A DETECTION AND MITIGATION SYSTEM FOR UNINTENDED ACCELERATION: AN INTEGRATED HYBRID DATA-DRIVEN AND MODEL-BASED APPROACH

A Dissertation

by

HONGTAO YU

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Reza Langari
Co-Chair of Committee,	Swaroop Darbha
Committee Members,	Sivakumar Rathinam
	Wei Zhan
Head of Department,	Andreas Polycarpou

December 2016

Major Subject: Mechanical Engineering

ABSTRACT

This study presents an integrated hybrid data-driven and model-based approach to detecting abnormal driving conditions. Vehicle data (e.g., velocity and gas pedal position) and traffic data (e.g., positions and velocities of cars nearby) are proposed for use in the detection process. In this study, the abnormal driving condition mainly refers to unintended acceleration (UA), which is the unintended, unexpected, uncontrolled acceleration of a vehicle. It is often accompanied by an apparent loss of braking effectiveness. UA has become one of the most complained-about vehicle problems in recent history.

The data-driven algorithm aims to use historical data to develop a model that describes the boundary between normal and abnormal vehicle behavior in the vehicle data space. At first, several detection models were created by analyzing historical vehicle data at specific moments such as acceleration peaks and gear shifting. After that, these models were incorporated into a detection system. The system decided if a UA event had occurred by sending real-time vehicle data to the models and comprehensively analyzing their diagnostic results. Besides the data-driven algorithm, a driver model-based approach is proposed. An adaptive and rational driver model based on game theory was developed for a human driver. It was combined with a vehicle model to predict future vehicle behavior. The differences between real driving behavior and predicted driving behavior were recorded and analyzed by the detection system. An unusually large difference indicated a high probability of an abnormal event.

Both the data-driven approach and the model-based approach were tested in the Simulink/dSPACE environment. It allowed a human driver to use analog steering wheels and pedals to control a virtual vehicle in real time and made tests more realistic. Vehicle models and traffic models were created in dSPACE to study the influences of UA and ineffective brakes in various roadway driving situations. Test results show that the integrated system was capable of detecting UA in one second with high accuracy. Finally, a brake assist system was designed to cooperate with the detection system, which reduced the risk of accidents.

ACKNOWLEDGEMENTS

I would like to begin by expressing my profoundest gratitude to my advisor, Dr. Reza Langari, for his guidance, encouragement and support throughout my graduate studies. He not only gave me the freedom to pursue my research interests, but also patiently guided me through difficulties. I could not have hoped for a better advisor.

I am also indebted to Dr. Swaroop Darbha, Dr. Sivakumar Rathinam and Dr. Wei Zhan for serving on my committee. Their insightful suggestions and comments helped me look at problems from different perspectives and have been absolutely invaluable.

I would like to thank my colleagues and friends from Texas A&M Transportation Institute, University of Michigan Transportation Research Institute, Texas A&M University and University of Houston as well. It was a pleasure working with them and we built good team chemistry.

Finally, I want to give special thanks to my family for their unconditional love. In particular, I would like to thank my parents. This journey would not have been possible without their consistent support.

NOMENCLATURE

ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
ECM	Engine Control Module
ECU	Electronic Control Unit
EPS	Electric Power Steering
ESC	Electronic stability Control
ETC	Electronic Throttle Control
FDIR	Fault Detection, Isolation and Recovery
FN	False Negative
FP	False Positive
ITS	Intelligent Transportation Systems
KNN	K-Nearest Neighbor
LDA	Linear Discriminant Analysis
ML	Machine Learning
MLP	Multilayer Perceptron
MPC	Model Predictive Control
OBD	On-Board Diagnostics
PID	Proportional-Integral-Derivative
RRT	Rapidly Exploring Random Tree
SVM	Support Vector Machine

TN	True Negative
TP	True Positive
TPS	Throttle Position Sensor
UA	Unintended Acceleration

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

1.1 Background

Automotive electronics have evolved enormously over the past decades. Numerous powerful electronic control units (ECUs) were developed to equip modern vehicles, which made driving much easier and more comfortable. As can be expected, a growing number of autonomous vehicles or partially autonomous vehicles will come to the market in the next few decades [1]. However, driving is an inherently complex task and cars can be very dangerous if not handled properly. As the intelligence level of vehicles increases, people gradually lose direct control of vehicles and raise worries about the safety and reliability of vehicle systems.

One of the typical problems related to electronic systems is unintended acceleration (UA). The US *National Highway Traffic Safety Administration* (NHTSA) defines UA as "the occurrence of any degree of acceleration that the vehicle driver did not purposely cause to occur." An apparent loss of braking effectiveness often happened at the same time [2]. UA first came to public attention due to its higher-than-usual occurrence as reported by drivers [3]. It has been a serious problem across the auto industry for many years. Audi cars received an unusually large number of complaints about UA in the 1980s. However, Audi and other vehicles manufactured at that time had comparatively few electronics systems and used mechanical throttle control. NHTSA attributed the cause of Audi's problems to drivers' misapplication [4].

Examination Past 10 Years of VOQs (Mechanical and Electronic Throttles)	
Drilldown into UA VOQs	**No. of VOQs**
Total VOQs received 1/1/2000 to 3/5/2010	426,911
UA related VOQs identified by key-word search	19,269
VOQs remaining after manual review	11,454
Model years 1998 – 2010	9,698
Toyotas (Model years 1998 – 2010)	3,054
Camry with Electronic Control System (2002-2010)	831

Figure 1.1 Unintended acceleration complaints [5]

In contrast to traditional simple and straightforward mechanical control systems, electronic systems have become more and more popular in vehicles in the past 25 years. One of the widespread systems is the electronic throttle control (ETC) system. Unfortunately, the promotion of ETC was accompanied by an increasing number of complaints about UA problems, which was doubted to be caused by software or other electronic failure. The National Aeronautics and Space Administration (NASA) and NHTSA conducted several analyses to study the cause of UA, as shown in Figure 1.1. Four major factors were deduced including pedal misapplication, unresponsive pedals, electronic throttle control or cruise control failure and stuck throttle. Pedal misapplication was concluded to be the most plausible reason. However, the possibility of vehicle defects still could not be ruled out [6, 7]. Mariani [8] claimed that soft errors should be considered for drive-by-wire automotive components. Koopman [9] argued that software bugs such as excessive use of global variables, concurrency bugs and race conditions could also be the cause of UA. Sood et al. [10, 11] stated that shorting of tin

whiskers could be a cause of UA. ETC is not the only widely used electronic system that has received a large number of complaints. With continued emergence of novel intelligent systems and people's growing dependence on them, fault detection, isolation and recovery (FDIR) has become vitally important. A well-designed vehicle should be able to find the causes of potential problems in real time and take action to prevent accidents.

1.2 Objective

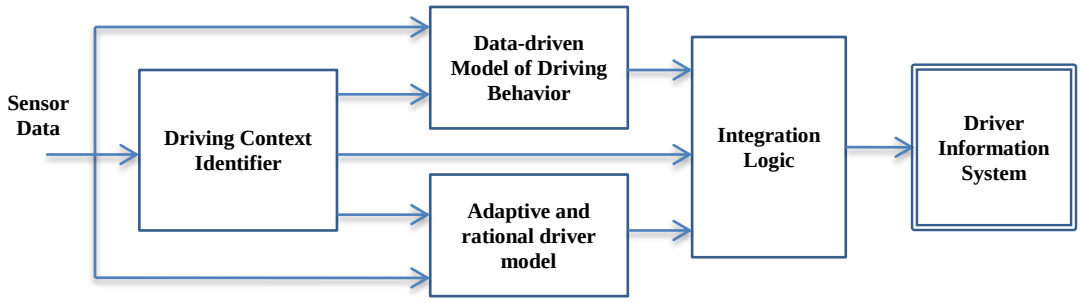

Figure 1.2 Schematic architecture for driving condition identification

The objective of this project is to develop an integrated hybrid model-based and data-driven approach to detecting abnormal driving conditions, as illustrated in Figure 1.2. Vehicle data (e.g., velocity, yaw rate and throttle position) and traffic data (e.g., velocities of cars nearby and space headway) are proposed for use in the detection process. In this task, the abnormal driving condition mainly refers to UA.

As one of the most complained-about problems in recent history, UA has drawn wide public attention and raised worries of vehicle safety. While human actions are involved in UA events, behavior of vehicular systems and human-vehicle interaction

also play an important role and need to be analyzed. This task aims to further study the causation of UA events and create corresponding detection and mitigation systems. Some findings from this research can also be extended to the emerging autonomous vehicle technology domain.

1.3 Literature review

Intelligent and electronic vehicle systems have been developed rapidly over the past few decades, and it is still a dominant trend of future vehicles. With the gradual transition from manual control to smart control, improvement of reliability and safety has become increasingly important. Automatic fault diagnosis systems need to be well-designed to identify abnormal vehicle behavior in real time. A lot of research has been done to study fault diagnosis algorithms for automotive components. In particular, it is possible to efficiently collect and use traffic data in traffic modelling, remote monitoring and wireless diagnosis with the rapid development of intelligent transportation systems (ITS) [12, 13]. In the following sections, previous studies about fault diagnosis, vehicle systems related to UA, driver models and game theory are reviewed respectively.

1.3.1 Fault diagnosis

Fault diagnosis is an important and challenging task in various disciplines. Extensive efforts have been devote to this field. Generally speaking, fault diagnosis consists of three tasks. The first one is fault detection. The occurrence of faults in the functional units, which results in undesired behavior of the system, should be detected. The second task is fault isolation, which aims to identify the position and type of the fault. The third task is recovery. It allows the system to continue operation to a level of satisfaction

when a fault occurs [14].

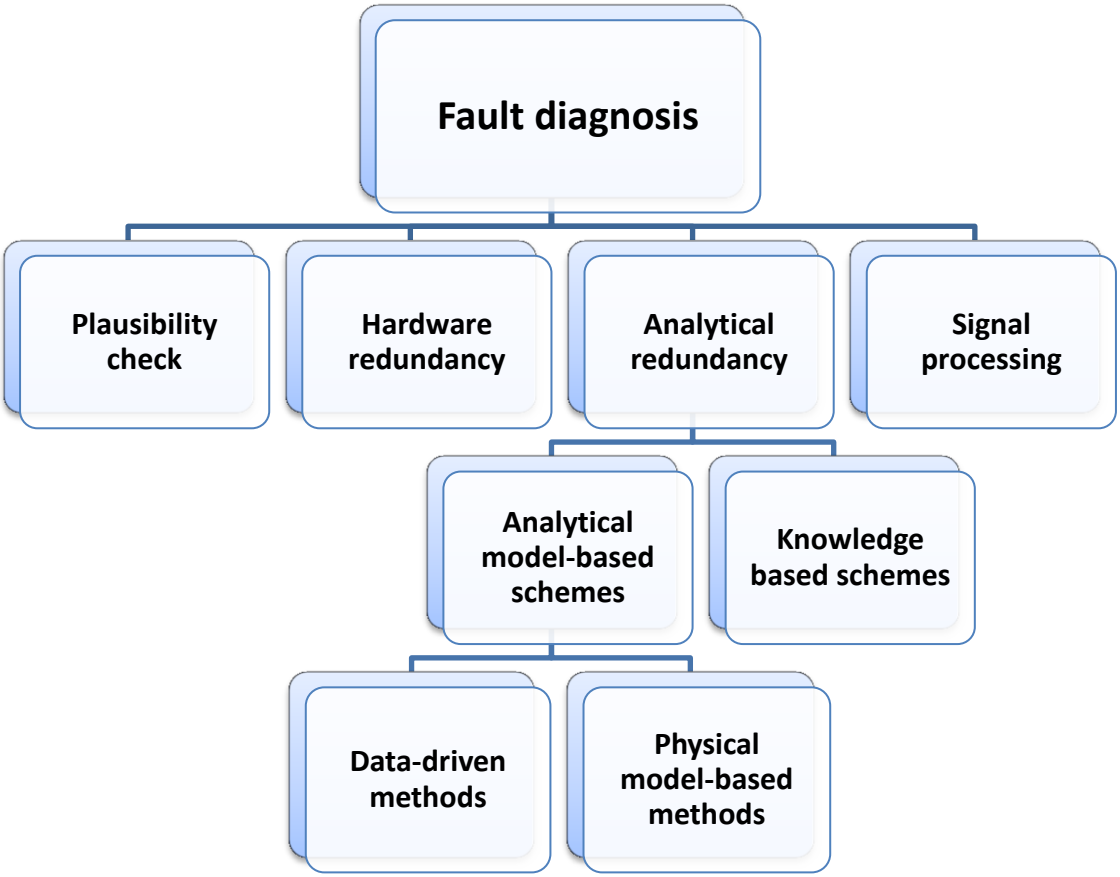

Figure 1.3 Fault diagnosis methods

Numerous fault diagnosis algorithms have been proposed in the past decades [15-17]. A rough classification of these algorithms is shown in Figure 1.3. These methods can be classified into four categories, which are plausibility check, hardware redundancy, analytical redundancy and signal processing. Plausibility check analyzes if a process component obeys some simple physical laws. For example, a system state is usually monitored by one sensor. A threshold of this state value is set according to experience and a fault is considered to occur when the state value exceeds the threshold.

This method is popular because of its simplicity and reliability. It is extensively used in the on-board diagnostic system (OBD), which was designed to maintain low-emissions of in-use vehicles. However, it can only detect large faults and does not allow a deep insight into the process behavior [18]. In addition, UA is a multi-dimensional problem, which is influenced by gas pedal positions, brake pedal positions, road grades and so on. An error may occur when all the state values stay within their own normal ranges. Hence, limit value checking is not a good choice for detecting UA. Hardware redundancy is another traditional way to diagnose faults. It reconstructs process components (i.e. sensors and actuators) using identical components. It is concluded that an error has occurred if process components and redundant components have different outputs. This method has high reliability, but its cost is relatively too high. Therefore, hardware redundancy is not very popular and it is only used in certain key components (e.g., the actuator of an air pressure regulator of an airplane).

To conduct in-depth fault diagnosis, a model-based fault diagnosis approach was proposed in the early 70's and great progress has been made since then. Basically, a process model of the system is usually created first, as shown in Figure 1.4. Subsequently, algorithms are applied to processing real-time data and diagnosing faults based on the model [19]. Würtenberger et al. [20] proposed a method to supervise vehicle motion using a discrete parity space. Moseler et al. [21] presented a parameter estimation technique for fault detection on a brushless dc motor. This method usually relies on a highly developed model, which can represent the complete system dynamics, e.g., models of electric motors. However, a vehicle's acceleration is affected by multiple

factors, e.g., environmental conditions and engine status. It is quite difficult to build a process model for all the vehicle systems related to UA. First of all, engines and tires are difficult to model accurately. Furthermore, ECUs, which may be the most important parts of modern vehicles, are often kept as commercial secrets and very difficult to model. Finally yet importantly, it is computationally complex to simulate the whole vehicle in real time. In summary, it is difficult to model the complete driving loop accurately in real time. Nevertheless, a combination of driver models and simplified vehicle models is still useful for evaluating the vehicle's status.

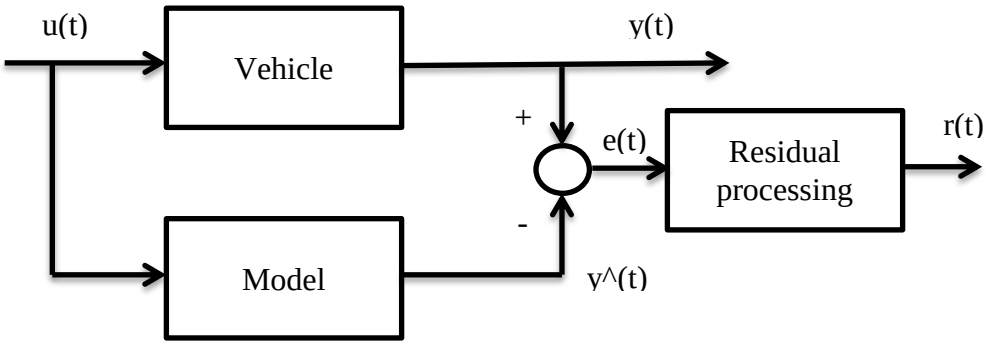

Figure 1.4 Model-based fault diagnosis

In contrast to the physical model-based fault diagnosis methods, data driven methods are based on a large amount of process data, including both historical data and online measurement data [22]. It mainly consists of two steps. The first step is training, in which the historical data sets are processed and transformed to a diagnostic system or algorithm. The second step is online running, in which data are collected and processed in the diagnostic system in real time for reliable fault diagnosis. In addition to these algorithms, signal processing based approaches are also widely used in fault diagnosis

[23-25]. It assumes that certain process signals carry the information about faults and the information can be obtained after processing signals properly. Symptoms are generated in the time domain (e.g., mean values and amplitudes) or frequency domain (e.g., power spectral density). Signal processing based algorithms are usually applied to the processes in the steady state.

Fault diagnosis algorithms have been widely used in the automotive field [26]. Schmidt et al. [27] built a framework to enable vehicles to communicate with other vehicles in the vicinity, using Vehicular Ad-Hoc Networks. Vehicles were classified as trustworthy, untrustworthy or neutral in regards to traffic safety. Sang et al. [28] identified abnormal behavior by performing video-based detection and creating an information chain of tracked vehicles. Bouttefroy et al. [29] introduced a paradigm for abnormal behavior detection, relying on the integration of contextual information in Markov random fields. This technique models the local density of object feature vector and leads to simple and sophisticated criterion for behavior classification. Chen et al. [30] presented an approach to detect potential ECU application software abnormal behavior based on the Mahalanobis Distance, the Mahalanobis-Taguchi System and vehicle driving data playback capability with a simulator. Jeppesen et al. [31] demonstrated an observer-based technique for monitoring an active suspension control system. Several observers were created for vehicle subsystems and the observer residuals were normalized and combined to perform fault diagnosis.

1.3.2 Vehicle systems related to UA

A growing number of electronic systems have been added to automobiles in the past

twenty-five years. While the vehicle becomes increasingly intelligent, the safety and reliability of these electronic components become more and more important.

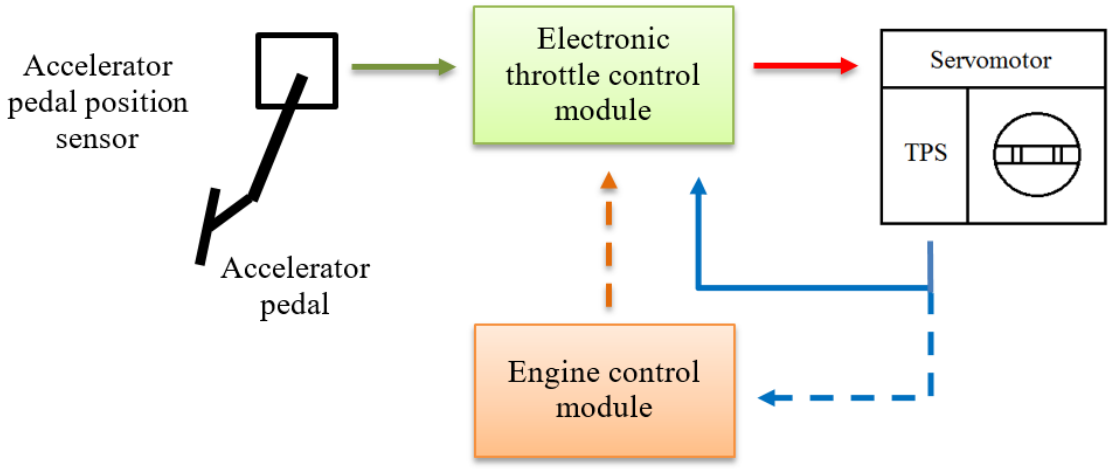

Figure 1.5 Electronic throttle control system

One of the widely used vehicle electronic systems is ETC system. ETC is a drive-by-wire technology which connects the gas pedal to the throttle electronically [32], as shown in Figure 1.5. An ETC system usually consists of three major parts: (1) a gas pedal module, (2) a throttle valve which can be controlled by an electric motor and a throttle position sensor (TPS), (3) an engine control module (ECM) [33]. ETC aims to make the vehicle power-train characteristics seamlessly consistent regardless of conditions such as engine temperature. It is also capable of adjusting throttle positions irrespective of the driver's gas pedal positions. This feature facilitates the integration of other electronic systems, which need torque management such as cruise control system and ESC system. However, ETC was criticized since it overruled driver decisions. Its safety and reliability has been suspected.

Michael Barr worked as an expert witness and testified in the Toyota UA litigation. He analyzed Toyota's ETC source code and concluded that the code was defective and contained bugs, which could cause UA [34]. Single bits in memory controlled each task. Therefore, hardware or software faults could result in bit flips, which suspended desired tasks or even started unwanted tasks. Michael Barr argued that one particular dead task could cause loss of throttle control and UA. The driver might have to fully remove the foot from the brake during UA events before being capable of ending UA, which was confirmed in tests. In addition, mirroring was used to protect variables against software- and hardware-caused corruptions. However, he found that some critical variables were not mirrored. In summary, fail safes may be inadequate and misbehaviors of ETC could be a cause of UA [34].

Apart from ETC faults, the failure of cruise control and adaptive cruise control (ACC) can also result in UA. Cruise control is an electronic system that controls the speed of a vehicle automatically. It takes over the throttle control of a car to maintain a steady speed set by the driver. ACC goes one step future by adding automatic braking or dynamic set-speed type controls. The system uses a radar or laser setup to detect cars nearby and calculate the safe following distance. When the vehicle in front gets too close, the brake is controlled to let the car slow down [35]. There were complaints about the failure of cruise control or ACC. For instance, a jammed cruise control trapped a driver in his car for half an hour at it raced along the Eastern Freeway in Melbourne, which was very threatening [36].

According to the traffic and accident reports, UA is often accompanied by ineffective

brakes [2]. A typical brake system contains a brake pedal, a pushrod, a master cylinder assembly, reinforced hydraulic lines and brake caliper assemblies. Many vehicles are also equipped with vacuum brake boosters to provide power assist to the brake system, as shown in Figure 1.6.

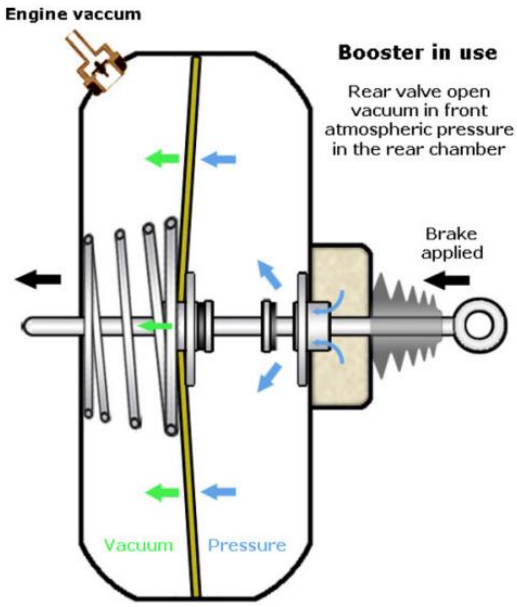

Figure 1.6 Brake booster [37]

A brake booster usually has two or more chambers, which are divided by flexible diaphragms. When the driver is not pressing the brake pedal and chambers are attached to the intake manifold of the engine, a two-way valve let the pressure in both chambers decrease. In this state, the booster stores the engine vacuum and works like a reservoir. When the driver presses the brake pedal, the two-way valve moves and let the vacuum only apply to the front. At the same time, atmospheric pressure air flows into the rear chamber and pushes the diaphragm. This force pushes the master cylinder piston and

assists braking. It can be seen that the brake booster highly relies on the engine vacuum. Based on the data from NHTSA [38], with depleted vacuum, it requires average of one hundred and seventy five pounds of force on the brake pedal to hold against wide open throttle. In contrast, it only needs fifteen to forty-three pounds of force with vacuum. If the throttle is not controlled properly, for example, it is wide open when the driver presses the brake pedal, it will be very difficult to stop the car, which is extremely dangerous.

1.3.3 Driver model

Driving is a complicated task and a driver is the most important component in the driving loop. Abnormal driving could be detected by analyzing the difference between the behavior of a rational driver model and the behavior of a human driver. An accurate driver model is drawing increasing interest from transportation researchers. Numerous driver models have been developed since the 1950s, including car-following models and lane-changing models.

Car following is one of typical driving tasks and has been extensively studied [39-41]. Newell [42] presented a car-following model based on the assumption that a human driver had a desired speed and tended to converge to this speed in an exponential fashion. Tyler [43] assumed that a human driver was similar to a linear optimal controller which tried to minimize a quadratic cost function of range error and range rate error. Burnham [44] improved Tyler's model by adding the reaction time of human drivers and vehicle nonlinearities. Gipps [45] presented a behavioral car-following model based on the assumption that each driver sets the limits of his desired braking and

acceleration rates. Lee et al. [46, 47] made a comparison between several classic longitudinal driver models and argued that the Gipps' model had the best performance. They also modified the Gipps' model by considering the reaction time of the following driver to the deceleration of the lead vehicle. In the modified Gipps' model, the velocity of the car is given by

$$V_n(t+\tau) = min\begin{cases} v_n(t) + 2.5a_n\tau\left(1 - \frac{v_n(t)}{V_n}\right)\sqrt{0.025 + \frac{v_n(t)}{V_n}} \\ b_n\tau + \sqrt{b_n^2\tau^2 - b_n\left[2[x_{n-1}(t) - s_{n-1} - x_n(t)] - 2v_n(t)\tau - \frac{v_{n-1}(t)^2}{\hat{b}}\right]} \end{cases} \quad (1.1)$$

where a_n is the vehicle n driver's allowable maximum acceleration, b_n is the vehicle n driver's allowable maximum deceleration, $\hat{b}$ is the estimated value for b_{n-1}, s_{n-1} is the effective size of vehicle n, V_n is the vehicle n driver's desired speed, $x_n(t)$ is the location of the front of vehicle n at time t, $v_n(t)$ is the speed of the vehicle n at time t.

Besides car-following models, efforts have also been devoted to studying drivers' lateral behavior [48-50]. Ahmed [51] proposed a gap acceptance model to decide if the adjacent gap is sufficient for lane changes. Laval et al. [52] presented an algorithm to capture the relaxation phenomena commonly observed in lane changes. Lane changes took place based on a stochastic process, whose mean value was a function of lane-specific macroscopic quantities. Yoo et al. [53] considered the application of a Stackelberg game to a driver behavior model in a merging situation. Yang et al. [54] proposed a rule-based lane-changing model. Lane changes were classified as either mandatory (MLC) or discretionary (DLC). A gap acceptance model was developed to

decide if it was safe to change lanes. The acceptable gap decreased linearly as the driver got closer to the location where a MLC had to take place. Toledo et al. [55] suggested that the subject vehicle changed lanes when both the lead gap and lag gap were bigger than predefined thresholds., as shown in Figure 1.7.

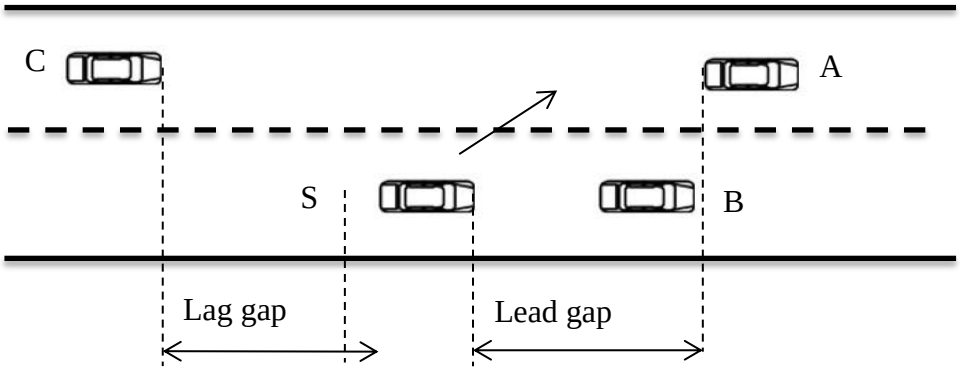

Figure 1.7 Gap acceptance model

1.3.4 Game theory

Game theory is "the study of mathematical models of conflict and cooperation between intelligent rational decision-makers" [56]. It has been extensively developed and applied to various disciplines since the 1950s. Besides its applications in the fields of economics, political science and psychology, game theory has been implemented in the engineering field to study human reasoning and interactions.

In game theory, games are usually well-defined mathematical objects, which include the players of the game, available strategies and payoffs for outcomes. A simple example is demonstrated in Table 1.1. This game has two players, P1 and P2. P1 has two choices of strategies (i.e. a, b) and P2 has two strategies (i.e. c, d). The payoffs of combinations

of strategies are given in the interior. The first number is the payoff for the row player (i.e. P1) and the second number is the payoff received by the column player (i.e. P2). A lot of research has been done to study particular sets of strategies called "solutions" or "equilibria". A solution is a stable state in which no player can profit by unilaterally changing his strategy.

Table 1.1. Nash equilibrium

Game theory		P2	
		c	d
P1	a	6; 1 →	4; 3 ↑
	b	5; 4 ←	3; 2

A famous solution of the non-cooperative game is the Nash equilibrium [57]. In this game, it is assumed that each player acts simultaneously or at least not earlier than players' actions. The Nash equilibria represent strategy pairs such that no player can obtain a better payoff by changing their own strategy. In the game given in Table 1.1, arrows indicate each player's preferred course of action. For instance, player P1 tends to choose the strategy pair (a, c) instead of (b, c) because (a, c) has a higher payoff. It can be easily seen that the pair (a, d) is the Nash equilibrium because both players P1 and P2 cannot benefit by unilaterally deviating from the strategies. Arrows also point to that strategy pair.

In contrast to simultaneous games, later players have a prior knowledge about earlier actions. These games are called sequential games. One classical solution of these kinds of games is the Stackelberg equilibrium [58]. A typical Stackelberg game is shown in Figure 1.8. In this game, P1 is the leader and acts first. The follower P2 chooses its strategy after that. It is assumed that the leader P1 knows that the follower P2 observes its action. Meanwhile, the leader also knows that the follower will not commit to a future non-Stackelberg follower action. Under these assumptions, it can be seen that the Stackelberg equilibrium of the game is (b, c). P1 knows that after it chooses strategy a, P2 will choose strategy d because its payoff is higher.

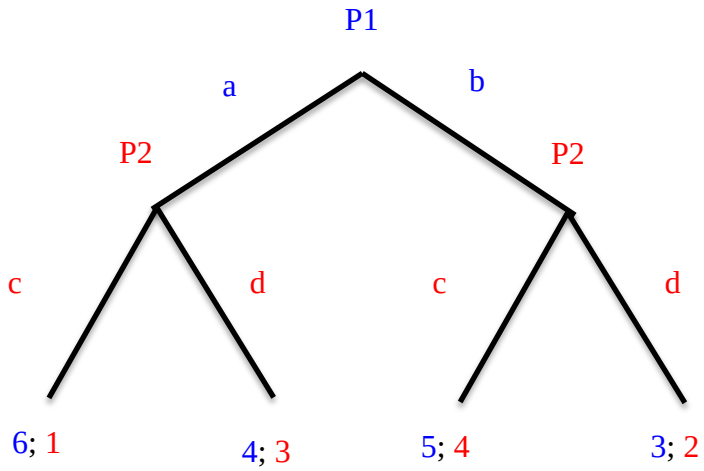

Figure 1.8 Stackelberg equilibrium

One of the main applications of game theory in the engineering field is traffic modelling [59]. Fisk [60] argued that two game theory models (i.e. Stackelberg game and Nash non-cooperative game) could be utilized in the decision-making process for transportation modelling. Talebpour et al. [61] presented a lane-changing model based

on a game-theoretical approach in a connected vehicular environment. The prediction capability of the model is tested and results show that the framework is able to predict lane-changing behavior with limitations that still need to be addressed. Alvarez et al. [62] proposed to model signalized intersections based on game theory and a Markov chain model.

1.4 Outline of the dissertation

The rest of this dissertation is organized as follows. Section 2 describes the development of the data-driven detection and mitigation system. The detection system was validated by both the data collected from vehicles controlled by a drive model in Simulink and the data collected from virtual vehicles in the dSPACE driving simulator. Section 3 presents a driver model-based detector for UA. A rational and adaptive driver model was created and its behavior was compared with the behavior of a human driver in real time in dSPACE. Finally, a summary of this work is given in Section 4.

2. DATA-DRIVEN DETECTION SYSTEM*

2.1 Overview

In this chapter, we are going to develop a data-driven method to detect UA, which does not need a complete prior knowledge of the vehicle structure. Its scheme is shown in Figure 2.1.

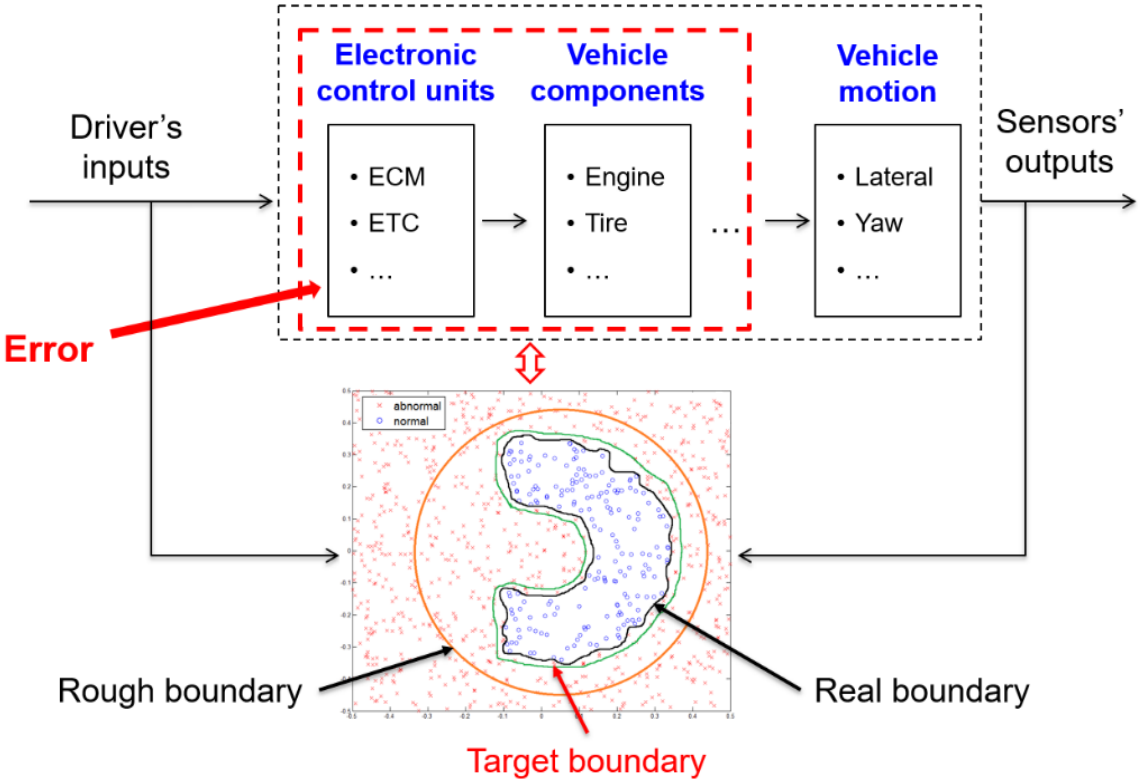

Figure 2.1 Data-driven detection

* Reprinted with permission from "Detection of Unintended Acceleration in Longitudinal Car Following" by Hongtao Yu and Reza Langari, 2015. SAE International Journal of Passenger Cars-Electronic and Electrical Systems, Volume 8, Pages 306-313, Copyright 2015 by SAE and permission from "A Detection and Warning System for Unintended Acceleration" by Hongtao Yu and Reza Langari, 2015. ASME 2015 Dynamic Systems and Control Conference, pp. V002T31A002-V002T31A002, Copyright 2015 by ASME.

Modern vehicles have a series of sensors. The data collected by these sensors form an n-dimensional data space. Each point in this data space represents a vehicle state. It is assumed that if normal vehicle behavior and abnormal vehicle behavior can be separated, a boundary between them can be founded in this data space or the transformation of this space. Since it is impossible to collect infinite data points which cover the whole space and even cars of the same type may have different performance characteristics, the objective is only to find an approximate boundary rather than the real boundary [63].

The first step to build a data-driven model to define the boundary was to collect vehicle data. Experiments were designed and conducted in the Simulink[1]/dSPACE[2] environment to gather data for both normal and abnormal vehicle behavior. The data set was then divided into the training part and test part randomly. Features were extracted from the training data set and the optimal combination of features was selected to train data-driven models. These models were combined to formulate a detection system for UA. Finally, the test data set was used to validate the system.

Besides the classification unit, a preprocessing unit and a brake assist system were also designed and combined to prevent accidents. A flowchart of the whole system is shown in Figure 2.2. At first, related vehicle signals are sampled and smoothed. Processed signals are sent to the classification unit in real time. If the classification unit doubts an error occurs, the system cuts the engine power and adjusts the braking force to

---

[1] Simulink is a registered trademark of The MathWorks, Inc.
[2] dSPACE is a registered trademark of dSPACE GmbH.

reduce the risk of accidents.

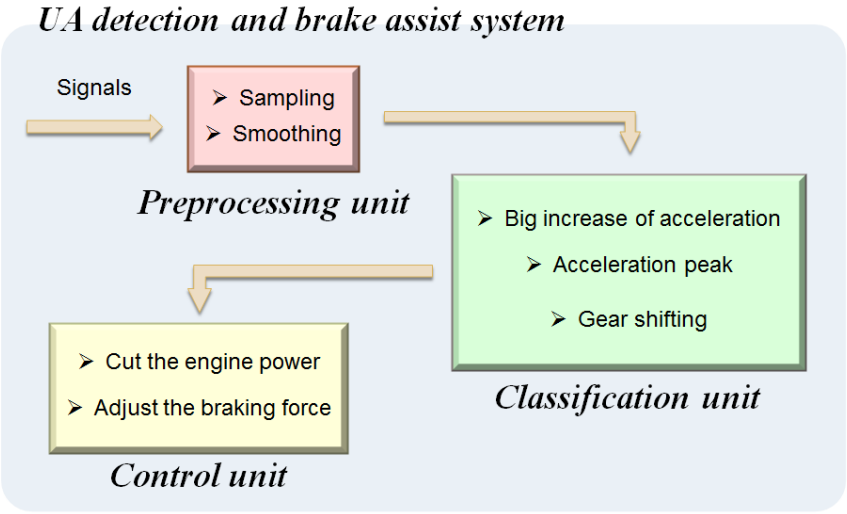

Figure 2.2 UA detection and brake assist system

2.2 Collect data from vehicle models in Simulink

A large amount of representative data points are needed to create a robust data-driven model. Efforts have been devoted to collecting and analyzing vehicle data. For instance, the University of Michigan Transportation Research Institute (UMTRI) has developed large driver-vehicle databases since the mid-1990s [64]. However, these data sets only contain data points from normal vehicles. If these data points cannot cover the whole space of normal behavior, the boundary obtained from them could deviate largely from the real one. In contrast, a few data points of abnormal vehicle behavior could help transform the data space and find a boundary much closer to the real boundary. Furthermore, data points of abnormal vehicle behavior are needed to validate the model. In order to collect data for both normal and abnormal vehicle behavior, new experiments in which UA is simulated need to be designed.

At the first stage, experiments were conducted in Matlab / Simulink. A three dimensional vehicle model was built. The model contained tire models, an engine, an automatic transmission system and other powertrain components. Vehicle problems and drivers' reactions were simulated and added during the simulation. Sensor noise was also considered and corresponding filters were designed and applied. After performing traffic simulation in Simulink, vehicle data including RPM, acceleration, brake pedal positions, velocity, and accelerator pedal positions were collected under both normal and abnormal driving conditions.

2.2.1 Powertrain model

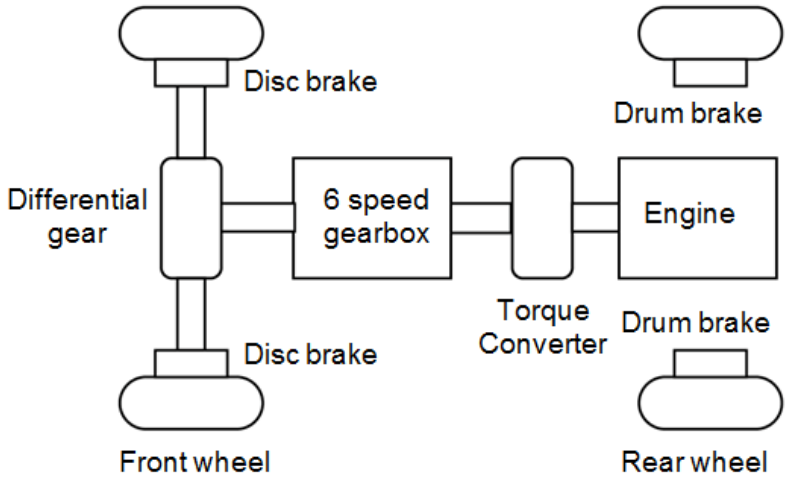

Figure 2.3 Powertrain model

A three dimensional automatic front-wheel drive (FWD) vehicle model was built in Simulink. The model contains basic powertrain components including engine, torque converter, gearbox, differential gear, and brake, as shown in Figure 2.3. Front wheels use disc brakes and rear wheels use drum brakes. Independent rear suspension is applied.

The car is assumed to use symmetric camber angles and toe angles so that it can move along a straight line when the steering angle is zero. The reason for choosing automatic transmissions is that, based on reports, UA most often happens with automatic vehicles.

2.2.2 Vehicle dynamics

In this study, a Society of Automobile Engineers (SAE) standard coordinate system is used [65]. The positive x axis points forward, the positive y axis points to the right and the positive z axis points down. Positive rotations are determined by the right-hand rule for these axes, as shown in Figure 2.4.

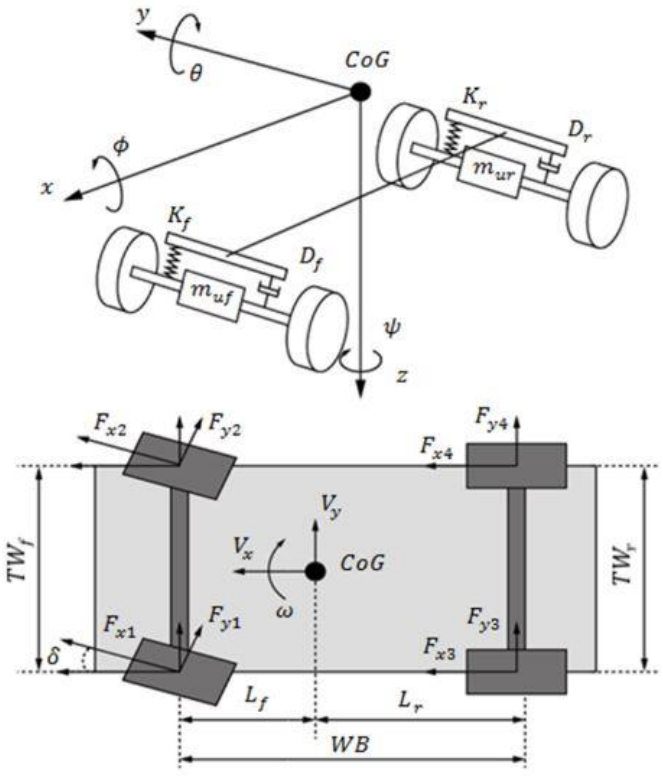

Figure 2.4 Vehicle dynamics

The longitudinal, lateral and yaw movement of a vehicle is described by

$$m(a_x - V_y\omega) = \sum_i FX_i - F_{drag} - F_{slope} \tag{2.1}$$

$$m(a_y + V_x\omega) = \sum_i FY_i \tag{2.2}$$

$$I_z\alpha_z = \sum_{i=1}^{2} FY_i L_f - \sum_{i=3}^{4} FY_i L_r + (FX_1 - FX_2)\frac{TW_f}{2} + (FX_3 - FX_4)\frac{TW_r}{2} + \sum_{i=1}^{4} M_{zi} \tag{2.3}$$

where m is the vehicle mass, a_x and a_y are longitudinal and lateral acceleration, V_x and V_y are longitudinal and lateral velocities, α_z represents the yaw angular acceleration, I_z is the yaw moment of inertia, F_{drag} is the aerodynamic drag force, F_{slope} is the force caused by the road grade and gravity, ω denotes the yaw rate, L_f and L_r are the distance from CoG to the front and rear axles, TW_f and TW_r are the front and rear track width, and M_{zi} is the tire aligning moment.

The longitudinal and lateral wheel forces are calculated as:

$$FX_i = \sum_i F_{xi}cos\delta_i - \sum_i F_{yi}sin\delta_i \tag{2.4}$$

$$FY_i = \sum_i F_{xi}sin\delta_i + \sum_i F_{yi}cos\delta_i \tag{2.5}$$

where F_{xi} and F_{yi} are longitudinal and lateral tire forces, which are determined from the

tires normal forces, and δ_i is the steering angle. The rolling resistance is included in F_{xi}. For a small roll angle ϕ and pitch angle θ, roll and pitch equations are given by:

$$\begin{aligned}(I_x + m_c d_{roll}^2)\ddot{\phi} + \left(D_{rollf} + D_{rollr}\right)\dot{\phi} + \left(K_{rollf} + K_{rollr}\right)\phi \\ = m_c g d_{roll} sin\phi - m_c\left(a_y + V_x\omega\right)d_{roll} cos\phi\end{aligned} \tag{2.6}$$

$$\begin{aligned}\left(I_y + m_c d_{pitch}^2\right)\ddot{\theta} + D_{pitch}\dot{\theta} + K_{pitch}\theta \\ = m_c g d_{pitch} sin\theta + m_c\left(a_x - V_y\omega\right)d_{pitch} co\, s\, \theta\end{aligned} \tag{2.7}$$

where d_{roll} is the length of the roll moment arm, K_{rollf} and K_{rollr} are front and rear roll stiffness, D_{rollf} and D_{rollr} are front and rear roll damping, d_{pitch} is the length of the pitch moment arm, K_{pitch} is the pitch stiffness, and D_{pitch} is the pitch damping.

Tire normal forces consist of four parts: the static vertical load F_s , the longitudinal load transfer due to acceleration F_p, the lateral load transfer due to rolling F_r and the lateral load transfer due to acceleration of the unsprung mass F_u. The tire normal force of the left front tire is given by

$$\begin{aligned}F_{z1} = F_{s1} + F_{p1} + F_{r1} + F_{u1} \\ = \frac{-hF_{drag} - \left(m_c h + m_{uf}h_{uf} + m_{ur}h_{ur}\right)gsin\gamma + mgL_r cos\gamma}{2WB} \\ - \frac{\left(m_c h + m_{uf}h_{uf} + m_{ur}h_{ur}\right) * a_x}{2WB} \\ + \frac{\left(a_y cos\phi + gsin\phi\right) * m_c d_{roll} * K_{rollf}}{TW_f\left(K_{rollf} + K_{rollr}\right)} + \frac{m_c a_y h_{uf} * \frac{L_r}{WB} + m_{uf}a_y h_{uf}}{TW_f}\end{aligned} \tag{2.8}$$

where m_c is the sprung mass, m_{uf} and m_{ur} are the front and rear unsprung mass, h is the height of sprung mass CoG, h_{uf} and h_{ur} are the height of front and rear unsprung mass CoGs, WB is the wheel base, and γ is the road inclination angle. The normal loads on other tires are similar. The values of some vehicle parameters are shown in Table 2.1.

Table 2.1 Vehicle parameters

m	1450 kg	h	0.5 m
TW_f	1.585 m	L_F	1.07 m
TW_r	1.575 m	L_R	1.605 m
m_{uf}	95 kg	m_{ur}	95 kg

2.2.3 Tire model

Tires are one of the most important parts of vehicles and tire modelling is crucial to the accuracy of simulation. Various tire models have been developed, including ring tire models [66, 67] and brush tire models [68]. In this study, Pacejka tire model (Magic Formula) [69] is used for the calculation of tire dynamics since it has proved to be reasonably accurate and computationally fast. One critical term in the Pacejka tire model is the slip ratio, which measures the difference between the rotational speed of the wheel and the translational velocity of the wheel center. To avoid singular values at low speed, the slip ratio is defined as:

$$\sigma_i = \begin{cases} \dfrac{\omega_i r - V_{xi}}{|V_{xi}|} & |V_{xi}| > V_{threshold} \\ \dfrac{\omega_i r - V_{xi}}{V_{threshold} + \dfrac{(\omega_i r)^2}{V_{threshold}}} & |V_{xi}| < V_{threshold} \end{cases} \tag{2.9}$$

where ω_i is the rotational velocity of the wheel, r is the effective radius of the wheel and V_{xi} is the longitudinal velocity of the wheel.

2.2.4 Traffic simulation

Traffic simulation is a relatively effective approach to capture the patterns for abnormal vehicle behaviors and analyze the effects of vehicle problems. A complete traffic simulation has three parts. First, environmental conditions are determined. Then, different kinds of vehicle faults are simulated and added to the simulation. Finally, a driver model to mimic the reaction of real drivers in this scenario is built.

In this study, it is assumed that drivers have specific driving patterns in certain traffic scenarios. To generate representative vehicle data that can cover the whole space of normal and abnormal behaviors, several typical traffic scenarios need to be chosen. This section focuses on longitudinal car following, in which drivers only control the gas pedal and brake pedal. Numerous car-following models have been developed since the 1905s. Gipps [45] proposed a behavioral car-following model based on the assumption that each driver sets the limits of his desired braking and acceleration rates. Lee et al. [46, 47] compared a few longitudinal driver models and suggested that the Gipps' model was the most promising. They also modified the Gipps' model by considering that the following driver would take time to observe the deceleration of the lead vehicle. Modified Gipps'

model was adopted because its calculated distances between two cars are closer to the assumed safe distance, which is one car length per ten miles per hour. However, the Modified Gipps' model can only estimate the velocity of the following car according to the lead car's velocity. To generate reasonable throttle/brake position data, a feedback control system is applied, as shown in Figure 2.5.

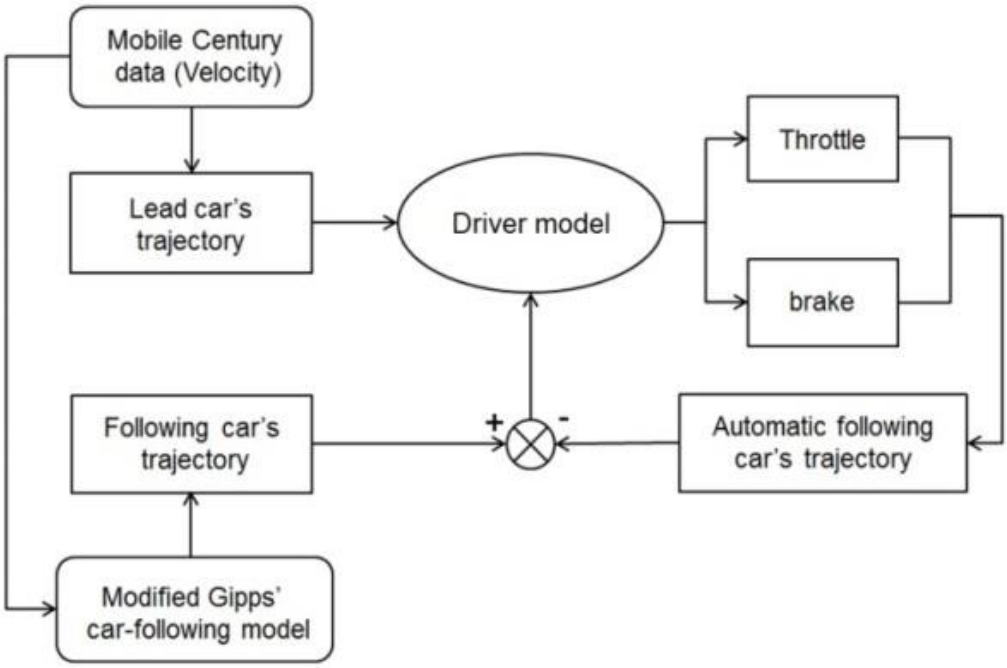

Figure 2.5 Car-following scheme

The velocity of the lead car was obtained from the Mobile Century data [70], which collected vehicle velocities on one section (NB and SB) of highway on Interstate 880 in California. Ninety cars were chosen from the data set. Interpolation was performed to synchronize the data of different cars.

The velocities from the Mobile Century data were integrated to get the trajectories of

lead cars. They were also sent to Modified Gipps' car following model to calculate the velocities of the following cars, which were then used to calculate the trajectories of the following cars. The automatic driver controlled throttle positions and brake positions by comparing Gipps' trajectories and automatic trajectories. The trajectories of the lead cars were also sent to the automatic driver to further determine throttle/brake positions and prevent accidents.

2.2.5 Driver model

The driver model plays an important role in the traffic simulation and directly affects the representativeness and comprehensiveness of the generated data. This study concentrates on generating realistic throttle position and brake position data rather than minimizing the tracking error. Therefore, several logical assumptions were made to mimic a rational human driver, which are shown in the following.

a) Safe distance = one car length per 10 miles per hour.
b) The gas pedal and brake pedal are not pressed at the same time. Since it is easy to detect if the gas pedal and brake pedal are pressed at the same time, this kind of issue can be dealt with separately and is not considered in this section.
c) If $V_{following} > V_{lead}$ when $headway \approx safe\ distance$, the driver of the following car begins to brake. Otherwise, the driver stops pressing the accelerator pedal and lets the car decelerate.
d) The time for moving between the gas pedal and brake pedal is around 0.6 s.
e) Brake position is defined as

$$Brake\ position = 1 - e^{-\theta} \tag{2.10}$$

where θ is obtained from

$$\theta = \frac{V_{following} - V_{lead}}{Distance\ \text{headway}} \tag{2.11}$$

This equation simulates the process that the brake position increases as the headway distance decreases, and the velocity difference between the following car and lead car increases.

f) Throttle positions are given by

$$u(t+\tau) = u(t) + K_1 * error(t+\tau) + K_2 * (error(t+\tau) - error(t)) + K_3 * (\frac{derror(t+\tau)}{dt} - \frac{derror(t)}{dt}) \tag{2.12}$$

where τ is the human driver reaction time. The error is defined as

$$error(t) = X_{Gipps}(\text{t}) - X_{simulation}(t) \tag{2.13}$$

where X_{Gipps} is the position of the Gipps' model and $X_{simulation}$ is the position of the driver model.

In summary, driver behaviors are simulated using a logical model and the amount of throttle position change is modelled using a discrete PID controller. The simulation result is shown in Figure 2.6. Blue stars represent the trajectory of the lead car. Green circles represent the trajectory of Gipps' model. Red crosses represent the trajectory of the real following car. The automatic following car can track the trajectories of the

Gipps' model without crashing.

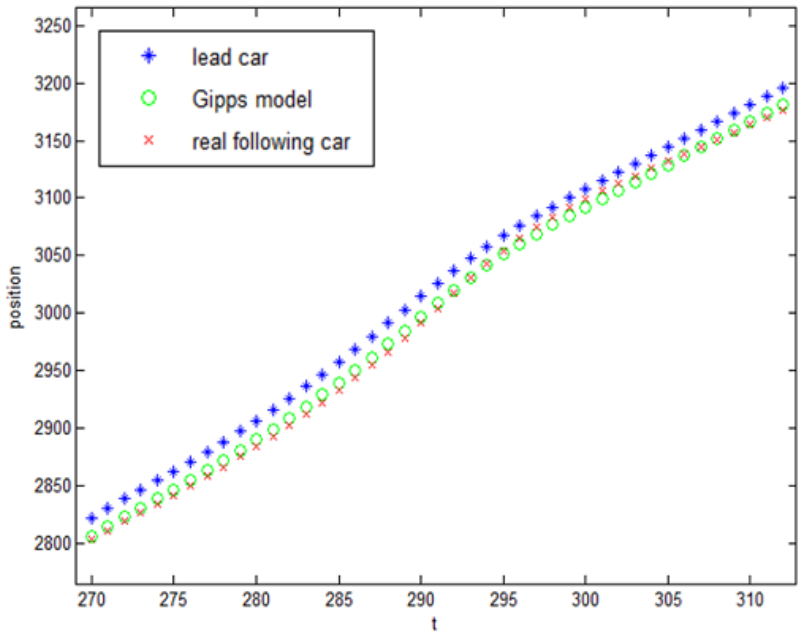

Figure 2.6 Automatic following

2.2.6 Simulation of vehicle problems and experiments

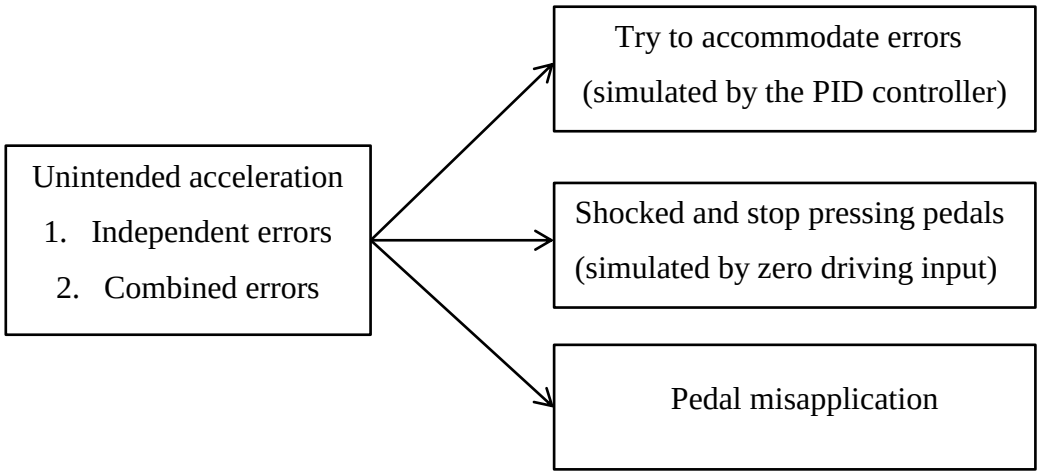

Figure 2.7 Common drivers' reactions to UA

Vehicle problems were simulated to help find the boundary of normal and abnormal vehicle behavior. This study focuses on the detection of UA. Since UA is often accompanied by loss of the brake control, brake problems are also taken into account. Figure 2.7 shows some common drivers' reactions to UA. The first two cases are caused by vehicle problems. The third one is due to human error. In this case, some human features such as drivers' heart rates are needed for reliable prediction. This section studies the first two cases that result from vehicle problems.

The vehicle problems are simulated in two ways. The first one is by generating random errors, which are defined as

$$real\ V_{frontwheel} = V_{frontwheel} + 0.1\ \ (m/s) \tag{2.14}$$

$$real\ brake\ torque = brake\ torque * \frac{1}{2}\ (N * m) \tag{2.15}$$

UA is simulated by giving an instantaneous increase of the velocity to drive wheels. Brake problems are modelled as a decrease of the brake torque. The real brake torque is equal to half the original torque, which means that the driver has to press the brake pedal very hard to stop the car.

The second approach to modelling the faults considers the vehicle-driver interaction, as shown in Figure 2.8. It is a combination of brake problems and UA. When the driver presses the brake pedal, there is a sudden increase in acceleration. At the same time, the driver loses control of the brake so that the real brake torque is zero. The next part simulates the hesitation of the driver experiencing UA. There is neither brake input nor throttle input for a short period of time, which assumes that the driver is shocked and has

no idea what to do. Finally, everything goes back to normal.

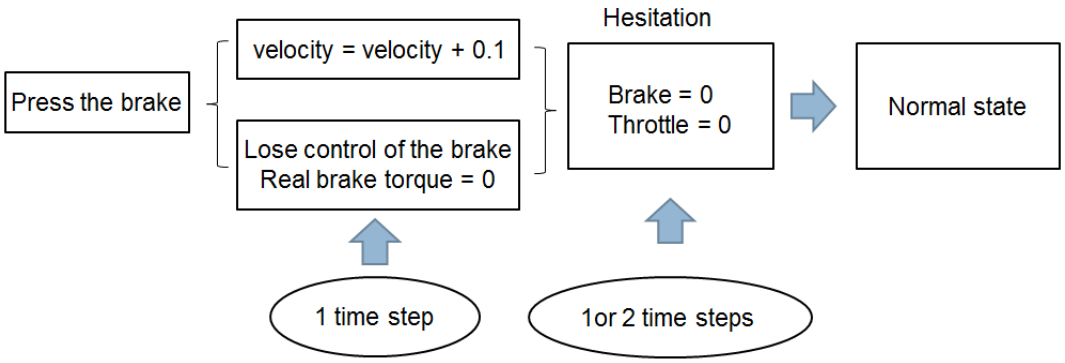

Figure 2.8 Combined faults

Finally, experiments were conducted in Simulink to collect data for both normal and abnormal driving. Each data set records the data of ninety cars and each car has three hundred and thirty time points. Sensor noise was also simulated and added.

2.3 Collect data from virtual vehicles in dSPACE

At the second stage, experiments were designed and conducted to collect data in the Simulink/dSPACE environment. dSPACE software allows a human driver to control a virtual vehicle using analog steering wheels and pedal in a 3D environment. It could be connected to Simulink so that features such as positions of vehicles could be extracted and recorded. A complete experiment process contains four parts, which are road design, traffic simulation, error and noise simulation and data collection.

2.3.1 Road design

Road types play a crucial role in driving. Vehicles' acceleration is affected by the curvature and slope of the road. Moreover, human drivers may change their driving

styles under different road conditions. To collect representative data, typical road environments should be covered. The road in the test has two lanes. One is the inner lane and the other is the passing lane. The road is designed to contain several straight sections and curved sections. The road grade also varies. Some parts are steep while others are flat. In the steep part, a car is able to have a considerable increase of acceleration without its driver's inputs.

2.3.2 Traffic model

Another critical factor in the driving environment is the traffic condition. Driving patterns of a human driver are affected by cars nearby. UA is extremely dangerous in the dense traffic. To perform realistic simulation and a comprehensive study of UA, a traffic model should be well constructed. Besides the car driven by the human driver, several virtual cars were created to move autonomously under several logic rules, as shown in Figure 2.9.

Initially, these cars move at preset desired velocities, which were obtained from the Mobile Century data [70]. The Mobile Century data set was collected from vehicles on Highway I-880 in California. Desired velocities are updated according to this data set in real time. In such a way, the desired speed of these virtual cars changes over time, which is similar to real cars. A virtual car starts to check if there is enough space for changing lanes when there is a car ahead and its velocity is much lower than the desired speed. If there is no space for changing lanes, the car switches to the following mode. Meanwhile, cars are not allowed to stay in the passing lane all the time. The traffic model was tested in dSPACE and proved sufficient to mimic the real traffic in this work.

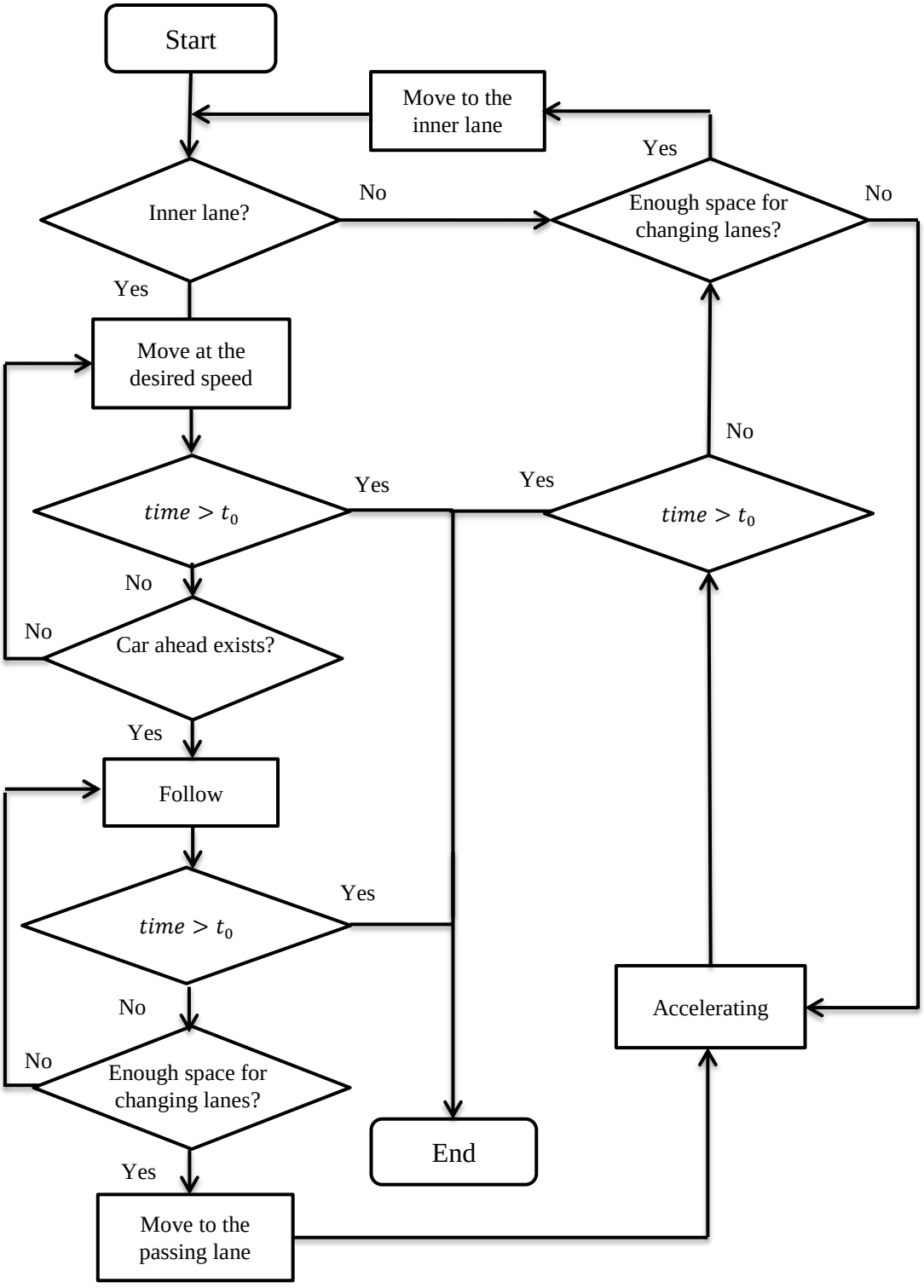

Figure 2.9 Traffic model

2.3.3 Simulation of unintended acceleration and sensor noise

As discussed above, UA refers to an unexpected sudden increase of acceleration. It could be caused by software failure. In this work, UA was simulated by adding an error to gas pedal positions randomly, as shown in

$$GP_{real} = min(GP_d + GP_{error}, GP_{max}) \tag{2.16}$$

where GP_d is the gas pedal position input from the driver. GP_{real} is the real gas pedal position value received by the electronic control unit. It is the sum of GP_d and an error and has an upper bound GP_{max} of one hundred. GP_{error} is an error function. The range of this error is between zero and fifty. The duration of the error should be long enough to make sure that drivers can notice abnormal vehicle behavior. The error was finally programmed to last for three seconds every time. Since the error was added randomly, the real duration of the error was a multiple of three seconds. It should be noted that Equation (2.16) actually works by adding a random error to the engine output torque, as shown below.

$$T_{engine} = T_{GP_d} + T_{error} \tag{2.17}$$

where T_{engine} is the engine output torque, T_{GP_d} is the torque caused by the driver's input and T_{error} is the torque caused by GP_{error}. The map between GP_{error} and T_{error} is nonlinear. The whole equation tries to reproduce the problem in the linkage between pedals and the engine. For example, when the ECU is interfered with extraneous signals, it may send a wrong signal to increase the throttle position while the driver does not

press the gas pedal. Another important part in the fault detection is the sensor noise. A poor detection system may generate false alarms due to sensor noise. In order to make the simulation more realistic, white noise is added to noisy sensors like accelerometers.

2.3.4 Experiments

The final step of data collection is conducting experiments. A front-wheel-drive virtual car with automatic transmission was created in dSPACE for tests because UA often appeared in this kind of cars. There were 120 tests in total. In each test, a human driver controlled a virtual car through analog steering wheels and pedals, as shown in Figure 2.10.

Figure 2.10 dSPACE simulator

Important signals were chosen and recorded at a sampling rate of 50 Hz. The selected data channels are accelerator pedal position, throttle position, brake pedal

position, steering angle, rpm, overall velocity, yaw rate, longitudinal acceleration, longitudinal velocity, lateral velocity and vertical velocity. The yaw rate and longitudinal acceleration can be measured or estimated by the sensors of electronic stability control system (ESC). Since most modern vehicles are equipped with ESC, it should be relatively straightforward to measure these two states. The longitudinal velocity, lateral velocity and vertical velocity can be measured by a GPS receiver.

2.4 Building detection systems

A relatively large data set was collected in experiments, but the data channels are very noisy and the data do not directly identify UA. To find the boundary between normal and abnormal vehicle behavior, certain information that is relevant to UA needs to be extracted and combined. There is a variety of ways to generate rules from the data and design models. Different detection models are compared by two major indicators, as shown in (2.18) - (2.19). The first indicator is the detection rate. An error is regarded as detected if at least one time point of it is found abnormal by the system. It should be noticed that the system could detect multiple points of an error. Therefore, the number of correctly detected points is larger than or equal to the number of detected errors. The second indicator is precision. The system gives one false alarm when one normal time point is considered abnormal.

$$Detection\ rate = \frac{N_{detected}}{N_{error}} \tag{2.18}$$

$$Precision = \frac{N_{detected}}{N_{detected} + N_{falsealarm}} \tag{2.19}$$

where $N_{detected}$ is the number of detected errors, N_{error} is the number of errors and $N_{falsealarm}$ is the number of false alarms.

2.4.1 Smoothing

Though raw data reflect the trend of driver behavior and vehicle behavior to some extent, they are still inadequate for detecting UA. First, sensors such as accelerometers are very noisy. The noise interferes in deciding if UA happens. Second, drivers had diverse driving patterns. One example is shown in Figure 2.11.

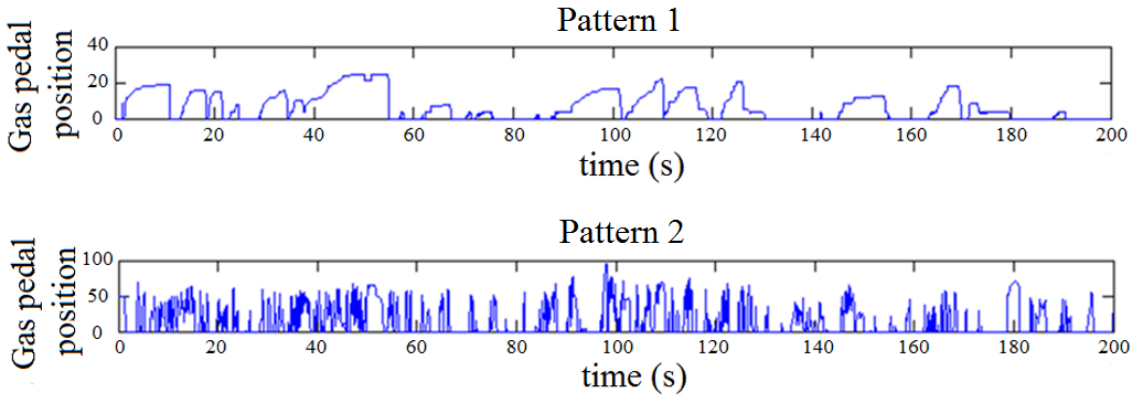

Figure 2.11 Different driving patterns

Driver one changed throttle positions smoothly while driver two changed throttle positions frequently. It was found that it was very difficult to detect UA using the data from drivers who changed their inputs at high frequencies. The reason might be the decrease of correlation between pedal positions and other vehicle states when the pedal positions changed frequently. To address the above problems, it is essential to find an approach to capturing important patterns of the data and leaving out noise or other rapid

phenomena in real time. However, not all the signals are noisy and need to be filtered. Additionally, many filters introduce a lag to the signal while filtering. As a result, there will be a mismatch between different data channels after filtering, which severely disrupts classification. In this work, oversampling is combined with exponential weights to reduce the impact of mismatch, which is shown in Figure 2.12.

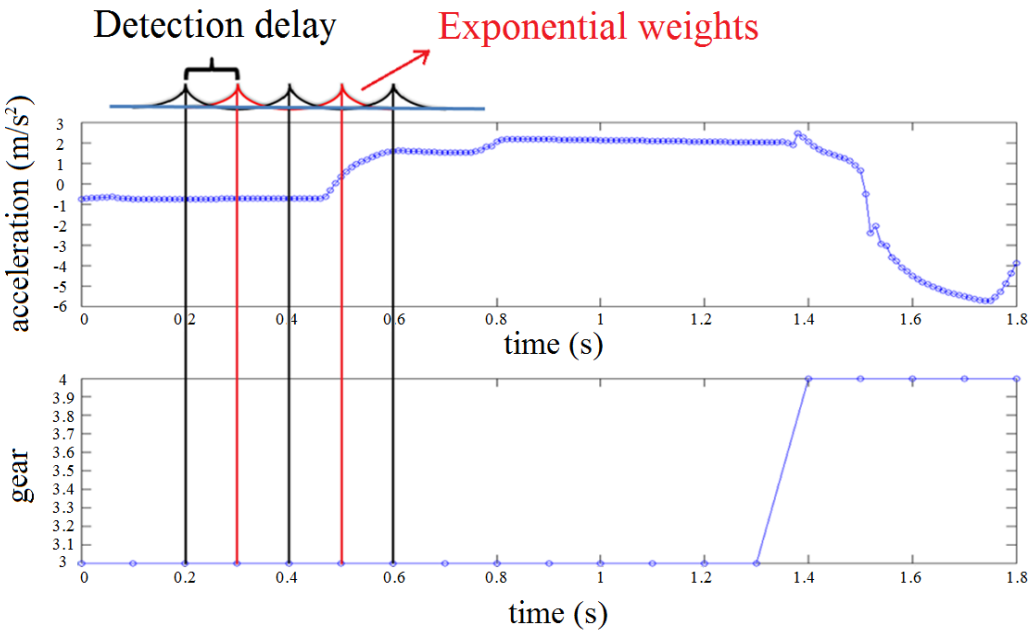

Figure 2.12 Smoothing

The sampling rate of noisy signals is set to be much higher than that of other signals. Exponential weights are assigned to noisy signals and the average value of each section is calculated and stored. This process actually combines a forward moving average filter with a backward moving average filter. The lead and lag introduced by two filters cancel each other out. It is similar to the approach proposed by Chen et al. [71]. In their research, a filter was developed for offline application. In this work, exponential weights

are applied in real-time detection. It causes a little detection delay because the system can only decide if UA occurs after the backward filtering of the current section is finished. However, the delay is very short and the smoothing process can greatly reduce the mismatch between data channels and improve the detection result.

2.4.2 Feature extraction and selection

The boundary between normal and abnormal behavior is concealed in abundant data points. Model-based fault diagnosis methods generate residuals or symptoms to detect faults. Similarly, features that are relevant to faults need to be extracted from the data set in data-driven approaches. One advantage of the data-driven approaches is that complex features can be handled even if the relationships between features are not completely available.

The first few features were selected by analyzing vehicle dynamics, as shown in Figure 2.4. It is assumed that UA results from a problem in the linkage between pedals and the engine. Therefore, tire forces increase when UA occurs. Though tire forces cannot be measured directly, vehicles' acceleration and other states can be estimated. Acceleration is an important vehicle variable in this work. To take it one step further, longitudinal acceleration is combined with other variables to show a vehicle state that is relevant to UA. The modified acceleration is calculated by

$$\begin{aligned} MA &= a_x - V_y\omega + \frac{F_{slope}}{m} \\ &= a_x - V_y\omega + gsin(\theta_{slope}) \end{aligned} \tag{2.20}$$

where g is the gravitational acceleration and θ_{slope} is the slope angle. a_x, V_y and ω can

be measured by vehicle sensors directly. However, the road grade is usually not available. Research efforts have been devoted to the road grade estimation. Ohnishi et al. [72] tried to estimate the road grade by comparing the differential of the wheel speed with the longitudinal acceleration of the vehicle. However, this method actually obtains the sum of the road slope and the pitch motion, which is not accurate. They also estimated the road grade by comparing the results of the vehicle's longitudinal model and the wheel output torque. However, this approach does not work while braking and gear shifting. Sebsadji et al. [73] presented a method to reconstitute the road slope using an Extended Kalman Filter and a Luenberger observer. Bevly [74] demonstrated the capability of a standard low-cost GPS receiver to estimate the slope angle of the road. In this research, the car is assumed equipped with a GPS receiver. The slope angle of the road is estimated by

$$\theta_{Slope} = tan^{-1}\left(\frac{-V_z}{\sqrt{V_x^2 + V_y^2}}\right) \tag{2.21}$$

where V_x, V_y and V_z are the longitudinal velocity, lateral velocity and vertical velocity.

The modified acceleration MA can be calculated after the road grade is estimated. It is used to represent the current vehicle state. However, a single acceleration value is not sufficient to represent the vehicle state accurately. According to the Takens' time-delay embedding theorem, a chaotic dynamic system can be reconstructed from the time-delayed versions of one generic state of the system [75]. This theorem was also extended to deterministic systems forces by some stochastic process [76-78]. In light of these

theorems, the time-delayed versions of acceleration would probably be capable of representing the current vehicle state. The embedding dimension of the data set is evaluated by the false nearest neighbors (FNN) [79].

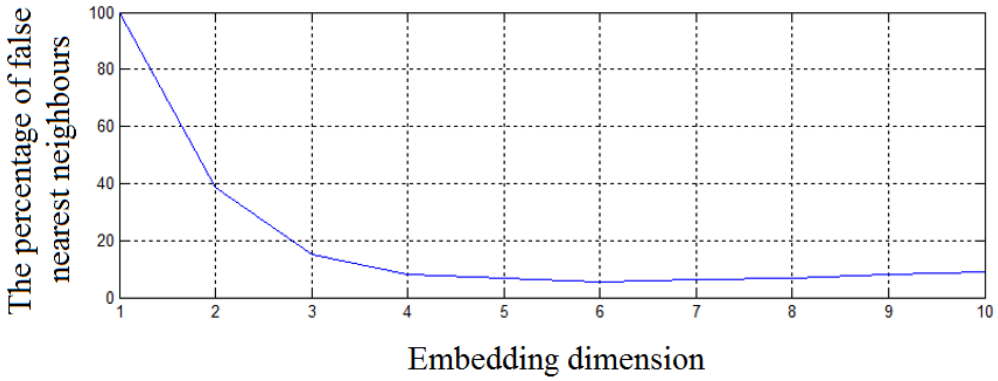

Figure 2.13 Estimating the embedding dimension

The main idea of FNN is that a sequence of observations of the state forms a signal trajectory in an n-dimensional space. Successive points will be mapped onto neighbors in the delay space due to the smoothness of the dynamics. In a low embedding dimension, many of the neighbors will be false. However, these false neighbors will no longer be neighbors as the embedding dimension increases. The test result of FNN is illustrated in Figure 2.13. It shows that the embedding dimension of the data set should be at least three to get an acceptable percentage of false nearest neighbors.

The second important term is the aerodynamic drag, which is given by

$$F_{drag} = \frac{1}{2}\rho v^2 C_D A \tag{2.22}$$

where ρ is the air density, v is the relative speed of vehicles, C_D is the drag coefficient

and A is the cross sectional area. Velocity squared is proportional to the aerodynamic drag and it is chosen as a feature. In this work, it is assumed that the wind speed is zero. When the wind speed is high, an extra sensor to estimate the wind speed is needed to recalculate the relative vehicle speed.

Other primary vehicle inputs, which affect vehicles' acceleration, are gas pedal positions, brake pedal positions and gear shifting. The engine time lag and brake time lag τ_b should also be considered. The engine speed response is dominated by the rotor inertia. The lag caused by the dynamics of combustion process is very small compared to the primary engine time constant τ_e [80]. Though τ_e and τ_b vary in different operating environments, their ranges are relatively small. Rough estimates of average τ_e and τ_b are enough in this approach. Gear shifting is also a key factor. Some cars have the gear shift shock, which may interfere with the detection of UA. Gear positions are not directly accessible for some types of cars. In this case, this feature can be obtained by monitoring the gear ratio, which is represented by

$$GR = \frac{RPM}{Velocity} \tag{2.23}$$

In summary, each time point t is regarded as an observer. If UA happens at this time point, the label of this point is one. Otherwise, this point is labelled as zero. The features of each time point are shown below.

$$GP_d(t-\tau_e), GP_d(t-\tau_e-\tau), GP_d(t-\tau_e-2\tau)$$

$$BP_d(t-\tau_b), BP_d(t-\tau_b-\tau), BP_d(t-\tau_b-2\tau)$$
$$V^2(t), V^2(t-\tau), V^2(t-2\tau)$$
$$MA(t), MA(t-\tau), MA(t-2\tau), MA(t-3\tau) \quad (2.24)$$

where τ is the time gap between two successive points, GP_d denotes the driver's gas pedal input, τ_e is the engine time lag, BP_d is the driver's brake pedal input, τ_b is the brake time lag, V^2 represents velocity squared and MA denotes the modified acceleration. Besides the above features, other variables such as the frequency components of steering angles and lateral acceleration were also taken into account. Different combinations of features were tested using machine learning models. The above set of features proved to have the best performance.

2.4.3 Initial test

Signals were smoothed and features and labels of all the time points were calculated. An initial test was conducted to compare the performance of several common machine learning models, including K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), Naïve Bayes, Random Forest and Multilayer Perceptron (MLP) [81]. The data set was normalized for some algorithms according to their needs. Then it was divided into training and test parts randomly. One hundred trials were used to train the model and the remaining twenty trials were sent to test the model. Classification results are shown in Table 2.2. It can be seen that although detection rates are acceptable, the precision is too low. It is not too complicated to design rules to detect UA. Nonetheless, the rules generated by a sophisticated detection model should not only be able to detect

faults, but also avoid false alarms under different conditions. If a detection system has too many false alarms, customers may just turn it off instead of using it.

Table 2.2 Initial test results

	Detection rate	Precision
KNN	99.19%	25.95%
LDA	98.26%	22.50%
Naïve Bayes	97.82%	10.87%
Random Forest	98.14%	26.00%
MLP	93.23%	39.34%

Among the tested algorithms, LDA is a linear model. But the dynamics of vehicles are highly nonlinear. That may be reason why LDA has low detection precision and it is not a good choice to monitor vehicles. Naïve Bayes assumes independence of features and is not suitable in this task. KNN does not really generate new rules from the training data and is not robust without a considerable amount of data. Random Forest requires a large number of trees and is computationally time-consuming. It may not be appropriate in real-time prediction of UA. MLP has the highest precision and is capable of solving nonlinear problems. It is adopted in the following work.

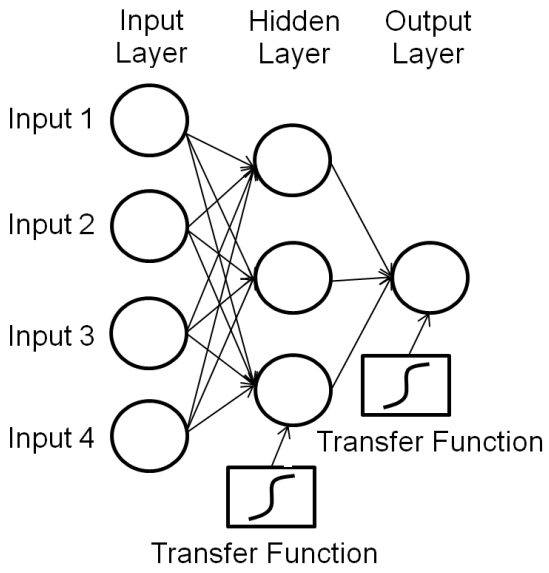

Figure 2.14 Multilayer perceptron

MLP is one of the well-developed artificial neural-network algorithms. It has been widely used in classification and control fields [82-84]. The structure of MLP is illustrated in Figure 2.14. It emulates the central nervous system by creating a high-dimensional mapping from the input layer to the output layer. Weights between neurons are tuned based on the training data. Transfer functions are placed after hidden layers and the output layer. The last transfer function of the neural network model is a threshold and is adjustable. An error is considered to occur when the calculation result of previous steps is higher than this threshold.

2.4.4 Multilayer sequential detection

It can be seen from Table I that applying ML algorithms to data processing directly has poor performance. The precision is too low. It is reasonable because one data-driven

model is not powerful enough to represent the whole vehicle dynamics. To detect UA, representative time points need to be chosen from the whole time series.

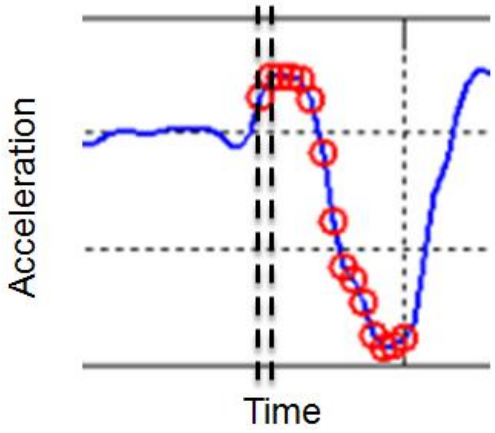

Figure 2.15 Acceleration

Figure 2.15 shows an example of the acceleration changes when UA happens. Red circles mean errors occur at those time points. UA usually starts with an obvious increase of acceleration. After that, the aerodynamic drag increases as the velocity increases. Meanwhile, the driver may press the brake pedal. Consequently, the acceleration of the vehicle will reach a peak value in a short period of time. Afterwards, the acceleration may decrease remarkably or go to a steady value, which makes it difficult to differentiate normal and abnormal vehicle behavior. Therefore, the rising edge of acceleration contains the most obvious symptoms of UA and is the easiest time interval to detect UA accurately and quickly. The beginning of UA can be detected by analyzing big increases of the acceleration. The end of the rising edge can be detected by comparing acceleration peaks. The whole detection scheme is illustrated in Figure 2.16.

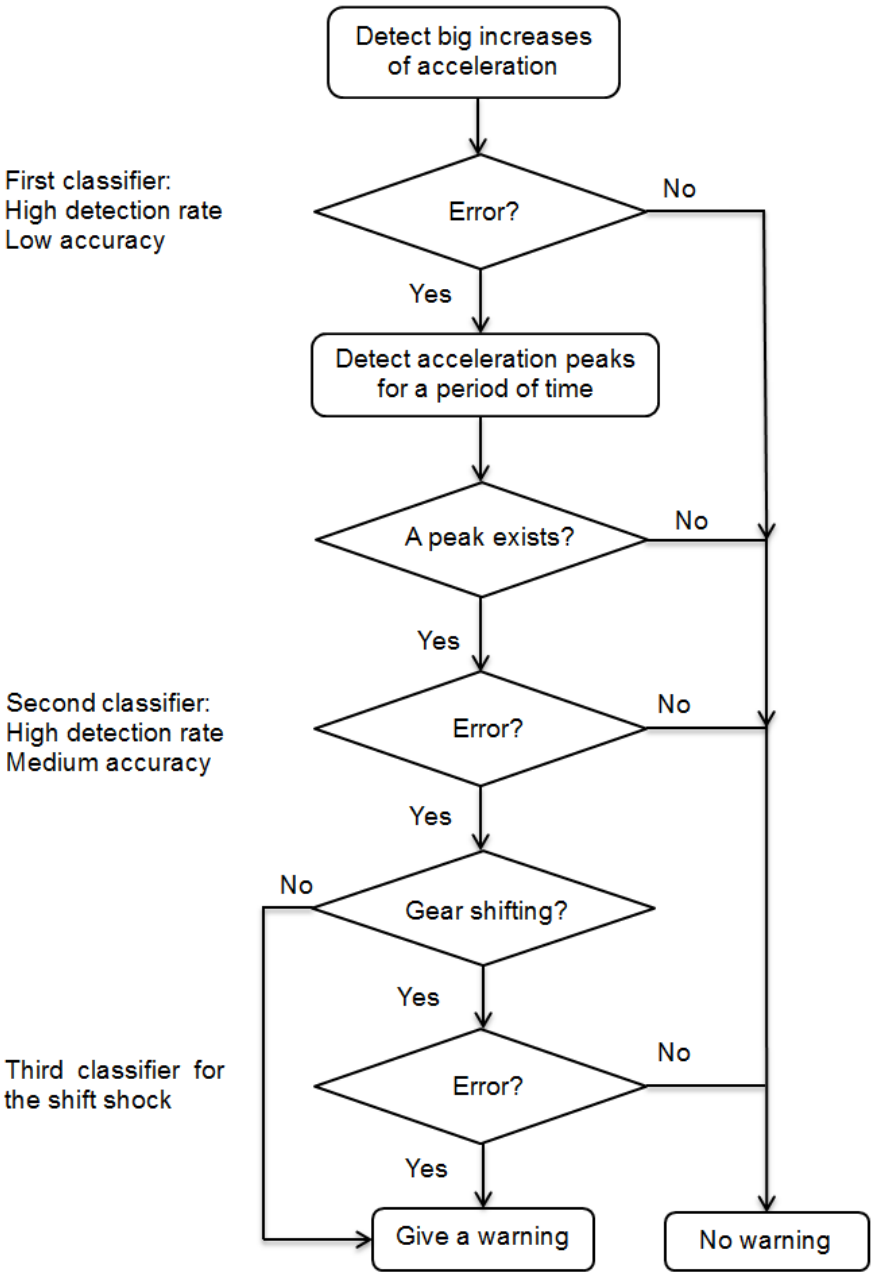

Figure 2.16 Detection scheme

Three models were built using the training data. The first classifier was built by analyzing time points where acceleration increases obviously. Its final transfer function was adjusted to ensure a high detection rate regardless of its accuracy. The second

classifier was trained by time points at acceleration peaks. Its final transfer function was adjusted to keep a balance between the detection rate and accuracy. During the test, it was found that a single classifier could not distinguish acceleration peaks caused by the shift shock from those caused by UA and generated false alarms. To solve this problem, a third classifier was created by analyzing time points at gear shifting.

When the system starts, it tries to detect big increases of acceleration at first. The features of time points where acceleration increases rapidly are sent to the first classifier. This classifier works like a filter and most normal time points cannot pass through it. If the first classifier doubts that an error might occur, the system begins to detect acceleration peaks. If an acceleration peak is found in a short period of time, the features of this peak are sent to the second classifier. If the second classifier also realizes a high possibility of UA, the system checks if this acceleration peak is caused by the shift shock using the third classifier. If the possibility of the shift shock is ruled out, the system gives a warning to the driver.

It may be doubted that if such a complicated data-driven system is necessary to detect UA. A straightforward idea to detect UA is described in Equations (2.25) - (2.26). Generally speaking, there is a strong possibility that UA happens if there is a substantial increase in acceleration while the gas pedal position does not increase obviously. Similarly, UA is quite likely to occur if the acceleration increases as the brake pedal position increases.

If

$$GP_d(t) - GP_d(t-\tau) \leq a$$
$$MA(t) - MA(t-\tau) > c \tag{2.25}$$

Or

$$BP_d(t) - BP_d(t-\tau) > b$$
$$MA(t) - MA(t-\tau) > c \tag{2.26}$$

then give a warning.

The above equations contain four parameters, which are a, b, c and τ. Extensive search was adopted to find the optimal parameter values. Various combinations of parameter values were tested one by one and their detection rates and precision were all calculated. Some illustrative results are shown in Table 2.3.

Table 2.3 Set thresholds directly

	Detection rate	Precision
Test 1	13.59%	69.74%
Test 2	31.54%	33.61%
Test 3	93.59%	1.59%

It can be seen that the performance of these equations is very poor. When the

thresholds are set big enough to obtain a satisfactory detection rate, many false alarms are sent. When conservative thresholds are defined to guarantee high precision, only a small number of errors can be detected. It is understandable because vehicle systems are very complicated. Besides the above two rules, many factors like gear shifting also play an important role. Moreover, constant thresholds are not sufficient to cover nonlinear characteristics of vehicles. In addition, estimation of the road grade using a GPS receiver fluctuates sometimes and deviates from the real values. This error plus sensor noise increases detection difficulties. In summary, it is quite difficult to manually design a complete set of rules that separate UA from normal behavior accurately. Setting thresholds directly can only fulfill the demand of some low-dimensional simple problems.

2.5 Validation and discussion

This section presents the test results of the proposed detection algorithm. Both the data collected from vehicles controlled by the driver model in Simulink and the data collected from virtual cars in dSPACE were used to validate the proposed detection algorithm.

2.5.1 Validation using data collected in Simulink

At first, data sets extracted from traffic simulation in Simulink are used to validate the detection system. Each data set records the data of 90 cars and each car has 330 time points. Sensor noise is also simulated and added. Finally, each data set is divided into the training part and test part. Sixty-three cars are chosen randomly for training. The remaining twenty-seven cars are used in the test.

The first data set only contains UA errors. The result is shown in Table 2.4. There are 67 errors in 8910 time points. The detection system gives 66 warnings and 63 of them are correct.

Table 2.4 Independent errors

	Condition positive	Condition negative
Test outcome positive	TP = 63	FP = 3
Test outcome negative	FN = 4	TN = 8840

The second data set uses a combination of UA errors and brake problems. The driver's reaction is also considered. Classification result is shown in Table 2.5. There are 49 errors in 8910 time points. The detection system gives 47 warnings and 46 of them are correct.

Table 2.5 Combined errors

	Condition positive	Condition negative
Test outcome positive	TP = 46	FP = 1
Test outcome negative	FN = 3	TN = 8860

To further validate the model, the detection rates are defined as

$$DR_{UA} = \frac{TP}{TP + FN} \tag{2.27}$$

$$DR_{normal} = \frac{TN}{TN + FP} \tag{2.28}$$

$$DR_{overall} = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.29}$$

where $DR_{UA}, DR_{normal}, DR_{overall}$ are the detection rates of UA, normal vehicle behaviors and the whole data set. Each test is repeated 30 times to obtain both the mean and standard deviation of the classification accuracy. The detection rates are summarized in Table 2.6. The average detection rates range from 92% to 99%.

Table 2.6 Simulink validation

Detection rate	Independent errors	Combined errors
Normal behavior	99±0.64%	99±0.81%
Unintended acceleration	93±2.34%	92±3.68%
Overall	97±0.92%	96±1.17%

2.5.2 Validation using data collected in dSPACE

Data sets collected in dSPACE are used to future validate the detection system. The detection rate and precision are calculated and compared. Apart from the detection rate and precision described in Equations (2.18) - (2.19), the reaction time is important for a detection system. A well-designed detection system should be able to detect errors as soon as possible and give ECU or drivers enough time to solve the problem and prevent accidents. The reaction time of the system is defined as

$$Reaction\ time = t_{firstwarning} - t_{errorbeginning} \tag{2.30}$$

where $t_{firstwarning}$is the time of the first warning and $t_{errorbe}$ginning is start time of errors.

Two methods were used to evaluate the algorithm. In the first part, the original data set was divided randomly to test the proposed algorithm. In the second part, the whole detection system was embedded into Simulink and tested by human drivers in real time. Table 2.7 shows 95% confidence intervals of the results of detection. Two tests were conducted. The first one chose one hundred trials for training and twenty trials for the test randomly. It is the same as the information in Table 2.3. As can be seen, both the detection rate and precision of the improved system are higher than those of using MLP directly. Meanwhile, warnings can be given within one second. The second test aimed to analyze if the generated model was robust. In each round, the data set of one driver was used to validate the model while the remaining data were used to train the model. This process was repeated twenty times so that each driver was used in the validation once.

The results show that both detection rates and precision are beyond 95% and all the errors were detected within one second. The system succeeds in separating abnormal acceleration from normal vehicle behavior.

Table 2.7 dSPACE validation

	Detection rate	Precision	Reaction time (s)
Training: 100 trials Test: 20 trials	97.21±0.55%	99.57±0.35%	0.82±0.19
leave-one-out cross-validation	97.83±1.50%	99.65±0.49%	0.81±0.17

The second major part of the validation comprises real-time tests of data-driven models generated from the original data set. The whole detection system including the smoothing unit and classification unit was embedded into Simulink. In these tests, signals were processed during the simulation and they were totally independent of the data used to train models. The error was a square wave whose amplitude was fifty. To have a thorough evaluation of the system, UA was programmed to occur under different conditions. Test results are shown in Figure 2.17 - Figure 2.19.

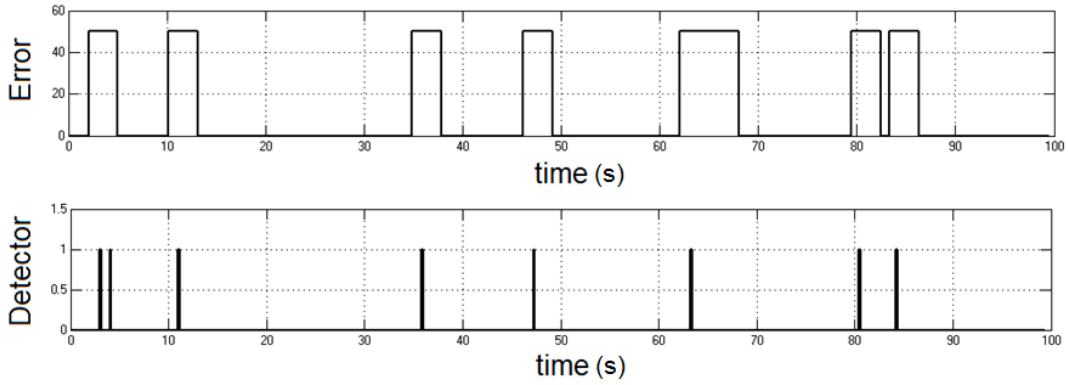

Figure 2.17 UA occurs when no pedal is pressed

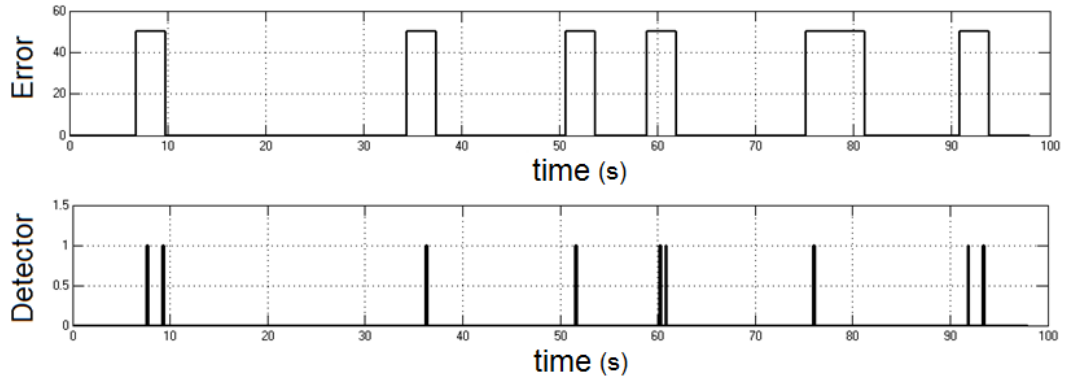

Figure 2.18 UA occurs when the brake pedal is pressed

In the first test, UA only occurred randomly when no pedal was pressed. In the second test, UA was programmed to occur randomly when the brake pedal was pressed. The braking force was adjusted to mimic an ineffective brake. It can be seen from Figure 2.17 and Figure 2.18 that the system gave as least one warning to an error and there was no false alarm.

In the third test, UA occurred randomly when the gas pedal position increased, which was the most difficult to detect. A ramp input was sent to the gas pedal to make sure that the pedal position kept increasing when the error appeared. It can be seen from Figure

2.19 that the system still succeeded in distinguishing the abnormal acceleration introduced by an error from the normal acceleration caused by pressing the gas pedal.

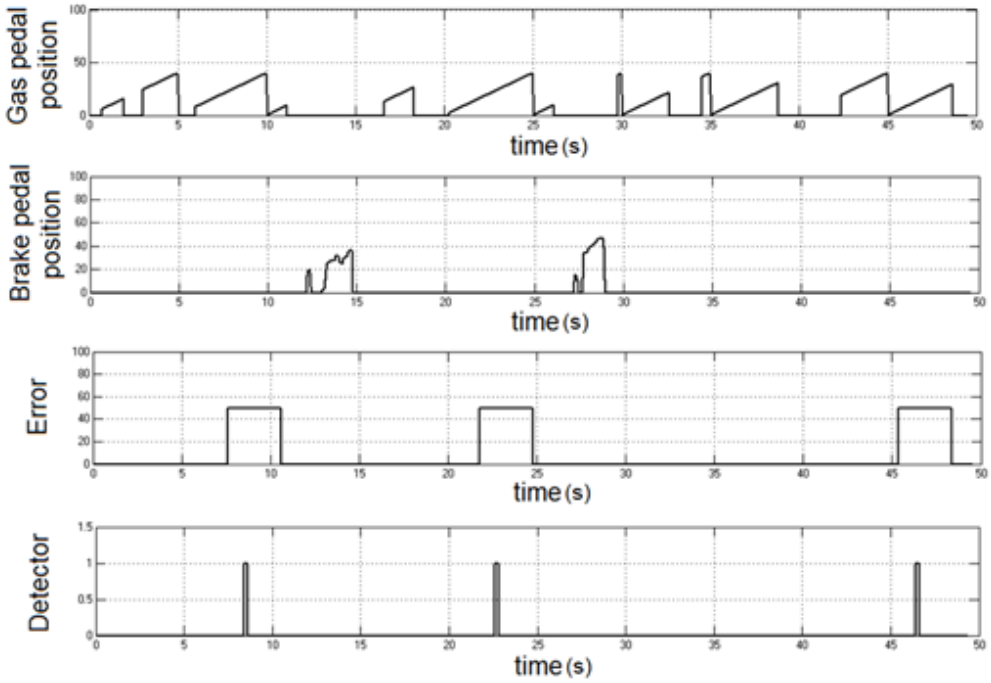

Figure 2.19 UA occurs when the gas pedal is pressed

Aside from the ability to detect UA, a decent detection system should avoid false alarms under any circumstances. When drivers increase their input frequencies, the correlations between acceleration and these inputs decline. It raises the possibility of false UA alarms. The fourth test intends to assess the system performance when the driver changes inputs frequently. The result is shown in Figure 2.20. No false alarm was given even when the driver changed the gas pedal position, brake pedal position or steering wheel angle at a high frequency.

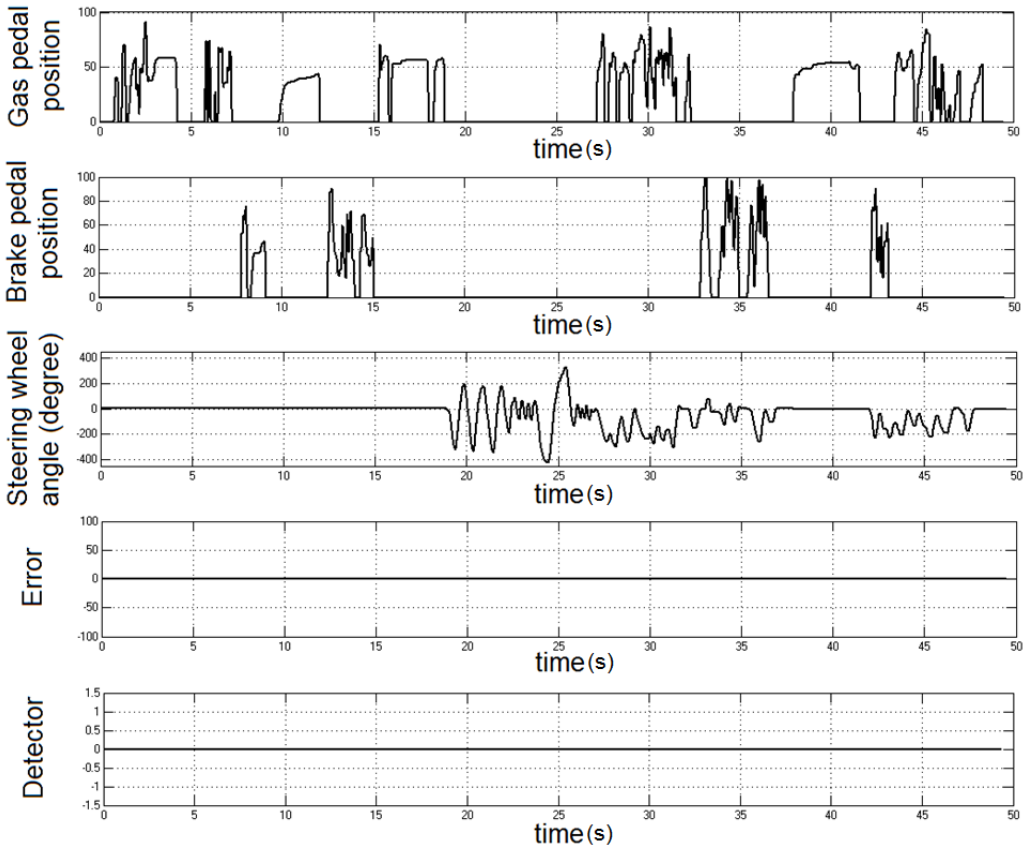

Figure 2.20 Normal condition

Based on the previous test results, the precision of the detection system is satisfactorily high. When the system gives an alert, the vehicle may encounter UA or an unwanted big increase of acceleration. To further avoid danger, an additional safety device was designed and integrated into the model to cut the engine power temporarily when the system detects an error and then reconnect the engine power. Since the detection system reacts very quickly, this process can slow down the vehicle before the driver takes some actions. It is helpful, for instance, when the driver is shocked by UA and does not know what to do. It should be noted that cutting the engine power and releasing the accelerator pedal have similar effects. For that reason, when the engine

power is disconnected, the detection system assumes the gas pedal position is zero irrespective of the real value. If the system detects UA multiple times in a short period of time, it sends a written message to the driver to pull off the road. (This is the current implementation in the simulation environment. In practice a more appropriate notification such as verbal, verbal and written, or a combination of an alarming sound and a verbal or written message can be used to inform the driver.) At the same time, it helps stopping the car by cutting the engine power and increasing the braking force when the brake pedal is pressed. A braking mechanism (e.g., an extra pump) which is different from the brake pedal is implemented in case of an ineffective brake.

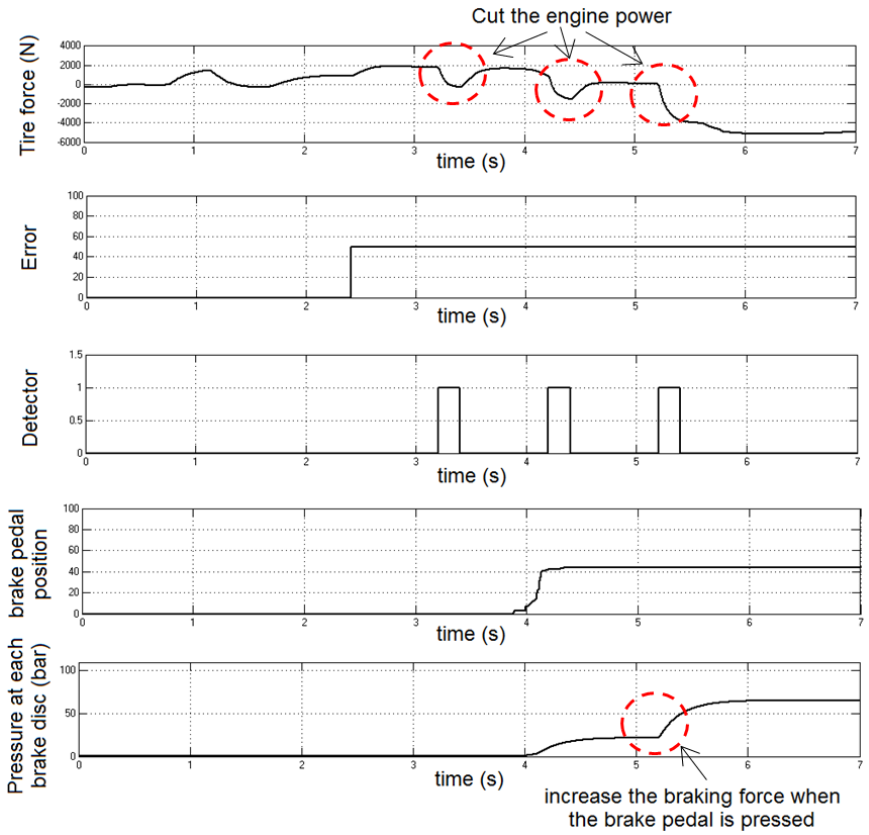

Figure 2.21 Brake assist

This safety system was simulated and tested in Simulink/dSPACE. One experiment result is illustrated in Figure 2.21. The braking system of the virtual car was modified so that it was not effective enough. When UA occurred suddenly, the detection system detected the error quickly and the safety device cut the engine power before the driver pressed the brake pedal. It can be seen in the graph that plots tire force vs. time that the tire force decreased due to this action. Since the UA event lasted for a relatively long period of time, the detection system gave multiple warnings. After the third warning, the safety device started to check if the brake pedal was pressed. When the pedal was pressed, it was assumed that the driver wanted to decelerate and the system helped increase the braking force. Figure 2.21 shows that the system successfully slowed down the vehicle and helped the driver stop the car finally.

Figure 2.22 A vehicle without the detector and brake assist

Figure 2.23 A vehicle equipped with the detector and brake assist

A comparison between a vehicle without the brake assist system and a vehicle with the brake assist system is shown in Figure 2.22 - Figure 2.23. The blue car is the subject vehicle and it is approaching an intersection. There is a stop sign ahead so that the subject vehicle needs to gradually slow down and finally stop before the stop sign. Two tests were conducted. In order to highlight the influence of the proposed detection and mitigation system, the subject vehicle was controlled using the same gas pedal inputs and brake pedal inputs in two tests. The only difference between these tests is that the second one contains brake assist while the first one does not.

It can be seen that even though the driver tried to press the brake pedal when UA occurred in the first test, the braking force was not big enough and the subject vehicle crashed into a vehicle, which had the right of way. In Contrast, the neural-network based detector gave multiple warnings quickly when UA occurred in the second test and sent

commands to the brake assist system. Finally, the system helped the driver stopped the car slightly before the stop sign and prevent an accident.

2.6 Summary

In this chapter, a data-driven detection system for UA is demonstrated. The system operates by smoothing vehicle signals and analyzing specific vehicle events like acceleration peaks and gear shifting. Corresponding models were created and combined to decide whether UA occurred. If UA was detected, the system cut the engine power and adjusted the braking force to prevent accidents. It should be noticed that though a robust detection system needs a large database of vehicle data to construct its models, these generated models only contain a few parameters. A low-cost processor is adequate for running these models to perform real-time detection.

Experiments were conducted under Simulink/dSPACE environment. UA events were simulated by adding a random error to gas pedal positions. Consequently, a human driver could find his virtual car accelerated by itself suddenly sometimes. Sensor noise was also introduced to vehicles during the simulation. Several tests were designed and run to evaluate the system under diverse conditions. Results are promising. The system reacted to errors very fast and both detection rates and precision are beyond 95%. It validates the possibility that well-built data-driven models can detect vehicle problems such as UA with high accuracy. Though these results are encouraging, caution should be taken while transferring the system to the real vehicles. It is possible, for example, that some uncommon working patterns of vehicles or driving styles were not covered by any of the collected subjects. Likewise, the driving environment in the experiments did not

span the entire range of real-world driving. However, it is believed that the simulator and experiments are sufficiently representative and realistic and comparable detection results can be achieved in real vehicles.

3. RATIONAL AND ADAPTIVE DRIVER MODEL-BASED DETECTION SYSTEM

3.1 Overview

In this chapter, a model-based approach to detecting unintended acceleration (UA) is proposed. The scheme of model-based detection system is shown in Figure 3.1. The basic idea of this approach is to develop an adaptive and rational driver model for a human driver. This driver model is combined with a vehicle model to predict the future behavior of the vehicle. Real driving behavior is recorded and compared with predicted driving behavior. If the real behavior deviates largely from the predicted behavior, the system concludes that an abnormal event has occurred.

In this study, it is assumed that sensors such as radar sensors, laser sensors and cameras are installed on vehicles so that the positions of cars nearby are available. In other words, the subject vehicle moves in a connected environment. At the same time, vehicle data such as velocities and throttle positions are also collected. Both the traffic data and vehicle data are sent to the driver model so that the model can adjust its parameters and adapt to the driver's behavior. The most difficult part of the model-based detection is to mimic the behavior of human drivers, especially the interaction between drivers. In this study, game theory was adopted to capture the patterns of human-vehicle interaction and driver-to-driver communication. The adaptive driver model was developed by combining multilevel PID controllers with the game theoretical model. Finally, a simplified vehicle model is combined with the driver model to guarantee real-

time prediction. The objective of the adaptive model is to keep the differences between the driver model and human drivers relatively small under normal conditions rather than to minimize the differences.

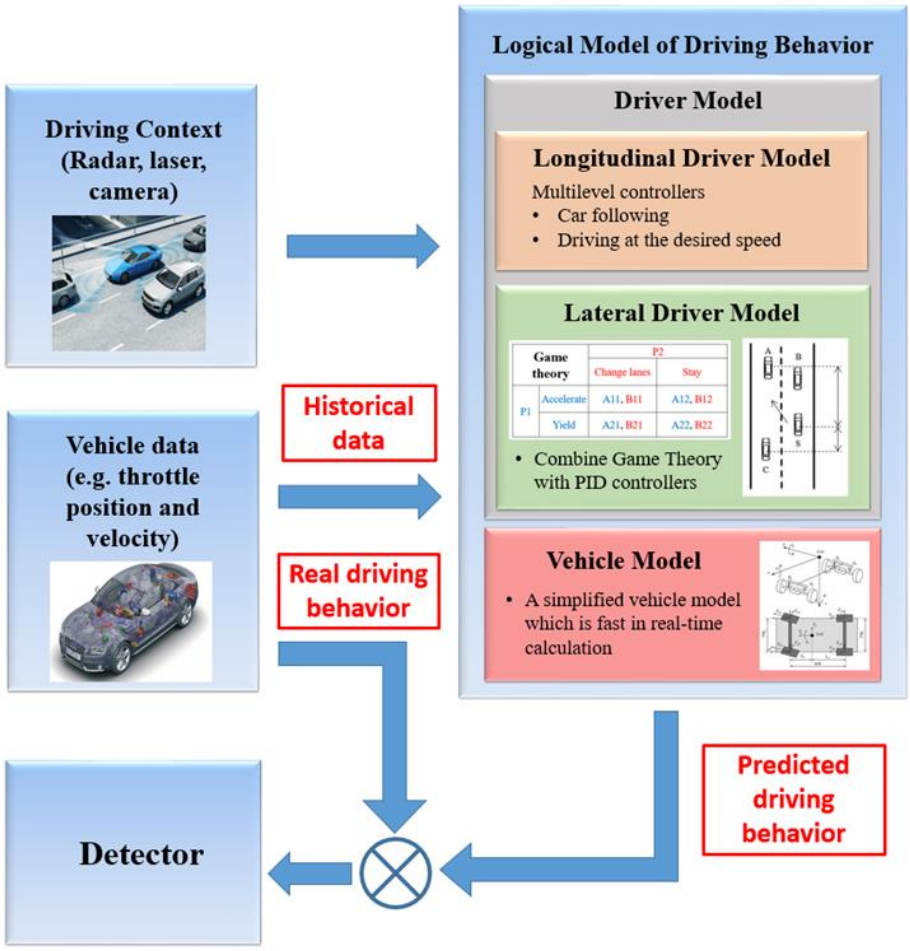

Figure 3.1 Driver model-based detection

3.2 Vehicle dynamic model

The first step for simulating the entire driving system is to develop a vehicle model. Since this model is used in real-time calculation, a simplified model is more appropriate than a complete, but time-consuming model. A Society of Automobile Engineers (SAE)

standard coordinate system is adopted [65]. The positive x axis points forward and the positive y axis points to the right. Positive rotations are determined by the right-hand rule for these axes, as shown in Figure 3.2.

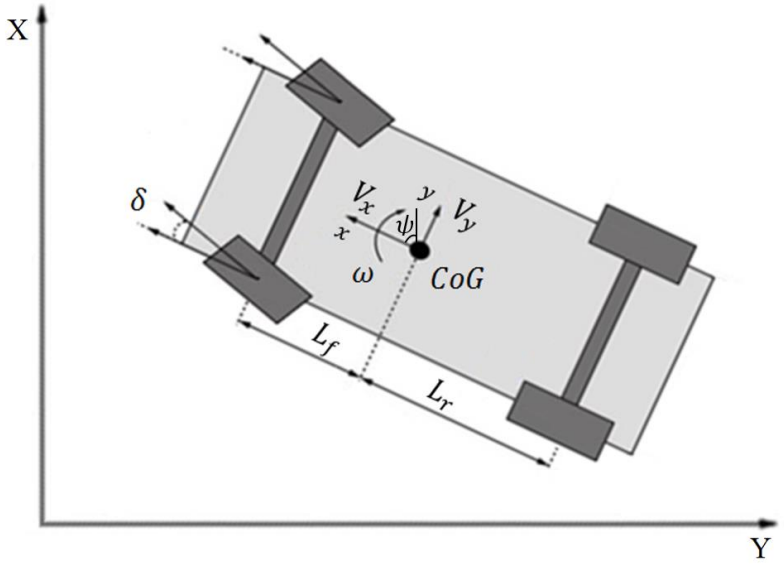

Figure 3.2 Vehicle model

The longitudinal, lateral and yaw movement of vehicles is given by [85]:

$$\dot{x} = v_x \tag{3.1}$$

$$\dot{v_x} = \frac{F_x - F_{drag}}{m} + \omega v_y \tag{3.2}$$

$$\dot{y} = v_y \tag{3.3}$$

$$\dot{v_y} = \frac{1}{m v_x}\left(-L_f C_{\alpha f} + L_r C_{\alpha r}\right)\omega - \frac{1}{m}\left(C_{\alpha f} + C_{\alpha r}\right)\beta + \frac{1}{m} C_{\alpha f}\delta - \omega v_x \tag{3.4}$$

$$\dot{\psi} = \omega \tag{3.5}$$

$$\dot{\omega} = \frac{1}{I_z v_x}\left(-L_f^2 C_{\alpha f} - L_r^2 C_{\alpha r}\right)\omega - \frac{1}{I_z}\left(L_f C_{\alpha f} - L_r C_{\alpha r}\right)\beta + \frac{1}{I_z} L_f C_{\alpha f}\delta \tag{3.6}$$

where v_x, v_y and ω represent the longitudinal velocity, lateral velocity and yaw velocity of the vehicle, x, y and ψ are longitudinal position, lateral position and yaw angle of the vehicle, m denotes the vehicle mass, $C_{\alpha f}$ and $C_{\alpha r}$ are front and rear cornering stiffness, I_z denotes the yaw moment of inertia, δ is the steering angle, L_f and L_r are the distance from CoG to the front and rear axles, F_x denotes the longitudinal tire force, F_{drag} is the aerodynamic drag and it is proportional to the relative speed of the vehicle, β denotes the vehicle sideslip angle, which is given by:

$$\beta = \frac{v_y}{v_x} \tag{3.7}$$

This state space model contains six states, which are x, y, ψ, v_x, v_y and ω. There are two inputs, which are F_x and δ. Equations (3.1) - (3.7) are created in the vehicle-fixed coordinate system. In order to simulate vehicles in the earth-fixed coordinate system, a transformation matrix is applied, which is calculated by:

$$\begin{bmatrix} V_X \\ V_Y \\ \Omega \end{bmatrix} = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{3.8}$$

where V_X, V_Y and Ω represent the longitudinal velocity, lateral velocity and yaw velocity of the vehicle in the earth-fixe coordinate system. Positions in the earth-fix coordinate system can be calculated by integrating these velocities.

3.3 Drive model

A common driving loop is shown in Figure 3.3. A driver sends control inputs to

electronic control units. Subsequently, these control units send commands to other vehicle components such as the throttle and power is transmitted from the engine to wheels. Finally, generated tire forces make the vehicle move. Vehicle motion could be estimated by a vehicle model. Other than that, drives' inputs, electronics control units and other vehicle components could be represented by a driver model. The driver model collects information from the environment and controls tire forces and steering angles to let the vehicle move safely. A complete driver model contains longitudinal and lateral control.

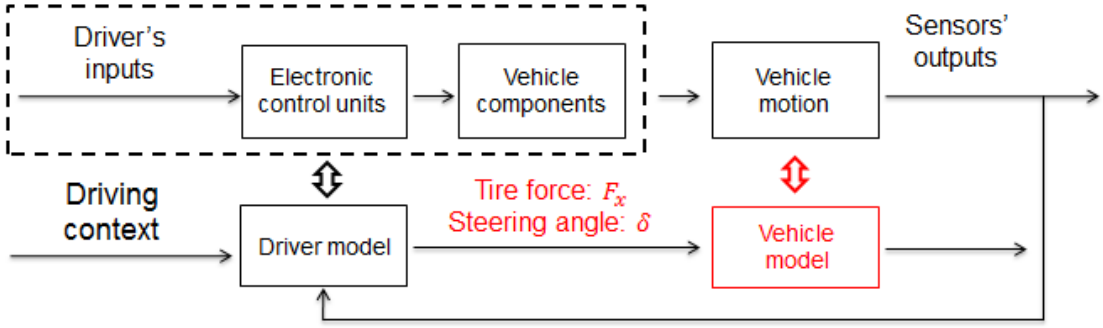

Figure 3.3 Driver model

3.3.1 Longitudinal driver model

The first part involves the longitudinal driving logic. The longitudinal movement of a vehicle can be divided into two types. The first type is driving with no car ahead. In this situation, the driver controls the car according to his desired velocity. The second type is car following. When there is a car ahead and it is not convenient to change lanes, the driver has to estimate a safe following distance and drives his car based on it. Three-second rule [86, 87] is used to estimate a normal following distance, which means the

time headway of the vehicle should be at least be three seconds. Time headway is calculated by

$$TH = \frac{P_{ahead} - P_{subject}}{V_{subject}} \tag{3.9}$$

where TH is the time headway of the subject car, P_{ahead} is the position of the car ahead, $P_{subject}$ and $V_{subject}$ are the position and velocity of the subject car. The driver model needs to know how to choose the driving mode according to the driving environment. The mode transition process is illustrated in Figure 3.4.

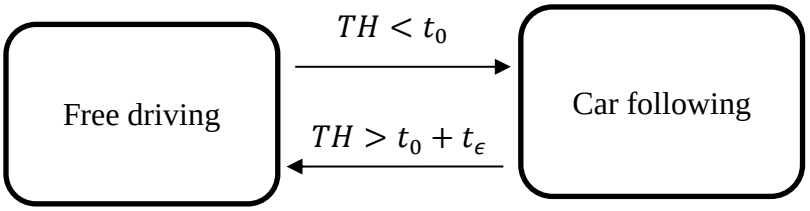

Figure 3.4 Driving mode transition

The variable t_0 denotes the desired following distance. When the time headway is smaller than t_0, the subject car enters the car-following mode. In order to prevent frequent mode changes, the subject vehicle only goes back to the free driving mode when the time headway is bigger than $t_0 + t_\epsilon$, where t_ϵ is a positive number. In such a way, the vehicle can stay in the car-following mode when its time headway fluctuates near t_0. Besides choosing modes, the longitudinal driver model should also be capable of controlling the vehicle in two modes. To accomplish these tasks, an adaptive cruise control model was designed and applied to the longitudinal movement, which is shown

in Figure 3.5.

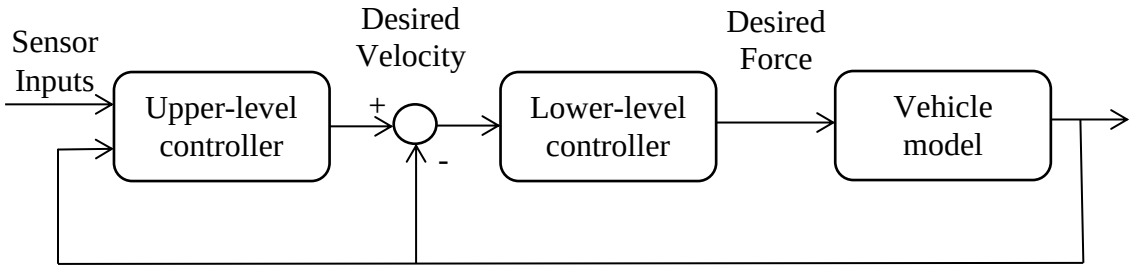

Figure 3.5 Longitudinal driver model

The model contains an upper-level controller and a lower-level controller. The lower-level controller is responsible for cruise control. It implements feedback PID control to adjust tire forces and let the vehicle move at the desired velocity. The upper-level controller is in charge of calculating the current desired velocity. During car following, this controller estimates a safe following distance and uses it to calculate the desired velocity. When there is no car ahead, this controller estimates the human driver's desired velocity in real time. Details about estimating the driver's desired speed are discussed in section 3.3.3.

3.3.2 Lateral driver model

In addition to a model that controls the longitudinal movement, a complete logic driver model still needs to decide when to change lanes and control the steering wheel. This part of the model was designed by mimicking the behavior of a rational human driver. A flowchart of the logic process is shown in Figure 3.6.

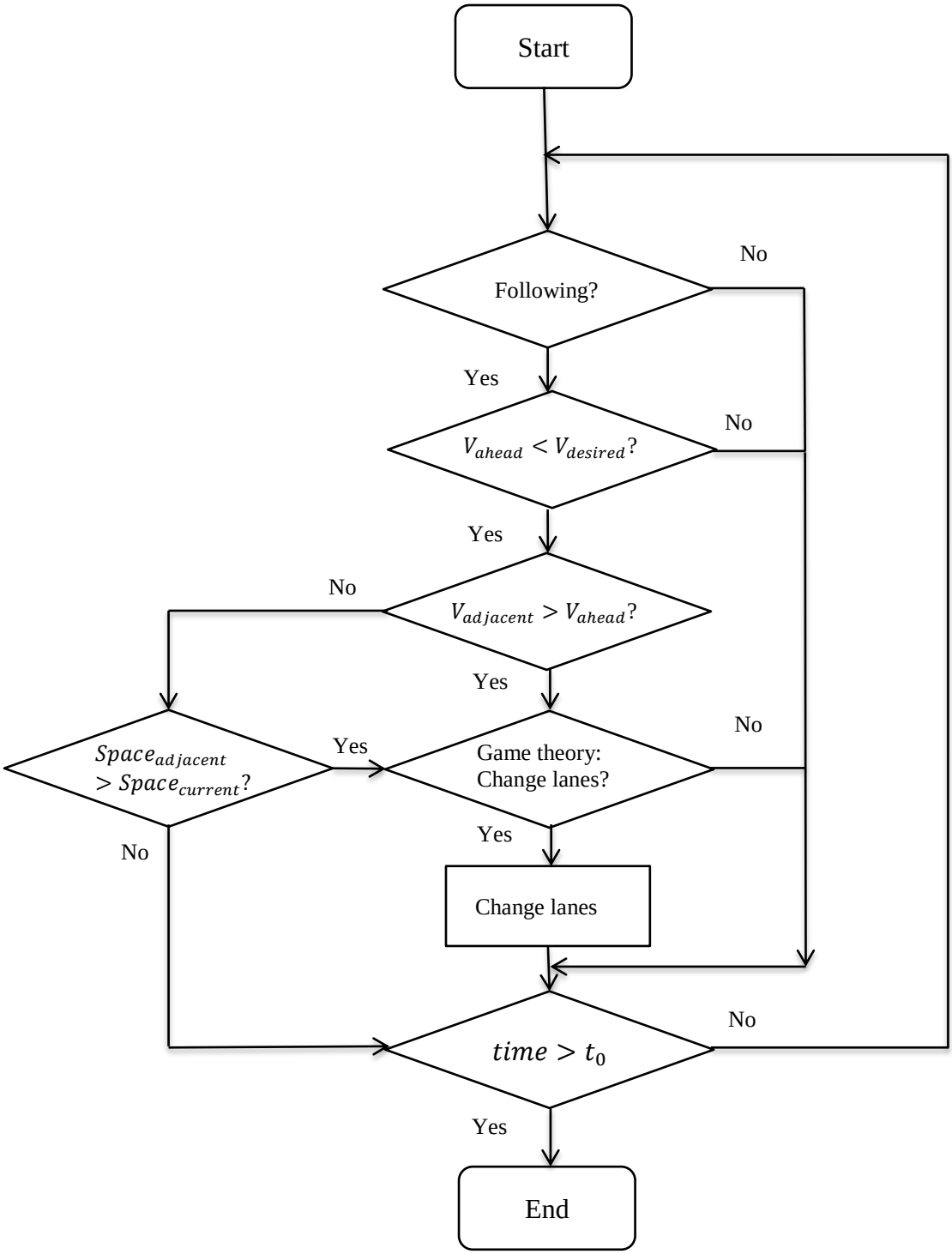

Figure 3.6 Lateral driver model

This whole process is imitated by combining PID controllers with a game theoretical model. PID controllers are used to control steering angles. The game theoretical model is used to mimic the interaction between drivers. Generally speaking, a human driver begins to considering changing lanes when the car ahead is moving too slowly. Before changing lanes, the driver looks at the rear-view mirror and side-view mirror to check the surrounding traffic. If cars in the adjacent lane are faster than the car ahead or the space in the adjacent lane is big enough for overtaking the car ahead, the driver desires to change lanes. If there is no following car in the target lane, changing lanes is relatively easy. The driver only needs to turn the steering wheel and move to the target lane smoothly. However, changing lanes in heavy traffic is relatively complicated since the driver has to predict the trajectories of vehicles nearby.

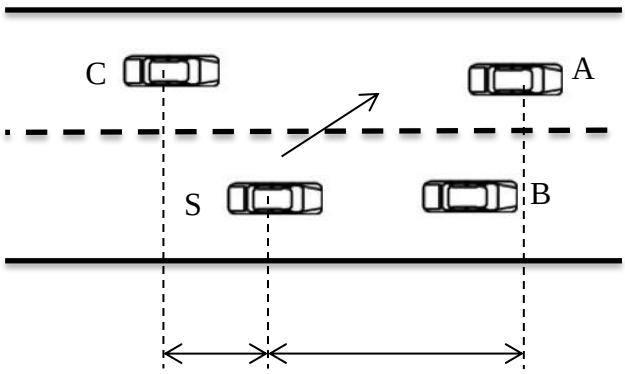

Figure 3.7 Lane-changing scenario

In recent years, increasing efforts have been made to study lane-changing behavior. A typical lane-changing scenario is shown in Figure 3.7. Car S is the subject vehicle. It wants to move to the left lane. However, Car C is following in the adjacent lane and Car

S needs to predict the trajectories of surrounding vehicles if it wants change lanes. Many popular algorithms such as rapidly-exploring random tree (RRT) [88-90], model predictive control (MPC) [91-93] and the gap acceptance model [94-96] assume that Car C has constant velocities or acceleration during the lane changing process of Car S. Liu et al. [97] adopted recursive regression to predict the trajectory of Car C.

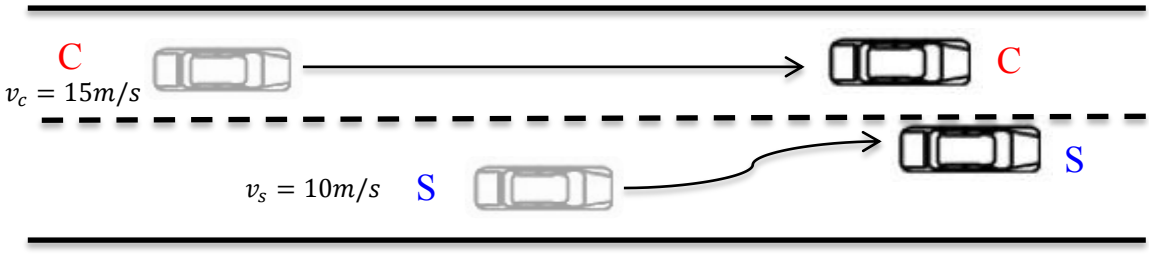

Figure 3.8 Aggressive competing vehicle

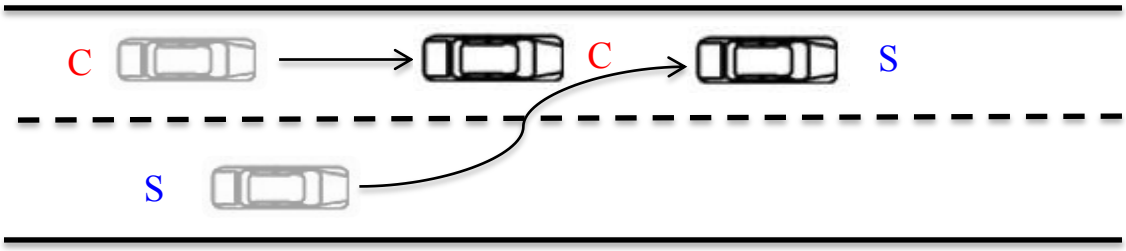

Figure 3.9 Timid competing vehicle

However, all these algorithms have one common problem. There is delay between the real signals and estimated signals, which could cause some trouble. An example is shown in Figure 3.8. Car S is the host vehicle and wants to move to the left lane. Car C, a following vehicle in the adjacent lane, is far behind. Car S thinks that if Car C has a constant velocity, it is safe to change lanes since the distance between two cars is big

enough. However, Car C is much faster. If Car C is aggressive, it may not want to slow down and follow Car S in this lane. Actually, when Car C finds that Car S starts to change lanes, Car C may accelerate and prevent Car S from changing lanes. It is relatively dangerous. Figure 3.9 shows another scenario. Car C is close to Car S at the beginning. Car S thinks that it could not change lanes if Car C has a constant velocity. Otherwise, it would crash into Car C. However, Car C may be timid in the real world. When it finds that Car S starts to change lanes, it may slow down and let Car C change lanes successfully and safely. In summary, the interaction between vehicles is not considered in these scenarios. The host vehicle needs to interact with emotional human drivers. The assumption that the competing vehicle has a constant velocity during the lane changing process of the host vehicle is not always true in the real world.

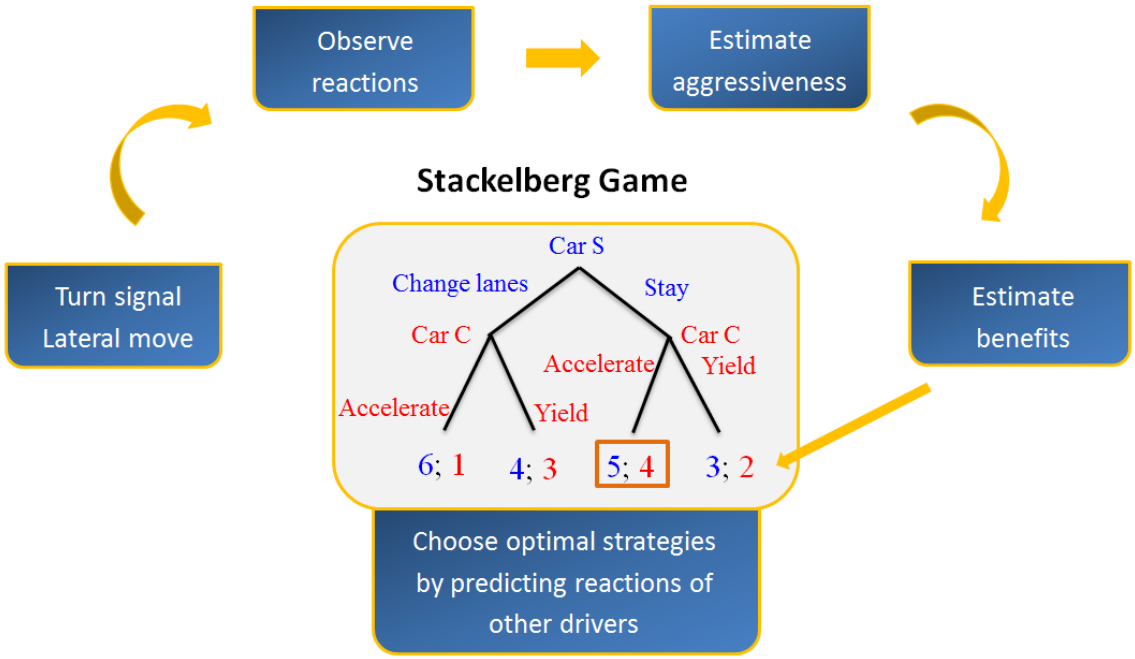

Figure 3.10 Mimic the lane changing process of a human driver

Game theory is a powerful tool to study the interaction between decision makers. The whole process mimics the lane changing process of a human driver, as shown in Figure 3.10. At the beginning, the host vehicle starts to interact with the competing vehicle by using the turn signal or making a small lateral move. Then the driver of the host vehicle looks at side view mirrors and the rear view mirror and observes the reactions of the competing vehicle. The driver is able to see if the competing is accelerating or decelerating and how fast the acceleration changes. Based on the observation, the subject vehicle is able to estimate the aggressiveness of the competing vehicle to some extent. Subsequently, the host vehicle starts to estimate the benefits of different strategies according to the estimated aggressiveness of the competing vehicle. Finally, Car S chooses its optimal strategy by predicting the reactions of Car C. This process is like playing chess. Both players are trying to maximize their payoffs. For instance, if Car S changes lanes and Car C reacts by accelerating, Car S' payoff is six and Car C's payoff is one. If Car S changes lanes and Car C reacts by yielding, Car S' payoff is four and Car C's payoff is three. It is obvious that Car C will choose to yield to Car S when Car S changes lanes due to its higher payoff. Therefore, Car S' payoff is only four if it changes lanes. Similarly, Car S' payoff is five when it chooses to stay in the current lane because Car C tends to react by accelerating. The optimal strategy of Car S is staying in the current lane.

The real scenario of changing lanes is a little more complicated than the game in Figure 3.10. Besides the following car in the adjacent lane, there may be other cars, which have big influence on the decision of the host vehicle, as shown in Figure 3.7. Car

B is the lead vehicle in Car S' lane while Car A moves in front of Car S in the adjacent lane. Car S has to consider Car A's positions to prevent crashes while changing lanes. But Car A is not required to react to the future actions of Car S. In other words, the interaction between Car A and Car S is not guaranteed. Only Car S and Car C are interacting with each other when Car S changes lanes. Car A and Car B only provide constraints to the host vehicle. Therefore, there are only two players in this game.

The structure of the game is shown in Table 3.1, where *a* represents the longitudinal acceleration of the vehicle and *U* denotes the payoff of the combination of strategies. Car S decides not only if it will change lanes, but also if it will accelerate or decelerate while changing lanes or staying in the current lane. Since acceleration is continuous, there are infinite combinations of strategies. Car C also has an infinite number of choices. It can choose acceleration within the physical limits of the vehicle. Both two cars are trying to maximize their payoffs.

Table 3.1 Structure of the game

Decision making		Car S	
		Change lanes $-6 \leq a \leq 4$	*Stay* $-6 \leq a \leq 4$
Car C	$-6 \leq a \leq 4$	U_{c0}, U_{s0}	U_{c1}, U_{s1}

It can be seen that payoff functions play an important role in the decision making process. In order to reproduce drivers' logic in lane changes, two payoff functions are defined in the game. The first function U_{safety} measures the safety payoff that a player can obtain in the game. The second function U_{space} estimates the space payoff that a player can get in the game.

U_{safety} is defined as the change of the safety factor in the lane-changing process, as shown below.

$$U_{safety} = \frac{1}{2}\left(SP_{t=T_{cl}} - SP_{t=0}\right) \tag{3.10}$$

where $SP_{t=T_{cl}}$ is the safety factor after the lane changing process ends, $SP_{t=0}$ is the initial safety factor and T_{cl} is the time needed to change lanes, which is given by

$$T_{cl} = \frac{y_s T_s}{LW} \tag{3.11}$$

where y_s is the lane position of car S, LW is lane width and T_s is the duration of complete lane changes. The safety factor is a function of the time headway. Its range is between negative one and one. The smaller the time headway, the smaller the safety factor. The safety factor of a vehicle is defined by

$$SP = \begin{cases} 1 & T_{headway} \le -T_b \\ \dfrac{2 * \left|T_{headway}\right|}{T_b} - 1 & -T_b < T_{headway} \le T_b \\ 1 & T_{headway} > T_b \end{cases} \tag{3.12}$$

where $T_{headway}$ is the minimum time headway between the subject vehicle and its surrounding vehicles and T_b is a breakpoint, which represents the desired time headway. When two cars are far enough from each other, the time headway is bigger than T_b. It is always safe under this condition and the safety factor reaches its maximum value. T_b of the competing vehicle is given by

$$T_b = \min(3, T_0) \tag{3.13}$$

where T_0 is the initial time headway between the competing vehicle (i.e. Car C) and the vehicle in front of it (i.e. Car A) when the host vehicle (i.e. Car S) starts to change lanes. The number three comes from the three-second rule [98], which suggests that a minimum of a three second interval between the host vehicle and the car ahead should be kept. Equation (3.13) means if the initial time headway of the competing vehicle is bigger than three seconds, it is assumed that the desired time headway of competing vehicle is three seconds. Otherwise, the desired time headway of the competing vehicle is T_0.

The initial time headway between Car S and Car C is defined by

$$T_{headway} = \frac{P_s - P_c}{v_c} \tag{3.14}$$

where P_s and P_c are initial longitudinal positions of Car S and Car C relative to the road coordinate system and v_c is the velocity of Car C. The time headway of the following

vehicle after T_{cl} seconds is given by

$$T_{headway} = \begin{cases} \dfrac{P_{sl} - P_{cl}}{v_c + a_c T_{cl}} & P_{sl} \geq P_{cl} \\ \dfrac{P_{cl} - P_{sl}}{v_s + a_s T_{cl}} & P_{sl} < P_{cl} \end{cases} \tag{3.15}$$

where v_s and v_c are velocities of Car S and Car C relative to the road coordinate system, a_s and a_c are acceleration of Car S and Car C, P_{sl} and P_{cl} are longitudinal positions of Car S and Car C, which are calculated by

$$P_{sl} = P_s + v_s T_{cl} + \frac{1}{2} a_s T_{cl}^2 \tag{3.16}$$

$$P_{cl} = P_c + v_c T_{cl} + \frac{1}{2} a_c T_{cl}^2 \tag{3.17}$$

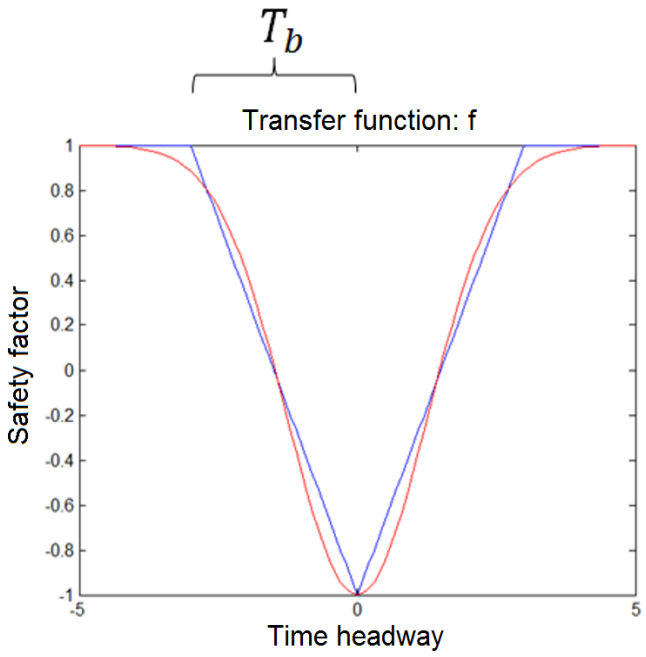

Figure 3.11 Safety factor

In summary, the safety factor is a piecewise linear function of the time headway. However, a piecewise function is relatively slow in the optimization process. In order to make the model more efficient, a function that is differentiable everywhere is used to approximate the piecewise function. In this study, Gaussian probability distribution functions are adopted. The amplitude is fixed and the standard deviation is adjusted to approximate the original function, as shown in Figure 3.11.

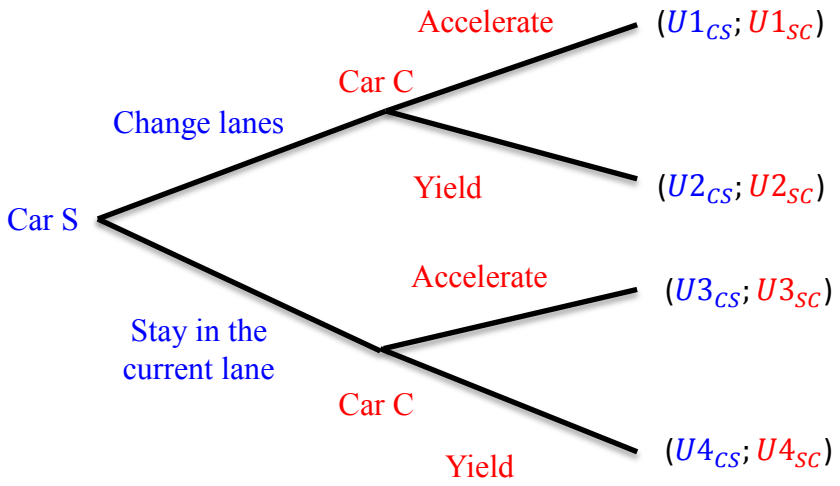

Figure 3.12 Space payoff tree

The second payoff function U_{space} estimates the change of space payoffs. A space payoff tree is shown in Figure 3.12. U_{space} is given by

$$U_{space} = \frac{1}{2}\left(RP_{t=T_{cl}} - RP_{t=0}\right) \tag{3.18}$$

where T_{cl} is the time needed to change lanes, $RP_{t=T_{cl}}$ is the space payoff after T_{cl} seconds and $RP_{t=0}$ is the initial space payoff. The space payoff is a function of relative positions. Its range is between negative one and one. Car S can get more space by changing lanes and Car C can get more space by accelerating and moving in front of Car S. The estimation of the space payoff of vehicles depends on if two vehicles move in the same lane.

When two cars move in different lanes, the space payoff of Car C in the instant *t* is calculated by

$$RP_{cs}(t) = \begin{cases} -1 & t_{cs}(t) \leq -3 \\ \frac{2}{3} t_{cs}(t) + 1 & -3 < t_{cs}(t) \leq 0 \\ 1 & t_{cs}(t) > 0 \end{cases} \tag{3.19}$$

where t_{cs} is the relative position of Car C with respect to Car S. For example, when car S is in front of Car C, Car C's relative position is defined as

$$t_{cs} = \frac{d_{cs}}{v_c} \tag{3.20}$$

where d_{cs} is the relative distance between Car C and Car S and v_c is the velocity of Car C. Car S' relative position with respect to Car C is given by

$$t_{sc} = \frac{d_{sc}}{v_c} = -\frac{d_{cs}}{v_c} = -t_{cs} \tag{3.21}$$

The sum of the space payoffs of Car S and Car C is zero, as shown below.

$$RP_{sc}(t) = -RP_{cs}(t) \tag{3.22}$$

Figure 3.13 illustrates why the space payoff is defined in this way. When car S is far ahead of car C, it is easy for car S to change lanes. As a result, car C does not lose a lot of space and tends to give way to car S. When car S wants to change lanes and car C is close to car S, car C has to decelerate a lot to give way to car S. In this case, car C tends to accelerate instead of giving way to car S.

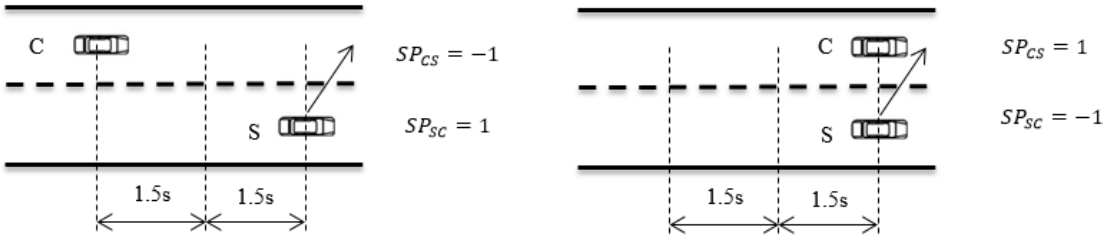

Figure 3.13 Space payoffs when two cars move in different lanes

When two cars move in the same lane, the space payoff of Car C in the instant *t* is calculated by

$$RP_{cs}(t) = \begin{cases} -1 & t_{cs}(t) \le -3 \\ \dfrac{t_{cs}(t)}{3} & -3 < t_{cs}(t) \le 3 \\ 1 & t_{cs}(t) > 3 \end{cases} \tag{3.23}$$

Equation (3.23) means that when a car is far behind the other car, it is relative position is negative one. When it is far ahead of the other car, it is relative position is one. The sum of two cars' space payoffs is still zero. For the sake of solving the game fast and efficiently, Equation (3.19) and Equation (3.23) are approximated by functions

that are differentiable everywhere as well. In this study, Gaussian cumulative distribution functions are utilized.

The total payoff function is a linear combination of the safety payoff and space payoff, which is given by

$$U_{payoff} = f_w((a - a_0)^2) * \Big((1 - \beta(q)) * U_{safety}\ (a) + \beta(q) * U_{space}(a) + 1\Big) - 1 \qquad (3.24)$$

where a is the future acceleration (i.e. future strategy) of the vehicle and a_0 is the current acceleration of the vehicle, f_w is the jerk penalty, q is the aggressiveness of the driver and β is the weight of payoffs. It can be seen that β is the most important parameter in the total payoff function. It affects the weights of space payoffs and safety payoffs, as shown below.

$$\beta \sim \frac{\beta(q) * U_{space}}{(1-\beta(q)) * U_{safety}}, \qquad 0 \le \beta \le 1 \qquad (3.25)$$

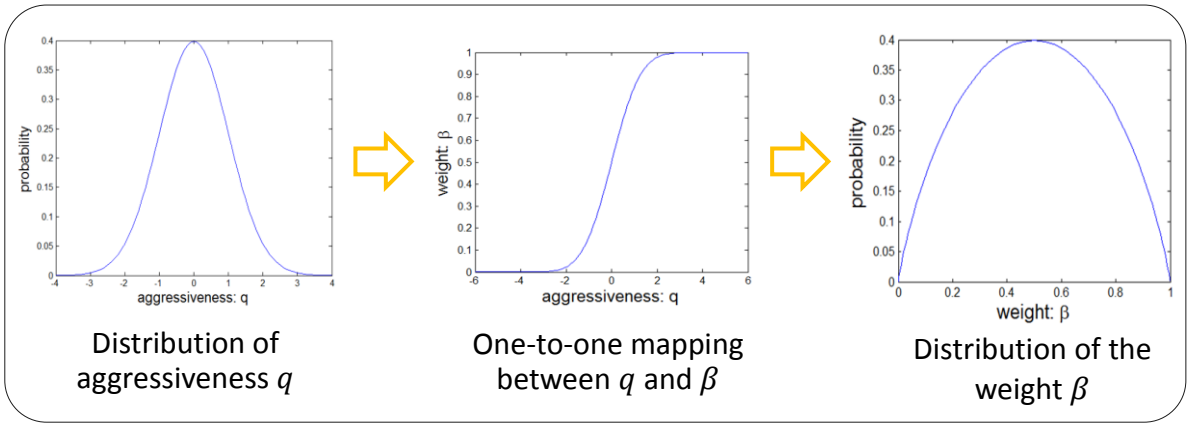

Figure 3.14 Distribution of aggressiveness

One parameter called aggressiveness is introduced into the equation. Aggressiveness of drivers has a significant influence on driving. A lot of research has been done to study human factors including aggressiveness in driving [99-102]. One of the widely-used assumption is that aggressiveness of drivers obeys Gaussian distribution [103, 104], as shown in Figure 3.14. There is a one-to-one mapping between aggressiveness q and weight β. Aggressive drivers care more about space than safety. They have big β and a small weight of safety payoffs. Timid drivers care more about safety than space. Their weights of safety payoffs are bigger than those of space payoffs. The jerk penalty function is defined as

$$f_w = e^{-T_{cl}^2 * \frac{(a-a_0)^2}{w}} \tag{3.26}$$

where a is the future acceleration of the vehicle and a_0 is the current acceleration of the vehicle, w is the penalty parameter and T_{cl} is the time needed to change lanes.

After estimating the safety payoff and space payoff, the last step is to solve the game. In the lane changing process, the host vehicle that wants to change lanes and the competing vehicle in the adjacent lane follow a 2-person Stackelberg game. This is a bilevel optimization problem, as shown below.

$$(a_s^*, c_l^*) = \underset{a_s, c_l}{argmax}\left(\min_{a_c \in \gamma^c(a_s, c_l)} U_s(a_s, c_l, a_c, q_s)\right) \tag{3.27}$$

$$\gamma^c(a_s, c_l) \triangleq \{\xi \in \Gamma^c : U_c(a_s, c_l, \xi,\ q_{sc}) \geq U_c(a_s, c_l, a_c, q_{sc}), \forall a_c \in \Gamma^c\} \tag{3.28}$$

subject to

$$v_c \geq 0,$$

$$v_s \geq 0,$$

$$a_{min} \leq a_s \leq a_{max},$$

$$a_{min} \leq a_c \leq a_{max},$$

$$\begin{cases} P_s < P_A & after\ Car\ S\ changes\ lanes \\ P_s < P_B & when\ Car\ S\ stays\ in\ the\ current\ lane \end{cases}$$

$$U_{safety}(s) > K$$

where Car S is the host vehicle that wants to change lanes, Car C is the competing vehicle following in the adjacent lane, Car A is the car ahead in the adjacent lane and Car B is the car ahead in host vehicle's lane, U_s denotes the payoff of the host vehicle, U_c is the payoff of the competing vehicle, a_s denotes the possible acceleration of Car S, c_l shows if Car S is changing lanes, a_c is the possible acceleration of Car C, a_s^* is the optimal acceleration of Car S, a_c^* is the optimal acceleration of Car C, c_l^* shows if changing lanes is appropriate for Car S, $\gamma^c(a_s, c_l)$ denotes the optimal action candidates of Car C given the actions of Car S, Γ^c indicates the action candidates of Car C, $U_{safety}(s)$ is the safety payoff of Car S, a_{min} and a_{max} are the minimum and maximum acceleration that a vehicle can reach, q_s is the aggressiveness of Car S, q_{sc} is the estimated Car C's aggressiveness, P_A, P_B and P_s are positions of Car A, Car B and Car S and v_c and v_s are velocities of Car C and Car S.

The solution of this game is the strategy that maximizes lower limit of the payoff

from the viewpoint of the leader with the consideration of the follower's reacting strategy. In other words, the solution indicates the pair that maximizes their payoffs in the worst case in order. If high accuracy of strategies is not required, this bilevel optimization problem can be solve by extensively searching the discrete payoff matrix. For example, one decimal place is accurate enough for the acceleration of vehicles in this game. A payoff matrix consists of a set of discrete acceleration can be built, which is similar to Table 3.1. Each element of the matrix represents the payoff of the combination of one of Car S' strategies and one of Car C's strategies. This matrix only has finite elements and the Stackelberg equilibrium can be found easily. If the accuracy of strategies is particularly important, this bilevel optimization problem can also be solved by Bilevel Evolutionary Algorithm (BLEAQ) [105]. However, BLEAQ is slower than extensive search. In should be noticed that the solution of the game is calculated in every instant. It may change when the game is played. For example, the solution may change from changing lanes to staying in the current lane after getting the latest traffic information, which is similar to real drivers.

The game in Equations (3.27) - (3.28) simulates the reasoning process of the leader while changing lanes. If the host vehicle knows the aggressiveness of the competing vehicle, it can predict the future reactions of the competing vehicle. Based on the predicted reactions, the host vehicle can find its optimal strategies. However, there is still one unknown, the aggressiveness of the competing vehicle. In order to find the optimal strategy accurately, the host vehicle needs to estimate the aggressiveness of the competing vehicle, as shown in Figure 3.15.

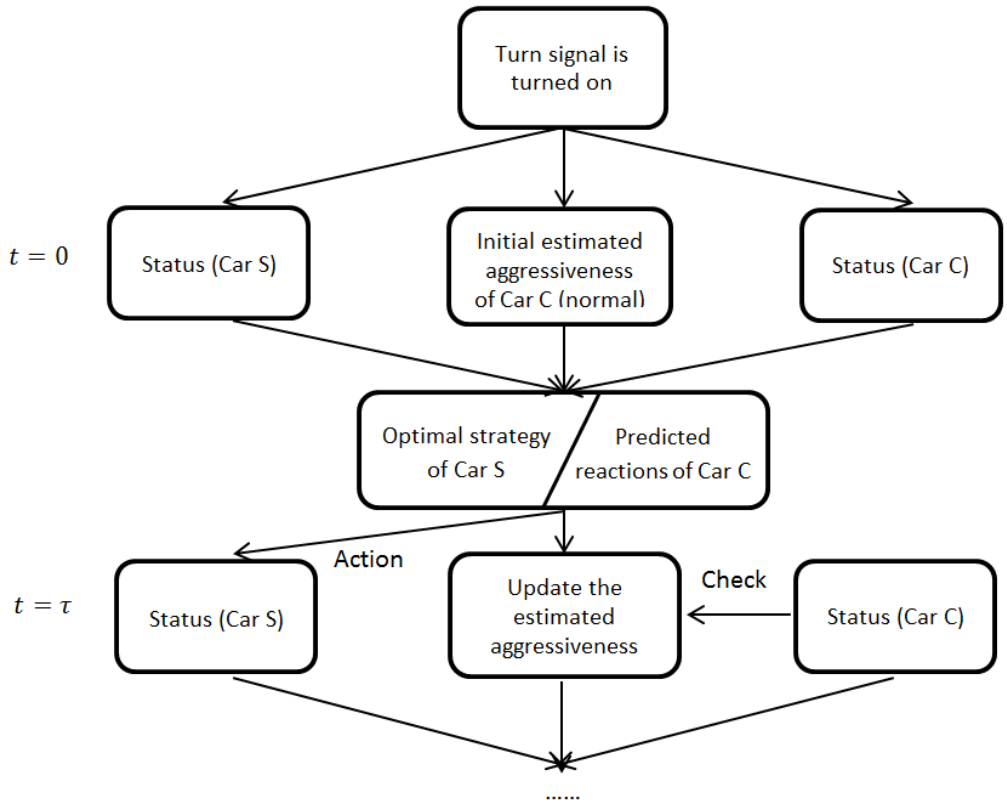

Figure 3.15 Estimate the aggressiveness of the competing vehicle

The host vehicle starts to interact with the competing vehicle by using the turn signal or making a small lateral move. At the beginning, the host vehicle knows nothing about the competing vehicle. It is reasonable to assume the competing vehicle is driven by a normal driver, whose aggressiveness is zero according to the distribution in Figure 3.14. The host vehicle predicts how a normal driver will react to its future actions and finds its own optimal strategy. In the next instant, the host vehicle is able to observe the real action of the competing vehicle. The real action is compared with the predicted action and the difference is used to update the estimated aggressiveness. The estimated aggressiveness is only updated when the difference is bigger than $0.3m/s^2$. It is updated

by solving the following equation.

$$U_{payoff}(q_c, Real_{a_c}) \geq U_{payoff}(q_c, Predicted_{a_c}) \tag{3.29}$$

where U_{payoff} is the total payoff of Car C, q_c is the aggressiveness of Car C, $Real_{a_c}$ is the real acceleration of Car C and $Predicted_{a_c}$ is the predicted acceleration of Car C. There is only one unknown q_c. The range of the aggressiveness of Car C can be estimated by solving the inequality.

3.3.3 Estimate the driver's desired speed

One of the most important parameters of the driver model is the driver's desired speed. The driver model compares the velocity of the car ahead with the driver's desired speed so that it can decide if it needs to change lanes. The desired speed is also used as the reference signal in the feedback control loop in the free driving mode. However, the desired speed of a human driver cannot be measured by a sensor directly. Furthermore, the desired speed varies a lot during driving. To address these challenges, a fixed-size queue is used to memorize the velocities of the car and estimate the driver's desired speed. The objective is to obtain a signal, which tracks the real desired speed. An example of the queue is shown as below.

$$V_d = [V(t-3), V(t-6), V(t-9), \ldots, V(t-30)] \tag{3.30}$$

where V_d represents the velocity queue, V is the velocity of the car and t denotes the time. The queue records the velocities of the car in the past few seconds. If the previous

velocities are not available, the queue is filled in with the speed limit of the current road. The size of this queue is fixed, which means when a new number is added to the rear terminal position, the oldest number at the front terminal position is removed. The estimated desired velocity of the driver is calculated by:

$$V_{estimated} = \frac{\sum_i (V_{di})}{N} \tag{3.31}$$

where V_{di} is the element of the velocity queue and N is the size of the queue. The estimated desired velocity is the mean value of the elements of the queue. When the car moves, the queue keeps being updated and the estimated velocity is recalculated in real time. A flowchart of the update process is shown in Figure 3.16.

At the beginning, the system checks if the driver model works in the car-following mode. If the model does not work in the following mode, there is no car ahead nearby. In this case, the system enqueues the newest velocity and dequeues the oldest velocity. The estimated desired speed is recalculated using the new queue in real time. This process is similar to a moving average filter and the estimated speed will track the vehicle's speed. If the model shows that it is following another car, the system starts to check whether the driver model and the human driver stay in the current lane or change lanes together. If the model and human driver keep staying in the same lane, the estimated speed is the same as or close to the real desired speed. Consequently, the elements of the queue do not need to be changed.

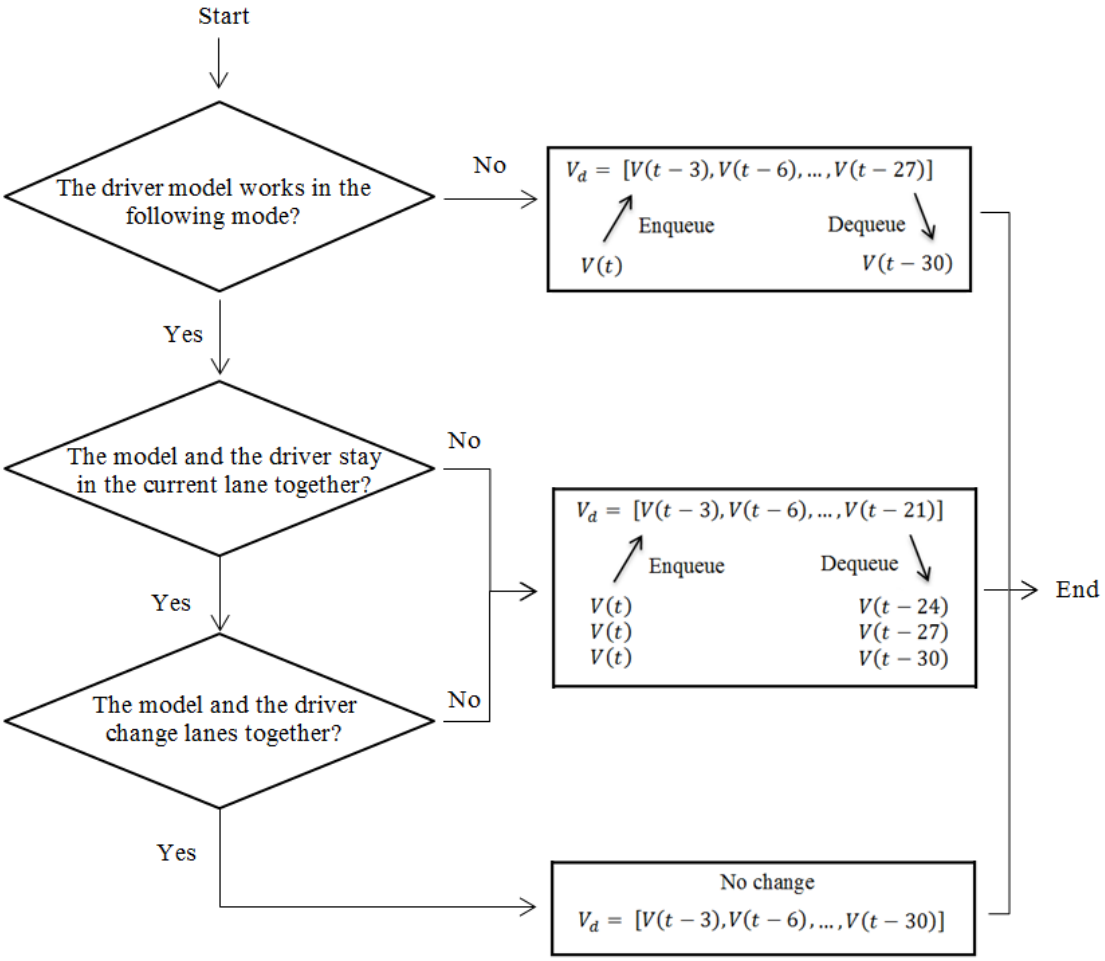

Figure 3.16 Estimate the driver's desired speed

In contrast, the model and the human driver may act differently. If the model changes lanes while the human driver stays in the original lane, the estimated velocity is bigger than the desired velocity. If the human driver changes lanes and the model does not change lanes, the estimated velocity is smaller than the desired velocity. In either case, the difference between the estimated velocity and the real desired speed is too big. To make the mean value of the queue elements converge to the real desired speed faster, the latest velocity is enqueued multiple times instead of only once and corresponding

oldest velocities are removed. In summary, the system selectively records velocities and estimates the desired speed using the historical data.

3.4 Detection system

In this section, the longitudinal driver model, the lateral driver model and the vehicle model are integrated into one fault detection system, which is able to predict the rational future positions of vehicles. The prediction results are compared with real positions measured by sensors. If the vehicle movement deviates largely from prediction results, the system concludes that an abnormal situation or an anomaly has occurred.

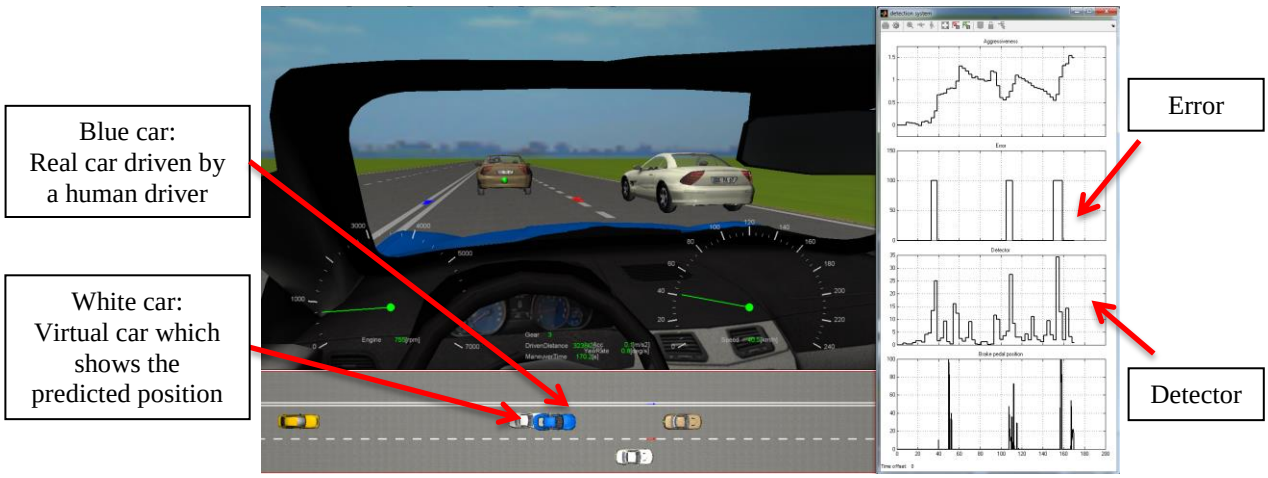

Figure 3.17 Prediction results of the driver model

The complete fault detection system was embedded into Simulink/dSPACE for tests. At the beginning, several virtual cars were assigned random initial positions, velocities, desired velocities and aggressiveness. Apart from these virtual cars, a human driver controls one car using analog steering wheels and pedals. This car is equipped with sensors such as LIDAR sensors and cameras so that it is able to detect surrounding cars.

An adaptive driver model was combined with the simplified vehicle model to predict the future positions of this vehicle, as shown in Figure 3.17.

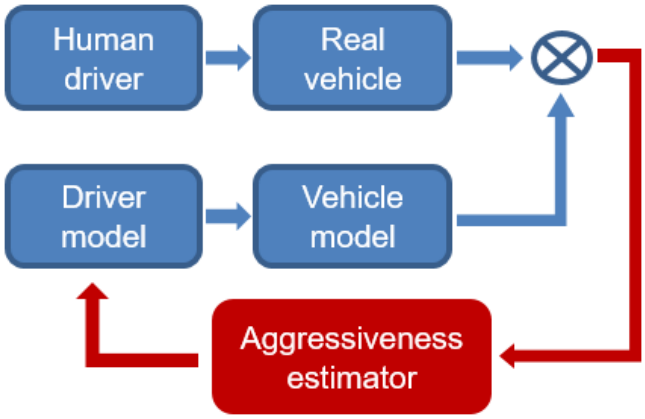

Figure 3.18 Estimate the aggrssivness of the subject vehicle

Since the aggressiveness of a human driver may not be constant, this parameter should be estimated and updated in real time, as shown in Figure 3.18. Initially, it is assumed that the aggressiveness of the driver is zero (i.e. normal). After that, two variables are calculated to indicate how much the car driven by the human driver deviates from the car controlled by the rational driver model. Since a transient difference is not big enough to represent the deviation, these two variables are calculated every three seconds during the simulation. The variables are defined as

$$ex(n) = \frac{\sum_{t\in[3(n-1),3n]}\big(model(t) - car(t)\big)}{N_t} \qquad n\epsilon Z \tag{3.32}$$

$$ey(n) = \sum_{t\in[3(n-1),3n]} (model_{lc}(t) - car_{lc}(t)) \qquad n\epsilon Z \tag{3.33}$$

where *ex* is the average difference between predicted longitudinal positions and real longitudinal positions in the current time interval, n is the index of time intervals, $model(t)$ is the predicted longitudinal position at instant *t*, $car(t)$ denotes the real longitudinal position at instant *t*, N_t is the number of sampled points in the current time interval, *ey* represents the difference between predicted number of lanes changes and real number of lane changes in the current time interval, $model_{lc}(t)$ is one if the model predicts that the car will change lanes at time *t* and is zero otherwise, $car_{lc}(t)$ is one if the car changes lanes at time *t* and is zero otherwise. The vehicle model and driver model are also reset every three seconds so that initial values of *ex* and *ey* are always zeros in each time interval. Aggressiveness is estimated by combining *ex* and *ey*, where *ey* is dominant. It is updated in real time, as shown below.

$$\begin{aligned} q(t_k) = q(t_{k-1}) + K_{px}[e_x(t_k) - e_x(t_{k-1})] + K_{ix}e_x(t_k) \\ + K_{dx}[e_x(t_k) - 2e_x(t_{k-1}) + e_x(t_{k-2})] + K_{iy}e_y(t_k) \end{aligned} \tag{3.34}$$

where q is the aggressiveness, K_{px}, K_{ix}, K_{dx} and K_{iy} are estimation parameters. Ag is a cumulative value. If the real driving behavior does not deviate obviously from the predictive driving behavior, Ag will decay to zero due to the integral term. The fault detector works by observing the transient difference between the predictive model and real car. It is given by

$$detector = \sqrt{w_1 * D_{fx}^2 + w_2 * D_{fy}^2} \tag{3.35}$$

where D_{fx} is the difference between the real longitudinal position and predicted longitudinal position and D_{fy} is the difference between the real lateral position and predicted lateral position, w_1 and w_2 are weights of D_{fx} and D_{fy}.

3.5 Test results and discussion

Several tests were conducted by a human driver in Simulink/dSPACE. In the present study, the road has two lanes. UA was simulated by adding an error to throttle positions randomly.

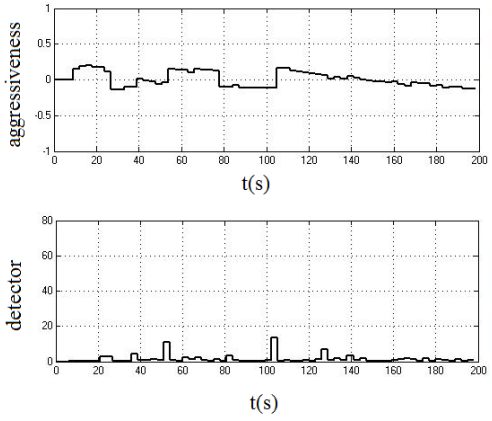

Figure 3.19 Normal driving

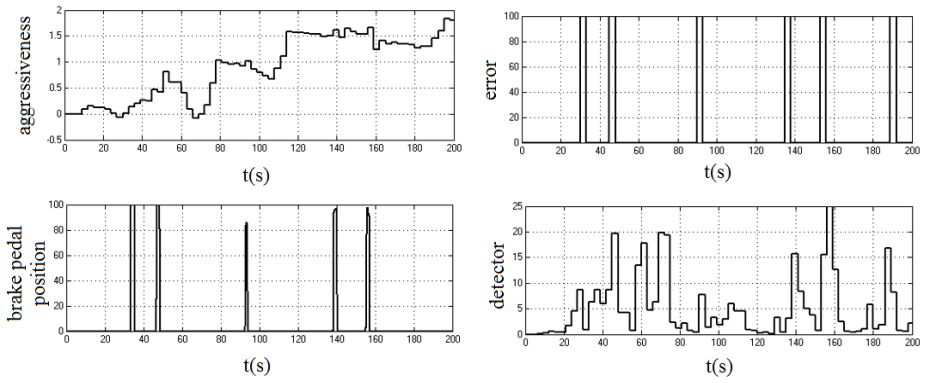

Figure 3.20 UA occurs randomly

Figure 3.19 shows the test result of normal driving. It can be seen that the aggressiveness does not fluctuate a lot and stays at a low level. The detector curve is also relatively flat, which meets expectations. Figure 3.20 shows driving with random UA errors. The duration of the error is three seconds. It can be found that the estimated aggressiveness keeps increasing and reaches a big value before it decays. The aggressiveness is a cumulative value and indicates the status of the driver and vehicle based on historical driving data. As expected, the aggressiveness of the test with UA is much higher than that of normal driving. Another important finding is that when UA occurs, the detector values are relatively high, which means the detector is helpful for detecting UA. However, there are also some problems. The test result shows that if the driver presses the brake pedal and controls the car carefully when UA occurs, there is no big difference between the predicted vehicle position and real vehicle position. In addition, human drivers change driving patterns suddenly sometimes. It leads to high detector values, which are not caused by vehicle errors.

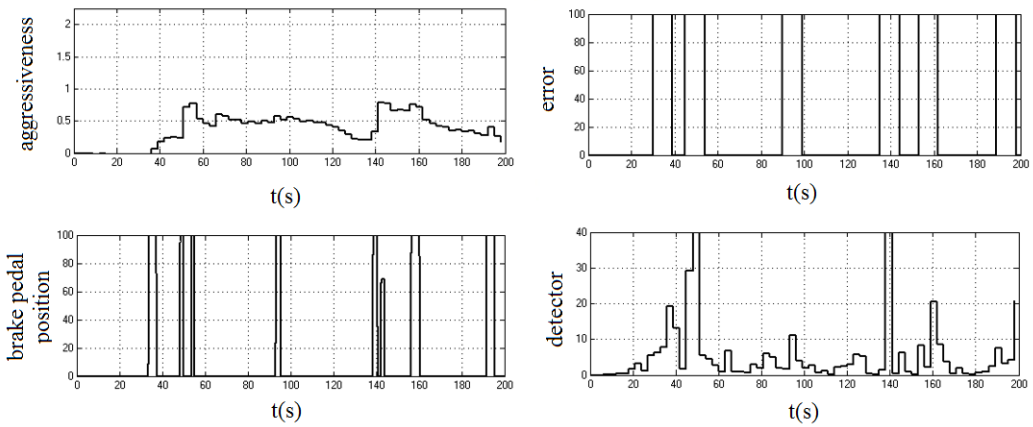

Figure 3.21 First test for UA lasting for seven seconds

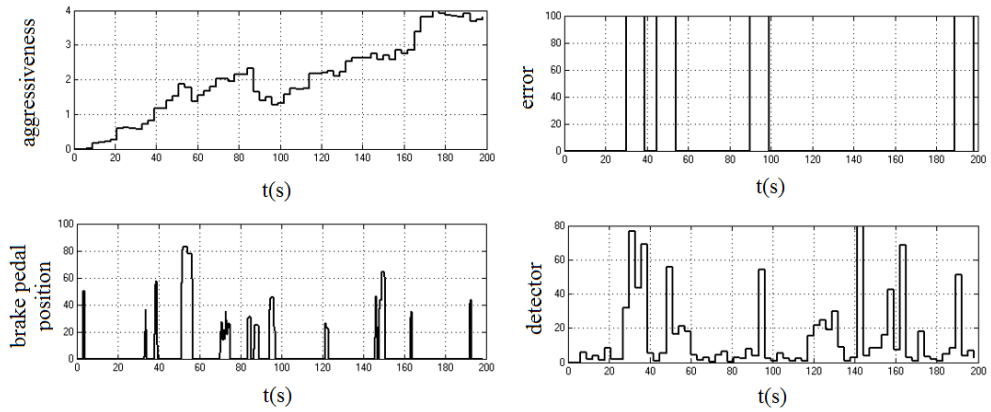

Figure 3.22 Second test for UA lasting for seven seconds

Figure 3.21 - Figure 3.22 show the test results when the duration of errors is extended to seven seconds. It can be seen that the relationship between errors and detector values is more obvious. It is reasonable because when the error lasts longer, it is more difficult to control the car. Therefore, real vehicle positions deviate considerably from predicted positions, which result in large detector values. Another interesting finding is that the driver in the second test could not control the visual car well and the estimated aggressiveness is much higher than that of the driver in the first test.

3.6 Summary

In this chapter, a model-based detection system for UA and other abnormal driving conditions is demonstrated. A rational and adaptive driver model is developed and combined with a vehicle model to predict the future behavior of the vehicle. The results of prediction are compared with real measurements from sensors. If the car driven by a human driver deviates largely from the trajectory of the model, the system concludes that an abnormal event has occurred. The detection system is validated in the dSPACE

driving simulator. Test results show that it is capable of indicating abnormal driving conditions. A persistent pattern of anomalies may imply a more drastic situation such as UA or else due to the driver becoming seriously erratic (e.g., incidents of road rage). Whether it is possible to differentiate the two cases, it is not clear at this point. Integration of physiological sensing may help differentiate these cases [106] [107] but this itself requires additional evaluation (and more sophisticated and complex experiment design). In summary, though the accuracy of the model-based detection system is not extremely high, it is a strong indicator of abnormal events and a good complement to the data-driven detector.

4. CONCLUSIONS

In this study, both a data-driven detector and a driver model-based detector were developed for vehicle anomalies originating in UA. These two detectors were integrated to increase the detection accuracy.

In the first part, a neural network-based detection system was designed. At first, the model was tested using the data collected from Matlab / Simulink. A vehicle model with six degrees of freedom was built. The model also contained tire models, an engine, an automatic transmission system and other powertrain components. A driver model was created to control steering wheels and pedals. Simulation results showed that the detector was able to differentiate abnormal acceleration from normal acceleration and its accuracy was beyond 95%. After that, the detection model was also tested in dSPACE. A brake assist system was developed to cooperate with the detection system. Test results showed that the detection system was able to detect anomalies in one second with an accuracy of 99%. The brake assist system received warnings from the detector and was capable of helping the driver stop the car safely.

In the second part, a rational and adaptive driver model was created. The longitudinal model is similar to an adaptive cruise control system while the lateral model is based on a Stackelberg game, which mimics the decision-making process of a human driver in the lane-changing scenario. The behavior of this driver model was compared with the behavior of a human driver. When the difference between the driver model and the human driver was abnormally large, the system concluded that an abnormal event had

occurred. Simulation results show that the driver model-based detector could give timely warnings when UA occurred. The driver model could also be extended to the emerging field of autonomous driving.

In summary, effective models that can detect and mitigate UA have been presented. The proposed approaches could be extended to other vehicle problems. For example, similar structures could be used for detecting the failure of electric power steering. The findings of this study could increase safety on the roadway.

REFERENCES

[1] J.M. Anderson, K. Nidhi, K.D. Stanley, P. Sorensen, C. Samaras, and O.A. Oluwatola, *Autonomous vehicle technology: A guide for policymakers*. Rand Corporation, 2014, ISBN: 0833084372.

[2] NHTSA, "Technical assessment of toyota electronic throttle control (ETC) systems," 2011, Available: https://acc.dau.mil/adl/en-US/494771/file/62425/NHTSA-UA_report.pdf, Accessed on: March 30,2016.

[3] S. Kane, E. Liberman, T. DiViesti, and F. Click, "Toyota sudden unintended acceleration," Safety Research & Strategies, 2010, Available: http://www.rightinginjustice.com/media/2010/02/SRS-Report-on-Toyota-Sudden-Unintended-Acceleration.pdf, Accessed on: March 30, 2016.

[4] J. Pollard and E.D. Sussman, "An examination of sudden acceleration," NHTSA, 1989, Available: http://ntl.bts.gov/lib/33000/33400/33403/33403.pdf, Accessed on: March 30, 2016.

[5] NASA, "Technical support to the national highway traffic safety administration (NHTSA) on the reported toyota motor corporation (TMC) unintended acceleration (UA) investigation," 2011, Available: www.nhtsa.gov/staticfiles/nvs/pdf/NASA-UA_report.pdf, Accessed on: March 30, 2016.

[6] National Research Council, *The safety promise and challenge of automotive electronics: Insights from unintended acceleration*. Transportation Research Board, 2012, ISBN: 0309223040.

[7] K.S. Linebaugh, Dionne (2010). *Cause of sudden acceleration proves hard to pinpoint*. Available: http://www.wsj.com/articles/SB10001424052748703510204575085531383717288, Accessed on: March 31, 2016.

[8] R. Mariani, "Soft errors on digital components," in *Fault injection techniques and tools for embedded systems reliability evaluation*: Springer, 2003, pp. 49-60.

[9] P. Koopman, "A case study of toyota unintended acceleration and software safety," presented at the TSP Symposium, Pittsburgh, PA, 2014. Available: https://users.ece.cmu.edu/~koopman/pubs/koopman14_toyota_ua_slides.pdf

[10] B. Sood, M. Osterman, and M. Pecht, "Tin whisker analysis of toyota's electronic throttle controls," *Circuit World,* vol. 37, no. 3, pp. 4-9, 2011.

[11] E. George and M. Pecht, "Tin whisker analysis of an automotive engine control unit," *Microelectronics Reliability,* vol. 54, no. 1, pp. 214-219, 2014.

[12] Q. Bi, P. Li, M. Newbury, and Q. Zhang, "Method of wireless vehicle diagnosis," ed: Google Patents, 2005.

[13] M. Kapolka, S. Chang, B. Crull, A. Ditchfield, and W. Bromley, "Remote monitoring, configuring, programming and diagnostic system and method for vehicles and vehicle components," ed: Google Patents, 2006.

[14] S. Ding, *Model-based fault diagnosis techniques: Design schemes, algorithms, and tools*. Springer Science & Business Media, 2008, ISBN: 354076304X.

[15] I. Hwang, S. Kim, Y. Kim, and C.E. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *Control Systems Technology, IEEE Transactions on,* vol. 18, no. 3, pp. 636-653, 2010.

[16] I. Samy, I. Postlethwaite, and D.-W. Gu, "Survey and application of sensor fault detection and isolation schemes," *Control Engineering Practice,* vol. 19, no. 7, pp. 658-674, 2011.

[17] F. Meinguet, P. Sandulescu, X. Kestelyn, and E. Semail, "A method for fault detection and isolation based on the processing of multiple diagnostic indices: Application to inverter faults in ac drives," *Vehicular Technology, IEEE Transactions on,* vol. 62, no. 3, pp. 995-1009, 2013.

[18] R. Isermann, "Model-based fault-detection and diagnosis–status and applications," *Annual Reviews in control,* vol. 29, no. 1, pp. 71-85, 2005.

[19] R. Isermann, "Diagnosis methods for electronic controlled vehicles," *Vehicle System Dynamics,* vol. 36, no. 2-3, pp. 77-117, 2001.

[20] M. Würtenberger and R. Isermann, "Supervision of lateral vehicle motion using a discrete parity space approach," *Advances in Automotive Control 1995,* vol. 75, 1995.

[21] O. Moseler and R. Isermann, "Application of model-based fault detection to a brushless dc motor," *Industrial Electronics, IEEE Transactions on,* vol. 47, no. 5, pp. 1015-1020, 2000.

[22] M. Schwabacher, "A survey of data-driven prognostics," in *Proceedings of the AIAA Infotech@ Aerospace Conference*, 2005, pp. 1-5.

[23] S. Bahrampour, B. Moshiri, and K. Salahshoor, "Weighted and constrained possibilistic c-means clustering for online fault detection and isolation," *Applied Intelligence*, vol. 35, no. 2, pp. 269-284, 2011.

[24] C. Furse, P. Smith, C. Lo, Y.C. Chung, P. Pendayala, and K. Nagoti, "Spread spectrum sensors for critical fault location on live wire networks," *Structural Control and Health Monitoring,* vol. 12, no. 3 - 4, pp. 257-267, 2005.

[25] J. Liu, "Shannon wavelet spectrum analysis on truncated vibration signals for machine incipient fault detection," *Measurement Science and Technology,* vol. 23, no. 5, pp. 055604, 2012.

[26] H. Yu and R. Langari, "A detection and warning system for unintended acceleration," in *ASME 2015 Dynamic Systems and Control Conference*, 2015, pp. V002T31A002-V002T31A002: American Society of Mechanical Engineers.

[27] R.K. Schmidt, T. Leinmüller, E. Schoch, A. Held, and G. Schäfer, "Vehicle behavior analysis to enhance security in vanets," in *Proceedings of the 4th IEEE Vehicle-to-Vehicle Communications Workshop (V2VCOM2008)*, 2008.

[28] S. Hai-Feng, W. Hui, and W. Dan-Yang, "Vehicle abnormal behavior detection system based on video," in *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on*, 2012, vol. 1, pp. 132-135: IEEE.

[29] P.L.M. Bouttefroy, A. Beghdadi, A. Bouzerdoum, and S.L. Phung, "Markov random fields for abnormal behavior detection on highways," in *Visual*

Information Processing (EUVIP), 2010 2nd European Workshop on, 2010, pp. 149-154: IEEE.

[30] Y. Chen and J. Phillips, "ECU software abnormal behavior detection based on mahalanobis-taguchi technique," *Detroit, Michigan: SAE International*, pp. 474-480, 2008.

[31] B. Jeppesen and D. Cebon, "Application of observer-based fault detection in vehicle roll control," *Vehicle System Dynamics*, vol. 47, no. 4, pp. 465-495, 2009.

[32] D. McKay, G. Nichols, and B. Schreurs, "Delphi electronic throttle control systems for model year 2000: Driver features, system security, and OEM benefits: ETC for the mass market," *SAE Technical Paper*, 2000, Art. no. 2000-01-0556. doi: 10.4271/2000-01-0556.

[33] R. Garrick. (2006). *Sensitivity of contact electronic throttle control sensor to control system variation*. Available: http://scholarworks.rit.edu/article/477, Accessed on: March 30, 2016.

[34] M. Barr. (2013). *2005 camry l4 software analysis*. Available: http://www.safetyresearch.net/Library/BarrSlides_FINAL_SCRUBBED.pdf, Accessed on: March 1, 2016.

[35] R.K. Jurgen, "Adaptive cruise control," *Training*, vol. 2011, pp. 04-08, 2006.

[36] L.G. Ham, Robyn. (2009). *Cruise control terror for freeway driver*. Available: http://www.smh.com.au/national/cruise-control-terror-for-freeway-driver-20091215-ktxn.html, Accessed on: March 1, 2016.

[37] AGCO Automotive Corporation. *Vacuum brake boosters*. Available: http://www.agcoauto.com/content/news/p2_articleid/129, Accessed on: March 1, 2016.

[38] NHTSA, "Vehicle characterization and performance tests of toyota camrys," 2011, Available: https://www.nhtsa.gov/staticfiles/nvs/pdf/NHTSA-Toyota_vehicle_characterization.pdf, Accessed on: March 1, 2016.

[39] M. Brackstone and M. McDonald, "Car-following: A historical review," *Transportation Research Part F: Traffic Psychology and Behaviour,* vol. 2, no. 4, pp. 181-196, 1999.

[40] R. Benekohal and J. Treiterer, "Carsim: Car-following model for simulation of traffic in normal and stop-and-go conditions," *Transportation Research Record,* no. 1194, 1988.

[41] A.D. Mason and A.W. Woods, "Car-following model of multispecies systems of road traffic," *Physical Review E,* vol. 55, no. 3, pp. 2203, 1997.

[42] G.F. Newell, "Nonlinear effects in the dynamics of car following," *Operations Research,* vol. 9, no. 2, pp. 209-229, 1961.

[43] J. Tyler, "The characteristics of model-following systems as synthesized by optimal control," *IEEE Transactions on Automatic Control,* vol. 9, no. 4, pp. 485-498, 1964.

[44] G. Burnham, J. Seo, and G. Bekey, "Identification of human driver models in car following," *IEEE transactions on Automatic Control,* vol. 19, no. 6, pp. 911-915, 1974.

[45] P.G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological,* vol. 15, no. 2, pp. 105-111, 1981.

[46] K. Lee and H. Peng, "Identification of a longitudinal human driving model for adaptive cruise control performance assessment," in *ASME 2002 International Mechanical Engineering Congress and Exposition*, 2002, pp. 675-682: American Society of Mechanical Engineers.

[47] K. Lee and H. Peng, "Identification and verification of a longitudinal human driving model for collision warning and avoidance systems," *International Journal of Vehicle Autonomous Systems,* vol. 2, no. 1-2, pp. 3-17, 2004.

[48] P. Hidas, "Modelling lane changing and merging in microscopic traffic simulation," *Transportation Research Part C: Emerging Technologies,* vol. 10, no. 5, pp. 351-371, 2002.

[49] P.G. Gipps, "A model for the structure of lane-changing decisions," *Transportation Research Part B: Methodological,* vol. 20, no. 5, pp. 403-414, 1986.

[50] S. Moridpour, M. Sarvi, and G. Rose, "Lane changing models: A critical review," *Transportation Letters,* vol. 2, no. 3, pp. 157-173, 2013. doi: 10.3328/TL.2010.02.03.157-173.

[51] K. Ahmed, M. Ben-Akiva, H. Koutsopoulos, and R. Mishalani, "Models of freeway lane changing and gap acceptance behavior," *Transportation and Traffic Theory,* vol. 13, pp. 501-515, 1996.

[52] J.A. Laval and L. Leclercq, "Microscopic modeling of the relaxation phenomenon using a macroscopic lane-changing model," *Transportation Research Part B: Methodological*, vol. 42, no. 6, pp. 511-522, 2008.

[53] J.H. Yoo and R. Langari, "A stackelberg game theoretic driver model for merging," in *ASME 2013 Dynamic Systems and Control Conference*, 2013, pp. V002T30A003-V002T30A003: American Society of Mechanical Engineers.

[54] Q. Yang and H.N. Koutsopoulos, "A microscopic traffic simulator for evaluation of dynamic traffic management systems," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 3, pp. 113-129, 1996.

[55] T. Toledo, H. Koutsopoulos, and M. Ben-Akiva, "Modeling integrated lane-changing behavior," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1857, pp. 30-38, 2003.

[56] R. Myerson, "Game theory: Analysis of conflictharvard univ," *Press, Cambridge*, 1991.

[57] J.F. Nash, "Equilibrium points in n-person games," *Proc. Nat. Acad. Sci. USA*, vol. 36, no. 1, pp. 48-49, 1950.

[58] H. Von Stackelberg, *Market structure and equilibrium*. Springer Science & Business Media, 2010, ISBN: 3642125867.

[59] J.H. Yoo and R. Langari, "Stackelberg game based model of highway driving," in *ASME Dynamic Systems and Control Conference (DSCC)*, 2012, pp. 499-508.

[60] C. Fisk, "Game theory and transportation systems modelling," *Transportation Research Part B: Methodological*, vol. 18, no. 4, pp. 301-313, 1984.

[61] A. Talebpour, H.S. Mahmassani, and S.H. Hamdar, "Modeling lane-changing behavior in a connected environment: A game theory approach," *Transportation Research Part C: Emerging Technologies*, vol. 59, pp. 216-232, 2015.

[62] I. Alvarez, A. Poznyak, and A. Malo, "Urban traffic control problem a game theory approach," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, 2008, pp. 2168-2172: IEEE.

[63] H. Yu and R. Langari, "Detection of unintended acceleration in longitudinal car following," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 8, no. 2015-01-0208, pp. 306-313, 2015.

[64] D.J. LeBlanc, M. Sivak, and S. Bogard, "Using naturalistic driving data to assess variations in fuel efficiency among individual drivers," UMTRI, Rep: UMTRI-2010-34, 2010, Available: https://deepblue.lib.umich.edu/bitstream/handle/2027.42/78449/102705.pdf, Accessed on: March 1, 2016.

[65] SAE International Surface Vehicle Recommended Practice, "Vehicle dynamics terminology," ed: SAE Standard J670, 2008.

[66] H. Sugiyama and Y. Suda, "Non-linear elastic ring tyre model using the absolute nodal coordinate formulation," *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 223, no. 3, pp. 211-219, 2009.

[67] N. Frey, "Development of a rigid ring tire model and comparison among various tire models for ride comfort simulations," M.S. Thesis, Dept. Mechanical Engineering, Clemson University, Clemson, SC, 2009.

[68] J. Svendenius and M. Gäfvert, "A brush-model based semi-empirical tire-model for combined slips," *SAE Technical Paper*, 2004, Art. no. 2004-01-1064. doi: 10.4271/2004-01-1064.

[69] H.B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle System Dynamics,* vol. 21, no. S1, pp. 1-18, 1992.

[70] J.C. Herrera, D.B. Work, R. Herring, X.J. Ban, Q. Jacobson, and A.M. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: The mobile century field experiment," *Transportation Research Part C: Emerging Technologies,* vol. 18, no. 4, pp. 568-583, 2010.

[71] D. Chen, P. Lo, and W. Swan, "Zero-lag exponential moving average for real-time control and noisy data processing," *Hydrocarbon Processing,* vol. 86, no. 10, pp. 61-61, 2007.

[72] H. Ohnishi, J. Ishii, M. Kayano, and H. Katayama, "A study on road slope estimation for automatic transmission control," *JSAE Review*, vol. 21, no. 2, pp. 235-240, 2000.

[73] Y. Sebsadji, S. Glaser, S. Mammar, and J. Dakhlallah, "Road slope and vehicle dynamics estimation," in *American Control Conference, 2008*, 2008, pp. 4603-4608: IEEE.

[74] D.M. Bevly, "Global positioning system (GPS): A low-cost velocity sensor for correcting inertial sensor errors on ground vehicles," *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, no. 2, pp. 255-264, 2004.

[75] F. Takens, *Detecting strange attractors in turbulence*. Springer, 1981, ISBN: 3540111719.

[76] J. Stark, D. Broomhead, M. Davies, and J. Huke, "Takens embedding theorems for forced and stochastic systems," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 30, no. 8, pp. 5303-5314, 1997.

[77] J. Stark, D.S. Broomhead, M. Davies, and J. Huke, "Delay embeddings for forced systems. Ii. Stochastic forcing," *Journal of Nonlinear Science*, vol. 13, no. 6, pp. 519-577, 2003.

[78] J. Stark, "Delay embeddings for forced systems. I. Deterministic forcing," *Journal of Nonlinear Science*, vol. 9, no. 3, pp. 255-332, 1999.

[79] M.B. Kennel, R. Brown, and H.D. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical Review A*, vol. 45, no. 6, pp. 3403, 1992.

[80] B. MacIsaac and R. Langton, *Gas turbine propulsion systems*. John Wiley & Sons, 2011, ISBN: 1119976146.

[81] C.M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006, ISBN: 9780387310732.

[82] W.T. Miller, P.J. Werbos, and R.S. Sutton, *Neural networks for control*. MIT press, 1995, ISBN: 026263161X.

[83] T.M. Mitchell, *Machine learning. Wcb*. McGraw-Hill Boston, MA:, 1997, ISBN: 0070428077.

[84] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern classification*. John Wiley & Sons, 2012, ISBN: 111858600X.

[85] R.N. Jazar, *Vehicle dynamics: Theory and application*. Springer Science & Business Media, 2013, ISBN: 1461485444.

[86] M.E. Russell, M.J. Delcheccolo, W.G. Woodington, H.B. Van Rees, J.M. Firda, and D. Lippert, "Safe distance algorithm for adaptive cruise control," ed: Google Patents, 2004.

[87] G. Yan, W. Yang, M.C. Weigle, S. Olariu, and D. Rawat, "Cooperative collision warning through mobility and probability prediction," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 2010, pp. 1172-1177: IEEE.

[88] Y. Kuwata, G.A. Fiore, J. Teo, E. Frazzoli, and J.P. How, "Motion planning for urban driving using RRT," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1681-1686: IEEE.

[89] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J.P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105-1118, 2009.

[90] J. Bruce and M. Veloso, "Real-time randomized path planning for robot navigation," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, 2002, vol. 3, pp. 2383-2388: IEEE.

[91] K.-D. Kim and P.R. Kumar, "An mpc-based approach to provable system-wide safety and liveness of autonomous ground traffic," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3341-3356, 2014.

[92] J. Nilsson and J. Sjöberg, "Strategic decision making for automated driving on two-lane, one way roads using model predictive control," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013, pp. 1253-1258: IEEE.

[93] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "Mpc-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems*, vol. 3, no. 2-4, pp. 265-291, 2005.

[94] D.R. Drew, L.R. LaMotte, J.A. Wattleworth, and J.H. Buhr, "Gap acceptance in the freeway merging process," *Highway Research Record*, no. 208, pp. 1-36, 1967.

[95] H. Mahmassani and Y. Sheffi, "Using gap sequences to estimate gap acceptance functions," *Transportation Research Part B: Methodological*, vol. 15, no. 3, pp. 143-148, 1981.

[96] H. Farah, S. Bekhor, A. Polus, and T. Toledo, "A passing gap acceptance model for two-lane rural highways," *Transportmetrica*, vol. 5, no. 3, pp. 159-172, 2009.

[97] C. Liu and M. Tomizuka, "Enabling safe freeway driving for automated vehicles," in *2016 American Control Conference (ACC)*, 2016, pp. 3461-3467: IEEE.

[98] F. Qiao, R. Rahman, Q. Li, and L. Yu, "Safe and environment-friendly forward collision warning messages in the advance warning area of a construction zone,"

International Journal of Intelligent Transportation Systems Research, pp. 1-14, 2016.

[99] K.I. Ahmed, "Modeling drivers' acceleration and lane changing behavior," Ph.D. dissertation, Dept. Civil and Environmental Engineering, Massachusetts Institute of Technology, Boston, MA, 1999.

[100] D. Shinar and R. Compton, "Aggressive driving: An observational study of driver, vehicle, and situational variables," *Accident Analysis & Prevention,* vol. 36, no. 3, pp. 429-437, 2004.

[101] W. Vanlaar, H. Simpson, D. Mayhew, and R. Robertson, "Aggressive driving: A survey of attitudes, opinions and behaviors," *Journal of Safety Research,* vol. 39, no. 4, pp. 375-381, 2008.

[102] K.H. Beck, M.Q. Wang, and M.M. Mitchell, "Concerns, dispositions and behaviors of aggressive drivers: What do self-identified aggressive drivers believe about traffic safety?," *Journal of Safety Research,* vol. 37, no. 2, pp. 159-165, 2006.

[103] S. Huang, K.K. Tan, and T.H. Lee, "Autonomous cruise control using neural networks in platooning," *Advanced Robotics,* vol. 19, no. 2, pp. 169-189, 2005.

[104] S.-N. Huang, S.C. Chan, and W. Ren, "Mixed traffic control involving manually-controlled and automatically-controlled vehicles in ivhs," in *Advances in intelligent autonomous systems*: Springer, 1999, pp. 507-528.

[105] A. Sinha, P. Malo, and K. Deb, "Efficient evolutionary algorithm for single-objective bilevel optimization," *arXiv preprint arXiv:1303.3901,* 2013.

[106] J. Ogorevc, G. Geršak, D. Novak, and J. Drnovšek, "Metrological evaluation of skin conductance measurements," *Measurement,* vol. 46, no. 9, pp. 2993-3001, 2013.

[107] A. Purwandini Sutarto, M.N. Abdul Wahab, and N. Mat Zin, "Resonant breathing biofeedback training for stress reduction among manufacturing operators," *International Journal of Occupational Safety and Ergonomics,* vol. 18, no. 4, pp. 549-561, 2012.

APPENDIX A VEHICLE DYNAMICS

Section 2 gives a brief introduction of vehicle dynamics. This section presents some more details, which are also important in a vehicle model, but not discussed in section 2. Tire modelling is critical for driving simulation. It could be up to 50% of a vehicle model. One of the most widely-used tire models is Pacejka tire model (Magic Formula) [69], in which tire forces and torques are functions of slip angles and slip ratios. A slip angle is the angle between the direction towards which a rolling wheel is pointing and the actual direction of travel of the wheel, as shown below [69].

$$\alpha = \begin{cases} \delta - \arctan\left(\frac{V_{yi}}{V_{xi}}\right) & Front\ wheel \\ -\arctan\left(\frac{V_{yi}}{V_{xi}}\right) & Rear\ wheel \end{cases} \tag{A.1}$$

where α denotes the slip angle, δ is the steering angle, V_{yi} represents the lateral velocity and V_{xi} denotes the longitudinal velocity of the wheel. V_{xi} and V_{yi} of different wheels are shown in Figure A-1.

$$\mathrm{V_{x1}} = V_x + \omega_z * \frac{TW_f}{2} \tag{A.2}$$

$$\mathrm{V_{x2}} = V_x - \omega_z * \frac{TW_f}{2} \tag{A.3}$$

$$\mathrm{V_{x3}} = V_x + \omega_z * \frac{TW_r}{2} \tag{A.4}$$

$$\mathrm{V_{x4}} = V_x - \omega_z * \frac{TW_r}{2} \tag{A.5}$$

$$\mathrm{V_{y1}} = V_y + \omega_z * L_f \tag{A.6}$$

$$\mathrm{V_{y2}} = V_y + \omega_z * L_f \tag{A.7}$$

$$\mathrm{V_{y3}} = V_y - \omega_z * L_r \tag{A.8}$$

$$\mathrm{V_{y4}} = V_y - \omega_z * L_r \tag{A.9}$$

where V_x and V_y are longitudinal and lateral velocities of the vehicle, TW_f and TW_r are front and rear track width, L_f and L_r are the distance from center of gravity (CoG) to the front and rear axles and ω_z denotes the yaw rate.

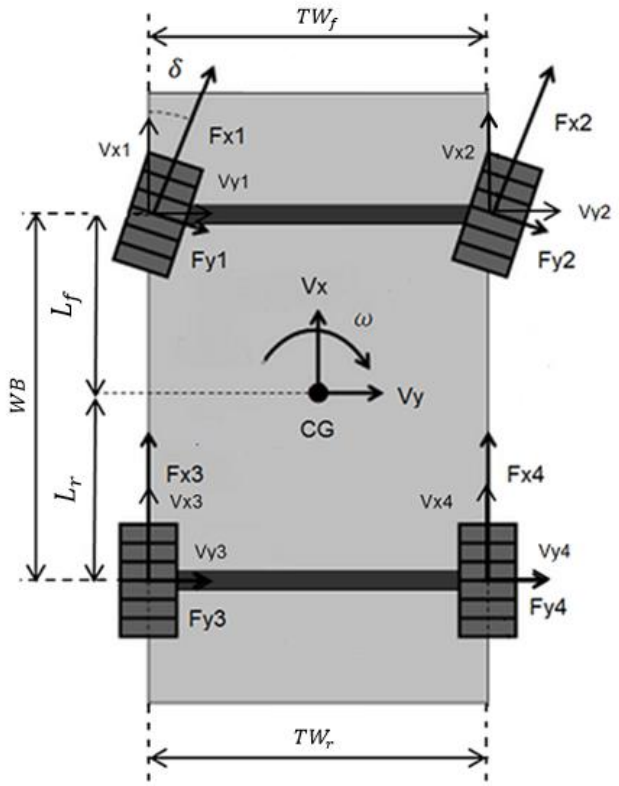

Figure A-1 Wheel forward velocities and lateral velocities

Slip ratios are calculated by Equation (2.9). Longitudinal tire forces, lateral tire forces and aligning moments are given by [69]

$$F_{xi} = D_x sin(C_x \arctan\big(B_x\sigma - E_x(B_x\sigma - arctan(B_x\sigma))\big) \tag{A.10}$$

$$F_{yi} = D_y sin(C_y \arctan\Big(B_y\alpha - E_y\big(B_y\alpha - arctan\big(B_y\alpha\big)\big)\Big) \tag{A.11}$$

$$\phi_i = (1 - E_z) * \alpha + \frac{E_z}{B_z} * \arctan(B_z * \alpha) \tag{A.12}$$

$$M_{zi} = D_z \sin(C_z * \arctan(B_z * \phi_i)) \tag{A.13}$$

where F_{xi} denotes the longitudinal tire force, F_{yi} denotes the lateral tire force, M_{zi} is the aligning moment, σ is the slip ratio, $B_x, C_x, D_x, E_x, B_y, C_y, D_y, B_z, C_z, D_z$ and E_z are functions of tire normal forces. Tire normal forces are affected by the slope angle of the road, as shown in Figure A-2.

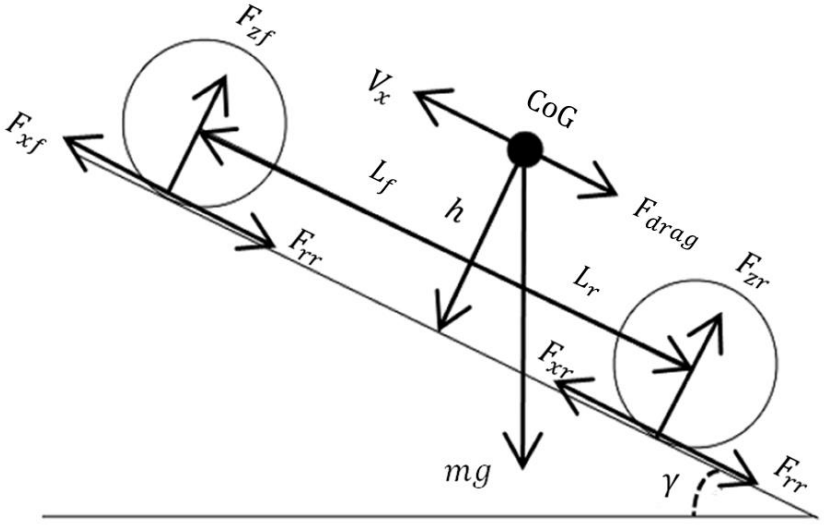

Figure A-2 Tire force

The road slope angle is denoted by γ and the tire normal force is denoted by F_z. F_z consist of four parts: the static vertical load F_s , the longitudinal load transfer due to acceleration F_p, the lateral load transfer due to rolling F_r and the lateral load transfer due to acceleration of the unsprung mass F_u. Tire normal forces are given by

$$F_{z1} = F_{s1} + F_{p1} + F_{r1} + F_{u1}$$

$$= \frac{-hF_{drag} - \left(m_c h + m_{uf}h_{uf} + m_{ur}h_{ur}\right)gsin\gamma + mgL_r cos\gamma}{2WB}$$

$$- \frac{\left(m_c h + m_{uf}h_{uf} + m_{ur}h_{ur}\right) * a_x}{2WB} \qquad \text{(A.14)}$$

$$+ \frac{\left(a_y cos\phi + gsin\phi\right) * m_c d_{roll} * K_{rollf}}{TW_f\left(K_{rollf} + K_{rollr}\right)} + \frac{m_c a_y h_{uf} * \frac{L_r}{WB} + m_{uf}a_y h_{uf}}{TW_f}$$

$$F_{z2} = F_{s2} + F_{p2} + F_{r2} + F_{u2}$$

$$= \frac{-hF_{drag} - \left(m_c h + m_{uf}h_{uf} + m_{ur}h_{ur}\right)gsin\gamma + mgL_r cos\gamma}{2WB}$$

$$- \frac{\left(m_c h + m_{uf}h_{uf} + m_{ur}h_{ur}\right) * a_x}{2WB} \qquad \text{(A.15)}$$

$$- \frac{\left(a_y cos\phi + gsin\phi\right) * m_c d_{roll} * K_{rollf}}{TW_f\left(K_{rollf} + K_{rollr}\right)} - \frac{m_c a_y h_{uf} * \frac{L_r}{WB} + m_{uf}a_y h_{uf}}{TW_f}$$

$$F_{z3} = F_{s3} + F_{p3} + F_{r3} + F_{u3}$$

$$= \frac{hF_{drag} + \left(m_c h + m_{uf} h_{uf} + m_{ur} h_{ur}\right) g sin\gamma + mgL_f cos\gamma}{2WB}$$

$$+ \frac{\left(m_c h + m_{uf} h_{uf} + m_{ur} h_{ur}\right) * a_x}{2WB} \tag{A.16}$$

$$+ \frac{\left(a_y cos\phi + g sin\phi\right) * m_c d_{roll} * K_{rollr}}{TW_r\left(K_{rollf} + K_{rollr}\right)} + \frac{m_c a_y h_{ur} * \frac{L_f}{WB} + m_{ur} a_y h_{ur}}{TW_r}$$

$$F_{z4} = F_{s4} + F_{p4} + F_{r4} + F_{u4}$$

$$= \frac{hF_{drag} + \left(m_c h + m_{uf} h_{uf} + m_{ur} h_{ur}\right) g sin\gamma + mgL_f cos\gamma}{2WB}$$

$$+ \frac{\left(m_c h + m_{uf} h_{uf} + m_{ur} h_{ur}\right) * a_x}{2WB} \tag{A.17}$$

$$- \frac{\left(a_y cos\phi + g sin\phi\right) * m_c d_{roll} * K_{rollr}}{TW_r\left(K_{rollf} + K_{rollr}\right)} - \frac{m_c a_y h_{ur} * \frac{L_f}{WB} + m_{ur} a_y h_{ur}}{TW_r}$$

where a_x and a_y denote the longitudinal and lateral acceleration of the vehicle, F_{drag} is the aerodynamic drag force, m_{uf} and m_{ur} are the front and rear unsprung mass, h_{uf} and h_{ur} are the height of front and rear unsprung mass CoGs, m_c is the sprung mass, , h is the height of sprung mass CoG, d_{roll} is the length of the roll moment arm, K_{rollf} and K_{rollr} are front and rear roll stiffness and WB is the wheel base.

APPENDIX B VEHICLE MODEL IN SIMULINK

A vehicle model with six degrees of freedom was built in Matlab / Simulink. The model contained tire models, an engine, an automatic transmission system and other powertrain components. The model was tested in a 3D environment. A traffic model was created. Side view mirrors and the rear view mirrors were also simulated. A human driver is able to drive a virtual vehicle using analog steering wheels and pedals.

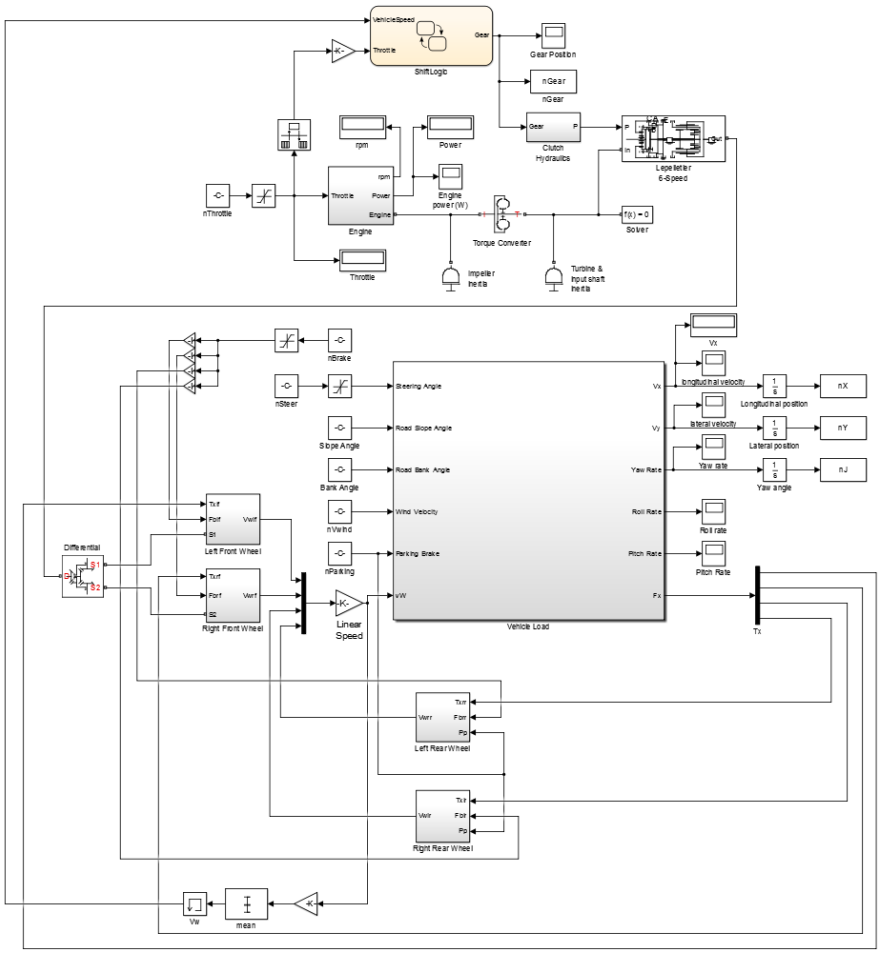

Figure B-1 Powertrain

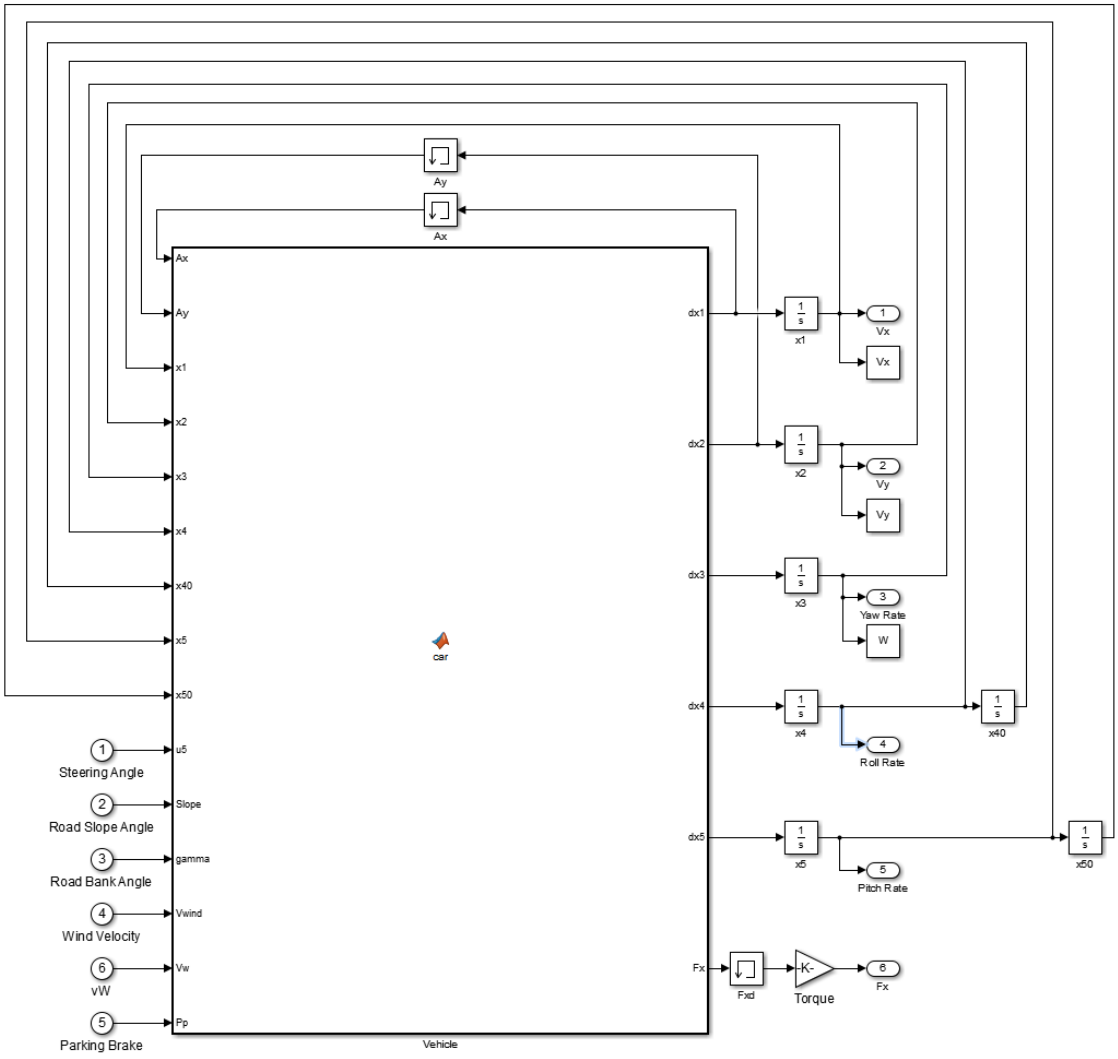

Figure B-2 Vehicle dynamics simulation

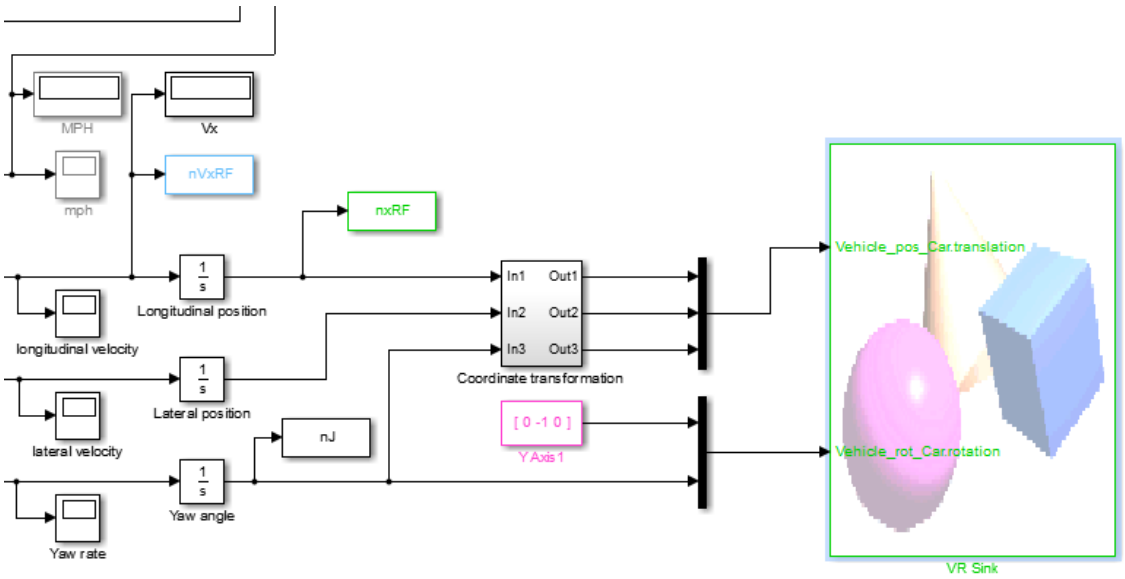

Figure B-3 Coordinate transformation and 3D simulation

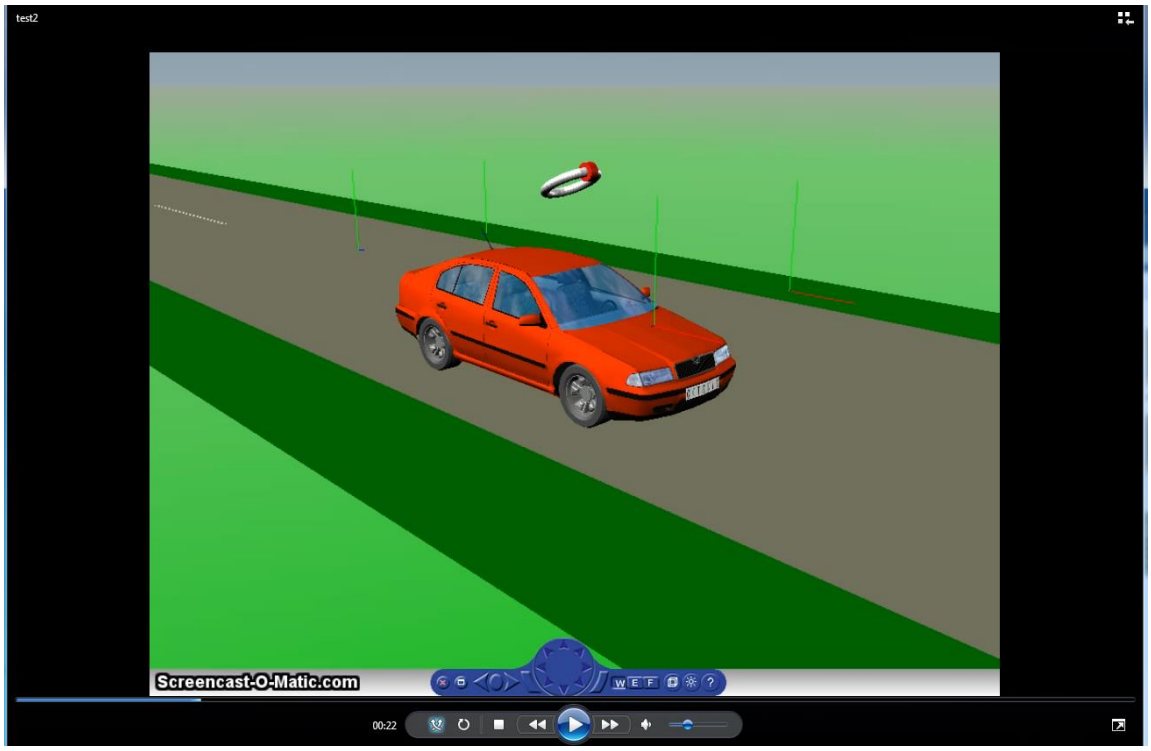

Figure B-4 Control the vehicle using analog steering wheels and pedals

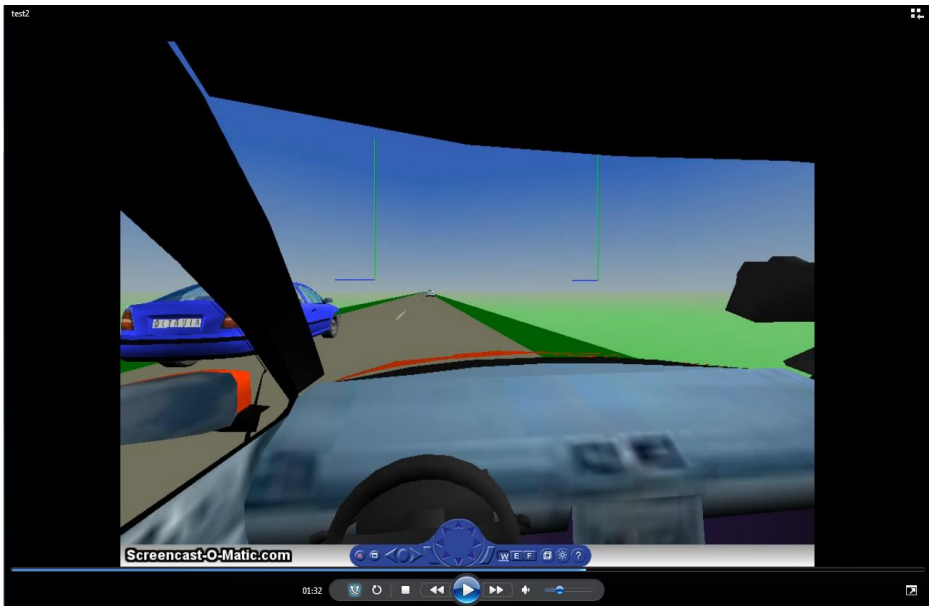

Figure B-5 Add traffic simulation

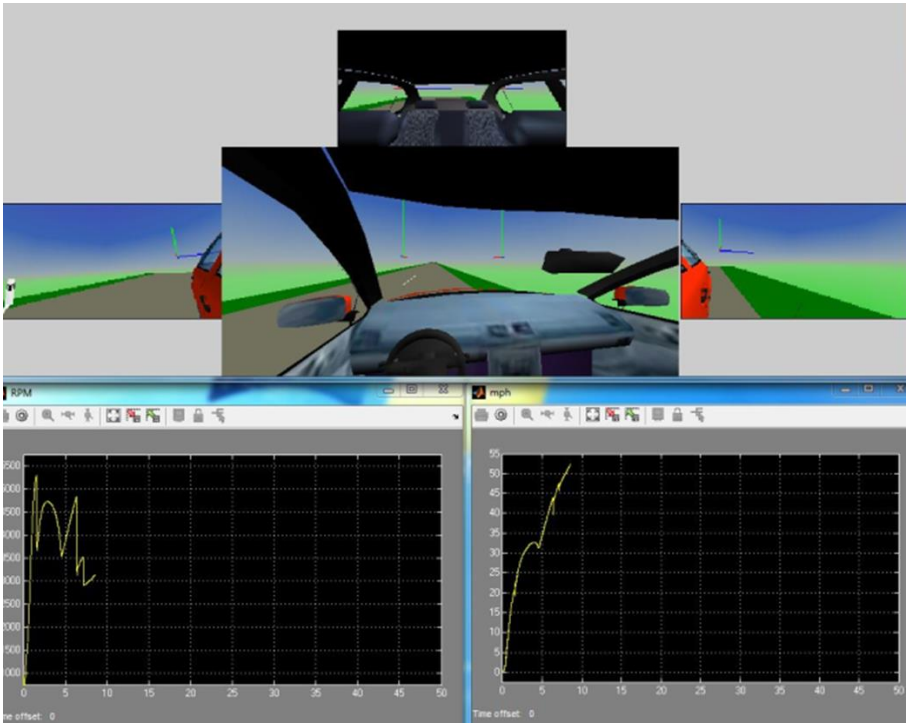

Figure B-6 Simulate mirrors

APPENDIX C MATLAB GUI

A Matlab GUI was created to visualize the movements of vehicles clearly, as shown in Figure C-1. It is similar to a simulation player and contains buttons such as run, step forward, step backward and stop. Velocities of vehicles are also shown in real time.

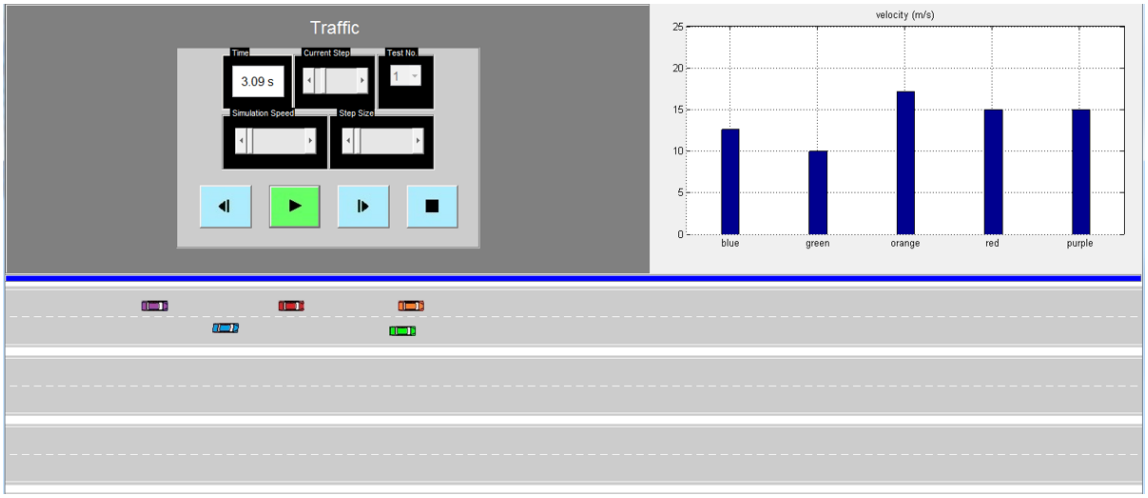

Figure C-1 Traffic simulation

Vehicle models described in section 3 are tested in Simulink. A few representative tests results are shown below. The red vehicle is the host vehicle. It wants to changes lanes because the blue car is moving slowly ahead. There is an orange car following in the adjacent lane. The red car needs to find the optimal timing for changing lanes. In the first test, the aggressiveness of the red car is positive (i.e. aggressive) and the aggressiveness of the orange car is negative (i.e. timid). Simulation results are shown in Figure C-2. The red car accelerates and starts to change lanes. The orange car finds that and decelerates. Finally, the red car change lanes successfully and safely.

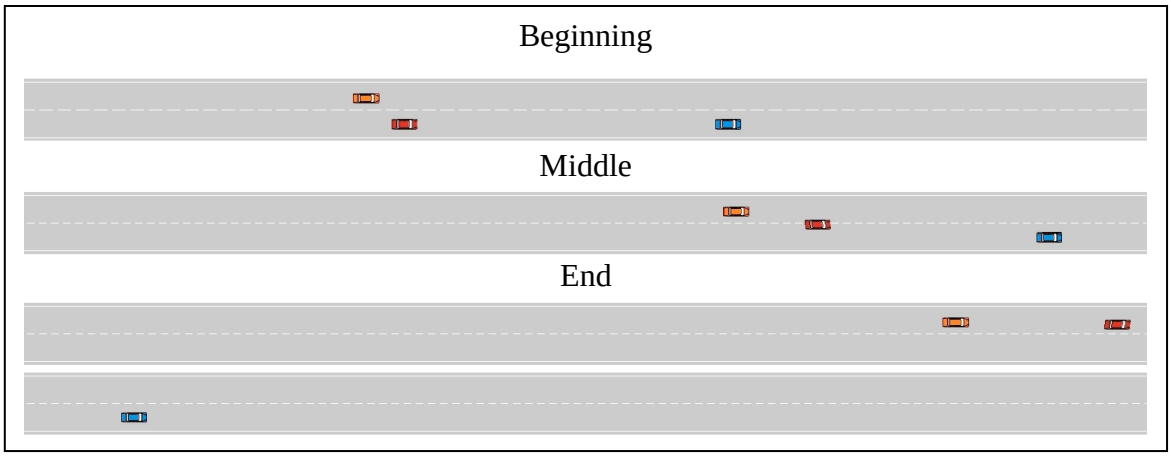

Figure C-2 Aggressive vs. timid

In the second test, the aggressiveness of the red car is negative (i.e. timid) and the aggressiveness of the orange car is positive (i.e. aggressive). Simulation results are shown in Figure C-3. The red car thinks it is dangerous to compete with the orange car and move in front of it. Therefore, it slows down and lets the orange car go first. Finally, the red car change lanes safely and follows the orange car in the new lane.

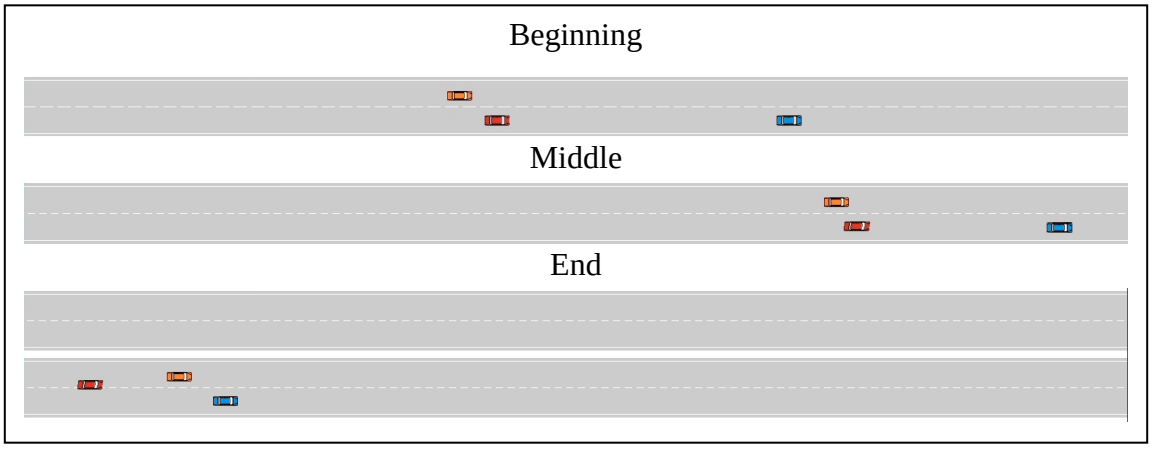

Figure C-3 Timid vs. aggressive

In the third test, the aggressiveness of the red car is smaller than that of the orange car. It means the orange car is more aggressive than the red car. Simulation results are shown in Figure C-4. At the beginning, the red car accelerates and tries to compete with the orange car. The orange car reacts by accelerating. After a period of time, the red vehicle gives in to the orange car and lets it go first. Therefore, it slows down and lets the orange car go. Finally, the red car change lanes and follows the orange car in the new lane.

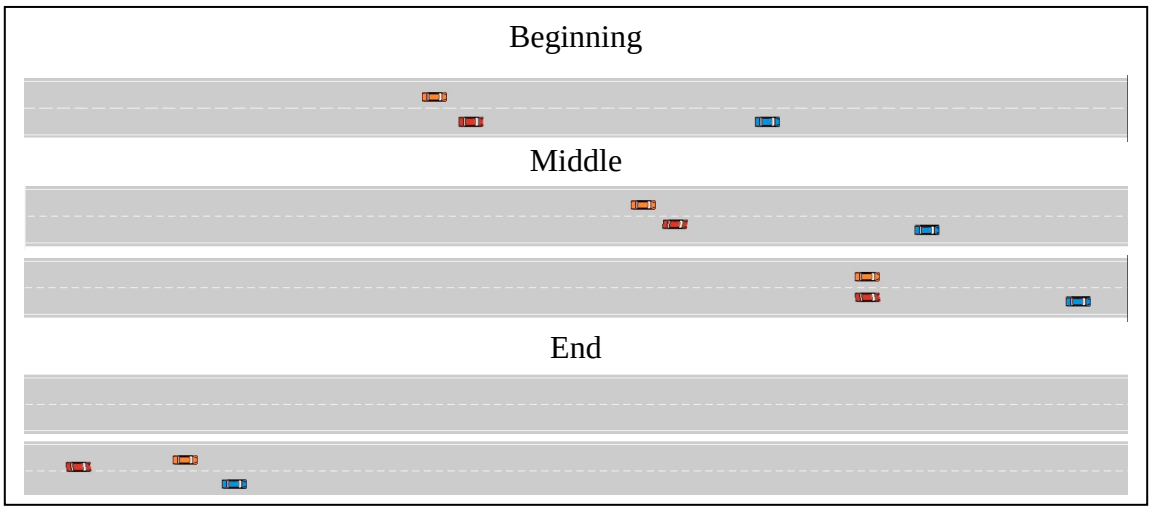

Figure C-4 Aggressive vs. aggressive

APPENDIX D ROAD DESIGN

A critical factor in the driving environment is the road type. Drivers may change their driving patterns under different road conditions. Various kinds of roads were designed and used in tests in the dSPACE driving simulator, as shown below. The first part is a top view. It can be seen that both straight and curved roads were created.

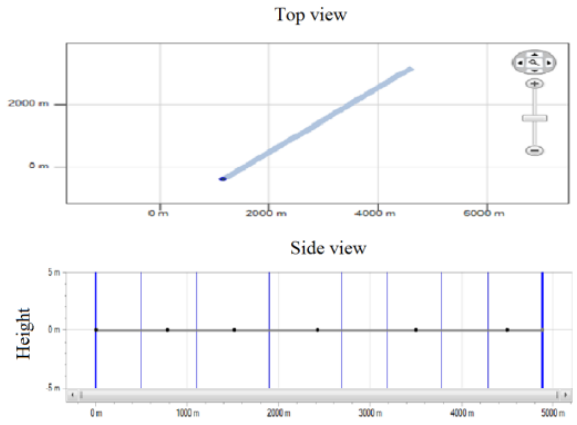

Figure D-1 Road one

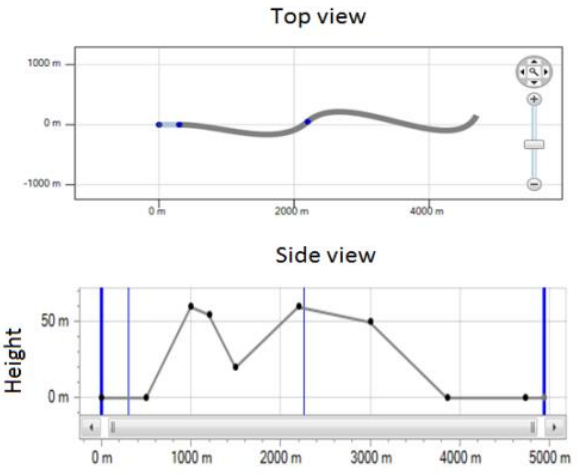

Figure D-2 Road two

There are two lanes, a passing lane and an inner lane. Since gravity affects vehicles' acceleration, the slope of the road is especially important in this work. Therefore, the road was designed to contain both steep and flat parts. In the steep part, there can be considerable changes of the acceleration even without drivers' inputs. It should be noticed that height curves of roads are smoothed (i.e. there is no sharp corner) in the 3D environment of the dSPACE simulator.

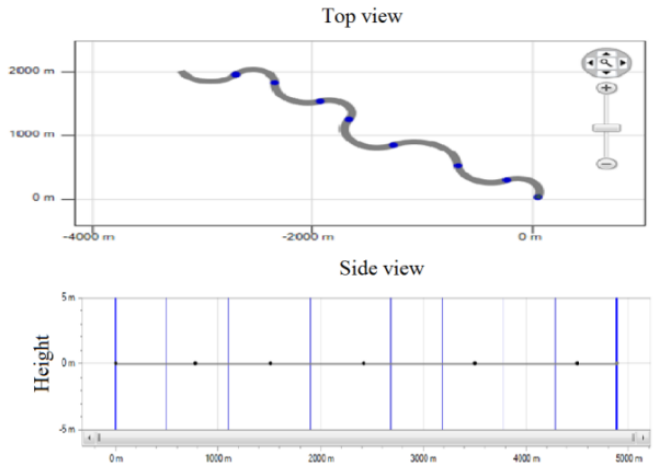

Figure D-3 Road three

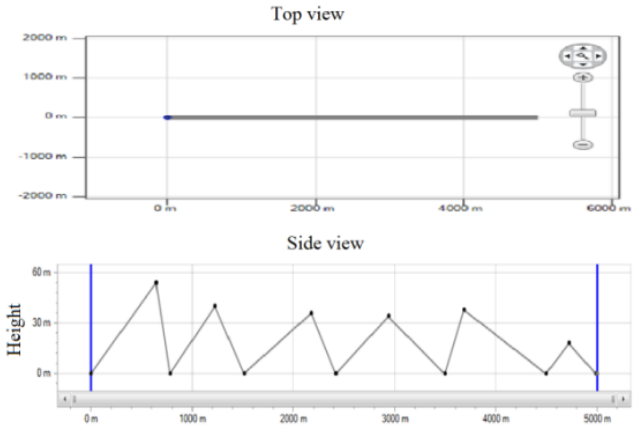

Figure D-4 Road four

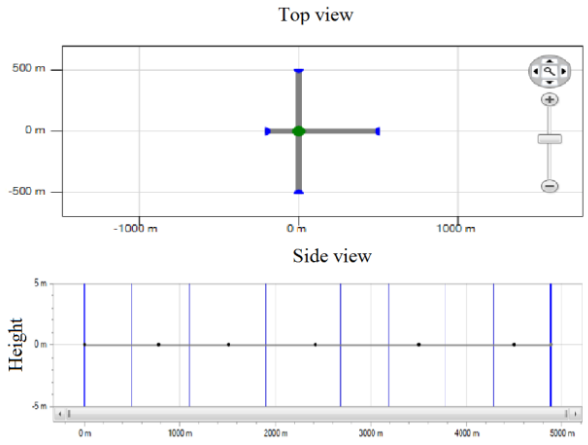

Figure D-5 Road five