DESIGN OF A VEHICLE AUTOMATIC EMERGENCY PULLOVER SYSTEM FOR AUTOMATED DRIVING WITH IMPLEMENTATION ON A SIMULATOR

A Thesis

Submitted to the Faculty

of

Purdue University

by

Wasif Javaid

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

May 2017

Purdue University

Indianapolis, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. Yaobin Chen, Chair

Department of Electrical and Computer Engineering

Dr. Stanley Chien

Department of Electrical and Computer Engineering

Dr. Renran Tian

Department of Electrical and Computer Engineering

Approved by:

Dr. Brian King

Head of the Departmental Graduate Program

To my Parents: Muhammad Javaid and Fakhra Shaheen.

ACKNOWLEDGMENTS

I would like to be thankful for the uncountable help and guidance provided by Dr. Yaobin Chen, Chancellor's Professor of Electrical and Computer Engineering Department and Director of Transportation Active Safety Institute (TASI). Dr. Yaobin Chen has been motivational and supportive in providing me the resources for the completion of this task. Dr. Stanley Chien and Dr. Lingxi Li have been helpful in this effort. I would also like to thank Dr. Stanley Chien and Dr. Renran Tian who are my committee members in my final defense.

I would like to acknowledge the contribution of Mr. Michael DeForge from Real-time Technologies, Inc. especially because of helpful behavior regarding the questions in the simulator operation.

This research was partially sponsored by the U.S. Department of Transportation through its University Transportation Centers program (Crash Imminent Safety UTC), the Transportation Active Safety Institute and the Department of Electrical and Computer Engineering.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

ABBREVIATIONS

RTI	Realtime Technologies, Inc.
ISA	Internet Scene Assembler
LDW	Lane Departure Warning
LKA	Lane Keeping Assist
FCW	Forward Collision Warning
CIB	Collision Imminent Breaking
AEB	Automatic Emergency Breaking
BSM	Blind Spot Monitoring
ACC	Adaptive Cruise Control
PAEB	Pedestrian Automatic Emergency Breaking
ALC	Automatic Lane Change
AEP	Automatic Emergency Pull-over
PID	Proportional Integral Derivative
CG	Center of Gravity
SV	Subject Vehicle
mph	miles per hour
RADAR	Radio Detection and Ranging

NOMENCLATURE

HeadingError	The error with the desired straight direction of motion for the subject vehicle.
LaneOffset	The value for the deviation of subject vehicle's CG with the center of the lane.
HDistance	The distance (in meters) of subject vehicle with the vehicle directly ahead of it.
HTime	The time required (in seconds) for SV to reach the location of the vehicle directly ahead of it.
PedDistance	The distance from the CG of subject vehicle to the CG of the actor/pedestrian in the scenario.
PedBearing	The angle CG of pedestrian makes with the CG of the subject vehicle (directly straight in the direction of motion of the SV is 0 degrees).
PedWarning	The audio warning on the detection of the nearby pedestrian.
VehVelocity	The velocity for the subject vehicle.
Trigger	The health device trigger to the electronic system of the Vehicle, it could be the trigger from a heart monitoring device (i.e. Implantable Cardioverter Defibrillator).
ObjDistance0	The distance (meters) of subject vehicle with the first roadside object (right side) in the typical scenario w.r.t. scenario starting point.
ObjDistance1	The distance (meters) of subject vehicle with the second roadside object (left side) in the scenario w.r.t. scenario starting point.

ObjDistance2	The distance (meters) of subject vehicle with the third roadside object (right side) in the scenario w.r.t. scenario starting point.
ObjDistance3	The distance (meters) of subject vehicle with the fourth roadside object (left side) in the scenario w.r.t. scenario starting point.
MPHVelocity	Subject vehicle velocity in miles per hour.

GLOSSARY

AltiaFace	An add-on for the SimCreator software used to create the dashboard and similar visual objects by using the items from its library.
SimCreator	The main software for the model development with the help of blocks from its standard library and the designing of model blocks itself using the programming with C++.
SimVista	The main software for the scenario development using the tiles from it's library, it also makes the scenario run with the help of sensors in its library which can be programmed with JavaScript.
GroundSim	The GroundSim component handles I/O from the instrumented (cab) vehicle, tires, dynamics and power train models.
VirtualDash	The Virtual Dash is the LCD for the instrument cluster. This runs Altia and the display is shared with the host computer.
Memory	This is a component in the software 'SimCreator' which acts like a storage for some data and to give the output on certain prescribed conditions. It uses C++ as programming language and can be edited.
Switch	This is a component in the software 'SimCreator' which acts like a two way switch. It can be edited for certain uses as per the requirement.
Controller	The part of a system which is added to correct the operation of a plant, usually by using the feedback techniques.

Feedback	The technique of taking a portion of output of a system and applying it to the input in some combination with the input signal- usually subtraction. If the output is modified by the transfer function before being applied back to the input, the feedback is said to be non-unity feedback.
PID	Proportional, Integral, Derivative; A control system where the error signal is formed as a linear combination of the input, the output, the derivative of the output, and the integral of the output. The design of such a system involves determining the appropriate value of the coefficients of combination - a process called tuning.
Plant	The part of a system which is to be controlled, having an input and an output. It is usually presented by a transfer function.
TransFunc	A mathematical representation of a system in the complex frequency domain as the ratio of two polynomials, G(s)=num(s)/den(s) satisfying the relationship Y(s)/R(s) = G(s) where, Y(s) and R(s) are the output and input of the system respectively. The 'm' roots of the numerator polynomial are referred to as zeros, where as the 'N' roots of the denominator polynomial are called poles. The roots may be simple, having only a real part, or be complex; having both a real part and an imaginary part.
Summer	A point of intersection for the multiple signals producing a combined output. It could be sum or difference of those inputs.
Vector	A mathematical construct composed of a magnitude and a direction, occurring in some n-space of $n \geq 2$. A vector may be represented by a magnitude and an angle, $\vec{x} = \sqrt{2}\angle 45°$, or by the lengths projected on the axis of the space, $\vec{x}$=[1,1].

ABSTRACT

Author: Javaid, Wasif. MSECE
Institution: Purdue University
Degree Received: May 2017
Title: Design of a Vehicle Automatic Emergency Pullover System for Automated Driving with Implementation on a Simulator
Major Professor: Dr. Yaobin Chen

This thesis addresses a critical issue of automotive safety. As traffic is increasing on the roads day by day, road safety is also a very important concern. Driving simulators can play an extensive role in the development and testing of advanced safety systems in peculiar traffic environments, respectively. Advanced Driver Assist Systems (ADAS) are getting enormous reputation but there is still need for more improvements.

This thesis presents a design of an Automatic Emergency Pullover (AEP) strategy using active safety systems for a semi-autonomous vehicle. The idea for this system is that a moving vehicle equipped with an AEP system can automatically pull over on the roadside safely when the driver is considered incapable of driving. Furthermore, AEP supporting features such as; Lane Keeping Assist, Blind Spot Monitoring, Vehicle and Pedestrian Automatic Emergency Braking, Adaptive Cruise Control are also included in this work. The designs for application of each system have been explained along with its algorithms, model development, component architecture, simulation results, vehicular/pedestrian behavior and trajectory precision on software tools provided by Realtime Technologies, Inc. All major variables which influence the performance of vehicle after AEP activation, have been observed and remodeled according to control algorithms. The implementation of AEP system which can control vehicle dynamics has been verified with the help of simulation results.

1. INTRODUCTION

1.1 Background and Motivation

The occurrence of traffic accidents is a very serious matter which points toward traffic safety. Around 1.24 million major or minor road traffic accidents occur annually on the world's roads, which makes road traffic injuries the eighth leading factor for death all over the world, mainly the prominent cause for death of young people aged between 15-29 years [1]. The decade of action for Road Safety 2011-2020, declared by the United Nations General Assembly, offers a broad structure for drawing attention towards the need for greater safety on the roads [1]. Active Safety has a lot of significance in avoiding such tragedy. Studies identify the importance as injury and fatality rates has accomplished new lows in 2009, with 75 injured and 1.14 people killed per 100 million miles of vehicle, compared to 120 injured people and 1.55 fatalities 10 years ago (NHTSA's National Center for Statistics and Analysis, 2010).

These fatalities can be avoided with the help of smart vehicles on the roads. Various active safety features have been developed in the past few decades; namely, anti-lock braking, traction and vehicle stability control, lane keeping assist, blind spot monitoring, automatic emergency braking, automatic lane change, adaptive cruise control and many more. AEP is an autonomous driving idea which could be implemented on vehicle control integrated with an implantable health device, for safety of patient's worst case scenarios. The intelligent vehicle systems might improve driving experience overall and can play a vital role in making automobile transportation safer.

1.2 Literature Review

Lane Keeping Assist System

It can warn and assist when driver unintentionally leave the road lane or when the driver change lanes without indication. The system monitors the vehicle position in the road lane. The Lane Departure Warning (LDW) system was estimated to reduce the number of crashes by 26.1% and the number of seriously injured drivers by 20.7%. In contrast, the aggressive steering to light steering Lane Keeping Assist (LKA) system were estimated to reduce the number of crashes by 32.7% to 37.3% and the number of seriously injured drivers by 26.1% to 31.2% [2]. The LKA system with characteristics of autonomous driving were estimated to reduce the number of crashes by 51.0% and the number of serious injuries by 45.9% [2]. This study represent that LDW and LKA could mitigate a considerable proportion of injuries and crashes in lane change accidents [2].

Blind Spot Monitoring System

Blind Spot Monitoring (BSM) system alerts driver whenever the traffic vehicles are in their side blind zone areas in the adjacent right or left lane. According to Studies; Overall, drivers are not able to execute over the shoulder (blind spot) glances for 85% and 68% for the right and left lane changes, respectively [3]. This recommends that BSM display provides indication to the driver that mostly fails to be obtained via over the shoulder glances [3]. In addition, when BSM was enabled there was a 23% reduction in right lane changes attempted without the driver checking the inside mirror and 31% reduction in left lane changes attempted without the driver checking the left mirror [3].

Forward Collision Warning/Automatic Emergency Braking System

It can detects at an early stage the danger of a potential collision with the vehicle in front and when there is no reaction from the driver observed it takes the preventive measures to mitigate the crash itself. Comparative with a no-warning condition, an early collision warning reduced the count of collisions by 80.7% [4]. Assuming collision severity is proportional to kinetic energy, the in time warning reduced collision severity by 96.5%. In contrast, the late warning reduced collisions by 50.0% and the corresponding severity by 87.5% [4].

Pedestrian Automatic Emergency Braking System

Pedestrian Automatic Emergency Braking (PAEB) system also uses various types of on board sensors (such as mono/stereo camera, radar, lidar, infrared etc.) in order to detect the potential crash to pedestrians. PAEB warns the driver if there is an imminent collision detected and if the driver does not take braking action it mitigates collision severity by applying the brake automatically [5]. Statistics show that in USA 11% of all traffic fatalities are pedestrian traffic fatalities (4,700 people) [6]. From research it is observed that a 10% reduction of velocity before impact can reduce fatal injuries in car crashes with approximately 30% [7] and that lowering the impact speed from 50 to 25 km/h before pedestrian collisions reduces fatality risk by 85% [8].

Adaptive Cruise Control System

Adaptive cruise control (ACC) is an advancement to Conventional Cruise Control. It assists the driver in controlling the longitudinal motion of their vehicle when driving on motorways [9]. The ACC system may be disturbed in several different ways, such as: driver alertness, road surface conditions, visibility, route curvature, or lead car velocity [9]. Only systems responding to lead car velocities are commercially available, but other disturbance systems are concerned with research and development [9]. The

ACC system employs certain switching logic which moves between two fundamental modes of operation [10]. In the primary mode, ACC maintains a velocity much like traditional cruise control. When necessitated by the presence of a slower lead car, the ACC system changes to headway management mode [10]. In this mode, velocity is controlled to maintain the distance/time relationship with the lead vehicle at the distance/time set point [10]. The system controls the accelerator, vehicle brakes and engine power train to maintain a desired time-gap to the vehicle ahead and ameliorate the driving comfort and experience [11]. On average, 90% of the complete driving time both the systems were used. For cruise control, the percentages ranged from 99% on rural roads during car following sections to 87% on rural roads with fog [12]. For ACC, the percentages ranged from 99% on the motorways with secondary tasks to 95% on rural roads with fog [12]. Although, there were driver delayed reactions in some critical situations, e.g., in a narrow curve or a fog bank. These results draw attention towards the safety effects of these systems, especially if their range of functionality (e.g., ACC Stop-and-Go) has been further increased [12].

Automatic Emergency Pullover Idea

According to National Highway Transportation Safety Administration's National Motor Vehicle Crash Causation Survey for Automated vehicles, there are many reasons for the road accidents and there may be remedial ways to avoid them using autonomous driving technologies.

As far as, physical impairment of the ability to act, is concerned. The report says, a driver who is suffering from a heart attack or other physical impairment is not able to take full control of a partially automated vehicle [13]. Observing that 2% of accidents are caused when some sort of physical impairment, such as a heart attack, inhibits the driver's ability to control his vehicle effectively. Depending on the trip and the automated vehicle's response, the technology could produce either a worse or better result for passenger in the car [13]. If only a minor accident has

caused by the heart attack or forces the driver to pull over to the roadside, having the driver in control may actually save his life. In these instances, the driver may get medical attention more quickly than if s/he were to continue all the way to his/her destination, assuming, of course, that his destination is not a hospital [13]. On the other hand, a fully automated vehicle will likely provide better protection to the other traffic drivers. Moreover, if the destination of driver is close at hand, delivering the driver safely to his destination may allow for medical attention to be delivered in a comparatively much safer way than causing a minor accident or forcing the driver to pull over off the road [13].

1.3 Problem Statement and Major Contributions

The objective of this research is to design and implement the automatic emergency pullover system which can prevent the potential collision in a scenario when driver gets an impairment. To achieve this purpose, active safety features were implemented to make a semi-automated vehicle on a driving simulator environment.

The following work in this thesis elaborates the approach and algorithm for features addition to the simulator. It can be really helpful in understanding the background architecture of the software, for emendation towards full ADAS or regarding other research work/study of human driver behavior under different driving circumstances. Simulation results are analyzed to provide a better understanding of vehicle performance and efficiency, so that it will be helpful in future development of scenario and improvement in vehicle dynamics model, outstandingly.

The supporting active safety systems whose implementation is mainly addressed in this thesis are mentioned below:

1. Lane Departure Warning and Lane Keeping Assist System.
2. Blind Spot Monitoring System.
3. Forward Collision Warning/Vehicle Automatic Emergency Braking System.

4. Pedestrian Automatic Emergency Braking System.

5. Adaptive Cruise Control System.

Automatic Emergency Pullover system is the major safety feature targeted in this work. Since, AEP working requires all the active safety systems. Therefore, implementation covers lane keeping assist, blind spot monitoring, automatic emergency braking (vehicle and pedestrian) and adaptive cruise control systems.

These all are to evaluate and enhance the autonomous systems for the real life vehicle testing scenarios. Although, the simulation helps in generation of the critical scenarios and testing for the efficiency of information processing and control algorithms ahead of time and with less cost. Various advancements can be proposed into the system with ease of editing in the model and its working can be perfection-ed under many conditions. However, there is always a difference between the simulation and actual world cases, the study is not able to solve each and every problem. There is always room for further in-depth research. We can focus on each specific problem separately and can identify the errors for further rectification based upon its complexity.

This thesis has been organized as follows. Chapter 2 will describe the Hardware and Software (scenario/model making) environment. Chapter 3 will present basic model, scenario developments and automatic emergency pullover system approach. Chapter 4 will go through the algorithm, simulation, model development and results for assisting active safety systems. Chapter 5 will elaborate the architecture and implementation of automatic emergency pullover system with its result analysis. Chapter 6 concludes this thesis and discusses the future work.

2. HARDWARE AND SOFTWARE ENVIRONMENT FOR A RTI DRIVING SIMULATOR

In this thesis we have used the Realtime Technologies Inc, hardware and software. It has provided us the capability to design scenarios and make implications according to our research needs among multidisciplinary researchers on a three channel desktop Simulator.

2.1 Driving Simulator Hardware Description

Realtime Technologies, Inc. hardware is shown in Fig. 2.1 below.

(a) Three channel and cabin with joystick

(b) Three channel CPUs

Fig. 2.1. Hardware provided by Realtime Technologies, Inc.

Three CPUs are synchronized with their respective channel screens and with an operating station (laptop). The laptop is used for running the Full Simulation Distributed Model on the rest of the channels. All the channels have SimCreator software, as well. The speakers are connected with center channel whereas, the joystick (steer-

ing, padels, shifter) is connected with the left channel. All channels run their part of scenario and accompany together, through SimCreator simulation model running on operator station. The block diagram for simulator working is shown below.

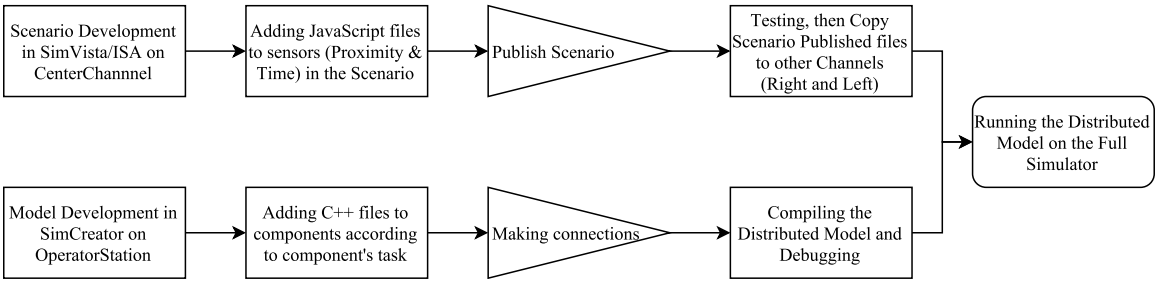


Fig. 2.2. The block diagram showing basic simulator working.

2.2 Software Configuration and Description

The main simulation model is shown in Fig. 2.3 below.

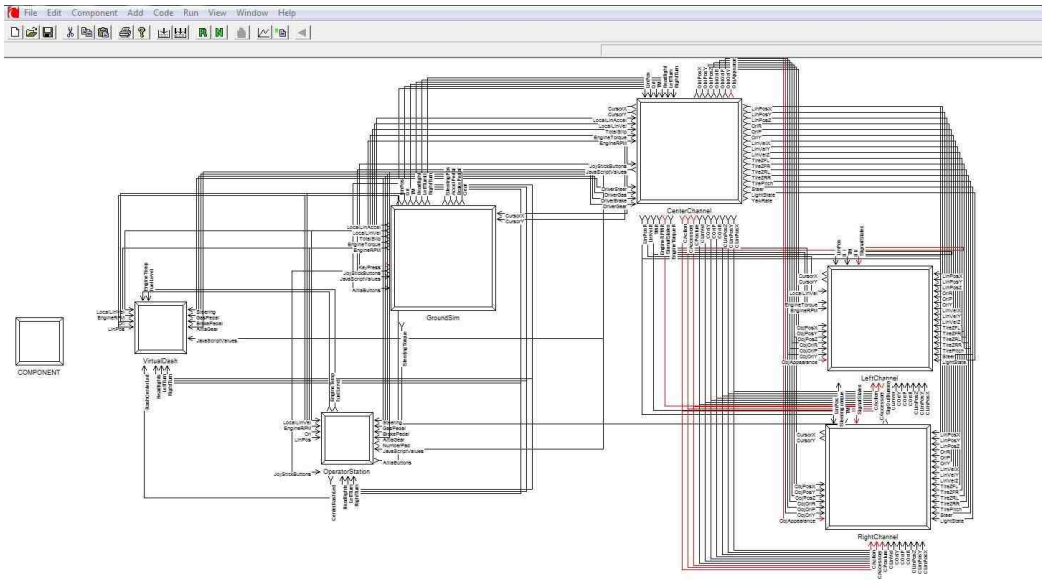


Fig. 2.3. A full distributed model diagram for blocks specified to channels and vehicle dynamics.

The full model contains the following blocks [14]:

- **GroundSim** This component handles the I/O from the instrumented (Logitech Joystick Hardware) vehicle, tires, dynamics, and power train models.
- **CenterChannel** The center visual display is computed from this component. This component also calculates the positions of all the moving models (vehicle and pedestrian) in the scene and sends the moving model information to the center display.
- **RightChannel** The right visual display is computed from this component. It also calculates the position of all the moving models (vehicle and pedestrian) in the scene and sends the moving model information to the right channel display.
- **LeftChannel** The Left visual display is computed from this component. It also calculates the positions of all the moving models (vehicle and pedestrian) in the scene and sends the moving model information to the left channel display.
- **Virtual Dash** The virtual dash is the LCD display for the instrument cluster. This runs Altia and display is shared with the host computer.

2.2.1 SimCreator Modeling Tools

Adding Connectors to Model Components

Each Component has associated to it a set of specified connectors. For Group components, these connectors can be interactively added, modified, and deleted. [14]. There are three kinds of connectors: input, state and output. Only output and input connectors can be added to a Group component [14]. For Groups all connectors are considered as matrices. To make a vector connector, set the rows to one. For a scalar connector, set the rows and columns to zero [14]. Only fixed dimension vector connectors are supported for group components. There are various associated data items with a connector [14].

They are:

- Name.
- Description.
- Units.
- Side.
- Location.
- Rows.
- Columns.
- Type.

Data Selection and Plotting

The software can be used for plotting the specific data as per your needs. Following are the ways to do so [14];

1. Select the plot flag on each output connector.
2. Select the plot variable for each component.
3. Select the plot variable for the model.

1. **How often to collect Data** After selecting what variables to collect, we must decide how often to collect data. How often data is collected is a combination of the DeltaT and the PlotMult values set when you run the model.

    - *DeltaT* is the time step we specify for the simulation. This value must be the same or smaller than the smallest update rate within model. For example, if model's tire query component is running at 400 Hz then the time step for the tire model is 0.0025 seconds or 2.5 milliseconds.

- The *PlotMult* is the value that tells the simulation, how often to collect data based on the *DeltaT* for the simulation. For example, if the PlotMult = 1, then for every time step we would collect data. However, if the PlotMult = 10, then for every 10^{th} time step we would collect data.

2. **Accessing the Data** There are several ways to access the data once we have set up the data collection. The first is through the SimCreator interface. The second is to access GnuPlot files (.plt and .hdr) directly. GnuPlot is a freely available plotting tool.

3. **Saving Data after experimental test** In order to save data from an individual drive we will need to duplicate that data. Using the *Duplicate* button with the data file selected we would like to save, we can make a copy of the collected data for future analyzing purpose.

2.2.2 Common Variables Description

When deciding to collect the data following are the two main methods [14]:

1. Plot flag is described above, when you select the plot flag on the connector of a component, it is written to the data stream.

2. Mostly the data can be extracted, such as accelerator, brake pedal, acceleration and velocity, etc. However, some specific scenario sub-system data, such as road position, headway, tail way, etc.

Below is a table of some variables and their associated descriptions. This is not at all inclusive list, different SimCreator model vary.

Table 2.1.: [14] Common Variables

Variable	**SimCreator Connector**	**JavaScript Command**	**Description**
Acceleration	Dynamics-LocalLinAccel[0]	getAcceleration()	Value in m/s/s
Lateral Acceleration	Dynamics-LocalLinAccel[1]	NA	Value in m/s/s
Accelerator Pedal	From the CabIO system	getThrottle()	Value between 0 and 90 deg. This is the angle of the pedal position.
Deceleration	Dynamics-LocalLinAccel[0]	getDeceleration()	Value in m/s/s
Break Pedal Force	From the CabIO system	getBrake()	Brake force. value between 0 and 170 representing N.
Gear	From the cabIO system	getGear()	Gear
Heading	NA	getHeading()	A value in degrees (deg) representing the heading of the vehicle. A value of 0 is North.
Heading Error	Scenario-HeadingError	getHeadingError()	Angle (deg) between road path and current heading.
Headway Distance	NA	getHeadwayDistance()	A value in meters (m). This is the headway distance between the CG of the vehicle to the CG of the vehicle ahead.
Headway Time	NA	getHeadwayTime()	A value in seconds (s). This is the headway time between the CG of the vehicle to the CG of the vehicle ahead.

continued on next page

Table 2.1.: *continued*

Variable	**SimCreator Connector**	**JavaScript Command**	**Description**
Lane Number	NA	getLane()	An integer value of the lane the vehicle is in. Lane numbers start from the center line and go out. Positive lane numbers are the positive direction of the road. When determining if you have crossed into on-coming traffic check for a sign change (e.g., -1 to 1)
Lane Offset	NA	getLaneOffset()	A value in meters (m) of the position of the vehicle from the center of the lane. Positive numbers are to the right of the lane and negative are to the left.
Road Offset	NA	getRoadOffset()	A value in meters (m) of the position of the vehicle from the center line. Positive numbers are to the right of the center line and negative are to the left.
Steering Wheel Position	From the CabIO system	getSteer()	A value in radians (rad) describing the steering wheel position.
Tailway Distance	NA	getTailwayDistance()	A value in meters (m). This is the tailway distance between the CG of the vehicle to the CG of the vehicle behind.
Tailway Time	NA	getTailwayTime()	A value in seconds (s). This is the tailway time between the CG of the vehicle to the CG of the vehicle behind.
Velocity	Dynamics-LocalLinVel[0]	getVelocity()	A value in m/s

Running the Test Model

A simple model is available on the SimCreator for preliminary testing purposes of the model, shown in Fig. 2.4 below.

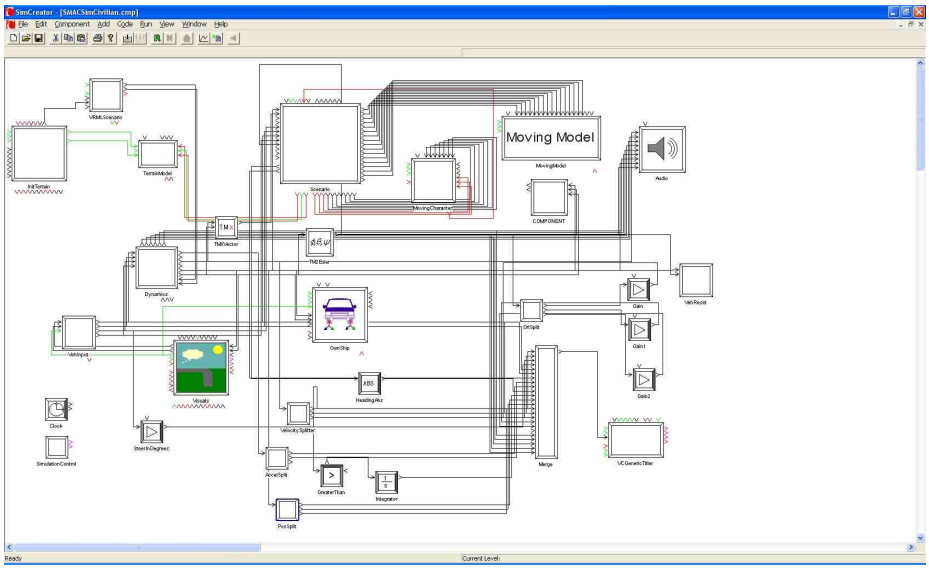


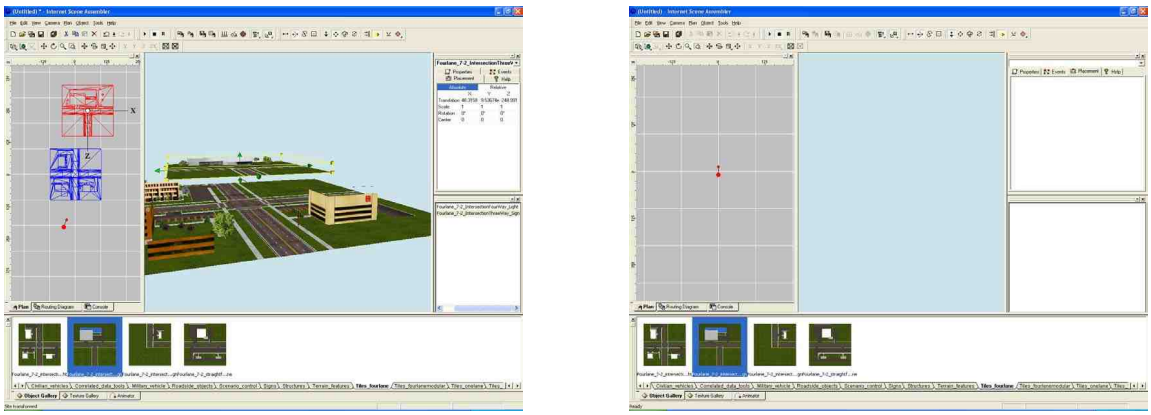
Fig. 2.4. [14] A Center Channel model diagram for the initial scenario testing.

2.2.3 SimVista Scenario Development Tools

The SimVista Scene and Scenario Development system provides a set of functionality that allows you to rapidly author your own driving simulation environments complete with traffic and pedestrian. The SMAC (Simulator Monitor Authoring and Control) model is designed to support users with limited technical skills while also enabling power users to create complex driving simulation scenarios for purposes of research and training [14].

Creating a Driving Environment

Environment authoring is accomplished using the included Internet Scene Assembler (ISA) authoring framework combined with the tiles, objects, and processes supplied with the SimVista system.



(a) ISA scene authoring software

(b) ISA window layout

Fig. 2.5. [14] Scene Development Environment for SimVista/ISA

ISA software has multiple windows available. Each window has a specific purpose and used with other ones to interact with the system altogether. The windows and their specific functions are defined below [14].

- **Main Window** This is a 3D visualization window that can be used to view and interact with the environment that is being developed.
- **Plan Window** This is a higher level view into the environment being developed. It shows the world in a wireframe rendering mode.
- **Object Gallery** The object gallery displays all objects that can be added to the environment that we are building.
- **Properties Window** The properties window shows specific information about the currently selected object.

- **Scene Tree** The scene tree contains a hierarchical inventory of objects that have been added to the current environment.

Adding Scenario Control Objects and Attaching a Script

Scenario control objects provide the ability to control various aspects of the simulation while it is executing. A number of different scenario control objects have been created to help you control the simulation. The most basic type is the proximity sensor. The proximity sensor is an object that has a volume. The JavaScript file associated with the sensor is defined in the sensors properties [14].

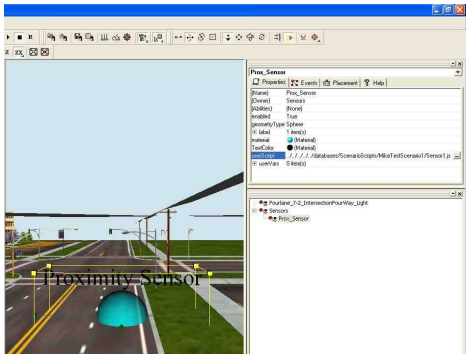
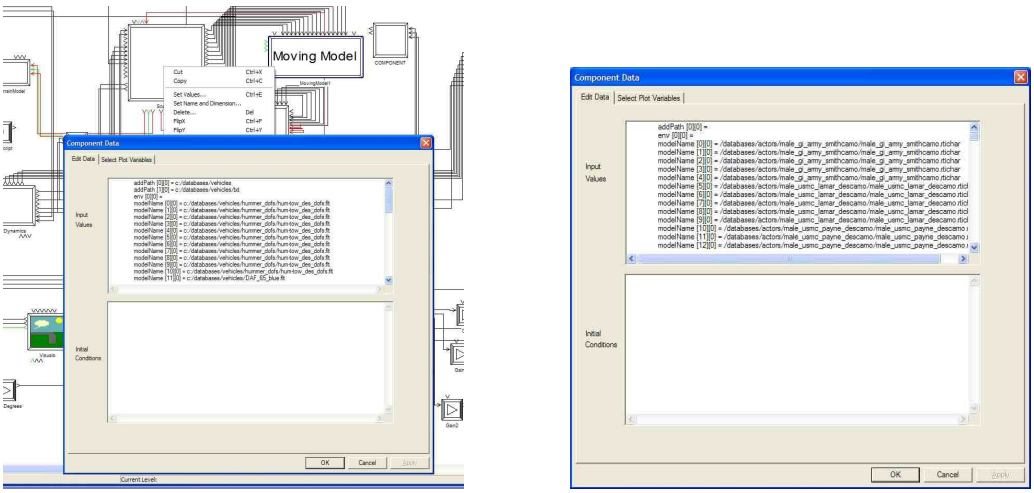


Fig. 2.6. [14] Assigning a JavaScript file to a sensor in the properties window

Dynamic Vehicles and Characters

Dynamic Vehicles, characters and objects can be used in the simulation. The *Moving Model* component includes a list of vehicles, The *Moving Character* component includes a list of actors and The *MovingObjects* component includes a list of static objects such as pylons. These all are available as the simulation is initialized. To use either the vehicles or characters or static objects we must use a Marker or Path object (path objects for vehicle and characters). One exception to this rule is the ambient traffic.



(a) Editing Dynamic vehicle assignments

(b) Editing Dynamic actor assignments

Fig. 2.7. [14] Dynamic Assignments for moving model and characters

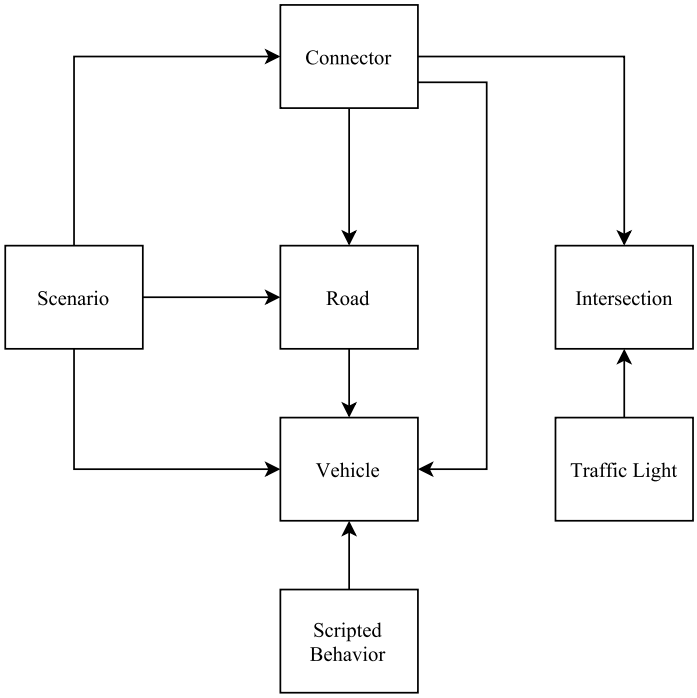


Fig. 2.8. Ambient Traffic Model and its behavior in scenario

Control Object Set

The following set of control objects has been developed to support the scripting of Scenario Control functions [14].

1. **Sensor Objects**

   SimCreator defines an object model inherited by all sensor objects. This object model includes built-in methods, properties and common event structure.

   All sensors contain the following built-in methods:

   - **enable()** Enables the object for processing and potentially generating events. Enabled objects are processed every frame by the Sensor controller, and they therefore expend system resources. Calling enable() generates the event *onEnable()* for that object, and the enabled property will be set to TRUE.
   - **disable()** Disables the object from processing or generating events. Disabled objects are removed from the Sensor controllers processing loop, and they therefore require minimal system resources. Calling disable() generates the event *onDisable()* for that object, and the enabled property will be set to FALSE.

2. **Events**

   Events are sent either by SimCreator or the sensor and are optionally captured within the users JavaScript code. Some events send arguments that may be optionally captured by the method.

   - **onEnter()** The *onEnter* method is called at the start of an operation for the sensor.
   - **onLeave()** The *onLeave* method is called at the end of an operation for the sensor.
   - **onActivate()** The *onActivate* method is called for continuous sensor events.

All sensors may define and generate the following generic events:

- **onInitialize()** It is called at the start of the simulation, after all sensors that were read have been created but before any other events have occurred.
- **onShutdown()** It is called at the end of the simulation. Any finalization necessary should be done in *onShutdown()*.
- **onEnable()** It is called when the sensor state changes from disabled to enabled. This event will not be generated at the start of the simulation.
- **onDisable()** It is called when the sensor state changes from enabled to disabled. Works after *onEnable()* only.

3. **Proximity Sensors**

   There are following sensor specific Events in the *Proximity Sensors*.

   - onInitialize()
   - onEnter(simEntity, direction)
   - onLeave(simEntity, direction)
   - onActive(simEntity, direction)
   - onEnable()
   - onDisable()

4. **TimeSensors**

   Time Sensors generate continuous or one-shot events at programmed points in time. Time Sensors have a notion of a start time and a stop time between which continuous events may be generated. Continuous events can be enabled and disabled at run-time, and the period with which these events are generated can be modified. The following diagram depicts the events as they would occur with a Time Sensor throughout its life cycle.

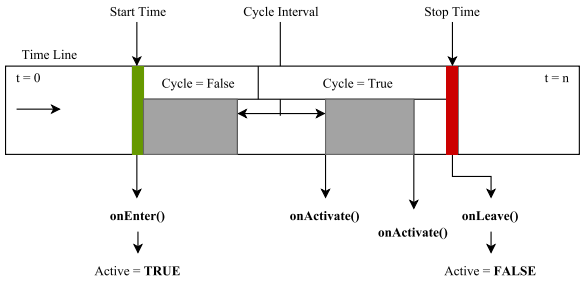


Fig. 2.9. Time sensor life cycle and functioning by events at programmed points

At Start Time and at Stop Time an onEnter() and an onLeave() event is generated respectively. Meanwhile, some continous Events may occur on cycle interval boundaries, as measured from the Start Time. When cycle is true, the remainder of the intervals generate an onActivate() Event. There are following sensor specific events in the Time Sensor.

- onEnter()
- onLeave()
- onActivate()

There are also several built-in methods for the Time Sensors

- setStartTime(*time*)
- *time* getStartTime()
- setStopTime(*time*)
- *time* getStopTime()
- setCycle(*value*)
- *value* getCycle()
- setCycleInterval(*time*)
- *time* getCycleInterval()

JavaScript Command Reference

The class hierarchy developed takes advantages of the object-oriented programming techniques. There are singleton classes included in the Scenario development suite. Below are mentioned:

- Audio
- Scenario
- Visuals

Moreover, all the objects that have a position associated with them inherit from the Spatial Datum base class. Fig. 2.10 [14] has the class description.

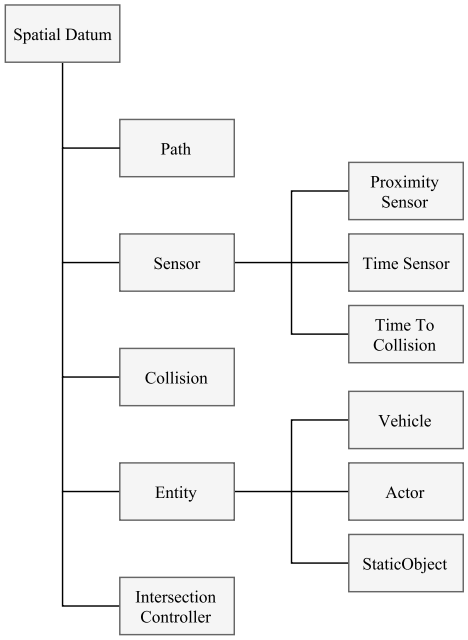


Fig. 2.10. Class Hierarchy for Spatial Datum class associated with scenario

Each class has subsections. There are major components of a class then come any state variables associated with the class, at the end are the functions associated with that class.

3. AEP SYSTEM STRUCTURE AND SCENARIO DEVELOPMENT

3.1 AEP Functionality Description

AEP states that, a vehicle on the road equipped with AEP system can automatically pullover in the roadside safely, when the driver is considered incapable of driving. The integration of systems for the AEP system is shown in Fig. 3.1 below.

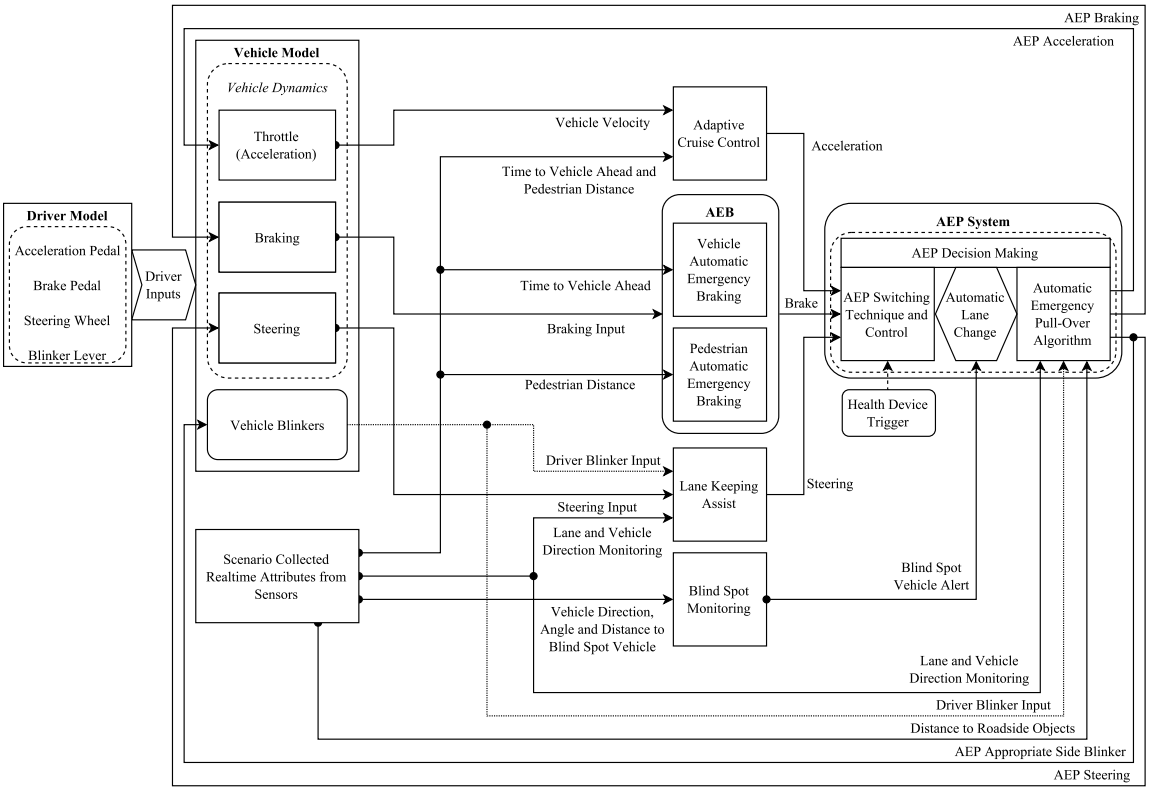


Fig. 3.1. The block diagram for automatic emergency pull-over system

After the AEP system receives trigger from health device, it automatically controls the vehicle pullover maneuver to avoid an accident. But if the trigger is a false alarm from the health device due to its malfunction, driver have a button in his/her hand to reverse back the trigger and take the control of the vehicle. We should also consider installing this manual button in actual vehicle systems for more driver control.

An example application of AEP is shown in the situation when the driver loses the ability of driving, such as faint or lost consciousness due to critical medical conditions which could result in a fatal accident. AEP operation consists of an interconnection of various active safety systems which continuously monitor the vehicle until safe pull over. The active safety systems involved include Lane Keeping Assist (LKA), Automatic Lane Change (ALC), Blind Spot Monitoring (BSM), Automatic Emergency Braking (AEB) and Adaptive Cruise Control (ACC). The design and implementation for all these supporting features has been explained in Chapter 4.

The functioning approach used for AEP is described in the flowchart in Fig. 3.2.

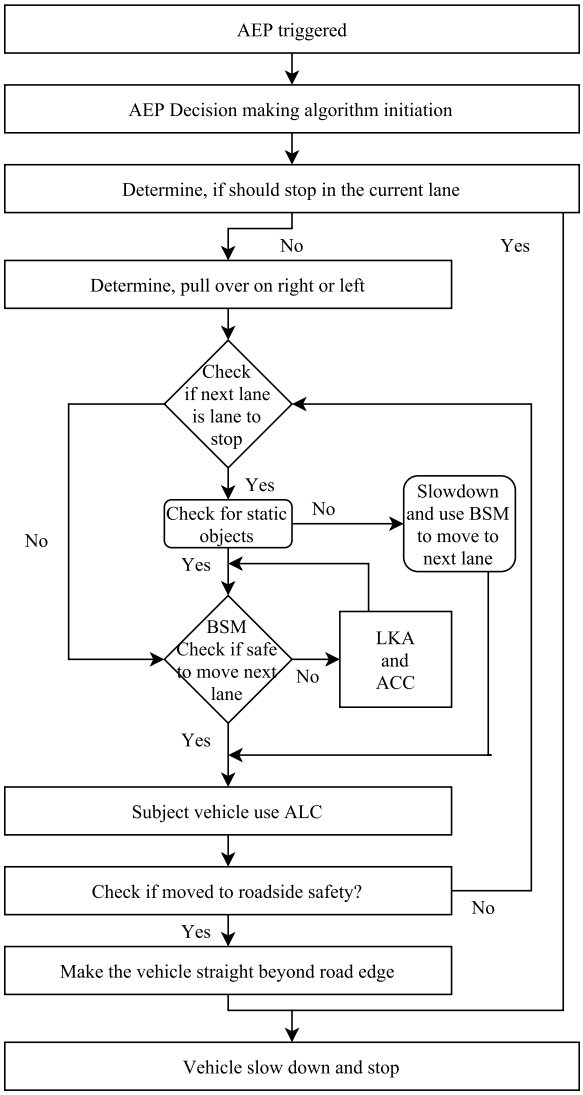


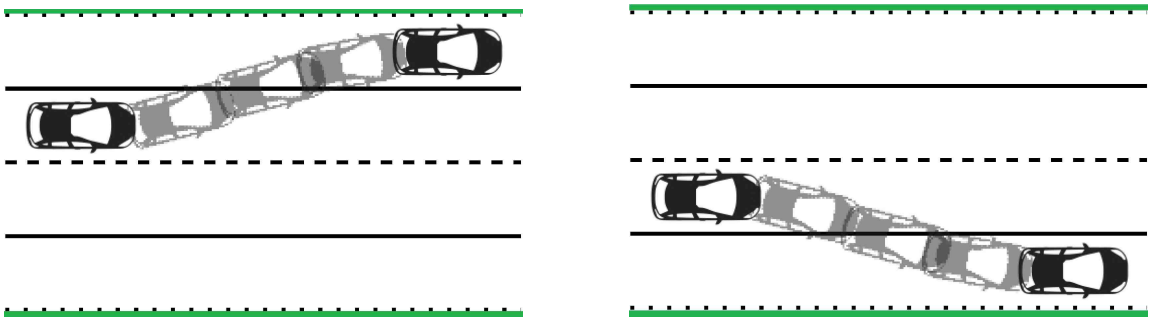
Fig. 3.2. The functioning of automatic emergency pull-over system

3.2 AEP Possible Maneuvers

There could be numerous roadside pullover scenarios but prominent of them are mentioned here. The possible maneuvers for AEP are explained in the following sections.

Right or Left Roadside Simple Maneuver

A simple roadside pullover is when the health device send trigger and the vehicle is steered to the left or right roadside safety lane to pullover, as shown in Fig. 3.3



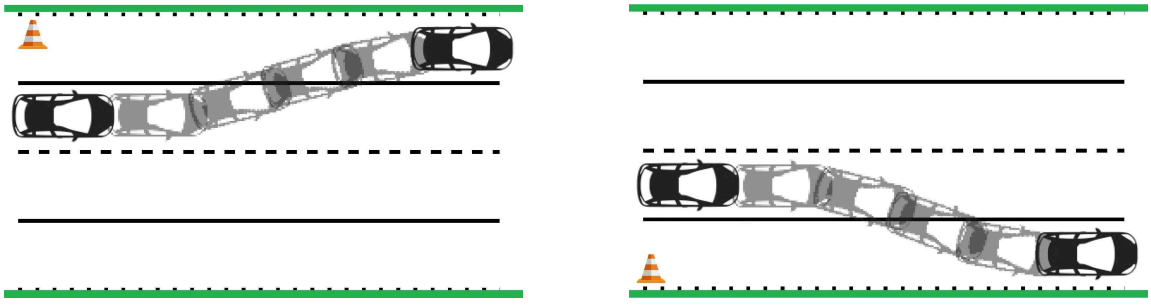
(a) AEP Simple Left Pullover

(b) AEP Simple Right Pullover

Fig. 3.3. Automatic emergency pullover simple maneuver

Right or Left Roadside Single Object (Far) Maneuver

A single object (far) roadside pullover is when the health device send trigger and vehicle tries to pullover on the nearest roadside but found a distant object. In such a condition, it keeps moving on and pullover at the adjacent roadside safety lane, a shown in Fig. 3.4.



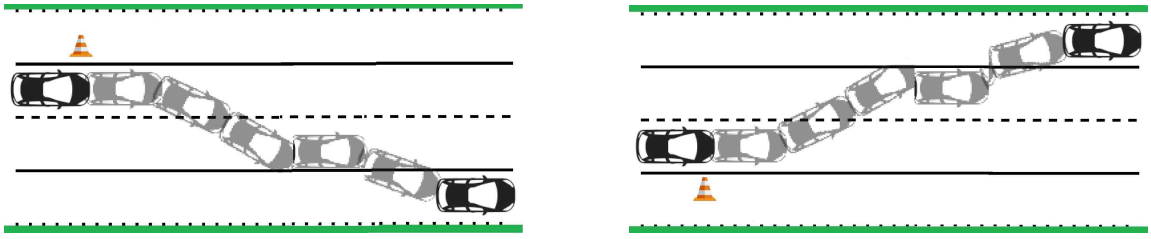
(a) AEP Single Object (far) Left Pullover

(b) AEP Single Object (far) Right Pullover

Fig. 3.4. Automatic emergency pullover single object (far) maneuver

Right or Left Roadside Single Object (Near) Maneuver

A single object (near) roadside pullover is when the health device send trigger and the vehicle tries to pullover on the nearest roadside but found object immediately on the nearest pullover location. Therefore, it changes the lane and tries to pullover to the roadside safety location other than the nearest one, as shown in Fig. 3.5.



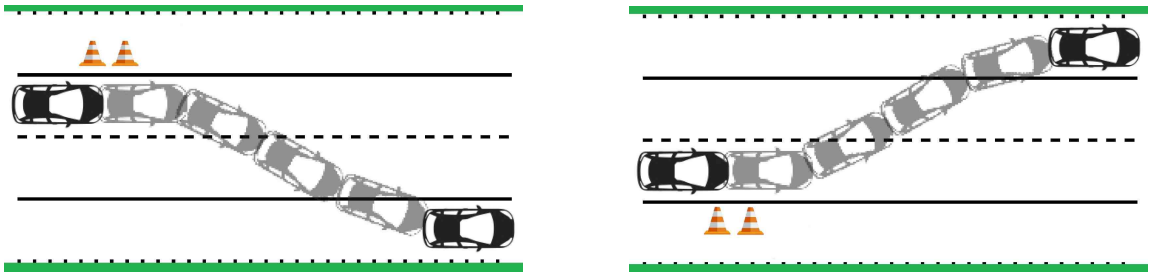
(a) AEP Single Object (near) Left Pullover

(b) AEP Single Object (near) Right Pullover

Fig. 3.5. Automatic emergency pullover single object (near) maneuver

Right or Left Roadside Multiple Objects Maneuver

A multiple objects roadside pullover means, after health device trigger vehicle tries to pullover on the nearest roadside location but found multiple objects there. Thus, it changes lane and tries to pullover to other roadside safety, as shown in Fig. 3.6.



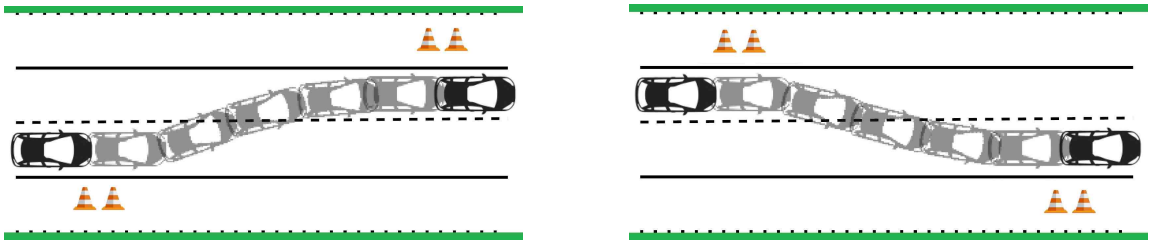
(a) AEP Multiple Objects Left Pullover

(b) AEP Multiple Objects Right Pullover

Fig. 3.6. Automatic emergency pullover multiple objects maneuver

Right or Left Both Roadside Multiple Objects Maneuver

This is a continuation for the previous maneuver. However, it changes the lane and tries to pullover to the roadside safety location other than the nearest one but still find multiple roadside objects on the other roadside location too and interpret a potential collision with these objects. Ultimately, it brakes on the road lane and stops on the current lane, as shown in Fig. 3.7.



(a) AEP Multiple Objects both side Left Pullover

(b) AEP Multiple Objects both side Right Pullover

Fig. 3.7. Automatic emergency pullover both side multiple objects maneuver

Two scenarios (Fig. 3.3 and Fig. 3.5) have been addressed in the latter chapter for AEP. The maneuvers are explained with the help of resulting plots.

3.3 Full Simulation Model

The main developed model is shown in Fig. 3.8. This figure illustrates the connections updated in the main model. Each connection has a specific purpose. The variables transfer will be explained in the following chapters.

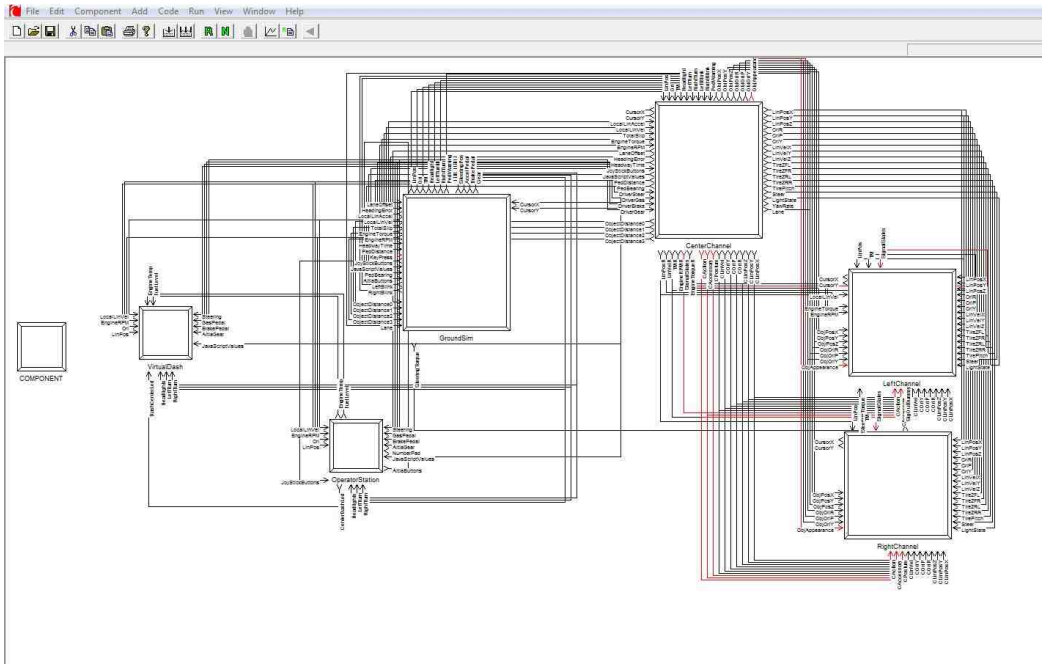


Fig. 3.8. FullSim model in SimCreator with the proposed modifications

There are major developments in the GroundSim model which regulates the dynamics of the vehicle and extracts the inputs from the scenario and make those variables as the inputs for the SimCreator model vehicle dynamics component blocks.

The variables are transferred in the specific fashion from Center Channel to the GroundSim through the transfer component block in each major component (i.e. GroundSim, CenterChannel, etc.).

The outputs from the blocks of active safety features are summed in an order as per the priority of the system. For example, PAEB has to work in every condition weather the ACC is ON or not but if there is some other feature which has to work only with ACC then its functionality becomes ON only when the cruise control key/switch has been pressed by the driver on the joystick. The sum arrangement as per the requirement of blocks is explained in Fig. 3.9.

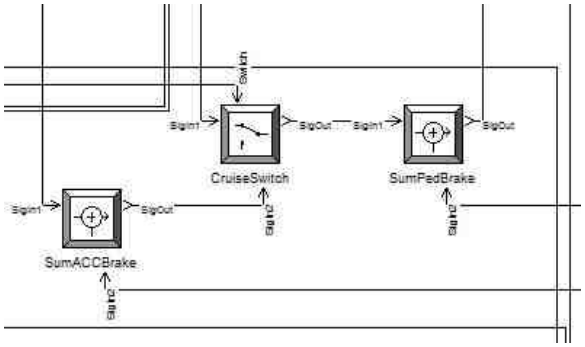
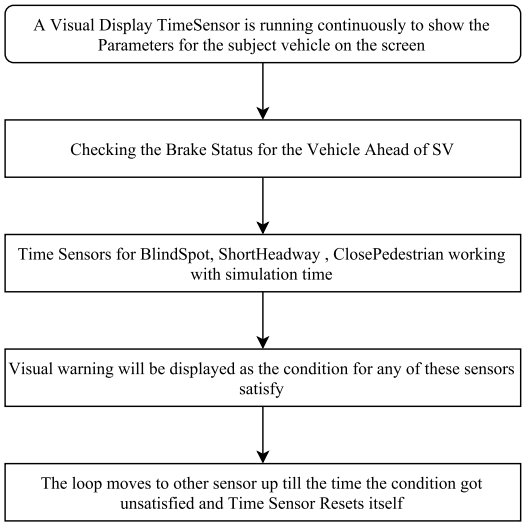


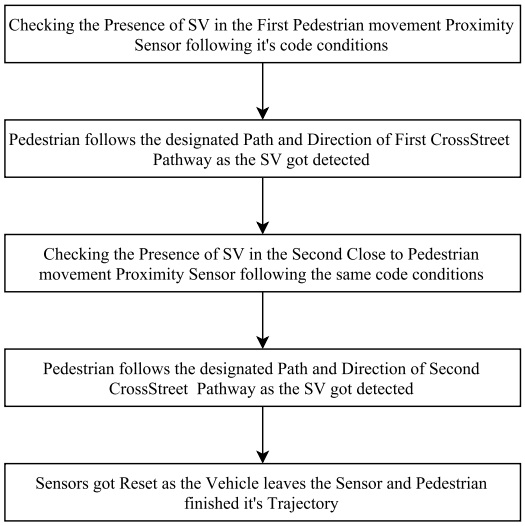
Fig. 3.9. Summation blocks placement for all the systems together

3.4 Simulation Scenario Development

The functioning of the sensors has been explained in the earlier chapter. Similarly, the role extracted from the sensors in the developed scenario is according to the functionality they offer for the desired purposes. The motive for the sensors used in the scenario is represented in the flowchart in Fig. 3.10.



(a) Time Sensors in the Developed Scenario



(b) Proximity Sensors in the Developed Scenario

Fig. 3.10. Sensors allocation and functionality in the developed scenario

The main developed scenario has been shown in Fig. 3.11. Apart from the static object placement in the scenario, there are some of the features which need to be added using invisible objects shown in the *Red Box*.

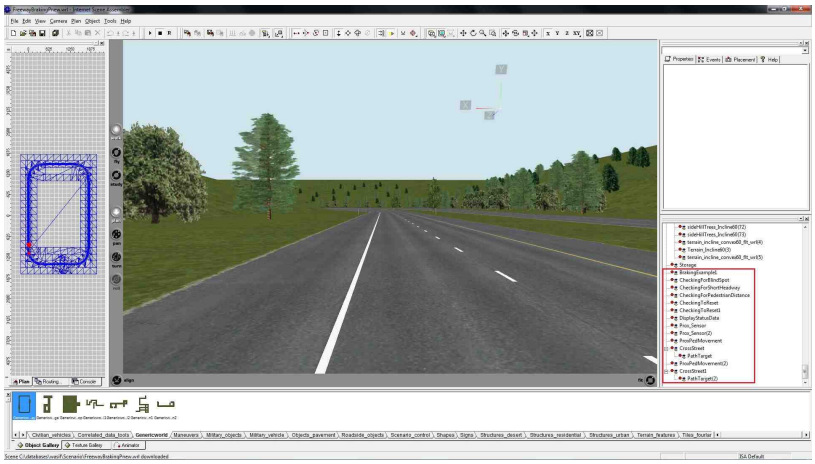


Fig. 3.11. Developed scenario showing sensors and path in SimVista/ISA

3.4.1 Time Sensors Inclusion and Usage

The basics of time sensor has been explained earlier. In this scenario, time sensors play a major role for the ambient traffic vehicle behavior to check the efficiency of installed systems.

Algorithm 1 Dynamic Braking

Vehicle Ahead SV (Brake Sensor Active)
Vehicle Ahead Velocity = oldVelocity - 5
if (Distance of SV to VehicleAhead < 10)
if (Vehicle Ahead Lane = 1)
gotoLane = 2
else if (Vehicle Ahead Lane = 2)
gotoLane = 1
end if

3.4.2 Proximity Sensors Inclusion and Usage

Different types of proximity sensors are used with respect to their functionality. The presence of proximity sensor has been shown in Fig. 3.12.

Fig. 3.12. Arrangement of a proximity sensor in typical scenario

The first spherical sensor initiates the pedestrian in the scenario and when the vehicle moves through the sensor, pedestrian starts moving with the defined velocity and acceleration. The sensor for pedestrian movement is shown in Fig. 3.13.

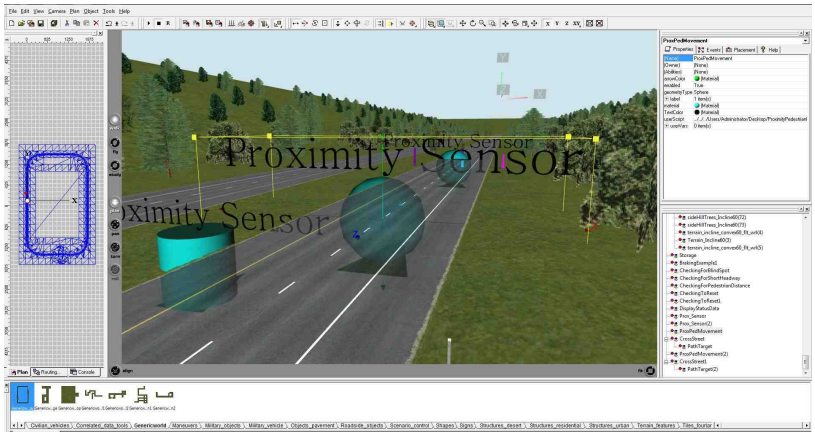


Fig. 3.13. Proximity sensor for the first movement of the pedestrian in the scenario

Now, after pedestrian movement initiation takes place, pedestrian requires a path to follow on and to do the prescribed action at the end of the path (i.e. to stop, etc.). This pathway stops the pedestrian in the middle of the lane the vehicle was initiated, in the start of simulation (i.e. Lane 2). The pathway defined for the pedestrian is shown in Fig. 3.14.

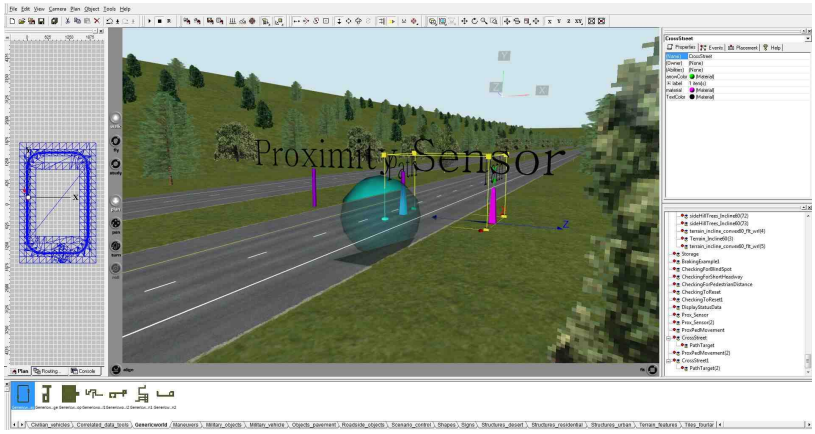


Fig. 3.14. Direction and cross street way for pedestrian movement

The second spherical proximity sensor is placed just as close to the pedestrian stop position because pedestrian stands there until the vehicle approaches to it and enters the second sensor which has the similar functionality except the pedestrian initialization into scenario (because pedestrian has been initialized already by first sensor).

The pedestrian is supposed to follow the next cross street path to reach the other end of the road and let the PAEB of the vehicle to turn Off and remove the pedestrian detection. The pathway defined for the pedestrian after the vehicle reaches close to it is shown in Fig. 3.15.

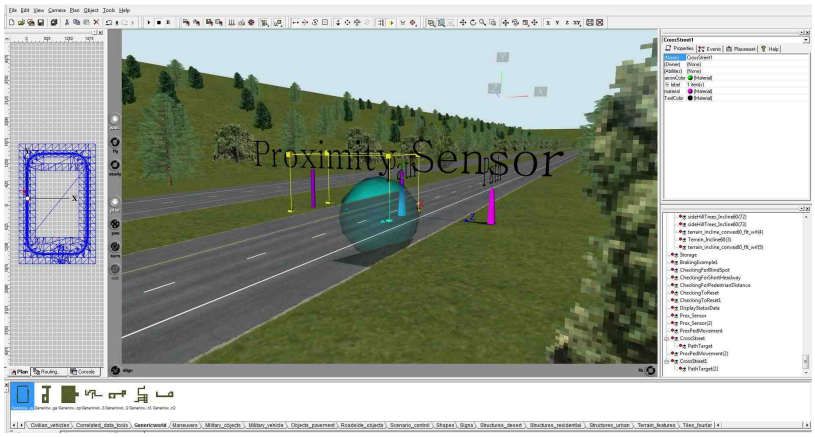


Fig. 3.15. Direction and cross street way for pedestrian after close vehicle

Ultimately, when the pedestrian moved away, vehicle regains its control whether it's in the driver's control or in the mode of ACC.

3.4.3 Static Objects Placement as Dynamic

The static objects are placed in the scenario as per the requirement but if it is required to extract the attributes like distance, angle from the objects then these objects need to be placed as a Dynamic Marker in the scenario. Afterwards, the Marker type should be identified (it can set, which object do you want to show in the scenario, e.g. pylon). Furthermore, the identified type axes offset, visibility and ambient traffic behavior towards the object should be specified through a Time Sensor in the scene. Its placement is just like the placement of any other object. The static object (dynamic) placement has been shown in Fig. 3.16.

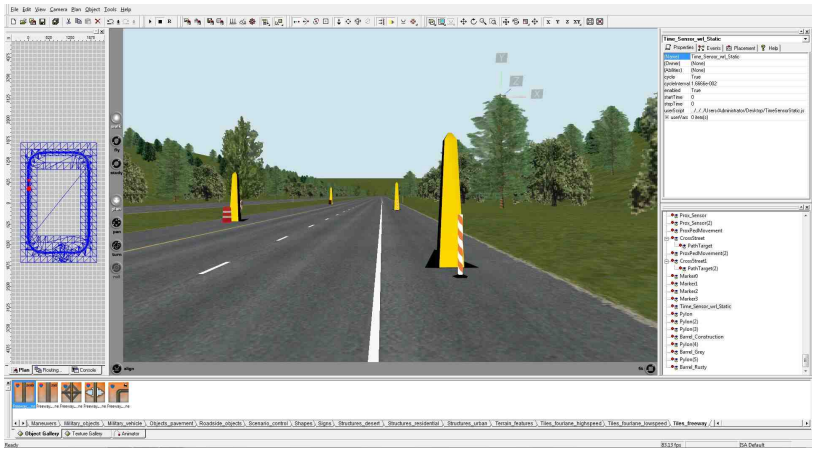


Fig. 3.16. Placement of static objects and dynamic objects in scenario

The lane numbers are defined for the particular lanes and type of lanes. This information is specified by the software ISA/SimVista. Considering the importance of this in the latter chapter (Automatic Emergency Pull-Over System), the lane numbers used for typical scenario are shown in the following Fig. 3.17.

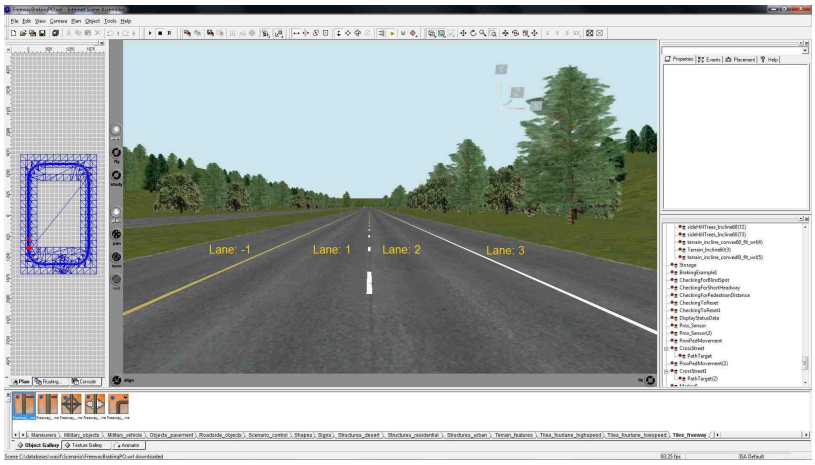


Fig. 3.17. Lane Numbers identification for a typical freeway scenario

Lane numbers are required for some particular system programming in the simulation environment (explained later).

4. DESIGN AND IMPLEMENTATION OF VEHICLE KEY ACTIVE SAFETY SYSTEMS

The Active Safety Systems whose design and implementation is mentioned in this chapter along with their results and analysis are below:

1. Lane Departure Warning and Lane Keeping Assist System.
2. Blind Spot Monitoring System.
3. Automatic Emergency Braking System.
    - Vehicle Automatic Emergency Braking.
    - Pedestrian Automatic Emergency Braking.
4. Adaptive Cruise Control System.

4.1 Lane Keeping Assist System

As far as lane keeping assist is concerned, the lane departure warning is a crucial component of this system. It provides an alert to the driver before taking any preventive action by itself.

The basic lane keeping assist approach and algorithmic approach used in the model is specified in the following Figs. 4.1 and 4.2, respectively.

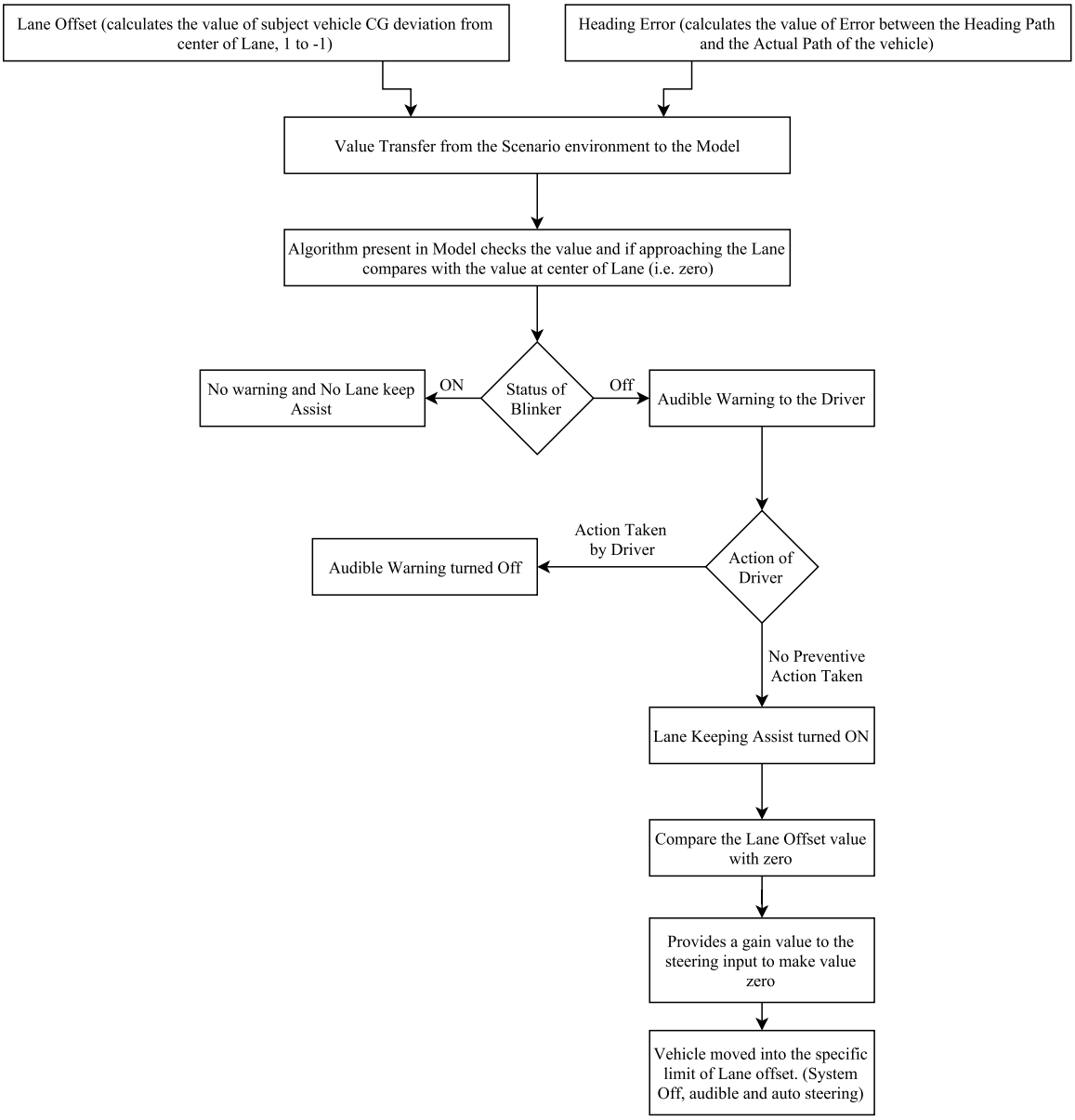


Fig. 4.1. The flowchart specifying functioning for lane keeping assist system

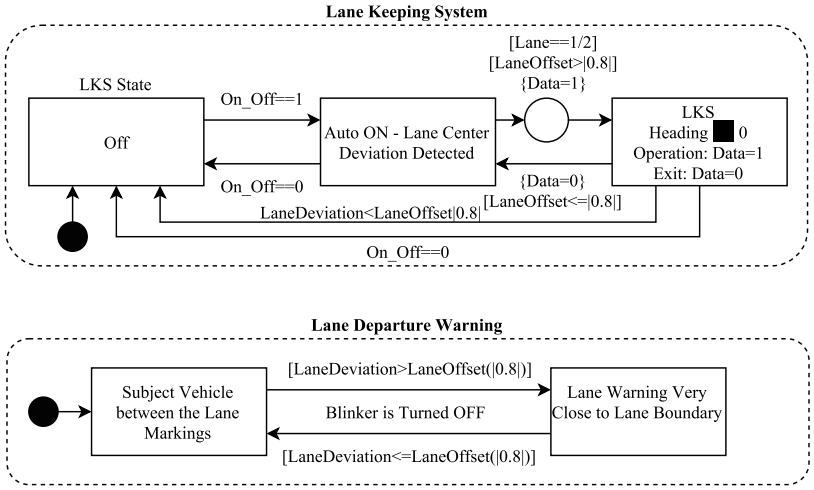


Fig. 4.2. The state flow for lane keeping assist system in the model

The main model has LaneOffset parameter transfered from CenterChannel to the GroundSim.

4.1.1 Lane Departure Warning

The LaneOffset variable is added to the Audio component in the CenterChannel and audio block uses Eq. 4.1 for lane departure warning alert.

$$Output = 1, if(-0.8 > LaneOffsetInputVariable > 0.8) \tag{4.1}$$

4.1.2 Lane Keeping Assist Module

The main approach of lane keeping assist in the GroundSim block is shown in Fig. 4.3.

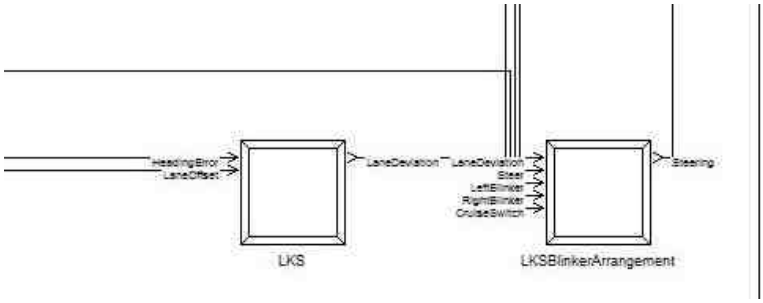


Fig. 4.3. The components consist of lane keeping assist system

The lane keeping assist system component block has two inputs. First one is the HeadingError whose definition has been given earlier the second one is the LaneOffset. The main algorithmic approach has been explained after Fig. 4.4.

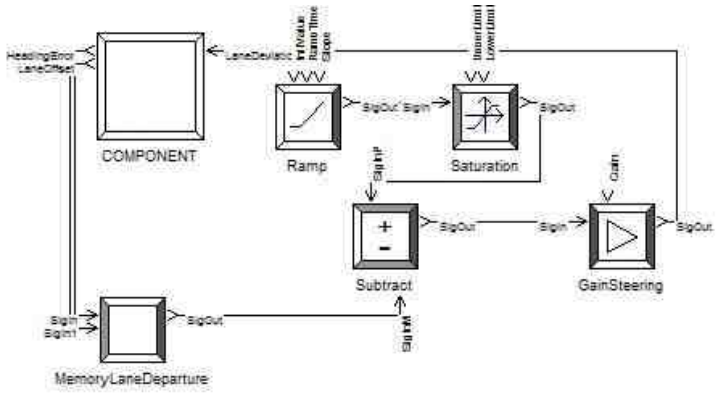


Fig. 4.4. The main components of the lane keeping assist system

This figure shows the two inputs for the *MemoryLaneDeparture* block. The computer program inside block follows with following two conditions:

$$StorageMemory = HeadingError, if(time! = 0) \tag{4.2}$$

$$Output = HeadingError, Checkcondition - 0.8 > LaneOffset > 0.8 \tag{4.3}$$

After satisfying this condition the output is inserted in the subtraction block where its difference has been calculated with the [absolute value = 0] then the GainSteering block tries to behave as a proportional controller to minimize the error and make the value tend to HeadingError = 0 (i.e. force the steering towards the center of the lane).

Lane Keeping Blinker Arrangement

The *SteeringGain* output is not yet transferred to the dynamics block of the subject vehicle, it enters next into the LKSBlinkerArrangement block which is shown in Fig. 4.5.

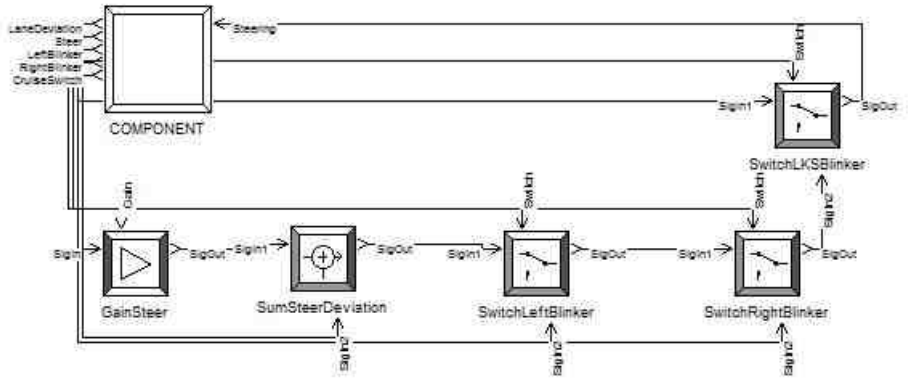


Fig. 4.5. The main components of the LKA blinker arrangement

The figure shows the two main switches for *LeftBlinker* and *RightBlinker* which removes or by-pass the HeadingError correction effect on the steering and removes the lane keeping assist system for the time being until you change the lane completely and put the blinker to OFF.

4.1.3 Simulation Results and Discussion

Results extracted from the simulation has been represented through the following plots.

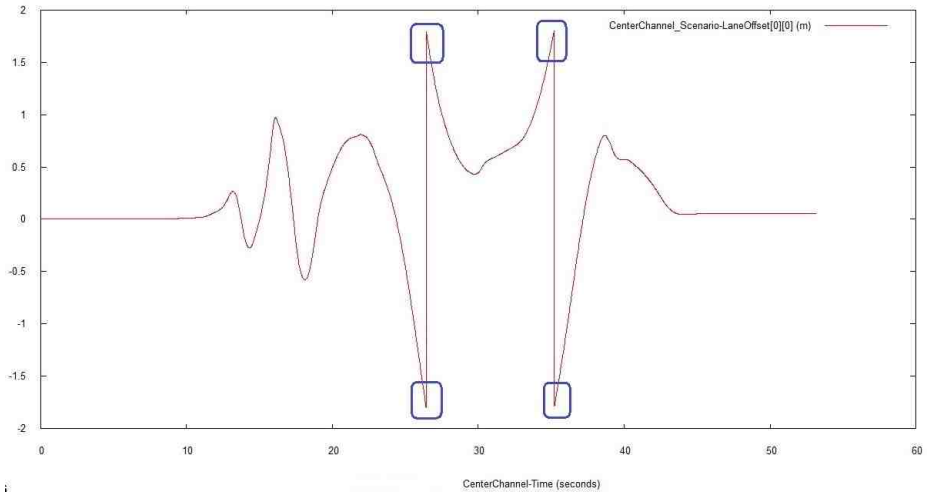


Fig. 4.6. Plot representing the value for variable LaneOffset

This plotting is showing the variation in value of *Lane Offset* variable. The starting peaks represent the position when LKS moves the vehicle towards the center of lane (i.e. LaneOffset = 0). The peaks represented with blue squares are the positions when the blinker was ON and the LKS turned OFF for that moment. Therefore, a sudden change can be noticed because the vehicle has departed the previous lane and changed to the other lane. The area between the peak values is representing the lane offset behavior in the other lane.

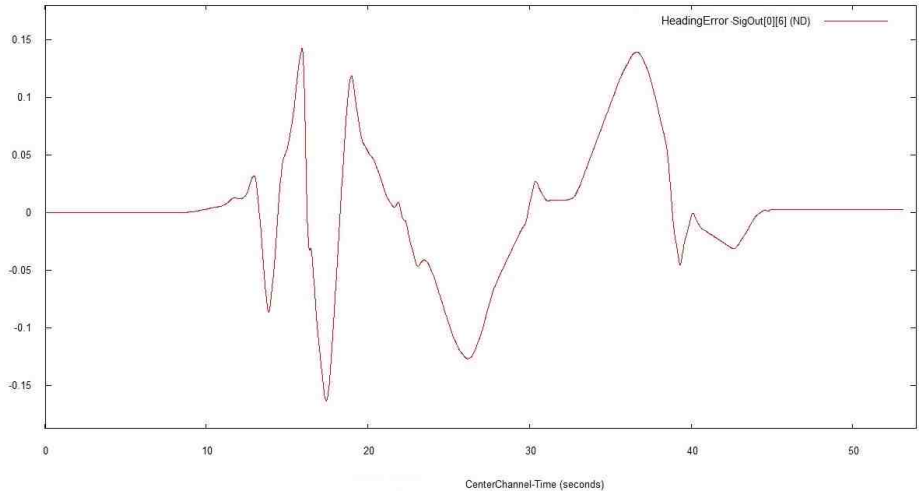


Fig. 4.7. Plot representing the value for variable HeadingError

This plot represents the value of HeadingError. The graph variation can be observed very closely as it is magnified and step size is small. A tiny variation in the HeadingError is considerable and LKS tries its best to make *Heading Error -> 0*.

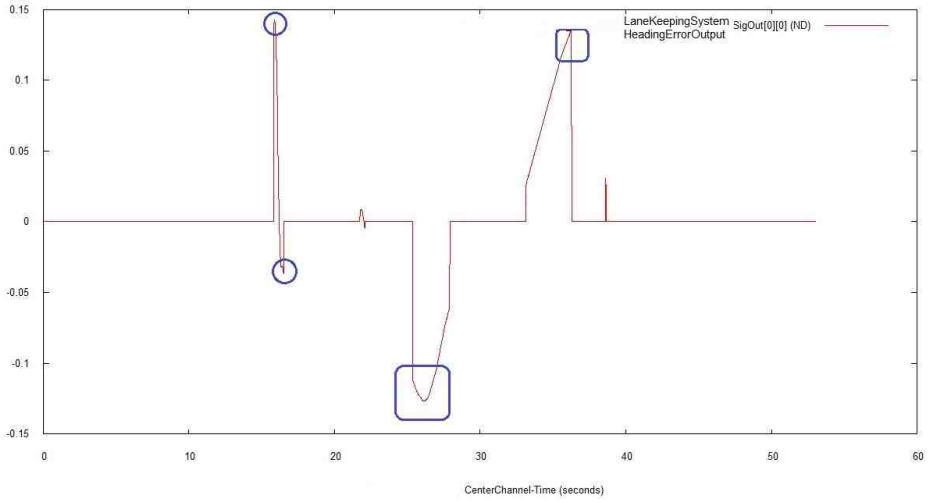
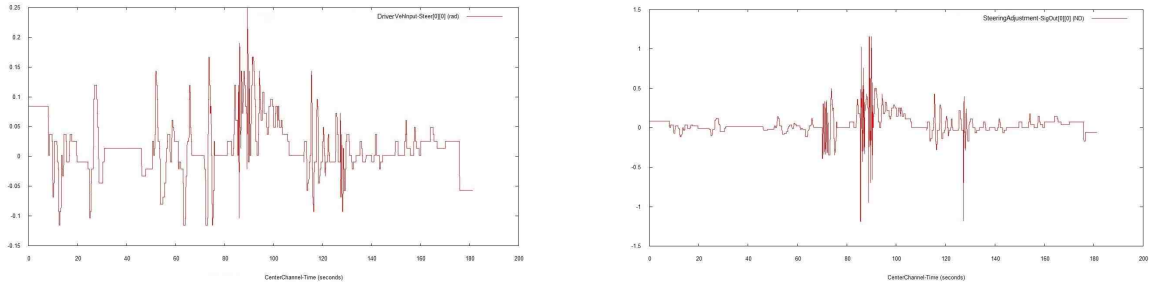


Fig. 4.8. Plot representing the value for output of lane keeping system

This plot represents a value of lane keeping memory block output. This block stores the value and gives the output of HeadingError only when the LaneOffset condition verifies. Circles indicate the output peaks with in one lane, squares indicate the output peaks at the instant when lane has been changed.



(a) Normal Steer Input (b) Adjusted Steer Input

Fig. 4.9. Steering adjustment through lane keeping assist system

The two plots are really important because they are showing continuous variations in the steering input by the driver and effort by the Lane Keeping system to minimize the variations when it is ON (keep in consideration that the units for plot 4.9(a) and 4.9(b) are different, driver steer input is in radians while adjusted steer input is force from LKA module). The peaks in the middle of "Adjusted Steer Input" are the variations when the driver was about to leave the lane and was not responding to lane departure warning hence, LKS has taken swift action to keep the lane.

One major advantage of such a lane keeping assist system which turns ON only in a scenario when the vehicle is near the lane marking, is more driver control. For instance, if driver is on a Highway and there is some long vehicle beside, then driver cannot keep the vehicle in exact middle of his/her lane because s/he needs some extra clearance distance with the long vehicle. This LKS turns ON only when driver is about to depart the lane and near the lane marking (i.e. LaneOffset value of $|0.8|$). Therefore, it prevents the annoying behavior of conventional Lane Keeping System and gives a more comfortable driving experience.

4.2 Automatic Lane Change System Concept

Automatic lane change is a system which performs freeway lane change maneuver by itself. During freeway driving lane change is a process which needs driver concentration significantly because a small mistake or negligence could result in accident. However, ALC is not autonomous driving it performs function only when the driver put the Blinker ON for either side but it provides comfort to driver and assist him/her in safe lane change. Sensors work instead of driver to avoid any potential collision. The basic approach used for ALC has been shown in Fig. 4.10 below.

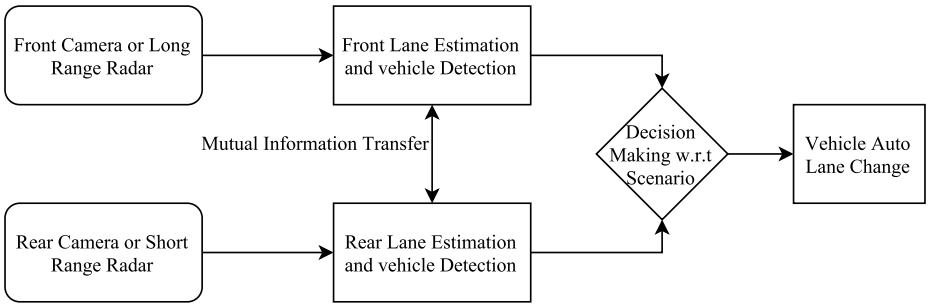


Fig. 4.10. The general approach for automatic lane change system

Different algorithmic and sensory approach can be used to achieve the purpose. An example is shown in Fig. 4.11. First, the knowledge about moving objects in the surrounding of the subject vehicle is necessary to avoid collisions. For that task, a long-range radar (LRR) mounted at the front bumper of the vehicle and two short-range radars (SRRs) at its rear end are used [15].

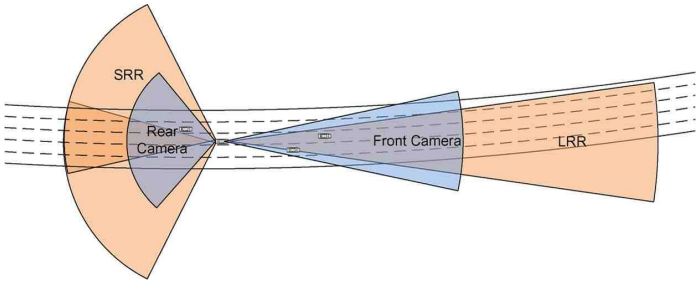


Fig. 4.11. [15] Sensor configuration of the system, consisting of an LRR, two overlapping SRRs, and two cameras.

There could be different range RADAR or different sight cameras used for ALC. *Our focus is to achieve an auto emergency pullover maneuver in a situation when driver is incapable of driving and s/he is not in a condition of putting the Blinker ON.* The AEP system only uses some functionality of ALC when it is supposed to change lane to avoid roadside objects (explained latter in AEP Chapter 5). Therefore, the description about ALC is provided here just for reader's knowledge as the study doesn't include a typical automatic lane change feature.

4.3 Blind Spot Monitoring System

Blind spot monitoring system detects the vehicle in the driver's blind spot and notify the driver with a visual or audible warning and some times with a vibratory warning. In more advanced cases it provides driver with some steering assist to prevent collision when there is a vehicle in the blind spot and the driver tries to make a lane change maneuver neglecting the vehicle in its blind spot. There is an advancement to the system introduced in the market known as 'Rear Cross Traffic Alert' which provides assistance to the driver in a scenario when the driver is trying to reverse the vehicle but overlook the vehicles moving in the back of it [16]. This is a system "which alerts drivers backing out of a parking space when traffic is approaching from the sides" [16].

The basic approach for the blind spot monitoring system is shown in Fig. 4.12.

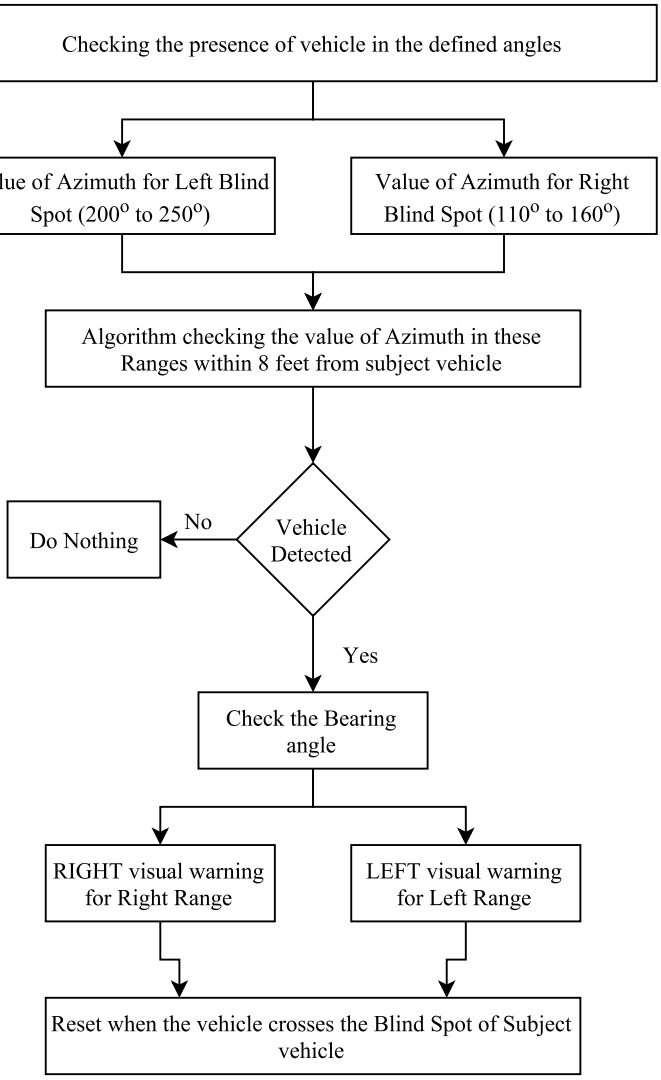


Fig. 4.12. The flowchart specifying basic blind spot monitoring system

4.3.1 Blind Spot Warning

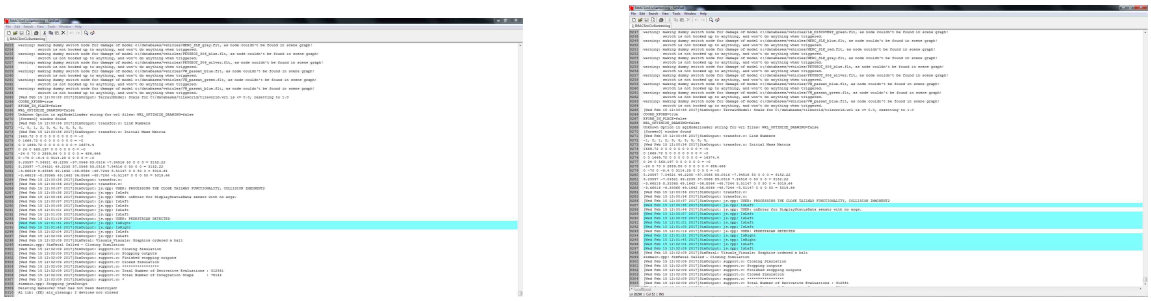
The algorithm implemented to take visual results on the desktop simulator is shown below:

Algorithm 2 BlindSpotWarning

Vehicle in SV Threshold Radius (Sensor Active)
Store the vehicle number in a string
if (Vehicle direction same as SV **and** Threshold Radius < 8)
if ($200 <$ SV bearing < 250)
VisualDisplay[Left]
else if ($110 <$ SV bearing < 160)
VisualDisplay[Right]
end if

4.3.2 Simulation Results and Discussion

Following are the results taken from run log of the simulation.



(a) Blind Spot Warning (right)

(b) Blind Spot Warning (left)

Fig. 4.13. Blind spot detection for vehicle and side identification

This warning is only an instant visual during the simulation, when the ambient traffic vehicles were observed on the right or left side blind zones. The warnings are on particular time when the vehicle was crossing its blind spot. The count of warning indicates the number of vehicles passed. Therefore, its results has been captured from the simulation output *Run log* which represents the identification of vehicle in the right or left blind spot.

4.4 Automatic Emergency Braking System

AEB is a combined name used for a mutual integration of two systems.

1. Vehicle automatic emergency braking system.

2. Pedestrian automatic emergency braking system.

4.4.1 Vehicle Automatic Emergency Braking System

An automatic emergency braking is an automotive safety system designed to mitigate the collision severity. It is known as pre-crash, collision imminent braking or collision mitigation system. Radar (all-weather), sometimes camera and laser (both sensor types are ineffective during bad weather) are used to detect an approaching crash. Once the detection is done, the system either provide an alert warning to the driver when there is an impending collision otherwise take action without any driver input (by braking or steering or both) autonomously [17]. As an emerging active safety system, AEB/CIB is already showing great benefits in improving road safety in the Real World [18]. A recent report performed by IIHS shows that the AEB technology can reduce insurance injury claims by as much as 35%. The 10 manufacturers committing to across-the-board AEB represented 57% of U.S. light duty vehicle sales in 2014 [18].

AEB systems usually improve the road safety in two ways [19]:

1. Use on-board sensors to detect objects and help to avoid accidents by identifying critical situations early and warning the driver.

2. Reduce the severity of collisions which cannot be avoided by mitigating the severity of crash and in some cases, by preparing the vehicle and restraint systems (e.g. seat belt etc.) for impact.

Cars with automatic emergency braking may also be equipped with adaptive cruise control, and use the same forward-looking sensors.

Adaptive cruise control system is explained later but the similar approach has been used for the AEB system here with the increased HeadwayTime calculation and increased magnitude for the braking when the time approaches to zero. The basic approach has been shown in Fig. 4.14.

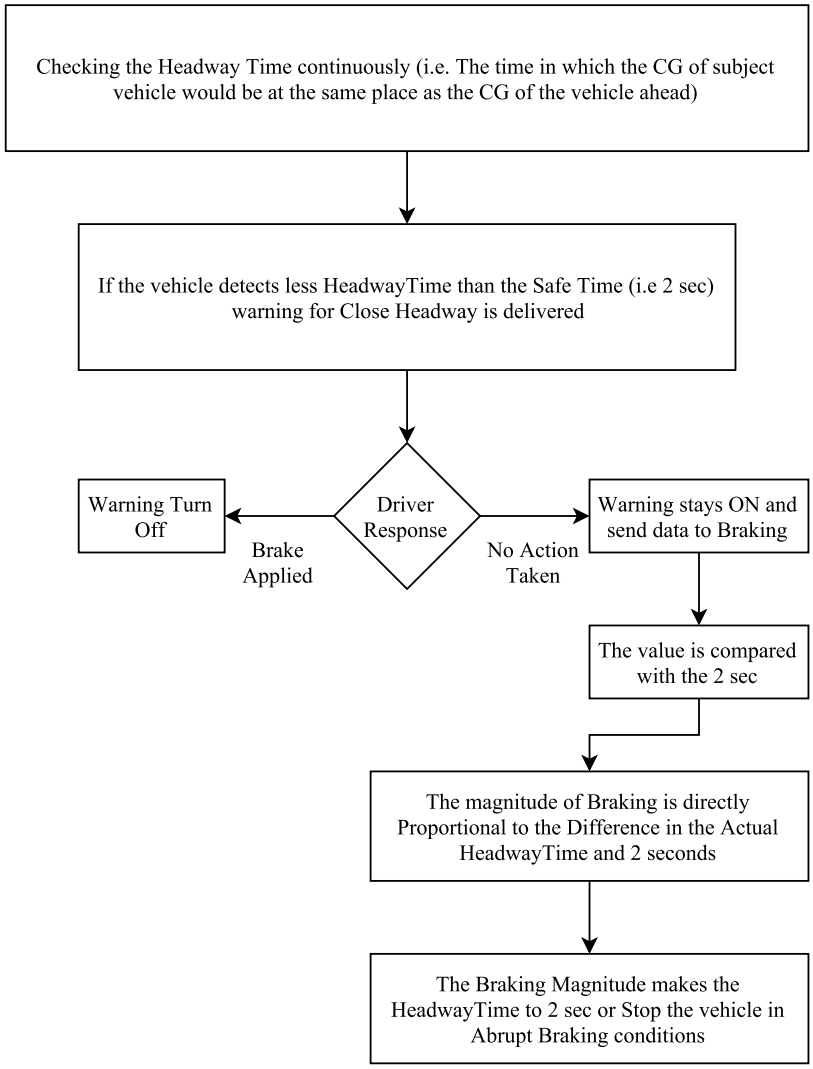


Fig. 4.14. The flowchart specifying functioning of vehicle automatic emergency braking system

4.4.2 Vehicle Automatic Emergency Braking Module

The automatic emergency braking is one of the very important features for the vehicle active safety. Fig. 4.15 shows the block input (HeadwayTime).

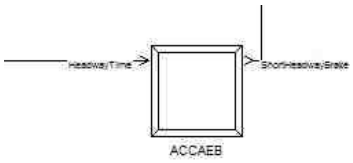


Fig. 4.15. The block inputs for vehicle automatic emergency braking

Below is the approach used to detect the less HeadwayTime to alert driver and transfer this warning to audio block. Eq. 4.4 defines the condition for short HeadwayTime with the vehicle directly ahead (Forward Collision Warning).

$$Output = 1, if(HeadwayTime < 2) \tag{4.4}$$

The internal representation of the AEB block is shown in Fig. 4.16 and braking conditions after it.

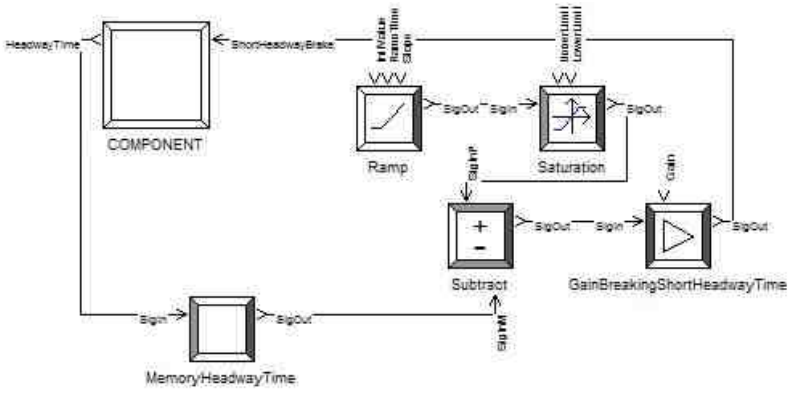


Fig. 4.16. The main components of the automatic emergency braking block

Eq. 4.5 and 4.6 represent the AEB activation criteria for vehicle.

$$StorageMemory = HeadwayTime, if(time! = 0) \tag{4.5}$$

$$Output = HeadwayTime, Checkcondition(HeadwayTime < 1.8) \tag{4.6}$$

After satisfying these conditions the output is inserted in the subtraction block where its difference has been calculated with the [absolute value = 2] then the Gain-BrakingShortHeadwayTime block tries to behave as a proportional controller and make the value tend to HeadwayTime = 2 (i.e. force the braking to be applied until the time to the vehicle ahead becomes 2 seconds). It keeps the magnitude of braking at some specific level if the vehicle ahead is slower and the system is supposed to keep current velocity less than the original velocity of the subject vehicle. Moreover, if the vehicle ahead is at complete stop then the system keeps the braking magnitude for SV at a higher value so that to avoid collision with the vehicle ahead. This process helps in the Stop-and-Go phenomenon for the ACC, which is explained in the latter section.

4.4.3 Pedestrian Automatic Emergency Braking System

The PAEB system is especially designed for protecting pedestrians and it is one of the key features of AEB system. The PAEB system judges the probability of a collision based on the position and relative speed of the vehicle with respect to a pedestrian, and either help the driver to avoid the collision by triggering proper warnings or help to mitigate collision damage by activating devices such as automatic brake assist, automatic steering, and so on [20].

Sensors mounted on vehicle are functional in detecting pedestrian and environmental objects. Although, visibility from the vehicle is limited. It is often the case that it is difficult or some times it is impossible to observe the object from the vehicle itself. Due to the range limitations of PAEB systems and speed variations for

vehicle sensory data processing, PAEB system cannot detect the actual conditions some times or other times there may be not enough time to prevent the potential crash [21]. For example, the potential collision with pedestrians is unavoidable if the vehicle speed is too high or the pedestrian enters in the path of the vehicle too quickly [21]. Moreover, there is another situation when the PAEB cannot detect the pedestrian when it is behind the obstacles, PAEB might not have enough time to react, one other variation role is played by weather or ambient circumstances [21].

The basic system working for the PAEB is shown in Fig. 4.17 flowchart.

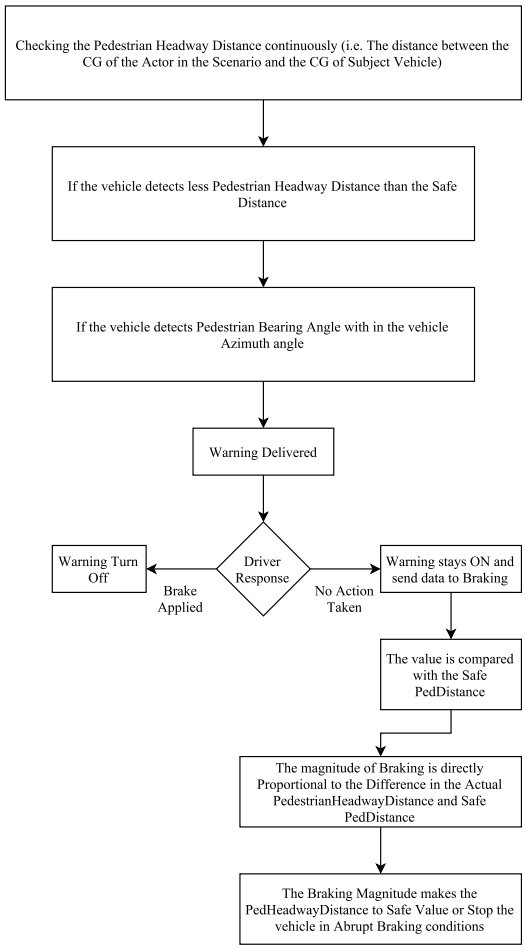


Fig. 4.17. The flowchart specifying basic pedestrian automatic emergency braking system

The algorithmic approach has been shown in Fig. 4.18

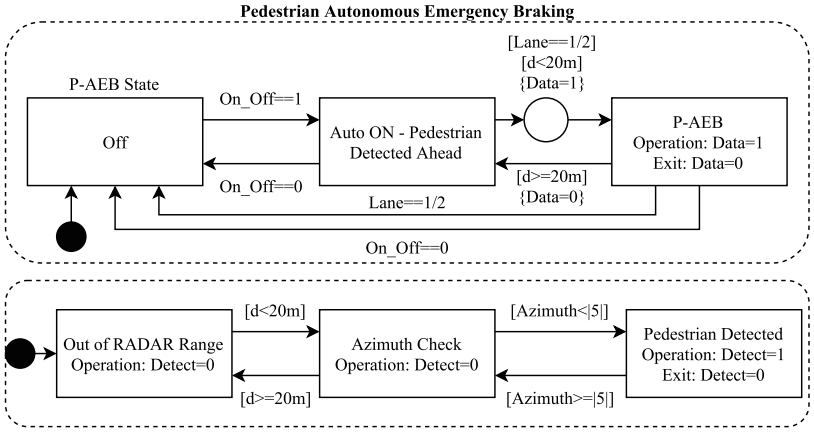


Fig. 4.18. The state flow for pedestrian automatic emergency braking system in the model

4.4.4 Pedestrian Automatic Emergency Braking Module

The main block for PAEB is mentioned in Fig. 4.19, which will clearly show the inputs entering the block. The internal approach for the PAEB block is shown in Fig. 4.20.

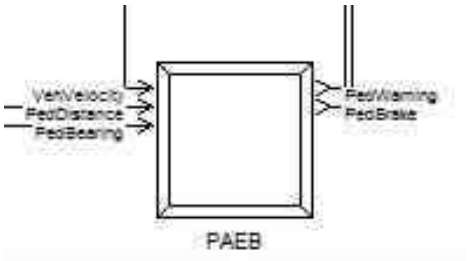


Fig. 4.19. The block inputs for pedestrian automatic emergency braking

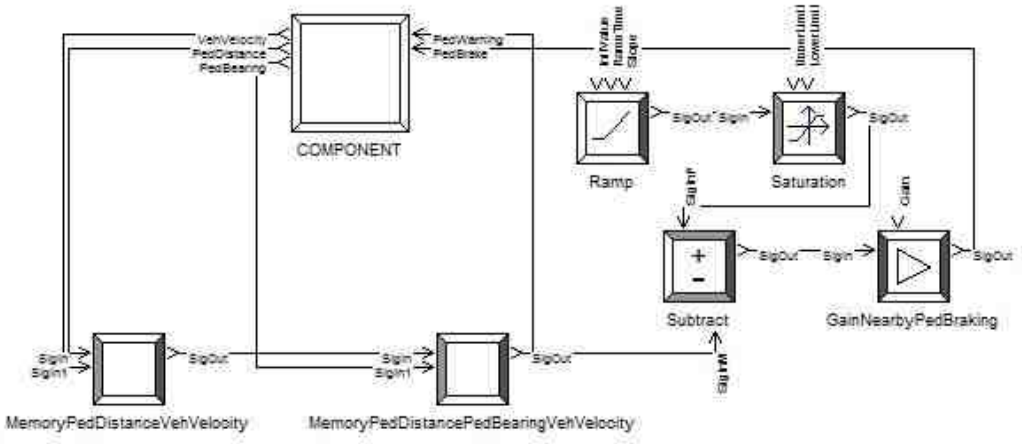


Fig. 4.20. Main components inside the Pedestrian AEB block

$$Output = 1, if(PedDistance < 20) and (PedBearing < |5|) \tag{4.7}$$

$$StorageMemory = PedDistance, if(time! = 0) \tag{4.8}$$

$$Output = PedDistance, checkcondition(PedBearing < |5|) \tag{4.9}$$

The equations are related with the condition to be followed for the particular function. Equation 4.7 defines the condition for the output of the PedWarning to be delivered to the driver before the imminent collision with the pedestrian. Furthermore, equation 4.8 and 4.9 describes the condition to be used for the calculation of the braking force to be transferred to the dynamics of the vehicle. The *MemoryPedDistanceVehVelocity* component stores the PedDistance for each distance and the *MemoryPedDistancePedBearingVehVelocity* component stores the PedBearing as well and if the PedBearing comes into the satisfactory limit it gives the output of the PedDistance which is compared with the value of 20 and their difference is moved to the gain block. The component *GainNearbyPedBraking* has a high gain value which is the most suitable for the abrupt high braking magnitude at such less distance. The values has been calculated by using fine tuning with hit and trial method. Now, there

is also a very important constraint to consider in this situation. The maximum value for speed of the subject vehicle is 40mph because above this speed the PAEB doesn't work due to the following reasons:

1. The braking force is too high which is unsafe for the driver.
2. The pedestrian detection with extremely long distance through a very high range radar is not practically possible.

Moreover, there is also a very important variable named as VehVelocity present as the *MemoryPedDistanceVehVelocity* component input. The purpose it serves is when the vehicle comes to a complete stop by detecting the pedestrian and making itself stop. Now, vehicle has to keep at stop until the pedestrian doesn't move away from front of the vehicle. This variable doesn't let driver to accelerate until the pedestrian is present and helps in keep applying the brake when velocity is more than 1 mph.

4.4.5 Simulation Results and Discussion

Vehicle Automatic Emergency Braking

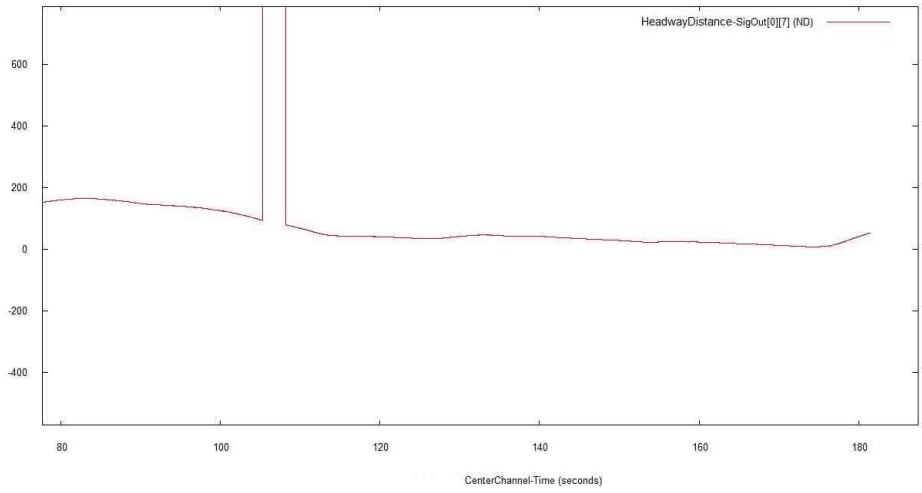


Fig. 4.21. Plot representing the value for headway distance from V to SV

The plot above showing the reduction in the headway distance with time as the vehicle ahead of subject vehicle approaching near it. There is peak in the middle which happens when there is no vehicle in front of the SV or the vehicle in front is very far.

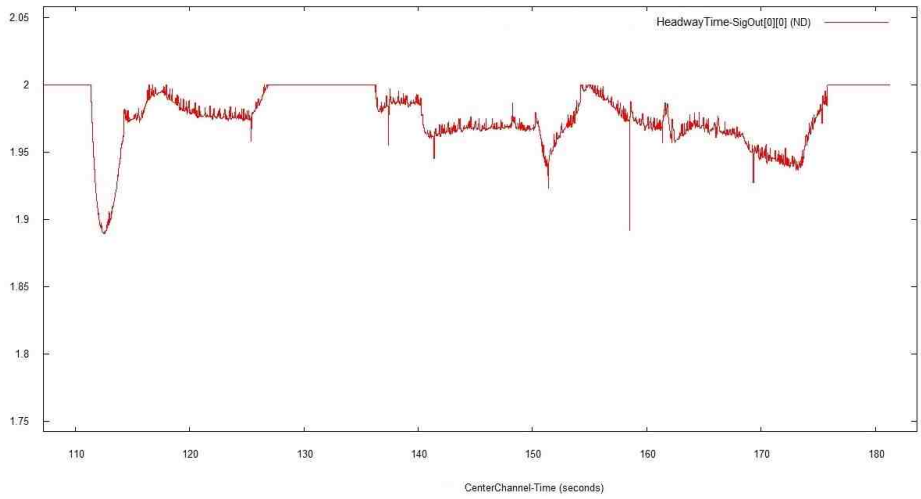


Fig. 4.22. Plot representing the value for output of the headway time

The plot above shows the exact output of memory block when the headway time is less than 2 seconds. This headway time behavior is important because it is required in order to calculate the braking magnitude accurately.

Next plot will show the braking force pattern (the scale should be observed in order to understand the difference between headway time and braking magnitude curves).

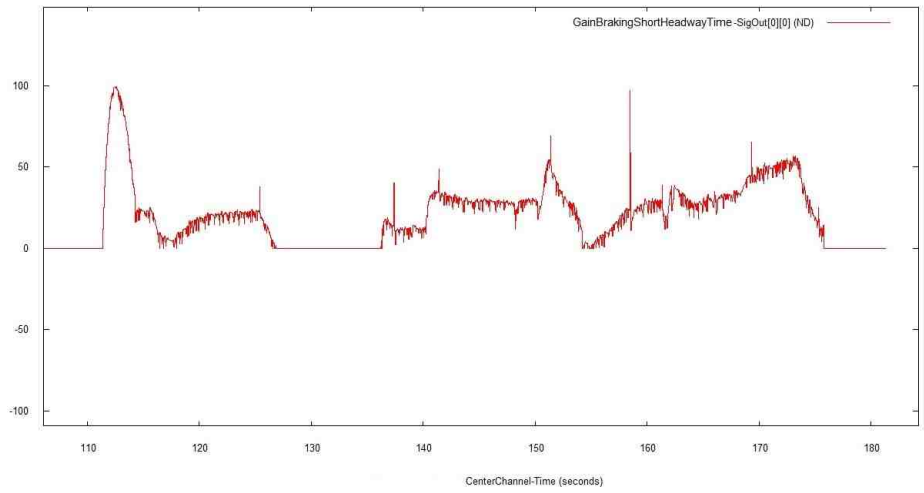


Fig. 4.23. Plot showing the value for braking when Headway Time $<$ 2s

The plot of Fig. 4.23 showing the value of braking magnitude in order to maintain headway time *(HeadwayTime = 2)*. A very clear relation can be observed between Headway Time and braking magnitude by the examination of these two plots above.

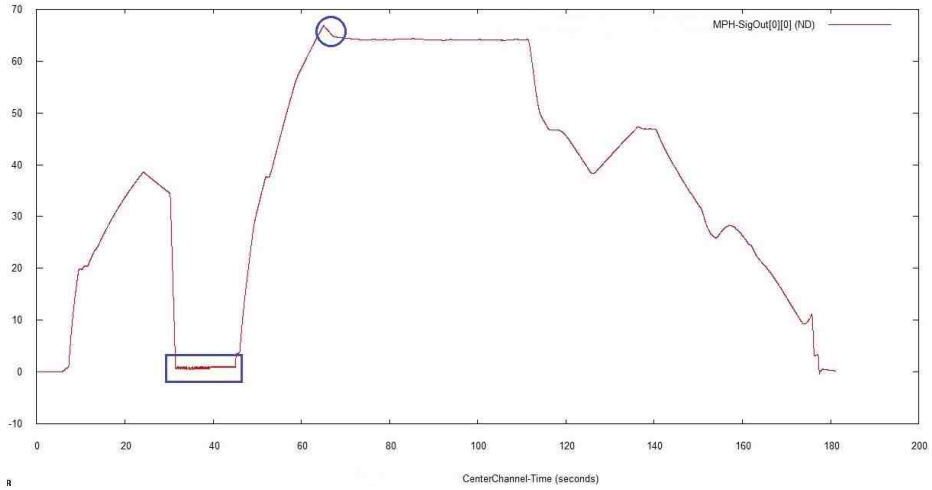


Fig. 4.24. Plot representing the value for variations in set point cruise control velocity

The straight line after the circle indicated in the plot in Fig. 4.24, has the variations in the set cruise control velocity. For this test, cruise control velocity has been set to 65mph and reduction in the velocity can be observed on the basis of short headway time as ACC and AEB both work on the preventive strategies. There are peaks in the slowing down partition because the ambient traffic is programmed to show the slowdown and speedup behavior during simulation (for checking ACC efficiency). The square indication on the plot has been explained later in the PAEB system plots.

Pedestrian Automatic Emergency Braking

Plotting results for PAEB are shown below.

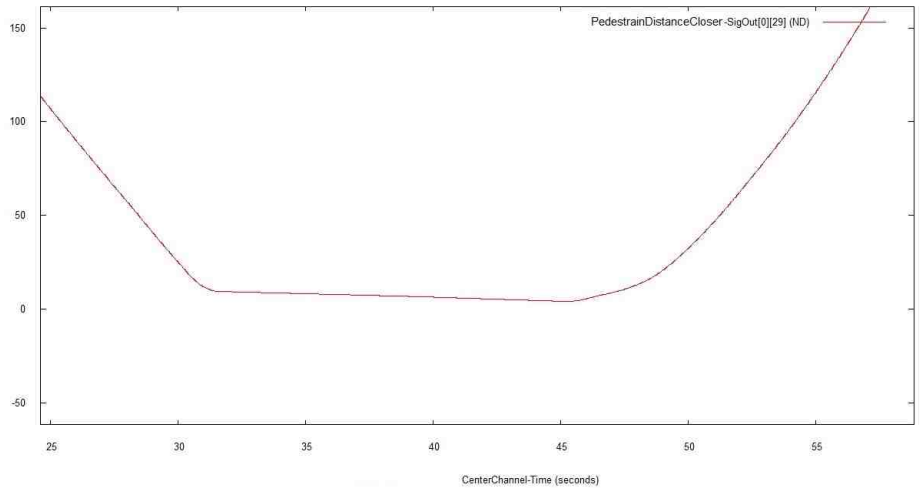


Fig. 4.25. Plot representing the value for pedestrian distance to SV

Plot in Fig. 4.25 is showing the PedDistance to SV. A small partition has been taken during which the vehicle was very close to pedestrian. The distance from CG of SV to CG of pedestrian can never be zero that's why distance starts increasing after a certain value, after the pedestrian has moved away from the front of vehicle and pedestrian collision has been avoided.

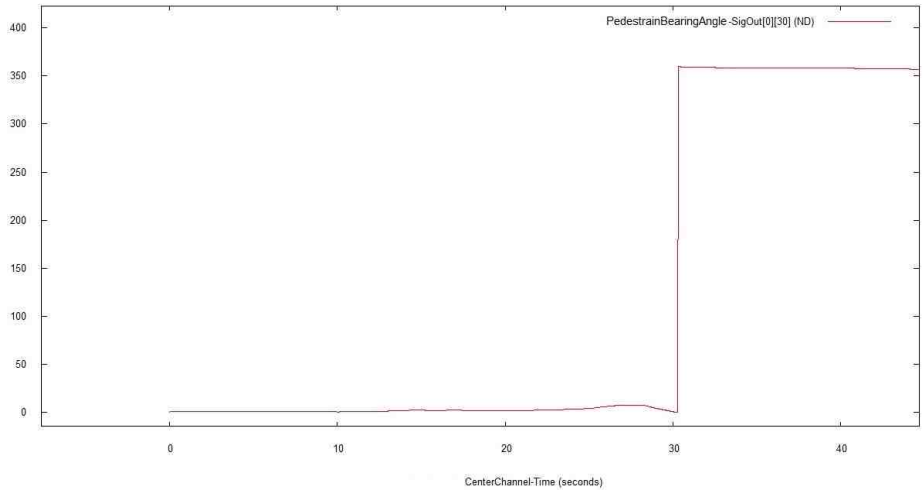
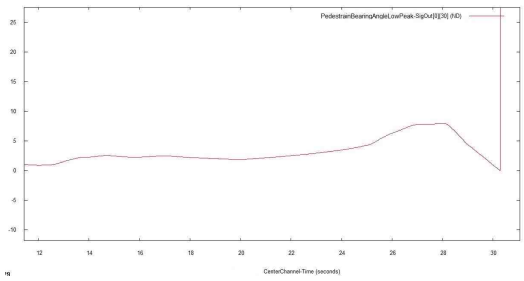
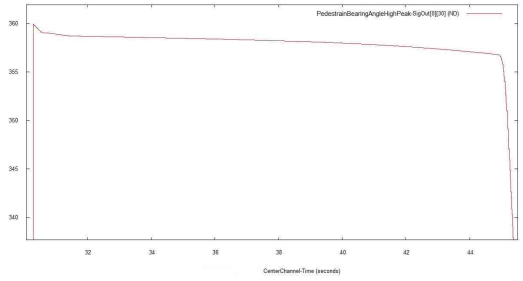


Fig. 4.26. Plot representing the value for pedestrian bearing angle to SV azimuth

Plot representing the PedBearing angle with vehicle azimuth. (0 at directly straight position on the vehicle front bumper). The limits to observe the PedBearing angle are (5 > PedBearing > 355).



(a) Lower end for the pedestrian azimuth angle to vehicle



(b) Higher end for the pedestrian azimuth angle to vehicle

Fig. 4.27. Pedestrian bearing angle with vehicle (low and high end).

These plots show the enlarged view of two ends which come into the PedBearing limits and show that pedestrian is moving in front of the vehicle when it has been detected.

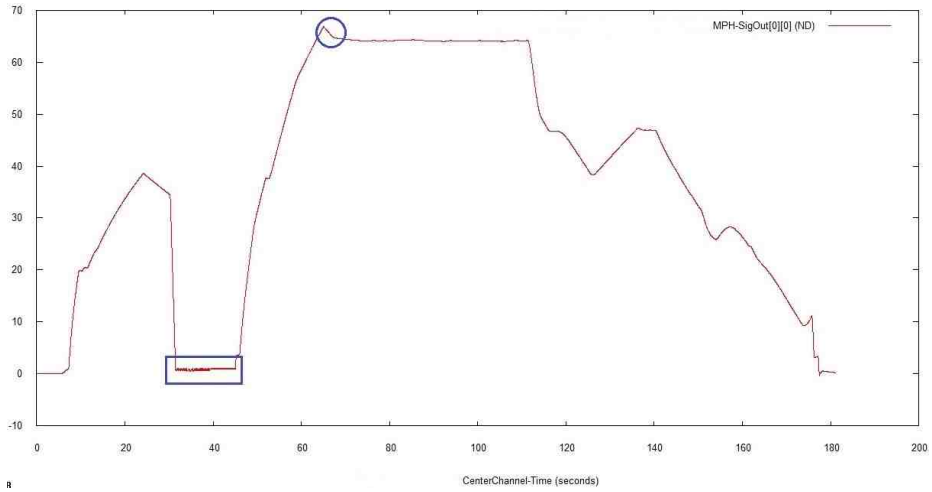
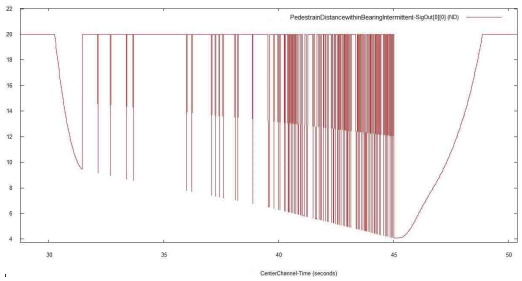
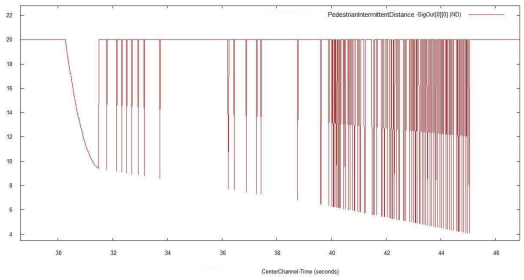


Fig. 4.28. Plot representing the value for variations in set point cruise control velocity

The square in plot above in Fig. 4.28 showing the behavior of set cruise velocity or the normal velocity (when the CC is not ON) at the instant of close pedestrian detection. Before the square, abrupt decrease in vehicle velocity from (40mph) with auto braking can be observed as it detects the pedestrian from distance. Further explanation is given in next plots.



(a) Close pedestrian distance detection



(b) Close pedestrian distance with bearing detection

Fig. 4.29. Pedestrian distance and bearing angle detection phenomenon

These plots have pedestrian distance detection as shown in Fig. 4.29(a) and PedDistance detection along with PedBearing as shown in Fig. 4.29(b). PedDistance is observed very quickly at every instant when the vehicle is close to pedestrian. These sudden peaks show the detection of close pedestrian very quickly when the driver is close to pedestrian and vehicle is stopped but driver is still trying to accelerate the vehicle, but the PAEB is not letting him/her to accelerate because the pedestrian is still in front of the vehicle and has not moved away.

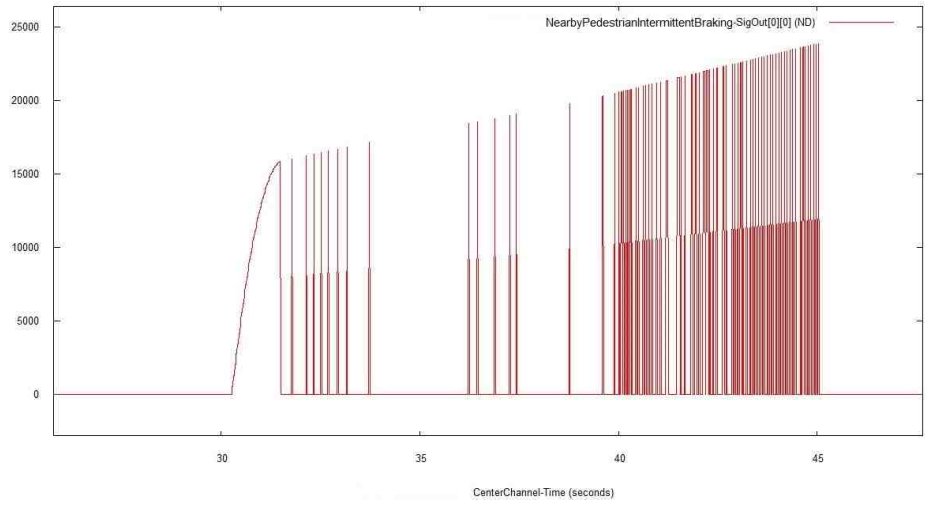


Fig. 4.30. Plot representing the value for magnitude of braking when nearby pedestrian is detected

The plot in Fig. 4.30 is showing the braking gain upon the detection of nearby pedestrian. It has a clear relationship with the close PedDistance detection plot. As the PedBraking magnitude increases and braking happens more often when the vehicle to pedestrian distance reduces.

4.5 Adaptive Cruise Control System with Stop-and-Go

ACC is the cumulative system of all the features described above. It is integrated system of lane keeping and pre-crash systems as well. The ACC integrated in this

model is advanced in a sense that it contains the stop and go technology as well because it can make your vehicle to complete stop as the preceding vehicle slows down to a complete stop.

In advancement to ACC system, a velocity controlling method of a subject vehicle for object detection might be installed. An object detecting device might be capable of detecting several target objects simultaneously. A separate control order is generated for each detected target object for influencing engine controls and brake controls of the SV to control its velocity [22]. At least one control signal based on the selected control orders is then sent to one of the engine controls and the brake controls of Subject Vehicle to control its velocity in accordance with the selected control order [22].

An adaptive cruise control system with stop-and-go capability has been invented to reduce the driver's load in expressways or in traffic jams. This section presents its effectiveness with the help of experimental results. The basic approach used for ACC is the additive effect of the systems described in earlier sections, algorithmic approach is shown in Fig. 4.31.

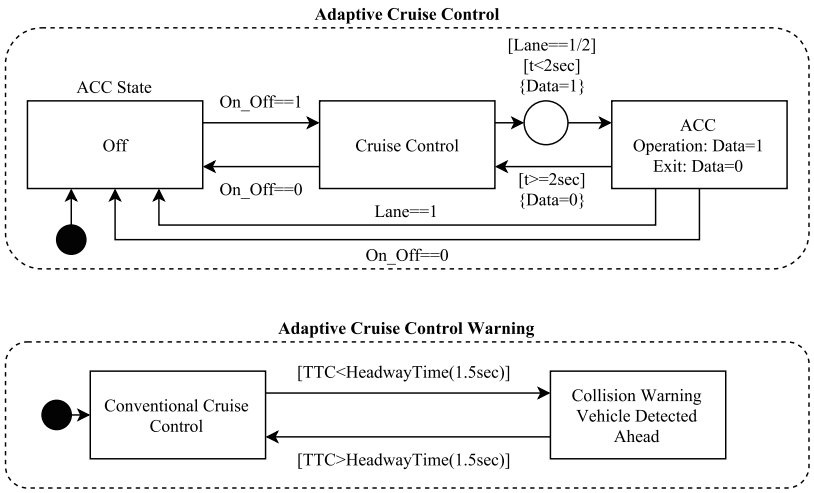


Fig. 4.31. The state flow for the ACC system in the simulation model

Time to Collision (Eq. 4.10) plays important role in the development of the ACC.

$$TTC = \frac{HeadwayDistance}{Velocity(HeadwayVehicle) + RelativeVelocity} \tag{4.10}$$

Following this equation is only basics but there are many enhancements required for making the ACC work as a normal vehicle ACC system.

4.5.1 Cruise Control Module

The ACC block and its interior are mentioned in Fig. 4.32 and 4.33, respectively. The description is provided afterwards.

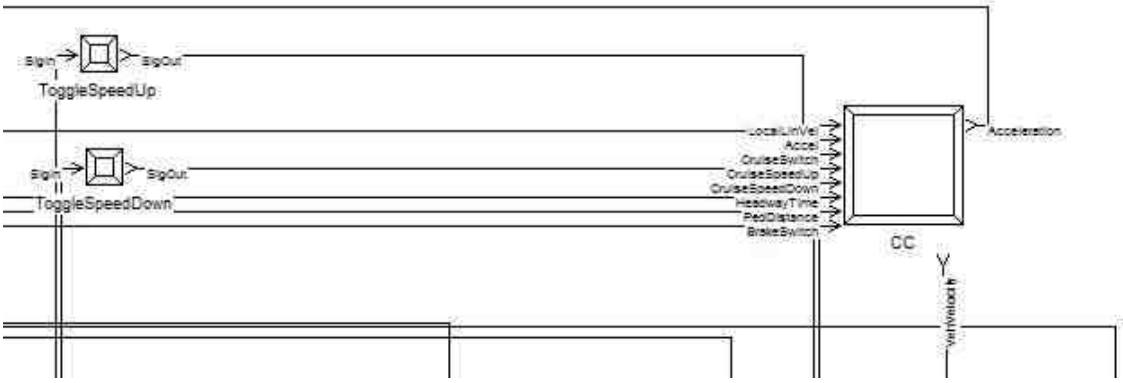


Fig. 4.32. Input variables for the conventional cruise control block

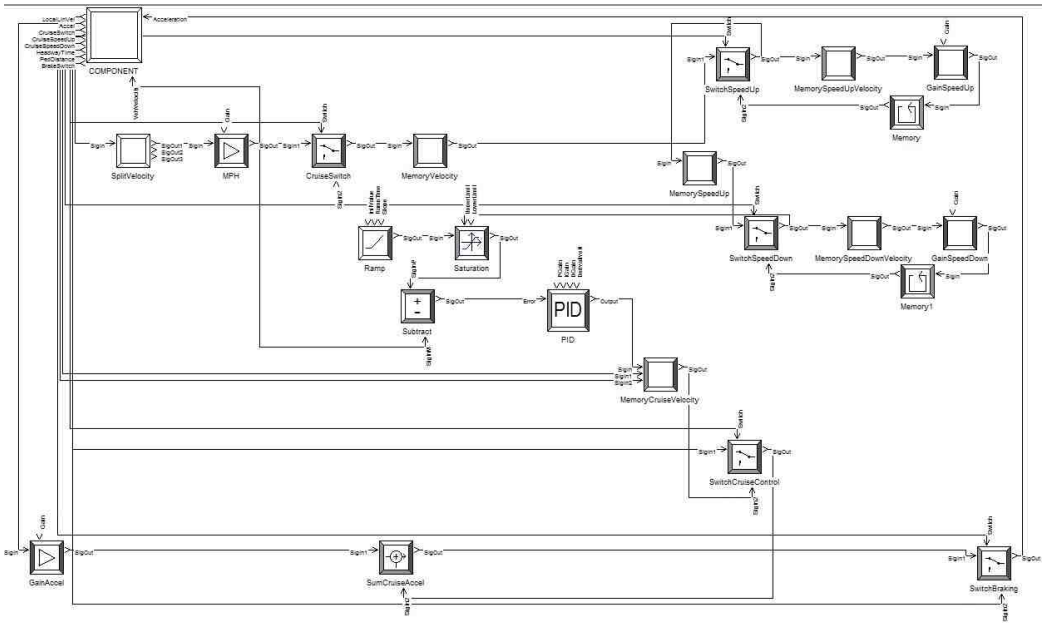


Fig. 4.33. Main components inside the conventional cruise control block

4.5.2 Explanation for Adaptive Cruise Control

Following are the important conditions to be fulfilled for the purpose:

$$Output = 1, if(HeadwayTime < 1.5) \tag{4.11}$$

$$MemoryVelocity = VehVelocity, if(input! = 0) \tag{4.12}$$

$$VelocityOutput = StorageVelocity, if(input == 0) \tag{4.13}$$

$$MemoryCruiseVelocity = CruiseVelocity, Checkcondition(CruiseVelocity! = 0) \tag{4.14}$$

$$CruiseVelocityOutput = MemoryCruiseVelocity, Checkif(HeadwayTime > 2) \tag{4.15}$$

$$CruiseVelocityOutput = MemoryCruiseVelocity, Checkif(PedDistance > 10) \tag{4.16}$$

Equation 4.11 serves as the condition for the output of the audible signal in case of short HeadwayTime and the vehicle ahead is closer than the desired distance. Equation 4.12 serves as the condition used for saving the velocity of the subject vehicle in the memory block component until the cruise control system is turned ON and then Equation 4.13 performs as the condition to turn system ON and supply this velocity as output to the *Speed Up* and *Speed Down* blocks.

The velocity first passes through *Speed Up* block which serve as the 1 mph increment block for the cruise control system. It works in a way that it increase the velocity by 1 mph as many times the button will be pressed on the joystick and keep saving the new incremented value. Afterwards, the velocity passes through the *Speed Down* block which serve as the 1 mph decrement block for CC. It works in a way that it reduces the velocity by 1 mph from the newly injected velocity as many times as the button is pressed on the joystick and keep saving the new decremented value.

Equation 4.14 defines the condition for saving the output of the PID controller (CruiseVelocity) in the appropriate memory block. Furthermore, equation 4.15 and 4.16 delineate two conditions to be fulfilled before giving the output of *MemoryCruiseVelocity* block. HeadwayTime should be more than 2 seconds in order to keep the cruise control system ON because braking system is working with ACC on the same time and as the HeadwayTime approaches to a value less than 2 seconds the AEB starts applying brakes. On the braking instant, acceleration should be negligible for this time period in order to make the AEB work most efficiently. Similarly, PedDistance should be more than 10 meters in order to keep the CC system ON because pedestrian emergency braking works along with ACC and as the PedDistance approaches to a value less than 10 meter the PAEB starts applying brakes. Similarly, acceleration should also be negligible for this time period in order to make the PAEB system work most efficiently and effectively. The output of SwitchCruiseControl is the parameter which transfers toward the SumCruiseAccel block, the objective of this internal summer is to provide an opportunity to the driver to accelerate the vehicle in the middle when the CC is ON but driver wants to accelerate according to the various

traffic behavior circumstances. The GainAccel component increases the magnitude of acceleration with specific gain value in order to nullify the effect of PID (for CC component) for a small instance of time because PID tends to make the velocity at the set point defined by cruise control. Further, comes the SwitchBraking component whose function is to disengage the cruise control system when the brake is applied because it's a normal phenomenon in the heavy traffic. But for the proposed automated vehicle it's unusual as it is equipped with Stop-and-Go feature (an advancement to adaptive cruise control system). Nevertheless, there could be some situations where very abrupt braking is required manually by the driver to prevent or ameliorate the collision. The output of SwitchBraking component is moved back into the vehicle dynamics in the form of acceleration in order to provide the desired acceleration to maintain the appropriate velocity.

The complete summer arrangement for the blocks as per their time of action requirement has been shown earlier in chapter 3 of this study.

4.5.3 Simulation Results and Discussion

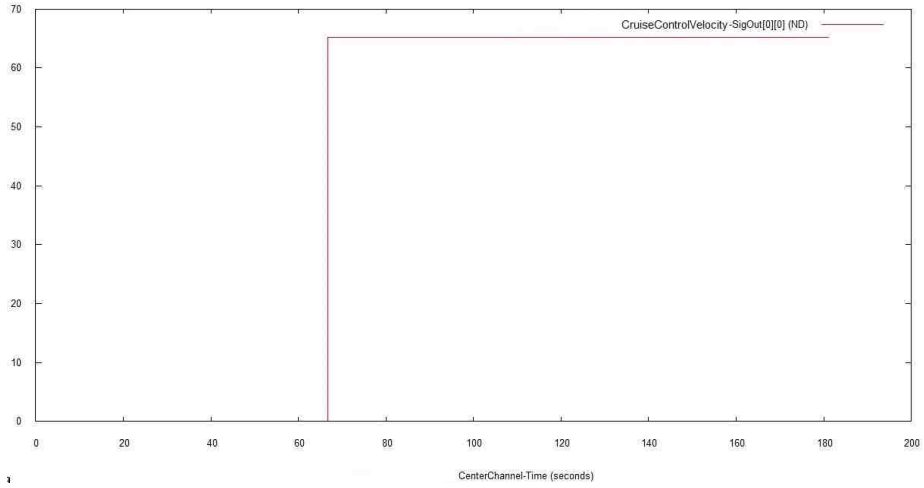


Fig. 4.34. Plot representing the value for set point cruise control velocity

Plot in Fig. 4.34 is showing the output of block which stores the velocity of SV and gives a steady output of the velocity to system only when the CC is turned ON.

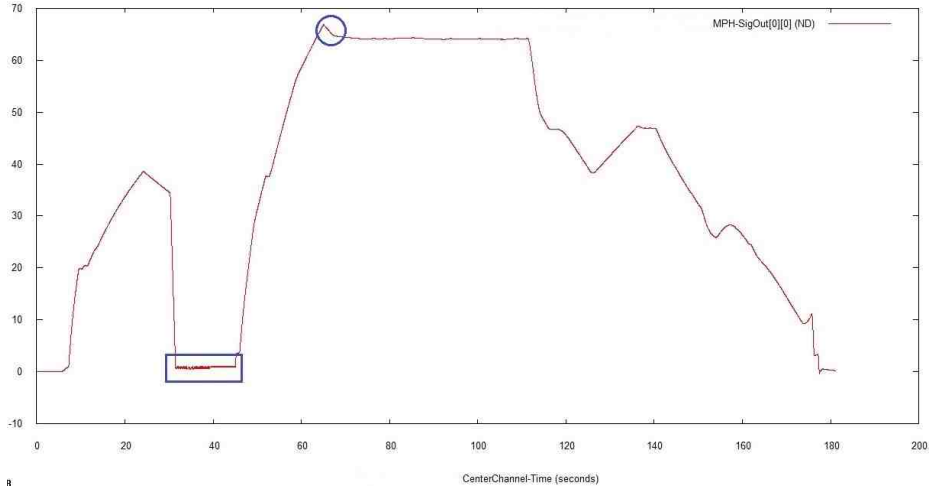


Fig. 4.35. Plot representing the value for variations in set point cruise control velocity

This plot has been explained for other sections earlier in this research work, but the straight line after the circular indication on the plot shows the normal stability and output of continuous cruise control velocity under normal traffic conditions when vehicle doesn't need to slow down due to alteration in ambient traffic behavior or actor movement on road.

5. AUTOMATIC EMERGENCY PULL-OVER SYSTEM

This chapter will describe the design and implementation of vehicle automatic emergency pullover (AEP) strategy, using active safety systems (described in the earlier chapter).

5.1 Automatic Emergency Pull-over Module

Automatic emergency pullover system consists of three internal parts namely; AEP Velocity Module, AEP Braking Module, AEP Steering Module. The main component block for AEP and the internal architecture of AEP block is shown in Figs. 5.1 and 5.2, respectively.

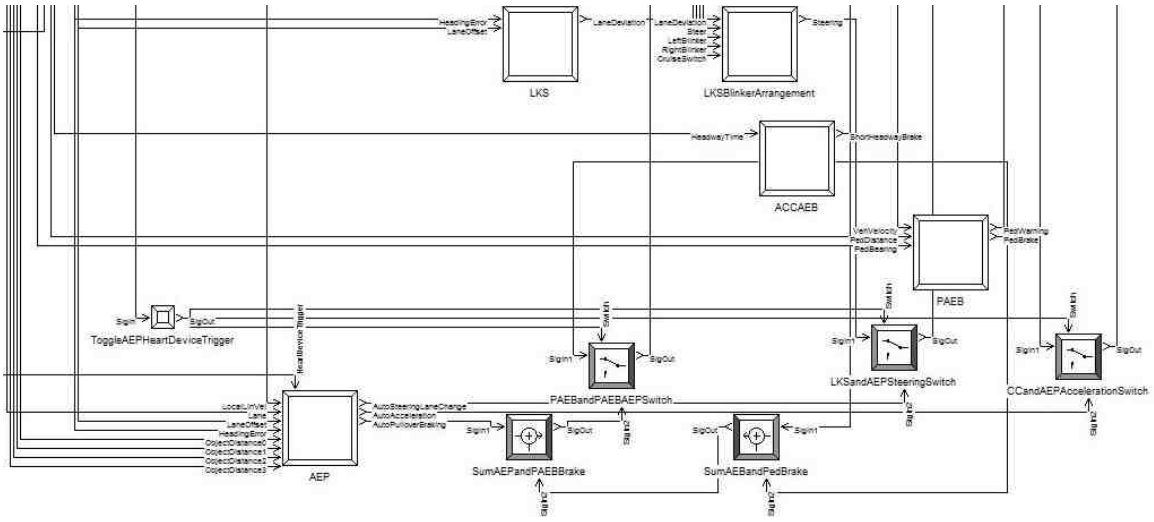


Fig. 5.1. The main component block for AEP and its inputs with correlation to other active safety systems

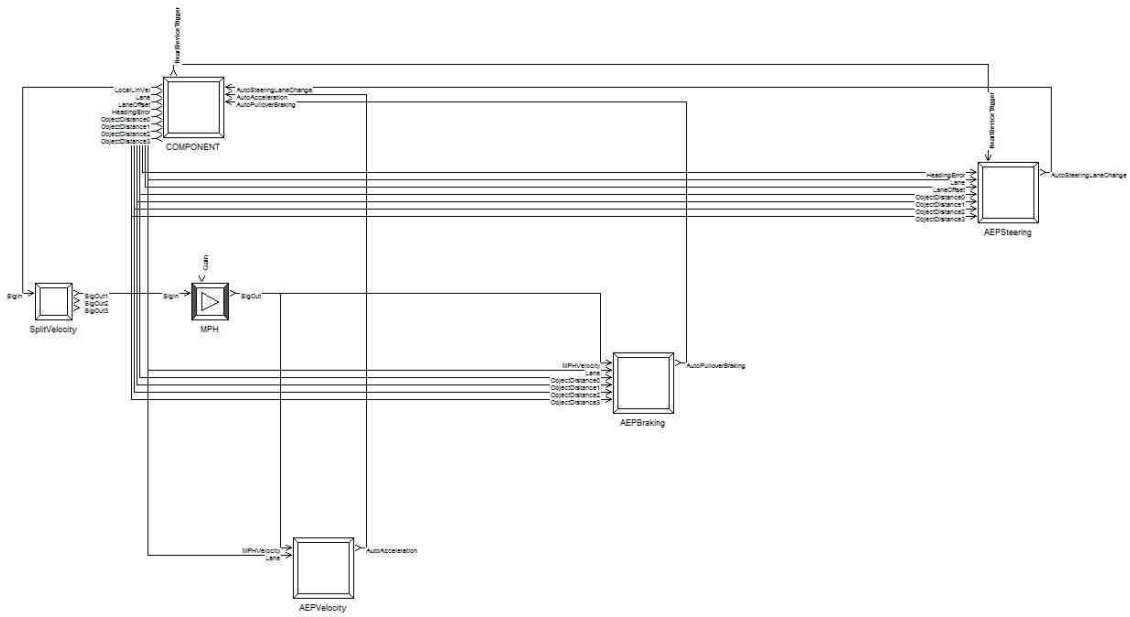


Fig. 5.2. Internal component representation and arrangement for AEP block

The AEP is a system made with the mutual correlation of the other systems as well. The sum blocks in Fig. 5.1 shows that AEB and Pedestrian-AEB are included in the AEP braking system for roadside safe lane braking. AEB and P-AEB are used when the vehicle is still on the road after the health device trigger has been received by the AEP system. It can be clearly seen in Fig. 5.1 that there are switches to toggle between the braking and AEP braking, acceleration and AEP acceleration, steering and AEP steering. Fig. 5.2 is showing that AEP systems use the vehicle velocity in miles per hour for its calculation purpose.

Automatic Emergency Pull-Over Blinker Arrangement

AEP need special blinker arrangement in the scenario when vehicle is autonomous after the driver unconscious behavior. Figs. 5.3 and 5.4 will show the blinker arrangement location in the model and the blinker component internal system, respectively.

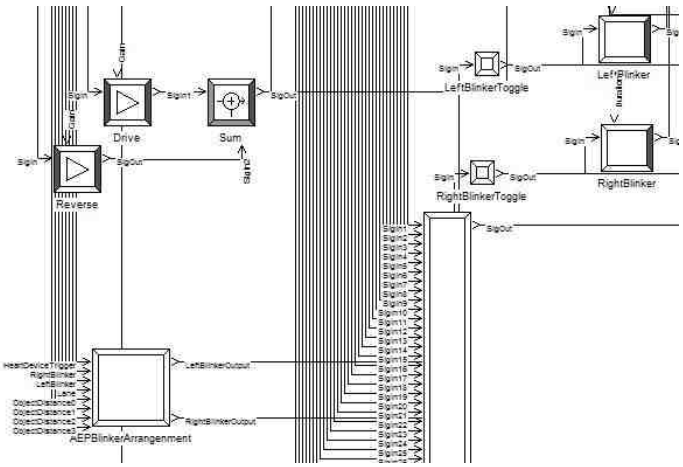


Fig. 5.3. AEP blinker component located in the FullSim model

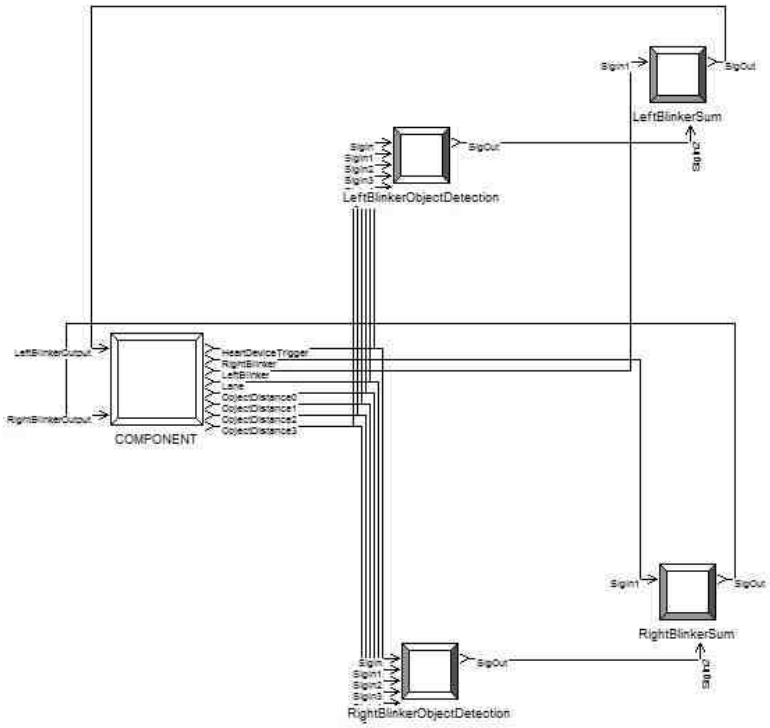


Fig. 5.4. Internal component representation of the AEP blinker block

AEP blinker arrangement receive a trigger signal from the health monitoring device, then it monitors the lane for the sake of understanding the position of subject vehicle on the road and it helps in identifying that on which side is the nearest roadside lane available (either right or left), after identification of present vehicle lane it gives the blinker to the nearest roadside safety lane direction. On the same time, blinker should also be monitoring the presence of roadside objects and whether there is safe distance from the roadside objects or not. For roadside object distance purpose the blinker arrangement calculate the distance and then analyze which side blinker it should give for a safe roadside maneuver (either right or left).

However, the manual blinker through lever remains in the hand of driver. If driver is in a condition to put blinker ON by himself/herself. S/he can do so but if driver is incapable then vehicle give appropriate blinker by itself. Additionally, head lights and tail lights start blinking and perform a function of hazard lights after the AEP system has been triggered. It helps traffic in getting alert about the hazard situation of subject vehicle at night time.

5.2 AEP System Design based on Active Safety features

As mentioned earlier AEP consists of three separate parts.

- AEP Velocity Control Module
- AEP Braking Control Module
- AEP Steering Control Module

5.2.1 AEP Velocity Module

AEP system needs automatic velocity calculation after the vehicle is in the control of system. The explanation is given after Fig. 5.5.

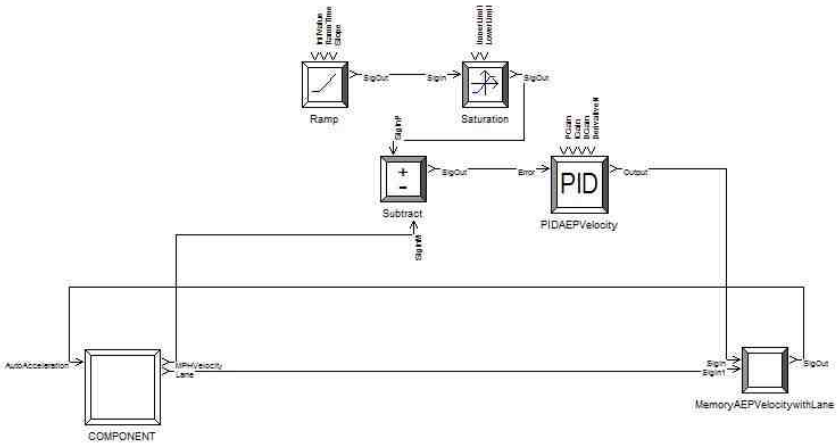


Fig. 5.5. Internal component representation of the AEP velocity module

After AEP got triggered the current MPH (miles per hour) velocity of vehicle is compared with a slow velocity (i.e. 20mph) then this occurrence is settled with a finely tuned PID block. *MemoryAEPVelocity* component in Fig. 5.5 monitors the lane in which the vehicle is present on the time of trigger. If the vehicle has a velocity less than 20 mph and is on the road lane, the system provides acceleration to the vehicle to achieve a velocity of 20 mph until the vehicle reaches the roadside safe lane (Eq. 5.1).

$$Velocity = 20, if(Lane! = 3|| - 1) \tag{5.1}$$

Lane numbers are given in Chapter 3. After reaching the roadside safe location it stops accelerating the vehicle and then braking algorithm comes intact, which is explained in the next section.

5.2.2 AEP Braking Module

The AEP braking plays a critical role because of its significance in such a situation when driver is incapable of control. The explanation is provided after Fig. 5.6.

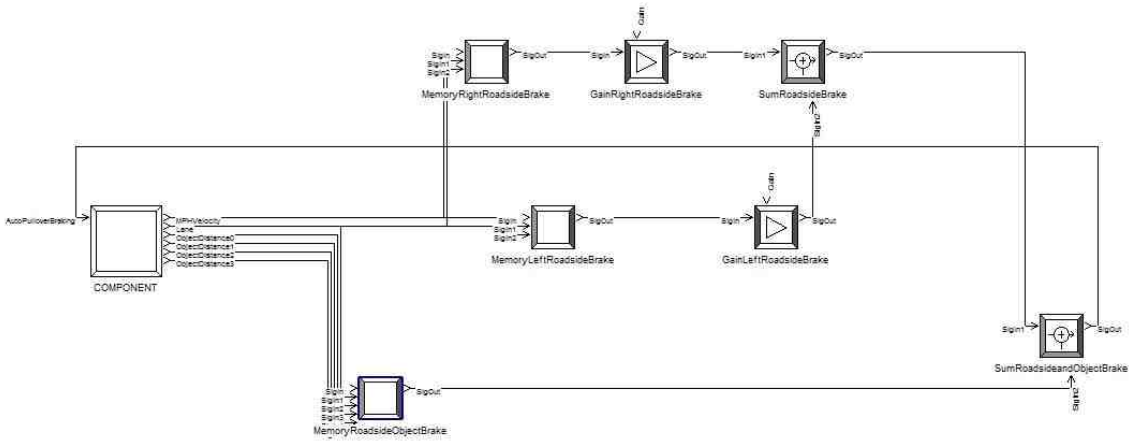


Fig. 5.6. Internal component representation of the AEP braking module

AEP braking system controls the vehicle braking after receiving trigger signal by monitoring the vehicle velocity at the instance of trigger. It also monitor the lane number at that instant along with distance to roadside objects. *Memory left/right roadside brake* component in Fig. 5.6 are monitoring the lane and velocity for the vehicle. They provide a specific braking force for a particular threshold of velocity after the trigger signal, which helps in deceleration of vehicle quickly. When the vehicle reaches roadside and attain a velocity of 10 mph then these blocks provide a high magnitude of braking force, in order to bring vehicle to a complete stop at roadside lane (typically high magnitude is applied just after the vehicle become straight on the roadside lane and have a velocity of less than 10 mph).

$$Brake = 160Nm, if(Velocity < 10mph) \tag{5.2}$$

$$Brake = 10Nm, if(Velocity > 10mph) \tag{5.3}$$

The *MemoryRoadsideObjectBrake* component in Fig. 5.6 is providing a high braking magnitude in such a situation when vehicle has lot of objects on the road side and they are very near to vehicle. Therefore, in such condition the vehicle doesn't have a

safe lane to pullover and it has to stop on the current lane to prevent collision with any other vehicle (it happens after the trigger signal has been received and it is a form of worst case road scenario).

$$Brake = 160Nm, if(DistancetoObject < 3m) \tag{5.4}$$

5.2.3 AEP Steering Module

The AEP steering is the most crucial part for such an auto driving situation. The explanation is provided after Fig. 5.7.

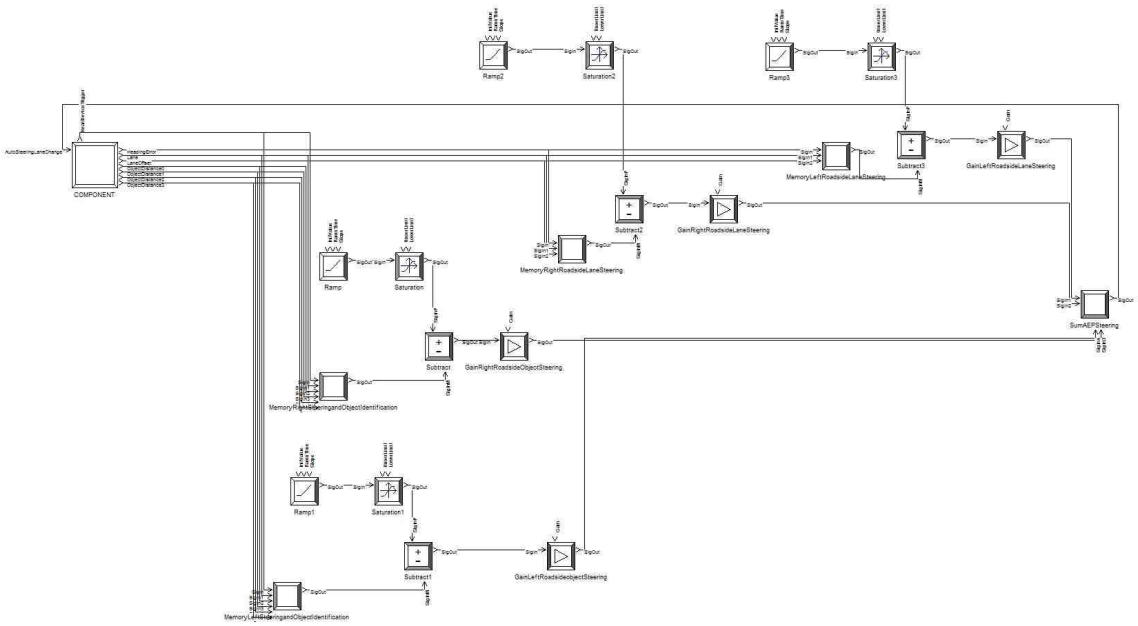


Fig. 5.7. Internal component representation of the AEP steering module

AEP steering mainly consists of four main controllers for four different maneuvers. Namely, right steering towards roadside, left steering towards roadside, steering in right roadside, steering in left roadside. *Right/Left Steering towards roadside*, both take the trigger from the health device. They monitor the distance from the roadside objects on that particular instant. Afterwards, according to the current lane of the

vehicle (the lane in which vehicle is present at the time of trigger), these blocks provide steering magnitude in the appropriate direction of roadside (either left or right). Due to complications, the control logic is explained through algorithm below.

Algorithm 3 AEP Steering

if (Trigger = 1) HealthDevice Trigger
if (Lane = 1)
Check the Distance from Roadside Objects
if (distance>8m)
Lane=-1;
else if (distance<8m)
Lane=3;
end if
if (Lane = 2)
Check the Distance from Roadside Objects
if (distance>8m)
Lane=3;
else if (distance<8m)
Lane=-1;
end if
end if

Further, when vehicle reaches the roadside safe lane, the *Right/Left roadside steering* blocks monitor the LaneOffset in roadside lane and provide sufficient steering torque in order to make vehicle straight in roadside lane and make HeadingError negligible (Eq. 5.5 and 5.6).

$$HeadingError = 0, if(Lane = 3\&\&LaneOffset = 0) \tag{5.5}$$

$$HeadingError = 3.1416, if(Lane = -1\&\&LaneOffset = 1.6) \tag{5.6}$$

Finally, the *SumAEPSteering* component adds-up all the appropriate steering magnitude torques and give an auto steering output to vehicle dynamics.

Note: The distances are always from the CG of vehicle to the CG of other entity (object).

5.3 Simulation Results and Discussion

One of the major benefits for this AEP system is that it can be triggered back and the driver can take the control back from AEP automated driving technique, in the case of false trigger alarm from the health monitoring device due to its malfunctioning.

Although, the implemented AEP system may work for other pullover scenarios but this section will provide the analysis over the automatic emergency pullover resulting plots for two possible cases. Both the cases are taken from a freeway driving situation.

1. **Case 1:** Right or Left Roadside Simple Pullover Scenario. (see Fig. 3.3)
2. **Case 2:** Right or Left Roadside Object (near) Pullover Scenario. (see Fig. 3.5)

Before going through it's necessary to learn the following parameters variation pattern.

1. Lane Number: -1,1,2,3 (left to right on a four lane freeway).
2. Lane Offset: 0 Center of lane; +ve Right; -ve Left (in meters).
3. Heading Error: 0 Straight on the lane; +ve Right; -ve Left (in degrees).
4. Steer Torque: 0 when steer angle in zero; +ve Left; -ve Right (in Newton-meter).

5.3.1 Results for Case 1 Maneuver

Steering

First, the simple pull-over scenario steer action is explained with the help of following figures:

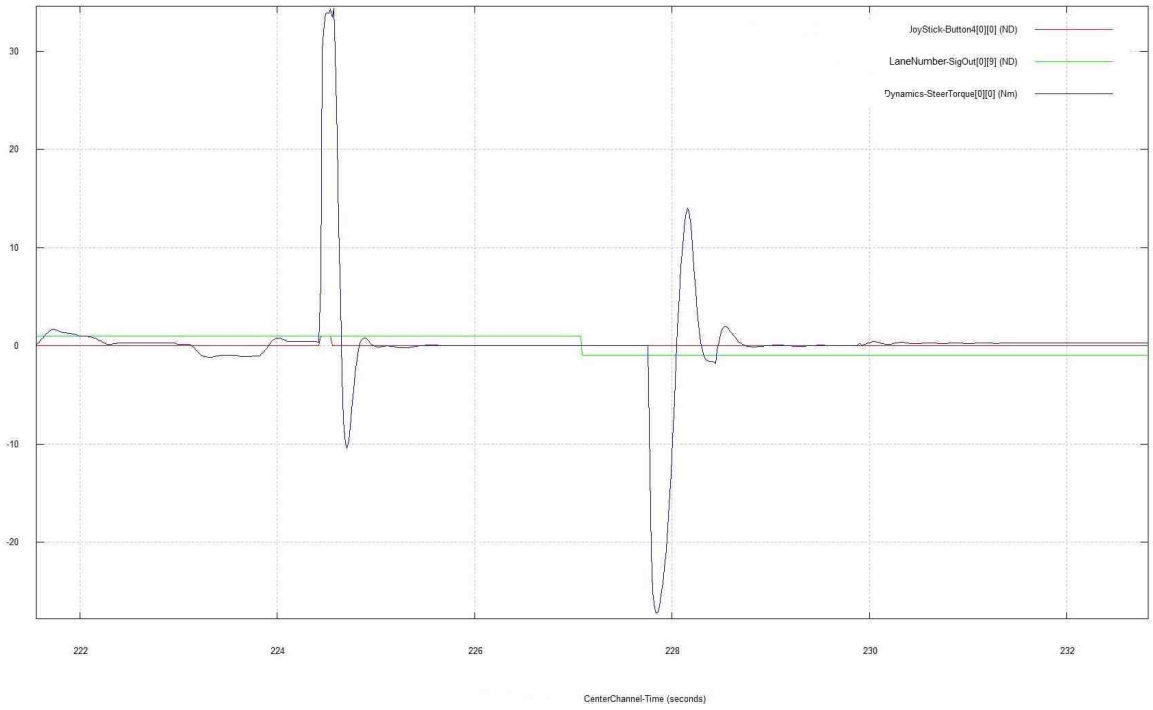


Fig. 5.8. Plot representing the value for variations in LaneNumber and SteerTorque for left roadside pullover

For left roadside pullover plot shown in Fig. 5.8 the green line is showing that vehicle is moving from LaneNumber 1 to LaneNumber -1, after the trigger is received using the button 4 (red line), the SteerTorque (blue line) is raised significantly in positive value and steer vehicle to left side, then it stabilizes itself until vehicle reaches Lane -1 (left roadside). Further, after reaching the Lane -1, the steer is raised in negative value significantly so that it can steer the vehicle on right side to make it straight on the roadside in the center of left safety lane. Now, it stabilizes itself to make the vehicle keep straight.

For right roadside pullover plot shown in Fig. 5.9, the operation is clearly opposite to the left roadside pullover. Vehicle is moving from Lane 2 to Lane 3. Therefore, after trigger the SteerTorque is negative (means steering is on right side). Then, after reaching the Lane 3 there is a nice stability curve for SteerTorque to make vehicle straight on right roadside safety lane.

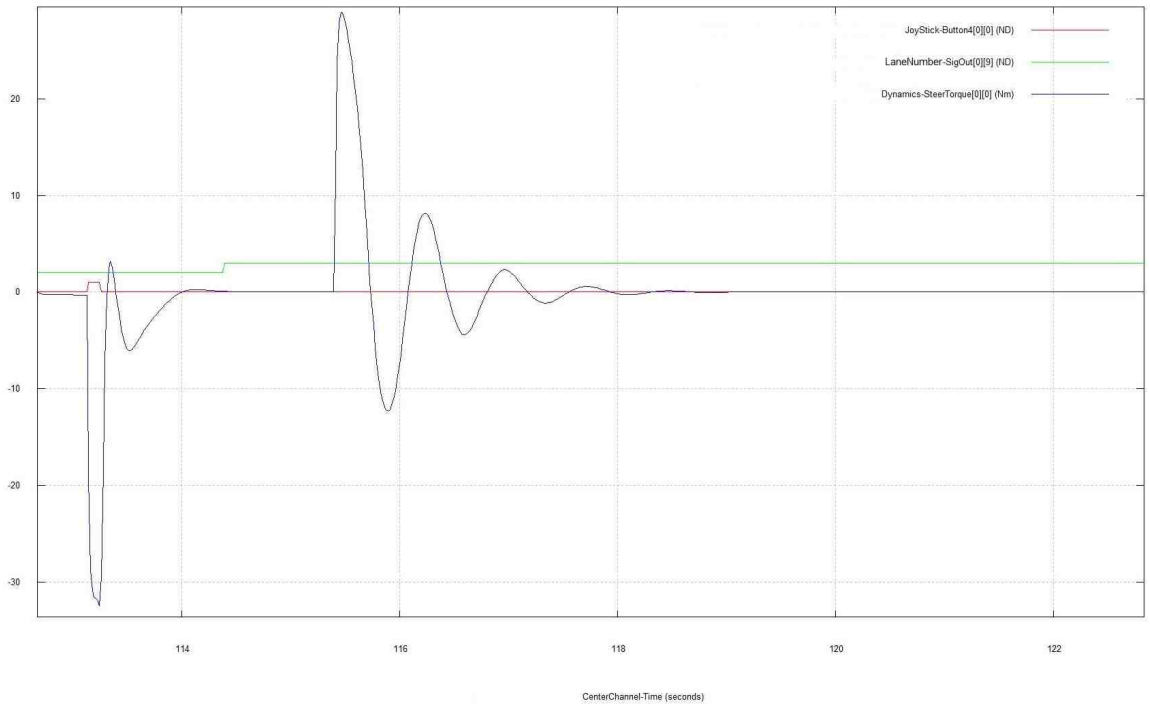


Fig. 5.9. Plot representing the value for variations in LaneNumber and SteerTorque for right roadside pullover

For left and right roadside pullover the steering torque magnitude can be observed approximately at a same level. In left roadside pullover the steering magnitude is 35 Nm after the health device trigger signal, on the plot. In right roadside pullover the steering magnitude is -35 Nm after the health device trigger. The steer torque can be observed in Nm and it has been mentioned earlier in the beginning of results. The reason for specific torque is that it has been extracted from the scenario (i.e. magnitude required to steer from Lane 1 to Lane -1 and required magnitude to steer from Lane 2 to Lane 3 for left and right pullover, respectively).

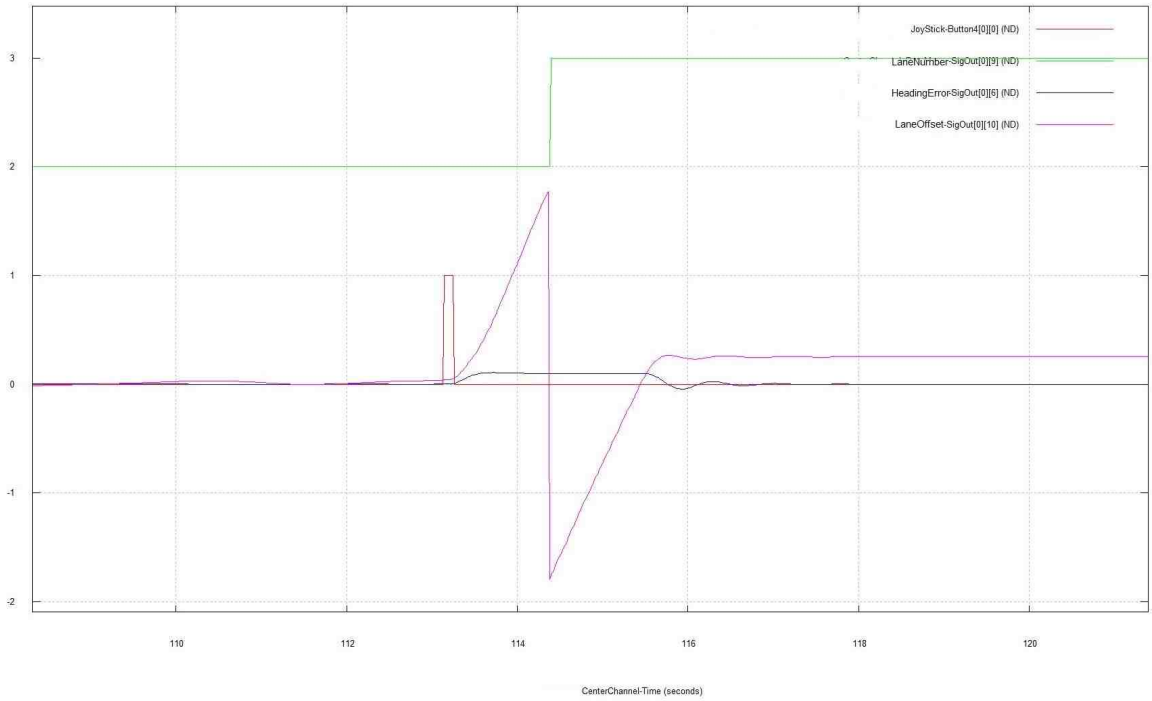


Fig. 5.10. Plot representing the value for variations in LaneNumber and HeadingError and LaneOffset for right roadside pullover

For right roadside pullover plot shown in Fig. 5.10, the operation is further explained (exactly opposite for left side). It can be seen through the plot that after trigger signal the LaneOffset value changes to positive (i.e. Lane 2 to Lane 3 movement) and suddenly after reaching Lane 3, its value is lowest negative because it is the least value for LaneOffset in a typical lane. After reaching the Lane 3, vehicle slowly moves to the center of lane and as the vehicle get a bit of value greater than 0 (i.e. +ve means vehicle is moving now from center of Lane 3 towards right of Lane 3), then the steering torque gives input to make the HeadingError zero abruptly.

Velocity and Braking

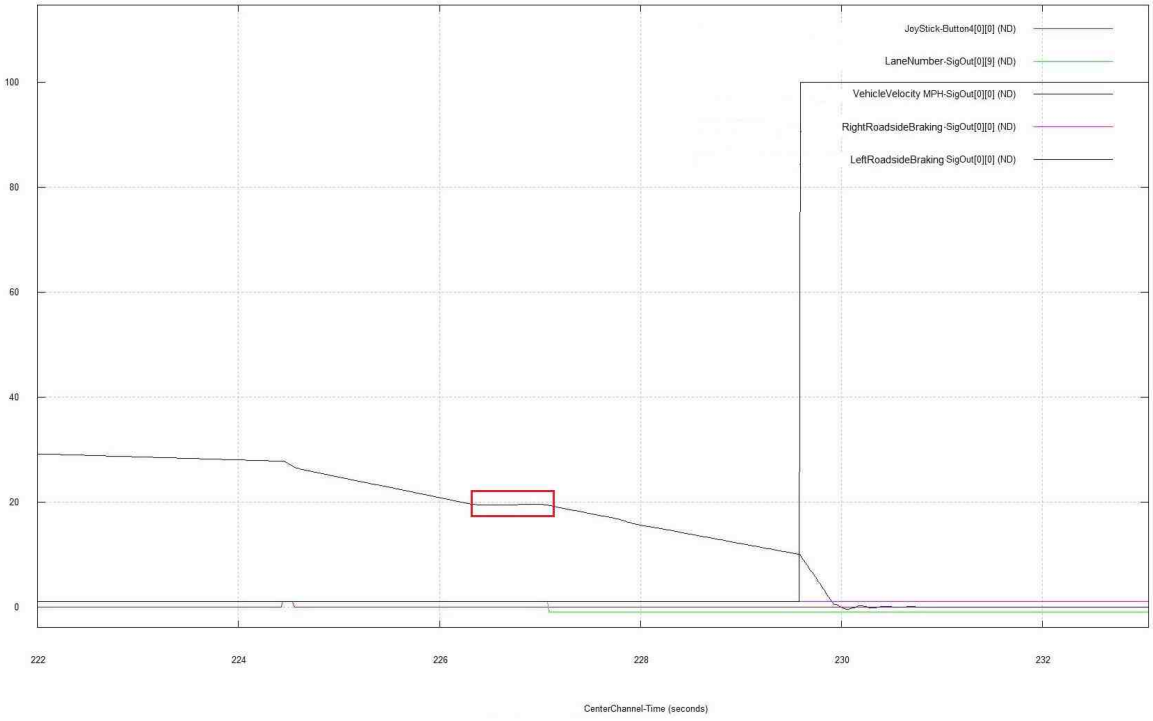


Fig. 5.11. Plot representing the value for variations in vehicle velocity and braking force for left roadside pullover

The left roadside braking plot shown in Fig. 5.11, braking can be observed after the trigger is pressed (red line). The vehicle velocity drops suddenly (blue line) but after it reaches the velocity of 20 mph, the AEPVelocity controller gives output to make it stable at 20 mph (it can be observed in plot from time 226 to 227 approx.). Afterwards, as soon as it enters the Lane -1 (green line) the vehicle decelerates quickly and as it reaches a velocity of 10 mph in the Lane -1 (left roadside safe lane), the braking controller provides a max braking force (black line) to make the vehicle stop completely.

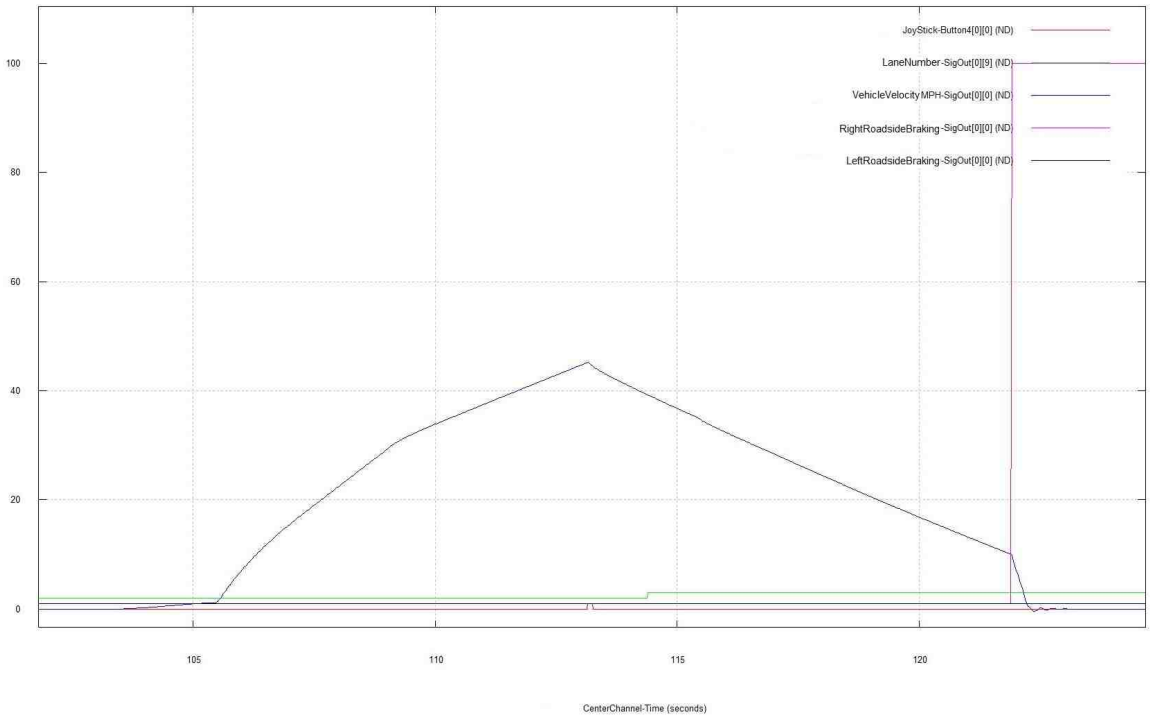


Fig. 5.12. Plot representing the value for variations in vehicle velocity and braking force for right roadside pullover

For right roadside braking plot shown in Fig. 5.12, the operation is opposite to that of left. The vehicle velocity is higher for this typical plot when the trigger was pressed so, there was no need to maintain a velocity of 20mph this time. The driver was accelerating the vehicle and trigger received in the middle (red line), after that the vehicle velocity decelerates quickly (blue line). Moreover, the vehicle velocity is still decreasing (green line) even after entering the Lane 3 (right roadside safety lane) and as early as it reaches the velocity of 10 mph, there is a braking force at its maximum magnitude (purple line) to make vehicle at complete stop on the right pullover lane.

Blinker Output

The blinker control results for both the cases are shown in the plot below and explained after it for each case separately.

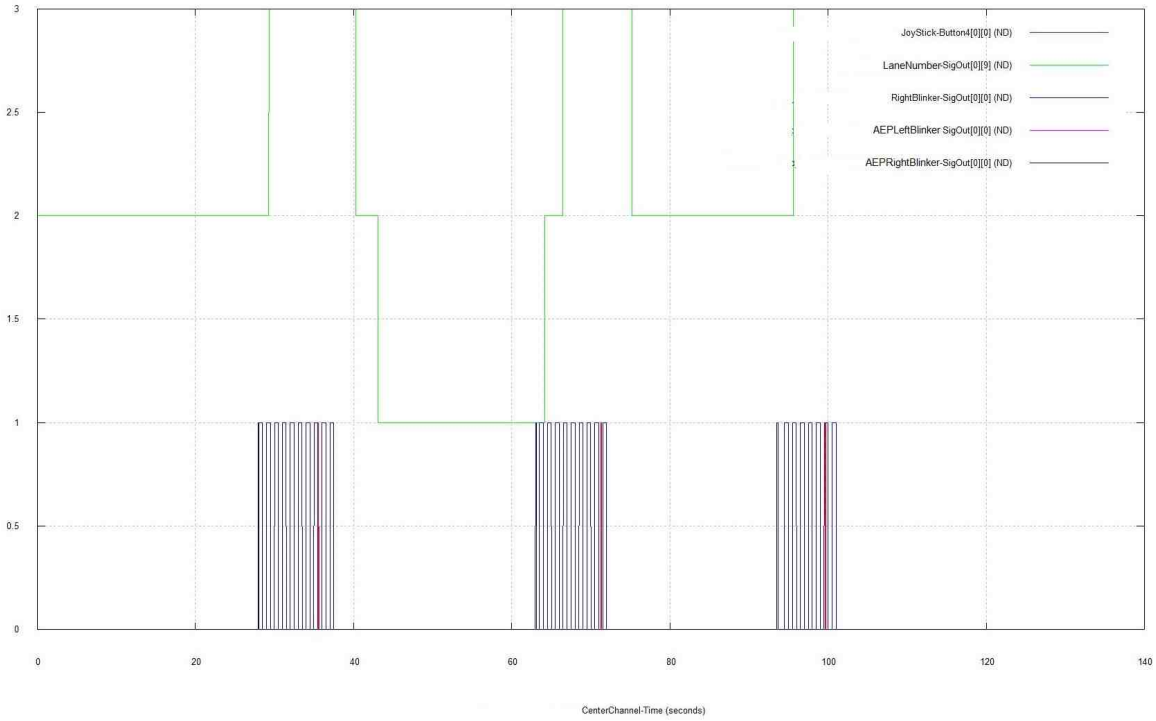


Fig. 5.13. Plot representing the value for variations in LaneNumber and blinker behavior for right simple roadside pullover

For AEP Blinker plot shown in Fig. 5.13, the situation for Case 1 can be observed between 20-40 seconds on plot. After trigger, when the vehicle was in Lane 2 and it has to move to roadside Lane 3 (green line), the AEP right blinker turned ON (black line). Afterwards, the right blinker (navy blue line) works in its typical manner. There is a red line in the middle to turn the health device trigger OFF, when vehicle has reached the right pullover lane and completed its maneuver.

5.3.2 Results for Case 2 Maneuver

Steering

The variations in the steer result based on the nearby roadside objects present in the scenario is represented below through following plots.

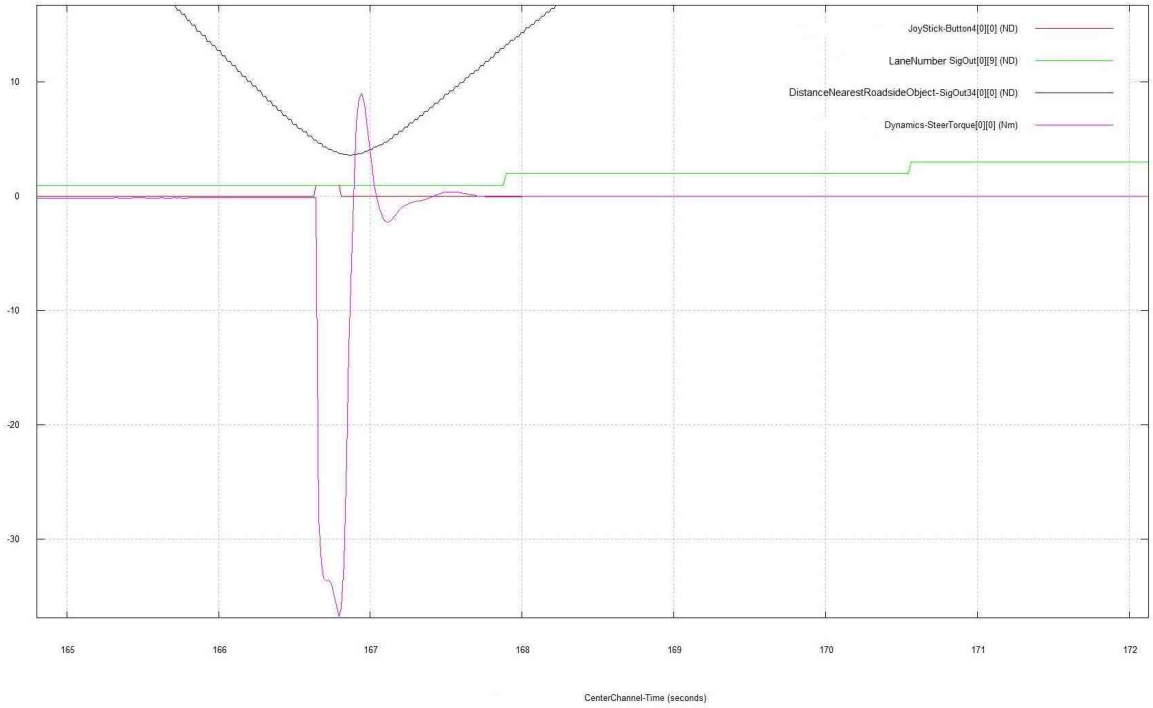


Fig. 5.14. Plot representing the value for variations in LaneNumber and SteerTorque based upon object distance (near) for right roadside pullover

Fig. 5.14 shows when the vehicle is in Lane 1 (green line) and the signal got triggered (red line), the vehicle is supposed to move to Lane -1 (left roadside) but the distance to nearest roadside object is very less (blue line), so it moves to Lane 3 (right roadside). After trigger signal (button 4), there is a negative change in the SteerTorque (right side steering) and vehicle moves to Lane 2 and then to Lane 3.

This is automatic lane change phenomenon when the vehicle crosses one lane in the middle (i.e. Lane 2) to get to the roadside lane (Lane 3) from its current lane.

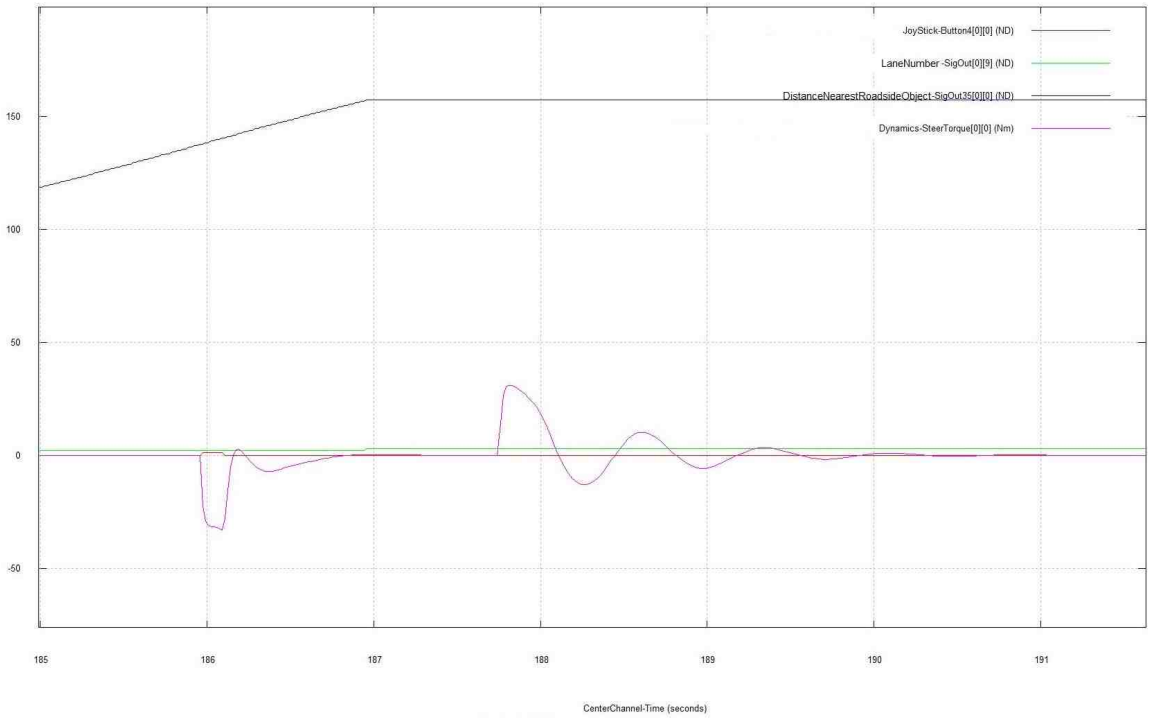


Fig. 5.15. Plot representing the value for variations in LaneNumber and SteerTorque based upon object distance (far) for right roadside pullover

Fig. 5.15 shows the similar pull-over scenario but it can be observed through plot that this time the distance to right roadside object is much more than clearance distance of 8 meters (blue line). In this case, when the health device trigger was received the vehicle was in Lane 2 and it was supposed to pullover in right roadside safety lane (Lane 3). Therefore, it pullover normally to right side and SteerTorque is negative (right side) too. After reaching the Lane 3, the vehicle straighten itself in the right roadside safety location with the help of auto steer technique.

Velocity and Braking

For velocity and braking, the results were similar in Case 2 as the Case 1 because the braking and velocity behavior would be same either there is roadside object present or not. See Figs. 5.11 and 5.12.

Blinker Output

The blinker control results are shown in the plot below and explained after it.

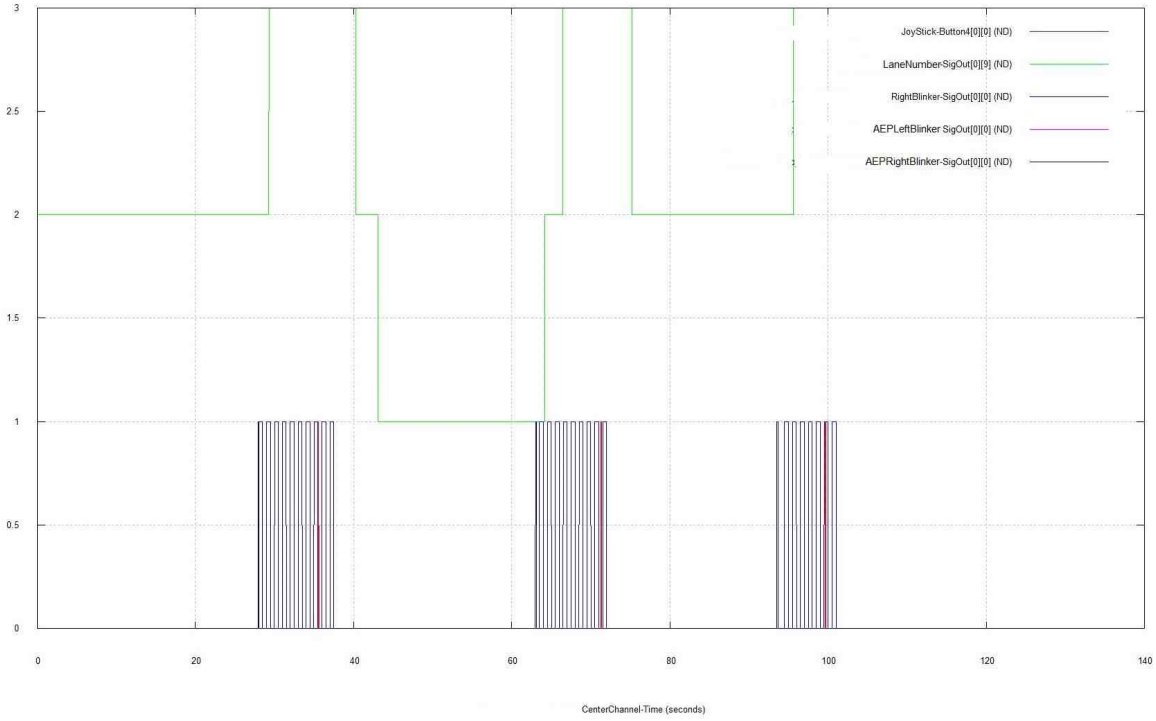


Fig. 5.16. Plot representing the value for variations in LaneNumber and blinker behavior for right roadside object pullover

For AEP blinker plot shown in Fig. 5.16, the situation for case 2 can be observed between 60-75 seconds on plot. Trigger signal received when the vehicle was in the Lane 1 and it has to pullover in Lane -1 but as there is an object in the Lane -1, so it has to pullover in Lane 3. In such situation AEP blinker again put the right side blinker ON and vehicle moves to Lane 2 and then Lane 3. Again, the red line is to put the health device trigger OFF, when vehicle has reached Lane 3 (right roadside safety lane).

Although, left side is same but mostly, right roadside maneuver on freeway is discussed because of the right side drive in the United States.

6. CONCLUSION AND FUTURE WORK

This research work discusses through the preparation and advancements in the scenario development and model development for the automated vehicles. Active safety features implementation is very important for now-a-days traffic environments where you can drive safely and comfortably. Lane keeping assist with lane departure warning, blind spot monitoring with warning, vehicle automatic emergency braking with forward collision warning, pedestrian automatic emergency braking with nearby pedestrian warning, adaptive cruise control with stop-and-go at low speed traffic and automatic emergency pullover system for health monitoring device are the key features whose implementation using Realtime Technologies, Inc. software has been explained in this thesis. The internal algorithms for model components which helped through development of each system separately and interconnected them has been explained explicitly. Scenario development procedures and software tools functioning have also been brought into readers knowledge briefly. Moreover, the limitations on these features and results discussions have also been provided.

There is still room for advancements. It can be observed that ACC has been made for any stage velocity fixation. This approach can be used on other features for advancements as well, likewise in the development of all speed pedestrian collision avoidance system which is really important but very difficult, considering the abrupt appearance and non predictive movement of human being and hard to identify the age and size constraints. The major advantage and freedom is that simulation provide an environment where new system development and behavior can be observed without significant cost.

Automatic emergency pullover is a system introduced in this research work which is not present in the actual vehicles yet. This system also has areas for improvements such as, roadside steering could be more smoother and comfortable, automatic lane

change could be available for any kind of scenarios. However, real vehicle systems have various differences than simulation and they might be much more challenging to install but simulation environment provide us a platform to implement the ideas and take a look at its potential and expected outputs.

REFERENCES

REFERENCES

[1] W. H. Organization *et al.*, "More than 270,000 pedestrians killed on roads each year," *Retrieved March*, vol. 1, p. 2015, 2013.

[2] J. M. Scanlon, K. D. Kusano, R. Sherony, and H. C. Gabler, "Potential safety benefits of lane departure warning and prevention systems in the US vehicle fleet," in *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, no. 15-0080, 2015.

[3] R. J. Kiefer and J. M. Hankey, "Lane change behavior with a side blind zone alert system," *Accident Analysis & Prevention*, vol. 40, no. 2, pp. 683–690, 2008.

[4] J. D. Lee, D. V. McGehee, T. L. Brown, and M. L. Reyes, "Collision warning timing, driver distraction, and driver response to imminent rear-end collisions in a high-fidelity driving simulator," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 44, no. 2, pp. 314–334, 2002.

[5] E. Coelingh, A. Eidehall, and M. Bengtsson, "Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 155–160.

[6] N. H. T. S. Administration *et al.*, "Traffic safety facts: 2007 data: pedestrians," *Annals of Emergency Medicine*, vol. 53, no. 6, p. 824, 2009.

[7] M. Krafft, A. Kullgren, A. Lie, J. Strandroth, and C. Tingvall, "The effects of automatic emergency braking on fatal and serious injuries," in *21st International Conference on Enhanced Safety of Vehicles*, 2009.

[8] E. Rosén and U. Sander, "Pedestrian fatality risk as a function of car impact speed," *Accident Analysis & Prevention*, vol. 41, no. 3, pp. 536–542, 2009.

[9] A. D. Meadows, "Plant error compensation and jerk control for adaptive cruise control systems," Master Thesis, Purdue University, 2012.

[10] Y. Zhai, L. Li, G. R. Widmann, and Y. Chen, "Design of switching strategy for adaptive cruise control under string stability constraints," in *Proceedings of the 2011 American Control Conference*. IEEE, 2011, pp. 3344–3349.

[11] G. Marsden, M. McDonald, and M. Brackstone, "Towards an understanding of adaptive cruise control," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 1, pp. 33–51, 2001.

[12] S. Schleicher, C. Gelau *et al.*, "The influence of cruise control and adaptive cruise control on driving behaviour–a driving simulator study," *Accident Analysis & Prevention*, vol. 43, no. 3, pp. 1134–1139, 2011.

[13] C. A. S. E-Forum, "Restating the nhtsa national motor vehicle crash causation survey for automated vehicles," vol. 1, 2014, Last accessed: 3/27/2017. [Online]. Available: https://www.casact.org/pubs/forum/14fforum/CAS%20AVTF_Restated_NMVCCS.pdf

[14] I. Realtime Technologies, "Simcreator and simvista user's manuals," 2008.

[15] R. Schubert, K. Schulze, and G. Wanielik, "Situation assessment for automatic lane-change maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 607–616, 2010.

[16] N. J. Harrington, "Passive method and apparatus for alerting a driver of a vehicle of a potential collision condition," June 2009, US Patent 7,545,261.

[17] S. A. Kanarachos, "A new method for computing optimal obstacle avoidance steering maneuvers of vehicles," *International Journal of Vehicle Autonomous Systems*, vol. 7, no. 1-2, pp. 73–95, 2009.

[18] D. P. Malone and J. F. Creamer, "Nhtsa and the next 50 years: Time for congress to act boldly (again)," SAE Technical Paper, Tech. Rep., 2016.

[19] B. Fildes, M. Keall, N. Bos, A. Lie, Y. Page, C. Pastor, L. Pennisi, M. Rizzi, P. Thomas, and C. Tingvall, "Effectiveness of low speed autonomous emergency braking in real-world rear-end crashes," *Accident Analysis & Prevention*, vol. 81, pp. 24–29, 2015.

[20] M. Heesen, M. Dziennus, T. Hesse, A. Schieben, C. Brunken, C. Löper, J. Kelsch, and M. Baumann, "Interaction design of automatic steering for collision avoidance: challenges and potentials of driver decoupling," *IET intelligent transport systems*, vol. 9, no. 1, pp. 95–104, 2014.

[21] B. Tang, "Pedestrian protection using the integration of v2v communication and pedestrian automatic emergency braking system," 2015, Last accessed: 3/27/2017. [Online]. Available: http://docs.lib.purdue.edu/dissertations/AAI10109204/

[22] K. Lindqvist and M. Eriksson, "Adaptive cruise control system," May 17 2011, US Patent 7,945,369.