Modeling the Impact of Connected and Automated Vehicles on Highway Operations

Mofan Zhou

BEng

School of Engineering
Griffith Sciences
Griffith University

Submitted in fulfillment of the requirements of the degree of Doctor of Philosophy

(March 2018)
Abstract

The purposes of this research are 1) studying the driving behavior of human drivers and Connected and Automated Vehicles (CAVs); 2) developing appropriate car-following models for CAVs to form a cooperative strategy in order to enhance traffic stability, reduce traffic oscillation and improve safety; 3) generating CAV’s driving model through a learning based method; and 4) using learning based method to develop a cooperative driving strategy in signalized intersections. Therefore, in the first part of this research, we show two demonstrations about the model development through an approach that modifying existing car-following models. The proposed methods are applied at highway sections with on-ramp and priority junction. By comparing with human drivers, the result shows that with a proper controlling mechanism, an increasing percentage of autonomous vehicles will reduce the total travel time and smooth traffic oscillations.

Developing driving models for Connected and Automated Vehicles through modifying a classical car-following model seems acceptable. However, those models are affected and constrained by empirical equations. The classical models, used to be applied for simulating human driving behaviors, may not be an ideal model for the Connected and Automated Vehicles due to the difference between machine and human.

Fortunately, technology innovations, most notably, machine learning techniques offer another modeling approach. In the second part of this research, we develop car-following controllers for Connected and Automated Vehicles based on reinforcement learning to dampen or eliminate traffic oscillations (or stop and go driving behaviors) caused by human drivers. By taking advantage of reinforcement learning, the controller has the capability of self-learning and self-correction.
Compared to traditional modeling approaches, it significantly reduces the modeling constraints. Two case studies are established to evaluate the model's performance. Our results demonstrate that the generated model from reinforcement learning is able to improve travel efficiency as well as reduce the negative impact of traffic oscillations.
Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Mofan Zhou
List of Publications

Included in this thesis are papers in Chapters 3, 4, 5 and 6 which are co-authored with other researchers. The bibliographic details for these papers including all authors are:

Chapter 3:


Chapter 4:


Chapter 5:


Chapter 6:

Appendix 1:


_________________________________ (Date)______________

Mofan Zhou

_________________________________ (Date)______________

Supervisor: Prof. Xiaobo Qu
Table of Contents

Abstract ......................................................................................................................... ii

Statement of Originality ............................................................................................... iv

List of Publications ........................................................................................................ v

Table of Contents ........................................................................................................... vii

List of Figures ................................................................................................................ xi

List of Tables .................................................................................................................. xv

List of Abbreviations ..................................................................................................... xvi

Chapter 1 Introduction .................................................................................................. 1

1.1 Background .............................................................................................................. 1

1.2 The objective .......................................................................................................... 5

1.3 Thesis layout .......................................................................................................... 7

Chapter 2 Literature Review ......................................................................................... 10

2.1 Traffic oscillation .................................................................................................... 11

2.1.1 Formation and propagation of traffic oscillation ............................................ 11

2.1.2 Attempts to mitigate traffic oscillation .......................................................... 12

2.1.3 CAVs in traffic oscillation .............................................................................. 13

2.2 Classical car-following models ............................................................................. 15

2.3 Reinforcement learning ........................................................................................ 16

2.3.1 Deterministic policy gradient ....................................................................... 19

2.3.2 Deep deterministic policy gradient ............................................................... 21

2.3.3 Multi-agent reinforcement learning ............................................................... 22
Chapter 3 A Modified Intelligent Driver Model for Optimizing Highway On-Ramps Merging

3.1 Introduction ........................................................................................................ 24

3.2 Microscopic traffic models ........................................................................... 26
   3.2.1 Modeling human driving behavior ......................................................... 26
   3.2.2 Modeling autonomous vehicular traffic .............................................. 28

3.3 An illustrative case study of Pacific motorway ............................................ 33
   3.3.1 Comparison of trajectories ................................................................. 33
   3.3.2 Simulation Setup .................................................................................. 34
   3.3.3 Trajectory ............................................................................................... 37
   3.3.4 Driving Sensitivity ............................................................................... 39
   3.3.5 Safety Analysis ...................................................................................... 39
   3.3.6 Travel Time ........................................................................................... 41
   3.3.7 Space Mean Speed ............................................................................... 42
   3.3.8 Sensitivity Analysis of Safe Time Gap ............................................... 43

3.4 Conclusions ..................................................................................................... 45

Chapter 4 A Modified Intelligent Driver Model for Optimizing Highway T-Junctions Merging ......................................................................................... 47

4.1 Introduction ..................................................................................................... 47

4.2 Experimental design ...................................................................................... 48
   4.2.1 Cooperative car-following strategy ..................................................... 49
   4.2.2 Scenarios ................................................................................................. 55

4.3 Results and discussion ............................................................................... 57
   4.3.1 Trajectory ............................................................................................... 57
   4.3.2 Global comparison ................................................................................ 59
   4.3.3 Local comparison .................................................................................. 60
4.3.4 Hard and soft safe time-gap replacement ........................................ 65
4.3.5 Location of the buffer zone .......................................................... 66
4.4 Conclusions .................................................................................. 69

Chapter 5 Dampen Traffic Oscillations: A Reinforcement Learning Based Approach 71

5.1 Introduction ................................................................................ 71
5.2 Experimental design ..................................................................... 75
  5.2.1 Training environment and parameter settings ................................ 75
  5.2.2 Reward function ...................................................................... 78
5.3 Results and discussion ................................................................ 80
  5.3.1 Training result ........................................................................ 80
  5.3.2 Comparing with manually-driven vehicles (MVs) .................... 82
  5.3.3 Mixed traffic flow .................................................................. 90
5.4 Conclusions ................................................................................ 92

Chapter 6 A Reinforcement Learning Based Approach to An Efficient Driving Strategy at Signalized Intersections ............................................. 94

6.1 Introduction ................................................................................ 94
6.2 Training environment and parameter settings ................................ 96
  6.2.1 Input features ........................................................................ 99
  6.2.2 Reward function .................................................................. 104
6.3 Results and discussion ................................................................ 106
  6.3.1 Training result ................................................................ ...... 106
  6.3.2 Different traffic demands ....................................................... 107
  6.3.3 Different light cycles ............................................................... 112
  6.3.4 Comparison with manually driven vehicles ......................... 115
6.4 Conclusions ................................................................................. 118

Chapter 7 Conclusion and recommendations ........................................ 120
Abstract

1 Introduction

2 Literature review

2.1 Oscillation

2.2 Neural networks and car-following models

3 Data description and pre-processing

3.1 Data description

3.2 Pre-processing

4 Neural network based model

4.1 Architecture

4.2 Results, applicability and critiques of existing neural network models

4.3 A gap-based neural network model

5 Recurrent neural network based model

5.1 Recurrent Neural Networks (RNNs)

5.2 Results and performance comparison

6 Conclusions

References
List of Figures

Fig. 2.1. Trajectory demonstration of traffic oscillation caused by traffic signal

Fig. 3.1. Illustration of AVs’ detection range

Fig. 3.2. Trajectories comparison between real data and IDM

Fig. 3.3. Scenario of merging simulation

Fig. 3.4. Trajectories of 0%, 5%, 15% and 25% of AVs

Fig. 3.5. Speed dispersion on a congested freeway segment (1.5 to 2.8 km), from 9th to 13th min, under different proportions of CIDM-based AVs

Fig. 3.6. Average travel time on the congested freeway segment (1 to 2.8 km)

Fig. 3.7. Space mean speed on the congested road segment (1 to 2.8 km)

Fig. 4.1. Illustration of the connected system

Fig. 4.2. Illustration of the cooperation strategy

Fig. 4.3. An illustration for a static \( T_i^c \)

Fig. 4.4. Trajectory result

Fig. 4.5. Average speed and queue length at each time step

Fig. 4.6. Example of the filtered trajectories

Fig. 4.7. Vehicular comparison of travel time, safety, and emission

Fig. 4.8. Vehicular comparisons of average values for travel time, iTTC, emission and queue length

Fig. 4.9. Hard and soft replacement for safe time gap

Fig. 4.10. Trajectory result of different buffer zone locations

Fig. 5.1. Learning diagram
Fig. 5.2. Vehicles follow each other in a circular driving loop. Note: the numbers next to vehicles refer to as “speed/reward” and these numbers are colored by the reward. A higher reward turns to blue and a lower one turns to red. ................................................................. 77

Fig. 5.3. Illustration of distance gap, time gap, and headway between successive vehicles ................................................................. 79

Fig. 5.4. The design of reward function .................................................. 80

Fig. 5.5. Moving averaged episode reward ............................................ 81

Fig. 5.6. Traffic oscillation when traveling under a high-speed setup. Note: The color in (a) and (b) demonstrates vehicle’s speed, red and blue means low and high speed respectively. ................................................................. 83

Fig. 5.7. Traffic oscillation when traveling under a low-speed setup. Note: The color in (a) and (b) demonstrates vehicle’s speed, red and blue means low and high speed respectively. ................................................................. 85

Fig. 5.8. Traffic oscillation when including a stop phase. Note: The color in (a) and (b) demonstrates vehicle’s speed, red and blue means low and high speed respectively ................................................................. 87

Fig. 5.9. Buffer time and distance for CAVs and MVs. Note: The color in (a) and (b) demonstrates vehicle’s speed, red and blue means low and high speed respectively ................................................................. 88

Fig. 5.10. Efficiency comparison with a baseline of MV ......................... 89

Fig. 5.11. Trajectory comparison under different CAV penetration rates. Note: The color in (a) and (b) demonstrates vehicle’s speed, red and blue means low and high speed respectively ................................................................. 90

Fig. 5.12. Comparison of travel efficiency under different CAV penetration rates .................................................................................. 91

Fig. 6.1. Learning diagram .................................................................... 96
Fig. 6.2. Circular training environment with CAVs generated from the center origin and removed at the end of the white line. The green square can turn red and green to represent the traffic light ......................................................... 97

Fig. 6.3. Illustration of the intersection scenario .................................................... 100

Fig. 6.4. Design of the feature representation for light cycle and its derivative term ......................................................................................................................... 102

Fig. 6.5. Reward function (dash line indicates an aggregated reward while solid line represents an absolute reward) ......................................................................... 104

Fig. 6.6. Moving averaged episode reward ................................................................. 107

Fig. 6.7. CAVs trajectory result for different traffic demands ....................................... 109

Fig. 6.8. CAVs average speed comparison for different traffic demands .............. 110

Fig. 6.9. CAVs trajectory and speed result in a dynamic traffic demand .............. 111

Fig. 6.10. CAVs trajectory result for different light cycles ........................................ 113

Fig. 6.11. CAVs average speed comparison for different light cycles (G: green; Y: yellow; R: red) ........................................................................................................ 114

Fig. 6.12. Emission and safety comparison between MVs and CAVs .............. 117

Fig. 6.13. Average speed, speed standard deviation and headway comparison between MVs and CAVs ........................................................................ 118

Fig. 1. Illustration of vehicle trajectories ................................................................. 137

Fig. 2. Comparison of filtered and unfiltered data for acceleration rate and velocity .......................................................................................................................... 138

Fig. 3. General neural network architecture with time-series inputs ............. 141

Fig. 4. Cross-Validation of NNa and NNv ................................................................. 144

Fig. 5. Comparison of predicted and real values .................................................. 147

Fig. 6. Comparison of predicted and real values .................................................. 149
Fig. 7. Comparison of the trajectories generated by neural network models and real data.................................................................150

Fig. 8. A typical car-following trajectory........................................153

Fig. 9. Cross-validation of NNpg model ....................................155

Fig. 10. Trajectory comparison between the real trajectories and the trajectories predicted by NNpg(1.0)..................................................156

Fig. 11. A typical RNN architecture, folded RNN (left) and unfolded RNN (right)...............................................................158

Fig. 12. Cross-validation for selecting the number of hidden units in RNNa...161

Fig. 13. Trajectories predicted by IDM and OVM .........................163

Fig. 14. Trajectories simulated with two different approaches ..........164

Fig. 15. Histogram of trajectories MSE for the IDM and RNNa models.....165

Fig. 16. Predict aggressive drivers in a traffic oscillation...............166

Fig. 17. Predict timid drivers in a traffic oscillation.........................167

Fig. 18. Predict normal drivers in a traffic oscillation .....................168

Fig. 19. Prediction in non-oscillation cases.................................170
List of Tables

Table 3.1. $\lambda_a, \lambda_s, \lambda_r$ and $\lambda_m$ values for the CIDM during merging................. 32

Table 3.2. Model parameters of the FVDM .................................................. 36

Table 3.3. Model parameters of the CIDM .................................................. 37

Table 3.4. Average Value of Speed Dispersion .............................................. 44

Table 3.5. Average Travel Time (Minute) .................................................. 44

Table 4.1. Default parameter values in IDM for simulating MVs (Treiber et al., 2000) ............................................ 51

Table 4.2. Default values for merging (Tian et al., 2000) ............................. 52

Table 4.3. Default parameter values in the modified IDM for CAVs ............... 54

Table 4.4. Experimental setup ................................................................. 56

Table 4.5. Buffer zone location for three cases. ........................................ 66

Table 5.1. Travel efficiency comparison between MVs and CAVs ................. 86

Table 5.2. Average results over all vehicles in traffic oscillations. ................. 91

Table 6.1. Features selected for the RL model .......................................... 100

Table 1. Neural network settings ............................................................ 143

Table 2. NNpg settings ............................................................................. 153

Table 3. Calibrated parameters of the IDM and OVM ................................ 162
List of Abbreviations

\( \alpha \) Scale factor of the buffer time

\( \alpha_\theta \) Learning rate for critic

\( \alpha_t \) Scale factor to normalize the derivative of light cycle

\( \alpha_w \) Learning rate for actor

\( \delta_t \) TD-error

\( \gamma \) Discount fact

\( \lambda \) Sensitivity factor

\( \lambda_a \) Scale factor for maximum acceleration rate

\( \lambda_b \) Scale factor for desired deceleration rate

\( \lambda_T \) Scale factor for safe time gap

\( \lambda_{\Delta x} \) Scale factor for merging impact

\( \mu \) Deterministic policy

\( \mu^\theta \) Target policy parameterized by \( \theta \)

\( \mu^{\tilde{\theta}} \) Target policy network parameterized by \( \tilde{\theta} \)

\( \pi \) Policy

\( \theta^\theta \) Parameters for Q function

\( \rho^\theta \) Behavior policy

\( \tau_c \) Critical gap

\( \tau_f \) Follow up time

\( \tau_\theta \) Soft-replacement ratio for policy network
$\tau_w$ Soft-replacement ratio for value network

$a$ Maximum acceleration rate

$a_{\text{brake}}$ Acceleration rate in a braking term

$a_{\text{free}}$ Acceleration rate for free road

$a_n$ Acceleration rate of the $n$th vehicle

$a_t$ Action at time $t$

$b$ Desired deceleration rate

$l_d$ Distance to light

$l_i$ Instant travel time to light

$l_B$ Length of the buffer zone

$i\text{TTC}$ Inverse time to collision

$n_m$ Instant queue length

$n_{mt}$ Threshold number of queuing vehicles

$p_n$ Position for vehicle $n$

$r_t$ Reward at time $t$

$s_o$ Jam distance

$s_t$ State at time $t$

$t_g$ Light duration of green phase

$t_r$ Light duration of red phase

$t_y$ Light duration of yellow phase

$v_{\text{EMA}}$ Exponential moving average speed
\( V_n \) Velocity of the \( n \)th vehicle

\( x_{EMA} \) Exponential moving average distance

\( Q^\mu \) State-action value based on a policy \( \mu \)

\( Q^\pi \) State-action value based on a policy \( \pi \)

\( Q^\psi \) Estimated state-action value by a function approximator parameterized by \( w \)

\( \tilde{Q}^\psi \) Target value network parameterized by \( \tilde{w} \)

\( R_d \) Detection range of the main-road automated vehicle

\( R_t \) Episodic reward

\( T \) Safe time gap

TTC Time to collision

\( T_l \) Light cycle

\( T'_l \) Derivative of light cycle

\( T^b_l \) Buffer time calculated from the length of the buffer zone \( l_b \) and the current velocity of this CAV

\( T^c \) Cooperative safe time gap

\( T^c_t \) Dynamic cooperative safe time gap at time \( t \)

\( T^{CAV} \) Default time gap for CAV

\( T_{max} \) Maximum time gap for CAV

\( V(\Delta v) \) Empirical optimal velocity function

\( \Delta v \) Difference of the leader's velocity relative to the follower
$\Delta x$  
Gap distance between two successive vehicles

$\Delta x_0$  
Remaining distance from the merging point

$\Theta(x)$  
Heaviside Function
Chapter 1
Introduction

1.1 Background

Urbanization is rapidly taking place globally. This tendency results in the increase in population and the total number of vehicles in cities. According to (United Nations, 2014), the urbanized population will contribute 66% in 2050. The rapid urbanization unavoidably causes severe transport and mobility challenges, especially in large cities like London, Beijing, Sydney and Shanghai. Transportation system designed ages ago becomes overloaded, which causes traffic congestion and turns to a problem that affects every traveler. There is no doubt that the transport challenges (safety, congestion, sustainability, etc.) significantly undermine a large city’s livability and the wellbeing of its residents:

- Traffic crashes result in 1.25 million fatalities and 50 million injuries worldwide every year and, more importantly, traffic crash is the leading cause of death for < 45 years old (World Health Organization, 2017);
- Gridlocks occur more and more frequently in our urban cities, especially during peak hours; and
- Transport sector contributes over 1/3 of the greenhouse gas (GHG) emissions (U.S. Environmental Protection Agency, 2017).
Researchers are putting their efforts in searching methods to increase current transportation capacity as well as to reduce the negative impact of the increasing tension in nowadays transportation system.

Firstly, the causes and factors related to this issue have been discovered and identified during recent decades. It has been well recognized that one of the influential factors is human driver’s limits (e.g. slow reaction time, limited information processing capability), heterogeneity (e.g. different reactions among drivers), and selfishness/non-cooperativeness substantially compromise the performance of our urban transport systems (Qu et al., 2017). On the other hand, the limitation in road infrastructure may form a capacity bottleneck and restrict the traffic flow. Therefore, bottleneck treatments have been analyzed and developed in such as railroad-highway crossings (Easa et al., 2017), roundabouts (Bie et al., 2016, Ren et al., 2016), shared bicycle cycleways (Jin et al., 2015), and rail lines (Wang and Qu, 2017). Therefore, it implies that the traffic capacity improvement in the future can be a vehicular-based and road infrastructure-based.

Most existing traffic control strategies and road design standards aim to regulate or control a collective and aggregated group of vehicles (e.g. traffic signal), with an attempt to accommodate the above-mentioned human driver’s deficiencies (Papageorgiou and Kotsialos, 2002, Papageorgiou et al., 2003, Qu et al., 2015, Qu and Wang, 2015, Zhao et al., 2012). By aggregating vehicles into a group, we pack the individual vehicle to an organized term, so that reducing the number of impact elements in this traffic network. Grouping and packing make use of the spatial advance as same as the packaging in commercial industry to increase the delivery efficiency.
Ramp metering, which utilizes the concept of group merging, is one of the techniques that has been applied to reasonably redistribute traffic flow in order to relieve overloaded traffic since the 1960s (May, 1965). Further, concepts such as Adaptive Cruise Control (ACC) and Cooperative Adaptive Cruise Control (CACC) are proposed and investigated recent years (Wang et al., 2014a, Milanes and Shladover, 2014, Naus et al., 2010, Zhou et al., 2017b). Cooperation in transportation system benefits the overall safety by preventing collision (Liu et al., 2018). Inter-Vehicle Communication (IVC) along with cooperation schemes have been paid due attention in consideration of the importance of using shared information for traffic optimization (Yang and Jin, 2014, Ma et al., 2017).

With the Vehicle to Vehicle communication (V2V) (Wang, 2007) and IVC, it is not sufficient in a long-term practice with only the vehicular level communication methods. To utilize the full potential of our urban transport infrastructure, we are witnessing a series of vehicle technology innovations in recent years, most notably, connected vehicles, and automated or self-driving vehicles.

Connected vehicles basically enable real-time information sharing and communications among not only individual vehicles but also infrastructure control units (Zhou et al., 2017a). It is believed that Connected and Automated Vehicle (CAV) can play a key role in traffic optimization by utilizing communication technologies. Ideas such as aforementioned V2V and CACC have been proposed to be a foundation of CAVs. Autonomous driving, which takes control from human and drives freely, is one of the characteristics of CAVs that requires a feasible control mechanism to achieve.

When turning into the CAV behavioral design, there are many approaches to achieving a suitable design for CAVs. A typical method is to redevelop or modify an
existing car-following model in order to be better applicable for CAVs. Our first two studies make use of this logic by modifying one of the most popular car-following models to gain a performance boost in reducing traffic oscillation on highway segments adjacent to on-ramp and priority junction.

Although these modifications of existing car-following models for CAVs do improve the efficiency of the transportation system, the classical car-following models themselves may be constrained by their empirical design and the close connection of human behaviors. In other words, those car-following models are developed to be best fitted for reflecting human drivers, which may be not desired to be adopted for the design of CAV’s behaviors due to which may have a totally altered control mechanism (Zhou and Qu, 2016).

The aforementioned concern let us consider other possible approaches. We notice that the recent development in Machine Learning (ML) has obtained enormous successes in many research fields, including image recognition (Girshick et al., 2014, Simonyan and Zisserman, 2014, Girshick, 2015, He et al., 2015a), generative works (Goodfellow et al., 2014, Radford et al., 2015), gaming (Mnih et al., 2015, Silver et al., 2016) and behavior shaping (Lillicrap et al., 2015, Mnih et al., 2016, Merel et al., 2017). Their successes are highly based on the breakthrough of both computer hardware and software. The increased hardware capacity such as GPUs allows computers to speed up large matrix calculation. Therefore, computer models such as Artificial Neural Networks (ANN), which has tens of thousands of parameters, can be optimized in a short amount of time. The data-driven, non-parameter and highly non-linearity properties make machine learning, especially ANN, being capable of solving hard real-life problems.
The Reinforcement Learning (RL) (Sutton and Barto, 1998), one of the developing branches in ML, provides the computer with a capability to learn behaviors from its interaction with the environment from scratch. The combination of ANN and RL enhances the performance of pure RL (Levine and Abbeel, 2014, Silver et al., 2014, Mnih et al., 2015), which also can be recognized as the desired method of developing CAV’s behavioral model.

1.2 The objective

We are witnessing how the traffic congestion can affect our daily life. The methods to mitigate or relieve its negative impacts are on demand. Autonomous vehicles to be seen as a desired tool, not only to free up drivers, but also to improve the current transportation system. Until now, there is no unique model for controlling CAVs under all driving scenarios. On the one hand, the driving behaviors of CAVs are remaining unknown. On the other hand, this is an opportunity for exploring and arguing the best behavior a CAV could have.

Developers and designers of CAVs are attempting to apply conventional car-following models on them. However, due to the fundamental difference between CAVs and human drivers, the car-following models designed to simulate human driving behaviors are argued to be not capable of addressing CAVs behaviors and underestimate CAVs capability. Therefore, there is a great room existing in improving CAVs’ driving behaviors. In here, we attempt to bridge the gap found in aforementioned designing procedure towards a case-specific behavioral design by two approaches including 1) modifying classical car-following models; 2) generate car-following models for CAVs. In the second approach, we explored the possible
ways in model developing to fulfill the CAV’s unique property through
reinforcement learning processes.

The objectives of this study are listed as follows.

• Modify a classical car-following model to be suitable for the application of
  maximizing the travel efficiency and safety on highway segments adjacent to
  an on-ramp;
• Modify a classical car-following model to improve the merging capacity on
  highway segments with a priority junction. Meanwhile, this model should
  also be able to minimize the traffic oscillation on these highway segments;
• Point out the important aspects of designing a modified car-following
  model for CAVs, such as the location of activation zone and the degree of
  cooperation;
• Adopt a reinforcement learning approach to generate CAV’s control model
  to maximize travel efficiency while minimizing risk level. This learned
  model should also be able to mitigate traffic oscillation triggered by a sudden
  change occurred in the leading vehicle;
• Evaluate the performance of the reinforcement learning based model in
  mixed traffic flow (CAVs with MVs); and
• Construct a reinforcement-learning framework to generate an appropriate
  driving behavior for CAVs to reduce the negative impact caused by the
  traffic oscillation triggered by the traffic signal in signalized intersections.

In addition to above objectives, we are also interested in utilizing artificial neural
networks, which can contain more than millions of hidden parameters, to train a
more accurate car-following model to reflect differences in human drivers. The
property of high non-linearity in ANN allows it to simulate many complicated
real-life phenomena. Therefore, we also adopt artificial neural networks as a
reinforced car-following model to simulate human driving behaviors, to replace the existing empirical car-following models. We will analyze the modeling process and establish a recurrent neural network model specially designed for the car-following scenario. Our goal in this study is to develop a single recurrent neural network model to simulate different driver characteristics, while the empirical car-following models are not capable of distinguishing that.

Due to our main purpose of evaluating CAVs, we leave this neural network study in Appendix 1 for a record of how to merge the neural network in car-following theory in order to replace empirical car-following models.

1.3 Thesis layout

This thesis is divided into seven chapters. Apart from the Introduction and Literature review chapters, we conducted case studies aiming to propose a specific car-following model for the CAV in each scenario. These scenarios are 1) on-ramp merging; 2) priority junction merging, 3) responding to sudden change in the leading vehicle; and 4) signalized intersection optimization. The detail description of each chapter is shown as follows.

Chapter 1, the Introduction, includes the background and the objectives of this thesis. In the background, it briefly introduces current transportation system and the issues found in this system. Later, we present our objectives and the method we adopted to achieve the objectives.

The literature review is formed by four aspects in Chapter 2. Firstly, we review the stop-and-go, or traffic oscillation phenomenon, describe its formation and
propagation properties. Secondly, some of the available classical car-following models are summarized. In the last two parts, we describe a typical reinforcement-learning framework, and the reinforcement learning method for a continuous action space we used in this thesis.

Chapter 3 presents our modified car-following model specified for optimizing on-ramp merging. Model developing procedure has been shown. An illustrative case study is conducted to evaluate the model capacity.

Another modification of a classical car-following model is presented in Chapter 4. This model modification is developed for mitigating traffic oscillation in priority junction scenario. Simulations are conducted to evaluate the model performance.

Chapter 5 reveals an innovative modeling process for developing CAVs car-following behavioral model. The reinforcement-learning framework is introduced in this study to learn an appropriate driving behavior in order to smooth traffic oscillation caused by the leading vehicle’s sudden changes. Environmental setup for training is demonstrated and the impressive results are shown in this chapter.

Chapter 6 is an extension of the study in Chapter 5, to develop a CAV model for a scenario with signalized intersections through the reinforcement learning approach. A training environment is established in this chapter together with a more complex system design. Evaluations under different scenarios are conducted to validate the learned model.

Chapter 7 summarizes all studies in this thesis. Present the limitations of this research and recommend possible solutions and suggestions for future studies.
Appendix 1 views car-following model from a different perspective. It shows the first attempt to use recurrent neural networks instead of a classical car-following model to simulate different driver characteristics.
Chapter 2

Literature Review

Transportation researchers and practitioners are paying more attention to the negative impact caused by the increasing number of vehicles on our road networks. The major issue has been identified as the limited road capacity being no longer fulfilled the increasing number of vehicles. Thus, problems such as stop-and-go traffic become more serious. Connected and automated vehicles are believed to be an ideal solution to mitigate the aforementioned issue. We will review some of the current methods developed for CAVs and their limitations. Moreover, in order to merge the state-of-art machine learning techniques in this study to our modeling procedure, we will review one of the machine learning methods called reinforcement learning.

In this chapter, we will introduce the details about stop-and-go traffic as known as traffic oscillation. Particularly, the formation, propagation and impact of traffic oscillation will be described in the next section. Further, in order to simulate traffic oscillation, we will list some classical car-following models available at this stage. Then the limitations of those models in simulating CAVs driving behaviors will be presented. Method of reinforcement learning then is introduced to the modeling process to address those limitations.
2.1 Traffic oscillation

2.1.1 Formation and propagation of traffic oscillation

Due to the heterogeneity of drivers’ responses, traffic oscillations are believed to be unavoidable under heavy traffic flow conditions, especially when bottleneck, e.g., on-ramp, lane closure, is involved (Qu et al., 2017). Indeed, human being’s perception-reaction time causes a delay in responding, and this delay can quickly form traffic oscillations in a dynamic traffic condition.

With the increased traffic demands caused by urbanization, traffic oscillations become more and more frequent, which subsequently imposes negative impacts on transport safety, efficiency and sustainability (Chen et al., 2012). The formation and propagation mechanisms of traffic oscillations have been intensively investigated during recent years. For example, Li et al. (2014) pointed out that drivers are often forced to be engaged in repeated deceleration-acceleration cycles in a congested highway segment. The trigger of this phenomenon includes ramp-merging, lane change and changes in roadway geometric features (Li et al., 2012). Signalized intersections due to alternating green and red light are another reason of oscillation forming (Ma et al., 2017, Zhou et al., 2017a). A congested pattern at a signalized intersection in Fig. 2.1 demonstrates that red light forces fast vehicles (blue) to slow down (red) and makes vehicles stop at the intersection. Oscillation is then formed and propagated to the upstream.
Fig. 2.1. Trajectory demonstration of traffic oscillation caused by traffic signal

2.1.2 Attempts to mitigate traffic oscillation

Ramp metering as a widely used method proposed since the early 1960s (May, 1965) aims to reduce the negative impact caused by heavy traffic demand. However, the experiment carried by Zhang and Levinson (2010) indicates that only a marginal improvement (2% increment for average flow rate and 3% increment for average queue discharge) is observed in a case of ramp metering. The possible inefficiency is a relay on the drivers on the highway. If they are not cooperating with the merging vehicles, ramp metering will still not effective in improving merging capacity.
Another approach to reducing the traffic oscillation is smoothing vehicle trajectories by Variable Speed Limits (VSL) that dynamically regulates traveling speed depending on real-time traffic flow information. VSL harmonizes overall speed in order to mitigate the negative effect caused by sudden brakes (Lu and Shladover, 2014). At signalized intersections, VSL is applied to alter a vehicle’s arrival time without fully stopped during the red phase (Sanchez et al., 2006, Wu et al., 2015). However, VSL’s performance in regulating traffic oscillation is constrained by the compliance of drivers and differences in vehicles’ acceleration profile (Fuhs, 2010).

2.1.3 CAVs in traffic oscillation

Technologies are improving; connected and automated vehicles (CAVs) will gradually enter the automobile market. Newly proposed technologies, such as Inter-Vehicle-Communication (IVC) (Wang, 2007, Schönhof et al., 2007, Schönhof et al., 2006, Yang and Recker, 2005) and Cooperative Adaptive Cruise Control (CACC) (Wang et al., 2014a, Milanés et al., 2014, Milanes and Shladover, 2014, Kachroo et al., 2014, Zhou et al., 2017b) are the foundation of future CAVs.

It is no doubt that the CAVs can take the advantages of a shared network and makes use of the public information to control individual vehicle’s trajectory precisely (Letter and Elefteriadou, 2017, Milanés and Shladover, 2016, Bagloee et al., 2016). Therefore, driving behaviors can be greatly shifted to a higher level that is significantly different to Manually driven Vehicles (MV). External conditions such as bottleneck location, overall traffic demand can be shared among CAVs. Furthermore, this shared information can possibly be formed as one part of the current driving strategy. Therefore, CAVs can be utilized in a way that is able to optimize the traveling condition and solve the aforementioned inefficient driving issues. To evaluate the performance of CAVs in this circumstance, simulations
conducted in van Arem et al. (2006), Zhou et al. (2017b), Letter and Elefteriadou (2017) show that CAVs can perform beyond human drivers when having a specific design corresponding to highway merging.

At a signalized intersection, by utilizing the shared information such as traffic light duration, CAVs trajectory can be pre-defined and optimized in a way to minimize fuel consumption, risk and increase efficiency (Zhou et al., 2017a, Ma et al., 2017). Thus, the traffic oscillation can be greatly mitigated in this case. Although their model can be implemented close to real-time, the optimized and planned trajectories have to change every time when surrounding driving environment changes. This could bring an overhead computational problem in real-time applications, as the model may need to do the same job repeatedly.

In this study, we aim to develop cooperative strategies for CAVs in a highway segment with bottlenecks, such as on-ramps and priority junctions, to achieve those maximization goals. We firstly develop the CAV’s car-following model by modifying a classical car-following model. The results indicate a great improvement in performance.

However, due to the behavioral differences existing between MVs and CAVs, a modified classical model may not able to overcome the modeling constraints. Typically, modeling CAVs to maximize travel efficiency in traffic oscillations is difficult due to many constraints and unknown parameters in classical car-following models. The state-of-art machine learning techniques make it possible to simplify this problem, and the reinforcement learning approach is more suitable for this problem.
In this study, we also attempt to address the inefficiency issue in aforementioned methods by adopting reinforcement-learning techniques. In reinforcement learning, it allows us to propose a self-learned behavioral model that can be applied in real-time and to optimize the trajectory without overhead issue even in a dynamic environment. The background and overview of reinforcement learning are presented in this review as well.

2.2 Classical car-following models

A car-following model is a mathematical expression with respect to how one car follows another. Different expressions have been established from the mid of twentieth century, e.g. the GHR model (Herman et al., 1959, Chandler et al., 1958), the CA model (Kometani and Sasaki, 1961) and Gipps’ model (Gipps, 1981).

In order to overcome the deficiencies of the above pioneering models, a few other car-following models are developed in order to better establish the relationship between vehicle motion and traffic conditions. The Optimal Velocity (OV) model (Bando et al., 1998, Mason and Woods, 1997, Bando et al., 1995) determines acceleration rates based on gap distance and velocity. The OV model extracts the distance information and converts it into a velocity representation, then compute the acceleration rate from the difference of velocity representation with real velocity. Models proposed by (Hasebe et al., 2004, Hasebe et al., 2003, Jiang et al., 2001) are modifications of the original OV model for a better performance.

Another widely used model is the Intelligent Driver Model (IDM) (Treiber et al., 2000). This model decomposes the acceleration into two aspects that consist of a free road acceleration and a brake deceleration. Other IDM improvements are such
as the Human Driver Model (HDM) (Treiber et al., 2007a, Treiber et al., 2006), the IDM with Constant-Acceleration Heuristic (CAH) (Kesting et al., 2010) and the IDM for cooperative adaptive cruise control (Milanes and Shladover, 2014).

Although these car-following models are initially developed to simulate human driving behaviors, these models also were applied to analyze CAV behaviors. Many studies (Zhou et al., 2017b, Yu and Shi, 2015, Milanes et al., 2014, Milanes and Shladover, 2014, Kachroo et al., 2014) built their CAV models by either improving or modifying an existing car-following model. However, most classical models have prescribed model structures and parameter settings that are independent of real-time and historical surrounding traffic and infrastructure conditions. In this regard, a mainstream of future CAV control algorithms is likely learning-based and adaptive to constant changing sensor feeds (Lefevre et al., 2015, Wei et al., 2014, Zhou et al., 2017c). Therefore, these models may not be flexible enough to describe adaptive CAV behaviors in real-world traffic.

2.3 Reinforcement learning

A widely accepted Reinforcement Learning (RL) framework contains an agent and an environment where the agent can perform an action and receive feedback from the environment. Particularly, at each time $t$, the agent receives an observation or state $s_t$, outputs a probability distribution over the actions $P(a_t)$ under a policy $\pi$, then receives a reward $r_t$ and a transition dynamics of the next state $s_{t+1}$ from the environment based on the current $s_t$ and $a_t$. 
The agent in RL attempts to learn a function to map the observation to its future reward. The goal in reinforcement learning is to learn a policy that maximizes the expected return from the current state. A discounting factor $\gamma \in [0, 1]$ is applied to compute the future return defined as the sum of discounted future rewards $R_t = \sum_{i=t}^{T} \gamma^{(i-t)} r(s_i, a_i)$. Many approaches to reinforcement learning use the Bellman equation (Equation (2.2)) to represent the recursive relationship in the future return.

$$Q^\pi(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim \mathbb{E}} \left[ r(s_t, a_t) + \gamma \mathbb{E}_{a_{t+1} \sim \pi} [Q^\pi(s_{t+1}, a_{t+1})] \right]$$

(2.2)

where $Q^\pi$ is the state-action value based on a stochastic policy $\pi$. Normally, a deterministic target policy $\mu: S \leftarrow A$ is applied instead of the stochastic policy $\pi$, so that the inner expectation in Equation (2.2) disappears and it can be rewritten as

$$Q^\mu(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim \mathbb{E}} \left[ r(s_t, a_t) + \gamma Q^\mu(s_{t+1}, \mu(s_{t+1})) \right].$$

(2.3)

With this deterministic policy $\mu$, the agent is possible to learn $Q^\mu$ off-policy, which makes use of the experience collected from other agents or from this agent but a different time. One of the famous off-policy algorithms is known as Q-learning.
(Watkins and Dayan, 1992) which computes the greedy policy 
\[ \mu(s) = \arg \max_a Q(s, a) \]. The \( Q(s, a) \) or Q function in this equation estimates the state-action value. Replacing the Q function with the state-of-art neural network, a non-linear function approximator, solves many challenging RL problems. Therefore, a policy \( \mu \) and the Q function can be parameterized by a neural network \( \theta^Q \) with a loss function described as below to apply gradient descent to optimize its parameters.

\[
L(\theta^Q) = E_{s_t, a_t, r_t} \left( \left( Q(s_t, a_t \mid \theta^Q) - y_t \right)^2 \right)
\]

where

\[
y_t = r(s_t, a_t) + \gamma Q\left(s_{t+1}, \mu(s_{t+1}) \mid \theta^Q \right).
\]

The \( y_t \) in Equation (2.4) and (2.5) is recognized as a Q-target in RL. It is found in the past that increasing the depth of a neural network can also increase its learning capacity but it introduces a non-convergence problem in RL. To address this issue, Mnih et al. (2015) proposed a variation of the Q-learning algorithm named DQN which learns to play video games from pixel inputs.

However, the Q-learning is an algorithm restricted by its discrete action output. In other words, it cannot output a continuous action such as to control acceleration rate or rotation angle. Fortunately, the Policy Gradient (PG) algorithm (Williams, 1992,
Williams, 1988) from another RL branch is capable of outputting an action in a continuous space.

Based on the concept of PG, Konda and Tsitsiklis (1999) proposed the Actor-Critic algorithm combining a policy function and value function to improve the stability of updating. By utilizing a function approximator such as a neural network to parameterize the Actor-Critic algorithm, it improves the performance dramatically in terms of speeding up training process and increasing the ability in non-linearity (Sutton et al., 1999).

2.3.1 Deterministic policy gradient

In PG, learning a stochastic policy may not be efficient and productive when the agent only needs a deterministic action to perform in a deterministic environment. Therefore, Silver et al. (2014) proposed a Deterministic Policy Gradient (DPG) method that leads to an efficiency boost in training. The key idea behind DPG is the same as Actor-Critic algorithm as it separates the training into following.

• Update value function (Critic) to have a better evaluation of current situation, which improves the understanding of surrounding environment and causal relationship; and
• Update policy function (Actor) directly from Critic to map the state to action in order to gain the better value.

These updates aim to maximize the future return an agent receives, so the objective function for this purpose is shown in Equation (2.6).

$$J_\beta(\mu^\theta) = \int_S \rho^\beta(s)Q^\theta(s, \mu^\theta(s))ds$$  \hspace{1cm} (2.6)
where $\mu^\theta$ is the target policy parameterized by $\theta$, $\rho^\theta(s)$ denotes the behavior policy at state $s$, $Q^\mu(s, \mu^\theta(s))$ represents the state-action value or Q-value evaluated from a critic and its action comes from the target policy. We can rewrite this objective function to obtain the parameter update in Equation (2.7).

$$
\nabla_\theta J_\rho(\mu^\theta) = \mathbb{E}_{s \sim \rho^\theta} \left[ \nabla_\theta \mu^\theta(s) \nabla_a Q^\mu(s, a) \right]_{a=\mu^\theta(s)} 
$$  \hspace{1cm} (2.7)

This equation gives the off-policy deterministic policy gradient. It indicates that the actor parameter updated by moving its parameter in the direction of the critic can maximize its Q-value. Finally, the actor and critic update in the DPG can be unified as following,

$$
\begin{align*}
\delta_t &= r_t + \gamma Q^w(s_{t+1}, \mu^\theta(s_{t+1})) - Q^w(s_t, a_t) \\
w_{t+1} &= w_t + \alpha_w \delta_t \nabla_a Q^w(s_t, a_t) \\
\theta_{t+1} &= \theta_t + \alpha_\theta \nabla_\theta \mu^\theta(s_t) \nabla_a Q^w(s_t, a_t) \right|_{a=\mu^\theta(s_t)} 
\end{align*}
$$  \hspace{1cm} (2.8)

where $\delta_t$ denotes the TD-error in the one-step update; $r_t$ represents the reward received at time $t$ when taking action $a_t$ and the state changes from $s_t$ to $s_{t+1}$; $Q^w$ is the estimated state-action value by a function approximator parameterized by $w$; $\alpha_w$ and $\alpha_\theta$ are the learning rates of actor and critic respectively; $\mu^\theta(s_t)$ denotes the target policy parameterized by $\theta$. 

20
2.3.2 Deep deterministic policy gradient

The convergence and update stability issue still exist in a complex RL problem even with the DPG algorithm, although it successfully improves the learning efficiency. The on-policy updating and nonlinear function approximator are found to be the cause. In other words, a correlation between two updates in successive states introduces unstable issue for challenging problems. The DQN mentioned above successively solves those issues, thus Lillicrap et al. (2015) brings the advanced concept in the DQN to DPG and rename it to Deep Deterministic Policy Gradient (DDPG).

The DDPG is a combination of DQN and DPG in terms of creating a memory buffer and a target network in addition to DPG aiming to de-correlate successive updates. Both of the actor and critic in DDPG keep evaluating networks ($\mu^\phi$ and $Q^\phi$) and target networks ($\mu^\psi$ and $Q^\psi$) throughout the training process. The parameters updated in a target network are delayed for the purpose of de-correlating successive updates. The formal updating rule is shown in following equations.

$$
\delta_t = r_t + \gamma Q^\psi(s_{t+1}, \mu^\psi(s_{t+1})) - Q^\phi(s_t, a_t)
$$

$$
w_{t+1} = w_t + \alpha_w \delta_t \nabla_{w} Q^\phi(s_t, a_t)
$$

$$
\theta_{t+1} = \theta_t + \alpha_\theta \nabla_{\theta} \mu^\phi(s_t) \nabla_{a} Q^\phi(s_t, a_t) \bigg|_{a=\mu^\phi(s_t)}
$$

There are two methods to update the target networks. The first one is named hard-replacement, which is to assign $\bar{w}$ and $\bar{\theta}$ in $Q^\psi$ and $\mu^\psi$ to $w$ and $\theta$ after a particular amount of time steps. In addition, the state-action-reward
transitions are stored in a memory buffer and randomly selected during the update process.

Another is to soft-replace the $\bar{w}$ and $\bar{\theta}$ in $\bar{Q}^{\bar{w}}$ and $\mu^{\bar{\theta}}$ to $w$ and $\theta$ with a certain ratio in each training step defined in Equation (2.10). In addition, the state-action-reward transitions are stored in a memory buffer and randomly sampled for computing gradients.

$$\begin{align*}
\bar{w} & \leftarrow \tau_w w + (1 - \tau_w) \bar{w} \\
\bar{\theta} & \leftarrow \tau_\theta \theta + (1 - \tau_\theta) \bar{\theta}
\end{align*}$$

(2.10)

Their results show that DDPG achieves a better control when the agent works on a continuous action space. In transportation research, different RL approaches have been applied to simulate human driving behaviors (Chong et al., 2013) or modeling CAVs (Desjardins and Chaib-Draa, 2011, Zhou and Qu, 2016). For CAVs control, one should be a continuous action problem but all of them discretize CAV’s action. While the discretized action in RL requires outputting multiple values, then chooses the action with the maximum value. By doing this, it requires more computational resources and becomes time-consuming when including a great deal of discretized actions. To bridge this gap, we focus on the continuous action method and apply the DDPG model for controlling CAV’s acceleration rate.

### 2.3.3 Multi-agent reinforcement learning

Multi-agent model is another branch in RL and becomes popular in recent years. In multi-agent RL, it provides each agent with different policy and value network to decentralize the control. By doing this, agents could learn different policy and be
able to cooperate without sharing central control system or one unique controller (Peng et al., 2017, Lowe et al., 2017). This concept offers agents a cooperative system, which shows potential in CAVs applications. However, from the result in Peng et al. (2017) and Lowe et al. (2017), their multi-agent models are trained in a fixed number of agents which constrains its scalability in a large system like CAVs. Although the cooperative mechanism in multi-agent RL has an attractive property, whereas due to the aforementioned shortcomings at this stage, in this study, we still apply a centralized control with all CAVs share the same learned controller.
3.1 Introduction

In order to improve the urban mobility, researchers and practitioners have made numerous efforts to improve freeway efficiency and counter traffic congestion. It has been well recognized that most freeway congestion results from traffic oscillations (or stop-and-go) near freeway ramps, caused by merging activities. Mauch and Cassidy (2002) use loop detector data to show that freeway oscillations frequently form and grow near ramps, suggesting that merging plays an essential role. Zheng et al. (2010) describe how merging near freeway ramps affects the growth of traffic oscillation. Indeed, freeway sections near ramps are considered as the bottlenecks of the freeway system. In this regard, solutions have to be sought that are capable of addressing the oscillations caused by freeway merging.

Ramp metering has been widely deployed in urban areas since the 1960s (May, 1965). By managing on-ramp traffic inflows, the application of meters not only reduces the average freeway delay (Zhang and Levinson, 2010), but also has positive impacts on travel time, emissions and highway safety (Levinson and Zhang, 2006). However, based on the Twin Cities freeways data, Zhang and Levinson (2010) point out that ramp metering increases the queue discharge flow rate by only 3%. Even with this increase, oscillation still occurs from time to time. In light of this,
additional efforts need to be made to improve the efficiency of freeway merging. The recently proposed merging technology of Inter-Vehicle Communication (IVC) and the connected vehicles (Guériau et al., 2016, Zhou et al., 2017a, Ma et al., 2017) provides the potential for managing the freeway ramps efficiently. This technology is proposed to be the basis for the next generation of vehicles on roads (Varaiya, 1993). In this research, we focus on IVC through detecting technology and Adaptive Cruise Control (ACC) as they have already been readily introduced to the market. The longitudinal control in ACC has been studied extensively. However, due to the complexity of the lateral control, it has not yet been well developed. Further, self-driving cars (or autonomous vehicles (AVs)) have recently been designed and introduced (Urmson et al., 2008). Many believe that a combination of autonomous vehicles (AVs) and manually driven vehicles (MVs) will soon share freeways and the AVs are most likely to be designed on the basis of detecting technology and ACC. In this regard, we assume the AVs, in this research, are not able to communicate but are capable of detecting the longitudinal and lateral driving conditions. This research is based on this critical assumption.

A Collaborative Driving System (CDS) with Cooperative Adaptive Cruise Control (CACC) has been proposed by Hallé and Chaib-draa (2005). CACC is based on ACC and designed for AVs to handle both longitudinal and lateral cooperation. In addition, van Arem et al. (2006) conduct a simulation study for CACC vehicles. Their results indicate that AVs could improve the traffic stability only when a high-CACC-penetration rate (>60%) occurs on freeways. Different from the above-mentioned two studies, this research is to improve the performance of CACC vehicles by modifying another widely accepted car-following model named Intelligent Driver Model (IDM). We incorporate a cooperative component within the well-established IDM developed by Treiber et al. (2000) and rename the new model
as Cooperative Intelligent Driver Model (CIDM) for AVs. In this research, the Full Velocity Difference Model (FVDM) is used to mimic MVs.

The contribution of this study is threefold. First, a new CIDM model is developed to incorporate the cooperation of AVs in the traditional IDM model. Second, we demonstrate that a proper vehicle-to-vehicle controlling mechanism with cooperation component could practically improve the freeway performance and smooth the traffic flow dynamics. Third, we conduct a sensitivity analysis to present the impact of AVs to freeway merging.

The rest of this chapter is organized as follows. Section 3.2 illustrates the proposed model and experimental setups. Section 3.3 shows the experimental results and Section 3.4 concludes.

3.2 Microscopic traffic models

3.2.1 Modeling human driving behavior

In recent decades, researchers have developed various microscopic car-following models. Some of the more prevalent are the Gazis-Herman-Rothery (GHR) (Chandler et al., 1958, Newell, 2002, Hoefs, 1972, Herman et al., 1959) and Optimal Velocity (OV) styles of models (Ge et al., 2004, Hasebe et al., 2003, Davis, 2003, Nakayama et al., 2001, Jiang et al., 2001, Lenz et al., 1999, Bando et al., 1998, Mason and Woods, 1997, Bando et al., 1995, Ge et al., 2008, Li and Liu, 2006, Hasebe et al., 2004). OV-type models are microscopic models that clearly show the dynamic formation of congestion (Nakayama et al., 2001, Bando et al., 1998, Bando et al., 1995). The FVDM, one of the improved OV-type models, is investigated by
Jiang et al. (2001). This car-following model improves the performance when simulating a transition in traffic flow and estimating the evolution of the congestion.

The governing equation of the FVDM is:

\[ a_n(t) = a \left[ V(\Delta x) - v_n(t) \right] + \lambda \Theta(s_c - \Delta x) \times \Delta \nu \]  

(3.1)

where \( a_n \) denotes the acceleration rate of the \( n \)th vehicle at time \( t \); \( a \) is the sensitivity constant; \( v_n(t) \) is the vehicle velocity at time \( t \); \( V(\Delta x) \) is an empirical optimal velocity function; \( \lambda \) is a sensitivity factor; \( s_c = 100m \); \( \Theta(x) \) denotes the Heaviside Function; \( \Delta x \) denotes the distance difference from the preceding vehicle; and \( \Delta \nu \) is the difference of the leader’s velocity relative to the follower.

Developing a car-following model needs to take into account both relative speed and headway to a leading vehicle in order to describe all traffic situations (Treiber and Kesting, 2013b), including the free non-interacting traffic which can be reflected by the extra \( \lambda \) term: \( \lambda \Theta(s_c - \Delta x) \times \Delta \nu \). By comparing speed trajectories generated by the OV model and the FVDM, the undesired high accelerations were avoided if the latter is chosen. Further, the vehicle motion delay time can be predicted more accurately using the FVDM (Jiang et al., 2001). In this regard, we choose the FVDM to simulate MVs.
3.2.2 Modeling autonomous vehicular traffic

A well-defined autonomous vehicle model would allow subsequent vehicles to follow preceding vehicles with an optimal velocity and safe headway. Furthermore, from the perspective of a passenger, traveling in AVs is supposed to be an enjoyable experience. In other words, the discomfort induced by higher rates of acceleration or deceleration needs to be minimized by smoothing the travel trajectory. For the sake of creating a smooth trajectory, multi-detecting devices, such as equipping radars, could be incorporated into AVs. The AVs can then cooperate with and pre-act in relation to surrounding vehicles, depending upon the understanding of the detected traffic conditions.

In order to simulate the driving behaviors of AVs, in this study, we use the CIDM with a cooperative driving strategy. The original IDM is first proposed by Treiber et al. (2000) to simulate bottleneck congestions. Different from other traditional microscopic models, IDM provides collision-free behavior as well as a self-organized characteristic. IDM is also utilized in Adaptive Cruise Control (ACC) (Kesting et al., 2008) for the following reasons:

- It provides collision-free and smooth manner traffic flow; and
- Environmental variable changes, such as deceleration of the preceding vehicle, will not result in turbulent traffic or, in particular, oscillation.

The IDM acceleration rate is presented as follows:

\[
 a_x = a \left[ 1 - \left( \frac{v}{v_s} \right)^3 - \left( \frac{s^*(v_s, \Delta v_s)}{\Delta x} \right)^2 \right] \tag{3.2}
\]
This equation combines both an accelerating term \( a_{\text{acc}}(v) = a \times \left[ 1 - \left( \frac{v}{v_s} \right)^2 \right] \) towards the desired speed \( v_0 \) on a free road, and a braking term \( a_{\text{brake}}(\Delta x, v_s, \Delta v_s) = -a(s^*/\Delta x)^2 \), where the \( s^* \) (Treiber, 2011) is given by:

\[
s^*(v_s, \Delta v_s) = s_0 + \max \left( 0, v_s T + \frac{v_s \Delta v_s}{2 \sqrt{ab}} \right)
\]  

(3.3)

The minimum distance \( s_0 \) is designed for vehicles in low-velocity circumstances; \( a \) denotes the same parameter as in (3.2) that represents maximum acceleration; \( T \) is a safe time gap; \( b \) is the desired deceleration rate.

In order for cooperative AVs to obtain traffic information from one or more preceding vehicles, the capability for spatial anticipation is required in the CIDM. Based on the concept of spatial anticipation in the Human Driver Model (HDM), which has been proposed as an extension of the IDM (Treiber et al., 2007a, Treiber et al., 2006), this anticipation can now also be applied to the proposed CIDM, which splits the IDM’s \( a_n \) into the following:

\[
a_n(\Delta x, v_s, \Delta v_s) = a_n^{\text{free}} + \sum_{\alpha} a_{\alpha}^{\text{int}}(\Delta x_{\alpha}, v_s, \Delta v_{\alpha})
\]  

(3.4)

where \( a_{\alpha}^{\text{int}} \) is \( a_{\text{brake}} \) with the consideration of vehicle-vehicle interaction, and \( a_n^{\text{free}} \) has the same definition as \( a_{\text{free}} \) above.
Given the multi-anticipation that strengthens their performance, AVs can theoretically detect more than one preceding vehicles in a longitudinal direction, but this still does not represent cooperative driving behavior well. Some critical scenarios, including forced merging from the on-ramp that triggers a large reduction in the preceding headway, would result in an unexpectedly high deceleration for following IDM-based vehicles. Therefore, more practical cooperative driving behavior needs to have greater predictive capabilities in both the longitudinal and latitudinal directions. Accordingly, cooperative rules are introduced here.

The first portion of these cooperative rules is from (Kesting et al., 2008, Treiber et al., 2007b), who conducted a traffic-state detection model using an Exponential Moving Average (EMA) concept (see (3.5)), by comparing $v_{EMA}$ as the derivative term of $x_{EMA}$ with the average velocity under a certain traffic condition. The best driving strategy then can be selected. Based upon this concept, the dynamic traffic changes that occur in front of AVs will be detected and responded (details can be found in (Kesting et al., 2008)).

$$x_{EMA}(t) = \frac{1}{\tau} \int_{t-\tau}^{t} e^{-\gamma(t-t')} x(t') dt'$$  \hspace{1cm} (3.5)

$a$, $b$ and $T$ are replaced by the parameters shown in (3.6), and then substituted into (3.2) and (3.3) according to a certain driving strategy (Kesting et al., 2008).

$$a = \lambda_\gamma a, \quad b = \lambda_\gamma b, \quad T = \lambda_\gamma T$$  \hspace{1cm} (3.6)
The second portion of the cooperative rules is a simplified Lane-Changing Impact (LCI) rule. Although LCI models have gradually been paid more attention over the last three decades, they normally focus on modeling macroscopic behavior (Zheng, 2014). Merging, as a mandatory type of lane-change, could significantly influence traffic dynamics on the main lane. With installing the detection equipment, ramp vehicles’ can be detected instantly once they have entered the detection range as shown in Fig. 3.1. Within the detection ranges, AVs on the main road could thereby gently adjust their acceleration rates and prepare a larger gap in advance if they needed to cooperate with ramp vehicles.

Fig. 3.1. Illustration of AVs’ detection range

Herein, we assume that AVs, to a certain extent, would give a larger gap for on-ramp vehicles in order to handle the uncertainty of human drivers and reduce the
collision risk. Therefore, we introduce another multiplication factor, $\lambda_{\Delta x}$, to imitate the merging impact on the following traffic flow. In this study, the aforementioned method is only being applied to AVs on the main road.

$$\lambda_{\Delta x} = \max \left[ 0.4, \left( \frac{\Delta x_0(t)}{R_d} \right)^2 \right] \quad \Delta x_0(t) \leq R_d \quad (3.7)$$

$$\Delta x = \lambda_{\Delta x} \Delta x \quad (3.8)$$

where $\Delta x_0(t)$ is the remaining distance at time $t$ from the merging point (or on-ramp entry), $R_d$ is the detection range of the main-road AV, seen in Fig. 3.1. During the time when the main-road vehicles are preparing to accept the merged vehicles, other multiplication factors are adjusted according to Table 3.1. The values for the multiplication factors are modified from (Kesting et al., 2008). The $\lambda_{\Delta x}$ is an extended factor to their original study in order to describe the merging cooperation.

<table>
<thead>
<tr>
<th>Merging condition</th>
<th>$\lambda_a$</th>
<th>$\lambda_s$</th>
<th>$\lambda_r$</th>
<th>$\lambda_{\Delta x}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before merge</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>Equation (3.7)</td>
</tr>
<tr>
<td>After merge</td>
<td>2</td>
<td>1</td>
<td>0.5</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.1. $\lambda_a$, $\lambda_s$, $\lambda_r$ and $\lambda_{\Delta x}$ values for the CIDM during merging
3.3 An illustrative case study of Pacific motorway

3.3.1 Comparison of trajectories

The CIDM is the model to simulate AVs’ driving behaviors. Unfortunately, there is no any existing data to compare with. In order to evaluate the validity of the CIDM, we have to evaluate the similarity of the IDM with real data. Therefore, the trajectories data extracted from video images of northbound traffic on I-80 in Emeryville, California (FHWA, 2008) and northbound traffic on M1 in Queensland are compared to the IDM trajectories. A section of trajectory data without lane-changing and merging are selected and shown in Fig. 3.2. A disturbance that is triggered by a lane-change maneuver transmits forward to the selected platoon. In the IDM trajectories, by fixing the trajectory of the leading vehicle from the real data, the remaining trajectories are generated by the IDM model with default settings of: \( v_0 = 110 \text{ km/h}; \ T = 1.1 \text{ s}; \ a = 1.4 \text{ m/s}^2; \ b = 2 \text{ m/s}^2; \) and \( s_0 = 2 \text{ m}. \)

![Fig. 3.2. Trajectories comparison between real data and IDM](image-url)
From Fig. 3.2 (a), the IDM trajectory successfully reflects the propagation of disturbance and the oscillation phase. However, with these parameter settings, the gap distance tends to be shorter in the congested phase and longer in the free-flow phase. Furthermore, the deceleration and acceleration stages are more evenly compared to the real data. In addition, from Fig. 3.2 (b), the IDM with default settings has the trajectories which are close to the traffic flow on the M1 if there is no disturbance. As a result, this property of IDM and IDM-based models could be utilized in the design of the AV car-following model.

### 3.3.2 Simulation Setup

We investigate a section of the freeway located in Brisbane, which is illustrated in Fig. 3.3. The freeway consists of three lanes in each direction. The figure only shows the northbound direction from the Gold Coast to Brisbane. At the on-ramp section, during traffic peak hours, vehicles from the Gold Coast and the southern suburbs of Brisbane contribute to a dense traffic flow.

![Fig. 3.3. Scenario of merging simulation](image-url)
In order to simulate the critical scenario, we assume the left blue lane (shown in Fig. 3.3) of the freeway to be the only lane affected by the merging maneuver, and we assume there is no lane changing from either the left or the right onto this blue lane. Apart from that, the main-road vehicles on the freeway are initialized with an incipient velocity (here taken to be 110 km/h as the speed limit) and time headway based on the flow circumstances. In this study, the traffic flow on the freeway in the lane adjacent to the on-ramp and the ramp are set at 1800 veh/h. For this simulation, we generate 300 main road vehicles with the same initial state.

Lane-change types can be classified to two classes: discretionary lane change (DLC) and mandatory lane changes (MLC) (Zheng, 2014, Wang et al., 2014b). For vehicles on the ramp, their merging motion is MLC and can be represented by a lane-changing model introduced by Hidas (2005). Hidas (2005) specified that the subject vehicle is the merging vehicle. The subject vehicle can merge into the target lane only if the gaps between the subject vehicle and leader on the target lane \( g_t \), and the subject vehicle and follower on the target lane \( g_f \) satisfy the following criteria:

\[
\begin{align*}
    g_t &\geq g_{t,\text{min}} \quad \text{and} \quad g_f \geq g_{f,\text{min}} \\
    g_{t,\text{min}} &= g_{\text{min}} + \begin{cases} 
        c_t(v_s - v_f), & \text{if } v_s > v_f \\
        0, & \text{otherwise}
    \end{cases} \\
    g_{f,\text{min}} &= g_{\text{min}} + \begin{cases} 
        c_f(v_f - v_s), & \text{if } v_f > v_s \\
        0, & \text{otherwise}
    \end{cases}
\end{align*}
\]
where \( c_i = c_f = 0.9 \); \( g_{\text{min}} = 2.0 \) m which is the average minimum safe constant gap. All the parameter values are selected from (Hidas, 2005) which are used in a simulation of on-ramp merging. Note that we assume the lane changing actions to be completed instantaneously while a lane change, in reality, is a continuous motion. Therefore, the estimated capacity could be overestimated, as the duration of the lane change is not taken into account.

Table 3.2. Model parameters of the FVDM

<table>
<thead>
<tr>
<th>Model parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>0.41</td>
</tr>
<tr>
<td>( s_c )</td>
<td>100</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>0.5</td>
</tr>
</tbody>
</table>

MVs, which are simulated by the FVDM, are assumed to follow the parameters listed in Table 3.2. The optimal velocity function in the equation (3.1) is determined by the following: \( V(\Delta x) = 16.8[\tanh 0.086(\Delta x - 25) + 0.913] \), which is calibrated from observed data on a Japanese motorway (Bando et al., 1998).
AVs, which are governed by the CIDM, are assumed to follow the parameters listed in Table 3.3. Milanes et al. (2014) and Milanes and Shladover (2014) have conducted a test regarding the CACC vehicles. Considering the advantages of quick response and action by AVs, they found the shortest time gap to be 0.6 s. As a result, we selected 0.6 s as the safe time gap for the CIDM. The other parameters remain the same as the IDM default. However, it should be noted that none of these parameters have been validated and tested. This needs to be accomplished in a future study. In addition, referring to Fig. 3.1, the detection range \( R_d \) is assumed as 30 m.

### 3.3.3 Trajectory

The trajectory of vehicles on the road represents the entire string stability and traffic dynamics. The trajectory graphs produced from the simulation results are depicted in Fig. 3.4. In this study, we compare four scenarios, with different proportions of AVs (0%, 5%, 15% and 25%) to present the transitional phase as AVs are gradually introduced.
Fig. 3.4. Trajectories of 0%, 5%, 15% and 25% of AVs

From (a) in Fig. 3.4, if there is 0% of autonomous vehicle penetration in the traffic stream, continuous and irregular merging behaviors would have a negative effect on string stability and result in an oscillation (stop-and-go scenario) in the upstream traffic flow. However, only a 5% AVs in the system can proportionately relieve the oscillation, as shown in Fig. 3.4 (b). Further increases in the AV’s proportion to 15% and 25% will lead to further improvements in the traffic flow stability.
AVs can smooth the trajectory and relieve the oscillation. However, the congestion occurs earlier and the oscillation may transmit further to the upstream when an AV-penetration-rate is less than 25%, which is illustrated in Fig. 3.4. In particular, the white gap between AV and MV trajectories, in Fig. 3.4 (b), (c), and (d), are evidence of how the oscillation is relieved and transmitted.

3.3.4 Driving Sensitivity

The previous IDM has the disadvantage of the high sensitivity due to the over-responding to lane-changing and merging. Kesting et al. (2010) attempted to reduce the sensitivity issue for IDM-based vehicles. Their approach is a post-act method by making an assumption: the leading vehicle will not change its acceleration for the next few seconds. Therefore, the acceleration adjustment will be delayed until after the cut-in maneuver. However, either a human or an AV should be able to take pre-action to decelerate in advance, in order to avoid the potential collision. Therefore, our approach to reducing sensitivity is a proactive method. Therefore, the CIDM leads to a positive driving strategy that is more reliable and logical in an autonomous driving system.

3.3.5 Safety Analysis
Fig. 3.5. Speed dispersion on a congested freeway segment (1.5 to 2.8 km), from 9th to 13th min, under different proportions of CIDM-based AVs

The speed dispersion is well recognized as a measure for estimating the risk of traffic accidents or the safety of traveling (Qu et al., 2014a). Previous researchers have often used standard deviation (SD) to describe speed dispersion. Disturbed traffic flow has a greater value of SD, and a higher SD value indicates a higher risk of a traffic accident. By contrast, a lower level of SD represents a lower risk due to the fact that most of the vehicles travel with similar speeds.

The SD results are shown in Fig. 3.5. These results indicate that without AVs on the road, the massive oscillation that is caused by merged-in vehicles results in a high SD (usually between 6 and 8). However, with an increasing number of AV
penetrating into freeways, the SD could be reduced progressively. This also implies that the increase of AVs has a positive impact on traffic dynamics.

### 3.3.6 Travel Time

The travel time is used as a measure of driving efficiency. The average travel time is a performance measure for the overall transport system, evaluating the efficiency by looking at the economic cost of traffic jams.

Based on the simulation result, the average travel times for the freeway traffic flow through the congested section, from 1 to 2.8 km, under different AV penetrations are shown in Fig. 3.6. In particular, it shows that the AVs can promote the efficiency of traveling on a congested road.
3.3.7 Space Mean Speed

The space mean speed takes a whole road segment into account and describes the speed more accurately than the time mean speed.

Fig. 3.6. Average travel time on the congested freeway segment (1 to 2.8 km)
The result of space mean speed is shown in Fig. 3.7. Due to the severe oscillations when no AV is on the freeway, the space mean speed of this scenario tends to be lower than other cases. In addition, the penetration rate of AVs has a positive impact on space mean speed.

### 3.3.8 Sensitivity Analysis of Safe Time Gap

The safe time gap in the IDM has essential impacts on travel efficiency and safety when AVs are designed. Thus, the sensitivity analysis of the safe time gaps (0.4s, 0.6s, 0.8s, 1s, and 1.2s) has been carried out. For each safe time gaps, we simulate all cases (0%, 5%, 15% and 25% AVs) under the same initial condition. The average
speed dispersion and average travel time for all cases are concluded in Table 3.4 and Table 3.5.

Table 3.4. Average Value of Speed Dispersion

<table>
<thead>
<tr>
<th>Percentage of AVs</th>
<th>0.4 s</th>
<th>0.6 s</th>
<th>0.8 s</th>
<th>1.0 s</th>
<th>1.2 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>6.86</td>
<td>6.86</td>
<td>6.86</td>
<td>6.86</td>
<td>6.86</td>
</tr>
<tr>
<td>5%</td>
<td>5.35</td>
<td>5.10</td>
<td>5.08</td>
<td>4.69</td>
<td>5.17</td>
</tr>
<tr>
<td>15%</td>
<td>3.59</td>
<td>3.61</td>
<td>3.51</td>
<td>2.86</td>
<td>3.07</td>
</tr>
<tr>
<td>25%</td>
<td>3.70</td>
<td>3.16</td>
<td>3.10</td>
<td>2.08</td>
<td>1.90</td>
</tr>
</tbody>
</table>

Note: These are average values of speed dispersion simulated by considering different safe time gap in the IDM for all cases.

Table 3.5. Average Travel Time (Minute)

<table>
<thead>
<tr>
<th>Percentage of AVs</th>
<th>0.4 s</th>
<th>0.6 s</th>
<th>0.8 s</th>
<th>1.0 s</th>
<th>1.2 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>3.32</td>
<td>3.32</td>
<td>3.32</td>
<td>3.32</td>
<td>3.32</td>
</tr>
<tr>
<td>5%</td>
<td>3.12</td>
<td>3.15</td>
<td>3.23</td>
<td>3.18</td>
<td>3.24</td>
</tr>
<tr>
<td>15%</td>
<td>3.09</td>
<td>3.01</td>
<td>3.12</td>
<td>3.32</td>
<td>3.36</td>
</tr>
<tr>
<td>25%</td>
<td>2.81</td>
<td>2.91</td>
<td>2.92</td>
<td>3.45</td>
<td>3.49</td>
</tr>
</tbody>
</table>

Note: These are average travel time simulated by considering different safe time gap in the IDM for all cases. The unit for travel time is minute.

The result in Table 3.4 shows that with the increase of AVs’ percentage in the mixed traffic flow, the overall speed dispersion for all cases reduces to a lower level, which indicates that the safety level is improved in all cases. However, the degrees of improvement are different, as a longer safe time gap can result in a greater
improvement in speed dispersion. It can further relieve the oscillation. This is explainable as a longer safe time gap increases the average headway. As such, the overall travel speed can be harmonized and the safety level can be continuously improved.

On the contrary, the trend of average travel time, shown in Table 3.5, differs from the trend of average speed dispersion. Specifically, the safe time gap is negatively related to the average travel time. On the other hand, in this study, when the safe time gap is less than 1.0 s, the increase of the percentage of AVs can result in improving travel efficiency. Nevertheless, when the safe time gap is further increased to greater than 1 s, a higher AVs percentage does not improve the travel efficiency.

3.4 Conclusions

It is anticipated that detecting technology is able to detect and foresee traffic conditions not only longitudinally (detecting several preceding vehicles) but also laterally (detecting vehicles in other lanes). A proposed CIDM-based controller determines the AV’s acceleration/deceleration rate as a response to the actions of surrounding vehicles, with an aim to improve road capacity and string stability. An extra term related to detecting merging point has been added to the original IDM to form the CIDM. The CIDM-based vehicle detects on-ramp vehicle and cooperates with them by preparing accepted gap for on-ramp merging. The on-ramp vehicle than be capable to merge without a full stop. The CIDM-based AVs are, therefore, more capable of maximizing the impact of on-ramp merging vehicles. In doing so, CIDM-based AVs are able to eliminate or relieve freeway oscillations.
A case study was conducted to evaluate the CIDM’s performance. From the trajectory result, we observe a pattern that with an increase in the proportion of CIDM-based vehicles, even with just 5% of CIDM-based vehicles, the traffic oscillation can be significantly mitigated. It continuously optimizes the traffic flow when the percentage of CIDM-based vehicle increases.

According to the result of speed dispersion, a higher AV penetration rate indicates safer freeways if the proposed CIDM-based controller is adopted. Further, a sensitivity analysis is carried out to illustrate that the safe time gap plays an essential role in improving travel efficiency and safety. We found that a longer safe time gap stabilizes traffic flow, but also increase average travel time. Furthermore, when the safe time gap is less than 1.0 s, the increase of the percentage of CIDM-based vehicles can result in an improvement in travel efficiency. On the other hand, when the safe time gap is further increased to greater than 1 s, a higher AVs percentage does not help the overall travel efficiency.

In sum, the results show that a proactive method for optimizing traffic efficiency is acceptable. A cooperative mechanism can be established once AVs detecting possible on-ramp vehicles. The cooperative mechanism compromises the small amount of travel time in main road vehicles to create a global benefit in the overall transportation system.
Chapter 4
A Modified Intelligent Driver Model for Optimizing
Highway T-Junctions Merging

4.1 Introduction

Research on optimizing travel efficiency and safety for highways with on-ramp segment has been conducted (van Arem et al., 2006, Zhou et al., 2017b, Letter and Elefteriadou, 2017), which shows an efficiency boost when Automated Vehicles (AVs) gain a certain level of automation in cooperation. However, on-ramp segments differ from T-junctions or priority junctions in a way that vehicles on-ramp can maintain certain speed before merging into the highway.

In a T-junction, vehicles have to stop before merging, which generally requires a longer move up time for merging vehicles. Traffic oscillation phenomenon, or stop-and-go, can easily form compared with on-ramp and other bottlenecks. A typical merging from priority junctions requires a greater gap for the first queuing vehicle and smaller gaps for the subsequent queuing vehicles. Therefore, if gaps sufficient for discharging a queue of vehicles (group merging) are provided, it will be more efficient than one-by-one discharging (single merging). By adopting this concept from IVC, Connected and Automated Vehicles (CAVs) can be used to optimize traffic conditions in priority junctions, because

- CAVs share information, such as the queue length on merging lane, transmitted from road infrastructure;
CAVs plan their driving strategies to prepare an acceptable gap for group merging rather than single merging.

In this study, we proposed a modified Intelligent Driver Model (IDM) to gain a cooperative driving strategy. The contribution of this study is fourfold:

• Proposed a modified IDM for cooperative driving for highway segments with priority junction;
• Conducted various simulations regarding a baseline benchmarking scenario, a metering scenario, and three mixed traffic scenarios to fairly compare the performance in safety, travel efficiency and emission;
• Evaluated two different methods for obtaining a cooperative strategy and compared their performance;
• Analyzed the best location on highways to apply the cooperative driving strategy.

This rest of this chapter is organized as follows. Section 4.2 illustrates the proposed model and experimental setups. Section 4.3 shows the experimental results and Section 4.4 concludes.

4.2 Experimental design

The connected system developed here is to maximize merging capacity and traveling efficiency. Fig. 4.1 illustrates a hypothetical T-junction consisting of one single line highway and one single merging lane attached to the highway mainline. In a connected scenario, a vehicle detector is installed at the merging point for counting the number of queuing vehicles here. When the length of the queue exceeds a certain threshold, the detector then sends a signal to the signal transmitter.
at the start of the buffer zone. After receiving the signal from the detector, the signal transmitter will be activated. A “cooperative” signal will be transmitted to the first CAV that enters the buffer zone. Once this selected CAV passes another signal transmitter located at the end of the buffer zone, another signal will be sent to this CAV to switch back to a normal mode. While traveling inside the buffer zone, CAVs can adjust their driving behavior based on a cooperative strategy.

![Diagram of the connected system](image)

**Fig. 4.1. Illustration of the connected system**

### 4.2.1 Cooperative car-following strategy

For CAVs, a cooperative car-following strategy is needed to jointly maximize merging capacity and minimize the negative impact caused by merging. Fig. 4.2 (a) demonstrates a conventional merging situation without a suitable cooperative strategy. With only a short gap provided by the following MV, few vehicles can merge into the mainline. On the contrary, if the following vehicle offers merging vehicles a sufficient gap, most of the vehicles on merging lane can be discharged at once. Fig. 4.2 (b) shows the latter idea as a CAV enlarges the gap in advance for a greater amount of merging acceptance. The latter case presents a higher merging capacity (about six times higher than the first case).
Fig. 4.2. Illustration of the cooperation strategy

To achieve that, an appropriate car-following model has to be carefully chosen among a wide range of available models. Widely accepted models, such as Optimal Velocity Model (OVM) (Bando et al., 1995), Intelligent Driver Model (IDM) (Treiber et al., 2000) and Newell’s model (Newell, 2002), are representatives in this model-family. Further model development and upgrading have been carried out for these models (Milanes and Shladover, 2016, Yu and Shi, 2015, Kesting et al., 2010, Treiber et al., 2007b, Treiber et al., 2006, Jiang et al., 2001). Due to properties including collision-free, meta-stability, reasonably interpreted model parameters, we select the IDM for simulating MVs as the IDM carries those properties. Additionally, we develop an IDM-based CAVs model for aforementioned cooperative strategy. The original IDM formula is defined as follows

\[
a_i = a \left[ 1 - \left( \frac{v_i}{v_0} \right)^4 \right] - \left[ \frac{s_i + \max \left( 0, T v_i + \frac{v_i \Delta v_i}{2 \sqrt{ab}} \right)}{s_i} \right]^2
\]

(4.1)
where $a_i$ represents calculated acceleration rate for a follower; $v_i$, $\Delta v_i$ and $s_i$ denote observed speed, relevant speed and gap distance; $v_0$, $T$ and $s_0$ denote desired velocity, safe time headway and jam distance respectively; $a$ and $b$ stand for the maximum acceleration and deceleration rate. The max[·] operation is adopted from (Treiber, 2011). All the parameter values are defined in Table 4.1.

Table 4.1. Default parameter values in IDM for simulating MVs (Treiber et al., 2000).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desired speed $v_0$</td>
<td>120 km/h</td>
</tr>
<tr>
<td>Safe time gap $T$</td>
<td>1.6 s</td>
</tr>
<tr>
<td>Maximum acceleration $a$</td>
<td>0.73 m/s²</td>
</tr>
<tr>
<td>Desired deceleration $b$</td>
<td>1.67 m/s²</td>
</tr>
<tr>
<td>Jam distance $s_0$</td>
<td>2 m</td>
</tr>
</tbody>
</table>

As discussed above, IDM can be a good model to simulate MVs. As for CAVs, in contrast, the IDM cannot be simply employed due to lack of a communicative property. In this regard, we modify the original IDM model to account for communications between vehicles and infrastructure. As can be seen in Fig. 4.1, we assume that CAVs can receive a signal from an activated roadside transmitter. Based on the signal, the selected CAV can decelerate to increase the safe time headway $T$ in order to create a larger headway to allow group merging.

An appropriate CAVs cooperative design will need to satisfy the following criteria:
CAVs should gradually increase safe time headway $T$ to avoid traffic oscillation;
CAVs should accelerate and catch up after passing the signal transmitter at the end of buffer zone to avoid an unnecessary disturbance caused by merged vehicles;
CAVs can maintain a shorter time headway compared with MVs as they eliminate human error; and
In this study, decisions are made by roadside communicative infrastructure rather than CAVs.

Firstly, for merging from the T-junction, we select the value of critical gap and follow up time in Table 4.2, which has been adopted in the *Highway Capacity Manual* (Transportation Research Board, 1997). As can be seen in the table, the critical gap for the first queuing vehicle is indeed much higher than the follow-up times of the subsequent vehicles.

<table>
<thead>
<tr>
<th>Critical gap $\tau_c$</th>
<th>Default value</th>
<th>4.1 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Follow up time $\tau_f$</td>
<td>Default value</td>
<td>2.2 s</td>
</tr>
</tbody>
</table>

The cooperative safe time gap $T^c$ for CAVs can be defined as

$$T^c = (n_m + 1)\tau_c$$  \hspace{1cm} (4.2)
where $n_{mt}$ denotes a threshold number of queuing vehicles. For example, if set $n_{mt} = 5$, the signal transmitter at the start of the buffer zone will be activated once more than 5 vehicles are detected queuing to merge, and the transmitter sends a “cooperation” signal to the nearest CAV that has not entered the buffer zone.

However, a hard replacing of $T^c$ from $T$ will introduce a sudden change and result in a disturbance or even oscillation for following vehicles. We will discuss more this in the next section. Here, instead of a hard replacement, we introduce a soft $T$ replacement in Equation (4.3).

$$T_i^c = \min \left[ T_{\text{max}}, \frac{T_{\text{max}} - T^{CAV}}{T_{tb}} t + T^{CAV} \right]$$

where:

\[
\begin{align*}
T_{tb} &= \alpha l_b / v_{CAV} \\
T_{\text{max}} &= (n_m + 1) \tau_c
\end{align*}
\]

$T_i^c$ denotes a dynamic cooperative safe time gap at time step $t$. $T_{tb}$ is a buffer time calculated from the length of the buffer zone ($l_b$) and the current velocity of this CAV. $\alpha$ is a scale factor of the buffer time. $T^{CAV}$ and $T_{\text{max}}$ are the default and maximum time gap for CAVs respectively. $n_m$ is an instant value that records a current queue length. The whole system activates if and only if $n_m \geq n_{mt}$.

Intuitively, if we fix the $T_{tb}$ to five seconds and $n_m$ to five vehicles, this dynamic equation becomes static and the graph can be visualized in Fig. 4.3 for illustrating the relationship between $t$ and $T_i^c$. CAV’s parameters used in this study are described in Table 4.3.
Fig. 4.3. An illustration for a static $T_1^c$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safe time gap for CAVs $T^C_{AV}$</td>
<td>0.8 s</td>
</tr>
<tr>
<td>Merge threshold $n_{mt}$</td>
<td>5 vehicles</td>
</tr>
<tr>
<td>Scale factor $\alpha$</td>
<td>0.7</td>
</tr>
<tr>
<td>Buffer zone length $l_b$</td>
<td>800 m</td>
</tr>
<tr>
<td>Cooperative time gap $T_i^c$</td>
<td>Equation (4.3)</td>
</tr>
<tr>
<td>Buffer time $T_i^b$</td>
<td>Equation (4.3)</td>
</tr>
<tr>
<td>Maximum cooperative gap $T_{max}$</td>
<td>Equation (4.3)</td>
</tr>
</tbody>
</table>

Note: The rest of parameters in the modified IDM are the same in Table 4.1.
The 0.8 s safe time gap is obtained from Zhou et al. (2017b) for CAVs. The selection of buffer zone length and location affects overall performance, and we use a simulation to further validate suitable options in the next section. The cooperative scheme involves communication between infrastructure and CAVs, and its procedures are presented as below.

- Vehicles on merging lane do not need to wait for cooperation. They can merge once an acceptable gap is presented. Otherwise, following procedures apply;
- Vehicle detector finds more than \( n_{\text{cr}} \) vehicles queue at merging point and activates signal transmitter at the start of buffer zone;
- Signal transmitter sends a cooperation signal to the next CAV which is about to enter the buffer zone;
- Signal is received by a CAV, and this CAV calculates a dynamic \( T_i^c \) to obtain an acceleration rate from the modified IDM; and
- This CAV switches \( T_i^c \) back to \( T^{CAV} \) after it passes the buffer zone (end of cooperation).

### 4.2.2 Scenarios

In order to show the results, we run experiments in multiple scenarios. It covers

- a case without any CAV (Baseline);
- a signalized T-junction (With signal);
- with 10% of the vehicles are CAVs (10% CAVs);
- with 50% of the vehicles are CAVs (50% CAVs); and
- all vehicles are CAVs (100% CAVs).

All the scenarios’ initialization status are presented in Table 4.4.
Table 4.4. Experimental setup

<table>
<thead>
<tr>
<th>Settings</th>
<th>Value</th>
<th>Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highway mainline traffic flow</td>
<td>1600 veh/h</td>
<td></td>
</tr>
<tr>
<td>Merging lane traffic flow</td>
<td>160 veh/h</td>
<td></td>
</tr>
<tr>
<td>Highway mainline initial speed</td>
<td>110 km/h</td>
<td>All</td>
</tr>
<tr>
<td>Merging lane initial speed</td>
<td>60 km/h</td>
<td></td>
</tr>
<tr>
<td>Mainline length</td>
<td>1300 m</td>
<td></td>
</tr>
<tr>
<td>Merging point</td>
<td>1300*0.9=1170 m</td>
<td></td>
</tr>
<tr>
<td>Mainline red phase</td>
<td>12 s</td>
<td>With signal</td>
</tr>
<tr>
<td>Mainline green phase</td>
<td>120 s</td>
<td></td>
</tr>
<tr>
<td>Buffer zone start</td>
<td>1170-900=270 m</td>
<td>Scenarios with CAVs</td>
</tr>
<tr>
<td>Buffer zone end</td>
<td>1170-100=1070 m</td>
<td></td>
</tr>
</tbody>
</table>

We show a congested setting, which the combined traffic can reach the flow of 1760 veh/h, or an average headway of 2.04 s/veh. The merging traffic flow is assumed to be 10% of that in mainline. Therefore, in “With signal” scenario, we use the green time allocation of 1:10 too. For the scenarios with CAVs, we select that the buffer zone starts and ends at 900 m and 100 m from the merging point, respectively. A different buffer zone segment may affect overall performance, which will be analyzed and included in next section.
4.3 Results and discussion

4.3.1 Trajectory

Here we show the trajectory result for all scenarios. We shift the view window between 450 s and 1000 s to omit the initialization phase (0 s to 450 s) and focus only on a critical phase where traffic oscillation may form.
Fig. 4.4. Trajectory result
In Fig. 4.4, MVs and CAVs are colored in blue and red, respectively. The first two scenarios are based purely on MVs. In case (a), it is clear that MVs on mainline travel with high speed but rarely cooperate with vehicles on merging lane, which causes few successful merging. When the traffic light is involved, it creates a time gap for merging. However, with the periodical light cycle, a periodical traffic oscillation is formed. In the scenarios of mixing CAVs in traffic flow, no obvious traffic oscillation is formed. Additionally, with the increase in the percentage of CAVs in the flow, fewer impacts apply on following vehicles.

### 4.3.2 Global comparison

A quantitative global comparison is carried out in this section to compare travel efficiency and merging capacity. In here, we plot the value at each time step and the result is shown in Fig. 4.5.

![Fig. 4.5. Average speed and queue length at each time step](image)

Fig. 4.5 (a) and (b) indicate average speed on highway mainline and merging lane for each scenario. The “With signal” scenario shows the lowest average speed
(about 30 km/h) on mainline and less than 10 km/h on merging lane. The baseline presents a high mainline speed (approximately 90 km/h) but low speed on merging lane (similar to “With signal” scenario). Average speed for scenarios with CAVs is high in both lanes (more than 60 km/h and 20 km/h on the mainline and merging lane, respectively). In addition, a higher percentage of CAVs penetration rate tends to show a higher average speed on mainline.

Fig. 4.5 (c) suggests that although the baseline has a similar average speed as scenarios of 50% and 100% CAVs, its discharging capacity of queuing vehicles is the worst. Further, unlike CAVs scenarios that can discharge almost queues, the queue length in the baseline and “With signal” cases keep raising and accumulating.

### 4.3.3 Local comparison

The local comparison provides with a vehicular view. To achieve this, a filtering procedure is required. We only keep the vehicles that travel through the whole road section, leave the truncated data abandoned. Fig. 4.6 demonstrates a filtered trajectory result.
Firstly, we compare vehicles in three aspects including travel time, safety, and emission. For illustration purposes, this study uses inverse time-to-collision (iTTC) for measuring safety. The equation of iTTC is written in Equation (4.4). Compared with other measures such as time-to-collision (TTC) (Hoffmann and Mortimer, 1994) that are widely implemented in many studies for analyzing safety (Kuang et al., 2017, Weng et al., 2015, Qu et al., 2014b, Qu and Meng, 2014, Meng and Qu, 2012, Kuang et al., 2015b, Kuang et al., 2015a), iTTC, even with a negative value, is more accurately addresses collision risks (Balas and Balas, 2006). Additionally, iTTC
solves the infinity problem in TTC when the speed of leading vehicle and following vehicle is equal.

\[
iTTC_n = \int \max \left[ 0, \frac{v_n(t) - v_{n-1}(t)}{p_{n-1}(t) - p_n(t) - l} \right] dt \quad (4.4)
\]

where \(v_n(p_n)\) and \(v_{n-1}(p_{n-1})\) are the velocity (position) for follower and leader, respectively; \(l\) denotes the car length, in this study, the length equals 5 m.

There are many models for evaluating fuel consumption and emissions such as the models in Xu et al. (2018) and Ahn et al. (2002). We select the widely used VT-Micro model (Ahn et al., 2002) for evaluating fuel consumption in this study. The \(e\) function is formulated in Equation (4.5). It is noted that truncation of speed and acceleration range of this model may cause underestimation of fuel consumption.

\[
e(v_n(t), a_n(t)) = \exp \left[ \sum_{i=0}^{3} \sum_{j=0}^{3} K_{ij} \left( a_n(t) \vphantom{\left( v_n(t) \right)} \right) \left( v_n(t) \right) \left( a_n(t) \vphantom{\left( v_n(t) \right)} \right) \right] \quad (4.5)
\]

where coefficient \(K_{ij}(a_n(t))\) depends on the sign of \(a_n(t)\). This study uses values of \(K_{ij}(a_n(t))\) in Ma et al. (2017).
On highway mainline, the vehicular result shows in Fig. 4.7 indicates a similar pattern that the “With signal” scenario leads to the longest travel time, highest risk and greatest fuel consumption. It also suggests that increasing CAVs percentage doubles the travel efficiency, safety and reduces fuel consumption to half on mainline. The baseline also indicates a result similar to CAVs’, but the high performance can only occur while rare vehicles merging into mainline. Similar results show in the Fig. 4.7 (2) for merging lane. However, the iTTC result for merging lane illustrates that the risk level is relatively higher than mainline except “With signal” scenario. The primary cause is that the cooperative strategy is activated only when vehicles are queuing on merging lane. The stopping and queuing effect, hence, makes a higher risk level.
Fig. 4.8. Vehicular comparisons of average values for travel time, iTTC, emission

and queue length

Fig. 4.8 reveals a hidden relationship among all scenarios. On the highway mainline, the “Baseline” scenario has an average performance similar to CAVs’ scenarios only by sacrificing its merging capacity. The figure also indicates that by mixing only 10% of traffic flow with CAVs, the travel performance on mainline will close to the baseline, while the average queue on merging lane reduces dramatically to 1/5 of it in the baseline case. The further increment of CAV rate from 50% shows a less improvement in performance. Values of emission and travel time are similar for both mainline and merging lane under CAVs cases. However, these two values are
nearly doubled for vehicles on the merging lane under the “With signal” case and go to an unexpected height under “Baseline” case.

### 4.3.4 Hard and soft safe time-gap replacement

Method of replacing the safe time gap is crucial and will influence following conditions. We have introduced two methods (hard and soft) shown in the last section for demonstrating the difference. In this section, we conduct two numerical tests for both methods and evaluate their performances.

![Graph showing hard and soft replacement for safe time gap](image)

**Fig. 4.9.** Hard and soft replacement for safe time gap
The first difference illustrated in Fig. 4.9 is changing the amplitude of traveling condition. In the zoom in the window of Fig. 4.9 (a) hard replacement, the CAV changes its following strategy in a sudden with a clear transition is observed. This would result in an uncomfortable traveling experience. On the contrary, CAVs’ transition is smooth with minimal disturbances. On the other hand, the hard replacement easily forms traffic oscillations starting from the location close to merging point due to the fixed safe gap configuration.

### 4.3.5 Location of the buffer zone

Location selection for the buffer zone is also important and affects overall performance. Three representative tests are conducted for this purpose and the test configurations for location selection are described in Table 4.5.

<table>
<thead>
<tr>
<th>Case</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100 m</td>
<td>0 m</td>
</tr>
<tr>
<td>2</td>
<td>900 m</td>
<td>800 m</td>
</tr>
<tr>
<td>3</td>
<td>900 m</td>
<td>100 m</td>
</tr>
</tbody>
</table>

Note: the start and end of the buffer zone are measured from merging point in this section.

These three cases represent three situations regarding buffer zone at near end, far end, and middle section. Showing in Fig. 4.10 (a) is the near end buffer zone, which allows CAVs to change their status in a short amount of time and results in traffic oscillations. A far end buffer zone shown in Fig. 4.10 (b) eliminates traffic oscillations, whereas the zoom in the window shows a gap, used to be created for
merging maneuvers, vanishes or disappears due to a back-propagated disturbance in front, even the disturbance is insignificant. Case 3 solves the problems found in Cases 1 and 2. A longer buffer zone in Fig. 4.10 (c) gives CAVs longer time to prepare and they can distribute their changes evenly during this preparation time. As CAVs continuously adjusting their gap in an adequate period based on the cooperative strategy, there will always be a sufficient gap for merging and the problem in Case 2 no longer exists.
Fig. 4.10. Trajectory result of different buffer zone locations
4.4 Conclusions

Connected and automated vehicles offer new opportunities for smoothing highway traffic flow where priority junctions are presented. In this study, a cooperative scheme has been proposed and tested under simulations.

Analyses have been carried out to evaluate the performances in aspects of efficiency, safety and fuel consumption. The result indicates that a baseline with only MVs under heavy traffic introduces difficulty in merging. The metering method as a widely accepted solution has been evaluated as well. However, redistributing traffic flow using metering introduces periodical traffic oscillation on highway mainlines with only a slight performance improved in merging capacity.

An IDM-based cooperative driving strategy for CAVs are proposed and investigated in this study. Results show efficiency and merging capacity boost with CAVs mixed in traffic flow and the benefit can be enhanced by increasing percentage of CAVs. Specifically, the problem of queue accumulation on merging lane under heavy traffic flow is solved by CAVs. Namely, traffic flow combined with CAVs discharges queue faster with only a slightly negative impact on mainline. Furthermore, the locations of the buffer zone and parameter settings have been validated for CAVs. This study shows a soft parameter replacement, together with a wide range of buffer zone improves the overall performance and reduce the occurrence of traffic oscillations.

This study also draws some interesting points when considering modifying or redeveloping a car-following model based on an existing car-following model. Firstly, the way that two or more vehicles cooperated should be taken into account. For example, the follow up vehicles in the waiting queue generally have less
follow-up time during merging, we want the merging vehicle to form a group before merging in order to increase the discharging rate. Therefore, the problem about “what would be the discharging threshold in cooperative mechanism?” should be concerned when designing the cooperative model. Secondly, which region should be designed as the buffer zone? From our result, the carefully selected buffer zone will improve the performance and merging capacity. To ask these questions would help designers who want to design the next generation of CAVs.
Chapter 5

Dampen Traffic Oscillations: A Reinforcement Learning Based Approach

5.1 Introduction

Automated vehicles aims to replace a human driver with a robot that constantly receives environmental information via various sensor technologies (as compared to human eyes and ears) (Xu et al., 2012), and consequently determines vehicle control decisions with proper computer algorithms (as compared to human brains) and vehicle control mechanics (Zhou et al., 2017b). The development of connected and automated vehicles (CAVs) has far outpaced the existing traffic control systems in that individual vehicle can be controlled and regulated in a real-time manner (Ma et al., 2017). With these CAVs, individual vehicle-based control becomes feasible in fully overcoming or minimizing the negative effects caused by human driver’s limit, heterogeneity, and non-cooperativeness (Schakel et al., 2010, Zohdy and Rakha, 2016, Milanés and Shladover, 2016). In other words, the traffic flow management can be transformed from a reactive, aggregated/collective, and non-cooperative infrastructure-based to a proactive, disaggregated/individual, and cooperative vehicle-based paradigm (Zhou et al., 2017b).

A number of studies have been developed with an attempt to modify and improve classical models for CAVs (Milanes and Shladover, 2014, Wang et al., 2013, Treiber and Kesting, 2013b, Kesting et al., 2010, Zohdy and Rakha, 2016, Yu and
Shi, 2015, Wang et al., 2014a, Milanes et al., 2014, Zhou et al., 2017b). These models have yielded abundant knowledge and control methods in understanding and utilizing this emerging technology in highway traffic management. However, as most of these models were developed primarily based on human-behavioral theories without any room for self-learning and self-correction, they have limited flexibility and adaptivity, and further modifications and improvements are likely constrained by certain empirical equations.

Machine learning has been widely used in transportation research, such as using artificial neural networks to mimic human driving behaviors (Aghabayk et al., 2014, Zhang and Ge, 2013, Mathew and Ravishankar, 2012, Khodayari et al., 2012, Chong et al., 2011, Panwai and Dia, 2007, Jia et al., 2003, Morton et al., 2017, Zhou et al., 2017c). However, if CAV driving strategies were only constrained by human driving paradigms, it would be hard for CAVs to overcome the intrinsic limitations of human drivers (e.g., proneness to errors, low reaction time, unwillingness to collaborate, etc.), which have been widely criticized as causes to prevailing traffic issues (Guériau et al., 2016, Fajardo et al., 2011, Bagloee et al., 2016), and are arguably the challenges that the masterminds behind the concepts of CAV planned to overcome (Qualcomm Technologies, 2017, Office of the Assistant Secretary for Research and Technology (OST-R), 2017). Therefore, appropriate driving strategies beyond the human driving framework need to be designed in order to realize the full vision of the future CAV traffic.

Traffic oscillations refer to the stop-and-go driving conditions in congested traffic which typically form bottlenecks of transport infrastructure [32, 38]. With regard to controlling CAVs in terms of dampening or eliminating traffic oscillations, one approach is to create sufficient time buffer or shorten the responding time in traffic oscillations and stabilize overall traffic flow (Stern et al., 2017, Cui et al., 2017).
This idea has also been validated by field experiments (Milanes et al., 2014, Milanes and Shladover, 2014, Milanés and Shladover, 2016). Another approach to eliminating traffic oscillations is to access a future target state to make a driving plan from the current state onward. Ma et al. (2017) developed a trajectory design model to eliminate traffic oscillation by optimizing the motion of CAVs backward from a target driving state in the future. However, this methodology only optimizes the current and future motions of vehicles by taking advantage of the infrastructure communication to vehicle and awareness of a future target state. While in most cases, there is no target state or it is difficult to obtain one, the CAVs are limited to make a plan in advance, and the current state matters more in terms of decision-making. Moreover, it is expected that planning methods will need much computational time, as it has to search into the future. Differently, the study in this study focuses more on obtaining a general CAV model that considers current driving information only. A Reinforcement Learning (RL) approach is applied for this purpose.

A recent breakthrough in RL challenges human in gaming disciplines (Silver et al., 2016, Mnih et al., 2015, Silver et al., 2017). RL is capable of generating appropriate rules to achieve a certain goal without human supervision. Additionally, RL generated solutions may come from a great number of search attempts in a solution space while human may only be capable of accessing a subset of this space. In this regard, the RL approach can overcome human limitations. As such, a properly designed RL based model can be an ideal alternative for the design of CAV driving strategies. Zhou and Qu (2016) applied one of the RL methods named Deep Q-Network (DQN) (Mnih et al., 2015, Mnih et al., 2013) and Desjardins and Chaib-Draa (2011) applied Policy Gradient (PG) (Williams, 1992, Williams, 1988) to design a CAV driving controller. The result shows a specific driving strategy can be learned by providing an appropriate reward-guided system. However, the learned
CAV acceleration controller only works in a discrete action space due to the fact that a discrete action function approximation can simplify the possible action outputs. However, it is impractical that the CAV can only drive with a discrete acceleration rate. Further, discretized actions in RL require outputting multiple values and choosing the action with the maximum value. When including a great deal of discretized actions, it consumes more computational time compared with outputting a single but continuous action. Therefore, in this study, we extend aforementioned studies to a continuous action space, which resolves the demerits of these models and is more suitable for real-time control of CAVs. To the best of our knowledge, there is no research study on using RL to develop car-following controllers for CAVs in a continuous acceleration space. In this research, we will bridge this gap by applying Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015) in the learning process. Furthermore, in addition to learning from the prior experiences, the designed CAV controller shall be able to mitigate or eliminate current traffic problems, particularly on traffic oscillations or stop-and-go traffic.

In this study, we define a novel reward-guided system for CAVs to learn and maximize its accumulative reward. After the training process, we find that the RL-based CAVs are able to dampen the effect of a sudden traffic disturbance. We further conduct simulations based on the proposed RL technique under traffic oscillations. A significant improvement is observed in the performances of transport systems as a whole.

This rest of this chapter is organized as follows. Section 5.2 introduces the proposed RL based methodology. Section 5.3 shows the experimental results and Section 5.4 concludes.
5.2 Experimental design

5.2.1 Training environment and parameter settings

The training procedure is demonstrated in Fig. 5.1 as CAVs can interact with the driving environment and simultaneously collect all traveling information, including their own speed, gap distance and the relative speed with their preceding vehicle. The traveling information is stored in a memory buffer in the RL (DDPG) system. A batch of experiences randomly sampled from this memory buffer is used to update the actor and the critic at each time step. The actor is responsible for choosing an appropriate acceleration for all CAVs and the inputs and output of the actor can be simplified as

\[ a \leftarrow Actor(\Delta v, \Delta x, v) \],

where \( \Delta v \) and \( \Delta x \) denote relative speed and distance gap with its preceding vehicle respectively; \( v \) denotes its own speed.
Note that there is only one actor in this system, so that all CAVs share the same DDPG model, and each of them contributes equally to the system update. A virtual environment is shown in Fig. 5.2 is built based on this training system. We choose 10-car platoon in order to simulate the oscillation effect in a consecutive traffic flow. The detailed setups regarding the physical property of CAVs are listed as follows.

- Acceleration range: \([-5m/s^2, 3m/s^2]\);
- Updating and responding interval: 0.1 seconds; and
- Car length: 5 meters.

The CAVs are trained on 2,000 episodes, each of which consists of 300 time steps. Each time step is equivalent to 0.1 second updating interval. We initialize each episode with randomness in order to reduce the sensitivity of the final model and the initialization setups are described as follows.

- Fix the initial speed of leading vehicle in each episode: 100km/h;
- Initial distance gaps for subsequent vehicles are randomly selected in a range of \([15m, 50m]\); and
- Initial speed for subsequent vehicles is randomly selected in a range of \([40km/h, 130km/h]\).
Fig. 5.2. Vehicles follow each other in a circular driving loop. Note: the numbers next to vehicles refer to as “speed/reward” and these numbers are colored by the reward. A higher reward turns to blue and a lower one turns to red.

The leading vehicle is not allowed to accelerate or decelerate during the whole episode, while other vehicles in this 10-car platoon adjust their acceleration rate by the actor in the DDPG model. The final goal for each episode is to stabilize the following condition from a disordered initialization. This is an indirect but faster
training procedure for CAVs to learn how to handle traffic oscillations compared with randomly disturbing the platoon’s leader.

The hyper-parameters for the DDPG model are carefully selected. We select $\alpha_c = 1e-5$ and $\alpha_o = 2e-5$ in Equation (2.9) as the learning rates for critic and actor. The discount factor $\gamma = 0.9$ in Equation (2.9). In order to cover the experience in several episodes, we choose the memory capacity as 100,000 transitions. The update frequency for target networks $Q^\bar{\pi}$ and $\mu^\bar{\pi}$ are selected as 1,000 time steps. The evaluation networks $Q^\nu$ and $\mu^\nu$ are updated each step using RMSprop (Tieleman et al., 2012) with a batch size of 64.

5.2.2 Reward function

In practice, it is common to carefully design a reward function to encourage a particular solution. In a car-following problem, without loss of generality, we apply a headway-based reward function such as the one mentioned in (Desjardins and Chaib-Draa, 2011, Zhou and Qu, 2016). By adopting a headway-based reward function, one can easily learn a car-following pattern. With the learned driving rule, a CAV can drive with collision-free behavior, which, however, may not also result in an increment of travel efficiency. This makes us to reconsider and to design a reward function from the perspective of optimizing traffic flow dynamics.
The headway is essential for a safety concern, but the time gap shown in Fig. 5.3 is more crucial which excludes the impact of preceding vehicle’s length. Therefore, in the following section, we adopt time gap instead of time headway as a part of the reward function illustrated in Fig. 5.4.

The reward function consists of two aspects. On the left of Fig. 5.4, we define a maximum speed of 110 km/h. In a homogeneous traffic follow, there is no doubt that all vehicles traveling with a higher speed increases efficiency. Further, in a traffic oscillation, a vehicle stabilizing its condition while maintaining a higher speed should be rewarded. Therefore, within the range of 0 km/h to 110 km/h, the reward is monotonically increasing from 0 to 1. In this research, we test three types of monotonic reward curve including “Linear”, “Concave” and “Convex” and results can be found in the next section. Wherever the speed exceeds the maximum speed, a reward of negative one is assigned as a punishment. Further, whenever the time gap is less than a minimum safe time-gap (here we choose 0.6 seconds), a reward of negative one is given to reduce the risk of collision. The value 0.6 s is
found in (Milanes and Shladover, 2014) as they conducted a road test about CAVs and found the shortest time gap to be 0.6 s.

![Graph showing the design of reward function](image)

Fig. 5.4. The design of reward function

5.3 Results and discussion

5.3.1 Training result

In order to validate the robustness of the DDPG model, we conduct six epochs with different random seeds and plot the moving averaged episode reward in Fig. 5.5. In a typical RL training, the reward for each episode can be noisy. Therefore, we apply a moving averaging method to show the tendency of the total reward change. The moving averaged episode reward is computed as
where $R_t$ denotes the reward at episode $t$.

The accumulated reward in an episode grows up quickly from the beginning of training. The linear reward function achieves 120 episode-rewards faster than other reward curves. The concave reward curve introduces higher variance in training and has about a half of the episode-rewards at the end of training compared with others.

In conclusion, both linear and convex reward curves are acceptable for training.
CAVs and the linear curve has the fastest convergence. Therefore, in the rest of the tests, we select the DDPG model trained by the linear reward-function.

5.3.2 Comparing with manually-driven vehicles (MVs)

Many car-following models have been developed for modeling MVs. The IDM (Treiber et al., 2000) is one of the classical car-following models that has been intensively studied (Kesting et al., 2010, Treiber et al., 2007b, Treiber et al., 2007a, Treiber et al., 2006, Milanès and Shladover, 2016). Therefore, in the following sections, we compare our trained CAVs with MVs (modeled by the IDM) in terms of travel efficiency in various circumstances. The default IDM parameters (Treiber et al., 2000) are used for the rest of evaluations as they perform well not only under free-flow but congested flow traffic (Zhou et al., 2017c, Zhou et al., 2017b).

5.3.2.1 Traveling with high speed

This test is to simulate the performance between MVs and CAVs in a disturbance when they traveling with a high speed. We fix the leading vehicle’s trajectory by a sequence of acceleration patterns:

• constant speed (100km/h) for 100 seconds;
• decelerate (−2m/s²) for 15 seconds (if the speed is decreased to zero, vehicle stops and the deceleration rate is set to zero);
• accelerate (1m/s²) to the original speed (100km/h);
• constant speed (100km/h) for 200 seconds.

Following vehicles are generated with a uniformly 2 seconds headway and 100km/h initial speed.
Fig. 5.6. Traffic oscillation when traveling under a high-speed setup. Note: The color in (a) and (b) demonstrates vehicle’s speed, red and blue means low and high speed respectively.

We plot the simulated trajectories and the traveling details of the leading vehicle, the first and last follower from CAVs and MVs respectively in Fig. 5.6. From the trajectory results, it is clear that the disturbance caused by the leading vehicle creates a chain reaction in both CAV and MV platoon. However, in the MV’s platoon, it is
shown that an obvious propagative oscillation throughout the whole platoon. In contrast, in the CAV’s platoon, the disturbance quickly dissipates and the oscillation gradually disappears. The acceleration, speed and time gap details draw the same conclusion. Additionally, the acceleration and speed details indicate that the first CAV follower is more responsive to the changes in the following condition, which result in a faster stabilization. The time gap results indicate that a CAV tends to maintain a time gap less than an MV does. We quantify the travel efficiency of this test in Table 5.1.

5.3.2.2 Traveling with low speed

We train the model for stabilizing traffic flow within a high-speed condition. It is worth testing the performance under a low-speed condition to evaluate the robustness of the trained model. We run another test by adopting the same testing configuration as the last section but with 40km/h initial speed for all vehicles.
Fig. 5.7. Traffic oscillation when traveling under a low-speed setup. Note: The color in (a) and (b) demonstrates vehicle’s speed, red and blue means low and high speed respectively.

Although the overall speed condition differs from the last test, the results shown in Fig. 7 draw the same conclusion as it is in the last test. We also compared the details of travel efficiency in Table 5.1.
Table 5.1. Travel efficiency comparison between MVs and CAVs

<table>
<thead>
<tr>
<th>Test</th>
<th>Vehicle type</th>
<th>Average travel time (min/km)</th>
<th>Time mean speed (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High speed</td>
<td>CAV</td>
<td>0.64</td>
<td>94.08</td>
</tr>
<tr>
<td></td>
<td>MV</td>
<td>0.69</td>
<td>87.60</td>
</tr>
<tr>
<td>Low speed</td>
<td>CAV</td>
<td>1.55</td>
<td>38.73</td>
</tr>
<tr>
<td></td>
<td>MV</td>
<td>1.62</td>
<td>37.05</td>
</tr>
<tr>
<td>Stopping</td>
<td>CAV</td>
<td>0.69</td>
<td>86.90</td>
</tr>
<tr>
<td></td>
<td>MV</td>
<td>0.79</td>
<td>75.93</td>
</tr>
</tbody>
</table>

5.3.2.3 Stop-and-go scenario

Traffic oscillations often consist of a series of stop-and-go phases. In this section, we evaluate the trained model under a long stopping phase followed by an acceleration phase. The trajectory of the first vehicle is fixed by:

- constant speed (100km/h) for 3 seconds;
- decelerate (−4m/s²) for 30 seconds (if the speed decreases to zero, vehicle stops and the deceleration rate is set to zero);
- accelerate (2m/s²) to the original speed (100km/h);
- constant speed (100km/h) for 200 seconds.

Fifty vehicles are initialized by the same method as last two tests.
From the result shown in Fig. 5.8, the trained DDPG model successfully eliminates the stopping effect. In contrast, traffic oscillations of MVs propagate to the last vehicle.
Fig. 5.9. Buffer time and distance for CAVs and MVs. Note: The color in (a) and (b) demonstrates vehicle’s speed, red and blue means low and high speed respectively.

Considering smoothing traffic oscillations through a ramp metering or traffic light control on this oscillated flow, the number of time intervals needed (Fig. 5.9) for the next vehicle are 34.4 seconds and 72.8 seconds for CAVs and MVs respectively. If it is measured by distance, the buffer distances are 955.3 meters and 2023.0 meters, respectively, which indicate an over 200% efficiency improvement for an oscillation-free CAVs followed by DDPG-based CAVs than MVs.
We also conduct a comparative analysis of efficiencies under the stopping effect in Table 5.1 and Fig. 5.10. We scale down the values in this table based on the MVs performance and use the performance of MVs under different cases as a baseline. The result indicates that when involving a great jump of speed such as in high speed and stopping cases, CAVs tend to be 105% to 120% more efficient. When considering only the last following vehicle in the platoon, we observe a higher improvement in traveling efficiency. This is caused by a greater negative impact of oscillation that propagates to the last vehicle. In the CAVs platoon, the following CAVs absorb a higher amount of oscillation so that the last CAV can travel almost freely. Oppositely, MV’s platoon cannot handle the oscillation, hence showing a limited performance. Note that this result includes only one-off traffic disturbance, a greater efficiency improvement will be expected on a congested road.
5.3.3 Mixed traffic flow

There is no doubt that CAVs will soon share roads with MVs. As such, we test not only the pure CAVs conditions but also in a mixed traffic flow of CAVs and MVs. As can be seen in Figs 11 and 12, with an increase of CAV’s penetration rate, the mixed flow can better accommodate traffic disturbances.

![Diagram](image)

Fig. 5.11. Trajectory comparison under different CAV penetration rates. Note: The color in (a) and (b) demonstrates vehicle’s speed, red and blue means low and high speed respectively.
Table 5.2 presents performances with respect to different penetration rates. Note the column “Efficiency improvement” represents the percentage of an average speed increment based on 0% CAV.

Table 5.2. Average results over all vehicles in traffic oscillations.

<table>
<thead>
<tr>
<th>AV rate</th>
<th>Average speed (km/h)</th>
<th>Average travel distance (km)</th>
<th>Average travel time (min/km)</th>
<th>Efficiency improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>71.12</td>
<td>2.77</td>
<td>0.85</td>
<td>0.00%</td>
</tr>
<tr>
<td>20%</td>
<td>74.59</td>
<td>2.91</td>
<td>0.81</td>
<td>4.89%</td>
</tr>
<tr>
<td>40%</td>
<td>76.42</td>
<td>2.98</td>
<td>0.78</td>
<td>7.45%</td>
</tr>
<tr>
<td>60%</td>
<td>79.11</td>
<td>3.09</td>
<td>0.76</td>
<td>11.23%</td>
</tr>
<tr>
<td>80%</td>
<td>82.49</td>
<td>3.22</td>
<td>0.73</td>
<td>15.99%</td>
</tr>
<tr>
<td>100%</td>
<td>87.78</td>
<td>3.43</td>
<td>0.68</td>
<td>23.43%</td>
</tr>
</tbody>
</table>
5.4 Conclusions

CAVs can not only free human drivers from driving, but also be considered as an optimization tool for improving traffic operations such as dampening or even eliminating traffic oscillations.

The design of CAVs should focus on not only levels of automation from the perspective of vehicle manufacturers but also efficient traffic operations from the perspective of transport managers. This study provides an alternative for CAV’s controller design other than following the template of classical car-following models. In doing so, the CAV’s driving rules are no longer constrained by a physical formwork in classical car-following models. Instead, it explores the possible combination of the reasoning relationship, using trial and error method, experiencing the transitions far more than any human can do. The final CAV model by using RL approach is possibly a solution beyond human’s knowledge and empirical equations.

The key to getting a well-performed RL behavioral model is the reward function design and the feature selection. We carefully developed the reward-guided system by considering travel efficiency and safety, in particular, to reward CAVs that maintain high speed and keep a following distance that greater than the minimum safety gap. In the feature selection, we continue to use the conventional features including relevant speed, gap distance, and its’ own velocity. Then the RL-based CAVs can learn a behavioral model to maximize the future return according to this reward signal and state as a feedback from the environment.

By generating an appropriate CAV driving strategy using RL approach, the CAVs can learn to drive not only automatically but also improve traffic efficiency of the transport infrastructure as a whole.
We evaluate the learned model under several scenarios to validate its performance. The model can be applied in a dense traffic flow traveling with both slow speed and high-speed traffic flow. The trajectory result implies that a sudden change in leading conditions can trigger an unavoidable traffic oscillation and this oscillation quickly propagated to upstream traffic in pure MV flow. In comparison, CAVs, although experience the same disturbance, show an absorption property in the traffic oscillation, preventing the oscillation propagating further to upstream traffic. Additionally, we tested a critical scenario of stop-and-go traffic. The leader decelerates to stop quickly then accelerates to its original speed. The result from this scenario reveals the learning model can still overcome the negative impact, which presents a strong robustness of the model.

Lastly, the implementation under mixed traffic flow illustrates that the increase in the proportion of CAVs improves oscillation absorption and more following vehicles will benefit from this to travel with free speed.
Chapter 6

A Reinforcement Learning Based Approach to An Efficient Driving Strategy at Signalized Intersections

6.1 Introduction

Previously, many of the control methods for Connected and Automated Vehicle (CAV) were developed from one of the existing classical car-following models, and empirical examinations were conducted to validate them (Milanes et al., 2014, Milanes and Shladover, 2014). However, when driving through signalized intersections, the mechanism in driving is fundamentally different between human drivers and CAVs because the resources they use are varied significantly. This can shift driving behavior of a CAV to an entirely different direction. As the same as in Chapter 5, therefore, the autonomous vehicle controller is not necessarily to be constrained by classical model-based approaches, the model of which has only been validated on human drivers. Therefore, new methodologies for developing CAV’s controller are required in road sections of signalized intersection.

Regarding trajectory optimization at intersections, where a road segment CAVs can take advantages of, Ma et al. (2017) proposed a CAV’s controller that pre-designs a smooth trajectory to encourage CAVs to arrive at an intersection only during green phase to avoid fully stop during red phase. However, this method computes the smoothed trajectories by given a known final status at an intersection and runs an optimization procedure repeatedly. It is no doubt that this method is suitable for making an optimized plan in a static environment. However, in a case of needing
dynamic adjustment to one’s trajectory, or whose final status is not accessible, this method may not be effective.

Meanwhile, the state-of-art machine learning gains enormous attention in many research areas. Reinforcement Learning (RL), a branch of this family, has achieved many successes in such as playing video game (Mnih et al., 2015) and board game Go (Silver et al., 2017, Silver et al., 2016) against human players.

Compared with re-developing a classical car-following model for CAVs for intersection optimization, RL iteratively tries enormous action combinations, experiences a number of trajectories far more than any human can do, and breaks the empirical constraint of a classical model. Additionally, applying RL, instead of the trajectory optimization mentioned in Ma et al. (2017), solves the computational issue caused by re-optimizing when changes occur in surroundings. Namely, RL exploits a learned behavioral model in real-time to mitigate re-planning overhead. This study bridges the gap towards a real-time CAV controller at road segments with signalized intersection by proposing an RL-based learning method.

In this study, we select Deep Deterministic Policy Gradient (DDPG) algorithm (Lillicrap et al., 2015) and define a meaningful reward-guided system for CAVs to form an appropriate behavior. Results show a regularized and robust control model is learned capable of being implemented under various traffic demands and different traffic light cycles. Compared with oscillated trajectories triggered by human drivers, CAVs’ trajectories indicate a significant improvement in the travel efficiency and the traffic oscillation can be largely eliminated. Additionally, it shows that one single behavioral model can run in a dynamic system in real-time and eliminate the effort of re-doing optimization process for each trajectory.
This rest of this study is organized as follows. Section 6.2 introduces the proposed RL based methodology. Section 6.3 shows the experimental results and Section 6.4 concludes.

6.2 Training environment and parameter settings

![Learning diagram](image)

Fig. 6.1. Learning diagram

Reinforcement learning based CAV learns a behavioral model from their interaction with the environment. Shown in Fig. 6.1 is a learning flowchart we use for training the CAVs. Every CAV collects observation data, stores that into the memory buffer where a mini-batch of data used in training will be sampled. The DDPG model applies those mini-batch data to update the critic and actor aiming to achieve a higher accumulated reward or a better performance in the environment. In this training framework, however, all CAVs are controlled by a centralized actor, which
means all CAVs perform actions only based on the same model parameters. This is similar to the typical car-following model in terms of calculating the output.

Fig. 6.2. Circular training environment with CAVs generated from the center origin and removed at the end of the white line. The green square can turn red and green to represent the traffic light.
A virtual environment shown in Fig. 6.2 is built for this training system. In order to train in this environment, assumptions made for CAVs are specified as follows.

- Updating and responding interval: 0.5 seconds (conventional 0.1 seconds interval works worse in the experiment as the action made with 0.5 seconds interval takes less number of iterations toward a future event, thus the back-propagation of the future impact becomes more effective in this case);
- Car length: 5 meters; and
- Acceleration bound: $[-4m/s^2, 2m/s^2]$.

We iteratively train the model for 250,000 steps. CAVs are trained on different episodes. In each episode, the light cycle and traffic flow conditions are randomly initialized based on a pre-defined range. This randomization can effectively diversify the experience used in training and increase the model robustness under different traffic conditions. Additional assumptions listed below for the training environment make effect globally.

- Traffic light at 1500 meters from the start point;
- Maximum episode step: 760;
- Light cycle is divided into yellow, red and green phases. The yellow phase as a buffer time remains unchanged (5 seconds), while the red and green phase can be randomly initialized within a range from 15 and 60 seconds for each episode; and
- Traffic demand, for each episode, at the initialization point can vary from 720 to 1800 veh/h indicating a headway range of 5 to 2 seconds (The critical traffic flow at the intersection can reach to about 3600 veh/h).

The hyper-parameters for the DDPG model are carefully selected. We choose the learning rates for critic and actor as $\alpha_c = \alpha_a = 0.0001$ and the discount factor
\( \gamma = 0.9 \) in Equation (2.9). In order to cover the experience in several episodes, we choose the memory capacity as 500,000 transitions to encourage well-distributed data. The update strength \( \tau_w \) and \( \tau_\theta \) for target networks \( Q^\tilde{\omega} \) and \( \mu^\tilde{\theta} \) are selected as 0.001 in Equation (2.10). The evaluation networks \( Q^\omega \) and \( \mu^\theta \) are updated each step using RMSprop (Tieleman et al., 2012) with a batch size of 64. The actor and critic both contain three-layer neural network with 128 neurons for each layer.

### 6.2.1 Input features

Features are crucial in the training process. Although RL can learn from raw or unprocessed features, e.g., images (He et al., 2015a, Girshick et al., 2014). A well-designed feature representation can maximize the learning efficiency. Feature selection is to find the observation terms to best correlate with our final goal. Feature engineering is necessary for many RL implementations in order to achieve fast convergence and precise control (Heess et al., 2017). In this study, we include the basic features adopted in conventional car-following models such as relative speed, gap distance, and vehicle speed. Additionally, the traffic light plays a key role in this setup. Therefore, we summarize some representative features in Table 4.1 and Fig. 6.3.
Fig. 6.3. Illustration of the intersection scenario

Table 6.1. Features selected for the RL model

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>Vehicle velocity (m/s)</td>
</tr>
<tr>
<td>$\Delta v$</td>
<td>Relevant speed to the preceding vehicle (m/s)</td>
</tr>
<tr>
<td>$\Delta x$</td>
<td>Gap distance to the preceding vehicle (m)</td>
</tr>
<tr>
<td>$l_d$</td>
<td>Distance to light (m)</td>
</tr>
<tr>
<td>$l_t$</td>
<td>Instant travel time to light (s)</td>
</tr>
<tr>
<td>$T_l$</td>
<td>Light cycle</td>
</tr>
<tr>
<td>$T'_l$</td>
<td>Derivative of light cycle</td>
</tr>
</tbody>
</table>
With three local and four global features, any CAV can understand its following condition regarding its preceding vehicle and relationship with the traffic light. The light cycle $T_i$ and its derivative term are to describe the current light state and time point in a continuous light cycle. We can visualize these two features in Fig. 6.4. If and only if providing $T_i$, it would not be sufficient to describe the time point at this light cycle. For instance, the value of zero points to two different times at the figure (one at the green phase and another at red). While by providing with the derivative term ($\dot{T}_i$) in addition to $T_i$, the combination of these two features precisely presents the property of the light cycle.
Fig. 6.4. Design of the feature representation for light cycle and its derivative term

Note that the future value based on light cycle is a demonstration, the duration of red and green phases can vary in the application.

The equation for light cycle considers the difference in cycle duration and is written as follows
where $t_r$, $t_y$ and $t_g$ represent the duration of red, yellow and green phases respectively, $t$ denotes the time step start either from the beginning of green phase or yellow phase. For instance, in Fig. 6.4, when the green, yellow and red durations are 40 s, 5 s and 40 s respectively, then $t = 20$ when at 20 s and 60 s. The derivative term $T'_i$ is written as follows

$$T'_i = \alpha_i \frac{dT_i}{dt}$$

$$= -\alpha_i \sin(map) \times \begin{cases} \frac{\pi}{t_r + t_y}, & \text{when not at green phase} \\ \frac{\pi}{t_g}, & \text{when at green phase} \end{cases}$$

where $\alpha_i$ denotes a scale factor used to normalize this feature. We apply scale factors for other features as well in order to normalize them into a similar order of magnitude before fitting into the neural network. The training of the neural network, therefore, is no longer sensitive to any of the features and a better convergence will be achieved.
6.2.2 Reward function

In practice, it is common to carefully hand-design the reward function to encourage a particular solution. Shown in Fig. 6.5 is a handcraft reward function presenting vehicle’s condition. This reward function is threefold consisting of aspects in velocity, time gap, and predicted arrival time at the intersection.

![Graph showing reward function](image)

Fig. 6.5. Reward function (dash line indicates an aggregated reward while solid line represents an absolute reward)

*Note that the reward function based on light cycle is a demonstration, the duration of red and green phases can vary in the application.*

Information such as vehicle velocity and time gap belongs to the local condition that shapes its driving behaviors locally. An interaction between a vehicle and traffic light forms a global condition, which should be distinguished from the interaction between surrounding vehicles. At a signalized intersection, both local and global
conditions are necessary and they work together to build a foundation of this reward function.

Specifically, we initialize the reward to zero for each step, and then aggregate other reward signals (dash line in Fig. 6.5) into it. Driving with higher velocity indicates less travel time, which should contribute to the reward (with a maximum of +1). However, punishment must be made to avoid speeding, so negative reward of -1 is given in this situation. Additionally, -1 is aggregated to the reward if the following vehicle maintains a time gap shorter than 0.8s to its leader. For a safety reason, the value 0.8s is selected slightly greater than the minimum value (0.6s) for CAVs found in (Milanes and Shladover, 2014). Lastly, if a vehicle predicted to arrive the intersection on a red phase based on its current speed and distance to the intersection, a -1 signal will be assigned (solid line) to fully replace the aggregated reward. Otherwise, when a CAV finds that it will pass through the green phase, it will accumulate +0.1 to the original reward. To emphasize the benefit of passing green phase and minimize the outliners, we include a buffer time of three seconds attached at the start and end of the green phase to re-distribute the traffic flow. Additionally, the yellow phase has the same effect as the red phase to be considered as another buffer to reduce outliners. Note that we use a fixed five seconds for yellow phase and three seconds start and end buffers for green phase, let the duration of green and red phases free in all experiments.
6.3 Results and discussion

6.3.1 Training result

In order to validate the robustness of the DDPG model, we conduct six epochs with different random seeds and plot the moving averaged episodic reward in Fig. 6.6. In a typical RL training, the reward for each episode can be noisy. Therefore, we apply a moving averaging method to show the tendency of the total reward change. The moving averaged episodic reward is computed as

\[ R_t \leftarrow 0.8R_{t-1} + 0.2R_t \]

where \( R_t \) denotes the reward at episode \( t \).
The accumulative reward in an episode grows up quickly from the beginning of training. Especially at around 80,000 steps, the episodic reward grows up rapidly. After 100,000 steps, the model tends to converge at the value of 400. We randomly select one of the trained models and conduct several tests in the following sections.

### 6.3.2 Different traffic demands

A model should serve a general purpose and be exploited under different traffic demands. We conduct this test to evaluate the model’s performance by giving three
different traffic demands. In addition, a dynamic demand is also evaluated to simulate a short-term effect.

Firstly, three simulated cases, including 600, 900 and 1200 veh/h, are built using a light cycle of 40 s for red and green phases and 5 s for yellow phase. The trajectory result is shown in Fig. 6.7 indicating that CAV trajectories are re-distributed in order to minimize the stopping effect or traffic oscillation. CAVs receive signals from both traffic light and their preceding vehicle. While handling the local changes between preceding vehicles, it also considers the global changes of the traffic signal, which can be seen as the CAV learns to take a gentle deceleration in advance for the sake of a long-term benefit. This learned behavior is shaped by the reward function we designed as we expect CAVs to maximize their speed while still go through the green phase.
Fig. 6.7. CAVs trajectory result for different traffic demands
The re-concentration and grouping maneuver stabilized the traffic flow. The flow capacity nearly doubled at the intersection where requires CAV switches to a shorter headway to pass through. The worst scenario is demonstrated in (c) which the maximum traffic flow can reach to approximately 2400 veh/h at the intersection, which can be classified as traffic congestion. The figure also indicates that each platoon leader can trigger a disturbance and its impact propagates to followers in the front of the platoon. With the decrease in the demand, this propagated impact is reduced and vehicles at the end of the platoon can actually travel with free-speed.

Fig. 6.8. CAVs average speed comparison for different traffic demands

Fig. 6.8 shows a view from a different perspective. The average speed follows a circular pattern according to the light cycle. The average speed peaks at the middle of both red and green phases, which reveals that CAVs can learn a special strategy to improve the efficiency. The figure on the right is the average speed for each
scenario, which indicates that the average speed can only be slightly affected by the
different demands. In other words, travel efficiency remains similar no matter how
the traffic volume changes in the range we specified.

Traffic demand can vary during a short period. Fig. 6.9 shows a gradually increased
traffic flow. As the conclusion drawn above, unlike the pre-defined method in Zhou
et al. (2017a), the trained CAVs can easily handle the uncertainty of demand in
real-time. Additionally, the trained model adapts the dynamic flow changes without a significant speed drop.

6.3.3 Different light cycles

Traffic light duration is not a fixed value for different intersections. Even at the same intersection, the duration of the light phase can change during a day based on the traffic demand. We highlight this duration variability in the training of the model so that the learned model can handle this kind of uncertainty. In order to validate the learned model among different light cycle scenarios, we conduct the following evaluation with a fixed 900 veh/h (about 1800 veh/h at the critical section) in various light cycles.
Fig. 6.10. CAVs trajectory result for different light cycles
The model only takes an instant input of the observation a CAV collected and output an action to perform. Unlike the method described in Ma et al. (2017), there is no plan made in advance. Therefore, without a pre-determined plan, the DDPG model takes the advantage of real-time action to adjust driving behavior in order to cope with the uncertainty in the environment. As shown in Fig. 6.10, it would require a different driving behavior for dealing with the different light cycles, whereas one DDPG model covers all the behaviors in one place.

Fig. 6.11. CAVs average speed comparison for different light cycles (G: green; Y: yellow; R: red)

Fig. 6.11 summarizes the average speed changes in those scenarios. It is clear that with the red and green durations being shortened, the travel efficiency improves. In
particular, the average traveling speed is 84 km/h with red and green phases of 25 s, nevertheless it is reduced by 16% (70 km/h) in the scenario of 45 s red and green phases. The cause of the average speed drop is revealed in the comparison shown in Fig. 6.10. The time point the leading CAV decides to reduce speed and pass the next green light is earlier in the case (a) and (b) at around 350 m compared with 800 m in (c), which means the leading CAV triggers a speed reduction to the following CAVs earlier as well. However, in terms of passing-light rate, the CAVs in all scenarios pass the intersection without stopping.

6.3.4 Comparison with manually driven vehicles

Manually driven vehicles (MVs) are recognized to limited by human errors (e.g. slow and different reaction time, limited information processing capability and non-cooperativeness) (Qu et al., 2017). More importantly, MVs cannot communicate over distance nor cooperate with others, while CAVs share the information from traffic and road infrastructure then exploit it in the decision making process. The following simulation evaluates the performance improvement by CAVs in terms of emission, safety, and efficiency.

For illustration purposes, this study uses inverse time-to-collision (iTTC) (Balas and Balas, 2006), which is based on the widely applied TTC (Kuang et al., 2017, Weng et al., 2015, Qu et al., 2014b, Qu and Meng, 2014, Meng and Qu, 2012, Kuang et al., 2015b, Kuang et al., 2015a) for measuring safety. Additionally, we apply the VT-Micro model (Ahn et al., 2002) for evaluating fuel consumption. The details of iTTC and VT-Micro can be found in Equation (4.4) and (4.5) respectively.

In the following simulation, we use the Intelligent Diver Model (IDM) proposed by Treiber et al. (2000) to simulate MVs driving behavior as it is easy to implement and
performs well not only under free-flow but congested flow with the default parameters (Zhou et al., 2017c, Zhou et al., 2017b). We extract the first 45 vehicles to compare their emission and safety under 1200 veh/h traffic flow. A serious traffic oscillation can be found in Fig. 6.12 (a) indicating the inefficient traveling caused by human divers. With all MVs are involved in oscillation, the emission reaches to the highest level just below 0.03 liter and the overall fuel consumption per car is 0.02 liter for driving through this road section, while the fuel consumption is controlled below 0.015 liters in the CAV group. A special pattern is observed in each CAV platoon. The leading CAVs produce relatively more emission than the followers do due to decelerating in advance to pass the intersection without stop, but this deceleration effect only propagates backward to a limited number of vehicles. Therefore, the last few of CAVs in that platoon do not need to modify their speed, thus consume less fuel. When comparing traveling safety based on iTTC in Fig. 6.12 (d), the overall CAVs maintain on a similar risk level. On the other hand, the stop-and-go traffic in MV flow introduces a high level of risk.
Along the time axis, the result from Fig. 6.13 presents that the traffic signal plays a role like a wall, blocking MVs from passing them. The majority of MVs have to queue at the intersection, so the average speed drops quickly from 100 km/h to only 20 km/h during the simulation. Additionally, MVs tend to maintain 2.5 seconds headway due to the delay in response of human drivers. Lastly, the large speed variance in the MVs traffic flow is another evidence of the existing of traffic oscillation. CAV overcomes those shortcomings by keeping a low-speed variance, shorter headway and maintaining relatively high traveling speed.
6.4 Conclusions

The technology of CAV can mean more than autonomous driving which is merely a replacement for human drivers. It is not only an individual that drives safely among the surroundings but also a participant in a group receiving and sharing information for global benefits. In this study, we proposed a DDPG-based model and trained CAVs from scratch to obtain an appropriate behavioral model to fulfill the requirement of being a cooperative and automated vehicle at signalized intersections.

The result shows that the trained CAVs utilize the advantage in communication between surrounding vehicles and road infrastructure to improve the travel efficiency, safety and reduce fuel consumption. A single behavioral model is learned to be applied to different traffic demands or traffic light cycles in real-time without the re-planning overhead. However, necessary assumptions about the model, such as 0.5 s updating interval, are made to encourage efficient training. Moreover, the CAV...
model is trained in an environment without any MV participant. A mixed traffic flow with CAVs and MVs has a different property, which may need a re-design of the training procedure, thus it has not been tested in this study yet. Apart from that, each CAV can learn a decentralized policy or an individual driving model to form a cooperative scheme. Multi-agent RL has potential in this field, which can be extended from this study. We will continue to study and discuss these possible implementations in our future work.
Chapter 7
Conclusion and recommendations

7.1 Conclusion

It is no doubt that the future vehicles should be safe and automated. Extend from that, vehicles should be connected to allowing instant information sharing. Therefore, the connected and automated vehicle is believed to be our next generation of vehicle. In above chapters, we explored many approaches to developing CAV’s car-following model to enable beyond human performance.

The first two studies proposed modified car-following models based on the IDM with extra formula integrated into the original IDM to engage a CAV’s functionality. To maximize the travel efficiency, the desired time gap for car following has to be shortened. Unlike human, CAV responses instantly to the following condition without delaying. The instant responding remedies the shortened time gap so that CAV can remain in a safe time distance. Further adjustment of the IDM corresponding to vehicle-to-vehicle cooperation has been applied to strengthen the cooperative effect. The results show, with the appropriate cooperative scheme being proposed, the merging capacity at the on-ramp and priority junction can be improved to a new level. Meanwhile, traffic oscillation can be greatly mitigated when only a few CAVs mixed in the traffic flow.

The simulations using these modified IDMs indicate that the increase in CAVs proportion in the market will encourage homogeneity in traffic flow. More
importantly, in the transition phase (close to merging point), a soft safe-time gap replacement performs better in the T-junction case study. It allows a gentle and smooth phase-transition during the preparation for merging, which also contributes to the mitigation of traffic oscillation. Lastly, the location of the buffer or preparation zone plays a key role in the T-junction optimization. A closed end and far end buffer zone is less likely to be affected than the middle long-distance buffer zone.

Despite the benefit that we observed from the modified IDM, the classical model itself may not be perfect in representing CAVs’ behavior due to the empirical limitations. Therefore, we explored another approach to modeling CAVs behavior through reinforcement learning. Through trial and error learning process, a behavioral model can be learned in order to map current state to the action that maximizes the future return. The learned model carries the behavior policy then be used as a car-following model to solve the issues caused by traffic oscillation.

We conducted two studies based on the reinforcement learning process. The first one, with regard to minimizing traffic oscillation triggered by the sudden change of a leading vehicle, shows that the behavioral model that CAVs learned greatly improves travel efficiency in high speed, low speed, and stopping scenarios. Further examination of the learned model under mixed traffic with MVs indicates that the reinforcement learning based approach can also achieve the state-of-art performance in mitigating traffic oscillation. The great advantage of using this approach is that we eliminate the handcrafted model developing. Instead, we can design a reward-guided system for reinforcement learning to learn an appropriate behavioral model, which simplifies the modeling process and can possibly push the model capacity to a higher level due to the parameter exploration by eliminating classical model constrains.
In the last study, we continue to explore the reinforcement learning applications in transportation optimization. The signalized intersection can be treated as another bottleneck that affects the travel efficiency due to periodical light changes. We redefined the environment and reward function based on this scenario in order to train the CAVs to obtain a real-time behavioral model that improves the efficiency and avoids forming traffic oscillation. The learned model is regularized to be capable of being implemented under different traffic demands and light cycles. Further, the property of real-time implementation enhances the model dynamic stability in smoothing trajectory. Compared with MVs simulated by the IDM, the learned CAV model performs beyond MVs in terms of energy saving, minimizing risk and efficient traveling.

Overall, we applied two major approaches, including classical model modifications and reinforcement learning, to obtain CAV’s models in a way of minimizing the negative impact from traffic oscillation. With additional terms extended to the original classical model or specific reward guided system, the CAV models created from these two approaches work as well as expected and outperform existing models in specified cases.

7.2 Limitations and recommendations for future research

Some limitations are founded in above studies. In order to complete and improve in the future, those limitations have to be addressed. Hence in this section, we list some limitations and corresponding recommendations for future research.

In Chapter 3 and 4, it has a limitation in the application with multi-lane traffic. In this study, we only consider single lane with a merging lane attached in on direction.
However, in reality, a highway is built consisting of more than one lane. The lane-change impact on those lanes is excluded for a convenience of this study whereas the lane-change behavior near merging points is frequent. For developing a more feasible and real AV/CAV model in the future, this multi-lane factor is necessary and should be taken into account.

In addition to that and as mentioned in the literature review, the model modification from a classical car-following model may also unnecessarily inherit the empirical limitations from the model itself. In spite of the better performance shown in the results, we have to ask how we can further improve it and break the inherited limitations. One way is to adopt a different modeling process like reinforcement learning allowing the self-learning and self-correcting mechanism.

However, approaches based on reinforcement learning require building a virtual environment where CAVs can interact and get feedback. In this regard, these are the necessary works for obtaining a workable RL model. For the RL model, at this stage, we only apply a centralized control similar with classical car-following models, which all CAVs have the behavioral model constructed with the same parameters. In a case that CAVs can fully use the cooperation attribute, a decentralized control mechanism is required because each CAV will have different behavior to be integrated into a cooperative unity. The multi-agent reinforcement learning is considered as a decentralized control mechanism, which can be a future study direction for CAV networking.

It is noted that the car-following model discussed above is a longitudinal model that only responsible for single lane control. The lateral control including lane changing and merging behaviors (other than on-ramp and T-junction merging) has not been discussed. In reality, both longitudinal and lateral mechanisms should be combined
in order to form a completed car-following model. Therefore, the lateral control mechanism should be extended in the future study.
Appendix 1

A Recurrent Neural Network Based Microscopic Car-Following Model to Predict Traffic Oscillation¹

Abstract

This paper proposes a recurrent neural network based microscopic traffic flow model that is able to accurately capture and predict traffic oscillation. Neural network models have gained increasing popularity in many fields and applied in modeling microscopic traffic flow dynamics due to their parameter-free and data-driven nature. We investigate the existing neural network based microscopic traffic flow models, and find out that they are generally accurate in predicting traffic flow dynamics under normal traffic operational conditions. However, they do not maintain accuracy under conditions of traffic oscillation. To bridge this research gap, we first propose four neural network based models and evaluate their applicability to predict traffic oscillation. It is found that, with an appropriate structure and objective function, the recurrent neural network based model has the capability of perfectly re-establishing traffic oscillations and distinguish drivers characteristics. We further compare the proposed model with a classical car following model (Intelligent Driver Model). Based on our case study, the proposed

¹ This section has been published as Zhou, M., Qu, X. & Li, X. 2017. A recurrent neural network based microscopic car following model to predict traffic oscillation. Transportation Research Part C: Emerging Technologies, 84, 245-264.
model outperforms the classical car following model in predicting traffic oscillations with different driver characteristics.

**Key Words:** Recurrent neural networks; traffic flow dynamics, car following; oscillation
1 Introduction

Traffic oscillation, or stop-and-go traffic, has raised many concerns to both transport practitioners and researchers. Its negative impacts include deteriorating highway mobility, increasing safety concerns, excessive fuel consumption and greenhouse gas emissions (Bilbao-Ubillos, 2008, Zheng et al., 2010, Song et al., 2013). Various car-following models have been proposed and improved in order to accurately capture the traffic oscillation. See for example Brackstone and McDonald (1999) and Saifuzzaman and Zheng (2014) for reviews.

On the other hand, machine learning techniques, especially artificial neural networks, have been widely applied in many modeling fields, such as speech recognition (Sak et al., 2014), handwriting recognition (Graves et al., 2009), and self-driving cars (Santana and Hotz, 2016). With the increases in data quantity and computational power, the machine learning approach is more efficient and accurate than ever before. Using machine learning to model car-following behaviors has also been studied intensively (Papathanasopoulou and Antoniou, 2015, He et al., 2015b, Aghabayk et al., 2014, Chong et al., 2013, Mathew and Ravishankar, 2012, Khodayari et al., 2012, Chong et al., 2011, Panwai and Dia, 2007, Jia et al., 2003). Compared with classical car-following modeling approaches, one unique benefit of machine learning is its nonparametric characteristic. Typically, in classical car-following models, one has to calibrate the parameters in the empirical equation to achieve a high level of accuracy. Whereas neural network based models can learn from field data and automatically generate the car-following model without many artificial parameters. They often produce superior results (Aghabayk et al., 2014, Mathew and Ravishankar, 2012, Khodayari et al., 2012, Chong et al., 2011).
Unfortunately, these neural network based models have not been tested under oscillating traffic conditions.

Therefore, we implement different neural network architectures under traffic oscillations, and the results indicate that neural network based models can produce highly accurate results when the tests are based on the inputs from field data. However, when being tested based on the inputs from the previous predictions similar to iterative simulation, the test results yield unsatisfactory performance. The results suggest that the unsatisfactory performance is mainly caused by insufficient data inputs and inaccurate measurements.

This paper aims to bridge the gap using a Recurrent Neural Network (RNN) based car-following model. RNNs are a variation of the neural network family, with greater strength in predicting sequential data (Lipton et al., 2015, Mikolov et al., 2010). The car-following behaviors can be treated as a behavior sequence due to the fact that the order of driving conditions matters when a driver is going to take their next action. For example, an acceleration phase followed by a deceleration phase is different to the reverse. With this strength, an RNN-based model considers not only the local information (such as short-term or immediate traffic condition used in classical car-following models), but also the global information which is a long-term condition stores in a RNN memory. In other words, RNN model takes action more like human in terms of the memory-based decision-making. In the car-following literature, a memory-based car-following model has also been paid attention to due to the driving history is an essential factor in predicting the next action. Treiber and Helbing (2003) address the historical impact by introducing an additional dynamical variable to a classical model. From this perspective, the memory effect does matter regarding developing a more accurate car-following model. Therefore, we develop an RNN-based model to capture car-following behaviors in oscillating scenarios and extended it to non-oscillating cases. The results illustrate that the RNN-based model
can successfully predict both oscillating and non-oscillating car-following rules and increase accuracy by learning driver behaviors.

The rest of the paper is organized as follows. Section 2 briefly reviews the literature on traffic oscillation and neural network based car-following models. Section 3 shows the data and the pre-processing method we adopted to regularize data noise. Section 4 compares existing neural network based car-following models under oscillating traffic circumstances, and raises the inaccuracy and other potential issues produced from using these models. At the end of this section, we attempt to fix these issues by proposing a new neural network model that predicts partial gaps. This new model improves the performance but has a shortcoming in handling less informative data. In section 5, we propose the RNN-based model to further improve modeling capacity in capturing driving behaviors. Section 6 concludes this paper.

## 2 Literature review

### 2.1 Oscillation

With the urbanization of our cities, traffic volume has been increasing continuously. A negative consequence of our daily travel is the notorious phenomenon known as traffic oscillation. The formation and propagation mechanisms of traffic oscillation, also known as stop-and-go traffic, have been intensively investigated during recent years in the literature. A general car-following model may not describe oscillation accurately. Therefore, researchers have proposed or improved upon existing models to better describe and illustrate the oscillating patterns on congested highways.

The measuring and analyzing methodologies for traffic oscillation have been developed and extended, which has improved the understanding of the mechanisms
of oscillations. The traffic flow on congested highways can be described as three phases including free flow, synchronized flow, and wide moving jam. Hence the three-phase traffic theory was proposed to fully explain the nonlinearity of traffic flow on highways (Kerner, 2012, Kerner, 1999). However, Treiber et al. (2010) discuss the inconsistent use of the term “traffic phases” and show the three-phase traffic theory can be reproduced with simple two-phase models with suitably specified model parameters and a consideration of factors characteristic for real traffic flows. In the term of vehicular speed, Tian et al. (2016) suggests that the standard deviation of speed grows concavely along vehicles in the oscillation, which they find is a universal property of traffic oscillation. Li et al. (2010) propose a frequency spectrum analysis approach that enhanced the measurement of the periodicity and magnitude in a traffic oscillation. The oscillation triggers at the vehicle level, and the propagation features of oscillation, were analyzed using the Wavelet Transform (WT) (Zheng et al., 2011a, Zheng et al., 2011b), which enables investigation of the oscillation behaviors down to the micro level.

Also, several car-following models have been proposed to capture traffic oscillation. Bando et al. (1998) investigate the properties of congestion and the delay time of car motion using optimal velocity model. Furthermore, the behavior of each driver differs, especially in a traffic oscillation. An oscillation is more likely to be instigated by an aggressive driver who may maintain a small response time and minimum spacing, which may also lead to capacity drop (Chen et al., 2014). According to this finding, to achieve more accurate car-following simulation, it is necessary to separate drivers depending on their driving behaviors (Chen et al., 2012, Laval and Leclercq, 2010). Laval and Leclercq (2010) introduce an additional parameter $\eta$ to the original Newell car-following formula (Newell, 2002) to distinguish driver’s behaviors. However, Chen et al. (2012) find the behavior model proposed in (Laval and Leclercq, 2010) captures one ideal case of driver behavior.
but far from sufficient to cover the general patterns of traffic oscillation. Therefore, they further improve this model and group drivers with respect to their different driving behaviors including originally aggressive, originally timid and originally Newell. Their results indicate more precise microscopic car-following patterns. On the other hand, Li and Ouyang (2011) presented the Describing-Function Approach (DFA) to predict the propagation properties in oscillations for nonlinear car-following behavior, and validated the approach given in (Li et al., 2012). Li et al. (2014) further employ this approach to investigate fuel consumption and emissions and explore ways of using connected autonomous vehicles to dampen traffic oscillation (Li et al., 2014). Built upon the describing function method, Rhoades et al. (2016) propose a method for calibrating nonlinear car-following model to quantitatively reproduce traffic oscillation. Laval et al. (2014) suggest that driver error contributes to oscillation formation and propagation. They improve the kinematic wave model to better reproduce driver behaviors in an oscillation. To capture the time-varying properties of oscillations, Zhao et al. (2014) focus on the analysis of divided short-term windows.

The above approaches and models perform better in oscillating traffic cases, due to their calibrated parameters and model constraints. Most of these existing studies are based on parametric car-following dynamic models constructed based on physics, behavior and sometimes artificial parameters for tuning model predictions. While these models may yield insightful explanations to traffic oscillation in certain settings, the fixed structure of these models may limit their flexibility and adaptivity to general traffic states and infrastructure settings. It is in general difficult to construct a universal parametric model compatible with various traffic scenarios, infrastructure types and data observations. Recently emerged artificial neural networks have been shown powerful, flexible and adaptable in addressing large-scale nonlinear problems with general settings. They are found suitable to be
applied to studies of traffic oscillation. Moreover, neural network based models are parameter-free and data-driven, and thus they are compatible with different traffic states, infrastructure configurations and available data. Therefore, neural network based car-following models hold promise for an alternative solution to predicting car-following and traffic oscillation with higher flexibility and robustness.

2.2 Neural networks and car-following models

As is the case with many well-known classical car-following models, the inputs into the network vary across studies. However, the predicted output from such models is usually determined as acceleration rate, velocity or gap distance for the target vehicle at the next time point. The following subsection reviews the existing research methods predicting each of those outputs.

2.2.1 Predicting acceleration

The acceleration rate has a direct link to the control of engine power. Many classical car-following models have already made great contributions towards predicting acceleration rates, such as the Gazis–Herman–Rothery model, known as the GHR model (Herman et al., 1959, Chandler et al., 1958), the Optimal Velocity model (OVM) (Bando et al., 1995, Bando et al., 1998), the Intelligent Driver Model (IDM) (Treiber et al., 2007b, Treiber et al., 2000), and the Full Velocity Difference Model (FVDM) (Jiang et al., 2001). Therefore, the acceleration rate would also be an ideal predicted output from a neural network.

Jia et al. (2003) introduce a four-layer neural network (including an input layer and an output layer). This neural network takes four input elements that consist of relevant speed, desired speed, follower speed and gap distance at the current time step and uses them to predict the acceleration at the last time step. The delay in
prediction is caused by the consideration of drivers’ reaction times. They also defined a nonlinear function for the hidden neural network layers, to improve the predictive accuracy. The data set they adopted was collected using the Five-Wheel System. The trained neural network is particularly capable of simulating human driving behaviors when predicting acceleration rates.

Chong et al. (2011) illustrate that it is still possible to predict acceleration rates accurately using a smaller-scale neural network. In their study, the neural network was built with only one hidden layer, and the number of inputs was reduced to three (speed, gap distance, and relevant speed). They trained the neural network on an episodic data set filtered from the Naturalistic Truck Driving Study (NTDS) (Olson et al., 2009). The predictive accuracy depended greatly on the episodes they selected for training. They declared that the accuracy of the output was restricted by the training episodes and it was not possible to generalize the results to different driving conditions.

To further improve the predictive accuracy, Khodayari et al. (2012) take all the inputs that Chong et al. (2011) considered and added the estimated instantaneous reaction delay. By training the network on the U.S. Federal Highway Administration’s noise-filtered Next Generation SIMulation (NGSIM) (FHWA, 2008) data set, they obtained a highly accurate simulated result.

2.2.2 Predicting velocity

Predicting velocity is another option since it is shown directly on the car dashboard for the driver to see and control. Therefore, the car-following model tends to mimic the speed a driver would maintain instead of controlling the acceleration rate directly. To be in line with this idea, many classical car-following models attempt to
calibrate velocity-based models, of which the Gipps model (Gipps, 1981) is one of the best-known representatives.

Panwai and Dia (2007) train neural networks for different driving modes to predict speed under various driving conditions. The inputs they selected were spacing headway and leading speed in different driving modes. Their data were collected from a congested single-lane road in Germany (Manstetten et al., 1997). However, the total amount of data used for training and testing is limited to 2100 samples. This amount may not be enough for neural networks to learn a general model, and can be compared with the NGSIM database that has more than a million samples.

Mathew and Ravishankar (2012) build three different neural network architectures with different inputs to predict vehicle-type-dependent following behaviors. In other words, the first neural network architecture include only two inputs, the leader velocity and the gap distance. The other two neural networks are based on the first one and contained additional inputs, namely the leader and follower vehicle types. They declare that the two neural networks with vehicle type inputs could capture the field velocity data closely.

Analyzing car-following models with respect to vehicle type has become a tendency due to the better results it produces. Aghabayk et al. (2014) focus on the modified neural network model only for heavy vehicles. The input vector consists of three elements: gap distance, follower speed and leader speed. Their results show that the modified neural network model fits the NGSIM velocity data better than the Gipps model.
2.2.3 Predicting gap distance

To predict the gap distance that a driver will maintain is another accepted method in the car-following model family. Because the gap is positional information, it can easily be extracted from trajectory data. In line with this idea, Helly (1961) develop a car-following model to describe drivers’ desired following distance. Another gap-based model is based on seeking to maintain a minimum safe distance from the vehicle, and is called the Collision Avoidance (CA) model (Kometani and Sasaki, 1961). Moreover, Newell (2002) develop a simple car-following model considering space and time shifts, which also involved gap information.

The aforementioned models have also been widely tested in simulations. However, to the best of our knowledge, this type of model has not yet been transformed into a neural network based model. Using a neural network to predict gap distance can be studied further.

3 Data description and pre-processing

3.1 Data description

To be consistent with most previous research, we adopt the data set from NGSIM (FHWA, 2008) to train and test our models. Specifically, we select the trajectory data on the northbound direction of Interstate 80 (I-80) in Emeryville, California. The data record frequency is 0.1 seconds per frame. The trajectory data example in study site is shown in Fig. 1.

In order to exclude the correlation between the neural network training process and the test process, we separated the data into a training data set and a test data set.
Vehicles travel between 4:00 and 4:15 experience fewer oscillations as shown in Fig. 1. Therefore, we choose the data on lane two collected from 4:00 to 4:15 p.m. on April 13, 2005 as the test data, and leave the rest of lanes in that period and all data from 5:00 to 5:15 p.m. to be the training data.
Fig. 1. Illustration of vehicle trajectories
3.2 Pre-processing

The trajectory data appear unfiltered and exhibited some noise artefacts (Khodayari et al., 2012, Thiemann et al., 2008), as can be seen in Fig. 2. Therefore, we apply a moving-average filter (Thiemann et al., 2008) for a duration of 0.8 s to all raw trajectories, and obtain the numerical velocity/acceleration data from the first-/second-order finite differences of the position respectively. We also attempt to train a neural network with the unfiltered data. The network yield poor predictive performance, especially when predicting the turbulent acceleration rate. This proves that the neural network model is sensitive to the training data used.

![Fig. 2. Comparison of filtered and unfiltered data for acceleration rate and velocity](image)

Further data pre-processing is applied and listed as follows:

- Filtered trajectory data less than one second;
• Filtered the first (last) two seconds trajectory data for followers who experience a cut-in (move-out) lane-change;
• Filtered fake collision/suddenly jump data segments;
• Set negative vehicle movements to zero which is caused by moving average smoothing; and
• Refined differentiated accelerations to a range of (-3.41376, 3.41376), which is consistent with the range in NGSIM acceleration data.

For training neural network models, the fitting procedure is data-point-based. We find previous 20-time steps for each data point and filter those points that do not have entire 20 previous steps. Therefore, the total training data contains 2,117,015 data points and randomly select 90% (2,011,164) of them for training, and the rest 10% (105,851) of them are used for validation.

To calibrate classical model, the calibration is trajectory-based. We further filter the trajectories less than five seconds to reflect a clear accumulative impact at the trajectory level. Hence, the training and test data for calibrating classical models contain 5,725 and 477 trajectories respectively. Randomly selected 90% (5,153) of the training data is for training and the rest 10% (572) is for validation. For RNN models, the sequence length is one of the key elements that affect the model’s performance. We set the sequence length to 600 steps (60 seconds) in order to continuously pass the hidden state through long sequences. Therefore, the number of total training trajectory for RNN models is reduced to 917 (826 for training and 91 for validation) as we only keep those trajectories longer than 60 seconds, while the number in the test set remains as 477.
4 Neural network based model

In the above literature review, we see that neural networks have the capacity to predict the acceleration rate and velocity. However, these highly accurate prediction methods have rarely been applied to real traffic data including oscillating traffic flow. We do not know what car-following behaviors will be exhibited when using those neural network models. Especially, there is a need to examine how a car follows another when encountering traffic oscillations. Therefore, in this section, we describe what happened when we applied different neural network architectures to test the performance of car-following manoeuver at the trajectory level.

4.1 Architecture

The efficient back-propagation algorithm (Rumelhart et al., 1988) is the foundation for today’s neural networks. A standard neural network feeds forward the input values for calculating the predicted outputs, and then uses the objective function to compute predicted errors from the real values. Lastly, it applies these errors as the gradients passing through the network to update the network parameters. Through this procedure, the neural network with updated parameters can produce more accurate predictions.

To develop a general neural network model that is compatible with most of the neural network models mentioned in the literature review, we decide to adopt the neural network architecture shown in Fig. 3. $i$, $j$ and $k$ at time $t$, and the input types can include leader speed, gap distance and the relevant speed of a following vehicle. Additionally, the inputs take time-series data into account in order to capture the driver’s reaction time. Therefore, the inputs not
only consist of one data point for each input type, but also batch data across the previous time steps \((T)\) denotes the number of time steps).

In the extreme case where \(T=1\), the neural network model outputs \(O_{t+1}\) only depend on the data of the last time step. The model, in this case, has the same architecture as instant models (Panwai and Dia, 2007, Chong et al., 2011, Mathew and Ravishankar, 2012). On the other hand, when taking \(T\) as a value greater than 1, the model has the same functionality as the models mentioned in other papers (Khodayari et al., 2012, Aghabayk et al., 2014, Jia et al., 2003), which consider drivers’ reaction time or delay.

So as to model the nonlinearity of car-following behaviors, we chose the Rectified Linear Unit (ReLU) (Nair and Hinton, 2010) as the activation function in this neural network model. The equation is as follows:

![General neural network architecture with time-series inputs](image-url)

**Fig. 3. General neural network architecture with time-series inputs**
where \( h_n \) denotes the \( n \)th hidden unit in the hidden layer, \( W_n \) and \( b_n \) are the weight matrix and bias for \( h_n \), respectively, and \( x \) is a vector of all inputs from the last layer. In general, if more than one hidden layer, the output vector of the last layer becomes the input vector of the next layer. The output \( O_{r+1} \) in Fig. 3 is computed as follows:

\[
O_{r+1} = W_o h + b_o
\]  

This equation is a linear function of \( h \). \( W_o \) and \( b_o \) are the parameters in the output layer, and \( h \) denotes the vector of all hidden units calculated using Equation (1). The predicted output \( O_{r+1} \) is related to all inputs in vector \( x \).

To quantify the quality of the prediction based on current network parameters, an objective function or so-called cost function is chosen, as follows:

\[
C(W, b) = (O_{r+1} - y)^2
\]  

\( h_n = \text{ReLU}(W_n x + b_n) = \begin{cases} W_n x + b_n; & x > 0 \\ 0; & x \leq 0 \end{cases} \)  

\( \text{ReLU}(x) = \begin{cases} x; & x > 0 \\ 0; & x \leq 0 \end{cases} \)
where \( y \) denotes the real value corresponding to the outputs. In our case, the cost function is also known as the mean squared error. This cost is then used as the source for updating all network parameters through back-propagation.

### 4.2 Results, applicability and critiques of existing neural network models

We conduct four simulations under different neural network architectures to test the predictive performance for the acceleration rate and velocity. For every simulation, the prediction is based only on the preceding trajectory from field data and the previously generated trajectory from the simulated vehicle. The neural network settings can be found in Table 1. \( T \) is fixed to one time step (or 0.1 seconds) for the neural networks, so that they predict based only on the information from the previous time step. The second scenario is used to consider drivers’ reaction delay, and here \( T \) is extended to the previous ten time steps (or 1 second). In the second case, the neural network considers more than a single time step so as to weight the inputs according to their importance to the output. Based on our expectations, the one-second scenario should perform better than the 0.1-second case, because it involves more input information. To completely reflect the impact of a driver’s reaction delay, various steps \( T \) are tested and compared.

<table>
<thead>
<tr>
<th>Model</th>
<th>( T ) (0.1s)</th>
<th>Prediction</th>
<th>Number of hidden units</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNa(0.1)</td>
<td>1</td>
<td>Acceleration</td>
<td>( 3 \times 3 = 9 )</td>
</tr>
<tr>
<td>NNa(0.1)</td>
<td>1</td>
<td>Velocity</td>
<td>( 3 \times 3 = 9 )</td>
</tr>
<tr>
<td>NNa(1.0)</td>
<td>10</td>
<td>Acceleration</td>
<td>( 3 \times 10 \times 3 = 90 )</td>
</tr>
</tbody>
</table>
In this study, the number of hidden units is based on the number of inputs. Normally, when we increase the number of hidden units, the network capacity for describing the nonlinearity will also improve. In the literature review, almost all neural network models have a relatively smaller number of hidden units. In order to increase the network capacity, we choose the number of hidden units to be three times greater than the number of inputs. For example, NNa(0.1) has three inputs, namely leader speed, gap distance and relevant speed, and the number of hidden units is nine. Moreover, as $T$ increases to 10 steps, the total number of inputs increases to 30. Therefore, NNs with $T=10$ and $T=20$ have 90 and 180 hidden units respectively. In the following subsections, these neural network models are tested and compared in detail.

In this study, the number of hidden units is based on the number of inputs. Normally, when we increase the number of hidden units, the network capacity for describing the nonlinearity will also improve. In the literature review, almost all neural network models have a relatively smaller number of hidden units. In order to increase the network capacity, we choose the number of hidden units to be three times greater than the number of inputs. For example, NNa(0.1) has three inputs, namely leader speed, gap distance and relevant speed, and the number of hidden units is nine. Moreover, as $T$ increases to 10 steps, the total number of inputs increases to 30. Therefore, NNs with $T=10$ and $T=20$ have 90 and 180 hidden units respectively. In the following subsections, these neural network models are tested and compared in detail.

<table>
<thead>
<tr>
<th>Model</th>
<th>Inputs</th>
<th>Type</th>
<th>Hidden Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNa(1.0)</td>
<td>10</td>
<td>Velocity</td>
<td>$3 \times 10 \times 3 = 90$</td>
</tr>
<tr>
<td>NNa(2.0)</td>
<td>20</td>
<td>Acceleration</td>
<td>$3 \times 20 \times 3 = 180$</td>
</tr>
<tr>
<td>NNv(2.0)</td>
<td>20</td>
<td>Velocity</td>
<td>$3 \times 20 \times 3 = 180$</td>
</tr>
</tbody>
</table>

Fig. 4. Cross-Validation of NNa and NNv
A cross-validation is further conducted to verify the performance of all Neural Network (NN) models. We again randomly select 90% of the training data to train the model and the rest of 10% of the training data to validate the cost. All models are trained on mini-batches of 128, and the average cost of the validation data at training step \( t \) is shown in Fig. 4 by using Equation (4).

\[
\bar{C}_t = \frac{\sum_{i=1}^{n} C_{i,t}(W, b)}{n}
\]  

(4)

where \( C_{i,t}(W, b) \) denotes the cost for validation data \( i \) with weights and biases at time step \( t \). The average cost \( \bar{C}_t \) at time \( t \) is to average all costs in the validation data.

After averaging several runs for each model, it is clear that for all NN model types, the number of historical inputs has its influence on the model performance. Generally, involving more historical inputs increases the accuracy the model predicts. However, the results also indicate that models with \( T \) greater than 1.0 second in general converges faster than others but will have a similar cost with whose \( T \) equals 1.0 second at the end of training.

4.2.1 Predicting acceleration

The above-mentioned neural network structures NNa(0.1), NNa(1.0) and NNa(2.0) belong to the family of the neural networks that predicting acceleration (Jia et al., 2003, Chong et al., 2011, Khodayari et al., 2012). We select a road segment from the
test data set to test the accuracy of these models. The predicted acceleration and velocity (obtained by integrating the predicted acceleration with the initial velocity from the field data) results are shown in Fig. 5.
Fig. 5. Comparison of predicted and real values

Note: the number next to y axis represents the ID of a vehicle in NGSIM
From Fig. 5, it indicates that NNa(0.1) under-fits the real acceleration rate and velocity curve due to the less informative inputs. NNa(1.0) and NNa(2.0) with more historical data inputs, on the other hand, successfully predict the acceleration rate. However, a small difference in predicted acceleration may lead to an unacceptable integrated velocity through time. This accumulative integration error becomes clear while we compare the velocity data. Therefore, training by this method fails to obtain an accurate car-following model.

4.2.2 Predicting velocity

The architecture of neural networks that predicts velocity is another architecture family (Panwai and Dia, 2007, Mathew and Ravishankar, 2012, Aghabayk et al., 2014). Similar to the prediction of acceleration, the predicted velocity and acceleration rate (obtained by differentiating the predicted velocity) are compared in Fig. 6. Because the inputs already contain a velocity factor, which makes them more informative for predicting velocity, the accuracy of the velocity prediction is higher than that of the pure acceleration rate. Based upon the real input data, all types of NNv output very accurate results. To some extent, the number of time steps \( T \) does not greatly affect this case.
Fig. 6. Comparison of predicted and real values

Note: the number next to y axis represents the ID of a vehicle in NGSIM
4.2.3 Shortcomings of existing neural network based model types

The results from NNa(1.0), NNa(2.0), NNv(0.1), NNv(1.0) and NNv(2.0) reflect a good predictive accuracy. However, the model performance under oscillated traffic flow has not been examined. The traffic oscillations have more uncertainty and worth analyzing. A model consuming fewer inputs has advantages on increasing calculating speed and reducing data dependency. Compared with models with $T=20$ and $T=10$, they both lead to a similar validation result (Fig. 4). We, therefore, conducted simulations to evaluate the performance of NNa and NNv with $T=10$ under traffic oscillation and the results are illustrated in Fig. 7.

Fig. 7. Comparison of the trajectories generated by neural network models and real data.
In this simulation, the NN models predict the first (velocity) and second (acceleration) order finite differences of positional data. We then obtain the predicted trajectories by the time integration of velocity and acceleration. Additionally, there are generally two approaches to simulate trajectories which can be defined as the types of leader the follower depending on. The first approach is to assume leaders’ trajectories are always given and we need to generate their followers’ trajectories by the model. The other approach is to fix only the first leader and all followers’ initial/boundary conditions by the data, and calculate trajectories of a group of subsequent vehicles by the model (Treiber and Kesting, 2013b). In this section, we choose the first approach to run the simulation. Eight followers’ trajectories are simulated with respect to their initial trajectory points and their data-driven leader. The result in Fig. 7 indicates that both models fail to predict a continuous trajectory.

Knowing this, the comparisons between the predicted acceleration rate and the real acceleration rate, and between the predicted velocity and the real velocity, become less significant. This result also indicates the rest of tests and evaluations in this paper should be done in the trajectory level to precisely reflect real car-following scenarios.

To investigate deeper, all current outputs from neural networks have an effect on the following inputs when testing at the trajectory level. Therefore, the error might accumulate. Hence, a potential solution is to break down the correlation between inputs and outputs, in order to break down the error-accumulating chain. This approach requires a reformation of the neural network architecture, which we will discuss in the next subsection.
4.3 A gap-based neural network model

The gap-based car-following models is another option for predicting following behaviors but at a trajectory level. The previous analyses from (Helly, 1961, Kometani and Sasaki, 1961, Newell, 2002) provide a theoretical foundation for a gap-based model. Based on this idea, we developed a gap-based neural network model and evaluated its performance.

4.3.1 Architecture

As we discussed above, in order to make a better prediction at the trajectory level, it is necessary to break down the link between inputs and outputs. The outputs are utilised by followers to change the following condition. Therefore, the inputs have to depend purely on the leader vehicle’s condition, because the decisions made by the leader will not relate (or only a little) to the decision made by the follower. Thus, from the perspective of the trajectory data and the leader’s behaviors, the neural network input we select is the leader’s displacements ($LD_t$ shown in Fig. 8). From the perspective of the follower’s behaviors, the neural network output we choose is the follower’s desired partial gap ($PG_{r+1}$ shown in Fig. 8). Therefore, the input and output are not associated with each other. Moreover, the follower’s velocity and acceleration rate can be calculated from the follower’s displacement ($FD_{r+1}$) which is related to $PG_{r+1}$ and $G_t$. 
We name this new type of neural network NNpg, referring to a neural network used to predict the desired partial gap distance \( (PG_{t+1}) \). \( G_t \) and \( LD_t \) denote the gap distance and leader displacement at time \( t \); \( LD_{t+1} \) and \( FD_{t+1} \) represent the leader displacement and follower displacement at time \( t+1 \).

Table 2. NNpg settings

<table>
<thead>
<tr>
<th>Model</th>
<th>( T ) (0.1s)</th>
<th>Prediction</th>
<th>Number of hidden units</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNpg(0.1)</td>
<td>1</td>
<td>Partial gap</td>
<td>( 3 \times 1 = 3 )</td>
</tr>
<tr>
<td>NNpg(1.0)</td>
<td>10</td>
<td>Partial gap</td>
<td>( 3 \times 1 \times 10 = 30 )</td>
</tr>
</tbody>
</table>

Fig. 8. A typical car-following trajectory
Similar to all NNa and NNv models, we define the network settings for NNpg in Table 2. The number of hidden units are shrunk to 1/3 of the number in NNa and NNv due to the reduction of the input dimension.

### 4.3.2 Results, applicability and critiques

The cross-validation of NNpg in Fig. 9 shows a similar pattern with other network types. Moreover, taking only one previous data as an input is not sufficient to predict accurate partial gaps. While NNpg(1.0) and NNpg(2.0) both outperform NNpg(0.1) and converge to a similar cost (around 70 calculated by Equation (4)) but with different converging speeds.
Similar to the reason of selecting NNa(1.0) and NNv(1.0) for the test at the trajectory level, we choose the NNpg(1.0) model for this test and the following summarize its settings:

- Input type: Leader’s displacement (LD);
- Input time steps (T): Last one second (10 steps);
- Output: Desired partial gap to maintain (PG); and
- Number of hidden units: 30.
Fig. 10. Trajectory comparison between the real trajectories and the trajectories predicted by NNpg(1.0)

The trajectory results generated by NNpg(1.0) are shown in Fig. 10. The simulation follows the same approach as it in Fig. 7 and the predicted oscillation phase outperforms previous models. However, due to the less informative inputs (only consist of the leader’s positional data) compared to NNa and NNv which have three types of inputs, the NNpg(1.0) can only predict a recommendation of $PG_{t+1}$ without a follower’s driving condition. In reality, the predicted driving behavior is not only conditional to leader’s behavior, but also conditional to follower’s. Additionally, in a control problem, this model is not fully functional and the partial gap is not capable of providing with a driving motion such as acceleration. Alternatively, the partial
gap is a raw information about controlling vehicles, but an appropriate and functional control mechanism would be acceleration-based. Hence, a valid car-following model should take these into consideration.

To solve the problem of uninformative inputs and focus on an appropriate control mechanism, we attempt to apply a variation of neural network type, as described in the next section.

5 Recurrent neural network based model

Through above simulations and tests, we find:

• A sequence of historical driving behaviors can improve NN models’ performance;
• Predicting and comparing at the trajectory level improves NN models’ performance; and
• A valid car-following model has to consider both the leader’s and the follower’s driving condition.

Through the above, we propose a recurrent neural network (RNN) based car-following model to satisfy those requirements. Namely, the RNN’s built-in mechanism for a sequential historical data as inputs is to eliminate the effort for tracking the previous data points and these inputs can contain both the leader’s and the follower’s information. Further, we set a trajectory-level objective function to capture a driver’s behavior.
5.1 Recurrent Neural Networks (RNNs)

The RNN has been applied in many fields, such as handwriting recognition (Graves et al., 2009) and speech recognition (Sak et al., 2014). The RNN has an internal state that represents the current situation. RNNs can build their internal memory as the foundation for the next prediction. A typical RNN’s architecture can be shown as in Fig. 11. It is clear that RNNs take a sequence of inputs to generate another sequence of outputs. All inputs and outputs are arranged in order. Therefore, RNNs learn the hidden sequence order as well as the corresponding output value.

![RNN Diagram]

Fig. 11. A typical RNN architecture, folded RNN (left) and unfolded RNN (right)

To achieve this RNN architecture, the equations for computing the output are also upgraded as follows:

\[ h_t = \text{ReLU}(W_h h_{t-1} + W_I I_t + b) \]  
\[ O_t = W_o h_t + b_o \]

(5)  
(6)
where $W_i$ and $b_i$ denote the input weights and biases, $h_t$ is the hidden state that represents the internal memory at time $t$, $I_t$ represents the input to the network at time $t$, $W_h$ are the weights of the hidden state, $W_o$ and $b_o$ stand for the output weights and biases, and $O_t$ represents the output at time $t$. The ReLU in Equation (5) is a nonlinear activation used to calculate $h_t$.

We expect a model for controlling vehicle, so it is necessary to predict acceleration rate rather than velocity or partial gap. Therefore, a RNNa model is proposed under this concern. Note that the RNN model predicts on a sequence of data and the sequence length can vary, so the constant sequence length $T$ used in NNa, NNv and NNpg is not needed.

The objective function or the cost function performing at the trajectory level is defined as follows:

\[
\begin{align*}
a_{t+1} &= RNNa(g_t, \Delta v_t, v_t) \\
x_{t+1} &= v_t \Delta t + (1/2) a_{t+1} \Delta t^2 \\
v_{t+1} &= v_t + a_{t+1} \Delta t \\
C(W, b) &= \frac{(x'_{t+1} - x_{t+1}) - g_{t+1})^2}{(g_{t+1})^2} 
\end{align*}
\]

The RNNa model takes inputs including gap ($g_t$), relevant speed ($\Delta v_t$) and vehicle speed ($v_t$) at time step $t$ and output acceleration ($a_{t+1}$) for the next time step. Then, the follower’s position ($x'_{t+1}$) and velocity ($v_{t+1}$) for the next step are updated based on
and the time interval $\Delta t$ which equals 0.1 in here. Based on the aforementioned discussion, we perform a comparison of the predicted gap with the gap in data. The predicted gap is calculated by subtracting the predicted follower’s position ($x_{i+1}^f$) from the data-driven leader’s position ($x_{i+1}^l$). The gap difference is squared then divided by the squared gap in data $(g_{r+1})^2$ to reduce the gap sensitivity (Kesting and Treiber, 2008). The $C(W,b)$ denotes the cost calculated based on current weights and biases in the RNNa model. The RNNa gradually minimizes this cost by backpropagating a small update through time in the direction of optimizing the weights and biases.

5.2 Results and performance comparison

We run a cross-validation about the average cost with different validating data for selecting an appropriate number of hidden units in RNNa’s cell. From Fig. 12, all three models with different unit numbers tended to converge to a cost near 0.07. Noted that the cost jump occurred at the 500th step on the curve of units = 100 was caused by mini-batch training as data in batches were different. The cost result indicates that the number of hidden units in the range of (20, 100) does not greatly affect the performance of the RNNa model. We choose the model with 60 units for the rest of tests as it performs slightly better than other models shown in Fig. 12 (b).
The following settings summarize the configuration for the selected RNNa:

- Input type: gap ($g_t$), relevant speed ($\Delta v_t$), follower’s velocity ($v_t$);
- Number of units in cell: 60; and
- Output: Follower’s acceleration rate ($a_{t+1}$).

Since the RNNa model has different property to classical car-following models, a comparison of RNNa with classical models is conducted. Firstly, we calibrate two of the most popular classical models, IDM and OVM, by Global Least-Squared Errors Calibration (Treiber and Kesting, 2013a) which is calibrated at a trajectory level to reduce the impact of the accumulative error. The model formulas are list as following and calibrated model parameters can be found in Table 3.
In Equation (10) and (11), the $v$, $\Delta v$ and $s$ represent the follower’s velocity, relevant speed and gap distance respectively. Others are the unknown parameters to be calibrated.

\[
a_{IDM} = a \left[ 1 - \left( \frac{v}{v_0} \right)^4 - \left( \frac{s_0 + vT + \frac{v\Delta v}{2\sqrt{ab}}}{s} \right) \right] \\
\]

(10)

\[
a_{OVM} = p_0 \left( (p_1 + p_2 \tanh(p_3s - p_4)) - v \right) \\
\]

(11)

Table 3. Calibrated parameters of the IDM and OVM

<table>
<thead>
<tr>
<th></th>
<th>IDM</th>
<th>OVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>2.01</td>
<td>$p_0$</td>
</tr>
<tr>
<td>$v_0$</td>
<td>27.19</td>
<td>$P_1$</td>
</tr>
<tr>
<td>$s_0$</td>
<td>6.73</td>
<td>$P_2$</td>
</tr>
<tr>
<td>$T$</td>
<td>1.53</td>
<td>$P_3$</td>
</tr>
<tr>
<td>$b$</td>
<td>1.77</td>
<td>$P_4$</td>
</tr>
</tbody>
</table>

Calibrated value

Calibrated value

The calibrated IDM and OVM are also compared in above-mentioned oscillation scenarios and their accuracy is quantified by calculating the Mean Squared Error (MSE) between data and predicted trajectory. The MSE is defined as below

\[
MSE = \frac{\sum_{i=1}^{n} (\bar{x}_i - x_i)^2}{n} \\
\]

(12)
where $\bar{x}_i$ and $x_i$ denote the $i$th predicted and field position respectively. Again, every follower’s trajectory is simulated based on its data-driven leader and the result is shown in Fig. 13.

![Trajectories predicted by IDM and OVM](image)

Fig. 13. Trajectories predicted by IDM and OVM

The IDM (MSE=6.94) has more accurate predictions compared with OVM (MSE = 11.03) in the traffic oscillation. Therefore, we choose the IDM as a reference to compare with the RNNa model.

For the subsequent tests, we look into the oscillation segment and run simulations based on different simulating approaches. The first approach is the same as it in above simulations which predicts a follower’s trajectory based on data-driven leader and follower’s initial state from data. The second approach is to simulate the trajectories of a group of subsequent vehicles based on one fixed first vehicle’s
trajectory and followers’ initial state. The results of these two approaches are shown in Fig. 14.

As illustrated in Fig. 14 (a), both RNNa and IDM fit field data very well with the MSE less than five when predicting by the first approach. However, Figure 14(b) indicates that the MSE of RNNa model is 58.3% of IDM’s, which means RNNa has a better prediction effect to a group of vehicles. In order to dig the reason behind, we conduct more detailed simulations to find the difference.

Firstly, we use test data sets to test two models by calculating their MSEs on every follower’s trajectory and show in a histogram (Fig. 15) to summarize models performance. As we can see, the RNNa model predicts trajectories with MSEs less than five more frequently. However, at a global level, these two models yield a similar MSE. The average MSE for all trajectories are very similar (29.08 for IDM
vs 27.25 for RNNa). We then decide to examine the models’ performances with respect to different driving behaviors.

As mentioned in the literature review, drivers’ characteristic varies. Drivers, therefore, can be divided into three different groups including “aggressive”, “timid” and “normal” (Chen et al., 2012). In the test data set, we distinguish trajectories into these three different characteristic tags and pick representative ones to plot in Fig. 16, Fig. 17 and Fig. 18.
Fig. 16. Predict aggressive drivers in a traffic oscillation
Fig. 17. Predict timid drivers in a traffic oscillation
Fig. 18. Predict normal drivers in a traffic oscillation

All these results indicate a special pattern of the RNNa model, which achieves a better fitness when a driver’s initial motivation is given. To distinguish drivers’
characteristics, the RNNa model can find the hidden information in a follower’s initial data point, while most classical models (except Chen et al. (2012)) cannot do so because they treat every driver as the same, regardless of initial data points. This can also be found in Fig. 19 which are tested under normal conditions. The RNNa model gathers the initial follower’s information from data and analyses it in order to understand which type of driver he or she is. The follower 2974 in Fig. 19 can be defined as a timid driver because it attempts to remain a larger gap compared with the gap predicted by the IDM. Another example is the trajectory of follower 3233, with an “aggressive” characteristic being identified at the initial point, the RNNa model attempts to predict an “aggressive” trajectory for this driver. However, after the time point 60, this driver tends to drive “normally” and RNNa can no longer fit to this “normal” trajectory. It indirectly proves that the RNNa model weights an initial given information more than what a classical model does and, in many scenarios, this initial information is a key for RNNa to understand the characteristic of a driver.
To summarize, we have the following findings:
• RNNa has a comparable performance with classical models in predicting the trajectory of the immediate subsequent vehicle;
• RNNa has a stronger performance than classical models in predicting the trajectories of subsequent eight vehicles;
• RNNa has a much stronger performance than classical models in predicting aggressive oscillations (oscillations caused by aggressive drivers);
• RNNa has a comparable performance with classical models in predicting timid and normal oscillations.

6 Conclusions

In this research, we adopt neural networks instead of classical car-following models to simulate traffic oscillations. We investigate the existing neural network based car-following models in congested traffic flow, and the results show that none of them is able to accurately predict drivers’ behaviors under traffic oscillations. We further identify potential reasons which are caused by insufficient inputs and inappropriate objective function for these neural network models. In order to overcome this deficiency, we apply a new type of neural network and design its architecture at the trajectory level (RNNa). The RNNa model gathers a sequence of historical data to approximate an acceleration rate and integrates it to positional data. We further use trajectories to train RNNa.

We test the RNNa model under both oscillated and non-oscillated traffic flow. The results indicate that RNNa has a stronger performance in predicting the trajectories of a group of subsequent vehicles given the trajectory of the first vehicle and initial/boundary conditions for following vehicles, while it has comparable performance with IDM model in predicting trajectory of the immediate subsequent vehicle. Compared to the fact that the same calibration parameters are used for all
vehicles in IDM model, NN-based models has advantages in identifying and differentiating different vehicles. We thus further compare the performances of RNNa and IDM in predicting the behaviors of different types of drivers. According to our results, RNNa significantly outperforms IDM in predicting oscillations caused by aggressive drivers. In this regard, the proposed RNNa model can be used to complement classical car following models in capturing traffic oscillations.
References


173


Qualcomm Technologies 2017. Accelerating C-V2X commercialization.


United Nations 2014. World’s population increasingly urban with more than half living in urban areas. New York.


Weng, J., Xue, S., Yang, Y., Yan, X. & Qu, X. 2015. In-depth analysis of drivers’ merging behavior and rear-end crash risks in work zone merging areas. Accident Analysis & Prevention, 77, 51-61.

Williams, R. J. 1988. Toward a theory of reinforcement-learning connectionist systems, Northeastern University.


World Health Organization 2017. Road traffic injuries.


