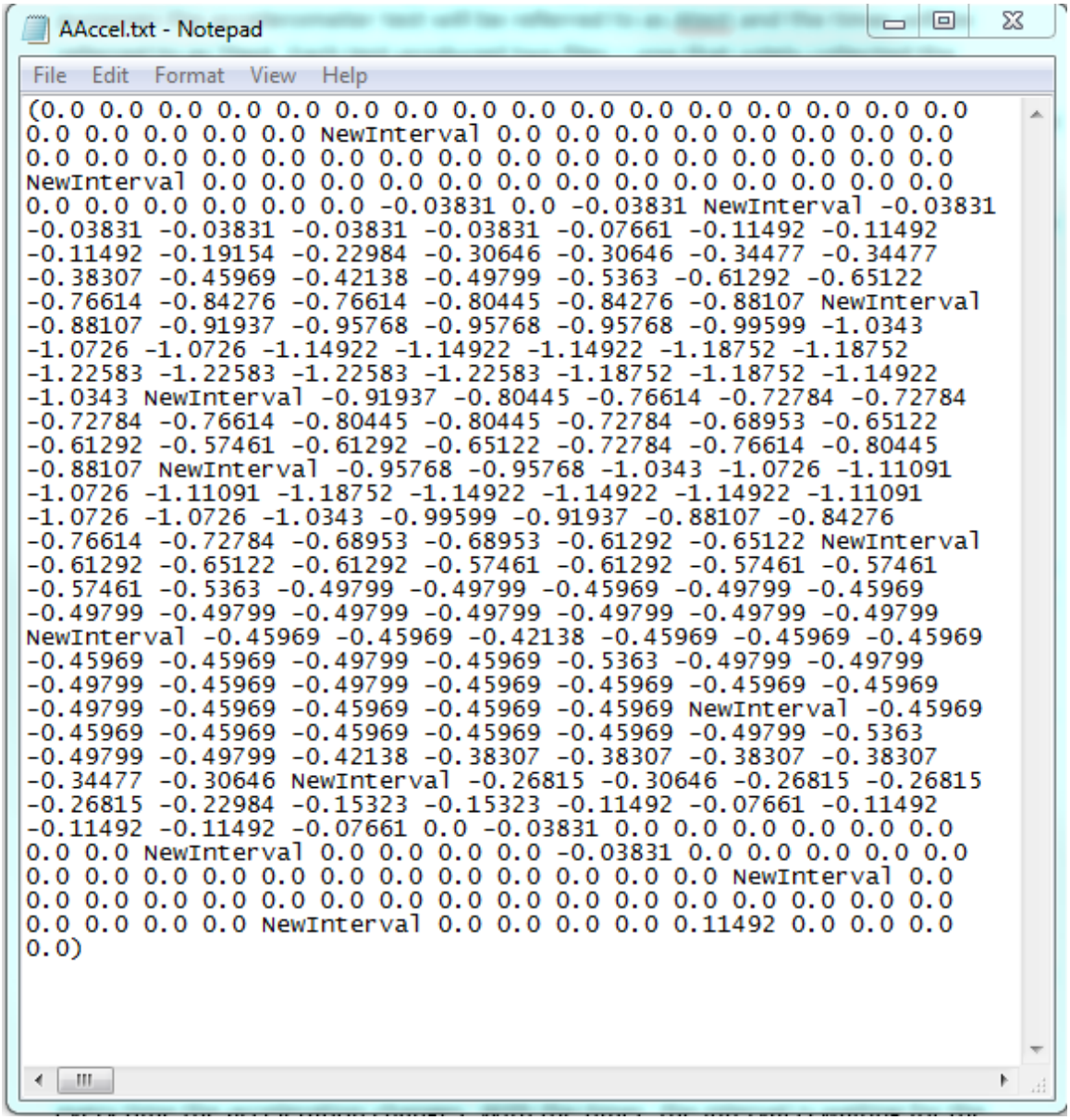


Figure A.3.3. File produced through the Atest to determine the number of acceleration values collected in each time interval.

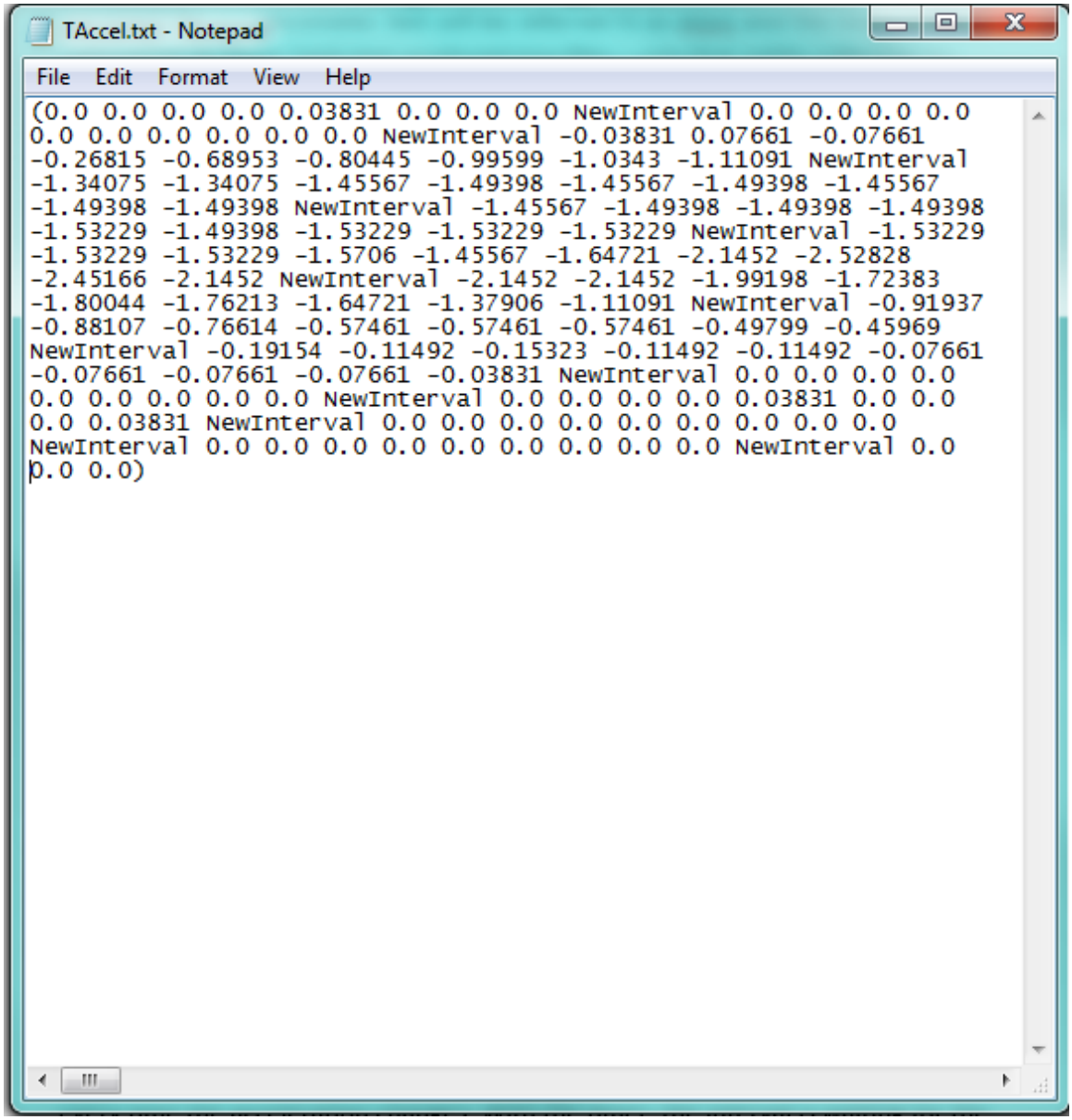


Figure A.3.4. File produced through the Ttest to determine the number of acceleration values collected in each time interval.