ABSTRACT

Title of dissertation: DYNAMIC TRAFFIC MANAGEMENT OF HIGHWAY NETWORKS

Fatemeh Alimardani,
Doctor of Philosophy, 2022

Dissertation directed by: Professor John S. Baras
Department of Electrical and Computer Engineering

Efficient operation of traffic networks via management strategies can guarantee overall societal benefits for both the humans and the environment. As the number of vehicles and the need for transportation grows, dynamic traffic management aims to increase the safety and efficiency of the traffic networks without the need to change the infrastructure of the existing roads. Since the highway networks are considered permanent investments that are expensive to build and maintain, the main scope of this dissertation is to propose traffic flow models and methods to improve the efficiency of the current highway systems without the need to change their infrastructure.

When all vehicles in a network are *Human-Driven Vehicles* (HDVs), and changing the infrastructure is either so expensive or impossible, then one reasonable approach to improve the efficiency of traffic networks is through the control of traffic signal lights specially because the behavior of the human drivers cannot be directly controlled. A literature review of highway traffic control demonstrate that *Ramp Metering* (RM) is one of the most commonly used approaches as it improves the network performance in regards to travel time, travel distance, throughput, etc and cost-wise, it is a very economical approach. As such,

in this research, the ultimate goal focus is to extend the current literature on traffic managements of highway networks by offering new models and algorithms to improve this field.

To reach this goal, the first step is to focus on improving and extending the current traffic flow models. There are two categories of traffic flow models in the literature: First-order models, and Second-order models. Many different extensions of the famous first-order model called the Cell-Transmission Model (CTM) have been proposed throughout the past decades, each one proposed based on different criteria and the specific needs of different applications. In the first part of this dissertation, a performance assessment of the most important extensions of CTM will be performed. Then, based on this evaluation, an extended version of the CTM, called the Piece-Wise Affine Approximation-CTM (*PWA-CTM*), will be offered which will be proven to have better performance regarding the evolution of traffic flow and computation time comparing to the previous versions of this model.

In the next step, the focus will be shifted to second-order models as they have better capabilities of modeling the behavior of traffic flow comparing to the first-order models. However, any optimization scheme for highway traffic control based on these models is highly nonlinear and computationally intensive. As such, in this part of the research, a linearization of the famous second-order model called the *METANET* will be offered which is based on PWA approximations and also synthetic data generation techniques. With extensive simulations, it will be shown that this linearized approximation can greatly impact the computational complexity of any optimization-based traffic control framework based on this second-order traffic flow model.

Moreover, to have significant traffic management improvements, not only the underlying traffic models, but also the control strategies should be enhanced. The availability of increasing computational power and sensing and communication capabilities, as well as advances in the field of machine learning, has developed *learning-based* control approaches

which can address constraint satisfaction and closed-loop performance optimization. In this chapter, *Reinforcement Learning* (RL) algorithms will be investigated to solve the optimal control problem of RM. In the case of RM, RL-based techniques offer a potentially appealing alternative method to solve the problem at hand, since they are data-based and make no assumptions on the underlying model parameters. Towards this direction, it is convenient to study the road model as a multi-agent system of non-homogeneous networked agents. In the following, a novel formulation of the RM problem as an optimal control problem based on a first-order multi-agent dynamical system will be offered. Then, applying policy gradient RL algorithms, a probabilistic policy will be found that solves the ramp-metering problem. The performance of the optimal policy learnt will be investigated under different scenarios to evaluate its efficiency.

DYNAMIC TRAFFIC MANAGEMENT OF HIGHWAY NETWORKS

by

Fatemeh Alimardani

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2022

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor Eyad H. Abed
Professor William Levine
Professor Dinesh Manocha
Professor Bruce L. Golden

Dedication

To my husband, Miead and my daughter, Jana.

Acknowledgments

I would like to express my deepest appreciation to my advisor, Professor John S. Baras, for the opportunity to be a part of his excellent research team. In the past 5 years, he has always been there with his invaluable patience, motivation, and immense knowledge. His insightful feed-back has always helped me to become a more critical thinker and be more innovative. His emphasis on identifying and solving real-world problems, and tireless passion for conducting research are some of the things I would like to thank him for. My dissertation research has been partially supported by the Office of Naval Research (ONR) grant N00014-17-1-2622, DARPA STTR contracts through Anthro Tronix Inc., and Lockheed Martin through the University of Maryland Foundation.

I am grateful to my committee members, Professor Eyad H. Abed, Professor William Levine, Professor Dinesh Manocha and Professor Bruce L. Golden, for sparing their valuable time and agreeing to serve on my committee.

Throughout my time at the University of Maryland, I have been fortunate to have wonderful friends and colleagues who deserve a special mention. First, I would like to thank Mrs. Kimberly Edwards for her unremitting help with any problem that may have occurred. Also, many thanks to my friends Christos, Nilesh, Usman, and Sandeep for their unconditional support, honest feedback, and heartfelt friendship.

Also, I would like to thank my friends Faezeh and Samaneh who have always been there for me like true sisters.

I wish to express my love to my mother, Nahid, my father, Reza, and my brothers for their love, support and encouragement.

Last, but not least, a special thank is given to my daughter, Jana, who has brought me more love and joy than I could have ever imagined possible. Finally, I would like to express my sincere gratitude to my husband, Miead Nikfarjam, for his faithful love and strongest support. This endeavor would not have been possible without him.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ITS	Intelligent Transport System
DTM	Dynamic Traffic Management
RM	Ramp Metering
RG	Route Guidance
VSL	Variable Speed Limit
FD	Fundamental Diagram
CTM	Cell-Transmission Model
ACTM	The Asymmetric CTM
PWA	PieceWise-Affine
RL	Reinforcement Learning
FHOCPs	Finite Horizon Optimal Control Problems
TTS	Total Time Spent
TTT	Total Travel Time
TWT	Total Waiting Time
TTD	Total Travel Distance
MSE	Mean Squared Error

List of Symbols

N	Number of cells	int
i	Cell index	$i = \{1, ..., N\}$
T	Sampling period	[h]
K	Time Horizon	int
k	Time index	$k = \{0, ..., K-1\}$
L_i	Length of each cell	[km]
v_i^f	Free-flow speed of cell i / (km/h)	
a_i	A parameter to be estimated from empirical data	
w_i	Congestion wave speed	[km/h]
q_i^{max}	Cell capacity	[veh/h]
λ_i	Lane numbers	int
ρ_i^{max}	Jam density	[veh/km]
ρ_i^{cr}	Critical density	[veh/km]
l_i^{max}	Maximum on-ramp queue length	[veh]
$r_i^{C,max}$	Maximum metering rate	[veh/h]
$\rho_i(k)$	Traffic density	[veh/km]
$\Phi_i^+(k)$	Total flow entering cell i	[veh/h]
$\Phi_i^-(k)$	Total flow exiting cell i	[veh/h]
$\phi_i(k)$	Mainstream flow	[veh/h]
$r_i(k)$	Flow entering from the on-ramp	[veh/h]
$s_i(k)$	Flow exiting through the off-ramp	[veh/h]
$\beta_i(k)$	Split ratio	$\in [0, 1]$
$l_i(k)$	Queue length in the on-ramp	[veh]
$d_i(k)$	Flow accessing the on-ramp	[veh/h]
$r_i^C(k)$	Ramp metering control variable	[veh/h]
r_i^*	Ramp metering set point	[veh/h]
K_R	Integral regulator gain	-
K_P	Proportional regulator gain	-
τ	Model parameter	-

ν	Model parameter	-
χ	Model parameter	-

Chapter 1: Introduction

1.1 Dynamic Traffic Management

As the number of vehicles and the need for transportation grow, cities around the world face serious traffic congestion problems: almost every weekday morning and evening during rush hours the capacity of many main roads is exceeded. Traffic jams do not only cause considerable costs due to lost time; they also raise the probability of accidents and have negative impacts on the environment (air pollution, lost fuel) and on the quality of life (health problems, noise, stress).

One possible approach to the ever growing traffic congestion problem is to extend the road network. Adding lanes and creating alternative new highway connections is possible but rather expensive. *Dynamic Traffic Management* (DTM) is an alternative approach that aims to increase the safety and efficiency of the existing traffic networks without the need to change the infrastructure of the network.

Road-based traffic control systems are nowadays the most commonly utilised approach to DTM. Such systems allow regulation of the traffic flow in a highway system by controlling all the vehicles together, i.e. by acting at a macroscopic level. These systems for

dynamic traffic control intervene in traffic in order to improve the performance of the traffic networks, i.e. to increase safety, to improve traffic flows, to reduce travel times, to make travel times more reliable, or to reduce emissions and noise production.

Any dynamic traffic control has two main ingredients:

1. A traffic flow model, and

2. A control technique.

This dissertation has the goal of contributing to the literature in both of these two categories.

1.2 Traffic Flow Models

The development of models to describe the dynamics of vehicles and their interaction (including the traffic behaviors and impact noted above) is fundamental for the design, prediction and control of traffic systems. Since the first traffic model developed by Lighthill and Whitham in 1955 [8], there has been a growing interest in the scientific community in modelling the dynamic behaviour of traffic systems. This has led to development of a large traffic models over the years, with different levels of detail, that are now available to scientists and traffic managers. Of course, in this huge variety of models, the most appropriate to be chosen for a given real case basically depends on the type and scope of the considered application.

A possible classification of traffic models is made according to their level of detail,

corresponding to the following cases [9]:

- *microscopic models*, in which the dynamics of all vehicles and their interactions are represented in detail. Normally each vehicle is described with a dynamic model, with different parameters (representing, for instance, the desired speed or acceleration capabilities of the vehicle, as well as the aggressiveness and reaction times of the driver). These types of models are very detailed and, consequently, can be computationally demanding for representing large road networks. Microscopic models are often embedded in simulation software tools;

- *macroscopic models*, in which the traffic dynamics is represented at an aggregate level. Specifically, the flow of vehicles is seen as a unique stream, in analogy with the flow of fluids or gases, and its dynamics is described by means of aggregate variables, such as density, mean speed and flow. These models are less computationally intensive than microscopic models and can allow fast simulations also for large-scale traffic networks. Macroscopic models are further classified according to the number of state variables, in first-order, second-order or higher-order models, and according to the number of represented vehicle types, in one-class or multi-class models;

- *mesoscopic models*, which present an intermediate level of detail. They do not distinguish individual vehicles but represent the heterogeneity of the drivers' choices in probabilistic terms.

It is important to emphasize that macroscopic models require a lower computational ef-

fort compared to microscopic models. In addition, it is normally easier to calibrate macroscopic models than microscopic models, since the former are characterised by fewer parameters. For these reasons, macroscopic models are surely the most suitable choice when they are used for control purposes, especially if the control law must be computed in real time, as, for instance, in optimal control schemes [9]. For this reason, this category of traffic flow models is chosen as the target models to focus on in this dissertation.

1.2.1 The Importance of Linear Traffic Flow Models

As aforementioned, one reason that macroscopic models are widely used for traffic management schemes is that they are less computationally intensive than microscopic models and can allow fast simulations also for large-scale traffic networks. However, available first-order and second-order traffic flow models in the literature are mainly nonlinear models and this causes computational complexities especially when it comes to optimization-based control methods. Nonlinear traffic flow models will highly affect the computation time of the simulation and also, for large-scale traffic networks, having nonlinear models will even make the problem difficult to solve. But why does simulation speed matter?

- Real-Time Decision Making: When using optimization to facilitate (or even automate) real-time decision making, the speed of the simulation and solving optimization problems really matters – as system operators can't afford to wait for their applications to generate solutions to their pressing, day-to-day business problems.

- Disruption Management: Another reason why having a high-speed simulations is so

pivotal is that it enables researchers and engineers to more effectively handle disruptions.

- Robust Scenario Analysis: Mathematical optimization technologies enable users to generate numerous what-if scenarios, evaluate and compare their potential impact on operations and objectives, and determine the best courses of action. Simply put, a faster simulations gives users the capability to run more scenarios in a shorter amount of time – and this has tremendous value.

As such, in this dissertation, after an analysis over the most famous first-order traffic flow models in chapters 2, in chapters 3 and 4 linear approximations of the most important first-order and second-order traffic flow models are offered.

1.3 Network-Oriented Traffic Control

The fact that the length, duration and the number of traffic jams is increasing has certain consequences for dynamic traffic control. When there are more congested locations, the available control measures have to solve more problems, which implies a higher complexity. Since nowadays the chances are higher that a vehicle encounters more than one traffic jam on its route, the traffic control measures influencing a vehicle in one traffic jam will also influence the other jam(s) that it encounters. Therefore, the spatial interrelations between traffic situations at different locations in the network get stronger, and consequently the interrelations between the traffic control measures at different locations in the network also get stronger. *Coordinated control strategies* are required in these cases, to make sure

that all available control measures serve the same objective.

One important ingredient of *network-oriented* traffic control is *coordination*. Since in a dense network the effect of a local control measure could also influence the traffic flows in more distant sections of the network, the control measures should be coordinated such that they serve the same objectives. Determining the effects of control measures on distant sections of the network also involves prediction, since the effect of the control measure has a delay that is at least the travel time between the two control measures in the downstream direction, and the propagation time of shock waves in the upstream direction.

Network-oriented traffic control has several advantages compared to local control. For example, solving a local traffic jam only can have as consequence that the vehicles more quickly enter another (downstream) traffic, while the same number of vehicles have to pass the downstream bottleneck (with a given capacity). In such a case, the average travel time on the network level will still be the same. A global approach would take into account and, if possible, ameliorate both traffic jams.

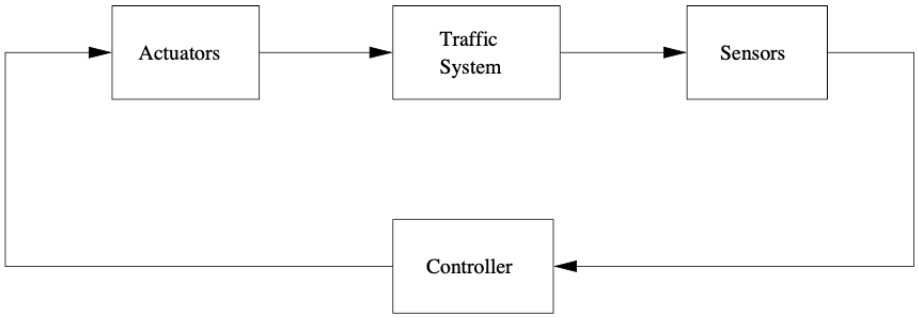


Figure 1.1: Schematic representation of the dynamic traffic management control loop

Dynamic traffic management systems operate according to the feedback control loop concept known from control systems theory (see Fig. 1.1). The traffic sensors provide information about the current traffic state, such as speed, flow, density, or occupancy . The controller determines appropriate control signals that are sent to the actuators. The reaction of the traffic system is measured by the sensors again, which closes the control loop. If new measurements show a deviation from the desired traffic system behavior (caused by unforeseen disturbances), the new control signals are adapted accordingly. Another problem is when the parameters of the traffic system change, e.g., when an incident occurs, or the weather conditions significantly change the system behavior. In that case the parameters of the model need to be adapted to the new situation. This is called *adaptivity* and is a very important feature for traffic control systems.

The control measures that are normally employed in highway networks are ramp management (in particular *Ramp Metering* (RM), applied using traffic lights at the on-ramps), mainstream control (including variable speed limits (VSL), lane control, congestion warning, keep-lane instructions, and so on), and route guidance (RG) (normally displaying specific indications at intersections). Amongst all these methods, coordination and adaptivity are two main factors that can make a traffic control algorithm to be more successful.

In this study, ramp metering (RM) has been chosen as the main control approach to focus on. Ramp metering uses traffic signals to regulate vehicle flows from on-ramps to the mainline of the highway. This technique can alleviate the negative impacts of "capacity drop" resulting from massive merging behaviors and reduce the average total time spent in

the traffic system. Several field tests have demonstrated the effectiveness of ramp metering in terms of throughput, vehicle- miles-traveled, vehicle-hours-traveled, and travel time reliability [9–13].

1.3.1 The Need for Learning-based Ramp Metering Algorithms

Ramp metering dynamically adjusts ramp flow of traffic merging into a highway mainline according to real-time traffic conditions to improve traffic operation. The main objective is to minimize the negative impact of ramp flow disturbance on mainline traffic to prevent the occurrences of traffic breakdown and capacity drop. Early RM studies and practice used the fixed-time control strategies, which decide the metering rate by setting the control cycle, green light phase and red-light phase [14]. Such strategies are easy to implement, but the major drawback is that the fixed parameters fail in responding to the changing traffic states, which can greatly reduce the strategy performances. Later, the feedback-based RM algorithms were proposed, which were traffic responsive in nature, adjusting the bottleneck occupancy at the expected value to maintain the maximum flow and prevent traffic breakdown [15–17]. However, the feedback-based RM adjusts the metering rate according to the traffic conditions. The performance is greatly limited by the feedback nature especially in the fast-changing traffic environments. In addition, setting control parameters in the feedback controller rely on human prior knowledge about the capacity drop and the breakdown probabilities.

Online optimization methods were used to calculate the metering rate by solving on-

line optimal control problems [18–20]. However, this approach requires accurate models to predict the traffic dynamics and contain large online computing workload which is considered not feasible for large-scale applications.

In this study, we investigate reinforcement learning algorithms to solve the optimal control problem of ramp-metering. Reinforcement learning methods are adaptive algorithms that approximate the solution to infinite-horizon optimal control problems [21]. In our case, they offer a potentially appealing alternative method to solve the problem at hand, since they are data-based and make no assumptions on the underlying model parameters.

1.4 Contributions and Outline of Dissertation

In Chap. 2, we analyze the performance of different first-order traffic flow models to provide a base model for the next chapters. To perform this assessment, different finite horizon optimal control problems (FHOCPs) with a linear combinations of multiple cost functions are formulated where each have different CTM versions as the underlying traffic flow model. For all FHOCPS, the PI-ALINEA RM metering control will be applied as the control approach. The performance of all FHOCPs will be investigated regarding the convergence of flow rates to the theoretical equilibrium flow vector, and the evolution of the metering rate of on-ramps w.r.t the changes of the exogenous demands profiles. This chapter concludes with some takeaways about the advantages and disadvantages of each CTM considered. This results will constitute the foundation of the next chapter.

In Chap. 3, a class of the Cell-Transmission Model (CTM) as the main first-order

traffic flow model is proposed which include the advantages of the previously offered CTM versions in the literature and at the same time, addresses their weakness. These extensions are based on a piecewise affine approximation (PWA) of the fundamental diagram (FD), and as they are linear models, it will be suitable to be used as the underling model of any optimal traffic control problem. To propose these models, a two-step identification methods will be used and then, different FHOCPs will be offered to evaluate the performance of these models. It will be shown that these models are superior to other extensions of the CTM as they can impose: 1) lower travel times in the mainline, 2) shorter vehicle queues at the on-ramp, and 3) higher traffic flows downstream of the merging area.

In Chap. 4, the focus shifts to second-order models as they have better capabilities of modeling the behavior of traffic flow comparing to the first-order models. However, optimization schemes for highway traffic control based on these models tend to be highly nonlinear and computationally intensive. As such, in this part of the research, a linearization of the famous model called the *METANET* will be offered which is based on piece-wise affine approximations and synthetic data generation techniques. With multiple FHOCPs, it will be shown that this linearized approximation can significantly reduce the computational complexity of any optimization-based traffic control framework based on this second-order traffic flow model while it shows an acceptable behavior regarding the accuracy and convergence to the equilibrium metrics.

In Chap. 5, *Reinforcement Learning* (RL) algorithms are offered to solve the optimal control problem of ramp metering (RM). In the case of RM, the RL approach enables the

agents to obtain optimal strategy in an unfamiliar environment without prior knowledge on traffic flow, reducing the need for human knowledge. Towards this direction, it is convenient to study the CTM traffic flow model as a multi-agent system of non-homogeneous networked agents. Next, a novel formulation of the RM problem as an optimal control problem based on a first-order multi-agent dynamical system is offered. Then, applying policy gradient RL algorithms, a probabilistic policy will be found that solves the ramp-metering problem.

Chap. 6 concludes remarks and discusses possible future work to the dissertation.

Chapter 2: Performance Assessment of Cell-Transmission Models for Ramp-Metered Highway Networks

Traffic flow models have been developed over the years and this field is still an ongoing research topic. Traffic flow models can be categorized into first order and second order models. The most frequently used models are first order models, such as the LWR model proposed in [22], which is a continuous model, and the cell-transmission model (CTM) proposed in [23, 24] which is a discretized version of the LWR model. The focus of this study is on first order models as the obvious disadvantage to second order models is that they lead to more complex optimization problems [25].

The CTM was first developed in [23] and then, through out the following years, many extensions of the original CTM have been proposed in the literature based on the applications and the purposes of researchers. The CTM for a freeway network ([26]), the lagged CTM ([27, 28]), the switched interpretation of the CTM ([29]), the asymmetric CTM ([25]), the link-node CTM ([30]), the CTM including capacity drop phenomena ([31, 32]), the graph constrained CTM ([33]), the CTM in a mixed-integer linear form ([34]), the variable-length CTM ([35]), width-based CTM for heterogeneous and undisciplined traffic streams ([36]), CTM for mixed traffic flow with connected and autonomous vehicles ([37]),

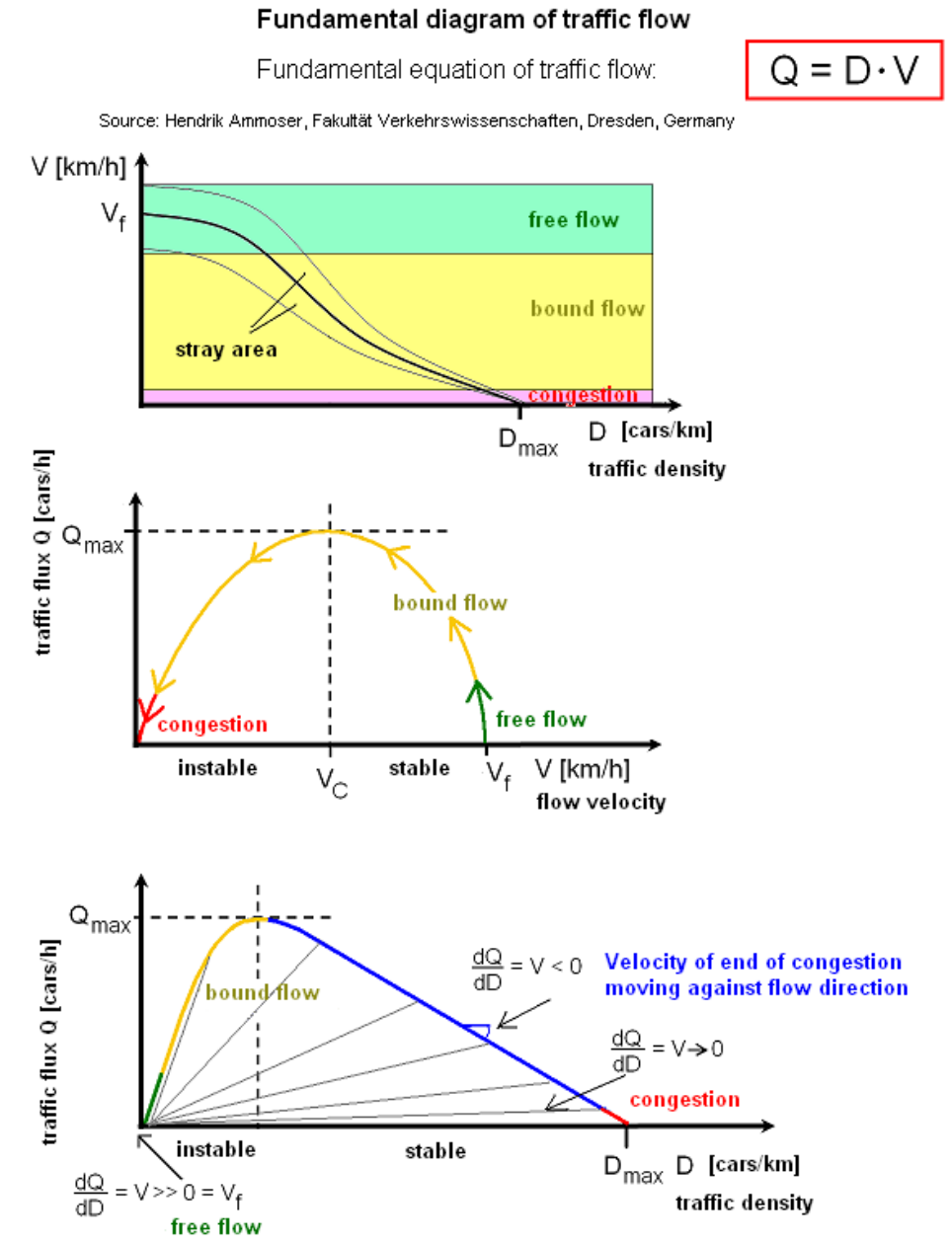
the CTM for heterogeneous disordered traffic ([38]), and multi-class CTM different traffic flow parameters ([39]) are some of these extended versions. Although these models have been proposed in different years and are suitable for different networks and applications, but still the original CTM [23], and most specifically the Asymmetric CTM (ACTM) [25] is the underlying model in all of them. This study aims at the comparison of the most commonly-used versions of the CTM.

In this section, first different versions of the CTM will be introduced. Then, after reviewing the PI-ALINEA RM control strategy, the formulation of multiple finite horizon optimal control problems (FHOCPs) will be explained. Last, the simulation results provide a thorough analysis of these models in details.

2.1 Preliminaries

2.1.1 The Fundamental Diagram

The fundamental diagram of traffic flow is a diagram that gives a relation between road traffic flux (vehicles/hour) and the traffic density (vehicles/km). A macroscopic traffic model involving traffic flux, traffic density and velocity forms the basis of the fundamental diagram. It can be used to predict the capability of a road system, or its behaviour when applying inflow regulation or speed limits [1].



V_f = "free velocity" - maximum velocity on free lane, selectable by the driver depending on car, skill etc.

V_C = "critical velocity" with maximum traffic flux (about 70...100 km/h)

Figure 2.1: Fundamental diagram of traffic flow [1]

There is a connection between traffic density and vehicle velocity:

- The more vehicles are on a road, the slower their velocity will be.
- To prevent congestion and to keep traffic flow stable, the number of vehicles entering the control zone has to be less than or equal to the number of vehicles leaving the zone in the same time.
- At a critical traffic density and a corresponding critical velocity the state of flow will change from stable to unstable.
- If one of the vehicles brakes in unstable flow regime the flow will collapse.

The primary tool for graphically displaying information in the study traffic flow is the fundamental diagram. Fundamental diagrams consist of three different graphs: flow-density, speed-flow, and speed-density. The graphs are two dimensional graphs. All the graphs are related by the equation "flow = speed * density"; this equation is the essential equation in traffic flow. The fundamental diagrams were derived by the plotting of field data points and giving these data points a best fit curve. With the fundamental diagrams researchers can explore the relationship between speed, flow, and density of traffic [1].

Speed-density

The speed-density relationship is linear with a negative slope; therefore, as the density increases the speed of the roadway decreases. The line crosses the speed axis, y, at the free flow speed, and the line crosses the density axis, x, at the jam density. Here the speed approaches free flow speed as the density approaches zero. As the density increases, the

speed of the vehicles on the roadway decreases. The speed reaches approximately zero when the density equals the jam density [1].

Flow-density

In the study of traffic flow theory, the flow-density diagram is used to determine the traffic state of a roadway. Currently, there are two types of flow density graphs: parabolic and triangular. Academia views the triangular flow-density curve as more the accurate representation of real world events. The triangular curve consists of two vectors. The first vector is the free-flow side of the curve. This vector is created by placing the free-flow velocity vector of a roadway at the origin of the flow-density graph. The second vector is the congested branch, which is created by placing the vector of the shock wave speed at zero flow and jam density. The congested branch has a negative slope, which implies that the higher the density on the congested branch the lower the flow; therefore, even though there are more cars on the road, the number of cars passing a single point is less than if there were fewer cars on the road. The intersection of free-flow and congested vectors is the apex of the curve and is considered the capacity of the roadway, which is the traffic condition at which the maximum number of vehicles can pass by a point in a given time period. The flow and capacity at which this point occurs is the optimum flow and optimum density, respectively. The flow density diagram is used to give the traffic condition of a roadway. With the traffic conditions, time-space diagrams can be created to give travel time, delay, and queue lengths of a road segment [1].

Speed-flow

Speed – flow diagrams are used to determine the speed at which the optimum flow occurs. There are currently two shapes of the speed-flow curve. The speed-flow curve also consists of two branches, the free flow and congested branches. The diagram is not a function, allowing the flow variable to exist at two different speeds. The flow variable existing at two different speeds occurs when the speed is higher and the density is lower or when the speed is lower and the density is higher, which allows for the same flow rate. In the first speed-flow diagram, the free flow branch is a horizontal line, which shows that the roadway is at free flow speed until the optimum flow is reached. Once the optimum flow is reached, the diagram switches to the congested branch, which is a parabolic shape. The second speed flow diagram is a parabola. The parabola suggests that the only time there is free flow speed is when the density approaches zero; it also suggests that as the flow increases the speed decreases. This parabolic graph also contains an optimum flow. The optimum flow also divides the free flow and congested branches on the parabolic graph [1].

2.2 The CTM and its extensions

In this section, first, a famous variant of the CTM, called the Asymmetric CTM (ACTM) is presented. This variant of CTM was proposed in [25]. Then, two of the most commonly-used extensions of the ACTM is introduced which are called the linear relaxation of ACTM ([25]) and the extended ACTM ([9]). As ACTM is the core of the other extensions, it won't be evaluated separately and the focus of this study, from now on, is on performance assessment of the two latter versions of the ACTM. Fig. 2.2 shows a stretch of a freeway with

queue length dynamics borrowed from [2]. The variables and parameters of cell i during time interval $[kT,(k+1)T)$ are described in the List of Symbols.

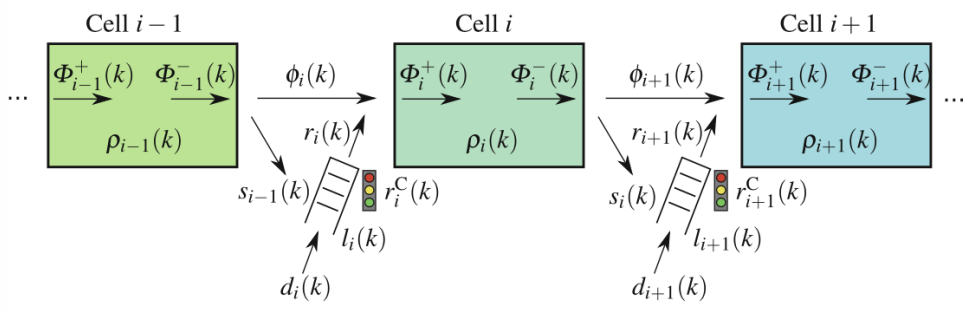


Figure 2.2: Sketch of a freeway stretch in the CTM ([2])

2.2.1 Version 1: The Asymmetric CTM (ACTM)

The ACTM proposed in [25], is a modification of the standard CTM proposed by [23]. Based on [25], the relevant difference between the two models is the treatment of traffic merges. More specifically, merges in the ACTM are considered as asymmetric connections, such as the junctions of the on-ramps into the mainstream. According to the logic of the standard CTM, the merge is oriented to move as much of the demand as possible from the two merging cells into the receiving cell. The ACTM, instead, makes separate allocations of supply for each merging flow. The flows can then be computed separately as the minimum among the demand, the allocated supply, and the capacity. Moreover, it is proved that the ACTM, as the CTM, ensures not to predict unrealistic behaviours such as backward moving traffic, negative densities and densities exceeding the jam density.

The ACTM is characterised by the following equations:

$$\rho_i(k+1) = \rho_i(k) + \frac{T}{L}(\Phi_i^+(k) - \Phi_i^-(k)) \tag{2.1}$$

$$\Phi_i^+(k) = \phi_i(k) + r_i(k) \tag{2.2}$$

$$\Phi_i^-(k) = \phi_{i+1}(k) + s_i(k) \tag{2.3}$$

$$s_i(k) = \frac{\beta_i(k)}{1-\beta_i(k)}\phi_{i+1}(k) \tag{2.4}$$

$$l_i(k+1) = l_i(k) + T(d_i(k) - r_i(k)) \tag{2.5}$$

$$\begin{aligned}\phi_i(k) = \min\{&(1-\beta_{i-1}(k))v_{i-1}(\rho_{i-1}(k) + r_{i-1}(k)),\\ &w_i(\rho_i^{max} - \rho_i(k) - r_i(k)), q_i^{max}\}\end{aligned} \tag{2.6}$$

For uncontrolled on-ramps:

$$r_i(k) = \min\{\frac{l_i(k)}{T} + d_i(k), \rho_i^{max} - \rho_i(k)\} \tag{2.7}$$

For controlled on-ramps:

$$r_i(k) = \min\{\frac{l_i(k)}{T} + d_i(k), \rho_i^{max} - \rho_i(k), r_i^{C,max}\} \tag{2.8}$$

$$0 \leq \rho_i(k) \leq \rho_i^{max}(k) \tag{2.9}$$

$$0 \leq \phi_i(k) \leq q_i^{max}(k) \tag{2.10}$$

$$0 \leq r_i(k) \leq r_i^{C,max} \tag{2.11}$$

$$0 \leq l_i(k) \leq l_i^{max} \tag{2.12}$$

Equations (2.1) through (2.12) describe the ACTM. Any optimization formulation based on this model will be non-concave and non-convex due to the *min* function in the mainline and on-ramp flow equations in (2.6)-(2.8).

2.2.2 Version 2: Linear Relaxation of ACTM

Gomes et a. in [25] offered this linear relaxed version of ACTM where (2.6)-(2.8) are replaced with linear equality and inequality equations and also upper bounds.

Equation (2.6) is replaced with:

$$\phi_i(k) \leq (1 - \beta_{i-1}(k))v_{i-1}(\rho_{i-1}(k) + r_{i-1}(k)) \tag{2.13}$$

$$\phi_i(k) \leq w_i(\rho_i^{max} - \rho_i(k) - r_i(k)) \tag{2.14}$$

$$\phi_i(k) \leq q_{i-1}^{max} \tag{2.15}$$

For uncontrolled on-ramps, (7) is replaced with:

$$r_i(k) = d_i(k), \quad 0 \leq r_i(k) \tag{2.16}$$

And, for controlled on-ramps, (8) is replaced with::

$$r_i(k) \leq l_i(k) + d_i(k), \quad 0 \leq r_i(k) \leq r_i^{C,max} \tag{2.17}$$

In summary, (2.1)-(2.5), and (2.9)–(2.17) describe the second version of ACTM.

2.2.3 Version 3: The Extended ACTM

The difference between this version presented in [9] and the original ACTM version presented in ([25]), named version 1 here, is how this version handles the merge between the on-ramp and mainstream cells. This version distinguishes between the flow rate equations of the free flow and congested case when on-ramp flow is merging to the mainstream flow. To do so, let's first introduce the demand and supply functions. The mainstream cell demand $D_{i-1}(k)$ is the flow that cell $i-1$ could send to the next cell i and the mainstream

cell supply $S_i(k)$ is the flow that cell i could receive from cell $i-1$.

$$D_{i-1}(k) = \min\{(1-\beta_{i-1}(k))v_{i-1}\rho_{i-1}(k), q_{i-1}^{max}\} \tag{2.18}$$

$$S_i(k) = \min\{w_i(\rho_i^{max} - \rho_i(k)), q_i^{max}\} \tag{2.19}$$

Also, the on-ramp demand $D_i^{ramp}(k)$ is the flow that can be sent from the on-ramp into cell i.

For uncontrolled on-ramps:

$$D_i^{ramp}(k) = \min\{d_i(k) + \frac{l_i(k)}{T}, r_i^{max}\} \tag{2.20}$$

For controlled on-ramps:

$$D_i^{ramp}(k) = \min\{d_i(k) + \frac{l_i(k)}{T}, r_i^{C}(k), r_i^{max}\} \tag{2.21}$$

The merge between the on-ramp and the mainstream is analogous to the merge of two generic cells. Two cases must be distinguished, corresponding, respectively, to free-flow and congested conditions:

If $D_{i-1}(k)+D_i^{ramp}(k) \leq S_i(k)$ (free-flow case), then

$$\begin{aligned} \phi_i(k) = &\quad D_{i-1}(k) \\ r_i(k) = &\quad D_i^{ramp}(k) \end{aligned} \tag{2.22}$$

If $D_{i-1}(k)+D_i^{ramp}(k) \geq S_i(k)$ (congested Case), then

$$\begin{aligned} \phi_i(k) = &\quad \text{mid}\{D_{i-1}(k), S_i(k)-D_i^{ramp}(k), p_i S_i(k)\} \\ r_i(k) = &\quad \text{mid}\{D_i^{ramp}(k), S_i(k)-D_{i-1}(k), p_i^{ramp} S_i(k)\} \end{aligned} \tag{2.23}$$

where the function mid returns the middle value. The parameters p_i and p_i^{ramp} model, respectively, the priority of the mainstream flow and the on-ramp flow in the merge and $p_i^{ramp}+p_i=1$.

In summary, (2.1)-(2.5), (2.9)-(2.12), and (2.18)-(2.23) describe the third version of the ACTM. Note that, the split ratios, i.e. $\beta_i(k)$, the demand in the cell before the first one, i.e. $D_0(k)$, and the supply of the cell after the last one, i.e. $S_{N+1}(k)$, and the flows accessing the on-ramp queues, i.e. $d_i(k)$ are the boundary conditions in this version.

It should be added here that all the three versions explained up to this point represent the necessary equations to model a freeway *stretch*. In order to model a freeway *network*, it is necessary to also add the merge and diverge models. Detailed explanations can be found in [9]. Since, these models are the same for all the versions stated above, and having them here would not affect the goal of this study, their details are not provided and interested readers are motivated to study the main reference.

2.3 Ramp Metering Control

Ramp metering is achieved by placing traffic signals at on-ramps to control the flow rate at which vehicles enter the freeway. The ramp metering controller computes the metering rate to be applied. This study applies the feedback local RM strategy PI-ALINEA proposed by [40] which is an extension of ALINEA developed by [15]. According to the stability analysis of the closed-loop RM system reported by [40], it can be stated that PI-ALINEA is able to guarantee a better control performance than ALINEA. PI-ALINEA is basically a PI-type controller in which the metering rate is given by

$$r_i^C(k) = r_i^C(k-1) + K_R[\rho_i^* - \rho_i(k)] - K_P[\rho_i(k) - \rho_i(k-1)] \tag{2.24}$$

where the flow that can enter section i of a freeway from the on-ramp of cell i during time interval $[kT,(k+1)T)$ is shown by $r_i^C(k)$. In case, the main objective of the traffic controller is to reduce congestion and to maximize the throughput, a good choice for the set-point is $\rho_i^* = \rho_i^{cr}$.

2.4 Finite horizon optimal control problems (FHOCPs)

The main objective of traffic control is to improve the network performance. However, network performance can be interpreted in many ways, and for every interpretation a different optimization problem can be formulated. To achieve the goal of this study, four FHOCPs will be proposed in this section and the above-mentioned ACTM versions will

play the role of the underlying model in these optimization control problems. In section 2.4.1, the cost functions considered for these problems will be explained and in section 2.4.2, the formulation of these FHOCPs will be presented in a compact form.

2.4.1 Cost functions

In this section, the definitions of several cost functions will be provided. In the formulation of FHOCPs presented in this section, a linear combination of these cost functions will make the final objective of each problem.

The most frequently used objective is to minimize the total time that all vehicles spend in the network (i.e., the Total Time Spent or TTS). Another advantage of the TTS is that it can easily be calculated for macroscopic models. Basically, the TTS is the time spent by all vehicles in the network (i.e., the total travel time or TTT), including the waiting time experienced at origins (i.e., the total waiting time or TWT). In other words, $TTS = TTT + TWT$.

$$J_1 = J_{TTS} = T \sum_{k=0}^{K-1} [\sum_{i=1}^{N} \rho_i(k) L_i \lambda_i + \sum_{i=1}^{N} l_i(k)] \tag{2.25}$$

The second objective function applied here is to maximize the sum of the traffic flows going through all sections and on-ramps. This objective function is also called the Total Travel Distance (TTD) since it is the total distance (veh mi) covered by all the vehicles in the considered time horizon.

$$J_2 = J_{TTD} = T \sum_{k=0}^{K-1} [\sum_{i=1}^{N} \phi_i(k) L_i + \sum_{i=1}^{N} r_i(k) L_i] \tag{2.26}$$

Another term that is often used in the objective function of traffic management problems is a term that penalizes control signal variations. This terms helps to suppress the high-frequency oscillations of the control trajectories. Since here RM provides the control variable, the following term is penalizing the RM control variable:

$$J_3 = J_{r_i^C} = T \sum_{k=0}^{K-1} \sum_{i=1}^{N} [r_i^C(k) - r_i^C(k-1)]^2 \tag{2.27}$$

Also, the maximum ramp queue constraints may be taken into account via the introduction of a penalty term in the cost criterion penalizing queue lengths larger than l_{max}, the maximum admissible queue for origin i.

$$J_4 = J_{l_i} = T \sum_{k=0}^{K-1} \sum_{i=1}^{N} [\max\{0, l_i(k) - l_{max}\}]^2 \tag{2.28}$$

Optimizing TTS and TTD are the main objective of this study and the contribution of the two penalty terms to the total cost criterion is very small as these two only penalize control signal variations. Therefore, they are not used as separate costs functions. For a traffic problem, a trade-off has to be made between the partial objective functions $J_i(k)$, $i = \{1,2,3,4\}$, which can be expressed by combining the objective functions into one objective function:

$$J_{total} = \sum_i \alpha_i J_i(k) \tag{2.29}$$

where α_i are appropriately chosen weights to express the trade-off between the several partial objective functions and in each problem $\sum_i \alpha_i = 1$.

2.4.2 Problem Formulation of FHOCPs

In this section, four FHOCPs is presented where the control strategy in all of them is the PI-ALINEA RM control. The difference between these problems is on the cost function and the version of the ACTM used as the model. For two of the problems a linear combination of TTS and TTD is used and for the other two problems, the two penalty terms are also considered. Regarding the ACTM version used, version 1 of the ACTM has the underlying equations used in the other two versions. As a result, it will not be used as a separate model for the simulation phase of this study. For FHOPC 1 and 2, the extended ACTM and for FHOPC 3 and 4, the linear approximation of ACTM is used. The formulation of the four FHOCPs proposed in this study is provided in a compact form as follows:

FHOCP 1:

Cost function:

$$\min_{r_i^C} \alpha_1 J_1 - \alpha_2 J_2 \tag{2.30}$$

Subject to:

Equations (2.1)-(2.5), (2.9)-(2.12), and (2.18)-(2.24).

FHOCP 2:

Cost function:

$$\min_{r_i^C} \alpha_1 J_1 - \alpha_2 J_2 + \alpha_3 J_3 + \alpha_4 J_4 \tag{2.31}$$

Subject to:

Equations (2.1)-(2.5), (2.9)-(2.12), and (2.18)-(2.24).

FHOCP 3:

Cost function:

$$\min_{r_i^C} \alpha_1 J_1 - \alpha_2 J_2 \tag{2.32}$$

Subject to:

Equations (2.1)-(2.5), (2.9)-(2.12), (2.13)-(2.17), and (2.24).

FHOCP 4:

Cost function:

$$\min_{r_i^C} \alpha_1 J_1 - \alpha_2 J_2 + \alpha_3 J_3 + \alpha_4 J_4 \tag{2.33}$$

Subject to:

Equations (2.1)-(2.5), (2.9)-(2.12), (2.13)-(2.17), and (2.24).

In the total cost function of each problem, every partial function with a positive sign will be minimized and every other one with a negative sign will be maximized.

2.5 Simulation Results and Analysis

2.5.1 Case study and model parameters

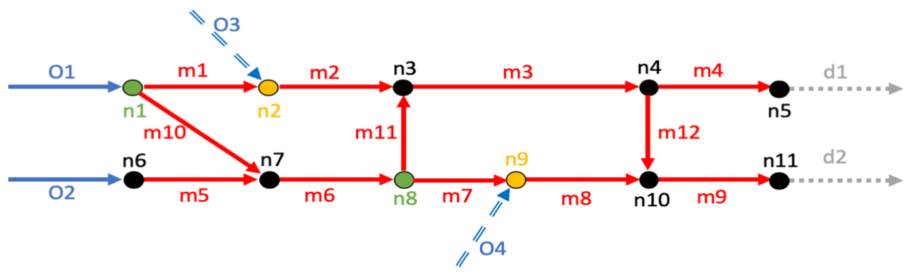


Figure 2.3: The hypothetical network under consideration

Simulation is performed for the network shown in Fig. 2.3 with two origins ($o1$ and $o2$), two controlled on-ramps ($o3$ and $o4$), 12 mainline links ($m1$ through $m12$) and two destinations ($d1$ and $d2$). One assumption about the network is that the proportion of turns at every junction, i.e. the split ratios $\beta_i(k)$, are fixed and known in advance. Also, it is assumed that the behavior of all the links can be described by a pre-known fundamental diagram with the parameters shown in Table 2.1 adopted from [41]:

Table 2.1: Model Parameters

Symbol	**Value**	**Unit/Range**
Period T	0.5	min
Length L_i	1	km
v_i^f	1	length/period
w_i	0.33	length/period
ρ_i^{max}	180	veh/length
ρ_i^{cr}	50	veh/length
q_i^{max}	50	veh/period

2.5.2 The Demand Profiles

The simulation horizon of 5 hours is considered by choosing the time horizon of K = 600 time steps and the simulation period of T = 0.5 minute ($K * T$ = 300 minute). For the origin links, two different types of demand profiles are applied: 1) the stationary, and 2) the time varying. Also, in order to approximate an empty final condition ([25]), a imaginary "cool down" period is considered at $K = 450$ till the end of the time horizon, in which all demands are set to zero. The demand profiles are shown in Fig. 2.4. It is important to notice that although the time-varying demand is changing at different time steps, however, the max values for each of the four demand functions are chosen to be fixed to the values of demand in the stationary demand profile. For example, the demand of origin 2 is equal to 9 veh/0.5 min in the stationary demand case between $k = 0$ to $k = 450$. On the other hand, in the time-varying case, this function also has the max value of 9 veh/0.5 min between $k = 100$ to $k = 400$. Also, all demand values has the min value of zero in the cool-down period. This holds true for all the four demand functions. The reason behind these choices is due to the fact that similar max and min values makes it easier to compare and analyze the solutions based on these two different demand patterns.

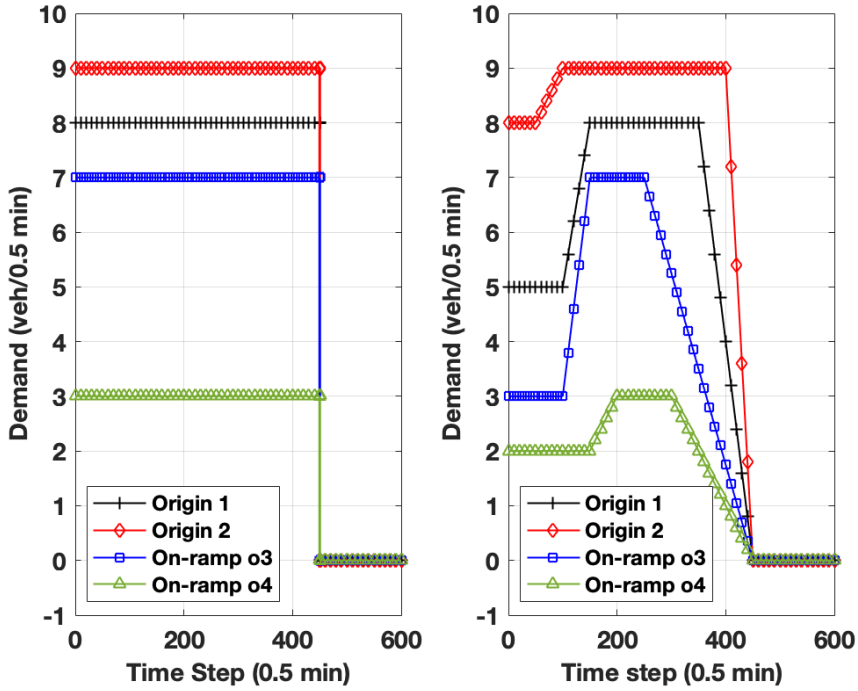


Figure 2.4: The demand profiles (Left: The stationary demand profile, Right: The time-varying demand profile)

2.5.3 Equilibrium State of the Network

According to [41], for each stationary demand vector $d(k) = (d0, ..., d_M)$, there exists a unique equilibrium flow rate $q(k) = (q_0, ..., q_N)$ and density vector $\rho(k) = (\rho_0, ..., \rho_N)$. Detailed explanation on the calculation of these vectors is concisely explained in [41]. Table 2.2 shows the theoretical equilibrium flow vector for the mainline links of this network based on the stationary demand vector shown in Fig. 2.4. Knowing this theoretical flow vector provides an insight on what should be expected to be seen in the simulation results.

Although based on [41], the equilibrium state of the network can only be computed for

the case of having stationary demand profiles, however, this information, if available, will help to analyze the behavior of models even for the case of applying time-varying demand profile. It can be assumed that at each time step, the values of the demand functions are kept constant between the time interval $[kT,(k+1)T)$. Based on this assumption, the equilibrium flow vector for the time-varying case was also computed with the same method applied before. However, in this case, this vector cannot be simply showed in a compact form like table 2.2 as the flow values are time-varying through the whole simulation horizon. Still, this information could help to analyze the performance of the FHOCPs in the time varying case. Further explanations are provided in section 2.5.4.

Table 2.2: Theoretical equilibrium flow rate (veh/0.5 min) vector

Link Number	**1**	**2**	**3**	**4**	**5**	**6**
Flow rate	4.8	11.8	15.46	9.27	9	12.2
Link Number	**7**	**8**	**9**	**10**	**11**	**12**
Flow rate	8.54	11.54	17.72	3.2	3.66	6.18

2.5.4 Simulation Results

The results of the simulation for the demand profiles of section 2.5.2 and the FHOCPs of section 2.4.2 are provided in this section. The FHOCPs were solved with Yalmip, the modeling and optimization language offered by [42], and the GUROBI non-commercial optimization solver via the interface of MATLAB. The simulations were performed on a device with 2.9 GHz Dual-core Intel Core i5 CPU with 8GB RAM.

Figures 2.5 and 2.6 show the boxplot of the mainline link flow rates (veh/0.5 min) with

the stationary and time-varying demand profiles, respectively. With a general glance, it can be seen that in both figures, all four problems are showing similar behaviors for the evaluation of the mainline flow rates. In Fig. 2.5, the top edge and the central mark of each box are in the same position and it is fixed at the value equal to theoretical flow of that link based on table 2.2. In other words, all four FHOCPs are able to reach the equilibrium state of the network in case of the stationary demand. In Fig. 2.6, the top edge is at the same position as the top edge in Fig. 2.5, however, the central mark is at a lower level. This matches the expectations in the sense that as the demand is changing, the flow rate values can not converge to a constant value. However, the top and bottom edge of each box are at the same position between these two figures for all the links.

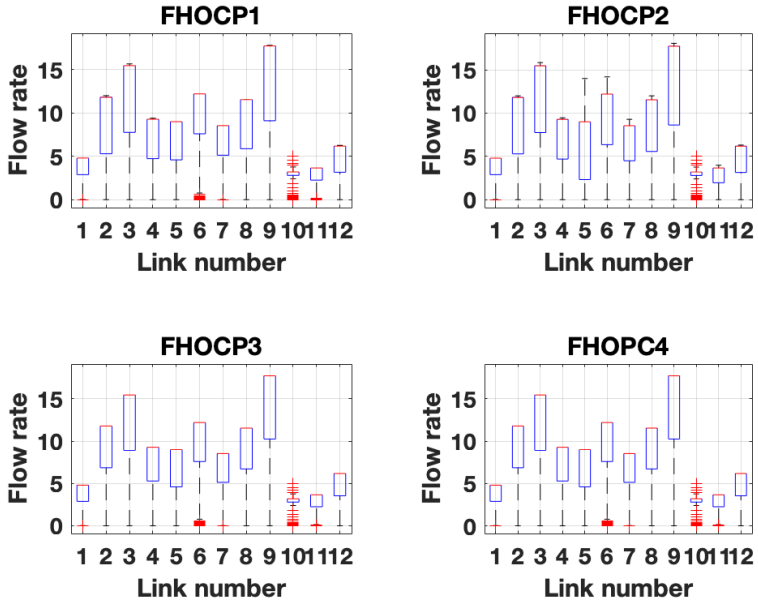


Figure 2.5: Boxplot of the mainline link flow rates (veh/0.5 min) with the stationary demand profile

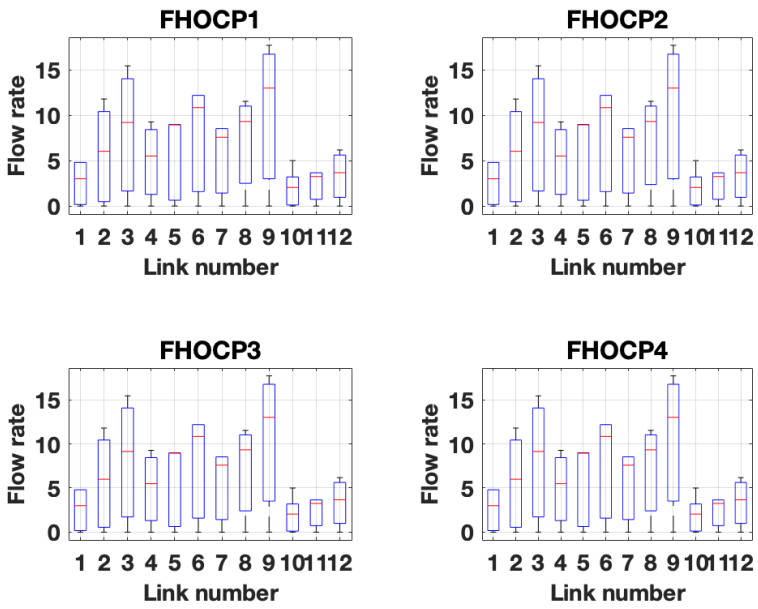


Figure 2.6: Boxplot of the mainline link flow rates (veh/0.5 min) with the time-varying demand profile

A method to compare the evolution of flow rates in these optimal problems, under the two defined demand profiles, is to compute the root-mean-square error (RMSE) of the flow rate values comparing to their theoretical equilibrium values. Basically, RMSE is a frequently used measure of the differences between values (sample or population values) predicted by a model and the values observed. Table 2.3 summarizes the RMSE values for each problem with the two demand profiles applied. As it can be seen, overall, the FHOCPs show less error if the demand pattern is stationary, indicating that the optimal control problems demonstrated higher accuracy in converging to the equilibrium state of the network. However, the mean value of the RMSE of the second case has also a fairly

low value, proving that even with the time-varying demand profiles applied, the FHOCPs can provide an acceptable performance. Also comparing performance of FHOCPs with different underlying ACTM versions, on average, the problems with the extended ACTM version are showing slightly better efficiencies.

Table 2.3: Comparison of the RMSE of the flow rate values

Problem	**RMSE**	
	Stationary	Time-varying
FHOCP 1	1.7224	2.5344
FHOCP 2	2.3112	2.2070
FHOCP 3	2.2949	2.9585
FHOCP 4	2.0179	2.8940

The analysis continues by providing the evolution of flow rate values of the network by simply adding the flow rate of all the mainline links, as shown in Fig. 2.7. Based on table 2.2, the total sum of the link flow rates at the steady state is expected to be equal to 113.37 veh/0.5 min. This matches perfectly with the experimental maximum value of the network flow rate as seen in Fig. 2.7 in both cases. A very considerable point to mention is the difference between the fluctuations of the flow rate in each problem. FHOCP 1 (blue color) is showing much more fluctuations comparing to FHOCP 2 (red color) as the cost function of FHOCP 2 has the two penalty terms explained in section 2.4.1 and they are successfully suppressing the high-frequency oscillations of the control trajectories. Same can be seen between the performance of FHOCP 3 and 4 where less oscillations are seen in the performance of FHOCP 4 due to the presence of the penalty terms in its cost function.

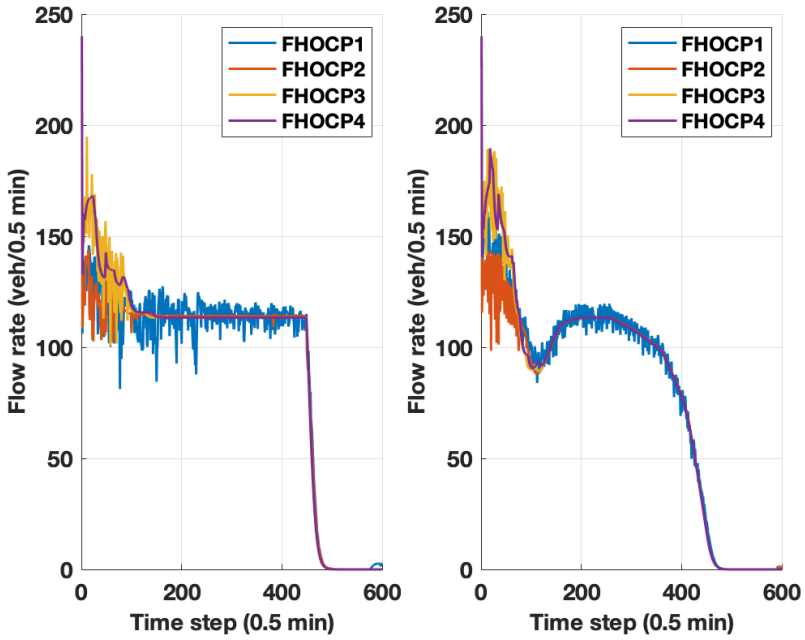


Figure 2.7: Network flow rates (veh/0.5 min) (Left: The stationary demand profile, Right: The time-varying demand profile)

The evolution of the ramp metering variables is also investigated here. Basically, the expectation is that the control applied to the on-ramps can satisfy the requirements of the network while considering the external demand of the on-ramps by not imposing too much waiting time to the vehicles on the on-ramps. As a result, a proper RM variable basically should replicate the overall demand pattern of that on-ramp. As an example, Fig. 2.8 and 2.9 show the RM variable of on-ramp $o4$. It can be easily seen that in both figures, all four problems are showing a behavior similar to the demand function of this on-ramp as seen in Fig. 2.4 (in green color with triangle marker). The interesting point is the reduction of oscillations between FHOCP 1 and 2 and also between FHOCP 3 and 4 which clearly

represent the impacts of the RM penalty terms in the cost function of FHOCP 2 and 4.

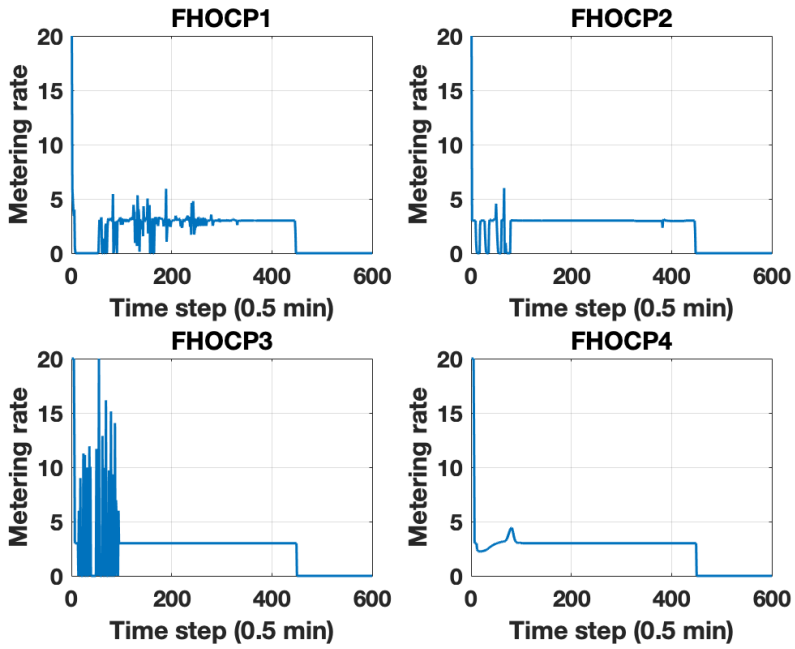


Figure 2.8: Metering rate of on-ramp o4 (veh/0.5 min) with the stationary demand profile

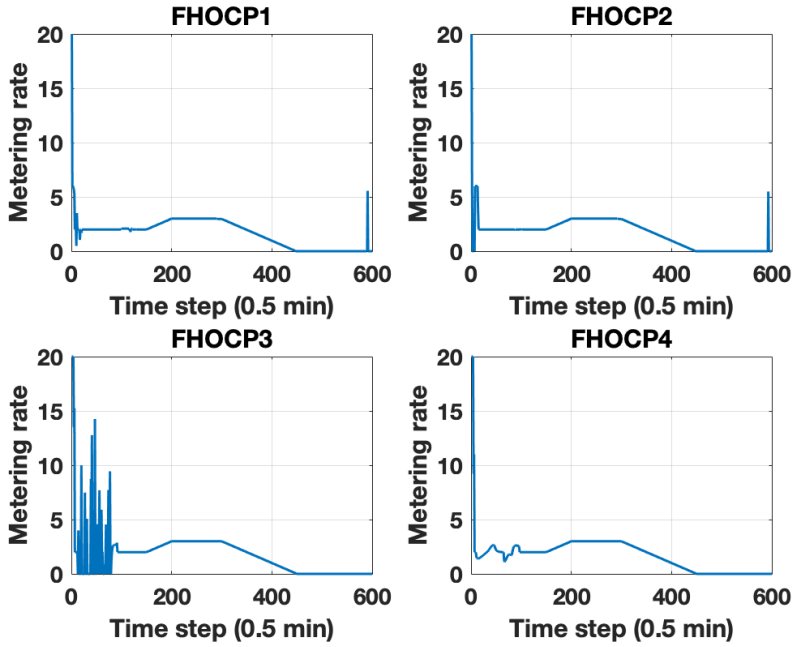


Figure 2.9: Metering rate of on-ramp o4 (veh/0.5 min) with the time-varying demand profile

2.5.5 Analysis of the Computation Times

To complete the analysis, computation time was also considered as an additional performance index. Table 2.4 provides the computation time of all problems under the two demand patterns applied. FHOCPs 1 and 2 had considerable higher computation times comparing to FHOCPs 3 and 4 with respect to both demand patterns. The reason is the complexity of extended ACTM, and to be more precise, due to the presence of *min* function in its equations. Also, FHOCPs 2 and 4 also had higher computation times in comparison with FHOCPs 1 and 3 because of the presence of the two penalty terms in their cost functions. To summarize, the extended CTM and the cost functions with penalty terms can

significantly increase the computation times.

Table 2.4: Comparison of the computation times

Problem	**Computation Time (sec)**	
	Stationary	Time-varying
FHOCP 1	165.3497	78.6285
FHOCP 2	324.0605	89.8481
FHOCP 3	11.4909	12.9580
FHOCP 4	28.0055	34.4486

2.6 Conclusion

This chapter provides a thorough analysis over the performance of different ACTM versions used as the underlying traffic flow models for the proposed FHOCPs. The FHOCPs had different cost functions, but for all of them, the PI-ALINEA RM metering control was applied. All 4 FHOCPs had promising performances regarding the convergence of flow rates to the theoretical equilibrium flow vector, and the metering rate of on-ramps and how they can follow the changes of origin demands. Simulation results have shown that using the linear relaxed version of the ACTM is motivated by computational consequences. The extended ACTM can show better capabilities in modeling the traffic flow variables and responding to the traffic control applied. However, it is complicated, and therefore it decreases the computational efficiency of the optimization scheme.

Chapter 3: PWA-CTM: An Extended Cell-Transmission Model based on Piecewise Affine Approximation of the Fundamental Diagram

3.1 Motivation

Throughout the past decades, many different versions of the widely used first-order Cell-Transmission Model (CTM) have been proposed for optimal traffic control. Highway traffic management techniques such as Ramp Metering (RM) are typically designed based on an optimization problem with nonlinear constraints originating in the flow-density relation of the Fundamental Diagram (FD). Most of the extended CTM models are based on the trapezoidal approximation of the flow-density relation of the Fundamental Diagram (FD) in an attempt to simplify the optimization problem. However, this relation is naturally nonlinear, and crude approximations can greatly impact the efficiency of the optimization solution. In this study, we propose a class of extended CTMs that are based on piecewise affine approximations of the flow-density relation such that (a) the integrated squared error with respect to the true relation is greatly reduced in comparison to the trapezoidal approximation, and (b) the optimization problem remains tractable for real-time application of ramp metering optimal controllers. A two-step identification method is used to approximate the

FD with piecewise affine functions resulting in what we refer to as PWA-CTMs. The proposed models are evaluated by the performance of the optimal ramp metering controllers, e.g. using the widely used PI-ALINEA approach, in complex highway traffic networks. Simulation results show that the optimization problems based on the PWA-CTMs require less computation time compared to other CTM extensions while achieving higher accuracy of the flow and density evolution. Hence, the proposed PWA-CTMs constitute one of the best approximation approaches for first-order traffic flow models that can be used in more general and challenging modeling and control applications.

3.2 Introduction

Optimal traffic control has been extensively studied since the beginning of the twentieth century, and is still a topic of ongoing research. Towards this direction, numerous traffic flow models have been developed and used to understand, describe, and predict traffic flow in different real-life scenarios, ranging from bi-directional roads to multi-lane highways with on-ramps [43–45].

Traffic flow models constitute dynamical models that can be classified as micro-, meso- or macroscopic models, depending on whether the model distinguishes the behavior of each individual vehicle or makes use of mean-field limit metrics [46]. In particular, the main advantage of macroscopic traffic models over microscopic models is the significantly lower computational costs due to lower complexity [47]. These models are typically described by the traffic density (average number of vehicles per unit length of road) and flow (average

number of vehicles per time unit) and can be further categorized with respect to the underlying model dynamics into first order or second order models. The most frequently used macroscopic traffic flow models are first order models, such as the Lighthill-Whitham-Richards (LWR) model [22], which is a continuous model, and the Cell-Transmission Model (CTM) [23] which is a discretized version of the LWR model. The simplicity and computational benefits of the CTM model has given rise to many CTM extensions, each one proposed based on different criteria and the specific needs of different applications. A review of the different CTM extensions and versions proposed over the years can be found in [9]. In [48], two of the mostly used CTM versions, namely the linear relaxation of CTM [25] and the extended CTM [9] are thoroughly evaluated. Throughout the manuscript, we will refer to these models as the *Relaxed* and the *Extended* CTM versions, respectively.

Many highway traffic management techniques, such as autonomous traffic signal control (Ramp Metering) [43, 44], are formulated as optimization problems with constraints originating in this flow-density relation of the fundamental diagram. However, in order to reduce the complexity of such methods, most of the CTM versions are based on triangular [24] or trapezoidal approximation [4] of the flow-density relation of the fundamental diagram. The trapezoidal FD is more commonly used in the literature and is shown in Fig. 3.1. Due to the fact that the flow-density relation is naturally a nonlinear relation (Fig. 3.1), crude approximations of the fundamental diagram can greatly impact the efficiency of the optimization solution, and, as a result, the performance of the traffic control methodology. On the other hand, control algorithms that take into account the original non-linear FD,

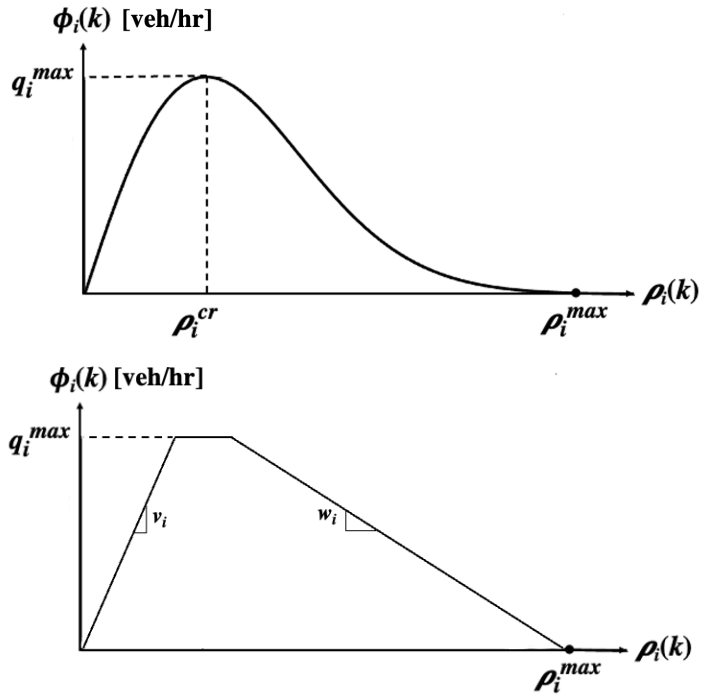


Figure 3.1: The original (non-linear) Fundamental Diagram (FD) [3] *(top)* and its trapezoidal approximation [4] *(bottom)*.

are computationally expensive and typically not suited for real-time application. In [48], it was shown that optimization problems based on the *Relaxed CTM* did indeed have computational advantages, while finite horizon optimal control problems (FHOCPs) based on the more complicated *Extended CTM* model showed better traffic control performance but reduced computational efficiency. In other words, there exists a trade-off between performance and complexity for the CTM-based traffic control algorithms, which has its roots on the approximation accuracy of the fundamental diagram.

In this work, we propose a class of extended CTMs that are based on piecewise affine approximations of the flow-density relation of the fundamental diagram (FD) such that:

(i) the integrated squared error with respect to the true relation is greatly reduced in

comparison to the trapezoidal approximation, and

(ii) the optimization problem remains tractable for real-time application of ramp metering optimal controllers.

A two-step identification method is used to approximate the FD with piecewise affine functions resulting in new extensions of the CTM that we will refer to as *PWA-CTM* traffic flow models. The proposed models are evaluated in the framework of finite horizon optimal control problems using ramp metering controllers, e.g. the widely used PI-ALINEA approach, in complex highway traffic networks. Hence, the proposed PWA-CTM models constitute one of the best approximation approaches for first-order traffic flow models that can be used in any finite horizon optimal control problem to enhance its performance regarding the computational time and the convergence to the equilibrium state.

The chapter is organized as follows: The CTM traffic flow model, different flow-density relations of the FD, and the ramp metering control scheme used in this study are briefly defined in Section 3.3. In Section 3.4, a two-step piecewise affine identification of nonlinear systems is described and applied to the nonlinear flow-density relation of the FD to find its PWA approximation. Also, the new CTM extensions based on this approximation of the FD are proposed in this section. The formulation of the FHOCPs are explained in Section 3.5 and in Section 3.6 the simulation results are reported and analysed in detail. Finally, some conclusive remarks are drawn in Section 3.7.

3.3 Preliminaries

In this section, we briefly introduce the Cell-Transmission Model (CTM), illustrate the importance of the fundamental relation between flow and density that constitutes the Fundamental Diagram (FD), and define a broadly used Ramp Metering (RM) control method to be used in our experiments.

3.3.1 The CTM model

The Cell Transmission Model (CTM) is a popular numerical method originally proposed by Carlos Daganzo [23, 24] to solve the kinematic wave equation. CTM models can be used to predict the macroscopic traffic behavior on a given road lane, by evaluating the traffic flow ϕ_i and density ρ_i at a finite number of intermediate points that result from dividing the lane into homogeneous sections $i \in [1, 2, \ldots, N]$ (hereafter called cells), as shown in Fig. 2.2. All notations used in this work, including all the model variables and parameters, are explained in the List of Symbols and the schematic of Fig. 2.2. Detailed explanation about the notation can be found in [9] and [48].

Initial and boundary conditions are appropriately defined to iteratively evaluate the values of the quantities at each cell. The flow across the cells is determined based on ϕ_i and ρ_i, which are the two monotonic functions that uniquely define the fundamental diagram (FD) as shown in Fig. 3.1. The update equations are given by:

$$\rho_i(k+1) = \rho_i(k) + \frac{T}{L}(\Phi_i^+(k) - \Phi_i^-(k)) \tag{3.1}$$

$$\Phi_i^+(k) = \phi_i(k) + r_i(k) \tag{3.2}$$

$$\Phi_i^-(k) = \phi_{i+1}(k) + s_i(k) \tag{3.3}$$

$$s_i(k) = \frac{\beta_i(k)}{1-\beta_i(k)}\phi_{i+1}(k) \tag{3.4}$$

$$l_i(k+1) = l_i(k) + T(d_i(k) - r_i(k)) \tag{3.5}$$

For uncontrolled on-ramps:

$$r_i(k) = d_i(k), \quad 0 \leq r_i(k) \tag{3.6}$$

and, for controlled on-ramps:

$$r_i(k) \leq l_i(k) + d_i(k), \quad 0 \leq r_i(k) \leq r_i^{C,max} \tag{3.7}$$

The boundary conditions are given by:

$$0 \leq \rho_i(k) \leq \rho_i^{max}(k) \tag{3.8}$$

$$0 \leq \phi_i(k) \leq q_i^{max}(k) \tag{3.9}$$

$$0 \leq l_i(k) \leq l_i^{max} \tag{3.10}$$

The flow variable $\phi_i = f(\rho_i)$ is given by the fundamental diagram (FD) which will be described in the next section. For a detailed overview of the model, the readers are referred to [9].

Table 3.1: Model Variables and Parameters

Symbol	**Description/Unit (Range)**
v_i^f	Free-flow velocity of cell i / (km/h)
a_i	A parameter to be estimated from empirical data
K_R	Integral regulator gain
K_P	Proportional regulator gain

3.3.2 Fundamental Diagram and the Trapezoidal Approximation

A key feature of the CTM is the assumption that there is some fundamental relation between the density and the flow variables, i.e.,

$$\phi_i = f(\rho_i) \tag{3.11}$$

This relation is known as the Fundamental Diagram (FD). Three families of this functional form have been proposed in the past decades: 1) Linear forms first proposed by Greenshields [49], 2) Nonlinear forms using logarithmic [50] or exponential form [3, 51], and 3) Multi-regime forms which use rather simple functional forms, e.g. linear or quadratic functions, for intervals of density [52], [53]. The interested reader can find a precise review of FDs in [4] and [54]. Equation (3.12) reads the exponential flow-density relation (Fig. 3.1).

$$\phi_i(k) = \rho_i(k) * v_i^f \exp[-\frac{1}{a_i}(\frac{\rho_i(k)}{\rho_i^{cr}})^{a_i}] \tag{3.12}$$

In some models, a different steady-state relation is considered for each cell i. The trapezoidal FD (Fig. 3.1) belongs to the third family of FD models and is described by eq. (3.13).

$$\phi_i(k) = \min\{v_{i-1}^f(\rho_{i-1}(k) + r_{i-1}(k)), q_{i-1}^{max}, w_i(\rho_i^{max} - \rho_i(k) - r_i(k))\} \tag{3.13}$$

In this study, the goal is to find a linear approximation of the FD which would have closer behavior to the nonlinear relation (3.12) but would make the constraints of optimization-based problems linear as the trapezoidal approximation (3.13) does.

3.3.3 Ramp Metering Control

Ramp metering (RM) is achieved by placing traffic signals at on-ramps to control the flow rate at which vehicles enter the freeway. The ramp metering controller computes the metering rate to be applied. For the purposes of this chapter, we are going to use the

feedback local ramp metering strategy PI-ALINEA, proposed by [40]. PI-ALINEA is a proportional-integral feedback control algorithm, in which the metering rate is given by

$$r_i^C(k) = r_i^C(k-1) + K_R[\rho_i^* - \rho_i(k)] - K_P[\rho_i(k) - \rho_i(k-1)] \tag{3.14}$$

where the flow that can enter section i of a freeway from the on-ramp of cell i during time interval $[kT,(k+1)T)$ is represented by $r_i^C(k)$. In case the main objective of the traffic controller is to reduce congestion and to maximize the throughput (see Section 3.5), a good choice for the set-point is $\rho_i^* = \rho_i^{cr}$.

3.4 Two-Step Piece-wise Affine Identification of Nonlinear Systems

In this section we consider a nonlinear function $f : \Omega \to \mathbb{R}$, $\Omega \subseteq \mathbb{R}$, and describe a two-step optimization-based approach to find a piece-wise affine (PWA) function:

$$\tilde{f} = \sum_{i=1}^{M} \mathbb{1}_{[x \in R_i]}(a_i x + c_i) \tag{3.15}$$

such that the approximation error $e = \|f - \tilde{f}\|^2$ is minimized. Here, the regions $\{R_i\}_{i=1}^M$ define a partition of Ω, $\mathbb{1}_{[\cdot]}$ represents the indicator function, and $\|f\|^2 = \int_\Omega |f(x)|^2 dx$ denotes the L^2 norm. The optimal parameters $\{a_i\}_{i=1}^M$, $\{c_i\}_{i=1}^M$, and $\{R_i\}_{i=1}^M$, are the output of the optimization process that minimize the approximation error e.

Following the two-step identification process described in [55], we split the search for the PWA function $\tilde{f}$ into two steps. In the first stage we will approximate the function f

with an analytic function $\hat{f} : \Omega \to \mathbb{R}$ of known form and lower complexity, by minimizing the error $\hat{e} = \|f - \hat{f}\|^2$. In the second step, an optimization-based procedure will be applied to derive an optimal PWA approximation $\tilde{f}$ such that the approximation error $\tilde{e} = \|\hat{f} - \tilde{f}\|^2$ is minimized. Assuming that $\Omega = [\underline{k}, \bar{k}]$, the partition $\{R_i\}_{i=1}^{M}$ of M non-overlapping parts is described by the regions $R_1 = [\underline{k}, r_1]$, $R_2 = [r_1, r_2]$, ..., $R_M = [r_{M-1}, \bar{k}]$, with $\bigcup_i R_i = [\underline{k}, \bar{k}]$. Then, the problem reduces to finding the slopes $a_i \in \mathbb{R}$, offsets $c_i \in \mathbb{R}$ and breakpoints $r_i \in \mathbb{R}$ such that the approximation error $\tilde{e}$ is minimized, i.e.

$$\min_{\{a_i, c_i, r_i\}} \sum_{i=1}^{M} \left(\int_{r_i - 1}^{r_i} (f(x) - (a_i x + c_i))^2 dx \right) \tag{3.16}$$

s.t.

$$\underline{k} \leq r_1 \leq \ldots \leq r_{M-1} \leq \bar{k} \tag{3.17}$$

$$a_i r_i + c_i = a_{i+1} r_{i+1} + c_{i+1}, \quad i = 1, \ldots, M-1 \tag{3.18}$$

To implement the first step, the exponential flow-density function $\phi_i = f(\rho_i)$ is approximated by a polynomial function of different degrees $n \in \{3, \ldots, 6\}$. Considering the trade-off between the complexity of the resulting polynomials and the mean squared error (MSE) of the approximation, the optimal polynomial degree was chosen as $n = 5$ for this step. For reproducibility and future reference, we provide the coefficients of the polynomial

approximation in eq. (3.19) below:

$$\phi_i(k) = \hat{f}_{p_5}(\rho_i(k)) = \sum_{j=0}^{5} c_j \rho_i(k)^j \tag{3.19}$$

where $\{c_j\}_{j=0}^{5} = \{373.77, 1249.24, 1110.26, 95.83, 436.69, 118.43\}$. The approximation function $\hat{f}$ is compared against the flow-density relation f in Fig. 3.2, as well, where we have set $\Omega = [0, 180]$.

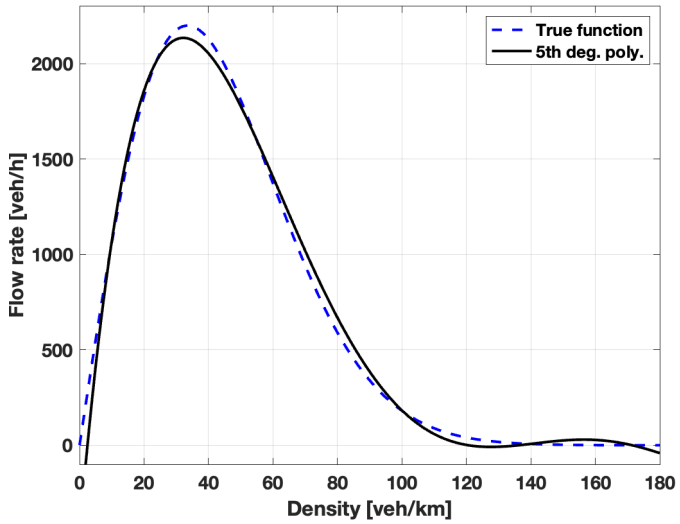


Figure 3.2: 5th-degree polynomial approximation of the FD.

For the second step, the optimization problem in (3.16), (3.17), (3.18) was solved, and the optimal, in terms of MSE, number of pieces $M \in \{5, 6, 7\}$ was chosen. The MSE of the approximations of the flow-density function are provided in table 3.2. Interestingly, in this problem, increasing the number of pieces does not necessarily translate to a similar decrease in the MSE (see table 3.2). The approximation with $M = 6$ pieces showed higher MSE. As a result, for this phase, the approximations with $M = 5$ and $M = 7$ pieces will

be considered in the experimental section that follows to develop new CTMs. Figure 3.3 shows the comparison between these PWA functions and the true fundamental diagram. Also, for reproducibility, the coefficients $\{a_i\}_{i=1}^{M}$, $\{c_i\}_{i=1}^{M}$, and $\{r_i\}_{i=1}^{M-1}$ of the functions $\phi_i = \tilde{f}_5(\rho_i)$, and $\phi_i = \tilde{f}_7(\rho_i)$ are provided in equations (3.20) and (3.21), respectively.

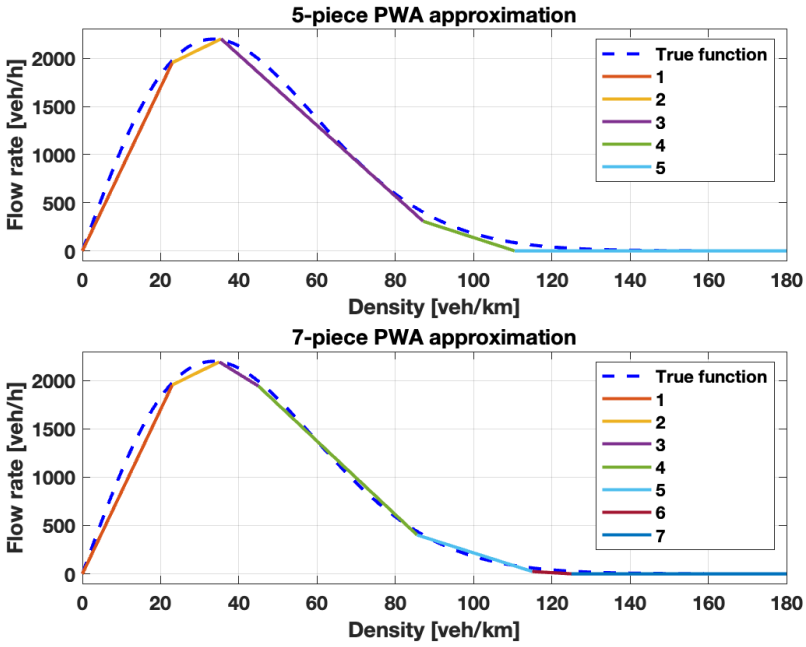


Figure 3.3: 5- and 7-Piece PWA approximation of the FD.

Table 3.2: MSE of different PWA approximations.

Piece Number	MSE
M=5	6433.14
M=6	7007.30
M=7	4375.17

$$\tilde{f}_5(\rho_i) = \begin{cases} 85.00 * \rho_i, & 0 \leq \rho_i \leq 23.00 \\ 19.78 * \rho_i + 1499.98, & 23.00 \leq \rho_i \leq 35.45 \\ -36.66 * \rho_i + 3501.06, & 35.45 \leq \rho_i \leq 87.12 \\ -13.12 * \rho_i + 1450.00, & 87.12 \leq \rho_i \leq 110.50 \\ 0, & \text{otherwise} \end{cases} \tag{3.20}$$

$$\tilde{f}_7(\rho_i) = \begin{cases} 84.99 * \rho_i, & 0 \leq \rho_i \leq 23.00 \\ 19.99 * \rho_i + 1490.00, & 23.00 \leq \rho_i \leq 35.00 \\ -20.00 * \rho_i + 2890.00, & 35.00 \leq \rho_i \leq 45.00 \\ -37.99 * \rho_i + 3700.00, & 45.00 \leq \rho_i \leq 93.57 \\ -13.90 * \rho_i + 1450.00, & 93.57 \leq \rho_i \leq 100.00 \\ -2.40 * \rho_i + 300.00, & 100.00 \leq \rho_i \leq 124.00 \\ 0, & \text{otherwise} \end{cases} \tag{3.21}$$

In the first step, the exponential flow-density function $\phi_i(k)$ can be approximated by any smooth regression function, e.g. a polynomial function or a feed-forward neural network. It is important to note that this step can be implemented as a data-based optimization algorithm, where actual traffic flow data are used to estimate the optimal fundamental diagram for the road and application at hand. This leads to learning-based traffic control approaches, which lie beyond the scope of this study.

3.4.1 The Piecewise Affine Cell-Transmission Models (PWA-CTMs)

Depending on the PWA approximation of the FD used, the CTM described in Section 3.3.1 can be extended to the models *PWA5-CTM* and *PWA7-CTM*, each using a PWA approximation with $M = 5$ and $M = 7$ pieces, respectively. The fundamental flow-rate relation $\phi_i = f(\rho_i)$, is now approximated by eq. (3.20) for the PWA5-CTM, and by eq. (3.21) for the PWA7-CTM model.

3.5 Finite Horizon Optimal Control Formulation

In this section, four Finite Horizon Optimal Control Problems (FHOCPs) with different underlying traffic flow models are formulated to investigate the evolution of traffic flow in each case. The traffic flow models used here define four different versions of the CTM. Two of the versions used are the Extended CTM [9] and the Relaxed CTM [25]. A thorough assessment on these two models is presented in [48]. The other two CTM versions are the PWA-CTMs proposed in the previous section. The traffic control strategy applied here is the PI-ALINEA ramp metering controller, and the analysis is performed by simulation for a hypothetical network of freeways with stationary demand patterns using the unique equilibrium state of each pattern.

Before developing the formulations of the FHOCPs, first the definitions of two objective functions will be provided. In the formulation of FHOCPs presented in this section, a linear combination of these cost functions will make the final objective of each problem. The most

widely used objective in traffic management is to minimize the total time that all vehicles spend in the network (i.e., the Total Time Spent or TTS). Basically, the TTS is the time spent by all vehicles in the network (i.e., the total travel time or TTT), including the waiting time experienced at origins (i.e., the total waiting time or TWT).

$$J_1 = J_{TTS} = T \sum_{k=0}^{K-1} [\sum_{i=1}^{N} \rho_i(k) L_i \lambda_i + \sum_{i=1}^{N} l_i(k)] \tag{3.22}$$

The second objective function applied here is to maximize the sum of the traffic flows going through all sections and on-ramps. This objective function is also called the Total Travel Distance (TTD) since it is the total distance (veh mi) covered by all the vehicles in the considered time horizon.

$$J_2 = J_{TTD} = T \sum_{k=0}^{K-1} [\sum_{i=1}^{N} \phi_i(k) L_i + \sum_{i=1}^{N} r_i(k) L_i] \tag{3.23}$$

A linear combination of these two objective functions is used as the objective function to express their priorities, i.e.

$$\min_{r_i^C} \alpha_1 J_1 - \alpha_2 J_2 \tag{3.24}$$

Here, $\alpha_1 > \alpha_2$ as minimizing the travel time is the main objective. Also, the α_i values are the same for all four problems.

The different FHOCP problems are defined by optimization problem (3.24), the controller (3.14), the constraints (3.1)-(3.10), and different flow-density relationship $\left\{\phi_i = f_j(\rho_i)\right\}_{j=1}^{4}$ given by:

FHOCP 1: $\phi_i = f_1(\rho_i) \sim$ Extended CTM [9, 48];

FHOCP 2: $\phi_i = f_2(\rho_i) \sim$ Relaxed CTM [25, 48];

FHOCP 3: $\phi_i = f_3(\rho_i) \sim$ (3.20); and

FHOCP 4: $\phi_i = f_4(\rho_i) \sim$ (3.21).

3.6 Simulation Results

In this section, we evaluate the performance and complexity of the proposed methodology on a complex simulation environment.

3.6.1 Case study and model parameters

Simulation is performed for the network shown in Fig. 2.3 with two origins ($o1$ and $o2$), two controlled on-ramps ($o3$ and $o4$), 12 mainline links ($m1$ through $m12$) and two destinations ($d1$ and $d2$). One assumption about the network is that the proportion of turns at every junction, i.e. the split ratios $\beta_i(k)$, are fixed and known in advance. Also, it is assumed that the behavior of all the links can be described by parameters shown in table 3.3 adopted from [41, 48] and converted to SI units for this study:

Table 3.3: Model Parameters.

Symbol	Value	Unit/Range
Period T	0.5	min
Length L_i	1	km
v_i^f	1	length/period
w_i	0.33	length/period
ρ_i^{max}	180	veh/length
ρ_i^{cr}	50	veh/length
q_i^{max}	50	veh/period

3.6.2 Simulation and Numerical Results

The results of the simulation of the four FHOCPs proposed in section 3.5 are provided in this section. The FHOCPs were solved with Yalmip, the modeling and optimization language offered by [42], and the GUROBI non-commercial optimization solver via the interface of MATLAB. The simulation horizon of 1 hour is considered by choosing the time horizon of K = 120 time steps and the simulation period of T = 0.5 minute ($K * T$ = 60 minute). For the simulation phase of this study, the following stationary demand vector was applied for origin links $o1$ and $o2$ and also on-ramp links $o3$ and $o4$: $d(k) = (8,9,7,3)$ *veh*/0.5 *min*. According to [41], for each stationary demand vector $d(k) = (d_0, ..., d_M)$, there exists a unique equilibrium flow rate $\phi(k) = (q_0, ..., q_N)$ and density vector $\rho(k) = (\rho_0, ..., \rho_N)$. Detailed explanation on the calculation of these vectors is explained in [41]. Table 2.2 shows the theoretical equilibrium flow vector for the mainline links of this network based on the chosen stationary demand. Adding all these values together, the network flow rate at the equilibrium state should be equal to 113.37 *veh*/0.5

min. Knowing this theoretical flow vector and having the overall value of the network flow rate provide an insight on what is expected to see in the simulation phase.

Figure 3.4 shows the comparison of the network flow rate, i.e. the summation of the flow rate of all the links, in all four FHOCPs during the 1-hour simulation horizon. In all problems, the network flow rate is showing convergence to the value of the theoretical equilibrium equal to 113.37 *veh*/0.5 *min*. However, the FHOCPs based on the PWA-CTM are showing faster convergence with much less error at the beginning of the simulation. The interesting result is that the network flow rate of the last two problems look to be the completely the same. To have a closer look at this comparison, the boxplot of the difference between the network flow rate of these two problems is provided in Fig. 3.5. It is clear that this difference is so small that can be neglected without any loss of generality.

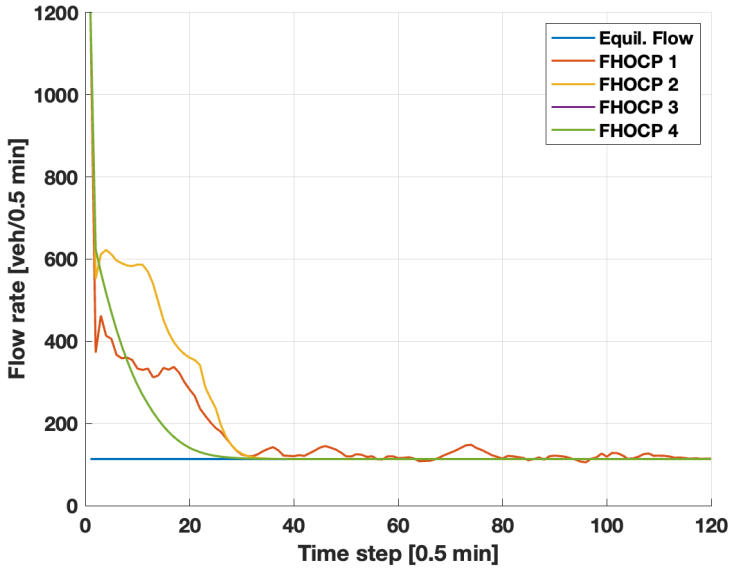


Figure 3.4: Comparison of the network flow rate in all four problems.

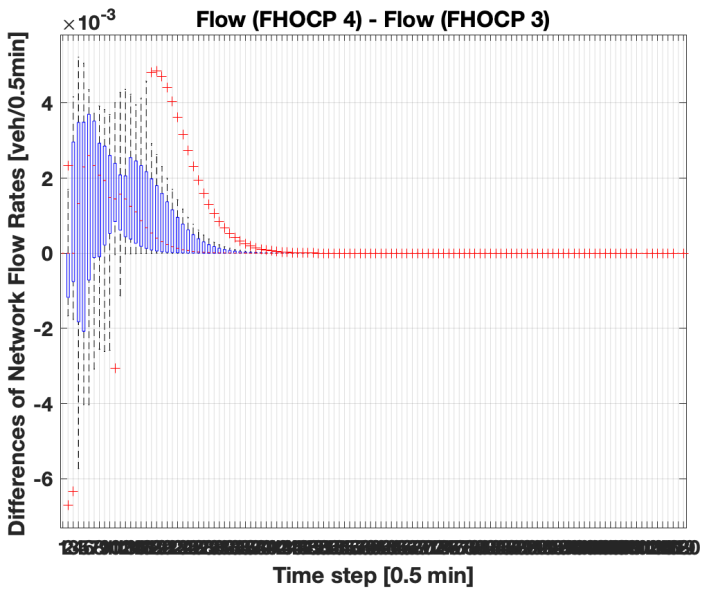


Figure 3.5: Boxplot comparison of the network flow rate in problems based on the PWA-CTMs.

Table 3.4: Comparison of the Computation Times and Iteration Numbers.

Problem	**Computation Time (sec)**	**Iterations**
FHOCP 1	245.88	618713
FHOCP 2	1.53	4212
FHOCP 3	1.62	746
FHOCP 4	1.64	749

Another factors used to evaluate the performance of these problems is the computation time and the number of iterations needed to solve each of the four problems. Table 3.4 shows the numerical value of these two factors. The value of these factor for the problem based on the Extended CTM is highly greater than those of the other problems. The reason is the large number of the nonlinear *min* function constraints in this model [48]. Also, considering the computation time of the other three problems, it can be seen that they almost have similar values, however, still the number of iterations needed to find a solution

for the problems based on the PWA-CTM is much less and indicates that these optimization problems remain more tractable for real-time application of ramp metering optimal controllers.

The evolution of the ramp metering variables are also investigated here. Fundamentally, the expectation is that the control applied to the on-ramps can satisfy the requirements of the network while considering the external demand of the on-ramps by not imposing too much waiting time to the vehicles on the on-ramps. As a result, a proper RM variable basically should replicate the overall demand pattern of that on-ramp. As an example, Fig. 3.6 shows the RM variable of on-ramp $o3$. In all four problems, the control variable is trying to converge to the value of the external demand applied (a constant demand of 7 *veh*/0.5 *min*). However, the RM variable in problems based on PWA-CTM have smoother and faster convergence with much less fluctuations while the control variable in the problem based on the Extended CTM never actually converges and in the problem based on the Relaxed CTM, a big overshoot happens before it starts to converge.

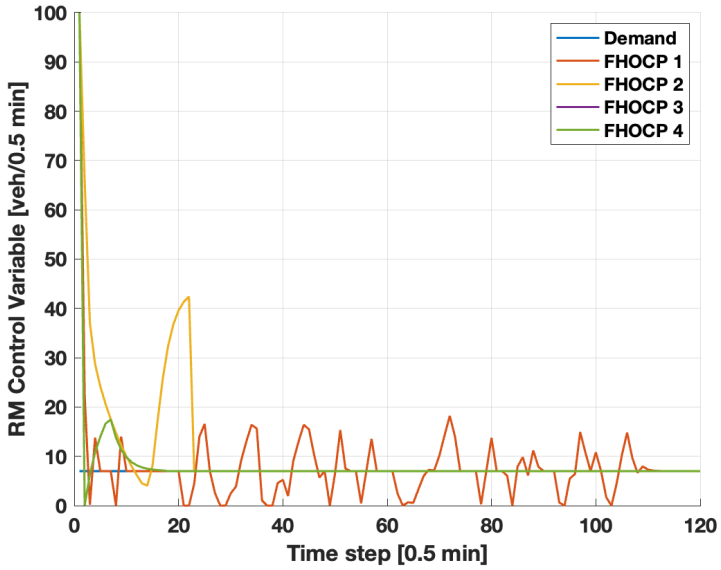


Figure 3.6: Evolution of the RM control variable of on-ramp $o3$.

Last but not least, the MSE between the network flow rate and density of the problems is compared with that of the equilibrium state. The results are provided in Fig. 3.7. The problem based on the Relaxed CTM has the highest MSE for both the flow rate and density of the network. It is because of the crude simplifications made in this version of the CTM [48]. Comparing the MSE of the problem based on the Extended CTM and the performance of the problems based on the PWA-CTM, the results show less MSE for the last two FHOCPs. It indicate the FHOCPs based on the PWA-CTM have less error comparing to the equilibrium values of the flow rate and density.

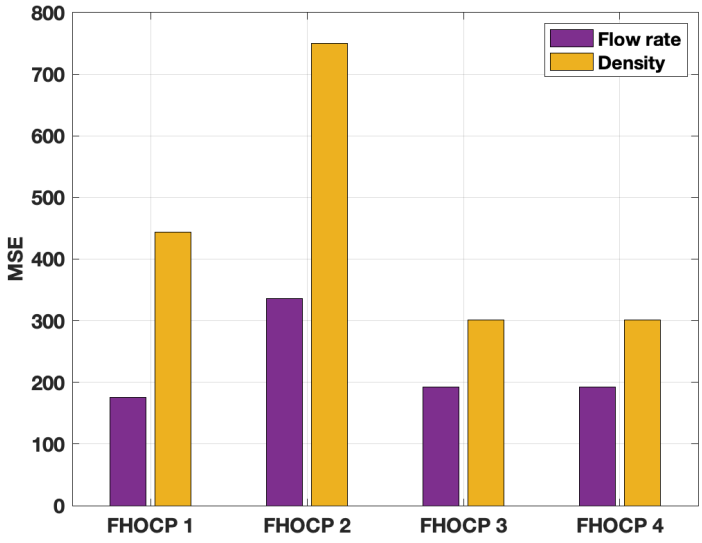


Figure 3.7: Bar plot comparison of the MSE of the network flow rate and density.

3.7 Conclusion

In this chapter, two enhanced CTM versions are proposed which constitute the PWA approximations of the flow-density relation of the FD. Most of the other extensions of CTM are based on the trapezoidal approximation of this relation in order to simplify optimization problems based on them. However, the flow-density relation is naturally nonlinear, and primitive approximations highly affect the efficiency of the optimization solution. The PWA approximation proposed here is computed based on a two-step identification method to minimize the integrated squared error with respect to the true relation. The so-called PWA-CTM versions are evaluated in the framework of FHOCPs for a hypothetical highway network using the PI-ALINEA RM strategy. Also, for better evaluation, the performance of these models are compared with that of two other widely used versions of the CTM in the

same optimization framework. Simulation results of these FHOCPs show that the problems based on the PWA-CTM models require less computation time and less iteration numbers compared to other problems. In addition, the problems based on the PWA-CTM models indicate higher accuracy of the flow and density evolution in comparison with the theoretical equilibrium state of the network. Also, the optimization problems based on the PWA-CTMs remains tractable for real-time application of ramp metering optimal controllers. Hence, the proposed PWA-CTMs constitute one of the best approximation approaches for first-order traffic flow models suitable for more challenging modeling and control applications.

Chapter 4: A Linear Approximation of the METANET model based on Piecewise Affine Approximation and Synthetic Data Generation of a Highway Network

4.1 Motivation and Related Work

For model-based optimal control of traffic networks one needs both a model that tracks the traffic states (flow, velocity, etc.) and an optimization approach that computes control variables yielding an optimal traffic performance. In the literature, the optimization schemes for highway traffic control which are based on the second-order traffic flow model METANET are highly nonlinear, computationally intensive and the solutions obtained are generally local optima. Such a model is commonly used as it has shown to provide good accuracy while it does not require as much computation time as microscopic models that take individual vehicles into account [56].

In this part of the study, as an alternative to the nonlinear METANET model in an optimization control framework, a linear approximation of the model will be proposed. The METANET model has different sources on nonlinearities. In contrast to what was discussed as the linearization approach for PWA-CTM, here, PWA approximations will not be able to address all the sources of nonlinearities by itself. As such, synthetic data generation method has been proposed to accompany the two-step identification method

to fully handle the nonlinearities of this model. For the purpose of traffic control, the linear approximation of the METANET model can be used in an optimization framework as the underlying traffic flow model. For a case study network, the accuracy of various approximations will be compared to the original nonlinear formulation of METANET.

4.2 The Standard METANET Model

The METANET model presented here is an improved version [57] of its original one first presented in [58]. However, the notation has been adopted from [9] in order to agree with the other notations of this dissertation. Detailed explanations of the variables and parameters of this model is provided in the List of Symbols.

Mainline Links

By the conservation law of vehicles we have,

$$\rho_i(k+1) = \rho_i(k) + \frac{T}{L_i\lambda_i}[\phi_{i-1}(k) - \phi_i(k) + r_i(k) - s_i(k)] \tag{4.1}$$

and the velocity equation is

$$v_i(k+1) = v_i(k) + \frac{T}{\tau}[V(\rho_i(k)) - v_i(k)] + \frac{T}{L_i}v_i(k)[v_{i-1}(k) - v_i(k)] - \frac{\nu T[\rho_{i+1}(k) - \rho_i(k)]}{\tau L_i[\rho_i(k) + \chi]} \tag{4.2}$$

where $i = 1, ..., N$, $k = 0, ..., K-1$.

The relaxation term,i.e. $\frac{T}{\tau}[V(\rho_i(k)) - v_i(k)]$, models the fact that vehicles tend to reach the steady-state velocity depending on the experienced density $\rho_i(k)$, according to a param-

eter τ , which represents the swiftness of drivers. Hence, vehicles accelerate if their actual velocity is lower than the steady-state value, and they decelerate otherwise.

The convection term, i.e. $\frac{T}{L_i}v_i(k)[v_{i-1}(k)-v_i(k)]$, represents the fact that vehicles arriving in section i from section $i-1$ cannot adapt immediately their velocity. If vehicles travel in section $i-1$ at a higher velocity than in section i , they decelerate when they reach section i but this change of velocity is not instantaneous. A similar argument applies in case of acceleration from section $i-1$ to section i.

The anticipation term, i.e. $-\frac{vT[\rho_{i+1}(k)-\rho_i(k)]}{\tau L_i[\rho_i(k)+\chi]}$, takes into account that drivers adjust their velocity also on the basis of the situation they see downstream, hence there is a deceleration if a higher density is seen ahead, and an acceleration in the opposite case.

Also, the traffic flow is given by

$$\phi_i(k) = \rho_i(k)v_i(k)\lambda_i \tag{4.3}$$

and the steady-state velocity–density relation is given by

$$V(\rho_i(k)) = v_i^f \exp[-\frac{1}{a_i}(\frac{\rho_i(k)}{\rho_i^{cr}})^{a_i}] \tag{4.4}$$

Origin Links

The dynamic equation of the on-ramp queue length of section i at time kT is

$$l_i(k+1) = l_i(k) + T[d_i(k) - r_i(k)] \tag{4.5}$$

Also , for the flow $r_i(k)$ entering the mainstream from the on-ramp, we have two cases:

For uncontrolled on-ramps:

$$r_i(k) = \min\{d_i(k) + \frac{l_i(k)}{T}, r_i^{max}, r_i^{max}\frac{\rho_i^{max} - \rho_{m,1}(k)}{\rho_i^{max} - \rho_i^{cr}}\} \tag{4.6}$$

For controlled on-ramps:

$$r_i(k) = \min\{d_i(k) + \frac{l_i(k)}{T}, r_i^{max}, r_i^{max}\frac{\rho_i^{max} - \rho_{m,1}(k)}{\rho_i^{max} - \rho_i^{cr}}, r_i^C(k)\} \tag{4.7}$$

where $r_i^C(k)$ comes from the RM control law.

4.3 The Linear Approximation of the METANET model

Given this nonlinear non convex model (METANET), any optimization problem based on this model will be difficult to solve quickly to optimality. For this reason the choice of a piecewise-affine (PWA) approximation of the nonlinear functions is made, which results in an optimization problem that can be more easily solved. Constructing a PWA formulation of the METANET model to be used in an optimization framework will make any optimization-based control problem formulations linear and this will greatly impact the computation time and efficiency of the optimal problems.

The METANET model explained in section 4.2 is a nonlinear model due to the following equations:

1. The traffic flow equation in eq. 4.3, where its non-linearity arises from the multiplication of mean velocity and density which makes the famous fundamental diagram as shown in Fig. 4.1 (bottom diagram). An extensive discussion about this diagram and a 5-piece and 7-piece PWA approximation of it was offered in equations 3.20

and 3.21 and shown in Fig. 3.3 in Chap. 3. As such, for this equation, the results offered before will be used accordingly.

2. The mean velocity equation in eq. 4.2 which has the following issues:

   (a) The relaxation term: the density variable arising in an exponential factor in eq. 4.4 (top figure in Fig. 4.1),

   (b) The convection term: multiplication of velocity variables,

   (c) The anticipation term: division of density variables by another density.

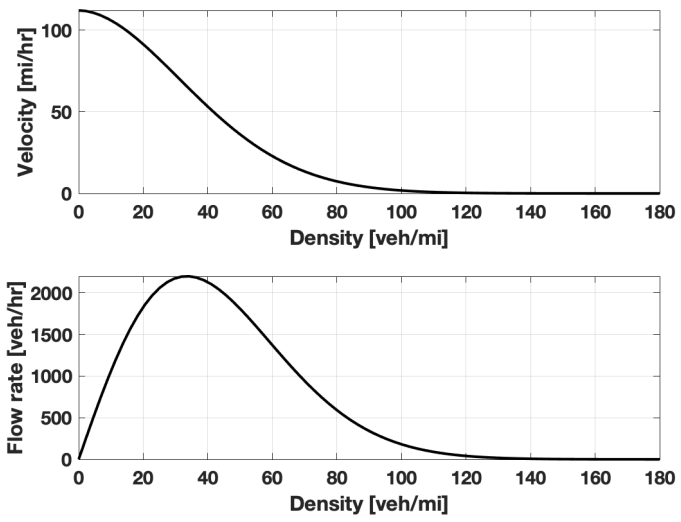


Figure 4.1: Nonlinear Fundamental Diagram.

In the remaining part of this chapter, further efforts will be made to resolve the remaining nonlinearity issues of the mean velocity equation. Then, in the framework of FHOCPs, this new linearized version of the METANET will be applied as the traffic flow model and the optimal problems will be solved to evaluate the performance of the proposed linear

METANET model.

4.4 Addressing the Nonlinearity of the Relaxation Term

In this part, the PWA approximation method mentioned in section 3.4 will be applied to approximate the nonlinear steady-state velocity relation $V(\rho_i(k)) = g(\rho_i(k))$ of the FD. This relation is shown in the top diagram in Fig. 4.1 and also stated in eq. 4.4). The nonlinearity source of this relation is due to the exponential function. The goal is to offer a 3-piece and 4-piece approximation of this exponential equation.

In [5], a 3-piece PWA approximation of the steady-state velocity has been offered. This approximation is provided in eq. 4.8. This will be used as a benchmark of comparison using the MSE with the original nonlinear equation.

$$\tilde{g}_{Groot}(\rho_i) = \begin{cases} -1.55 * \rho_i(k) + 112.17, & 0 \leq \rho_i(k) \leq 65.5 \\ -0.25 * \rho_i(k) + 26.61, & 65.5 \leq \rho_i(k) \leq 107.5 \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

This approximation is also shown in Fig. 4.2.

To implement the first step of the PWA identification method, the exponential velocity-density function was approximated by polynomial functions of degrees $n = \{3,4,5\}$. In this case, the MSE of polynomials of degree 4 and 5 did not show much lower value and using them would result in unnecessary complexity for step two. As such, the polynomial $n = 3$ was chosen for the result of step one. For reproducibility and future reference, we

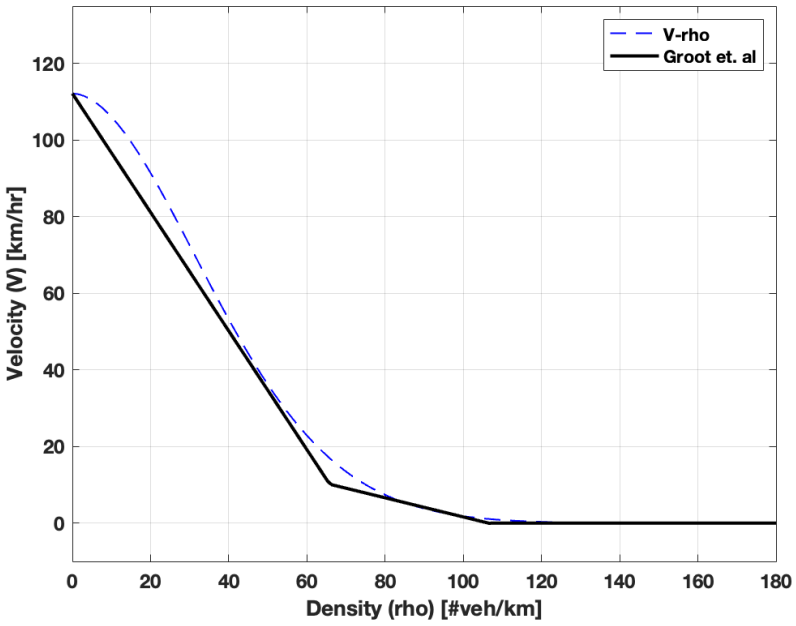


Figure 4.2: 3-piece PWA approximation of the velocity-density equation offered in [5].

provide the coefficients of the polynomial approximation in eq. 4.9 below:

$$\hat{g}_{p_3}(\rho_i(k)) = \sum_{j=0}^{3} c_j \rho_i(k)^j \tag{4.9}$$

where $\{c_j\}_{j=0}^{3} = \{5.78, -23.22, 20.39, -4.66\}$. The 3rd degree polynomial approximation function $\hat{g}$ is compared against the velocity-density relation g in Fig. 4.3, as well.

For the second step, the optimization problem in 3.16, 3.17, 3.18 was solved again for the velocity-density function g_i, and the optimal, in terms of MSE, number of pieces $M \in \{3,4\}$ was chosen. The MSE of the approximations of the velocity-density function are provided in table 4.1. It can be easily seen that the MSE of the approximations proposed here are much lower than that proposed in [5]. Also, Fig. 4.4 shows the comparison between these PWA functions and the true fundamental diagram.

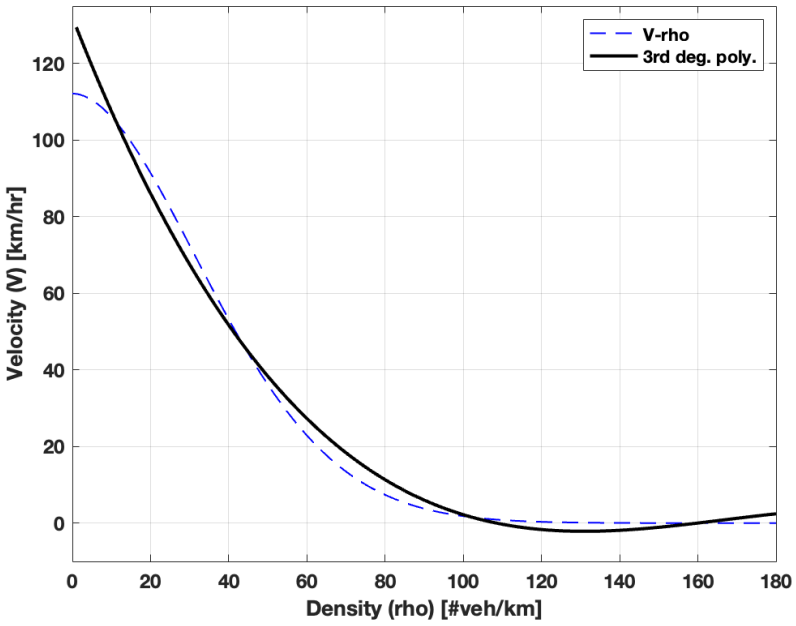


Figure 4.3: 3rd degree polynomial approximation of the velocity-density equation.

Table 4.1: MSE of different PWA approximations.

Number of Pieces	**MSE**
M=3 [5]	16.26
M=3	5.95
M=4	2.56

For reproducibility, the coefficients $\{a_i\}_{i=1}^{M}$, $\{c_i\}_{i=1}^{M}$, and $\{r_i\}_{i=1}^{M-1}$ of the functions $\tilde{g}_3(\rho_i)$, and $\tilde{g}_4(\rho_i)$ are provided in equations 4.10 and 4.11, respectively.

$$\tilde{g}_3(\rho_i) = \begin{cases} -1.53 * \rho_i(k) + 116.00, & 0 \leq \rho_i(k) \leq 70 \\ 0.12 * \rho_i(k) + 28.00, & 70 \leq \rho_i(k) \leq 101.00 \\ 0, & \text{otherwise} \end{cases} \tag{4.10}$$

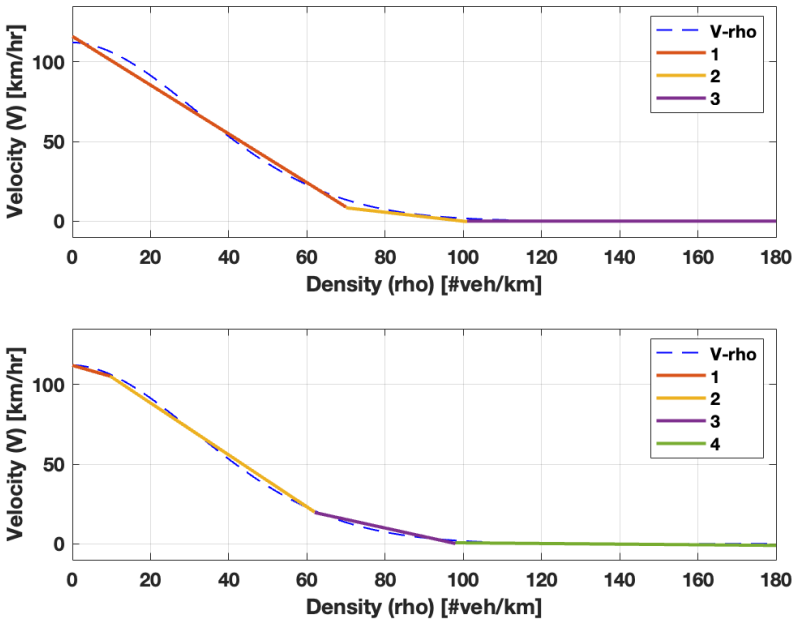


Figure 4.4: 3- and 4-Piece PWA approximation of the velocity-density equation.

$$\tilde{g}_4(\rho_i) = \begin{cases} -0.70 * \rho_i(k) + 112.00, & 0 \leq \rho_i(k) \leq 9.64 \\ -1.63 * \rho_i(k) + 121.01, & 9.64 \leq \rho_i(k) \leq 61.98 \\ -0.54 * \rho_i(k) + 53.13, & 61.98 \leq \rho_i(k) \leq 97.98 \\ -0.02 * \rho_i(k) + 2.63, & \text{otherwise} \end{cases} \tag{4.11}$$

The equations 4.10 and 4.11 are the proposed PWA approximation of the exponential velocity-density equation in this section and using them, the nonlinearity of the relaxation term will be resolved. As the MSE of the 4-piece approximation is lower, from this point on, this approximation will be used as the PWA approximation of eq. 4.4. At this point, we will move on to the next section to address the remaining nonlinearities of the mean

velocity equation.

4.5 Addressing the Nonlinearities of the Convection and Anticipation Terms

The convection term, i.e. $\frac{T}{L_i} v_i(k)[v_{i-1}(k) - v_i(k)]$, and the anticipation term, i.e. $-\frac{vT[\rho_{i+1}(k) - \rho_i(k)]}{\tau L_i[\rho_i(k) + \chi]}$ are the two nonlinearity sources left in the METANET model. In this section, a method will be offered to address these issues. Both of these terms are bivariate functions depending on two state variables. In the convection term, we have the multiplication of the mean velocity of cells $i-1$ and i variables. In the anticipation term, we have the division of the density variables of cells $i+1$ and i.

To address these types of nonlinearties, PWA approximation alone could not do the job of approximation as these two functions are bivariate functions and the mean velocity depends on more than one variable in these terms. As such, to handle this problem, we needed to take care of the extra variable present in each of these two terms. In the convection term, for $v_i(k+1)$, the variable $v_{i-1}(k)$ causes the trouble and in the anticipation term, for $v_i(k)+1$, the variable $\rho_{i+1}(k)$ is the source of difficulty. The goal is to find a way to take care of these variables. Then, the convection and the anticipation become one-variable functions and the PWA approximation method would give a linear approximation for these terms. To tackle this problem, the method of approximation by *piecewise constant functions* will be used which is explained in the next section.

4.5.1 Synthetic Data Generation

To handle the nonlinearity of the convection and anticipation terms, the method used in [5] is to simply keep the velocity term $v_{i-1}(k)$ and the density term in the denominator $\rho_i(k)$ constant at a value determined by historical data. However, in this study, we would to find a better approximation for these terms. Here, synthetic data will be generated for the network under consideration. Then, for each time step of the simulation, the convection and anticipation functions will be computed based on this data. Then, polynomial curve fitting method will be adapted to approximate these bivariate nonlinear functions with polynomial single-variable functions. After this step, the previously-introduced PWA approximation method will be used to find linear/affine functions for these two terms.

The following notations will be used for the convection term and anticipation term for the rest of this chapter: From the mean velocity equation $v_i(k+1)$ in eq. 4.2, and the part called the convection term, we define:

$$p(v_i(k)) \coloneqq v_i(k)[v_{i-1}(k) - v_i(k)] \tag{4.12}$$

which is dependent on $v_{i-1}(k)$ and $v_i(k)$. And from the anticipation term, we define:

$$q(\rho_i(k)) \coloneqq \frac{[\rho_{i+1}(k) - \rho_i(k)]}{[\rho_i(k) + \chi]} \tag{4.13}$$

which is dependent on $\rho_{i+1}(k)$ and $\rho_i(k)$.

In order to find linear approximations of $p(v_i(k))$, we take the following two steps:

- Approximate $p(v_i(k))$ with a polynomial curve , called $\hat{p}(v_i(k))$, fitted to function

resulted from synthetic data in a way that this function is only dependent only on one variable $v_i(k)$ such that the curve fitting error $\hat{p} = \|p - \hat{p}\|^2$ has the minimum value.

- Find the PWA approximation $\tilde{p}(v_i(k))$ based on the least squared method described in Chap. 3, such that $\tilde{e} = \|\hat{p} - \tilde{p}\|^2$ is minimized.

The same process will be repeated for $q(\rho_i(k))$, by finding the $\hat{q}(\rho_i(k))$ and $\tilde{q}(\rho_i(k))$

The process of the synthetic data generation, polynimial curve fitting of functions $p(v_i(k))$ and $q(\rho_i(k))$ and their PWA approximations is explained in the next section.

Eclipse SUMO traffic simulator

In this section, the process of generating synthetic data for the hypothetical network of this dissertation (Fig. 2.3) will be explained. For this process, Eclipse SUMO traffic simulator [59] will be used. SUMO is a free and open source traffic simulation suite. It is available since 2001 and allows modelling of intermodal traffic systems - including road vehicles, public transport and pedestrians. Included with SUMO is a wealth of supporting tools which automate core tasks for the creation, the execution and evaluation of traffic simulations, such as network import, route calculations, visualization and emission calculation. SUMO can be enhanced with custom models and provides various APIs to remotely control the simulation. SUMO is highly portable, microscopic and continuous multi-modal traffic simulation package designed to handle large networks.

To run a simulation with SUMO, three input files in the following order are needed:

- Road Network: For a simulation, a SUMO Road Network must be given using the

option –net-file. The network is normally built using *netconvert* or *netgenerate*.

- Traffic Demand (Routes): The vehicles to simulate must be given. Their description normally includes vehicle types, vehicles, and vehicle routes. Routes are normally given to the simulation modules using the option –route-files.

- Additional Files: All additional structures/definitions are given to the simulation using the –additional-files. Each file is read completely into memory and the list of files is processed from left to right. One or more additional-file(s) are used to load additional entities:

  - infrastructure related things: traffic light programs, induction loops and bus stops
  - additional visualization: POIs and polygons (i.e. rivers and houses)
  - dynamic simulation control structures: variable speed signs and rerouters
  - demand related entities: vehicle types and routes

Vehicles, Routes and Flows

Initially, we will define a vehicle with a route that belongs to it. A vehicle may be defined using the different attributes. Some of these attributes are:

- id: The name of the vehicle
- type: The id of the vehicle type to use for this vehicle
- route: The id of the route the vehicle shall drive along

- depart: The time step at which the vehicle shall enter the network
- departLane: The lane on which the vehicle shall be inserted
- etc.

It is possible to define repeated vehicle emissions (”flow”s), which have the same parameters as the vehicle or trip definitions except for the departure time. They are distributed either equally or randomly in the given interval. The following additional parameters are known:

- begin: first vehicle departure time
- end: end of departure interval (if undefined, defaults to 24 hours)
- vehsPerHour: number of vehicles per hour, equally spaced (not together with period or probability)
- etc.

Also, the attributes of a route are:

- edges: The edges the vehicle shall drive along, given as their ids, separated using spaces
- color: This route’s color
- cycleTime: When defining a repeating route with stops and those stops use the until attribute, the times will be shifted forward by ’cycleTime’ on each repeat

- etc.

A vehicle is defined using the vType-element. Some of the available vType attributes are:

- accel: The acceleration ability of vehicles of this type ($2.6 m/s^2$)
- decel: The deceleration ability of vehicles of this type ($4.5 m/s^2$)
- sigma: Car-following model parameter (0.5)
- length: The vehicle's netto-length (length) ($5.0 m$)
- maxSpeed: The vehicle's maximum velocity (55.55 ($200 km/h$) for vehicles, 1.39($5 km/h$) for pedestrians)
- etc.

There are different car-following models currently implemented in SUMO. The *"carFollowing-Krauss"* model is the default model. The default model says: Let vehicles drive as fast as possibly while maintaining perfect safety (always being able to avoid a collision if the leader starts braking within leader and follower maximum acceleration bounds).

Detectors

Sumo has the following detectors:

- Inductive loop detectors (E1): simulated induction loops
- Instant induction loops: simulated unaggregated induction loops

- Lane area detectors (E2): detectors that capture a lane segment (i.e. simulated vehicle tracking cameras)
- Multi-Entry-Exit detectors (E3): simulators that track traffic in an area by detecting entry and exit events at defined locations
- Route Detectors: detector for sampling route distributions

Lane Area Detector (E2)

For the SUMO simulation of this study, E2 detectors have been chosen. A laneAreaDetector is used to capture traffic on an area along a lane or lanes. In reality this would be similar to a vehicle tracking cameras. In contrast to an induction loop, a lane-area detector has a certain length which is specified by the length attribute or by the attributes pos, endPos. The outputs of an E2 Detector are tailored for measuring queues of standing/jammed vehicles and it keeps track of all vehicles which currently are on its area The following generated outputs from the E2 detectors were used for this study:

- nVehSeen: (num vehicles) The number of vehicles that were on the detector in the corresponding interval (were ”seen” by the detector).
- meanSpeed: (m/s) The mean velocity over all collected data samples.

SUMO Simulation

Running SUMO simulation for 5 hours and uploading the detector outputs every $T = 0.5min = 30sec$, the simulation time horizon was set to 18000 steps through TraCi. TraCI

is the short term for ”Traffic Control Interface”. TraCI uses a TCP based client/server architecture to provide access to sumo. Thereby, sumo acts as server that is started with additional command-line options. In other words, Traci is a programming language interface between SUMO and Python. The package ”tools/traci” allows to interact with sumo using Python. Having these settings, for each detector, 600 (=18000/30) rows of data were recorded. Max Speed of vehicles was chosen to be equal to 20.

Following figures are some snapshots from the network simulated in SUMO.

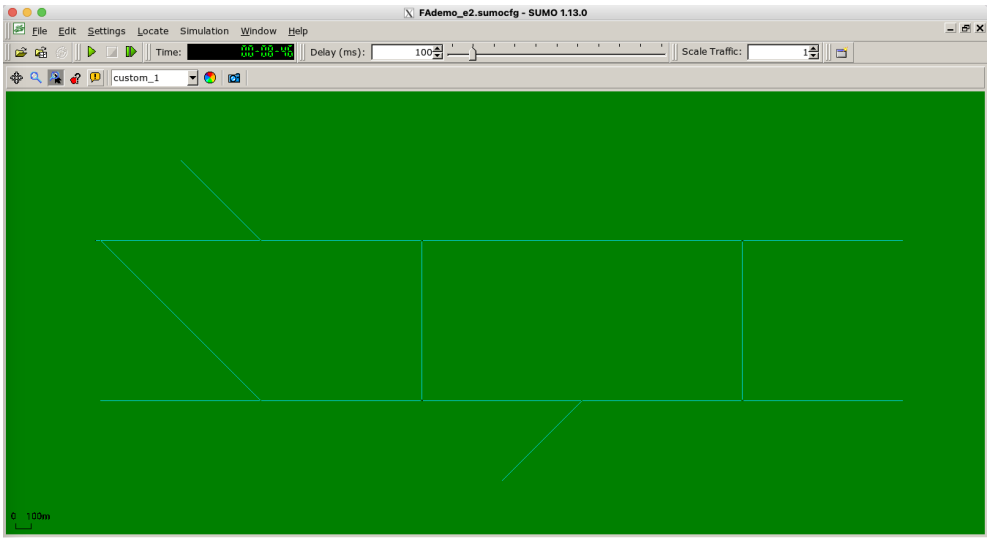


Figure 4.5: Simulated Network in SUMO

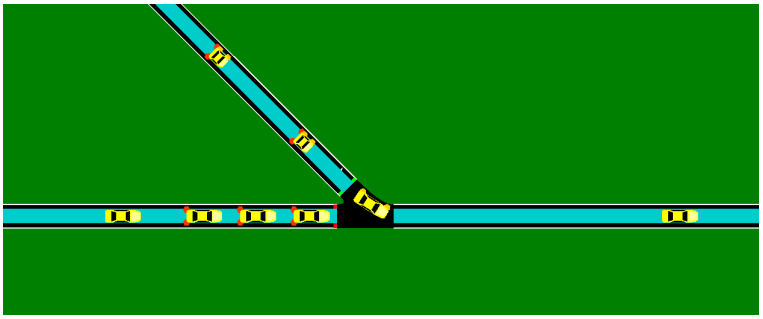


Figure 4.6: Movements of different flows of vehicle at the junction of on-ramp *o3* and link 2

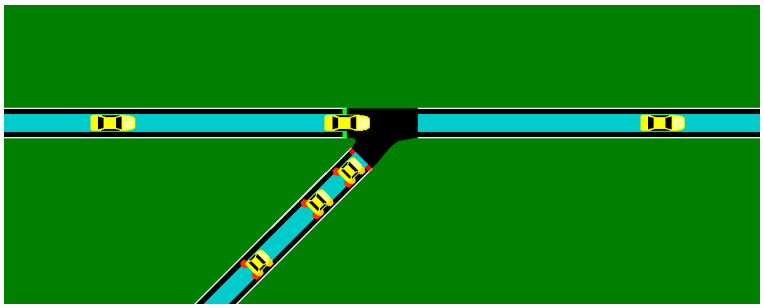


Figure 4.7: Movements of different flows of vehicle at the junction of on-ramp *o4* and link 8

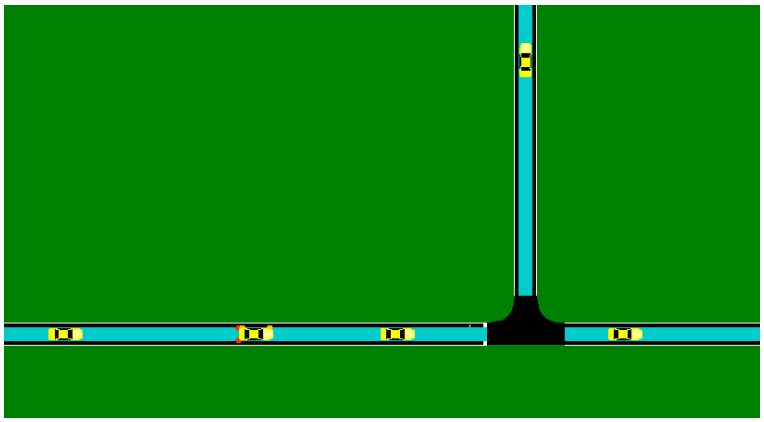


Figure 4.8: Flow of vehicles at a node with two outgoing links

4.5.2 Polynomial Curve Fitting

After the data generation process was finished, as the next step, the functions $p(v_i(k))$ and $q(\rho_i(k))$ were computed for all the links of the network. Then, to find the best polynomial curve fitting these functions, MATLAB *polyfit* function was used. Function *polyfit(x,y,n)* returns the coefficients for the polynomial $m(x)$ of degree n that is a best fit (in a least-squares sense) for the data in y. The coefficients in m are in descending powers, and the length of m is $n+1$.

$$m(x) = m_1 * x^n + m_2 * x^{n-1} + \ldots + m_n * x + m_{n+1}.$$

Then, the *polyval(p,x)* function was used which evaluates the polynomial p at each point in x. In other words, *Polyval* generates a curve to fit the data based on the coefficients found using polyfit.

Fit Polynomials for the Convection Term:

Figure 4.9 shows the result of fittig a curve to the function $p(v_i(k))$ for two links of the network. Blue points are the true functions values and red points are the polynomial curve fitted to the data.

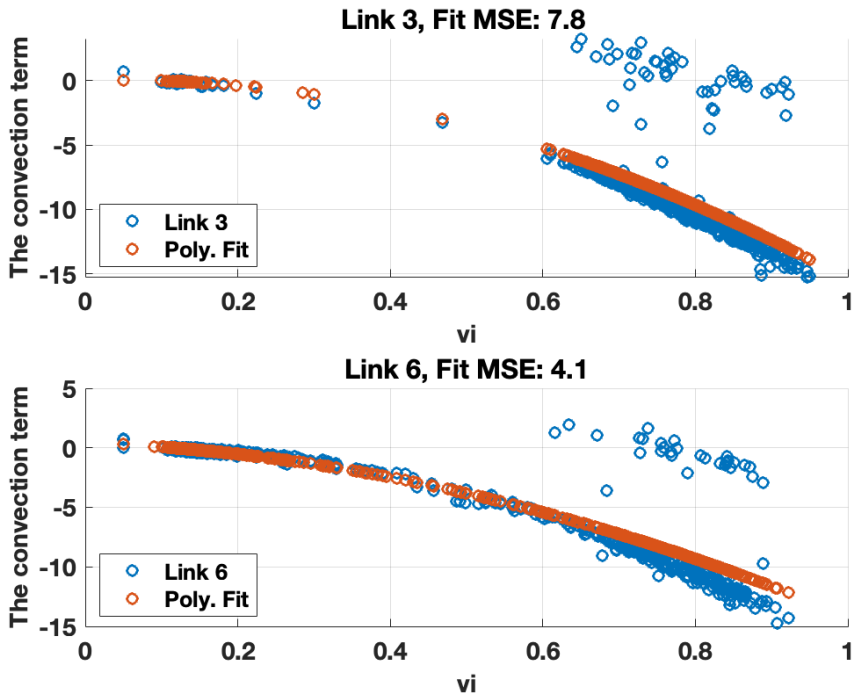


Figure 4.9: The convention term and its polynomial curve fitting of Link 3 (top) and link 6 (bottom)

The polynomial curves fitted to the $p(v_i(k))$ function of 6 links are shown in Fig. 4.10.

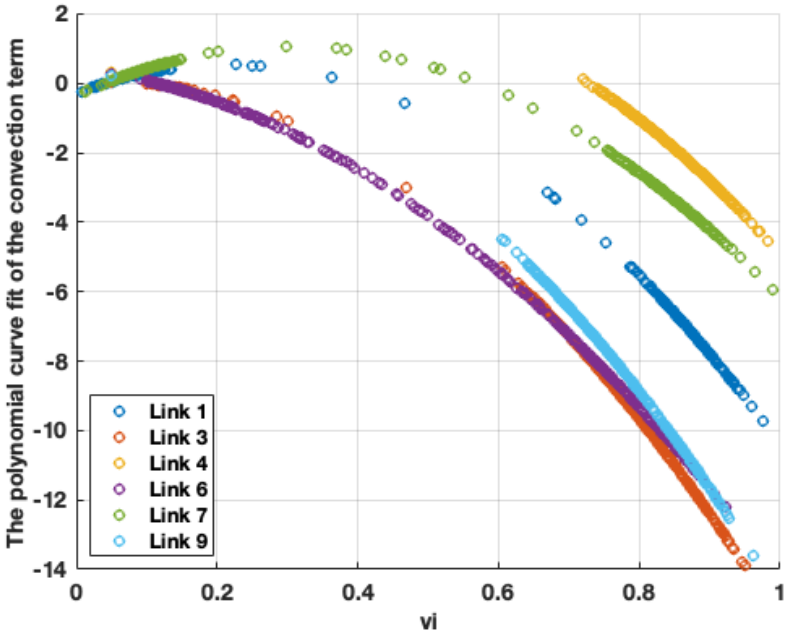


Figure 4.10: The polynomial curve fitting of 6 different links

Also, the MSE of fitting a curve to the true function of these 6 links are presented in Table 4.2.

Table 4.2: Comparison of the MSE of curve fitting for different links

Link	Curve Fitting MSE
1	0.2169
3	7.8208
4	8.2128
6	4.0640
7	5.0108
9	8.6864

At this point, the goal is to extract an approximate function which, with some reasonable error, would play the role of the $p(v_i(k))$ for all the links. We do not want to offer a model which is compatible with the behavior of only one link. As such, a curve should be chosen

that approximately can represent the behavior of all links to a good extent. Therefore, looking at Fig. 4.10 and considering the MSE errors of Table 4.2, the curve fitted to the $p(v_i(k))$ function of link 1 was chosen to move forward to the next step. The polynomial of the curve fitted for the $p(v_i(k))$ of link 1 is:

$$\hat{p}(v_i(k)) = -2.7277 * v_i(k)^2 - 2.5606 * v_i(k) - 0.0570 \tag{4.14}$$

As expected, the $\hat{p}(v_i(k))$ is a 2-nd order polynomial. Having this function, we would like to find a PWA approximation for it based on method explained in Chap. 3. Details of this process are skipped here and only the results are shown. Figure 4.11 shows the PWA approximation of the $p(v_i(k))$ in equ. 4.14. It has been approximated with 3 pieces and the function is provided in equ. 4.15. Also, the MSE of this approximation was equal to 0.0354 which is a very small error.

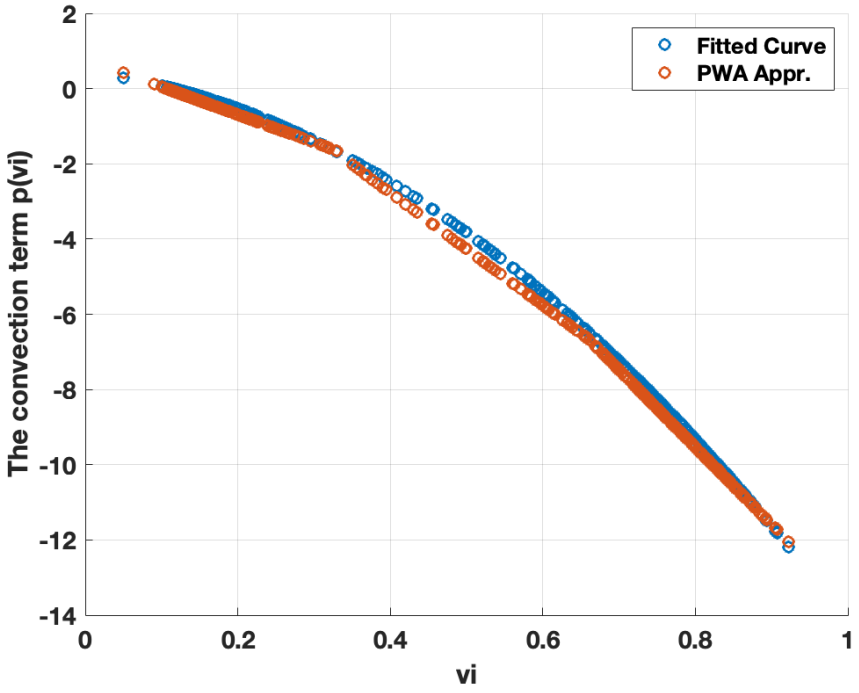


Figure 4.11: The PWA approximation of the convection term

$$\tilde{p}(v_i(k)) = \begin{cases} -7.39 * v_i(k) + 0.78, & 0 \leq v_i(k) \leq 0.32 \\ -14.94 * v_i(k) + 3.20, & 0.32 \leq v_i(k) \leq 0.65 \\ -20.53 * v_i(k) + 6.83, & 0.65 \leq v_i(k) \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{4.15}$$

Fit Polynomials for the Anticipation Term

The same process of the previous section is repeated for the $q(\rho_i(k))$ function. Figure 4.12 shows the true $q(\rho_i(k))$ function and the polynomial fitted to it for links 3 and 6 of the network.

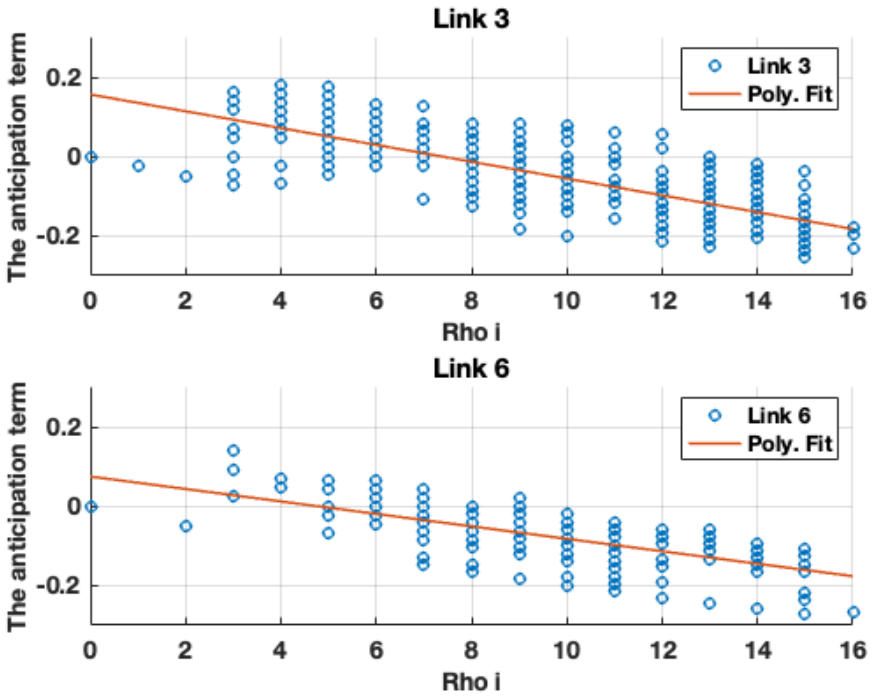


Figure 4.12: The anticipation term and the its polynomial curve fitting of Link 3 (top) and link 6 (bottom)

Figure 4.13 shows the polynomial curve fitted to the $q(\rho_i(k))$ function of 6 links and the MSE of this curve fitting are summarized in Table 4.3.

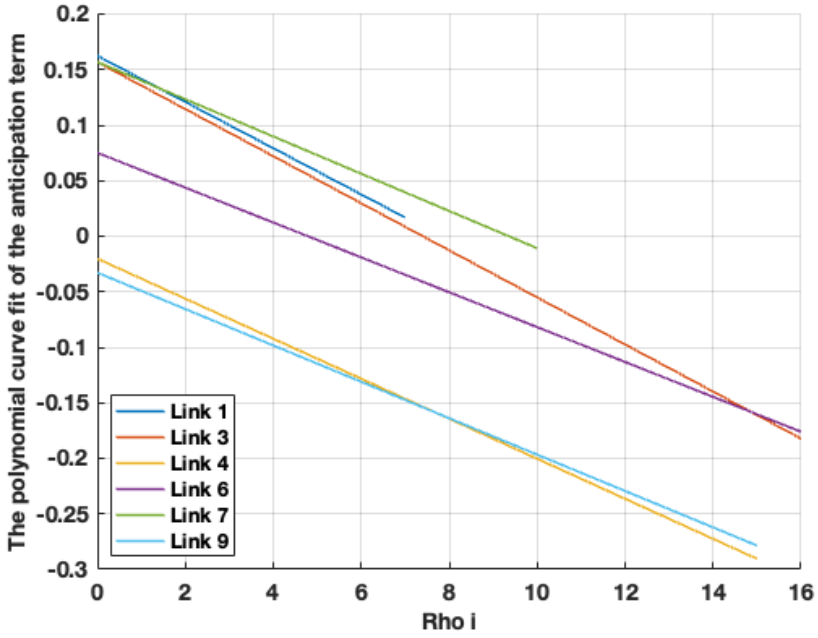


Figure 4.13: The polynomial curve fitting of 6 different links

Table 4.3: Comparison of the MSE of curve fitting for different links

Link	Curve Fitting MSE
1	7.79e-04
3	0.0027
4	1.71e-05
6	0.0013
7	0.0010
9	1.78e-05

At this point, the polynomial fitted for the $q(\rho_i(k))$ function of link 6 is chosen for the next step. This polynomial is provided in the following equation.

$$\hat{q}(\rho_i(k)) = -0.037 * \rho_i(k) - 0.06 \tag{4.16}$$

As expected, this polynomial is linear function, and it does not require any further PWA

approximations. As such, $\tilde{q}(\rho_i(k)) = \hat{q}(\rho_i(k))$.

4.6 The Piecewise Affine METANET Model (PWA-METANET)

The fully linearized METANET model will be called the PWA-METANET model and it constitute the following equations:

Density equation is similar with eq. 4.1 as it was a linear equation by itself.

As for the PWA velocity equation, using equations 4.11 for the PWA steady-state Velocity $\tilde{g}_3(\rho_i)$ in the relaxation term, eq. 4.15 for the PWA $\tilde{p}(v_i(k))$ function of the convection term, and eq. 4.16 for the PWA $\tilde{q}(\rho_i(k))$ function of the anticipation, we have:

$$v_i(k+1) = v_i(k) + \frac{T}{\tau} * [\tilde{g}_4(\rho_i) - v_i(k)] + \frac{T}{L_i} * \tilde{p}(v_i(k)) - \frac{\nu T}{\tau L_i} * \tilde{q}(\rho_i(k)) \tag{4.17}$$

Also, the traffic flow is given by its PWA approximation found in Chap. 3, eq. 3.21.

The queue length equation in eq. 4.5 will be used here as well as it was a linear equation too. As for the on-ramp control variable stated by equations 4.6 and 4.7, the *min* functions will be broken into 3 and 4 inequality (less than or equal) constraints, respectively, to also take care of the nonlinearity of the *min* function.

4.7 Finite Horizon Optimal Control Problem Formulation

For the performance evaluation of the PWA-METANET, four FHOCP problems are defined where in each of them, one of the nonlinear relations is replaced with its PWA approximation. All the FHOCP problems are defined by the cost function of eq. 3.24,

the RM controller of eq. 3.14, the density equation of eq. 4.1, the queue length equation in eq. 4.5, and the RM control variable in equations 4.6 and 4.7. The difference between these problems are in the steady-state velocity, mean velocity and traffic flow equation. The first is based on the original nonlinear METANET model. As we move on, we replace the three nonlinear equation with its PWA approximation one by one and define the rest of the problems. Definitions of the four problems formulated for this chapter are summarized in Table 4.4.

Table 4.4: Definitions of the FHOCPs and their problem type

	Steady-state Velocity $V(\rho_i(k))$	Flow Rate $\phi_i(k)$	Mean Velocity $v_i(k)$	Problem Type
FHOCP 1	Nonlinear	Nonlinear	Nonlinear	Nonlinear
FHOCP 2	PWA	Nonlinear	Nonlinear	Nonlinear
FHOCP 3	PWA	PWA	Nonlinear	Nonlinear
FHOCP 4	PWA	PWA	PWA	Linear

4.8 Simulation and Numerical Results

In this section, we aim to evaluate the performance and complexity of the proposed model on a complex simulation environment. The first step is to solve the optimization problems defined in Sec. 4.7.

4.8.1 Solvers and Algorithm for Nonlinear Programming Problems

A general optimization problem is to select n decision variables $x1, x2, ..., xn$ from a given feasible region in such a way as to optimize (minimize or maximize) a given objective function $f(x1, x2, ..., xn)$ of the decision variables. The problem is called a *nonlinear*

programming problem (NLP) if the objective function is nonlinear and/or the feasible region is determined by nonlinear constraints.

In this study, several of the FHOCPs are NLP due to having nonlinear constraints and only the problem based on the fully linearized approximation METANET was a linear programming problem.

General nonlinear programming solvers available are [60] BARON, FMINCON, IPOPT, KNITRO, and a few more. Amongst these solvers, the following solvers were tried:

- BARON: BARON full version was not free and its demo mode could only handle up to 10 constraints and variables and up to 50 nonlinear operations.
- IPOPT: It was not able to solve the large scale problem of this study.
- KNITRO: KNITRO full version was not free and its one-month full trial could solve a limited problem size (300 constraints and 300 variables).

As such, *Fmincon* solver (available with MATLAB academic license) was chosen as the solver for the nonlinear programming problems of this study.

Fmincon for Nonlinear Problems

Fmincon is the nonlinear programming solver function which finds the minimum of constrained nonlinear multi-variable functions. *Fmincon* has five algorithm options:

- 'interior-point' (default)
- 'trust-region-reflective'

- 'sqp'
- 'sqp-legacy'
- 'active-set'

The recommendations regarding the choice of algorithms are as follows:

Use the 'interior-point' algorithm first. "To run an optimization again to obtain more velocity on small- to medium-sized problems, try 'sqp' next, and 'active-set' last. Use 'trust-region-reflective' when applicable. Your problem must have: objective function includes gradient, only bounds, or only linear equality constraints (but not both)".

The reasoning behind these recommendations are:

'interior-point' handles large, sparse problems, as well as small dense problems. The algorithm satisfies bounds at all iterations, and can recover from NaN or Inf results. It is a large-scale algorithm; see Large-Scale vs. Medium-Scale Algorithms. The algorithm can use special techniques for large-scale problems. For details, see Interior-Point Algorithm in *fmincon* options. 'sqp' satisfies bounds at all iterations. The algorithm can recover from NaN or Inf results. It is not a large-scale algorithm; see Large-Scale vs. Medium-Scale Algorithms. 'sqp-legacy' is similar to 'sqp', but usually is slower and uses more memory. 'active-set' can take large steps, which adds velocity. The algorithm is effective on some problems with nonsmooth constraints. It is not a large-scale algorithm; see Large-Scale vs. Medium-Scale Algorithms. 'trust-region-reflective' requires you to provide a gradient, and allows only bounds or linear equality constraints, but not both. Within these limitations, the algorithm handles both large sparse problems and small dense problems efficiently. It

is a large-scale algorithm; see Large-Scale vs. Medium-Scale Algorithms. The algorithm can use special techniques to save memory usage, such as a Hessian multiply function.

Interior-Point Algorithm

Interior-point methods (also referred to as barrier methods or IPMs) are a certain class of algorithms that solve linear and nonlinear convex optimization problems. An interior point method was discovered by Soviet mathematician I. I. Dikin in 1967 and reinvented in the U.S. in the mid-1980s. In 1984, Narendra Karmarkar developed a method for *linear programming* called Karmarkar's algorithm, which runs in provably *polynomial time* and is also very efficient in practice. It enabled solutions of linear programming problems that were beyond the capabilities of the simplex method. Contrary to the simplex method, it reaches a best solution by traversing the interior of the feasible region. The method can be generalized to convex programming based on a self-concordant barrier function used to encode the convex set. Any convex optimization problem can be transformed into minimizing (or maximizing) a linear function over a convex set by converting to the epigraph form. The idea of encoding the feasible set using a barrier and designing barrier methods was studied by Anthony V. Fiacco, Garth P. McCormick, and others in the early 1960s. These ideas were mainly developed for general nonlinear programming, but they were later abandoned due to the presence of more competitive methods for this class of problems (e.g. sequential quadratic programming). Yurii Nesterov, and Arkadi Nemirovski came up with a special class of such barriers that can be used to encode any convex set. They guarantee that the number of iterations of the algorithm is bounded by a polynomial in the dimension and

accuracy of the solution. Karmarkar's breakthrough revitalized the study of interior-point methods and barrier problems, showing that it was possible to create an algorithm for linear programming characterized by polynomial complexity and, moreover, that was competitive with the simplex method. Already Khachiyan's ellipsoid method was a polynomial-time algorithm; however, it was too slow to be of practical interest. The class of primal-dual path-following interior-point methods is considered the most successful. Mehrotra's predictor–corrector algorithm provides the basis for most implementations of this class of methods [61].

The interior-point approach to constrained minimization is to solve a sequence of approximate minimization problems. The original problem is

$$\min_x f(x), \quad \text{subject to} \quad h(x) = 0 \quad \text{and} \quad g(x) \leq 0 \tag{4.18}$$

For each $\mu > 0$, the approximate problem is

$$\min_{x,s} f_\mu(x,s) = \min_{x,s} f(x) - \mu \sum_i \ln(s_i), \quad \text{subject to} \quad s \geq 0, \quad h(x) = 0, \quad \text{and} \quad g(x) + s = 0 \tag{4.19}$$

There are as many slack variables s_i as there are inequality constraints g. The s_i are restricted to be positive to keep the iterates in the interior of the feasible region. As μ decreases to zero, the minimum of f_μ should approach the minimum of f. The added logarithmic term is called a barrier function. The interior-point method is fully described in [62] and [63]. The approximate problem eq. 4.19 is a sequence of equality constrained problems. These are easier to solve than the original inequality-constrained problem eq. 4.18.

Subproblem Algorithm: Conjugate Direction Methods

When the *EnableFeasibilityMode* option is true and the iterations do not decrease the infeasibility quickly enough, the algorithm switches to feasibility mode. This switch happens after the algorithm fails to decrease the infeasibility in normal mode, and then fails again after switching to conjugate gradient mode. Therefore, for best performance when the solver fails to find a feasible solution without feasibility mode, set the SubproblemAlgorithm to Conjugate Gradient ('cg') when using feasibility mode. Doing so avoids fruitless searching in normal mode.

The feasibility mode algorithm is based on [64]. The algorithm ignores the objective function and instead tries to minimize the infeasibility, defined as the sum of the positive parts of the inequality constraint functions and the absolute value of the equality constraint functions.

Conjugate direction methods can be regarded as being between the method of steepest descent (first-order method that uses gradient) and Newton's method (second-order method that uses Hessian as well). The motivation behind developing such a method is that the steepest descent is slow. The goal of conjugate direction methods is to accelerate it.

Newton method is fast. But, we need to calculate the inverse of the Hessian matrix. Conjugate direction methods is something between steepest descent and Newton method. It's goal is to:

- Accelerate the convergence rate of steepest descent,
- To avoide the high computational cost of Newton's method.

As such, to solve the nonlinear programming problems of this study, *fmincon* is chosen as the solver, the interior-point approach is used as the main algorithm and conjugate direction method (or conjugate gradient ('cg')) is chosen as the algorithm to solve the sub-problems.

4.8.2 Solver and Algorithm for Linear Programming Problems

For linear programming, the following famous sovlers are available [60]:

CPLEX (free for academia), GUROBI (free for academia), LINPROG, MOSEK (free for academia), XPRESS (generous community trial license available). In [65], it is stated that the fastest and most powerful mathematical programming solver available for LP, QP and MIP (MILP, MIQP, and MIQCP) problems is *Gurobi*. Benchmarks consistently show that Gurobi finds both feasible and proven optimal solutions faster than competing solvers (including CPLEX and XPRESS). The Gurobi Optimizer is simply the best mathematical optimization solver out there [65]. As such, it was chosen as the main solver for this section.

The key Features of Gurobi are:

- Problem-Solving Power: The results of benchmark tests consistently show that the Gurobi Optimizer finds both feasible and proven optimal solutions faster than competing solvers, with the performance gap growing as model size and difficulty increase. The Gurobi Optimizer is capable of solving all major problem types (convex and non-convex): Linear programming (LP), Mixed-integer linear programming (MILP), Quadratic programming (QP), Mixed-integer quadratic programming (MIQP), Quadratically-constrained programming (QCP), Mixed-integer quadratically-

constrained programming (MIQCP).

- Ease of Use: The Gurobi Optimizer is an out-of-the-box solver that – without any customization of code – can be used in a vast array of applications to tackle a whole host of real-life business problems.

- State-of-the-Art Algorithms: The Gurobi Optimizer provides advanced implementations of the latest algorithms including:

  - LP algorithms – simplex, parallel barrier with crossover, concurrent and sifting.

  - QP algorithms – simplex and parallel barrier QCP algorithms – parallel barrier (SOCP).

  - MIP algorithms – deterministic parallel, non- traditional search, heuristics, solution improvement, cutting planes, and symmetry breaking.

- User-Friendly Development Environment

- Flexible Deployment

- Responsive Support

Gurobi developers come up with certain heuristics and rounding techniques, and then implement it using concepts of distributed computing. So all you have to do is to get a set of equations and throw it to the solver.

As such, to solve the linear programming problems of this study, Gurobi was the preferred solver. Still, to have a comparison between the performance of this solver and others,

MOSEK solver was also tried.

4.8.3 Numerical Results

In this section, the numerical results of solving the four FHOCPs of Sec. 4.7 are presented. The same network and model parameters of Chap. 3 was adapted here as well. Also, the simulation horizon was chosen equal to 1 hour (K = 120 times steps, T = 0.5 min) again. Table 4.5 shows a comparison between the iteration number and computation time of solving each problem with its suitable solver.

Table 4.5: Comparison of the Computation Times (sec) and Iteration Numbers

	Iteration Number	Computation Time	Solver
FHOCP 1	4570	6.80e+04 (18.91 hrs)	fmincon
FHOCP 2	1561	8.3788e+03 (2.32 hr)	fmincon
FHOCP 3	357	2.02e+03 (0.56 hr)	fmincon
FHOCP 4	94	3.54e+03 (0.98 hr)	fmincon
FHOCP 4	22	1.74	Gurobi

Figure 4.14 shows the comparison of the network flow rate for all problems. All four problems converge to the network equilibrium flow rate equal to 113.37 *veh*/0.5 *min*. However, FHOCP 4 has many few fluctuations and faster convergence.

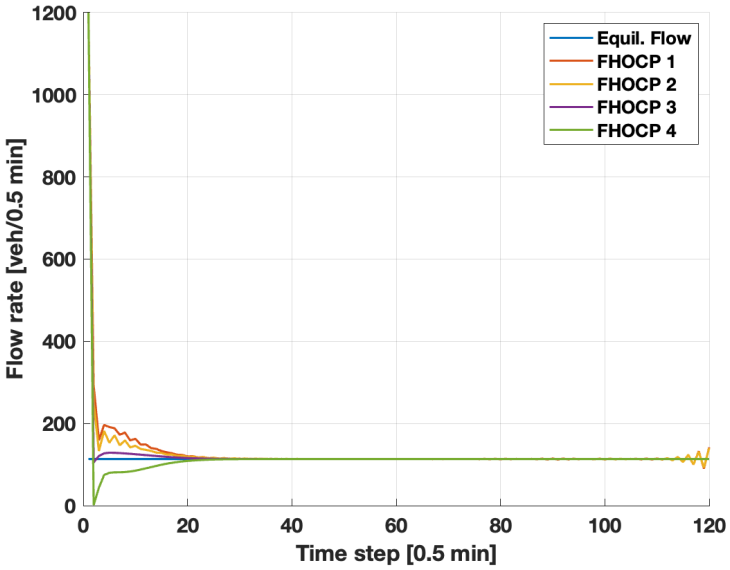


Figure 4.14: Comparison of the network flow rate in all four problems.

Figure 4.15 shows the network mean velocity evolution. As expected the problem based on PWA-METANET (FHOCP 4) is showing the largest deviation compared with FHOCP 2 and 3 with respect to the original problem (FHOCP 1) and the equilibrium value of the network mean velocity. It is because of the substitution of three terms of the mean velocity with their approximations.

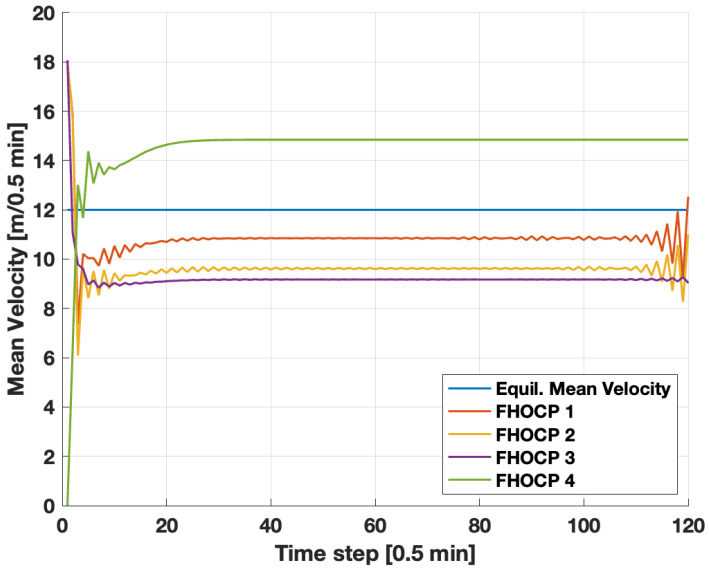


Figure 4.15: Evolution of the mean velocity in all four problems.

Also, Fig. 4.16 represents the evolution of RM control variable of on-ramp $o3$. As discussed in Sec. 3.6, a proper RM variable basically should replicate the overall demand pattern of that on-ramp. In all four problems, the control variable is trying to converge to the value of the external demand applied (a constant demand of 7 *veh*/0.5 *min*). However, the RM variable in problems based on PWA-METANET have smoother and faster convergence with much less fluctuations while the control variable in the other problems represents large fluctuations before they starts to converge.

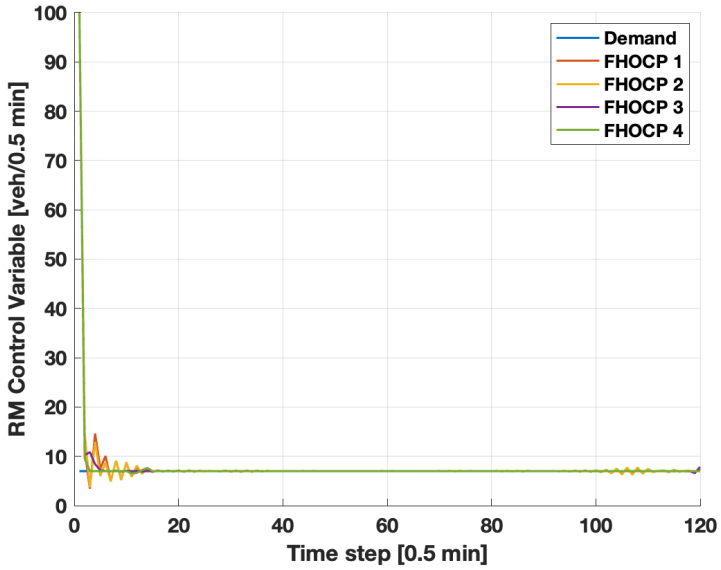


Figure 4.16: Evolution of the RM control variable of on-ramp $o3$.

In addition, a comparison between the approximation error of steady state velocity, flow rate and mean velocity of all these four problems was performed and the results are summarized in Table 4.6. The approximation error was computed as

$$\text{Error} = \left| \frac{x - x_{\text{approx}}}{x} \right| * 100 \tag{4.20}$$

Table 4.6: Error Comparison of different factors w.r.t the original model (*SS Vel. means steady state velocity).

	SS Vel.* (%)	Flow (%)	Mean Vel. (%)	TTS (%)	TTD (%)	Cost (%)
FHOCP 1	0	0	0	0	0	0
FHOCP 2	10.23	1.77	10.94	5.99	1.62	4.99
FHOCP 3	11.42	4.83	14.60	23.85	4.38	16.90
FHOCP 4	11.43	9.46	33.96	17.36	8.03	19.29

A glance to the table of errors show that overall, considering all the factors investigated,

the FHOCP 4 which is based on PWA-METANET represents larger percentage of error due to having more relations approximated with PWA functions. Still, if we consider the trade-off between the computation time and error percentage of FHOCP 4 and FHOCP 1, this PWA approximation could be a great substitute for the original nonlinear model for any traffic control optimization study. As mentioned in Sec. 1.2.1, the speed of simulation is a great factor for researchers in the field of traffic management and brings significant benefits like real-time decision making capabilities, disruption management and robust scenario analysis which are all critical features for a control scheme.

4.9 Conclusion

In this chapter, a linear approximation of the METANET model, called the PWA-METANET, was developed which was based on the PWA approximation of the steady-state velocity, the flow rate, and the mean velocity equations. The METANET model is a non convex nonlinear model and when utilized as the underlying model of an optimization problem, makes the problem nonlinear and solving it will be either difficult and time consuming and for larger networks, it could even become impossible.

To PWA approximation of the METANET model proposed here is computed based on two parts: 1) polynomial curve fitting of the nonlinear terms using the synthetic data generated by SUMO traffic simulator, 2) the two-step PWA identification method mentioned in the previous chapter. The linearized METANET model is evaluated in the framework of FHOCP for a complex hypothetical highway network.

To better investigate the effect of approximating each of the 3 nonlinear equations of the

METANET, the PWA approximate of these equations was substituted in the original nonlinear model one by one and overall, four FHOCPs were proposed. Simulation results of these FHOCPs show that the problem based on PWA-METANET model requires much less computation time and iteration numbers compared to other problems. Also, the optimization problems based on the linearized METANET remains tractable for real-time application of traffic-management optimal controllers. Hence, the proposed PWA-METANET constitute one of the best linear approximation approaches for second-order traffic flow models suitable for more challenging modeling and control applications.

Chapter 5: A Learning-Based Ramp Metering Algorithm Based on Policy Gradient Methods

5.1 Introduction

In this chapter, we are going to investigate reinforcement learning algorithms to solve the optimal control problem of ramp-metering. Reinforcement learning methods are direct adaptive algorithms that approximate the solution to infinite-horizon optimal control problems [21]. In our case, they offer a potentially appealing alternative method to solve the problem at hand, since they are data-based and make no assumptions on the underlying model parameters.

Towards this direction, it is convenient to study the traffic flow model as a multi-agent system of non-homogeneous networked agents that belong to two classes: the controlled links, i.e., ramps with a ramp metering controller, and the uncontrolled links, i.e., links that are not controlled and the density flows freely depending on the intrinsic dynamics. In the following, we offer a novel formulation of the ramp metering problem as an optimal control problem on a first-order multi-agent dynamical system. We then apply policy gradient reinforcement learning algorithms to find a probabilistic policy that solves the ramp-metering problem.

5.1.1 Related Work on RM

Ramp metering (RM) is one of the most important strategies for traffic control to reduce freeway delay and improve safety [9, 15, 25, 43, 66–68]. It is achieved by placing traffic signals at on-ramps to control the flow rate at which vehicles enter the freeway. The ramp metering controller computes the metering rate to be applied. Ramp metering has the following different goals [9, 69]:

- to improve or remove congestion;
- to alleviate freeway flow, traffic safety and air quality;
- to reduce total travel time and the number of peak-period accidents;
- to regulate the input demand of the freeway system so that a truly operationally balanced corridor system is achieved.

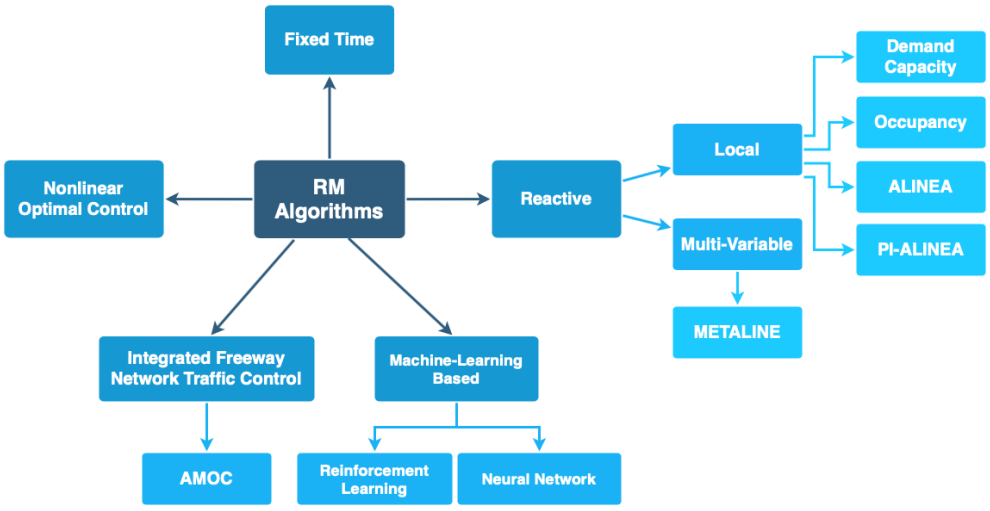


Figure 5.1: Ramp metering algorithms classification [6]

A classification of ramp metering algorithms is presented in the Figure 5.1.

Fixed time metering is the simplest strategy which is usually adjusted based on historical data and applied during particular times of day. Obviously, fixed time metering has very limited application due to its major shortcoming of not historical traffic data. The computational simplicity of the proposed method allows it to run on reacting to any real-time traffic metrics.

Reactive ramp metering techniques are based on real time traffic metrics. Two main types in this category are the local and multi-variable strategies.

Local ramp metering uses traffic measures collected form the ramp vicinity. Demand-capacity, and occupancy based strategies allow as much traffic inflow as possible to reach the freeway capacity. ALINEA and PI-AlINEA offer a more complex and more responsive strategy that unlike capacity and occupancy strategies generates smoother responses towards changes in metrics.

Multi-variable regulator strategies perform the same as local strategies, but more comprehensively and independently on a set of ramps and usually outperform local strategies. METALINE can be viewed as a more general and extended form of ALINEA.

Nonlinear optimal control strategy considers local traffic parameters and metrics as well as nonlinear traffic flow dynamics, incidents, and demand predictions in a freeway network and outputs a consistent control strategy.

Integrated freeway network traffic control is a more general approach to nonlinear control that extends application of optimal control strategies to all forms of freeway traffic

control.

Most of these methods lie under the category of online optimization methods which calculate the metering rate by solving on-line optimal control problems. These approaches require accurate models to predict the traffic dynamics and contain large online computing workload which is considered not capable for large-scale applications. Also, such approaches require deep understanding about the traffic flow characteristics and congestion mechanisms at the target bottlenecks. The obtained strategies may be site-specific and may not work well when applied on other freeway bottlenecks.

The optimization-based approaches optimize signal timings based on real-time traffic data. The model predictive control framework, which considers the interactions between ramp metering and future traffic states, is often employed to predict traffic evolution for proactive traffic controls. Nevertheless, the models used to describe traffic dynamics may lead to nonlinear control problems, which brings the difficulty in finding the optimal signal timings. In addition, the effectiveness of the ramp metering strategy depends on the degree of the fitness of the models to the actual traffic dynamics [70].

Artificial intelligence and *machine learning approaches* like *reinforcement learning* (RL) and *artificial neural networks* (ANN) are new techniques being implemented recently for RM control [71]. In the next section, more details on this category of RM algorithms will be explained.

5.1.2 Related Work on the Application of RL in RM

This section describes RL-adopted control schemes where the ramp metering is the control action, and the goal is mitigating congestion phenomena in a highway network.

The rule-based approaches update signal timings in real-time according to specific rules. The study in [72] proposed a feedforward ramp metering algorithm, aiming at keeping the downstream traffic flow below the capacity. Papageorgiou et al. in [15] proposed a feedback ramp metering algorithm, which is named as ALINEA, based on the occupancy obtained by inductive loop detectors downstream of the merging area. The ALINEA regulates the inflows from the on-ramp in the scheme of closed-loop control so that the detected occupancy would approach a predefined target value. The ALINEA was further extended by considering more traffic measures, such as queue lengths at the on-ramp and traffic volumes in the mainline [17]. Wang et al. in [40] proposed the proportional-integral-based ALINEA (i.e. PI-ALINEA), considering the cases where a bottleneck with a capacity lower than that of the merging area is present further downstream of the merging area. They demonstrated numerically that the PI- ALINEA performs better than the original ALINEA. The ALINEA has been incorporated in several coordinated ramp metering algorithms, such as METALINE and HERO [73].

The RL based approaches enables the agent to obtain optimal strategy in an unfamiliar environment without prior knowledge on traffic flow, reducing the need for human knowledge. Previous researchers have incorporated the Q-learning (QL) which is a basic RL algorithm, into the RM control tasks. Deep Q-learning (DQN) is a combined use of the QL

and the deep neural network (DNN). [70] incorporated the DQN algorithm with the RM control to reduce total travel time and to improve traffic operation at freeway bottleneck areas.

Existing RL-based ramp metering approaches were mainly developed based on value-based RL methods, such as the Q-learning algorithm. They evaluate the value of policy given each state-action pair and improve the policy with a heuristic search. The actions usually refer to the metering rate or signal phase selection. The states are represented by traffic measures, such as traffic density and volume. The downstream and upstream traffic flow, ramp queue length, and traffic density are often selected to define the control reward [74–77]. The RL has also been introduced for coordinated ramp metering [78, 79].

In this chapter, the optimal control problem of ramp-metering is formulated based on policy gradient reinforcement learning (RL) algorithms. Figure 5.2 shows the general structure of using RL for the purpose of traffic control. Details of this methodology is explained in the next chapters.

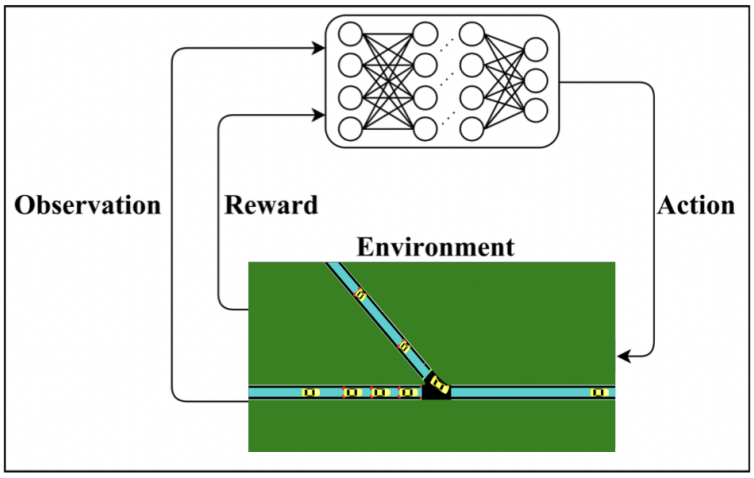


Figure 5.2: Traffic control with RL framework [7]

5.2 The CTM model as a Multi-Agent System

In this section, we introduce a multi-agent formulation of the CTM model. First, we start with the definition of the agent. An agent i is a section of the road (also referred to as a mainline link or an on-ramp), that is described by an one-dimensional system with:

- the density $\rho_i \in \mathbb{R}_+$ being the state of the system;
- the incoming flow $\phi_i \in \mathbb{R}_+$ being the input of the system; and
- the out-going flow $r_i \in \mathbb{R}_+$ being the output of the system.

The incoming flow ϕ_i of each agent is the sum of the out-going flows r_j, $j \in \mathcal{N}_i$, of the agents in its neighborhood N_i, in addition to a constant external demand d_i that may or may not exist. The neighborhood $\mathcal{N}_i$ of each agent i is defined with respect to the topology of the network, which, as will be shown later, corresponds to a directed graph. This is illustrated in Fig. 5.3.

Assuming N agents, the road model is formulated as a weighted directed graph $G = (V,E)$ where the nodes V is the set of all the agents in the system ($|V| = N$), and the edge set $E \subseteq V \times V$ is a set of ordered pairs of vertices that is defined by the specifications of the problem. Each edge $(i,j) \in E$ is associated with a weight w_{ij} that represents the percentage of the out-going flow r_j of agent j that flows towards agent i. This is also illustrated in Fig. 5.3. When $w_{ij} = 0$, the two agents are assumed disconnected. The neighborhood $\mathcal{N}_i$ is defined as $\mathcal{N}_i = \{j : w_{ij} \neq 0\}$. The weights w_{ij} form the non-symmetric weighted adjacency matrix W that encodes all the information needed about the structure of the network, and

thus, the road configuration.

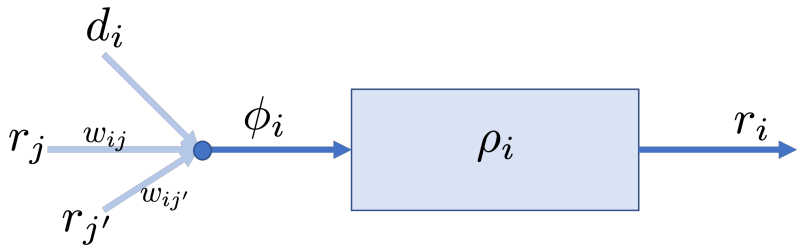


Figure 5.3: A CTM agent.

Given the definitions above, the dynamics of each agent i are given by:

$$\Delta\rho_i = \frac{T}{L}(\phi_i - r_i) \tag{5.1}$$

where

$$\phi_i = d_i + \sum_j w_{ij} r_j \tag{5.2}$$

and

$$r_i = \begin{cases} f_{FD}(\rho_i), \text{ mainline} \\ f_C(\rho_i), \text{ on-ramp} \end{cases} \tag{5.3}$$

Here $\Delta\rho_i$ is used to describe $\rho_i(k+1) - \rho_i(k)$ such that the time indices can be removed for clarity of the notation. The constraints

$$\begin{cases} \rho_i \leq \rho_i^{max}, \\ \phi_i \leq \phi_i^{max}, \\ r_i \leq \phi_i + \frac{L}{T}\rho_i \end{cases} \tag{5.4}$$

should also hold.

The functions f_{FD} and f_C define the controllers of each agent, depending on whether the agent is an uncontrolled link in the mainline or a controlled link in an on-ramp. According to the original CTM model, if an agent is a link in the mainline, the function f_{FD} is given by the fundamental diagram described in Chap. 3, eq. 3.20. However, if the agent is an on-ramp controlled by ramp-metering, the function f_C defines the ramp-metering controller. As a result, the function f_C is the solution of our optimal control problem.

In what follows, we will approximate the function f_C with a parametric model $f_\theta \simeq f_C$, where the vector $\theta \in \mathbb{R}^m$ will be the unknown parameter vector. As a result, in vector notation, we can represent $r := \{r_i\}$ as a function of the densities $\rho := \{\rho_i\}$ and the parameters θ, i.e., $r(\rho, \theta)$. Finally, in vector notation, we can represent the entire multi-agent system dynamics as:

$$\Delta\rho = \frac{T}{L}(W - \mathbb{I})r(\rho, \theta) + \frac{T}{L}d \tag{5.5}$$

To compare this multi-agent formulation with the classical road model, we present in Fig. 2.3 the same network used in the previous chapters, using the formulation described here.

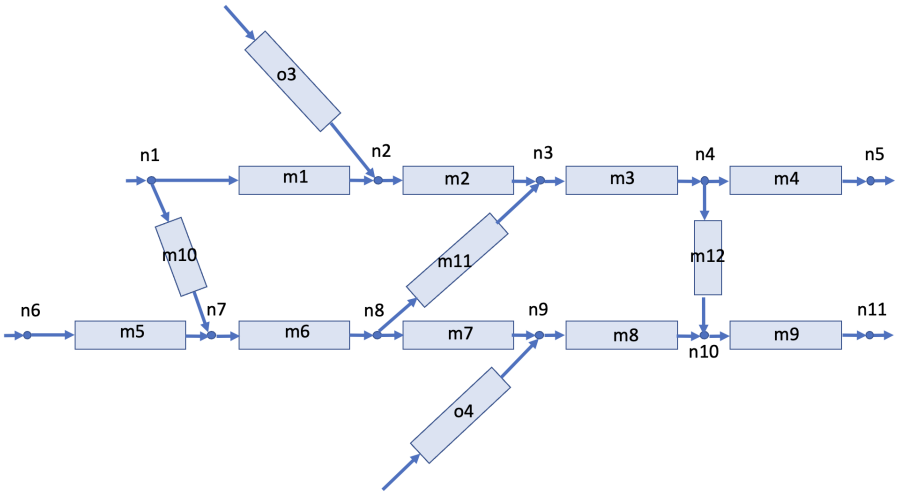


Figure 5.4: The network of previous chapter in multi-agent formulation.

5.3 Reinforcement Learning and Policy-Gradient (PG) Methods

Reinforcement Learning studies a sequential decision making problem in which an agent acts in a stochastic environment with unknown dynamics by sequentially choosing actions, so as to maximize a desired performance measure. A PG method tries to optimise a parameterised policy function by gradient-descent/ascent method. Indeed, PG methods are interested in searching policy space to learn policies directly, instead of estimating state-value or action-value functions. Unlike the traditional RL algorithms, PG methods do not suffer from the convergence problems of estimating value functions under non-linear function approximation or in the environments which might be partially observable Markov decision process (POMDP). They can also deal with the complexity of continuous state and action spaces better than purely value-based methods. PG methods estimate policy gradients using Monte Carlo estimates of the policy gradients. These methods are guaranteed to converge to a local optimum of their parameterised policy function. However, typically PG

methods result in high variance in their gradient estimates [7]. Hence, in order to reduce the variance of the gradient estimators, some methods subtract a baseline function from the policy gradients. By inspiring these features of PG methods and successes of neural networks in automatic feature abstractions, we use DNNs to represent an optimal traffic control policy directly in the RM control problem.

Reinforcement Learning (RL) considers the problem of computing a sequence of decisions under uncertainly in unknown or complex environments, in which an agent executes actions in a sequential manner, so as to maximize some desired performance measure [21, 80]. Recent impressive performance of RL algorithms in the domain of computer games [81], starting with the DQN [82] algorithm, as well as successful applications of RL systems in multiple domains, such as robotics, traffic control, interactive education, and health care, have brought RL to the forefront of current research.

Reinforcement Learning studies a sequential decision making problem in which an agent acts in a stochastic environment with unknown dynamics by sequentially choosing actions, so as to maximize a desired performance measure. In Reinforcement Learning, the interactive environment is modeled as a Markov Decision Process (MDP). An MDP is a tuple $\mathscr{M}=(\mathscr{S},\mathscr{A},p_0,P,r,\gamma)$, where $\mathscr{S}$ is the state space and $\mathscr{A}$ is the action space, which may each be discrete or continuous; p_0 is the initial state distribution; $P:\mathscr{S}\times\mathscr{A}\to\Delta(\mathscr{S})$ is the transition kernel , which is in general unknown, where $\Delta(\mathscr{S})$ denotes the space of probability distributions on $\mathscr{S}$; $r:\mathscr{S}\times\mathscr{A}\to\mathbb{R}$ is the reward function, defining a task; and $\gamma\in[0,1)$ is a discounting factor.

In such environment, the behavior of an RL-agent is characterized by its policy. A (randomized) policy $\pi(\cdot|s)$ is a probability distribution over action space given the state, which prescribes the probability of taking an action $a \in \mathscr{A}$ when in state $s \in \mathscr{S}$.

At each time-step t, when in state s_t, the RL-agent executes an action a_t according to a differentiable parameterized policy (possibly a Neural Network) $\pi(\cdot|s_t;\theta)$ where $\theta \in \mathbb{R}^d$ is a vector of d parameters. Upon the execution of the action, the system transitions to a successor state s_{t+1} according to transition probability $p(s_{t+1}|s_t,a_t)$, unknown to the agent, and the agent receives a reward $r_t := r(s_t,a_t)$. In the setting of finite horizon episodic RL, the agent aims to find a policy that maximizes some desired performance measure, $J(\theta)$, during an episode.

A trajectory τ is a sequence of states and actions. The agents' policy and the system transition probabilities induce a trajectory distribution, a probability distribution over the sequence of states and actions, i.e. the space of all possible trajectories $\mathscr{T}$. The probability distribution induced over the space of trajectories by following the policy parameterized by θ is denoted by $\rho_\theta(\tau)$ and is given by

$$\rho_\theta(\tau) = p_0 \prod_{t=0}^{|\tau|-1} \pi(a_t|s_t;\theta)p(s_{t+1}|s_t,a_t) \tag{5.6}$$

Policy gradient Reinforcement Learning (RL) algorithms are among the most well-studied [83–85]. The desired performance measure is a long-run expectation, such as the undiscounted cumulative reward, and is optimized with respect to parameterized policies using stochastic gradient descent/ascent. The standard REINFORCE algorithm [86] is a

prime example of a policy gradient algorithm that seeks a stochastic policy $\pi(a|s;\theta)$, a distribution over action space $\mathscr{A}$ given the system state $s \in \mathscr{S}$, so as to maximize the expected cumulative (discounted) reward, i.e.

$$\max_{\theta} J(\theta) := \mathbb{E}_{\tau\sim\rho_\theta}[R(\tau)] \tag{5.7}$$

where $R(\tau) = \sum_{t=0}^{|\tau|-1} \gamma^t r(s_t, a_t)$ is the trajectory's total reward; expectation is taken over space of trajectories $\mathscr{T}$ generated by following the policy, i.e. $s_0 \sim p_0$, $a_t \sim \pi(\cdot|s_t;\theta)$ and $s_{t+1} \sim p(\cdot|s_t, a_t)$. Thus, the optimal policy parameter can be obtained iteratively by using gradient ascent as follows

$$\theta_{t+1} = \theta_t + \alpha\widehat{\nabla J(\theta_t)} \tag{5.8}$$

where $\alpha\in\mathbb{R}$ is a constant step-size, i.e., learning rate, and $\widehat{\nabla J(\theta_t)}\in\mathbb{R}^d$ denotes an unbiased stochastic estimate of the gradient of the system's performance measure with respect to the policy parameter θ.

The gradient of the (risk-neutral) objective of (5.7) with respect to the policy parameters is given by Policy Gradient theorem [83]

$$\nabla J(\theta) \propto \mathbb{E}_{\tau\sim\rho_\theta}\Big[R(\tau)\sum_{t=0}^{|\tau|-1}\nabla\log\pi_\theta(a_t|s_t;\theta)\Big] \tag{5.9}$$

The Monte Carlo estimation of the expectation in Eq. (5.9) suffers from high variance. To reduce variance, by taking advantage of the temporal structure of the problem and causality, it can be shown that the gradient could be re-written in terms of reward-to-go

$R_t := \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r(s_{t'}, a_{t'})$ as follows:

$$\nabla J(\theta) \propto \mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1} R_t \nabla \log \pi_\theta(a_t|s_t;\theta)\Big] \tag{5.10}$$

Using Eq. (5.10), the update rule in standard REINFORCE algorithm is obtained, i.e.,

$$\theta_{k+1} = \theta_k + \alpha R_t \frac{\nabla \pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta)} \tag{5.11}$$

Note that the Monte Carlo estimate of expectation in Eq. (5.10) has a lower variance than that of the expectation in Eq. (5.9). However, Monte Carlo estimates of both expectations in Equations (5.9) and (5.10) typically have high variance and hinder learning, especially in complex environments. A well-studied variance reduction technique to reduce the variance associated with the gradient estimations of (5.9) and (5.10) is the subtraction of an appropriately chosen baseline, reducing the variance without introducing bias. To that end, the policy gradient theorem can be generalized to include a baseline as follows [80]:

$$\nabla J(\theta) \propto \mathbb{E}_{\tau\sim\rho_\theta}\Big[\sum_{t=0}^{|\tau|-1} \Big(R_t - b(s_t)\Big) \nabla \log \pi_\theta(a_t|s_t;\theta)\Big] \tag{5.12}$$

where $b(s_t)$ is a state-dependent (action-independent) function. While an optimal state-dependent baseline exist it is hard to find [87]. One particularly common and popular baseline is an estimate of the value function defined as

$$v(s_t) := \mathbb{E}_{\tau\sim\rho_\theta}[R_t|s_t]$$

The effect of action-dependent baselines over state-dependent baselines is subject to much debate [88], so we don't consider action-dependent baselines here. The basic REIN-

FORCE algorithm is depicted in Alg. 1.

Algorithm 1 REINFORCE algorithm

Input: a differentiable policy parametrization $\pi(a|s;\theta)$.
Algorithm parameters: step-size $\alpha>0$, and discount factor $\gamma>0$.
Initialization: Initialize policy parameters $\theta \in \mathbb{R}$ (e.g. to 0).
while True do
  Generate an episode following the policy $\pi(\cdot|\cdot;\theta)$, i.e., $s_0 \sim p_0$, $a_t \sim \pi(\cdot|s_t;\theta)$ and $s_{t+1} \sim p(\cdot|s_t,a_t)$, generating a sequence of state-actions $s_0, a_0, \ldots, s_{|\tau|-1}, a_{|\tau|-1}$
  for $t = 0$ to $|\tau|-1$ do
    $R \leftarrow \sum_{t'=t}^{|\tau|-1} \gamma^{t'-t} r_t$
    $\theta \leftarrow \theta + \alpha \gamma^t R \nabla \log \pi(a_t|s_t;\theta)$
  end for
end while

5.4 Ramp-Metering with Policy-Gradient Reinforcement Learning

Using the multi-agent system formulation introduced in Section 5.2, we are in place to use standard reinforcement learning approaches (see Section 5.3) to solve the optimal control problem of ramp-metering in arbitrary complex road sections.

In this case, the MDP $\mathscr{M} = (\mathscr{S}, \mathscr{A}, p_0, P, r, \gamma)$ is defined such that

- the state space $\mathscr{S} \in \mathbb{R}_+^N$ is the domain of all possible density values, i.e., $s := [\rho_1, \ldots, \rho_N]$;
- the action space $\mathscr{A} \in \mathbb{R}_+^N$ is the domain of all possible out-going flows, i.e., $a := [r_1, \ldots, r_N]$;
- p_0 is the initial state distribution;
- $P : \mathscr{S} \times \mathscr{A} \to \Delta(\mathscr{S})$ is the transition kernel which is assumed unknown for the RL algorithm;

- $\gamma \in [0,1)$ is a discounting factor; and

- $r : \mathscr{S} \times \mathscr{A} \to \mathbb{R}$ is the reward function which is defined as

$$r(s,a) := -J(s,a) = -(V + \alpha_1 J_1(s,a) + \alpha_2 J_2(s,a)) \tag{5.13}$$

  where $J_1(s,a) = T\sum_{i=1}^{N} s_i L_i \lambda_i$ is the travel time cost to be minimized, and $J_2(s,a) = T\sum_{i=1}^{N} a_i L_i$ is the travel distance objective to be maximized. The constant cost $V < 0$ is designed to provide a positive reward for every time step that the constraints in eq. (5.4) are not violated. This drives the system towards normal and safe operation, while the terms J_1 and J_2 drive the system towards optimal operation.

An episode of the RL algorithm starts with random initial conditions, i.e., initial policy $\pi(a|s;\theta)$ with random vector θ. The episode stops after a predefined number of time steps $|\tau|$, or if any of the constraints in eq. (5.4) are violated. While the episode is running, the RL algorithm collects the rewards r_t. When the episode is terminated, the average objective function $\mathbb{E}_{\tau\sim\rho_\theta}[\sum_{t=0}^{|\tau|-1} \gamma^t r(s_t,a_t)]$ is being approximated through Monte-Carlo simulation, and the gradient ascent update of the REINFORCE algorithm follows.

Experimental results are shown in Section 5.5.

5.5 Simulation and Numerical Results

In this section we illustrate the policy learned by the REINFORCE algorithm described in Section 5.4. In Fig. 5.5 and 5.7, we present snapshots of the evolution of the network of Fig. 5.4 under the policy learned by the REINFORCE algorithm. We compare the policies

learned using the the proposed CTM-PWA7 model to a baseline ramp-metering method and a REINFORCE policy learned using the standard non-linear CTM dynamics. The objective functions to be minimized are as defined in Section 5.4. For this task, we use a learning rate of $\alpha = 0.001$ and a discount factor of $\gamma = 0.99$. The action space $\mathscr{A} \in [0, \phi_i^{\max}]^N$ is discretized using bins of width $W_a = 1.0$. We use the 'Adam' optimization method with step sizes that evolve according to the stochastic approximation theory using $\alpha(n+1) = \frac{1}{1+\alpha_m n}$, where $\alpha_m = 0.99$. We train the agent for $n_e = 1000$ episodes, and run an additional testing run to evaluate the performance of the learned policy.

In both Fig. 5.5 and 5.7, we observe that the flows and densities of the network converge, and the output flow matches the total input flow. This is further illustrated in Fig. 5.6 and 5.8. This shows that reinforcement learning can be used as a method to find a ramp-metering policy when using the multi-agent formulation of the CTM model described above. In addition, the CTM-PWA7 model results in a policy that yields similar results in terms of safe operation and maximum throughput. We can observe that the policy learned for the CTM-PWA7 model allows slightly higher maximum density inside the road sections (see Fig. 5.6 and 5.8), but the convergence to that policy appears to be faster by a factor of 40% in terms of total episodes needed in the experiments conducted.

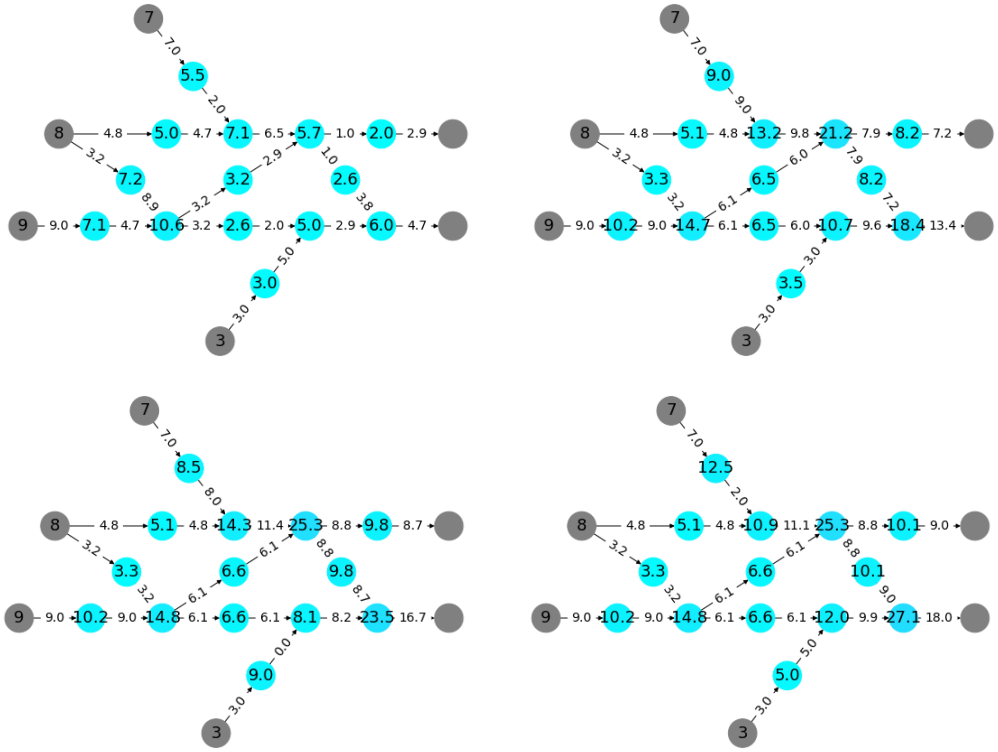


Figure 5.5: Evolution of density and flow at time steps $t \in \{0, 50, 100, 150\}$, using the ramp metering policy learned by the REINFORCE algorithm using the original non-linear model.

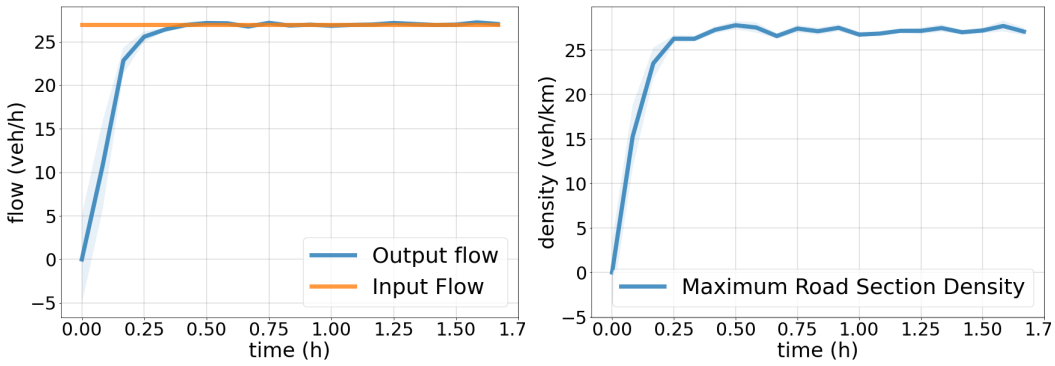


Figure 5.6: Evolution of the maximum density and output flow, using the ramp metering policy learned by the REINFORCE algorithm using the original non-linear model.

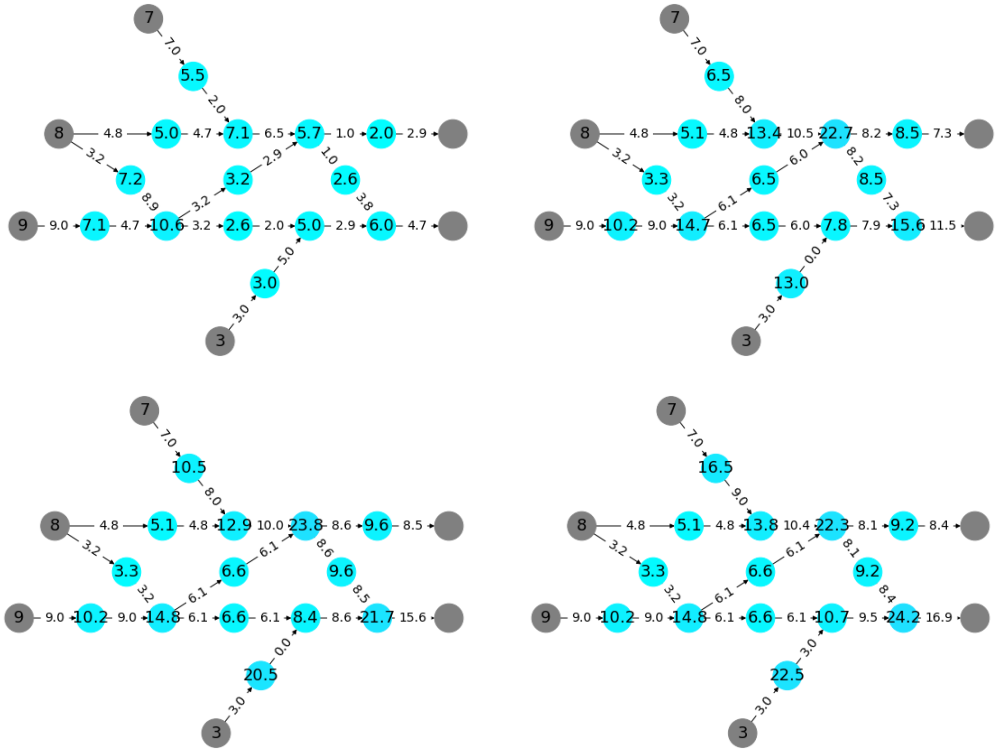


Figure 5.7: Evolution of density and flow at time steps $t \in \{0, 50, 100, 150\}$, using the ramp metering policy learned by the REINFORCE algorithm using the CTM-PWA7 model.

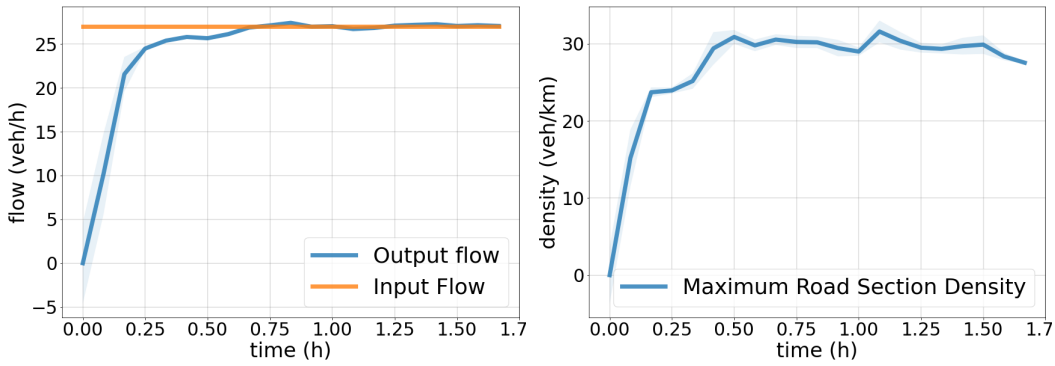


Figure 5.8: Evolution of the maximum density and output flow, using the ramp metering policy learned by the REINFORCE algorithm using the CTM-PWA7 model.

To understand what a naive ramp metering policy would look like without the formulation described here and the use of reinforcement learning approaches, we present in Fig.

5.7 and 5.10 a constant ramp metering algorithm which results from a predefined red-green light cycle that provides on average a flow of $(f_{c_1}, f_{c_1}) = (6,2)veh/km$. Observe the accumulation of vehicles on the on-ramps and the monotonic increase of the maximum density of the network in Fig. 5.10. This is a result of the output flow not being able to match the input flow for the entire simulation time.

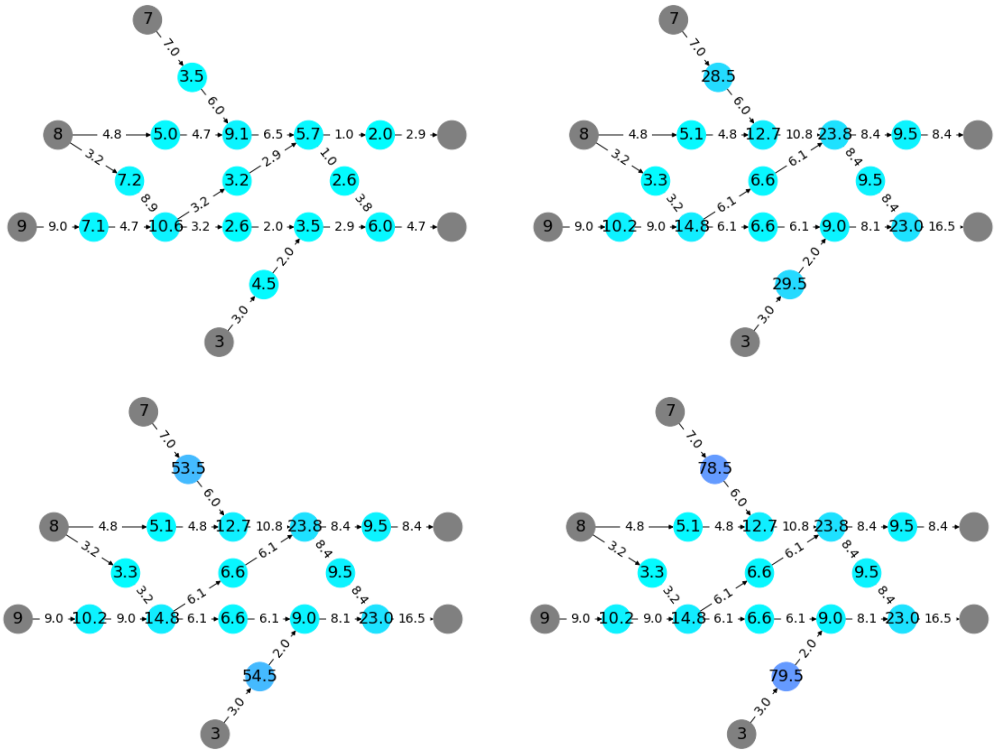


Figure 5.9: Evolution of density and flow at timesteps $t \in \{0, 50, 100, 150\}$, using a constant ramp metering policy.

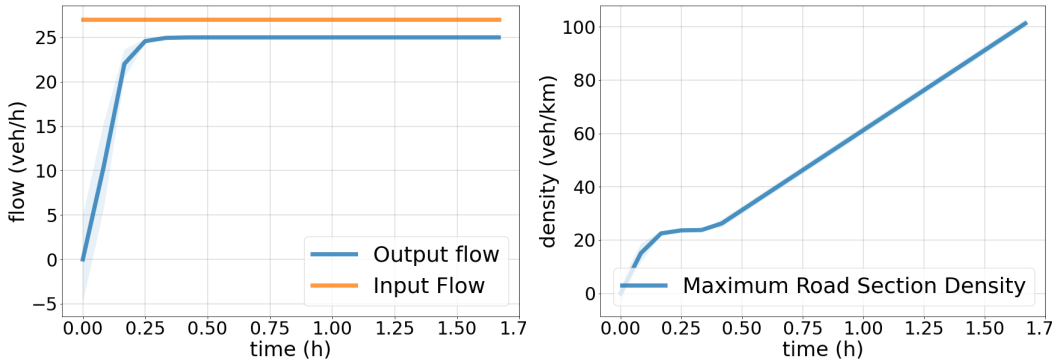


Figure 5.10: Evolution of the maximum density and output flow, using a constant ramp metering policy.

Finally, comparing the results of the Reinforcement Learning method with those of the optimization approach of chapters 3 and 4, it is important to keep in mind that the constraints of eq. (5.4) are not directly applied to the RL algorithm. Rather, the reward function is used to incorporate these constraints. This may result in the algorithm not being able to learn a policy that satisfies the constraints, or, on the other hand, a policy that does satisfy the constraints but does not take into account the optimality from the J_1 and J_2 objective functions. In other words, quite a few experimentation is necessary for the RL algorithms to converge to an acceptable solution, i.e., for the parameters above to be found through experimentation. This is in contrast to the model-based approach developed in Chapters 3 and 4.

Finally, we should make a comment on the complexity of the algorithm. A minimum of $n_e = 1000$ training episodes are required for the RL policy to converge. While the RL algorithm does not need a known model for the system, the PWA approach of chapters 3 and 4 serve as a better candidate for the trade-off between accuracy and complexity, since the

PWA approximation greatly reduces the computation time of the exhaustive optimization algorithm. We believe this to be an example of how prior information from the field can enhance the existing algorithms when handled in an appropriate way, instead of ignoring this information and only relying in non-robust generic algorithms that require high volume of parameter tuning.

5.6 Conclusion

In this chapter, a *Reinforcement Learning* (RL) algorithm was offered to solve the optimal control problem of ramp metering (RM). In the case of RM, the RL approach enables the agents to obtain optimal strategy in an unfamiliar environment without prior knowledge on traffic flow, reducing the need for human knowledge. To reach this goal, at first, the traffic flow model CTM was modified as a multi-agent system of non-homogeneous networked agents. In the following, a novel formulation of the RM problem as an optimal control problem based on the multi-agent dynamical system was offered. Then, applying policy gradient RL algorithm, a probabilistic policy was found that solved the ramp-metering problem. Simulation results show that the flow and density evolution of all the links of the network are fully compatible with the total demand of the network and it represent that the RM agents are able to perform based on an optimal policy which was learned without any prior knowledge on traffic flow. As such, our approaches have led to promising results.

Chapter 6: Conclusion

In this dissertation, after a performance analysis of different first-order CTM versions in chap. 2, a PWA extension of this model was proposed in Chap. 3 where its features enable this model to better match the needs of traffic management studies. Next, in Chap. 4, a linearized version of the second-order METANET traffic model was offered which will highly benefit the complexity and computational requirements of solving any optimization problem based on this model. In the next phase of this study, the focus was shifted to the design and evaluation of a learning-based framework for the RM strategy and a learning-based ramp metering algorithm was introduced in Chap. 5. The performance of this algorithm was evaluated from different points of view and it was shown that the RL approach enables the agents to obtain optimal strategy in an unfamiliar environment without prior knowledge on traffic flow, reducing the need for human knowledge.

6.1 Future Work

- For the class of PWA-CTMs and PWA-METANET offered in Chapters 3 and 4, the FD used was based on the benchmark parameters of this diagram widely used in the literature. This could be extended by deriving the parameters of the FD from the synthetic data generated in Chap. 4. Also, instead of assuming that all links follow

the same FD parameters, in other words, assuming to have homogeneous links, a unique FD can be extracted and utilized for each link. This will extend the traffic flow models offered in these two chapters to be compatible with real-world highway networks with non-homogeneous links.

- The models offered in chapters 3 and 4 can be simply extended to *multi-lane multi-class models* more suitable for real-world simulations. The term multi-class refers to the capability of the traffic flow model to represent the dynamics of different types of vehicles.

- For all the optimization problems in this dissertation, the RM variables of the optimization problems are considered as continuous variables. For real-world ramp metering control, the RM variables are the number of vehicles per hour entering the mainline from an on-ramp and they should be integer values. For all the simulation results, a future work would be to define an additional optimization problem which gets the continuous variables and finds their corresponding integer values by minimizing the error between the the continuous and integer values.

- PWA-CTM is a first-order model while the PWA-METANET model is a second-order model. A general comparison between the performance of the PWA-CTM and PWA-METANET could be done to investigate the advantages and disadvantages of each class.

- The main benefits of the PWA-METANET model is in having an extra dynamical

equation of the model (mean velocity equation). As such, for better evaluation of the proposed PWA model, it would be best to investigate its performance for Variable Speed Limit (VSL) control techniques rather than for RM control.

- The PWA-CTMs and PWA-METANET offered in Chapters 3 and 4 could be used in model predictive control or event-triggered control schemes and their performance could be compared with the original nonlinear CTM and METANET models in these receding horizon fashions as well.

- To solve the nonlinear programming problems of Chap. 4 other non-open source solvers could be tried and the computation time and iteration numbers could be compared with that of the fmincon solver.

- The performance of the policy gradient reinforcement algorithm offered in Chap. 5 could be compared with that of other RM algorithms like PI-ALINEA or other RL algorithm like deep Q-Learning and PPO to better evaluate the performance of the proposed method.

- We would like to investigate the effect of risk-sensitivity in the optimal solution, by applying new risk-sensitive policy-gradient reinforcement learning algorithms based on exponential criteria.

Bibliography

[1] Wikipedia contributors. Fundamental diagram of traffic flow — Wikipedia, the free encyclopedia, 2021. [Online; accessed 25-May-2021].

[2] A. Ferrara, A. Nai Oleari, S. Sacone, and S. Siri. Freeways as systems of systems: A distributed model predictive control scheme. *IEEE Systems Journal*, 9(1):312–323, 2015.

[3] Robin T Underwood. Speed, volume, and density relationships. 1960.

[4] Lukas Ambühl, Allister Loder, Michiel Bliemer, Monica Menendez, and Kay W Axhausen. A functional form for the macroscopic fundamental diagram with a physical meaning. *Arbeitsberichte Verkehrs-und Raumplanung*, 1306, 2017.

[5] Noortje Groot, Bart De Schutter, Solomon Zegeye, and Hans Hellendoorn. Model-based predictive traffic control: A piecewise-affine approach based on metanet. *IFAC Proceedings Volumes*, 44(1):10709–10714, 2011.

[6] Fatemeh Alimardani, Nilesh Suriyarachchi, Faizan M. Tariq, and John S. Baras. Models and methods for intelligent highway routing of human-driven and connected-and-automated vehicles. In Stefano de Luca, Roberta Di Pace, and Chiara Fiori, editors, *Models and Technologies for Smart, Sustainable and Safe Transportation Systems*, chapter 6. IntechOpen, Rijeka, 2020.

[7] Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intelligent Transport Systems*, 11(7):417–423, 2017.

[8] Michael James Lighthill and Gerald Beresford Whitham. On kinematic waves ii. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178):317–345, 1955.

[9] Antonella Ferrara, Simona Sacone, and Silvia Siri. *Freeway traffic modelling and control*. Springer, 2018.

[10] David Levinson and Lei Zhang. Ramp meters on trial: Evidence from the twin cities metering holiday. *Transportation Research Part A: Policy and Practice*, 40(10):810–828, 2006.

[11] Neila Bhouri, Habib Haj-Salem, and Jari Kauppila. Isolated versus coordinated ramp metering: Field evaluation results of travel time reliability and traffic impact. *Transportation Research Part C: Emerging Technologies*, 28:155–167, 2013.

[12] Lachlan Faulkner, Frans Dekker, David Gyles, Ioannis Papamichail, and Markos Papageorgiou. Evaluation of hero-coordinated ramp metering installation at m1 and m3 freeways in queensland, australia. *Transportation Research Record*, 2470(1):13–23, 2014.

[13] XY Lu, CJ Wu, and XY Lu. Field test implementation of coordinated ramp metering control strategy: a case study on sr-99n. In *Proceedings of the Transportation Research Board Meeting 96th Annual Meeting*, 2018.

[14] Markos Papageorgiou and Apostolos Kotsialos. Freeway ramp metering: An overview. *IEEE transactions on intelligent transportation systems*, 3(4):271–281, 2002.

[15] Markos Papageorgiou, Habib Hadj-Salem, Jean-Marc Blosseville, et al. Alinea: A local feedback control law for on-ramp metering. *Transportation Research Record*, 1320(1):58–67, 1991.

[16] Emmanouil Smaragdis, Markos Papageorgiou, and Elias Kosmatopoulos. A flow-maximizing adaptive local ramp metering strategy. *Transportation Research Part B: Methodological*, 38(3):251–270, 2004.

[17] Emmanouil Smaragdis and Markos Papageorgiou. Series of new local ramp metering strategies: Emmanouil smaragdis and markos papageorgiou. *Transportation Research Record*, 1856(1):74–86, 2003.

[18] Tom Bellemans, B De Schutter, and Bart De Moor. Model predictive control for ramp metering of motorway traffic: A case study. *Control Engineering Practice*, 14(7):757–767, 2006.

[19] Andreas Hegyi, Bart De Schutter, and Hans Hellendoorn. Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C: Emerging Technologies*, 13(3):185–209, 2005.

[20] Ioannis Papamichail, Apostolos Kotsialos, Ioannis Margonis, and Markos Papageorgiou. Coordinated ramp metering for freeway networks–a model-predictive hierarchical control approach. *Transportation Research Part C: Emerging Technologies*, 18(3):311–331, 2010.

[21] Dimitri Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

[22] Michael James Lighthill and G Be Whitham. On kinematic waves i. flood movement in long rivers. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178):281–316, 1955.

[23] Carlos Daganzo. The cell transmission model. part i: A simple dynamic representation of highway traffic. 1992.

[24] Carlos F Daganzo. Thecelltransmissionmodel:adynamicrepresentationofhighwaytraffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(2):269–287, 1994.

[25] Gabriel Gomes and Roberto Horowitz. Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Research Part C: Emerging Technologies*, 14(4):244–262, 2006.

[26] Carlos F Daganzo. The cell transmission model, part ii: network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93, 1995.

[27] Carlos F Daganzo. The lagged cell-transmission model. 1999.

[28] Wai Yuen Szeto. Enhanced lagged cell-transmission model for dynamic traffic assignment. *Transportation Research Record*, 2085(1):76–85, 2008.

[29] Laura Muñoz, Xiaotian Sun, Roberto Horowitz, and Luis Alvarez. Traffic density estimation with the cell transmission model. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 5, pages 3750–3755. IEEE, 2003.

[30] Ajith Muralidharan, Gunes Dervisoglu, and Roberto Horowitz. Freeway traffic flow simulation using the link node cell transmission model. In *2009 American Control Conference*, pages 2916–2921. IEEE, 2009.

[31] Anupam Srivastava and Nikolas Geroliminis. Empirical observations of capacity drop in freeway merges with ramp control and integration in a first-order model. *Transportation Research Part C: Emerging Technologies*, 30:161–177, 2013.

[32] Claudio Roncoli, Markos Papageorgiou, and Ioannis Papamichail. Traffic flow optimisation in presence of vehicle automation and communication systems–part ii: Optimal control for multi-lane motorways. *Transportation Research Part C: Emerging Technologies*, 57:260–275, 2015.

[33] Fabio Morbidi, Luis Leon Ojeda, Carlos Canudas De Wit, and Iker Bellicot. A new robust approach for highway traffic density estimation. In *2014 European Control Conference (ECC)*, pages 2575–2580. IEEE, 2014.

[34] Antonella Ferrara, Simona Sacone, and Silvia Siri. Event-triggered model predictive schemes for freeway traffic control. *Transportation Research Part C: Emerging Technologies*, 58:554–567, 2015.

[35] Carlos Canudas-de Wit and Antonella Ferrara. A variable-length cell road traffic model: Application to ring road speed limit optimization. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6745–6752. IEEE, 2016.

[36] Afzal Ahmed, Satish V Ukkusuri, Shahrukh Raza Mirza, and Ausaja Hassan. Width-based cell transmission model for heterogeneous and undisciplined traffic streams. *Transportation research record*, 2673(5):682–692, 2019.

[37] Yanyan Qin and Hao Wang. Cell transmission model for mixed traffic flow with connected and autonomous vehicles. *Journal of Transportation Engineering, Part A: Systems*, 145(5):04019014, 2019.

[38] Sai Kiran Mayakuntla and Ashish Verma. Cell transmission modeling of heterogeneous disordered traffic. *Journal of Transportation Engineering, Part A: Systems*, 145(7), 2019.

[39] Rongsheng Chen, Tab Zhang, and Michael W Levin. Effects of variable speed limit on energy consumption with autonomous vehicles on urban roads using modified cell-transmission model. *Journal of Transportation Engineering, Part A: Systems*, 146(7):04020049, 2020.

[40] Yibing Wang, Elias B Kosmatopoulos, Markos Papageorgiou, and Ioannis Papamichail. Local ramp metering in the presence of a distant downstream bottleneck: Theoretical analysis and simulation study. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2024–2039, 2014.

[41] Gabriel Gomes, Roberto Horowitz, Alex A Kurzhanskiy, Pravin Varaiya, and Jaimyoung Kwon. Behavior of the cell transmission model and effectiveness of ramp metering. *Transportation Research Part C: Emerging Technologies*, 16(4):485–513, 2008.

[42] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[43] Rodrigo C Carlson, Ioannis Papamichail, Markos Papageorgiou, and Albert Messmer. Optimal mainstream traffic flow control of large-scale motorway networks. *Transportation Research Part C: Emerging Technologies*, 18(2):193–212, 2010.

[44] Zhi Liu, Ye Wu, Shipeng Cao, Linan Zhu, and Guojiang Shen. A ramp metering method based on congestion status in the urban freeway. *IEEE Access*, 8:76823–76831, 2020.

[45] Yuwei Bie, Mudasser Seraj, Can Zhang, and Tony Z Qiu. Improving traffic state prediction model for variable speed limit control by introducing stochastic supply and demand. *Journal of Advanced Transportation*, 2018, 2018.

[46] Femke van Wageningen-Kessels, Hans Van Lint, Kees Vuik, and Serge Hoogendoorn. Genealogy of traffic flow models. *EURO Journal on Transportation and Logistics*, 4(4):445–473, 2015.

[47] Jan Hueper, Gunes Dervisoglu, Ajith Muralidharan, Gabriel Gomes, Roberto Horowitz, and Pravin Varaiya. Macroscopic modeling and simulation of freeway traffic flow. *IFAC Proceedings Volumes*, 42(15):112–116, 2009.

[48] Fatemeh Alimardani and John S Baras. Performance assessment of different cell-transmission models for ramp-metered highway networks. *IFAC-PapersOnLine*, 54(2):114–120, 2021.

[49] Bruce Douglas Greenshields, JT Thompson, HC Dickinson, and RS Swinton. The photographic method of studying traffic behavior. In *Highway Research Board Proceedings*, volume 13, 1934.

[50] Harold Greenberg. An analysis of traffic flow. *Operations research*, 7(1):79–85, 1959.

[51] Gordon Frank Newell. Nonlinear effects in the dynamics of car following. *Operations research*, 9(2):209–229, 1961.

[52] Michiel CJ Bliemer and Mark PH Raadsen. Continuous-time general link transmission model with simplified fanning, part i: Theory and link model formulation. *Transportation Research Part B: Methodological*, 126:442–470, 2019.

[53] Carlos F Daganzo and Nikolas Geroliminis. An analytical approximation for the macroscopic fundamental diagram of urban traffic. *Transportation Research Part B: Methodological*, 42(9):771–781, 2008.

[54] Malachy Carey and Michael Bowers. A review of properties of flow–density functions. *Transport Reviews*, 32(1):49–73, 2012.

[55] Juraj Stevek, Alexander Szucs, Michal Kvasnica, Miroslav Fikar, and Stefan Kozák. Two steps piecewise affine identification of nonlinear systems. *Archives of Control Sciences*, 22(4):371–388, 2012.

[56] Serge P Hoogendoorn and Piet HL Bovy. State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 215(4):283–303, 2001.

[57] Xiao-Yun Lu, Tony Z Qiu, Roberto Horowitz, Andy Chow, and Steven Shladover. Metanet model improvement for traffic control. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2148–2153. IEEE, 2011.

[58] Markos Papageorgiou, Jean-Marc Blosseville, and Habib Hadj-Salem. Macroscopic modelling of traffic flow on the boulevard périphérique in paris. *Transportation Research Part B: Methodological*, 23(1):29–47, 1989.

[59] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[60] Solvers.

[61] Wikipedia contributors. Interior-point method — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Interior-point_method&oldid=1069107991`, 2022. [Online; accessed 29-May-2022].

[62] Richard H Byrd, Mary E Hribar, and Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[63] Richard A Waltz, José Luis Morales, Jorge Nocedal, and Dominique Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical programming*, 107(3):391–408, 2006.

[64] Jorge Nocedal, Figen Öztoprak, and Richard A Waltz. An interior point method for nonlinear programming with infeasibility detection capabilities. *Optimization Methods and Software*, 29(4):837–854, 2014.

[65] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.

[66] Robert L Gordon. Ramp metering. In *Intelligent Freeway Transportation Systems*, pages 115–145. Springer, 2010.

[67] Rodrigo Castelan Carlson, Ioannis Papamichail, and Markos Papageorgiou. Local feedback-based mainstream traffic flow control on motorways using variable speed limits. *IEEE Transactions on intelligent transportation systems*, 12(4):1261–1276, 2011.

[68] Azita Dabiri and Balázs Kulcsár. Distributed ramp metering—a constrained discharge flow maximization approach. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2525–2538, 2017.

[69] Michael Zhang, Taewan Kim, Xiaojian Nie, Wenlong Jin, Lianyu Chu, and Will Recker. Evaluation of on-ramp control algorithms. 2001.

[70] Bing Liu, Yu Tang, Yuxiong Ji, Yu Shen, and Yuchuan Du. A Deep Reinforcement Learning Approach for Ramp Metering Based on Traffic Video Data. *arXiv e-prints*, page arXiv:2012.12104, December 2020.

[71] Saeed Ghanbartehrani, Anahita Sanandaji, Zahra Mokhtari, and Kimia Tajik. A novel ramp metering approach based on machine learning and historical data. *Machine Learning and Knowledge Extraction*, 2(4):379–396, 2020.

[72] Dale P Masher, DW Ross, PJ Wong, PL Tuan, HM Zeidler, and S Petracek. Guidelines for design and operation of ramp control systems. 1975.

[73] Ioannis Papamichail and Markos Papageorgiou. Traffic-responsive linked ramp-metering control. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):111–121, 2008.

[74] Mohsen Davarynejad, Andreas Hegyi, Jos Vrancken, and Jan van den Berg. Motorway ramp-metering control with queuing consideration using q-learning. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1652–1658. IEEE, 2011.

[75] Kasra Rezaee, Baher Abdulhai, and Hossam Abdelgawad. Application of reinforcement learning with continuous state space to ramp metering in real-world conditions. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1590–1595. IEEE, 2012.

[76] Ahmed Fares and Walid Gomaa. Freeway ramp-metering control based on reinforcement learning. In *11th IEEE International Conference on Control & Automation (ICCA)*, pages 1226–1231. IEEE, 2014.

[77] Hao Yang and Hesham Rakha. Reinforcement learning ramp metering control for weaving sections in a connected vehicle environment. Technical report, 2017.

[78] Francois Belletti, Daniel Haziza, Gabriel Gomes, and Alexandre M Bayen. Expert level control of ramp metering based on multi-task deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1198–1207, 2017.

[79] Fuwen Deng, Jiandong Jin, Yu Shen, and Yuchuan Du. Advanced self-improving ramp metering algorithm based on multi-agent deep reinforcement learning. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3804–3809. IEEE, 2019.

[80] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 2018.

[81] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level Control Through Deep Reinforcement Learning. *Nature*, 518(7540):529–533, 2015.

[82] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. *arXiv preprint arXiv:1312.5602*, 2013.

[83] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning With Function Approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063, 2000.

[84] Sham M Kakade. A Natural Policy Gradient. *Advances in Neural Information Processing Systems*, 14:1531–1538, 2001.

[85] Shixiang Shane Gu, Timothy Lillicrap, Richard E Turner, Zoubin Ghahramani, Bernhard Schölkopf, and Sergey Levine. Interpolated Policy Gradient: Merging On-policy and Off-policy Gradient Estimation for Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*, pages 3846–3855, 2017.

[86] Ronald J Williams. Simple Statistical Gradient-following Algorithms for Connectionist Reinforcement Learning. *Machine learning*, 8(3-4):229–256, 1992.

[87] Lex Weaver and Nigel Tao. The Optimal Reward Baseline for Gradient-based Reinforcement Learning. *arXiv preprint arXiv:1301.2315*, 2013.

[88] George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard Turner, Zoubin Ghahramani, and Sergey Levine. The Mirage of Action-Dependent Baselines in Reinforcement Learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5015–5024, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.