DAS Departamento de Automação e Sistemas
CTC Centro Tecnológico
UFSC Universidade Federal de Santa Catarina

Optimal Control of Semi-continuous Models for Freeway Congestion Mitigation

Relatório submetido à Universidade Federal de Santa Catarina
como requisito para a aprovação da disciplina:
DAS 5511: Projeto de Final de Curso

Angelo Frigeri Araujo

Florianópolis, Janeiro de 2018

Optimal Control of Semi-continuous Models for Freeway Congestion Mitigation

Angelo Frigeri Araujo

Esta monografia foi julgada no contexto da disciplina

DAS 5511: Projeto de Final de Curso

e aprovada na sua forma final pelo

Curso de Engenharia de Controle e Automação

Prof. Rodrigo Castelan Carlson

____

Banca Examinadora:

Prof. Eduardo Camponogara / GOS – DAS – UFSC
Orientador na Empresa

Prof. Rodrigo Castelan Carlson
Orientador no Curso

Prof. Hector Bessa Silveira
Responsável pela disciplina

Prof. Henrique Simas, Avaliador

Willian de Medeiros Galvani, Debatedor

Wesley André Bortolozo Junior, Debatedor

Acknowledgements

Firstly and mostly, I would like to express my very great appreciation to my wife, Daniela Frigeri, for her full support in this journey, comprehending the time I invested in studying and writing this work, helping save the precious time, so to have a full dedication to my Final Course Project.

I am particularly grateful to my father Edson, my mother Fanny, my brother Leo, and my grandmother Ita, for supporting me during my under graduation course, spending their time, energy and resources so I could spend mine to my career.

I offer my special thanks for the assistance given by my friend and professor Eduardo Camponogara, who adopted me as his student since my very beginning in under graduation, investing in my study and advising me through challenges.

My special thanks are extended to my advisor in this PFC and professor, Rodrigo Carlson, to my friends Marco Aguiar and Eduardo Hülse, who helped me when things seen impossible or unreasonable. When theory was too much difficult, they showed me the way out to understanding.

To the other members of GOS, Luiz Guardini, Leonardo Assis, Caio Giuliani, Thiago Lima, Bruno Benetti, Andres Codas and Jean Jordanou, I appreciate all your contributions and relaxing moments.

Last but not least, I thank all the DAS department (professor and technical staff) for making possible the transformation of students into engineers.

"For I am convinced that neither death nor life
nor angels nor governments nor things now here
nor things to come nor powers nor height nor depth
nor any other creation will be able to separate us
from God's love that is in Christ Jesus our Lord.
(Holy Scriptures, Romans 8:38, 39)

Resumo

Propõe-se um modelo de tráfego rodoviário macroscópico semi contínuo baseado no “Cell Transmission Model”. O modelo incorpora o limite de velocidade como parâmetro variável. Além disso, é proposto o emprego de algoritmos de controle ótimo para a resolução de problemas de otimização dinâmica que contenham o modelo proposto na forma de Equações Diferenciais Algébricas (DAEs). Testes em diferentes cenários de simulação, com condições de tráfego adversas, mostram que o sistema de controle é capaz de impedir ou, pelo menos, reduzir os efeitos prejudiciais do congestionamento no sistema de tráfego.

Keywords: Controle, Optimização, Controle Ótimo, Modelagem de Tráfego, Tráfego de Rodovias, Congestionamento.

Abstract

A semi-continuous model for macroscopic traffic in freeways is proposed, based on Cell Transmission Model and other second-order continuous models. The proposed model embraces the speed limit as the manipulated variable. Moreover, optimal control algorithms are applied in order to solve the dynamic optimization problem in the form of Differential Algebraic Equations (DAEs). Tests in different simulation scenarios with different adverse traffic conditions show that the controlled system is capable of avoiding, or at least decreasing, the harmful effects of traffic jams, traffic congestion and stop and go waves, commonly seen in traffic systems.

Keywords: Control, Optimization, Optimal Control, Traffic Modeling, Freeway Traffic, Traffic Jam.

List of Figures

Contents

1 Introduction

Congestion in traffic is increasingly found in a several cities worldwide. Drivers in Los Angeles, California, spent over 100 hours a year in traffic jams in 2016, 81 hours in 2015, and 64 hours in 2013 - more than any other city in the world [1] [2] [3].

Brazil's scenario is increasingly getting worse. São Paulo is the 6th city in the world IN which drivers spend more hours in traffic congestion. They spend an average of 77 hours a year in traffic congestion. In 2015, the same city was in the 10th rank position. Belém, Pará, ascended from 72nd (2015) to 24th (2016) in world rank position, spending an average of 54 hours a year. São José, Santa Catarina ascended from 69th (2015) to 39th (2016) in world rank position, spending an average of 47 hours a year [4].

In London, UK by spending 70 hours a year in traffic congestion have a cost of more than £1, 911.00 per driver, and more than £6 billion across the city as a whole [5]. Proportionally, drivers in São José, SC, Brazil spend R$ 4, 461.00 each a year due to traffic jams. [1]

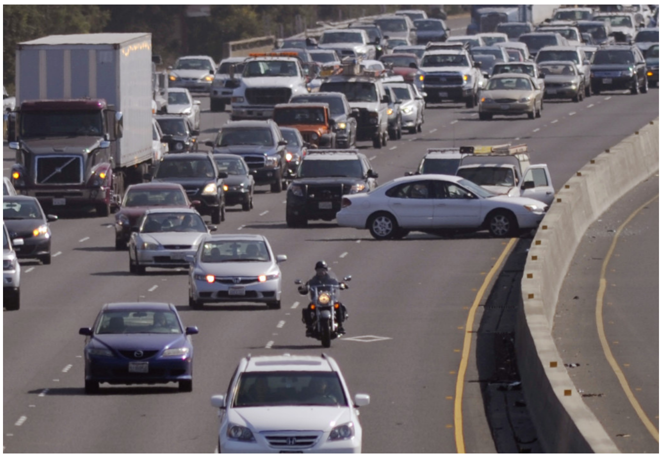


Figure 1 – Traffic congestion caused by an accident. Image source.

Traffic jams are also caused by traffic accidents, disabled vehicles, severe weather, road construction, etc. [6] Figure 1 shows a car (involved in an accident) that blocks two lanes of a freeway, resulting in a temporary reduction in the roadway capacity.

[1] It is computed the volume of fuel spent per driver by considering the fuel price as £1.21 per liter. Volume per driver = 1580 liters. Then, it is set the proportion "liters per hour" as constant for both cities = 22.6 liters of fuel per hour. In São José, SC, Brazil the time spent in traffic congestion was 47 hours, spending approximated of 1062 liters of fuel. The fuel price considered is R$ 4, 20 per liter (January/2018). Finally: 1062 liters × R$ 4, 20 per liter = R$ 4, 461.00

1.1 Motivation

"Engineering or technology is all about using the power of science to make life better for people, to reduce cost, to improve comfort, to improve productivity", said N.R. Narayana Murthy, the co-funder of Infosys Technologies Limited [7].

In fact, engineering has the power to bring a better life to people. For example, the automation of jobs performed by human work brought a solution with low operating costs, and optimized performance, to increasingly complex systems. Automatic Control has proved to be a very effective technique to not only solve challenging problems in engineering, but also to improve human life.

Although limited in their performance, Conventional Control Techniques – such as Proportional, P.I. and P.I.D. – had a very important role in the early years of automatic control implementation in industries and other areas [8].

Modern Control Techniques emerged from the conventional ones, but bringing more benefits. With prediction and optimization, it has been possible to achieve not only a good solution from a short-time perspective, but the best solution possible in a long-time perspective.

1.2 Objectives

This work aims to apply Optimal Control (a Modern Control Technique that uses optimization to achieve better performance) in traffic scenarios subjected to congestion.

More specifically, the research focuses on the manipulation of speed limits in all sections of a freeway to mitigate traffic jams, traffic congestions and stop-and-go waves. This control is applied in real time, changing the speed limits in time stamps, while measuring both vehicle speeds and the number of vehicles along the freeway (density, concentration).

Figure 2 exemplify the use of dynamic speed limits, that are updated in order to adapt speed limits to freeway conditions. This work presents a control technique and a traffic model for freeways that improve the speed in accidents situations, mitigating traffic jams and other damaging effects.

1.3 Document Structure

This document is composed of four chapters.

Chapter 2 explains the principles of traffic engineering, brings the main concepts, involved variables and equations, and the fundamental theory

Figure 2 – Electronic Speed Limit. Image source.

Chapter 3 gives a brief introduction to optimal control, explains the theory and tools that surround this area. Also, it shows what control features are present in Optimal Control.

Chapter 4 models a traffic system in a suitable way for the application of optimal control and its mathematical formulation for computational implementation.

Chapter 5 brings results of applying Optimal Control Techniques for the control speed limits in order to reduce damaging traffic effects, and the interpretation of results.

Chapter 6 concludes this work with further comments about the results, the conclusion of this research and future works.

2 Traffic Control

This chapter highlights the traffic dynamics theory and its main aspects regarding mathematical modeling. This includes the explanation of certain essential concepts, a discussion on the main differences between microscopic and macroscopic traffic modeling, as well as the most common control techniques found in literature and their models. Mostly of this chapter is based on [9]. Once the above is stated, conclusions about open-loop traffic behavior and the desired performance will be discussed.

2.1 Traffic Fundamentals

This section gives the basic background for understanding traffic dynamics, the commonly mentioned variables and parameters. The relationship between these variables and parameters will be shown in the fundamental equations. At last, their connection with a real scenario will be discussed.

2.1.1 Flow

Flow is one of the primary measures of traffic condition or state [10]. In Fluid Dynamics Theory, flow is described as the volume of liquid passing through a fixed area over time. Likewise, the flow, q (veh/h), in Traffic Dynamics is described as the quantity of vehicles that passes the cross-section at a location in space within a time interval:

$$q = \frac{N}{T} \tag{2.1}$$

with N (veh) being the number of vehicles, and T (h) the time interval.

This traffic characteristic is important not only because it ensures movement, but also because its relation with other variables can propose conditions which make easy to diagnose some key phenomena, such as stop and go waves, traffic congestions and traffic jams.

Differently from some traffic variables, the average flow is calculated by the harmonic mean:

$$q = \frac{1}{\frac{1}{N}\sum\limits_{i=1}^{N}\frac{1}{q_i}} \tag{2.2}$$

with q_i being the flow attributed to an individual vehicle i – a unit of flow – which can be understood as an elementary contribution to the road's total flow [9].

The proof for using harmonic mean in calculating traffic flow is provided in the next subsection.

2.1.2 Headway

Headway is the gap between two consecutive vehicles in the same lane – it can be measured as a time headway, in which case the gap is given in time units, or it can be measured as a space headway, in which case the gap is given in distance units.

Considering a single lane road and a vehicle as a point in the space (length equals to zero), in (2.1) the time T can be defined as the sum of the time headways, h_i (h), for all vehicles:

$$T = \sum_{i=1}^{N} h_i. \tag{2.3}$$

Replacing the new definition of time from (2.3) in (2.1), traffic flow can be defined as:

$$q = \frac{N}{\sum\limits_{i=1}^{N} h_i} = \frac{1}{\frac{1}{N}\sum\limits_{i=1}^{N} h_i} = \frac{1}{\bar{h}} \tag{2.4}$$

in which $\bar{h}$ (h) is the mean headway. In words, the flow is the inverse of the mean time headway.

Therefore, for a single vehicle, a unit of flow can be computed as:

$$q_i = \frac{1}{h_i}. \tag{2.5}$$

Replacing the unit of headway from (2.5), as $h_i = \frac{1}{q_i}$ in (2.4), it is shown that the average of flow is indeed the harmonic mean, proving the statement (2.2) in the previous subsection.

2.1.3 Speed

Speed is one of the most important variables in traffic. It assures movement: a non-zero speed means that a vehicle varies its position in time. The measured speed can be relative to one single vehicle or it can be relative to a set of vehicles. In case of multiple vehicles, two different but related speeds can be calculated.

Time Mean Speed

Time mean speed, v_t (km/h), is the average of the speed of all vehicles passing a fixed point in space over a duration of time:

$$v_\mathrm{t} = \frac{1}{N}\sum_{i=1}^{N} v_i \tag{2.6}$$

with v_i (km/h) being the speed of the i^{th} vehicle, and N the total number of vehicles with measured speed.

Space Mean Speed

Space mean speed v_{s} (km/h) results by considering the average time $\bar{t}$ (h) spent by a set of vehicles to travel a fixed length L (km) in space:

$$v_{\mathrm{s}} = \frac{L}{\bar{t}}. \tag{2.7}$$

Computing the average time as:

$$\bar{t} = \frac{1}{N}\sum_{i=1}^{N}\frac{L}{v_i} \tag{2.8}$$

and by replacing in (2.7), the Space Mean Speed can be defined as:

$$v_{\mathrm{s}} = \frac{L}{\frac{1}{N}\sum\limits_{i=1}^{N}\frac{L}{v_i}} = \frac{N}{\sum\limits_{i=1}^{N}\frac{1}{v_i}}. \tag{2.9}$$

2.1.4 Concentration

Concentration can be understood as the number of vehicles along a fixed segment of a road. Although this definition implies measurement along a distance, traffic engineers have traditionally estimated concentration k from point measurements, using the relationship:

$$k = \frac{q}{v_{\mathrm{s}}} \tag{2.10}$$

with q being the flow and v_{s} the space mean speed in the given segment of road.

This is acceptable because concentration varies with speed. The higher the speed, the larger the spatial headway – in regular traffic conditions, time headway is not affected by speed variation – making vehicles more distant from each other, therefore, less concentrated in the road.

2.1.5 Diagrams

The relation among the variables presented above can be more easily seen in graphics and diagrams, highlighting their behavior over their respective variance.

Figure 3 shows the relationship between flow and density (concentration). When the density is zero (origin), there are no vehicles in the road, therefore, no traffic, and flow equals to zero. When the density is maximum (k_{j}) – called "jam density" – the concentration is so high that vehicles are tightly together – flow is equal to zero, because there is not available space for vehicles to perform any movement in the road. Regular traffic conditions are denoted by densities k_{A} and k_{B}. This curve is not necessarily well defined, similar to a quadratic function. Different models produce different diagrams. Several non linear models are proposed in literature, each with different behaviors and different relations among variables, producing different diagrams.

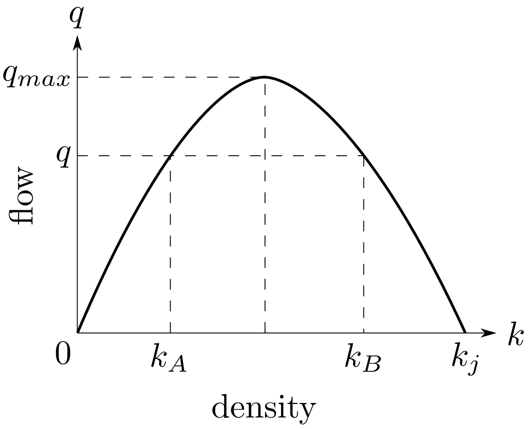


Figure 3 – Flow Density Curve

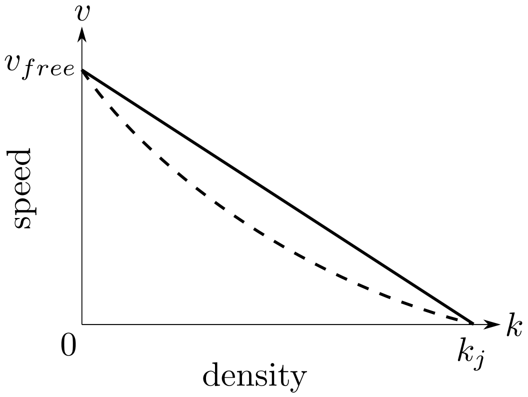


Figure 4 – Speed Density Curve

Figure 4 shows the relationship between speed and density (concentration), this diagram is also called **Fundamental Diagram**. When the density is zero (origin), the speed of vehicles is maximum because the road is free. In fact, this scenario is called *free flow speed*. When the density is at its maximum (k_{j}), the speed is equal to zero because there is not any movement or flow – same scenario stated in the previous diagram. For the same previous reason, the curve comes from modeling that can be non linear, as shown in the dotted curve.

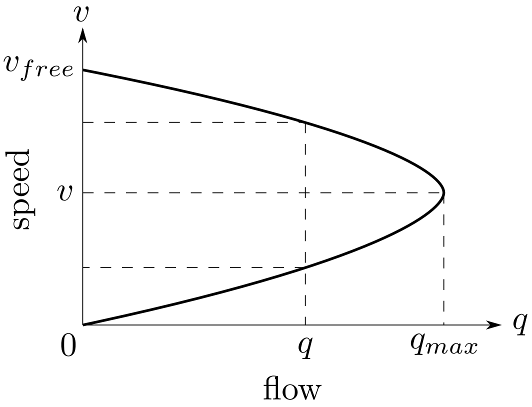


Figure 5 – Speed Flow Curve

Figure 5 shows the relationship between speed and flow. There are two scenarios to cause flow to be zero (origin in the diagram):

1. There are too many vehicles in the road, making it impossible to move: speed is also zero, therefore;
2. There is no vehicle in the road, allowing the desired speed to be equal to the *free flow speed*.

The maximum flow q_{max}, or capacity flow, is reached with a speed lower than the maximum *free flow speed*, or critical speed.

In fact, the three curves can be related in a single diagram given in Figure 6.

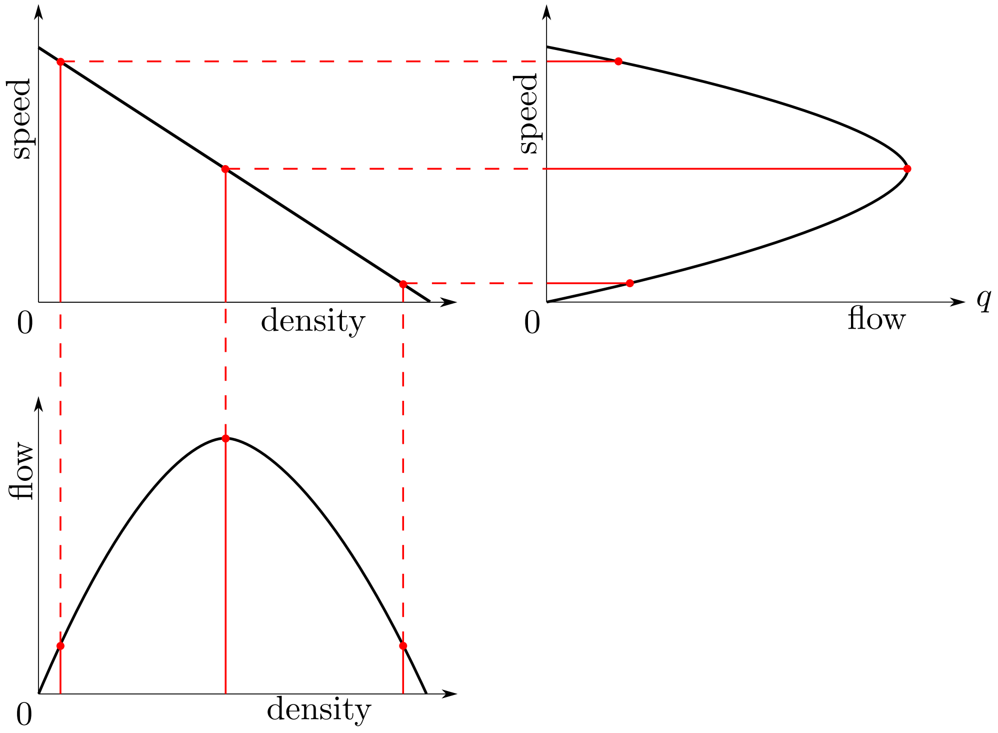


Figure 6 – Relation Between The Fundamental Flow Traffic Diagrams

Figure 6 shows the method to build the diagram in Figure 5: given a density, selecting the resulting flow and speed, then relating them in a new diagram. Hence, the two regular traffic conditions (previously mentioned and denoted by k_{A} and k_{B}) can be easily understood:

1. The first scenario is in roads with low car density – the concentration leads car to a higher speed than usual, that is balanced by the low volume of vehicles, maintaining flow in a regular state.

2. The other scenario happens in more highly concentrated traffic, with higher density, speed decreases, but is balanced by an increased number of cars, keeping flow in a regular state.

Considering equation (2.10), the non linearity of models, and the scenarios above, it can be seen that the best condition (when flow is maximum) is reached balancing both speed and density.

2.2 Microscopic vs Macroscopic Traffic

This section explains the concepts of Microscopic and Macroscopic Traffic Modeling, providing their respective definitions, commonly studied scenario, mathematical models and applications.

2.2.1 Microscopic Traffic

Microscopic Traffic (or Microsimulation) is a category of traffic modeling that simulates the behavior of individual vehicles within a predefined scenario.

Mathematically, Microscopic Traffic Modeling delineates for each vehicle i the position $x_i(t)$, velocity $v_i(t)$ and acceleration $\dot{v}_i$ in time. Moreover, the behavior of each vehicle, and the interactions between vehicles, are computed according with the scenario using the so-called car-following models, which describe the processes by which drivers follow each other in the traffic stream [1].

Microscopic Traffic usually embraces scenarios of city streets, where the interactions and the behavior of vehicles are tightly coupled, and should be taken into consideration. Networks can be used in applications of such models. These networks are mostly composed of nodes and arcs, which represent streets and crossings, with parameters regarding the physical construction.

The most common applications of Microscopic models are:

- Modeling how single vehicles affect traffic;
- Describing human driving behavior, such as estimation errors, reaction time, inattentiveness, and anticipation;
- Visualization of interactions between different types of vehicles (cars, trucks, buses, cyclists, pedestrians, etc.);

[1] Other models that are worthy of being mentioned are Lane Changing Models and Gap Acceptance Models.

- Validation and optimization of infrastructure projects of streets, crossings, roundabouts, and traffic signal timings.

2.2.2 Macroscopic Traffic

Differently from Microscopic Traffic, Macroscopic Traffic does not simulate an individual behavior. Although it considers vehicle parameters, what in fact is modeled is the dynamic state of roads in time and space. It is described analogously to liquids or gases in motion, to mention *hydrodynamic models*. Mathematically, Macroscopic Traffic delineates the collective vehicle dynamics in terms of concentration $\rho(x,t)$, flow $Q(x,t)$, mean speed $V(x,t)$ and acceleration $A(x,t)$ in space and in time. The interaction between vehicles is indirectly computed because road parameters include desired speed, capacity (maximum flow), and others parameters that interfere in the dynamics.

Macroscopic Traffic usually embraces scenarios of roads and highways. To simulate such scenarios, both continuous and discrete models can be used. For example, cellular automaton models sample time and space, dividing the highway in cells, where, according to maximum and minimum functions and parameters, vehicles travel from a given cell to the next.

The most common applications of Macroscopic models are:

- Physical project of on- and off-ramps, number of lanes, etc;
- Real-time control of speed-limit, on-ramp metering, stop-and-go waves avoidance, congested traffic avoidance, etc.

2.3 Modeling

Traffic flow models can be categorized by their mathematical structure, some are continuous in both time and space, others are discrete in one or both time and space. Three traffic flow models are presented below. Although not studied in this work, car-following is going to be mentioned briefly. Two macroscopic models are detailed next, the first order Cell Transmission Model, which is discrete in time and space, and a second-order model continuous in time and space.

2.3.1 Microscopic Model

As mentioned earlier, microscopic traffic models represent the interaction between vehicles under street constraints. A multitude of car-following models have been proposed, both for single-lane and multi-lane traffic including lane changes [11]. Therefore, only one model is presented.

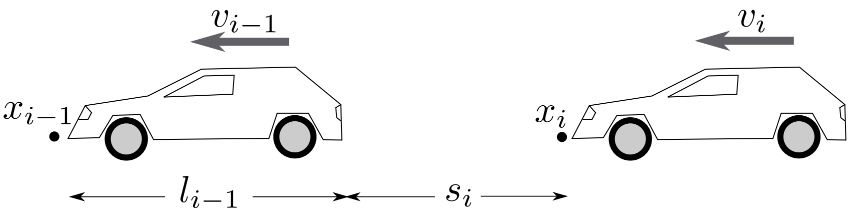


Figure 7 – Bumper to bumper gap

As the name suggests, car following models are based on the distance from one vehicle to another – the vehicle on the back "follows" with a distance the vehicle in front (leader). Since what the driver sees from the vehicle ahead is the car's bumper, the modeling starts by setting the bumper-to-bumper distance gaps:

$$s_i = x_{i-1} - l_{i-1} - x_i \tag{2.11}$$

where s_i is the gap between vehicles i (follower vehicle) and $i-1$ (leader vehicle), and l_{i-1} is the length of the leader vehicle. Figure 7 illustrates this configuration.

Many realistic car following models describe the response of the vehicle as a function of:

- the gap s_i to the leader vehicle;
- the leader vehicle's current speed v_{i-1};
- the vehicle's current speed v_i.

Considering a continuous time and space modeling, mathematically it can be expressed as:

$$\dot{x}_i = \frac{dx_i(t)}{dt} = v_i(t), \tag{2.12}$$

$$\dot{v}_i(t) = \frac{dv_i(t)}{dt} = a_{\text{mic}}(s_i, v_i, v_{i-1}). \tag{2.13}$$

One example to be mentioned is the Intelligent-Driver Model (IDM) [12], where the acceleration $a_{\text{mic}}(s_i, v_i, v_l)$ assumed is a continuous function of the speed v_α, gap s_i and leader vehicle's speed v_{i-1}:

$$\dot{v}_i = a\left[1 - \left(\frac{v_i}{v_0}\right)^\delta - \left(\frac{s^*}{s_i}\right)^2\right] \tag{2.14}$$

where a is the maximum acceleration, v_0 the desired speed, s^* is the desired gap, and the exponent δ is typically between 1 and 5 [9]. The desired gap s^* has the relation:

$$s^*(v_i, v_{i-1}) = s_0 + \max\left(v_iT + \frac{v_i(v_i - v_{i-1})}{2\sqrt{ab}}, 0\right) \tag{2.15}$$

in which s_0 is the minimum gap, also called “jam distance,” T is the safe time headway in congested but moving traffic, a is an acceleration term and b a deceleration term, where both reflects an “intelligent” braking strategy.

Further explanation about microscopic traffic theory, mathematical modeling and historical accounts can be obtained in the literature [13].

2.3.2 Cell Transmission Model

In the Cell Transmission Model (CTM) [14], the road is divided into homogeneous sections (cells), with lengths chosen such as the time traveled by a vehicle along a cell is equal to one clock tick. Each cell is numbered consecutively, starting with the upstream end of the road, from $i = 1$ to I. Time is discretized in time t.

Thus, the system’s evolution obeys:

$$n_{i+1}(t+1) = n_i(t) \tag{2.16}$$

where $n_i(t)$ is the number of vehicles in cell i at time t. It can be seen that the concept is of a queue, where, when the clock runs, vehicles advance from one cell to the other. To ensure the real traffic behavior, important constrains are necessary to be stated:

1 $n_i(t)$: the number of vehicles in cell i at time t;

2 $Q_i(t)$: the capacity of flow into cell i for time interval t;

3 $N_i(t) - n_i(t)$: the amount of empty space in cell i at time t.

where N_i is the storage capacity - the maximum density of cars in cell i.

Item 1 ensures that the number of vehicles to pass from one cell to the next cannot be greater than the existing number of vehicles in the respective cell. Item 2 ensures that there is a maximum flow of vehicles moving from one cell to the next. Item 3 ensures that the number of vehicles (density) in every cell will not be greater than the physically allowed.

Considering the conservation of vehicles, the number of vehicles at the next time, will be equal to the number of vehicles at the current time, plus the inflow, minus the outflow:

$$n_i(t+1) = n_i(t) + y_i(t) - y_{i+1}(t) \tag{2.17}$$

where $y_i(t)$ is the inflow into cell i. Notice that the outflow from cell i is the inflow into cell $i+1$.

Considering the 3 constraints stated in the items above, the outflow can defined as follows:

$$y_i(t) = \min\{n_{i-1}(t), Q_i(t), N_i(t) - n_i(t)\} \tag{2.18}$$

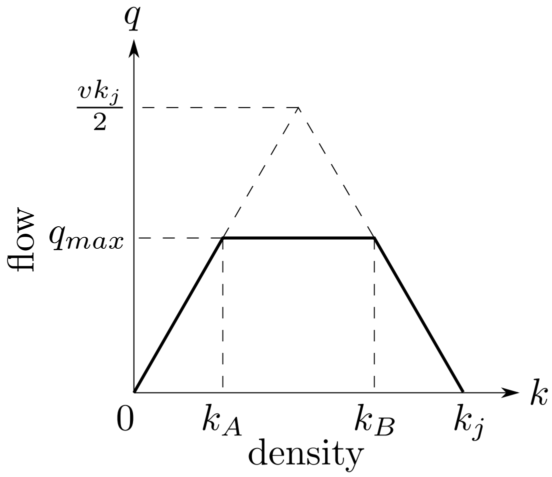


Figure 8 – CTM Flow Density Diagram

This model produces the diagram in Figure 8, which shows that the CTM model approximates the flow-density relation also given in Figure 3. Three important types of traffic behavior can be induced by the density (to enhance the understanding, numerical examples are considered: $k_{\mathrm{A}} = 50$, $k_{\mathrm{B}} = 100$, $k_{\mathrm{j}} = 150$, $q_{\max} = 50$):

- From 0 to k_{A}: the speed remains the same regardless of the traffic density (free traffic speed). The closer the density gets to k_{A}, the closer traffic approaches maximum flow.

  In this scenario, the minimum in Equation (2.18) is imposed by the first term, $n_{i-1}(t) \leq k_{\mathrm{A}} = 50$. Since the density in cell i is low, there is remaining space available for vehicles to flow in from the previous cell $i-1$ ($N_i - n_i \geq k_{\mathrm{A}}$). So, the number of vehicles in cell $i-1$, being smaller than the maximum flow ($q_{\max} = 50$), is the number of vehicles that flows in cell i;

- From k_{A} to k_{B}: the speed decreases while density increases. However, flow remains constant at its maximum. Although the speed decreases, the concentration of vehicles compensates for the traffic flow, remaining at its maximum.

  In this scenario, both the number of vehicles in the previous cell ($n_{i-1} \geq k_{\mathrm{A}} = 50$) and the available space ($N_i - n_i \geq k_{\mathrm{A}} = 50$) are larger than Q_i. Therefore, the amount of vehicles that flows in cell i is determined by the second term in Equation (2.18), $Q_i(t) = 50$;

- From k_{B} to k_{j}: the speed and the flow decrease while density increases. Traffic is saturated, and deprecated by the high density of vehicles.

  In this scenario, the minimum in Equation (2.18) is enforced by the term $N_i(t) - n_i(t)$. Because traffic is saturated, the current number of vehicles n_i is close to the maximum N_i, driving the available space in cell i close to zero. Numerically: $n_i(t) \geq k_{\mathrm{B}} = 100$,

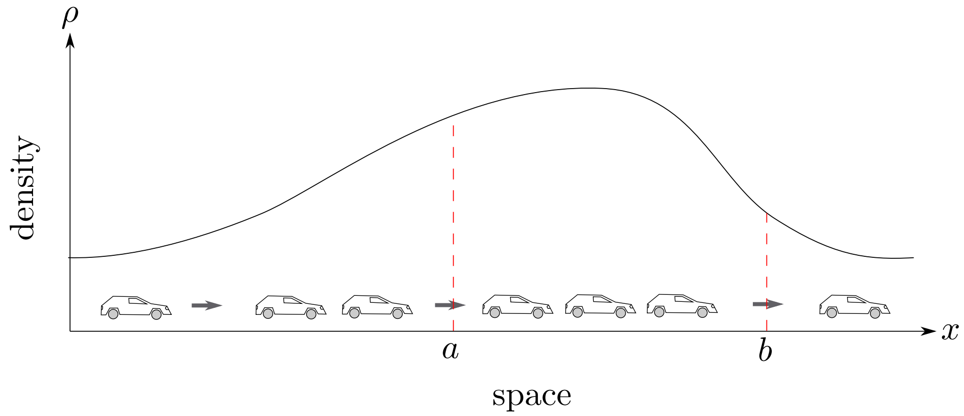


Figure 9 – Illustrative Example of Density Along Space

$q_{\text{max}} = 50$, $N_i - n_i \leq k_{\text{A}} = 50$. Therefore, the number of vehicles that flow in cell i is limited by the available space in it;

Further explanation about the cell transmission model and its different applications can be obtained in the literature [14, 15].

2.3.3 Continuous Macroscopic Model

Macroscopic Traffic is strongly related to Fluid Dynamics. The highway is decomposed into basic elements, which are modeled by similarity with hydrodynamic systems, by assuming that the flow of vehicles is similar to the movement of a fluid through a pipe [16]. Therefore, what is studied are the fundamental variables in roads (differently from the car-following model, and similar to CTM).

Figure 9 illustrates an example of vehicle distribution along a road, constituting a density along space.

By adding up the number of vehicles in a given length of the road, and then dividing it by the length, the density is obtained. Equation (2.19) shows the opposite: number of vehicles is a function of the density in a continuous space:

$$\int_a^b \rho(t,x)dx = [\text{number of vehicles between points } a \text{ and } b]. \tag{2.19}$$

By computing the number of vehicles that passes through point a and b in space during a given time, the flow of vehicles is defined. Equation (2.20) expresses the result by considering time infinitively small:

$$\frac{d}{dt}\int_a^b \rho(t,x)dx = [\text{inflow at } a] - [\text{outflow at } b] = -(q(t,b) - q(t,a)) \tag{2.20}$$

in which $q(t,x)$ is the flow of vehicles at point x and time t.

Algebraic manipulations lead to:

$$\frac{\partial}{\partial t}\int_a^b \rho(t,x)dx = -\left(\frac{\partial}{\partial x}\int_a^b q(t,x)dx\right) \tag{2.21}$$

$$\frac{\partial}{\partial t}\rho(t,x) = -\frac{\partial}{\partial x}q(t,x) \tag{2.22}$$

$$\frac{\partial}{\partial t}\rho(t,x) + \frac{\partial}{\partial x}q(t,x) = 0 \tag{2.23}$$

Equation (2.23) is a fundamental relation between density and flow in both space and time, establishing a law of dynamic conservation in traffic theory [9].

Combining with the Continuity Equation (2.10), Equation (2.23) can be recast as:

$$\frac{\partial}{\partial t}\rho(t,x) + \frac{\partial}{\partial x}\rho(t,x)v(\rho) = 0 \tag{2.24}$$

which renders the Conservation Equation (2.24) dependent only on density $\rho(t,x)$ and speed $v(\rho)$. To obtain a dynamic and continuous model, it is only needed an equation that models speed as a function of density: the acceleration equation.

The acceleration equation is characterized by speed V, density ρ, gradients and nonlocalities. Most acceleration equations from second-order models can be represented by a general form:

$$a = \frac{dV}{dt} = \frac{\partial V}{\partial t} + V\frac{\partial V}{\partial x} = \frac{V_e^*(\rho, V, \rho_a, V_a) - V}{\tau} + \frac{k}{\rho}\frac{\partial \rho}{\partial x} \tag{2.25}$$

where k is a constant, and sub-index a represent spatial anticipation.

It is important to highlight three important terms in the general form of the model equations:

- **Relaxation**: it is the local speed adaptation, which describes the acceleration fraction of vehicles according to the variation of concentration in the local neighborhood. Constant τ is the speed adaptation time, the time a vehicle takes to reach steady state;
- **Anticipation**: is the nonlocal speed adaptation, describes the acceleration fraction of vehicles according to the variation of concentration at an anticipated location, ahead of the actual position ($x_a > x$);
- **Traffic Pressure**: describes a response of the local ensemble of vehicles on density gradients. It is the acceleration fraction according to the change of position with different density.

The first fraction $\frac{V_e^*(\rho,V,\rho_a,V_a)-V}{\tau}$ in Equation (2.25) contains both **Relaxation** and **Anticipation** terms, because V_e^* is a function of concentration (and speed V) at the local neighborhood ρ and also dependent of density (and speed V_a) at an anticipated location ρ_a.

The current location (local neighborhood) defines the **Relaxation** term. The anticipated location (ahead of the actual position) defines the **Anticipation** term. The last fraction $\frac{k}{\rho}\frac{\partial\rho}{\partial x}$ defines the **Traffic Pressure** term, because the derivative models the density gradient, the contribution for speed when vehicles drives from different concentrated locations.

Several proposed models follow (2.25), such as Payne-Whitham Model, KK Model, Gas-Kinetic Model, to mention some.

2.4 Summary

This chapter shows the main idea of traffic modeling for both microscopic traffic (urban areas) and macroscopic traffic (freeways). It also emphasizes the main variables and the interpretation of them in each case. For traffic in freeways, the main equations and components are described and explained in order to propose new models.

3 Optimal Control

This chapter gives a simple presentation of optimization and non-linear problem formulation, optimality conditions, and numerical methods (based on [17] and [18]). Then, a brief introduction to dynamic systems and control systems. The chapter further states the definition of Optimal Control Problems (OCP), and some methods used to solve dynamic equations in an optimization problem.

3.1 Optimization

Optimization is a branch of mathematics that concerns the computation of optimal values for decision variables, that, in turn, assures optimal performance and satisfies constraints from a mathematical model.

Optimization Problems (OP) have three main elements:

- **Decisions Variables:** parameters whose values define a solution for the problem. In models, these parameters can be defined as production quantities, the electric voltage applied to a motor, resources consumed and other examples;
- **Objective Functions:** functions that contain decisions variables and other parameters, which measure the performance to be minimized or to be maximized. In models, these functions can represent the total cost in production, the number of hours spent to produce goods, and leftovers in resources, among others;
- **Constraints:** a set of functions that define the feasible solutions, establish limits and boundaries; the domain space within which the decisions variables represent a solution for the problem. In models, these constraints can express the maximum number of working hours for an employee, the minimum production amount of a certain product and maximum resources available, to name a few.

As in other other disciplines, in optimization, we seek models that represent the reality of a specific system, process or phenomenon. Despite not being a trivial task, modeling is very important in optimization as a poor model can lead to unsatisfactory results.

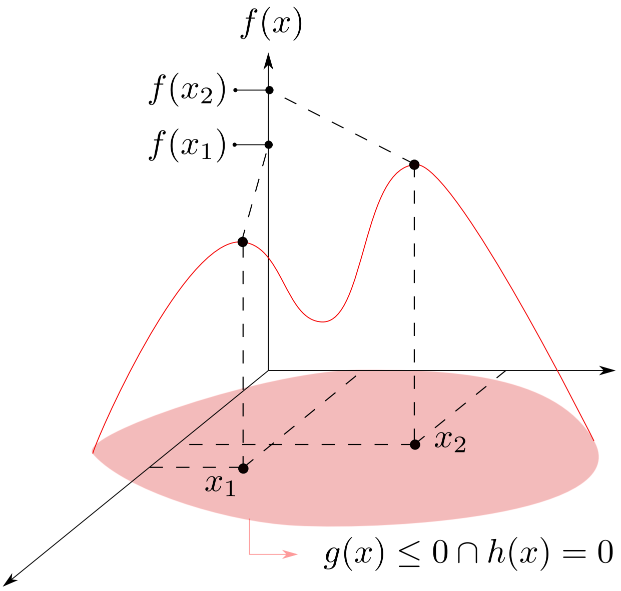


Figure 10 – Nonlinear objective function with a two-dimensional domain

A general formulation for an optimization problem (OP) is:

$$\mathcal{OP}_g : \min_{x} \ f(x) \tag{3.1a}$$

$$\text{s.t.} : \begin{cases} g(x) \leq 0 \\ h(x) = 0 \\ x \in \mathbb{R}^n \end{cases} \tag{3.1b}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function; $g : \mathbb{R}^n \to \mathbb{R}^m$ are inequality constraints; $h : \mathbb{R}^n \to \mathbb{R}^p$ are equality constraints; and $x \in \mathbb{R}^n$ is the vector with decision variables.

Figure 10 shows an objective function $f(x)$ within a feasible space: a local maximum $f(x_1)$ and a global maximum $f(x_2)$.

If the objective function is concave (maximization problem) or convex (minimization problem) and the constraint set is convex, then the problem can be defined as a convex optimization problem, for which the local optima are also global optima.

When concavity is not assured, the problem is said to be a non-convex optimization problem and a local optimum is usually sought.

Several numerical methods can solve convex optimization problems: Newton's (for equality constrained problems) and interior-point method (for inequality constrained problems) will be presented.

3.1.1 Newton Method with Equality Constraints

A general problem addressed by this method can be formulated as follows:

$$\begin{aligned} \mathcal{OP}_n : \ \min_x \ & f(x) && (3.2a) \\ \text{s.t.} : \ & Ax = b && (3.2b) \end{aligned}$$

where f is convex, twice continuously differentiable, and A is a $\mathbb{R}^{p \times n}$ matrix with rank p and $n > p$.

The Newton method is iterative and depends on the computation of two parameters:

- **Newton Step** Δx_{nt}: it is computed from the gradient $\nabla f(x)$ and Hessian $\nabla^2 f(x)$ of the objective function:

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{nt} \\ w \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ 0 \end{bmatrix} \tag{3.3}$$

- **Newton Decrement** $\lambda(x)$: it is computed from the Newton Step Δx_{nt} and Hessian $\nabla^2 f(x)$ of the objective function.

$$\lambda(x) = \sqrt{\Delta x_{nt}^T \nabla^2 f(x) \Delta x_{nt}} \tag{3.4}$$

The Newton Method for Equality Constrains is given by the following algorithm:

```
Input: $x^0$ = feasible solution, $\epsilon > 0$
let $x = x^0$ be the initial solution;
while $\frac{\lambda^2}{2} > \epsilon$ do
    compute Newton Step $\Delta x_{nt}$;
    compute Newton Decrement $\lambda$;
    update $x := x + t\Delta x_{nt}$;
    choose a step size $t > 0$ by backtracking line search
end
```

Algorithm 1: Newton Method with Equality Constraints

Since the initial step x^0 is feasible, the method ensures that all iterates x^k remain feasible, which converge to a local optimum.

3.1.2 Interior Point (Logarithmic Barrier) Method

A general problem addressed by this method can be formulated as follows:

$$\mathcal{OP}_{ip} : \min_{x} \quad f(x) \tag{3.5a}$$

$$\text{s.t.} : \quad \begin{cases} g_i(x) \leq 0, & i = 1, ..., m \\ Ax - b = 0 \end{cases} \tag{3.5b}$$

where both f and g_i are convex, twice continuously differentiable, and A is a $\mathbb{R}^{p \times n}$ matrix with rank p (full row rank), and $n > p$.

This method consists in removing the inequality constraints and accounting for them in the objective, using a function I that is equal to zero when the constraints g_i are satisfied, and equal to infinity otherwise.

Such a function is called "exact penalty", which is not differentiable. Thus, a smooth approximation of I is introduced with a logarithmic penalty Φ:

$$I \approx \left(\frac{1}{t}\right) \Phi(x) = -\left(\frac{1}{t}\right) \sum_{i=1}^{m} \log(-g_i(x)) \tag{3.6}$$

where t is the *barrier parameter*, which improves the approximation of function I as t increases.

For the problem defined in Equation (3.5), the interior-point method is stated below:

```
Input : $x^0$ = feasible solution, $t^0 > 0$, $\mu > 1$, $\epsilon > 0$,
let $x$ be initially equal to $x^0$;
let $t$ be initially equal to $t^0$;
while $\frac{m}{t} > \epsilon$ do
    compute $x^*(t)$ by minimizing $tf(x) + \Phi(x)$, subject to $Ax = b$;
    update $x := x^*$;
    update $t := \mu t$
end
```

Algorithm 2: Interior Point Method

3.2 Dynamic Systems

Dynamic systems are characterized by a set of dynamic equations (differential equations) that describe the behavior of physical systems (and their phenomena) with respect to time, and other variables that influence the dynamic.

A dynamic equation is a mathematical model that attempts to predict the state of the system. It is a mathematical description of the ratio of change as a function of other variables, for example:

$$\frac{dx}{dt} = -x \tag{3.7}$$

Equation (3.7) brings a dynamic equation example, which is described mathematically by the ratio of change of state x as a function of its own value.

This ratio of change can be better understood from the definition of derivative:

$$\dot{x}(t) = \frac{dx(t)}{dt} = \lim_{\Delta t \to 0} \frac{x(t + \Delta t) - x(t)}{\Delta t} \tag{3.8}$$

Therefore, a differential equation describes how a state changes in time, in an infinitesimal time interval.

The given example is simple, however, there are different types of differential equations of increasing complexity. Concerning linearity of the states, there are *linear* and *nonlinear* differential equations. Concerning the coefficients, there are *time invariant* and *time variant* differential equations. Also, a differential equation can be *ordinary*, which means that it is defined with respect to only one independent variable. To express the behavior of a system, one may combine differential equations and algebraic equations, resulting in differential-algebraic equations (DAE).

3.2.1 Ordinary Differential Equation

An Ordinary Differential Equation (ODE) is a differential equation with only one independent variable, usually time.

Generally, a system of ODEs can be defined as a set of differential equations, whose ratios of change are modeled with respect to only one variable, as a function of their states.

An ODE can be described by:

- an independent variable t, usually time;
- a state variable $x(t)$;
- a dynamic vector function $f(x(t), y(t), t)$.

Mathematically, an ODE can be defined as follows:

$$\dot{x} = f(x, t) \tag{3.9}$$

Equation (3.9) is a general representation for every ODE system, where the function f defines the system's behavior. The more complex the dynamic function f, the more difficult to find an analytical solution. In most real cases, the solution of a complex ODE system is found through numerical methods.

The example below shows an analytical solution of a simple ODE and the respective numerical solution.

Example: Acceleration based on referential speed

The rate of change of the speed is known as the acceleration. Also, it can be based on a referential speed V_0. When the state variable v is lower than V_0, the acceleration is positive. When state variable v is greater than V_0, the acceleration is negative. In fact, acceleration either increases or decreases according with the signal of the difference from the desired speed and the current speed:

$$\dot{v}(t) = \frac{dv}{dt} = \frac{V_0 - v}{\tau} \tag{3.10}$$

which the speed v has only time as the independent variable, and τ is the *time constant*. Time constant τ represents the response time of the system dynamics. The state variable v achieves 95% of the reference V_0 when time $t = 3\tau$.

The analytical solution of the ODE in Equation (3.10) is presented below. First, rewrite in the form of a first order separable ODE:

$$\frac{1}{-v + V_0} dv = \frac{1}{\tau} dt \tag{3.11}$$

Then, integrate both sides:

$$\int_{v(t_0)}^{v(t_f)} \frac{1}{-v + V_0} dv = \int_{t_0}^{t_f} \frac{1}{\tau} dt \tag{3.12}$$

Solving the integrals:

$$\ln(-v(t_f) + V_0) - \ln(-v(t_0) + V_0) = -\frac{t_f}{\tau} \tag{3.13}$$

Applying the exponential operator on both sides:

$$\frac{-v(t_f) + V_0}{-v(t_0) + V_0} = \exp\left(-\frac{t_f}{\tau}\right) \tag{3.14}$$

Isolating the state variable:

$$v(t_f) = V_0 + (v(t_0) - V_0) \exp\left(-\frac{t_f}{\tau}\right) \tag{3.15}$$

where $v(t_0)$ is the initial speed, usually zero. In this special case:

$$v(t_f) = V_0 \left(1 - \exp\left(-\frac{t_f}{\tau}\right)\right) \tag{3.16}$$

It is now easier to see that, since the exponential decreases to zero as t increases to infinity, speed (the state variable) tends to be equal to the referential speed V_0.

3.2.2 Differential Algebraic Equation

Differential Algebraic Equations (DAEs) can be understood as Ordinary Differential Equations (ODEs) with algebraic constraints, which improve the mathematical modeling of the system dynamics, thus, representing a larger variety of physical systems.

A DAE can be described by:

- an independent variable t, usually time;
- a state variable $x(t)$;
- an algebraic variable $y(t)$;
- a dynamic vector function $f(x(t), y(t), t)$;
- an algebraic vector function $g(x(t), y(t), t)$;

Mathematically, a DAE can be defined as follows:

$$\dot{x} = f(x, y, t) \tag{3.17a}$$

$$0 = g(x, y, t) \tag{3.17b}$$

where the function g represents the algebraic constraints. Since this function can assume different forms, the DAE is classified in indexes, which are defined as the number of differentiations needed for function g to represent y by its time derivative.

The DAE classification is important because it imposes mathematical conditions and properties that define the proper numerical method to obtain the DAE system.

Example: Acceleration based on referential speed

Take the same previous example, however, the referential speed V_0 is no longer a constant, but a function $g(t)$ of time:

$$V_0(t) = g(t) = \left(-\frac{V_a}{t_a^2}\right) t^2 + \left(\frac{2V_a}{t_a}\right) t + V_a \tag{3.18}$$

which generates the curve in Figure 11.

Therefore, the DAE system equation is:

$$\dot{v}(t) = \frac{V_0(t) - v(t)}{\tau} \tag{3.19a}$$

$$0 = V_0(t) - g(t) \tag{3.19b}$$

which can be numerically solved using IDAS from Sundials [19].

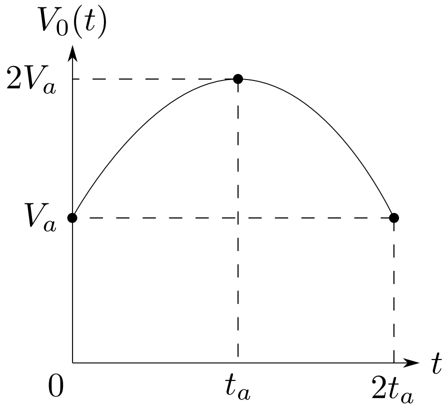


Figure 11 – Referential Speed Curve

3.2.3 Controlled Systems

A controlled system is a special case of dynamic system because it has feedback. This feedback consists in comparing the system output with a reference, generating an error (a difference). An input signal is computed from this error, and input to the system.

This process can be illustrated as a closed loop block diagram, as shown in Figure 12, where:

- y: is the system output;
- r: is the reference (also called set point - S.P.), what the signal output y should follow;
- e: is the error, the difference from reference and output signal;
- Ψ: is the control law, a logic or function that computes the control signal;
- u: is the control signal, the parameter that feedbacks into the dynamic equation;
- f: is the dynamic equation;
- g: is the algebraic equation.

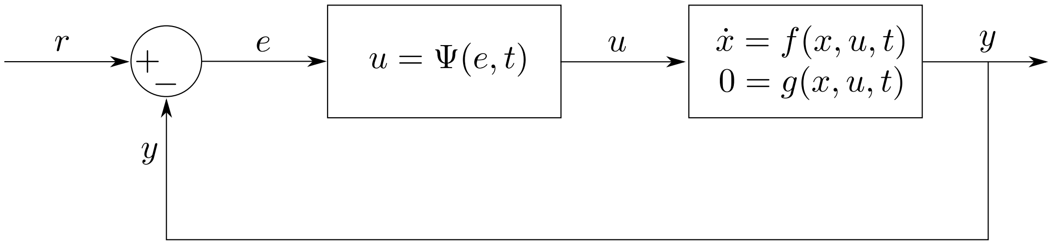


Figure 12 – Closed-loop block diagram.

Two variables are of greater importance:

Manipulated Variable: the variable which the controller inputs in the system; the variable that can be modified according to the control law;

Controlled Variable: the variable which the controller strives to draw toward the set point; the output of the system, usually the measured variable, which will produce an error when compared to the reference.

The control of this system is dependent on both the Controlled Variable and Set Point, ensuring that, in time, the Controlled Variable will follow the reference given. For this reason, systems with closed-loop controllers are also called feedback systems [20].

Example: Shower temperature control

A trivial and basic example of a feedback control system, used almost everyday by humans, is to adjust the temperature and flow of the water in the shower by opening two valves – the valve with hot water and the valve with cold water – until the temperature feels comfortable. In order to measure the temperature, hands or arms are used to feel whether it is too hot, too cold, or pleasant. When the temperature is too high, the hot water valve can be adjusted to decrease the hot water flow, or, the flow of cold water can be increased by adjusting the respective valve.

This example illustrates a controlled system, for we have a closed loop controller: the output variables (the controlled variables - y) are the water flow and water temperature of the shower. The input variables (manipulated variables - u) are the valve opening of both hot and cold water. A human uses their own body as sensor to measure the output variables and compare them with his referential temperature (r), creating an error signal (e) and adjusting the input variables until the output variables reach the set point (a comfortable temperature and a nice water flow).

A mathematical model could be extracted from this example, however, for instance, the theoretical approach is enough to illustrate a simple control problem. Of course, other factors are involved, which were not considered in the example above, such as the delay from the time instant the valve is opened or closed until the real effect it takes on the water temperature (temperature is a very common example to illustrate systems with delay). Another factor not considered is the constraint of valve opening; there is a maximum opening to the valve, which imposes boundaries to the manipulated variables.

There are plenty of controllers that account for the two factors above (delay and constraints), however, many of them require a model, a mathematical description of the system dynamic, in order to predict the output behavior towards the manipulation of the manipulated variable.

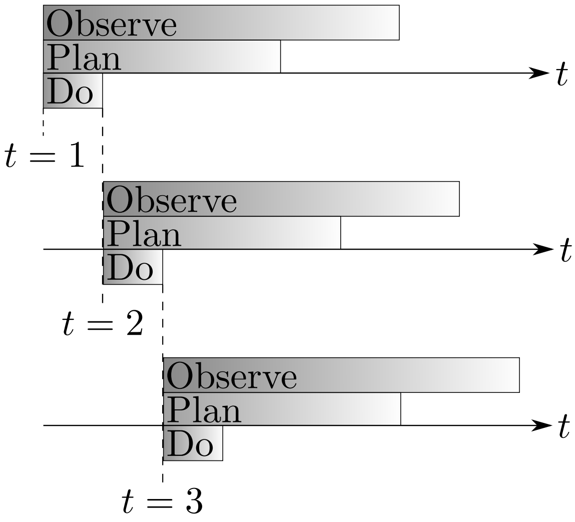


Figure 13 – Prediction horizon progress over time.

3.2.4 Predictive Control

The most popular Predictive Control method is the Generalized Predictive Control (GPC) [21]. The basic idea of GPC is to compute a sequence of future control signals that minimize the objective function of an optimization problem (cost function) defined over a prediction horizon.

This prediction horizon is finite and made from time windows, whose length is a function of the discretization sampling time. Figure 13 exemplifies the prediction horizon progress over time.

For a GPC, a linear and discretized model is necessary to simulate the system dynamics, which allows to compute output variables for every given input variable. Therefore, it is possible to predict the system behavior in future time instants, mounting a prediction horizon.

The future variables, such as the output variable y, can be denoted as follows:

$$y(t+k \mid t) \tag{3.20}$$

where $t+k$ is the time prediction in window k, computed at the current time t.

Since the set point is the desired behavior, it can be considered as known for future time windows. Therefore, the future error can also be computed:

$$e(t+k \mid t) = r(t+k \mid t) - y(t+k \mid t) \tag{3.21}$$

where e is the error signal computed by the future reference (known variable) and future output (computed by the mathematical model).

With the prediction of the error signal, it can be computed the control signal in the prediction horizon. Usually, the cost function is made from two weighted factors; the first is the sum of the error squared for each window in the prediction horizon (multiplied by the error weight factor - δ); the second is the sum of the squared control signals for each window in the prediction horizon (multiplied by the control weight factor - λ).

Considering that the cost function is to be minimized, the first term is responsible for reference following, and the second term responsible for the control behavior. For a conservative behavior, the weight on control signal should be larger than the weight on error signal ($\lambda > \delta$ for normalized variables). The opposite results in a non-conservative behavior ($\delta > \lambda$ for normalized variables).

Since the predictive control method computes the control signal by solving an optimization problem, constraints can be easily considered. Also, if the system time delay is within the prediction horizon, the GPC provides a satisfactory solution, where the output follows the reference in a smooth or fast way.

3.2.5 Lyapunov Stability

There are two important methods for stability proposed by Lyapunov for continuous-time systems [22].

Before addressing the first Lyapunov Stability Method, it is import to define equilibrium point:

Equilibrium point definition:

(x^e, u^e) is an equilibrium point for the system in (3.22)

$$\dot{x} = f(x, u, t) \tag{3.22a}$$

$$0 = g(x, u, t) \tag{3.22b}$$

if the conditions in (3.23) are satisfied.

$$0 = f(x^e, u^e, t) \tag{3.23}$$

The first Lyapunov Stability Method observes each pair of equilibrium point (x^e, u^e) given an initial condition (x_0, u_0). A given state can be defined as a stable equilibrium point if every point starting "close enough" to it remains close enough forever (within a certain distance).

Mathematically, an equilibrium point is *stable* if for each $\epsilon > 0,\ \exists\ \delta > 0$ such that, for each x_0 such that $||x_0 - x^e|| < \delta$, then for every $t > t_0$ it is satisfied $||x(t) - x^e|| < \epsilon$.

There are two stability definitions proposed by Lyapunov.

Asymptotic Stability:

If the solution that starts close enough to the equilibrium point not only remains close enough to it, but also eventually converges to the equilibrium point:

$$\lim_{t \to \infty} ||x(t) - x^e|| = 0 \tag{3.24}$$

which means that the state distance from the equilibrium point eventually decreases to zero.

Exponential Stability:

If the solution not only converges to the equilibrium point (asymptotically stable), but as fast as a known rate:

$$||x(t) - x^e|| \leq \alpha ||x(0) - x^e|| \exp(-\beta t) \tag{3.25}$$

for all $t \geq 0$, $\alpha > 0$ and $\beta > 0$.

The second Lyapunov Stability Method makes use of a function $V(x(t))$ which has an analogy to the potential function (or energy). Considering a system $\dot{x} = f(x, t)$ with an equilibrium point x^e, Lyapunov ensures stability if the following conditions for a function $V(x) : \mathbb{R}^n \to \mathbb{R}$ are satisfied:

- $V(x) = 0$ if, and only if, $x = x^e$;
- $V(x) > 0$ if, and only if, $x \neq x^e$;
- $\dot{V}(x) = \frac{dV(x)}{dt} = \sum\limits_{i=1}^{n} \frac{\partial V(x)}{\partial x_i} f_i(x) \leq 0$, $\forall x \neq x^e$.

in other words: $V(x)$ must be zero at the equilibrium point, positive definite, and $\dot{V}(x)$ must be negative semidefinite. Then, the equilibrium point is asymptotically stable.

Moreover, if the radial unboundedness condition is satisfied. : $V(x) < c, \forall x$, which means that $||x - x^e|| < d, \forall x$, then, the equilibrium point is globally asymptotically stable (GAS).

3.3 Optimal Control Problems

Optimal Control combines both optimization and control areas. The idea is to find the control actions that minimize a cost function. If the cost function can be related to a set points, then, the error, usually the quadratic error between the output variable and the set point, is minimized; for instance, it could minimize the error between desired speed and the current speed or to minimize the error between the desired altitude of an airplane and its current altitude.

An objective functions may not contain a set point, usually economic objectives. In this case the goal is to find the maximum or the minimum of a function depending on the state and control variables; it could minimize the waste of supplies or to maximize the (net or gross) profit.

The most used application of optimal control is within model predictive control, where an optimal control problem (OPC) is solved at each sampling time.

This section presents the mathematical formulation for an OPC, as well as numerical methods for solving this family of problems. This section is mainly based on [23], but also contains references from [24] and [25].

3.3.1 Formulation

Like in an optimization problem, an OCP consists of one or more objective functions to be minimized, and a set of constraints defined by the model itself and boundaries imposed on variables and parameters.

To formulate an OCP, it is needed the state vector $x(t) \in \mathbb{R}^{N_x}$, the control vector $u(t) \in \mathbb{R}^{N_u}$, the output vector $y(t) \in \mathbb{R}^{N_y}$, and the interval $T = [t_0, t_f]$ defined by the initial and final simulation time. Moreover, functions are needed, such as the differential equation f, the algebraic equation g, a final cost function Φ and a dynamic cost $\mathcal{L}$.

Constraints are formulated by the above functions, which model the problem, physical limits, variable boundaries and initial conditions.

Thus, an OCP can be formulated as follows:

$$\mathcal{P}_{OCP} : \ \min_{u} \ J(x,y,u) = \Phi(x(t_f), y(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(x,y,u,t)\, dt \tag{3.26a}$$

$$\text{s.t.} : \begin{cases} \dot{x} = f(x,y,u,t), \\ 0 = g(x,y,u,t), \\ h_{\text{eq}}(x,y,u,t) = 0 \\ h_{\text{ineq}}(x,y,u,t) \leq 0 \\ x(t_0) = x_0 \\ x_{\text{L}} \leq x \leq x_{\text{U}} \\ y_{\text{L}} \leq y \leq y_{\text{U}} \\ u_{\text{L}} \leq u \leq u_{\text{U}} \end{cases} \tag{3.26b}$$

where:

1. J is the objective function;
2. Φ is the final cost function (Mayer Term);
3. $\mathcal{L}$ is the dynamic cost (Lagrange Term);
4. f and g are differential and algebraic equations that formulate the system equation;
5. h_{eq} are the quality constraints;
6. h_{ineq} are the inequality constraints;
7. x_0 is the initial condition state vector;
8. x_{L}, y_{L} and u_{L} are the lower bounds for the state, output and control vectors, which impose minimum limits for these variables;
9. x_{U}, y_{U} and u_{U} are the upper bounds for the state, output and control vectors, which impose maximum limits for these variables.

Some basic properties on the underlying functions are required in order to problem $\mathcal{P}_{OCP}$ in (3.26) to be well defined:

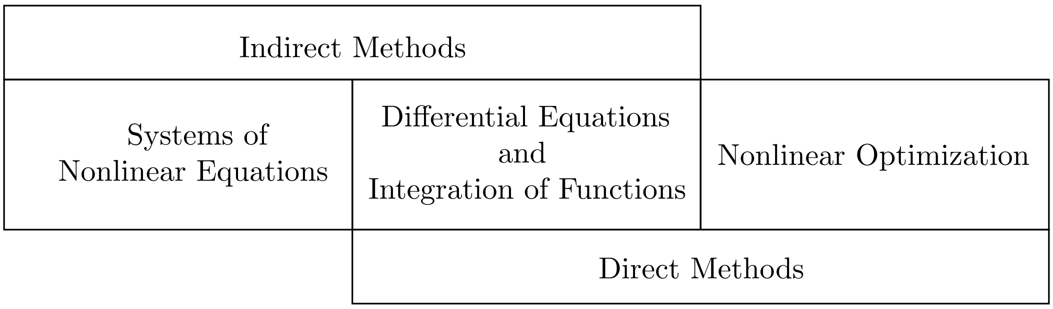
1. The function Φ is continuously differentiable with respect to x and t;
2. The function $\mathcal{L}$ is continuously differentiable with respect to x, y, u, and t;
3. The dynamic function f is continuously differentiable with respect to x, y, u, and t;
4. The function x is continuously differentiable with respect to t, and the function u is piecewise continuous with respect to t.

Then, the solution of such an OCP lies in numerical methods.[1]

3.3.2 Numerical Methods

Numerical methods for solving optimal control problems are divided into two major classes: indirect methods and direct methods. In an *indirect method* the original optimal control problem is turned into the problem of solving a system of nonlinear equations. In a *direct method*, the control function is parametrized, leading to a *nonlinear optimization problem* or *nonlinear programming problem* (NLP). The NLP is then solved using well known optimization techniques. A schematic with the breakdown of the components used by each class of optimal control methods is shown in Figure 14.

[1] It is important to differ g from h_eq: while g is an algebraic equality equation containing algebraic variables and state variables, h_{eq} do not contain algebraic variables, depending only on state variables.



Indirect Methods

Systems of Nonlinear Equations

Differential Equations and Integration of Functions

Nonlinear Optimization

Direct Methods

Figure 14 – The Three Major Components of Optimal Control and the Class of Methods

Direct Methods

The idea behind Direct Methods is “first discretize, then optimize”. The solution is direct in the sense that the optimization process iteratively produces a solution that minimizes an objective function, rather than a solution that satisfies the optimality conditions. Regarding the control variables, they are approximated by piecewise (constant, polynomial, . . .) functions. In some cases, the state variables may also be parametrized. Hence, the OCP is recast as a *NLP* [26].

The most common algorithms employed in the application of a direct method are *Direct Shooting Methods* (single and multiple) which result in small NLP problems, and the *Collocation Method* which is the most accurate, however, resulting in a very large NLP.

The main advantage of the direct methods is that they are very popular and easy to understand and apply (no calculus of variations needed). They are robust and a range of NLPs are available. The principal disadvantage is that direct methods produce only suboptimal or approximate solutions, unless the problem is convex or global optimization algorithms can be afforded.

Indirect Methods

Indirect Methods instead follow the idea “first optimize, then discretize”. These set of methods use calculus of variations to determine the first-order optimality conditions of the OCP. Then, the optimality conditions are used (typically the Pontryagin’s minimum principle – PMP) to find the optimal control variables, which reduces the OCP to a BVP. The BVP solution, in turn, yields candidate optimal trajectories (from with the one with the lowest cost is chosen). Because an indirect method requires solving a multiple-point boundary-value problem, the original optimal control problem is turned into the problem of solving a system of nonlinear equations [26].

The main advantage of the direct methods is that, different from Direct Methods, they do less assumptions regarding the control variables. The disadvantages lies on the fact that path constraint or inequalities makes it difficult to apply the PMP, this leads to state dependent switches, then, the resulting BVPs are difficult to solve.

In order to decrease the solution difficulty, the Hamiltonian is calculated from the Lagrangian function, because it summarizes the optimization problem (representing constraints on the objective function). By eliminating inequality constraints from the set of constraints, the resulting problem becomes easier to solve. Therefore, optimality conditions for optimization problems are derived from the Lagrangian function.

However, the analysis of the computed solution is mandatory to verify if it is truly a minimum. This can be accomplished by inspecting the problem (convexity, second variation and so on).

3.3.3 Sensitivity Analysis

Sensitivity Analysis are methods to obtain the partial derivatives of a function that depends on the solution of an ODE/DAE system. It provides a variation with respect to the control variable, without inducing errors. There are two methods for Sensitivity Analysis: *Forward Sensitivity* and *Adjoint (Backward) Sensitivity.*

Forward Sensitivity

The forward sensitivity calculation is a method for obtaining the derivatives based on a numerical simulation. The *forward* denomination comes from the fact that the derivatives are calculated in the positive direction of the time axis.

The method can be stated for a generic Initial Value Problem (IVP):

$$\dot{x} = f(x, y, u, t), \tag{3.27a}$$

$$0 = g(x, y, u, t), \tag{3.27b}$$

$$x(t_0) = x_0 \tag{3.27c}$$

The Forward Sensitivity derivatives are calculated using the dynamic function f, algebraic function g, boundary conditions $x(t_0)$, and the function of interest $\Phi(x(t_f), y(t_f), t_f)$.

First, the control variable $u \in \mathbb{R}^{N_u}$ is parametrized with respect to time, such as $u(t) = u^k, \forall t \in [t_k, t_{k+1})$. A new vector variable $p \in \mathbb{R}^{N_p}$ is now defined as follows:

$$p = \begin{bmatrix} u_1^1 \\ \vdots \\ u_1^{N_c} \\ u_2^1 \\ \vdots \\ u_2^{N_c} \\ \vdots \\ u_{N_u}^{N_c} \end{bmatrix}_{N_p \times 1} \tag{3.28}$$

which N_c is the number of samplings of u in time, such that $N_p = N_c \times N_u$.

For the sake of readability, the dependency of x, y, and p are omitted and the following matrix variables are defined:

$$R(t) = \frac{dx}{dp} = \begin{bmatrix} \frac{dx^1}{dp^1} & \cdots & \frac{dx^1}{dp^{N_p}} \\ \vdots & \ddots & \vdots \\ \frac{dx^{N_x}}{dp^1} & \cdots & \frac{dx^{N_x}}{dp^{N_p}} \end{bmatrix} \tag{3.29a}$$

$$S(t) = \frac{dy}{dp} = \begin{bmatrix} \frac{dy^1}{dp^1} & \cdots & \frac{dy^1}{dp^{N_p}} \\ \vdots & \ddots & \vdots \\ \frac{dy^{N_y}}{dp^1} & \cdots & \frac{dy^{N_y}}{dp^{N_p}} \end{bmatrix} \tag{3.29b}$$

The Sensitivity Analysis aims to obtain the partial derivative of the function of interest $\Phi(x(tf), y(tf), t_f)$ with respect to p: $\frac{\partial \Phi}{\partial p}$. This can be obtained by solving the following system equations:

$$\frac{df}{dp} = \frac{\partial f}{\partial x} R(t) + \frac{\partial f}{\partial y} S(t) + \frac{\partial f}{\partial p} \tag{3.30a}$$

$$\frac{dg}{dp} = \frac{\partial g}{\partial x} R(t) + \frac{\partial g}{\partial y} S(t) + \frac{\partial g}{\partial p} = 0 \tag{3.30b}$$

$$\frac{dx}{dp}(t_0) = \frac{dx_0}{dp} \tag{3.30c}$$

$$\frac{d\Phi}{dp}(t_f) = \frac{\partial \Phi}{\partial x} R(t_f) + \frac{\partial \Phi}{\partial y} S(t_f) + \frac{\partial \Phi}{\partial p}(t_f) \tag{3.30d}$$

The forward sensitivity has an advantage when the function of interest Φ has a large number of rows. However for problems where the vector p has high dimension, the calculation of such sensitivity can be costly, since the number of additional variables is $N_p(N_x + N_y)$. For these cases, there is a more efficient approach called the adjoint sensitivity.

Adjoint Sensitivity

Differently from the forward sensitivity, the cost for calculating the adjoint sensitivity does not increase with the number of parameters, however it increases with the number of rows in the interest function Φ. This property makes the adjoint sensitivity more suitable for direct methods for optimal control.

Backward Sensitivity is a more elaborated method, which requires a forward and a backward simulation. A detailed presentation of Backward Sensitivity can be found in the literature [27], being based on the same principle of Forward Sensitivity.

3.4 Examples

3.4.1 Direct Multiple Shooting Method

The general idea of the multiple shooting method is to break the simulation horizon into multiple smaller simulations ("first discretize, then optimize"), each being one shooting interval, and, as the optimization algorithm iterates, the state at the end of one shooting interval converges to the initial state of the following shooting interval.

The multiple shooting makes use of a function F that performs the simulation (shooting) of a given system for a given time interval. Given the state at the beginning of the simulation horizon (x) and the control vector (u) as input arguments, the function F returns the state and the algebraic variables at the end of the simulation. For some initial state x^k and the control vector u^k, the values of x^{k+1} and y^{k+1} are obtained by:

$$\begin{bmatrix} x^{k+1} \\ y^{k+1} \end{bmatrix} = F(x^k, u^k), \quad k \in 0, \dots, N_c - 1 \tag{3.31}$$

where N_c are finite elements extracted from the prediction horizon time interval $[t_0, t_f]$.

The control variable $u(t)$ is parametrized in time, $u(t) = u^k, \forall t \in [t_k, t_{k+1})$.

Since F is a simulated function, both Jacobian and Hessian are not easy to obtain. These could be obtained through an approximation (by perturbing the control vector), however, this would induce noise, poor convergence and numerical instabilities. Therefore, *Sensitivity Analysis* plays an important role in proving both functions (Jacobian and Hessian).

As an example, consider the following OCP:

$$\begin{aligned}
\mathcal{P}_{OCP}: \ \min_u \quad & J(x, y, u) = \Phi(x(t_f), t_f) + x_c(t_f) && \text{(3.32a)} \\
\text{s.t.}: \quad & && \text{(3.32b)} \\
& \dot{x} = f(x, y, u, t) && \text{(3.32c)} \\
& \dot{x}_c = L(x, y, u, t) && \text{(3.32d)} \\
& 0 = g(x, y, u, t) && \text{(3.32e)} \\
& h_{\text{eq}}(x, y, u, t) = 0 && \text{(3.32f)} \\
& h_{\text{ineq}}(x, y, u, t) \leq 0 && \text{(3.32g)} \\
& x(t_0) = x_0 && \text{(3.32h)} \\
& x_{\text{L}} \leq x \leq x_{\text{U}} && \text{(3.32i)} \\
& y_{\text{L}} \leq y \leq y_{\text{U}} && \text{(3.32j)} \\
& u_{\text{L}} \leq u \leq u_{\text{U}} && \text{(3.32k)}
\end{aligned}$$

Notice that (3.32) is related to (3.26) except for the dynamic cost function in (3.32d), which is placed as a constraint. To solve the OCP through *Direct Multiple Shooting*, both dynamic and algebraic equations have to be replaced by the simulated function F, which at the same time provides a discretization for the problem (with respect to the state, output and control vector variables). This transformation is denoted in (3.4.1).

$$\mathcal{P}_{NLP}: \min_{u} \; J(x,y,u) = \Phi(x(t_f), t_f) \tag{3.33a}$$

$$\text{s.t.}: \begin{cases} \begin{bmatrix} x^{k+1} \\ y^{k+1} \end{bmatrix} - F(x^k, u^k) = 0, \quad k \in 0, \dots, N_c - 1 \\ h_{\text{eq}}(x^k, y^k, u^k, t) = 0, \quad k \in 1, \dots, N_c \\ h_{\text{ineq}}(x^k, y^k, u^k, t) \leq 0, \quad k \in 1, \dots, N_c \\ x(t_0) = x^0 \\ x_{\text{L}} \leq x^k \leq x_{\text{U}} \\ y_{\text{L}} \leq y^k \leq y_{\text{U}} \\ u_{\text{L}} \leq u^k \leq u_{\text{U}} \end{cases} \tag{3.33b}$$

It can be seen that the original OCP in (3.32), which contained dynamic equations, has been transformed into a NLP , with can be solved with nonlinear optimization.

3.4.2 Indirect Methods

Consider the following optimal control problem:

$$\mathcal{P}_{OCP}: \min_{u, t_f} \; x(t_f)^2 + \int_{t_0}^{t_f} x^2 + u^2 dt \tag{3.34a}$$

$$\text{s.t.}: \begin{cases} \dot{x} = -y + u \\ y - x = 0 \\ x(t_0) = 1 \\ -1 \leq u \leq 1 \end{cases} \tag{3.34b}$$

The Hamiltonian can be written as:

$$H(x,y,u,t) = x^2 + u^2 + \lambda[-y + u] + \nu[y - x] \tag{3.35}$$

Then, the optimality conditions are:

$$-\lambda^* = 2x^* - \nu^* \tag{3.36a}$$

$$-\lambda^* + \nu^* = 0 \tag{3.36b}$$

$$u^* = \min_{u_\mathrm{L} \leq u \leq u_\mathrm{U}} \left[x^{*2} + u^{*2} + \lambda[-y^* + u] + \nu^*[y^* - x^*]\right] \tag{3.36c}$$

$$\dot{x}^* = -y^* + u^* \tag{3.36d}$$

$$y^* - x^* = 0 \tag{3.36e}$$

with the boundary conditions:

$$x^*(t_0) = x_0 \tag{3.37a}$$

$$\lambda^*(t_f) = 2x(t_f) \tag{3.37b}$$

Notice that H is convex with respect to u, therefore, the optimality conditions result by solving $\frac{\partial H}{\partial u} = 0$:

$$\frac{\partial H}{\partial u} = 2u^* + \lambda^* = 0 \Rightarrow u^* = \frac{\lambda^*}{2} \tag{3.38}$$

then, projecting the solution onto the feasible set, results in

$$u^*(t) = \min\left\{u_\mathrm{U}, \max\left\{u_\mathrm{L}, -\frac{\lambda^*}{2}\right\}\right\} \Rightarrow \min\left\{1, \max\left\{-1, -\frac{\lambda^*}{2}\right\}\right\} \tag{3.39}$$

3.5 Summary

This chapter offered a short overview of Optimal Control, presented algorithms that solve Optimal Control Problems, such as Newton's Method with Equality Constraints and the Interior Point Method. The chapter briefly explained the theory of dynamic systems and showed the model structure which is abstracted in the form of simulation, casting an OCP as a nonlinear programming problem.

4 Optimal Control Applied to Freeway Speed Control

This chapter proposes a semi-continuous model for macroscopic traffic, relates it to existing models in the literature and validates it through the fundamental diagrams.

Further, an OCP is proposed to manipulates the speed limits of a freeway, in order to avoid harmful effects of traffic in face of disturbances such as accidents and lane blocking.

4.1 Modeling

Traffic models lie in the differentiation of two variables: space and time. However, the computation of an optimal control law to a completely continuous model is a complex task. Therefore, the model must have derivative in only one variable, consequently, the other variable has to be discretized. In this work, the spatial variable x is discretized while time t is kept as a continuous variable. Spatial variance is easier to be discretized (the CTM model proposes a well defined formulation to be followed) and time is commonly continuous in dynamic systems.

Therefore, a semi-continuous model is proposed, where the spatial modeling is similar to the CTM model and the time continuous modeling derives from second-order models, such as Payne's model [28].

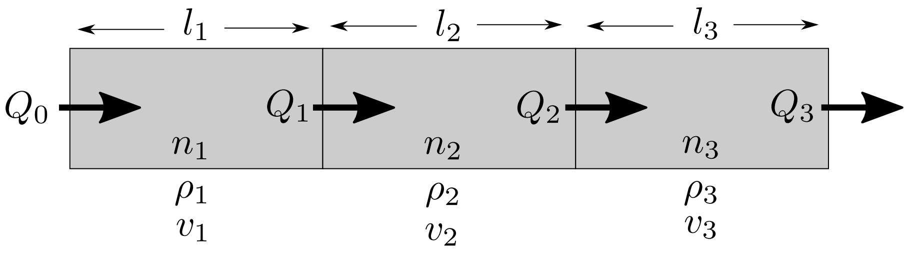


Figure 15 – Cell Transmission Model

Like in the CTM model, the freeway is divided into cells, whereby traffic variables (velocity, concentration, $\cdots$) vary from one cell to the other, as illustrated in Figure 15. Each cell i contains its own state variables, that vary continuously in time.

- Q_i: flow - vehicles leaving cell i and approaching cell $i+1$;
- ρ_i: concentration - number of vehicles in cell i;
- v_i: speed at cell i;

- l_i: length of cell i.

Therefore, a semi-continuous traffic model can be defined to a single cell i. Two important equations define this model: the conservation equation and acceleration equation.

4.1.1 Conservation Equation

The density in cell i varies if the inflow and outflow of vehicles are unbalanced. Density increases if the inflow is greater than the outflow, and decreases otherwise:

$$\dot{\rho}_i(t) = \frac{Q_{i-1}(t) - Q_i(t)}{l_i} \tag{4.1}$$

where $Q_i(t)$ is the flow of vehicles departing from cell i. The flow departing from cell $i-1$, $Q_{i-1}(t)$, becomes the inflow in cell i.

Notice that the right-hand side of equation (4.1) is an approximation of the original conservation equation (2.23): $\dot{\rho}_t + \dot{q}_x = 0$.

Since $Q = \rho v$, equation (4.1) can be written as:

$$\dot{\rho}_i(t) = \frac{\rho_{i-1}(t)v_{i-1}(t) - \rho_i(t)v_i(t)}{l_i} \tag{4.2}$$

in which the only dependent variable is time. This models the continuity of vehicles along the road.

4.1.2 Acceleration Equation

The acceleration equation models the vehicles' dynamics and the influence of the road condition towards it. From the second-order models, acceleration can be modeled as a combination of relaxation (a^{rel}), traffic pressure (a^{pre}), and anticipation (a^{ant}) terms [29], which leads to the following equation:

$$a_i = a_i^{\text{rel}} + a_i^{\text{pre}} + a_i^{\text{ant}}. \tag{4.3}$$

The modeling of acceleration lies on the modeling of each term, that will be explained individually:

Relaxation and Anticipation Terms

The speed depends not only on the local density ρ_i, but on the density ρ_{i+1} at an anticipated location ahead of the actual cell. So, the relaxation and anticipation terms can be expressed as a single equation:

$$a_i^{\text{rel}} + a_i^{\text{ant}} = \frac{V_i^e(\rho_i, \rho_{i+1}) - v_i}{\tau} \tag{4.4}$$

in which τ is the adaptation time (where 3τ is the time a vehicle takes to reach 95% of the target speed $V_i^e(\rho)$).

The speed $V_i^e(\rho_i)$ is the target speed, that is the speed which is comfortable for driving in cell i with prevailing traffic density:

$$V_i^e(\rho_i, \rho_{i+1}) = u_i V_i^{\text{free}} \sqrt{\left(1 - \frac{\rho_i}{\rho_i^{\max}}\right)^2 \left(1 - \frac{\rho_{i+1}}{\rho_{i+1}^{\max}}\right)^2} \tag{4.5}$$

where V_i^{free} is the speed limit in cell i. The control variable u_i is a multiplier to V_i^{free}. Since the *free speed* argument is a parameter it doesn't change over time and is not manipulated by the control law. Therefore, the control variable $u_i \in [0, 1]$ multiplies this *free speed* in order to perform a variable speed limit.

Notice that the closer the density in cell i is to the maximum density, the lower the driving speed will be. The same happens if the density in the following cell $i + 1$ gets near its maximum. This is important to notice, because is related to the CTM model, where the storage capacity available at the following cell $i + 1$ served as an upper bound to the outflow at ell i. In this case, if density ρ_{i+1} is close to storage capacity (maximum density) $\rho_{i+1}^{\max}$, then:

$$\lim_{\rho \to \rho^{\max}} 1 - \frac{\rho}{\rho^{\max}} = 0 \tag{4.6}$$

the **Relaxation and Anticipation** term will be zero. This is replicated when density in the current cell i gets close to maximum $\rho_i^{\max}$.

The square potential serves both to intensify the speed decrement while density increases and to maintain speed as positive when density overgrows the maximum boundaries (what happens in a short instance of time, since pressure term also contributes to acceleration). The square root maintain a feasible value for V_i^{free} when density is low. Otherwise, without square root, V_i^{free} would be too large for regular values of speed found in highways.

Pressure Term

The pressure term describes a response of vehicles on density gradients. The intuition behind the Pressure Term is derived from gas-kinetic, where the speed either increases or decreases when a particle passes from an environment with greater/lower pressure, to a greater/lower pressure. This difference of pressure (difference of density) causes the local ensemble of vehicles to change speed.

This term is also important because it considers the reality of drivers that increase their speed when a faster vehicle is behind, or when they leave the faster vehicle to pass.

This behavior is modeled by the pressure term:

$$a_i^{\text{pre}} = \beta \frac{\rho_{i-1} - \rho_i}{l_i} \tag{4.7}$$

where β is a weight factor that accounts for the influence of vehicles behind on the driver's acceleration. The greater the density in the cell $i-1$ is, the greater will be the pressure term in the cell i, leading the speed in the cell i to increase.

Consider the case where the speed in cell $i-1$ is greater than speed in cell i. Without the pressure term, cell $i-1$ would have a grater inflow of vehicles in cell i, increasing the density in cell i. The greater the density, the smaller the speed. Therefore, a cascade effect would take action, causing the density to increase more and more until the maximum is reached, and the speed decreases until zero.

The fact is that, traffic speed is not only dependent on the current cell's density (relaxation term), but its vehicles are influenced by the previous (pressure term) and future cells (anticipation term) as well.

4.1.3 Scenario Modeling

To simulate traffic scenarios in which there are accidents or lane blocking, a change of model parameters must be applied to the parameters in a given cell i, such as maximum density $\rho_{\max}$ and desired speed V_i^u.

When a lane is occupied by vehicles involved in an accident, it becomes inaccessible by vehicles in movement (coming from previous cells). If the accident is blocking one lane of a two-lane freeway in the segment l_i, by an certain time interval T_{accident}, then at this segment of road the maximum density is decreased to 50 percent of the original maximum density:

$$\widehat{\rho}_i^{\max}(t) = \begin{cases} \frac{\rho_i^{\max}}{2} & , \quad \text{if } t \in T_{\text{accident}} \\ \rho_i^{\max} & , \quad \text{otherwise} \end{cases} \tag{4.8}$$

However, an accident can lead to "curiosity", and it does in many occasions. So, not only density is decreased in such scenarios, but also the *free speed* in the accident location is decreased:

$$\widehat{V}_i^{\text{free}}(t) = \begin{cases} \frac{V_i^{\text{free}}}{2} & , \quad \text{if } t \in T_{\text{accident}} \\ V_i^{\text{free}} & , \quad \text{otherwise} \end{cases} \tag{4.9}$$

These two changes in the parameters can simulate the behavior of an accident or lane blocking, which render an interesting scenario to apply control techniques for speed control.

4.1.4 Model Comparison

The model validation proccess requires both a real scenario and an amount of data. However, since no specific scenario has been chosen and no amount of data has been collected, validation is let to further works. Even so, to prove the model replicates commom behaviors seen in other models and to increase the reliability of this proposed model, it will be compared with well accepted models from the literature. Also, reliability increases when the fundamental diagrams exhibit a behavior similar to the ones considered in Chapter 2.

The proposed semi-continuous model will be compared with the CTM. Both models were calibrated according to the following parameters:

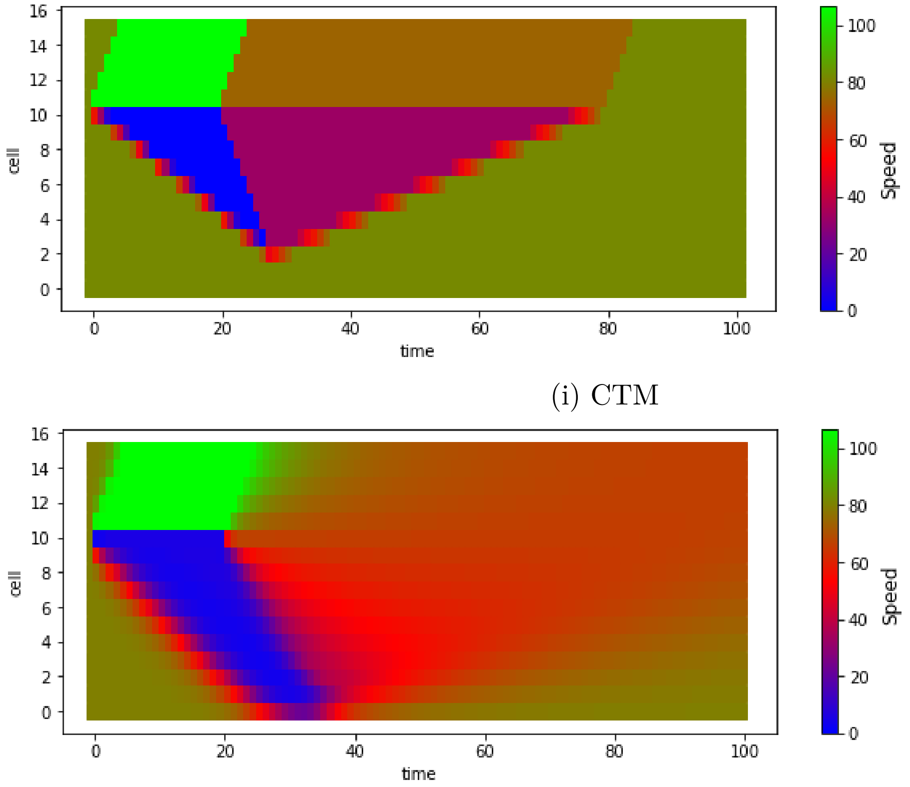
- Vehicle Speed $v = 80$ km/h;
- Freeway Flow $q = 2400$ veh/h;
- Cell length $\Delta x = 167$ m;
- Density $\rho = 4$ veh/cell;
- Maximum Density $\rho_{\max} = 15$ veh/cell;
- Maximum (in/out) flow between cells $Q_{\max} = 5$ veh/ Δt;
- Freeway Speed $V^{\text{free}} = 148.75$ km/h;
- Sampling Time $\Delta t = 6$ s.

The following scenario is considered:

- Number of cells $N = 16$;
- Cell $i = 10$ has maximum inflow of vehicles decreased to 1 vehicle a time, until simulation time $t = 20$;
- After simulation time $t = 20$, cell $i = 10$ has maximum inflow of vehicles restored to $Q_{\max} = 5$.

By simulating both the CTM and the proposed models, results were obtain which are shown in Figure 16.

The results indicate a similarity between the two models. They have similar traffic jam and traffic congestion behavior, similarly increasing speed in the cells after the accident, and a similar speed propagation over time (slope of colored areas). It can also be observed that the *Proposed Model* has a continuous distribution of speed along the time axis, while similar discrete distribution along the cell axis.



(i) CTM

(ii) Proposed Model

Figure 16 – Result of model comparison simulation

4.2 Model Fundamental Diagrams

To get the fundamental diagrams, a realistic scenario was considered (parameters are shown in Chapter 5). The model was simulated in different conditions. For each density (from minimum to maximum) the model parameters were held fixed until speed converged.

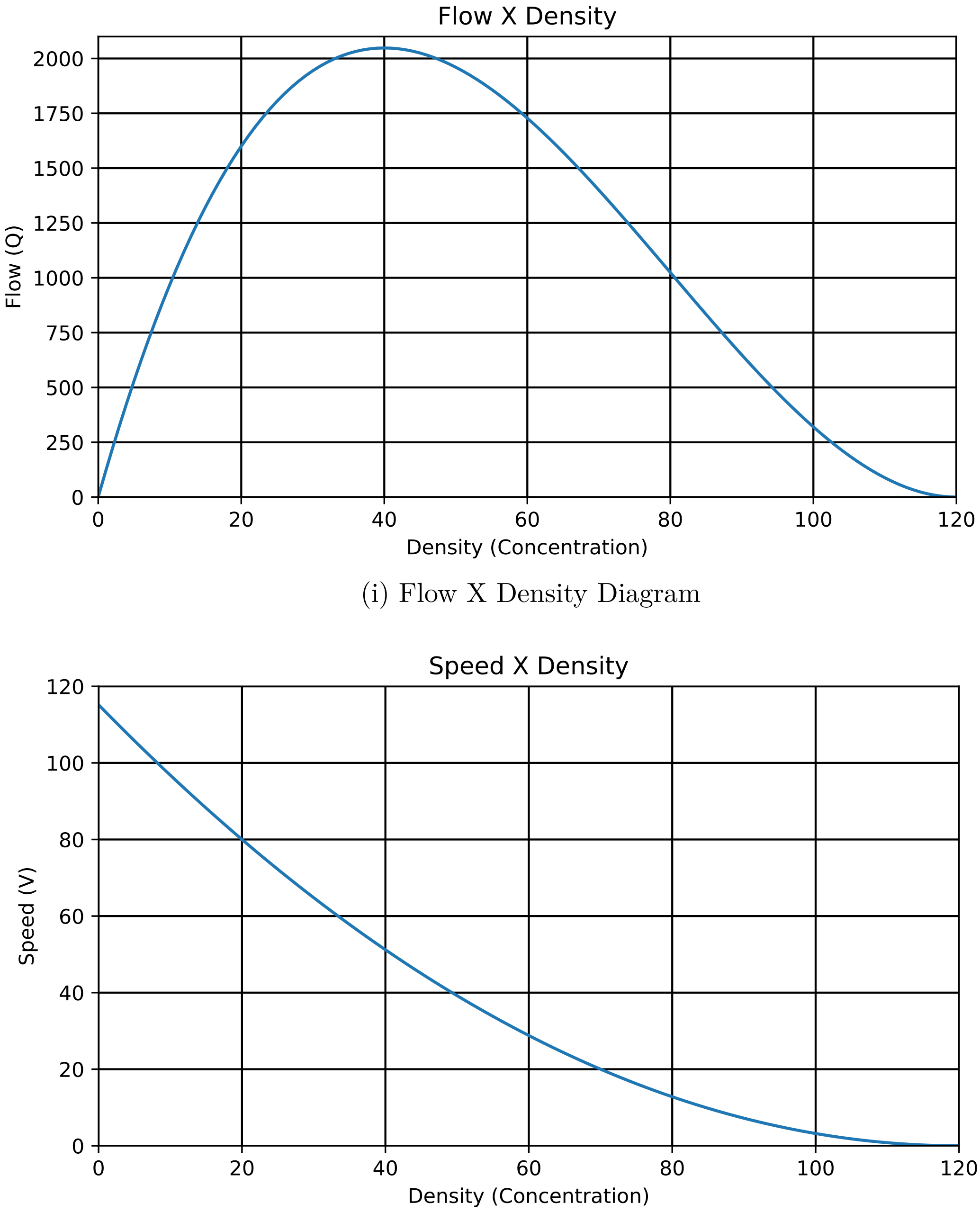
The behavior depicted in Figure 17 area as expected: both fundamental diagrams are very similar when compared to Figure 6.

4.3 Optimal Control Problem

As considered, an OCP is formulated with an objective function and a set of constraints which define both the model (through dynamic equations) and the variable bounds (limits imposed by physical capacity).

4.3.1 Objective Function

When an accident occurs, traffic may jam or become oscillatory (stop-and-go waves). This decreases speed for vehicles and delays trip progress.



(i) Flow X Density Diagram

(ii) Speed X Density Diagram

Figure 17 – Model resulting fundamental diagrams

In order to mitigate congestion effects, the following objective function is defined:

$$L = \sum_{i}^{N} -v_i(t) \tag{4.10}$$

where the negative sign implies a maximization of the speed, since the objective function L is conventionally minimized.

If speed is maximized, the control law will ensure to decrease the concentration of vehicles, decreasing the time spent in traffic congestion.

4.3.2 Constraints

To define the constraints that describe the model behavior, the proposed semi-continuous model needs to be expressed by dynamic equations:

$$\dot{\rho}_i(t) = \frac{\rho_{i-1}(t)v_{i-1}(t) - \rho_i(t)v_i(t)}{l_i} = f_i^{\rho}(t) \tag{4.11}$$

$$\dot{v}_i = \frac{u_i V_i^{\text{free}} \sqrt{\left(1 - \frac{\rho_i}{\rho_i^{\max}}\right)^2 \left(1 - \frac{\rho_{i+1}}{\rho_{i+1}^{\max}}\right)^2} - v_i}{\tau} + \frac{\beta}{\rho_i}\frac{\rho_{i-1} - \rho_i}{l_i} = f_i^v(t) \tag{4.12}$$

in which the control variable $u_i \in [0, 1]$ is the multiplier of the speed limit. The control variable changes dynamically over time for each cell. The main objective is to set the speed limit in each cell i over time, in order to mitigate traffic effects that decrease performance and smoothness.

Therefore, the Optimal Control Problem (OCP) can be formulated as follows:

$$\mathcal{P}_{ODE}: \min_{\rho,v,u} \; J = \sum_{i=1}^{N} \int_{t_0}^{t_f} -v_i(t)dt \tag{4.13a}$$

$$\text{s.t.}: \begin{cases} \dot{\rho}_i = f_i^{\rho}(t), \\ \dot{v}_i = f_i^{v}(t), \\ \rho_i(0) = \rho_i^0, \\ v_i(0) = v_i^0, \\ \rho_i \leq \rho_i^{\max}, \\ \rho_i \geq \rho_i^{\min}, \\ v_i \leq v_i^{\max}, \\ v_i \geq v_i^{\min}, \\ u_i < 1, \\ u_i > 0, \qquad \forall i = 1, \ldots, N \end{cases} \tag{4.13b}$$

This defines the problem formulation, where the dynamic equations ensure the model behavior, and the objective function imposes a better solution than a non-controlled (open-loop) problem.

Although simulations take no constraints in the state variables, it is convenient to the optimization problem to place boundaries in those variables, in order to facilitate the control computing, improving performance and optimality. Also, this ensures a non-negative concentration and speed, which cannot happen in real world conditions.

4.4 Summary

This chapter proposed a second-order model for macroscopic traffic scenarios in freeways. It also formulated the OCP using the proposed model, with the constraints and objective function for the optimization of the given traffic scenario.

5 Experiments and Results

In order to approximate simulation to reality, traffic scenarios commonly found in the real-world were considered [30]:

- Regular condition speed $v = 80$ km/h;
- Regular condition density $\rho = 20$ veh/km;
- Freeway Flow (2 lanes) $q = 1600$ veh/h;
- Time Headway $h_t = \frac{1}{q} = 4.8$ s;
- Spatial Headway $h_x = h_t \times v = 100$ m.

The model was calibrated according to the following parameters:

- Cell length $\Delta x = 1000\text{m} = 1$ km;
- Maximum Density $\rho_{\text{max}} = 120$ veh/km;
- Freeway Speed $V^{\text{free}} = 115$ km/h;
- Adaption Time $\tau = 10$ s;
- Total cells $N = 30, \quad i = [0, 1, 2, \ \dots\ , 29]$.

With the following initial conditions:

- Vehicle Speed $v = 80$ km/h;
- Density $\rho = 20$ veh/km.

Although the model is continue in time, the simulation is discretized. Simulation time interval is split into subintervals by considering time steps Δt (a typical value is 15 s); the discretization in space is obtained by dividing the freeway in sections (a typical length is $500 - 1000$ m). For stability reasons, the discretization needs to be such that $\Delta x > v\Delta t$ for all sections [31].

The accident occurs in cell 28, when simulation time $t = 20$ s, and it blocks one lane. When simulation time $t = 1800$ s, cell 28 is cleared up, and traffic in the blocked lane is freed. While one lane from cell 28 was blocked, both parameters Freeway Speed V^{free} and Maximum Density ρ_{max} were decreased to 50%.

The control variable u is defined as a 3-length variable. u_0 defines the multiplier for cells 0 to 27. u_1 defines the multiplier for cell 28 and u_2 is the multiplier for cell 29. Then, there is one control for each cell before the accident, one control for the accident cell and one control for the cell after the accident. Appendix B shows the implementation of this.

5.1 Implementation

To implement the OCP, the *Python* programming language [32] was used as it provides many scientific tools to implement algorithms and perform simulations. Multiple libraries were used for modeling, calculation and results exhibition.

Casadi [33] served as the tool for modeling and simulation over time, which is a symbolic framework for numeric optimization and automatic differentiation. It provides efficient implementation of algorithms for nonlinear numerical optimization. Of particular interest is dynamic optimization, using either a collocation approach, or a shooting-based approach using embedded ODE/DAE-integrators.

To solve the optimization problem (the OCP requires a nonlinear solver), it was used *IPOPT* [34], a very known solver designed to find (local) solutions of nonlinear optimization problems.

To present the results in three dimensions: time, space and one state (speed, concentration or flow) it was used *MatPlotLib* [35], which provides a simplified tool to make 2D plots using color as the third axis.

The abstracted pseudo-code is given in Algorithm 3 and the implementation code in Appendix B

```
Input: initial condition $x_i^0 = [v_i^0, \rho_i^0]$,
       warm start $u_i^0 = [1]$,
       dynamic model $f_i = [f_i^p, f_i^v]$,
       algebraic equations $g_i$,
       set of constraints $h_i = \left[h_i^{\text{eq}}, h_i^{\text{ineq}}\right]$
let $S = [x_i^0, u_i^0]$ be the initial solution;
let $time = 0$;
let $\Delta time = 5$;
let $time_{final} = 3000$;
while  $time < time_{final}$ do
    mount OCP $P_{OCP}$ using $S$, $f_i$, $g_i$, $h_i$;
    compute $u$ by solving the OCP;
    integrate model $f_i$ using $u$ from $time$ until $time + \Delta time$;
    update $S :=$ integration final solution;
    update $time := time + \Delta time$;
end
```

Algorithm 3: Optimal Control Problem

5.2 Simulation Results

To compare the control efficiency both open-loop and closed-loop simulations were implemented. In some works, *congestion* is considered when the freeway speed are less than 65% of free flow speeds [4]. In this particular case, a congested area will be considered when its concentration is larger than 40 veh/km (larger than the concentration for which flows is maximum).

Although only speed is shown is this section, Appendix A shows figures for density and flow.

5.2.1 Open-loop Simulation

The open-loop simulation (without control computation) result is presented in Figure 18.

It can be observed the accident duration in cell 28: from the beginning to 1800 s. However, the total effect of the accident lasts more than the accident itself. Color red represents a decrease in speed of around 50 km/h. Therefore, the accident only clears up the freeway as a whole when time $t = 3800$ s.

A second important effect to be highlighted is the stop-and-go waves. The blue area (representing a velocity equals – or very close – to zero) contains traces in red, which

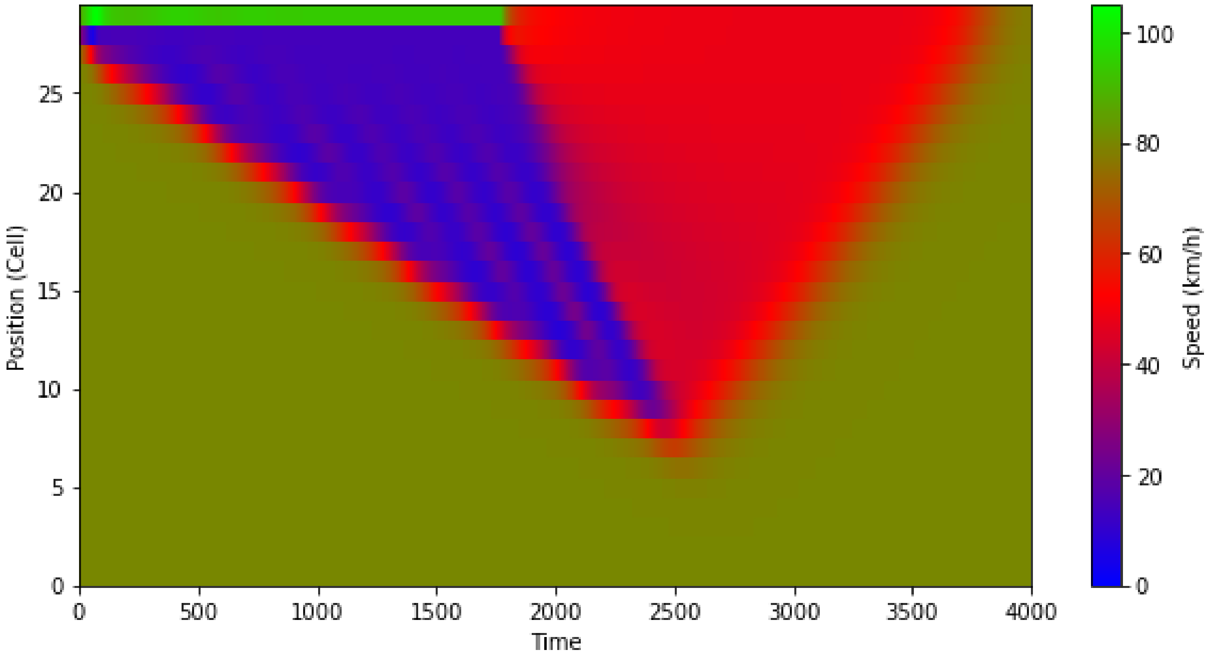


Figure 18 – Non-controlled Resulting speed

means that vehicles have their speed increasing and decreasing from time to time, while they advance in space.

A third important effect is the propagation caused by the accident. The effects are felt all the way to cell 6, which experiences a decrease in speed. Congestion is felt down to cell 9 (when concentration is larger than 40). Taking a time frame (drawing a vertical line) at time $t = 1800$ s, it can be measured the total queue of vehicles in the congestion area: 15 km (from cell 14 until cell 28).Therefore, the total propagation of an accident that blocks one lane from a two lane freeway during 30 minutes (1800 s) extendeds 23 km beyond the location of the accident itself.

Table 1 summarizes some data extracted from this simulation:

Data	Value
Congested Area	11.1km.h
Maximum Speed	105 km/h
Minimum Speed	4 km/h
Average Speed	62 km/h
Maximum Flow	2380 veh/h
Minimum Flow	316 veh/h
Average Flow	1600 veh/h
Maximum Concentration	92 veh/km
Minimum Concentration	4 veh/km
Average Concentration	33 veh/km

Table 1 – Open-loop simulation results

5.2.2 Closed-loop Simulation

The results from the closed-loop simulation (using optimal control computation) are presented in Figure 19.

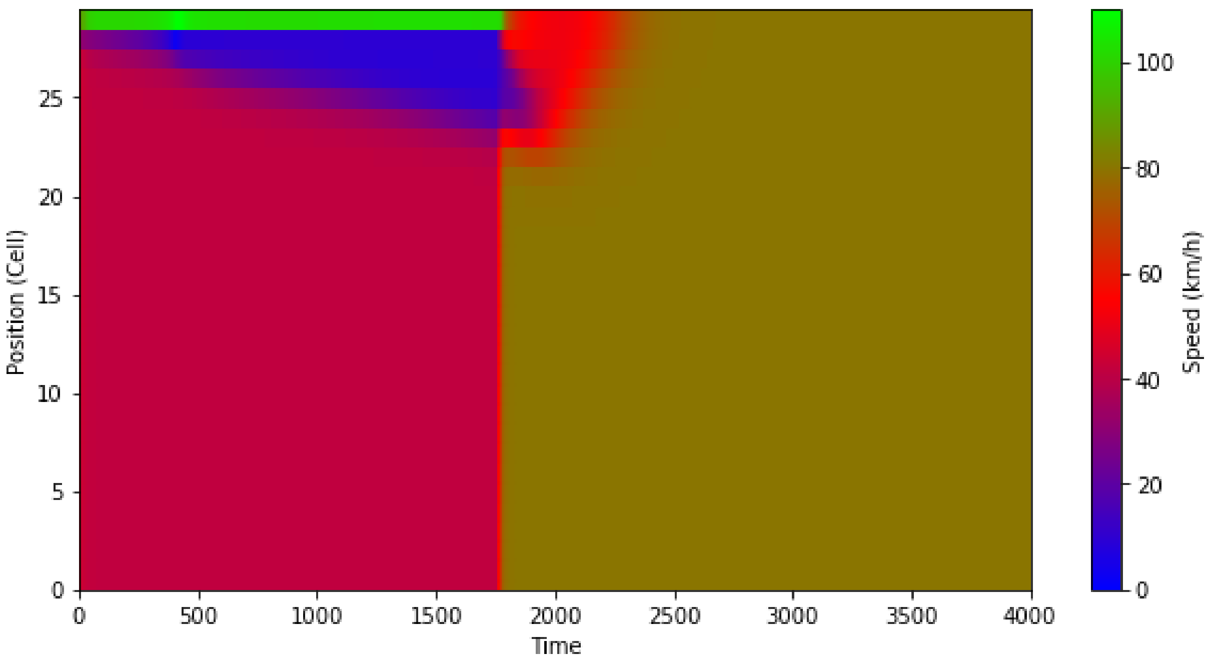


Figure 19 – Controlled Resulting speed

It can be observed that the accident duration in cell 28 is the same: from the beginning to 1800s. However, the total effect of the accident was decreased. Although it lasts more than the accident itself, speeds returned to normal condition at time $t = 2350$ s (70% decreas in relation to the open-loop case).

It can be seen that stop-and-go waves have been suppressed, as no shades of higher speed are found in the blue area.

The propagation caused by the accident was also reduced. The effects are felt only down to cell 22, which has experienced a decrease in speed (70% reduction). Congestion effects do not extend beyond cell 24 (75% reduction). Taking a time frame in $t = 1800$ s, the total queue of vehicles in the congestion area is 5 km (a 70% reduction in relation to the open-loop case).The total propagation of the congestion was reduced from 23 km to 7 km, which represents a 70% of improvement in traffic conditions.

Table 2 summarizes some data extracted from this simulation:

Data	Value
Congested Area	1.4 km.h
Maximum Speed	110 km/h
Minimum Speed	2 km/h
Average Speed	62 km/h
Maximum Flow	2423 veh/h
Minimum Flow	179 veh/h
Average Flow	1260 veh/h
Maximum Concentration	74 veh/km
Minimum Concentration	2 veh/km
Average Concentration	22 veh/km

Table 2 – Closed-loop simulation results

5.2.3 Drivers Standpoint

Drivers do not see the freeway as a whole, as it is seen from a simulation perspective. What really matters is to arrive at the destination faster. In fact, to reduce time spent driving, speed has to be increased.

To compute trips of vehicles in the simulation, two initial data were necessary: the starting point (position in space) and the starting time (position in time). From this information, the speed while a driver advances in position was integrated over time. Three scenarios were considered:

A. Starting at $x = 0$ km at time $t = 0$ s;

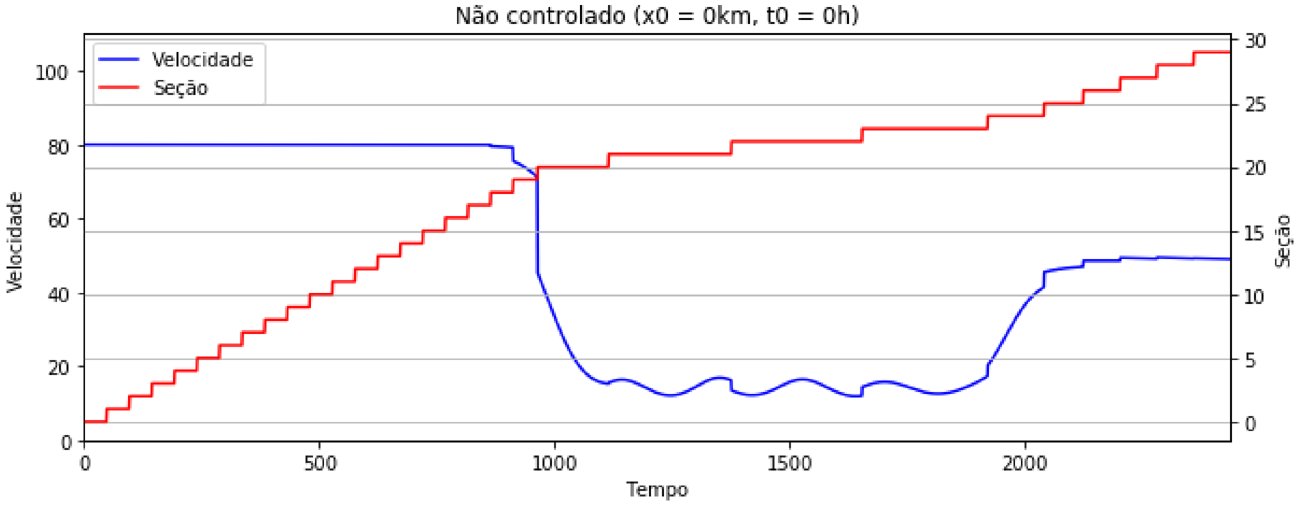
B. Starting at $x = 1$ km at time $t = 30$ minutes;

C. Starting at $x = 15$ km at time $t = 15$ minutes;

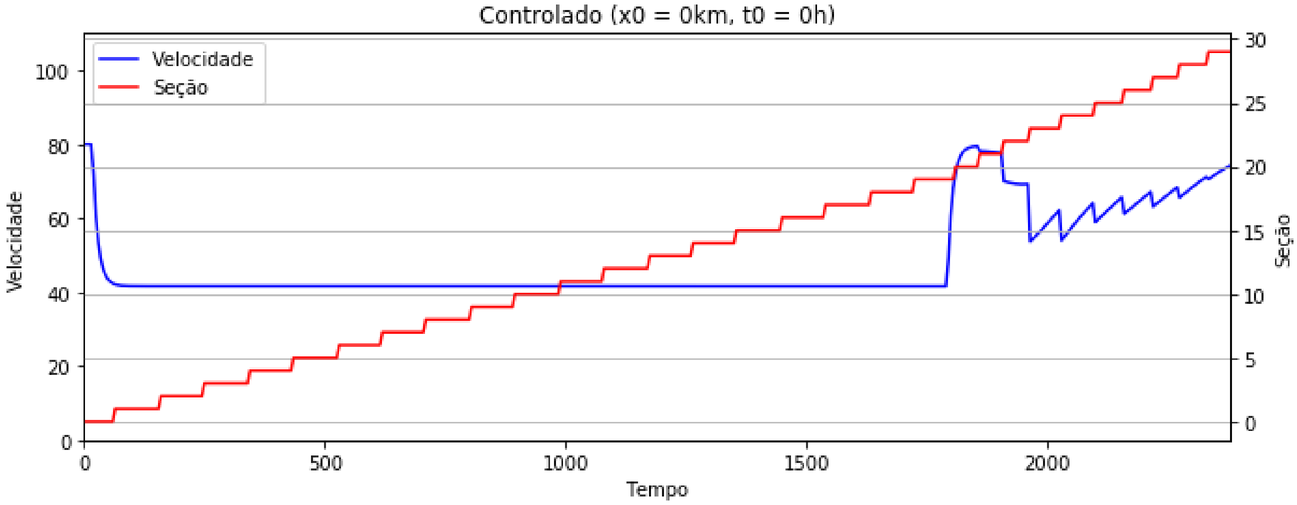
For all scenarios, the destination is to arrive at the end of the last cell. The end of each plot represents the driver arrived at the destination.

The results of the integration for scenarios (**A** – Figure 20), (**B** – Figure 21) and (**C** – Figure 22) are shown for both non-controlled and controlled simulations.

In Figure 20, it can be observed for scenario **A** that drivers in controlled simulation arrive at the end of the last cell when time $t = 2380$ s, slight before than drivers in open-loop simulation when time $t = 2437$ s (2% time reducing). This is due to the fact that in the controlled simulation speed (as seen in Figure 20ii) had decreased initially (when accident is detected and speed limit control begin to act) from 80 km/h to 42 km/h until time $t = 1800$ s (when the blocked lane is freed). This action provided a better performance when it comes to the time drivers arrive at their destination (end of cell 29).



(i) Trip for open-loop simulation.

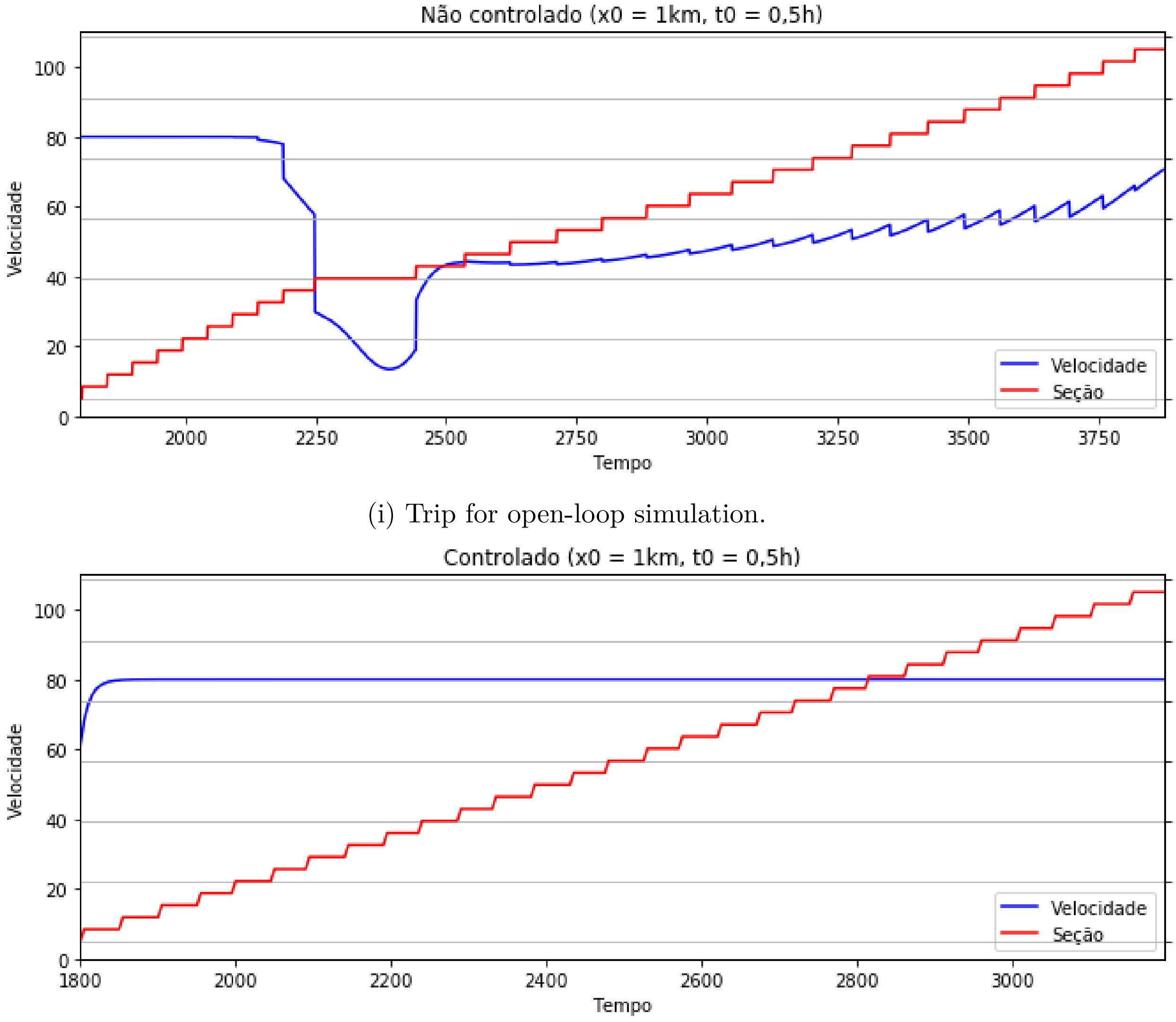


(ii) Trip for controlled simulation.

Figure 20 – Trips for scenario A, starting at 0 km in time 0 h.

Although a certain variation in speed happens in the end of the controlled simulation, it is not characterized as stop-and-go effect, because speed in much larger than zero (between 54 km/h and 75 km/h). However, in Figure 20i stop and go waves are clearly seen in the open-loop simulation between time $t = 1000$ s and time $t = 2000$s when speed varies from 10 km/h to 15 km/h.

For scenario **B**, shown in Figure 21, drivers in controlled simulation arrive when time $t = 3195$ s, much before than drivers in open-loop simulation when time $t = 3874$ s (33% time reducing). This scenario happens after the blocked lane is freed (when time $t = 1800$ s). In Figure 21ii can be seen that drivers are returning their speeds to 80 km/h, and keeping it constant when speed limit is controlled. In the open-loop simulation, Figure 21i shows that drivers take much longer to return to the original speed of 80 km/h. Hence, driver in controlled scenario arrived faster at their destination (end of cell 29).



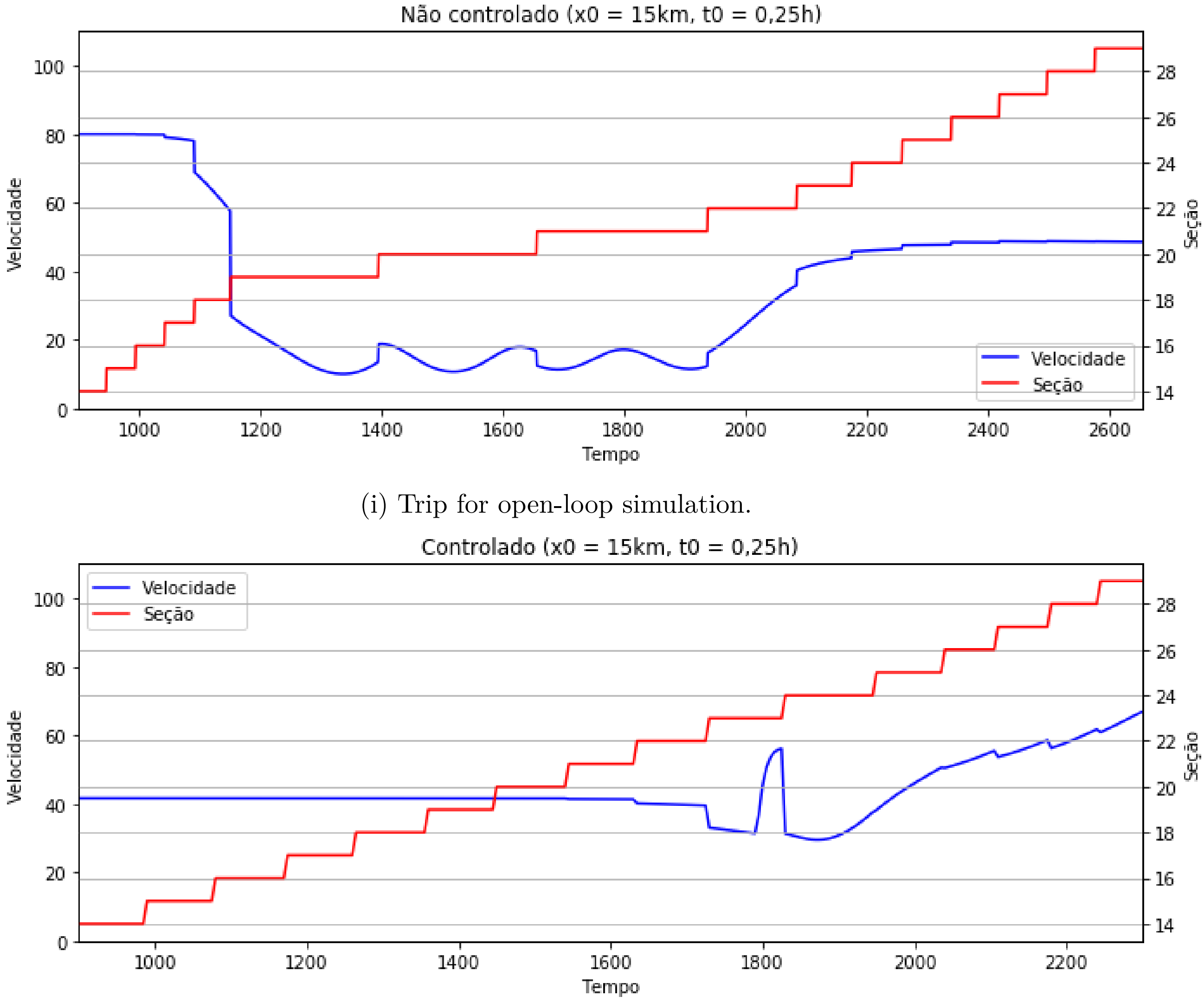
(i) Trip for open-loop simulation.

(ii) Trip for controlled simulation.

Figure 21 – Trips for scenario B, starting at 1 km in time 30 minutes.

In scenario **C**, shown in Figure 22, drivers in controlled simulation arrive when time $t = 2300$ s, before than drivers in open-loop simulation when time $t = 2655$ s (21% time reducing). In the controlled simulation, Figure 22ii, drivers come with a lower speed of 42 km/h, what avoided the stop-and-go effect detected in the open-loop simulation (Figure 22i) between time $t = 1200$ s and time $t = 2000$ s. This lower speed oncoming provided a faster approach to the destination (end of cell 29).

Therefore, in all scenarios considered, the controlled speed simulation guaranteed a better performance from driver standpoint, arriving at the destination in a shorter time than in the open-loop simulation. Thus, minimizing the effect of traffic jam, traffic congestion and stop-and-go wave effect.



(i) Trip for open-loop simulation.

(ii) Trip for controlled simulation.

Figure 22 – Trips for scenario C starting at 15 km in time 15 minutes.

5.3 Summary

This chapter showed the simulations for both controlled and non-controlled speed scenarios. It illustrated by figures the conditions of both simulations through time and space. Also, from a driver standpoint, the chapter provided plots of the speed through time until arriving at the destination for different scenarios and for both simulations. At last, the chapter highlighted the performance improvement assured in freeways by the controlled simulations.

6 Conclusion and Future Works

Sometimes slower means faster.

This is an important conclusion from this work. In traffic congestion, to decrease speed limits (the desired speed) increase the driving speed.

Non-controlled simulations made possible for drivers to choose their speed. As observed, they traveled at the speed of 80km/h, making a big shock wave when they hit the congested area. This shock wave propagated through space and time, causing damaging effects such as delaying trip progress, traffic jam and causing stop-and-go behavior.

However, in controlled simulations, it was observed the control in action by reducing the speed of ascending cells (before the accident), so as to reduce this shock wave in the congested area. This reduced the accident propagation, decreasing the trip delay, traffic jam, and avoiding damaging effects such as stop-and-go behavior. The slower entrance in the congested area guaranteed a faster recovery, taking speed to the regular value much faster.

From the driver standpoint, it could be seen the effects of those enhancements. While with non-controlled speed a specific driver took 2655 seconds to complete his trip, the same driver with controlled speed took 2300 seconds – 21% faster. In fact, in all scenarios considered, drivers with controlled speed could arrive earlier at their destination. Although some drivers were slower in the beginning, their speed was recovered in the advancement of time, decreasing trip time.

By maintaining time as a continuous variable and sampling space into cells, a semi-continuous model was obtained for application Optimal Control Problems (OCP). The OCP, in turn, computed the parameters that guaranteed the better performance to the given conditions in a rolling prediction horizon. The overall enhancement given by the controlled simulation was of 20% trip time reducing for drivers and of 70% freeway congestion reducing.

Of course, other traffic areas could be investigated for the application of optimal control, such as urban areas. Microscopic traffic is found to be a more complex area since it requires more computation, multiple variables for each vehicle in a street. In a scenario with connected streets, where flow is controlled by semaphores, distributed optimal control can be applied in real time to enhance performance. This is left for future works.

This work showed the importance of integrating many subjects in order to achieve an important result. Physics are important to comprehend, understand behaviors in nature and human life. Mathematics plays an important role to model these behaviors for numerical computation, and computer simulation, with the aim of improving traffic performance. Optimization provided the theory and tools to use the explicit model and simulations in order to compute parameters that assure the best solution for a given problem. Computation was the basis to run the simulations, obtain optimal control signals, apply to the simulation models, and obtain results. The study of Control and Automation Engineering made possible to use all these tools to achieve goals such as improving human life quality, making life better for people, while improving comfort, reducing cost and improving productivity.

Bibliography

1 INRIX. Los angeles tops inrix global congestion ranking. 2017. Disponível em: <http://inrix.com/press-releases/los-angeles-tops-inrix-global-congestion-ranking/>. Citado na página 10.

2 INRIX. Inrix 2015 traffic scorecard sets benchmark for u.s. cities as federal government accelerates smart city spending. 2016. Disponível em: <http://inrix.com/press-releases/scorecard-us/>. Citado na página 10.

3 INRIX. Inrix traffic scorecard reports u.s. congestion grew at three times the rate of u.s. gdp. 2016. Disponível em: <http://inrix.com/press-releases/2764/>. Citado na página 10.

4 INRIX. 2016 global traffic scored. 2017. Disponível em: <http://inrix.com/scorecard/>. Citado 2 vezes nas páginas 10 and 57.

5 INRIX. Traffic congestion cost uk motorists more than 30 billion in 2016. 2017. Disponível em: <http://inrix.com/press-releases/traffic-congestion-cost-uk-motorists-more-than-30-billion-in-2016/>. Citado na página 10.

6 CHUNG, Y. Development of an accident duration prediction model on the korean freeway systems. *Accident Analysis and Prevention*, v. 42, n. 1, p. 282 – 289, 2010. ISSN 0001-4575. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0001457509002139>. Citado na página 10.

7 RAJ, N. G. Interview with N.R. Narayana Murthy, Chairman and Chief Mentor, Infosys Technologies. *The Hindu*, 2010. Disponível em: <http://www.thehindu.com/opinion/interview/lsquoWe-want-youngsters-to-take-to-SampT-in-a-big-way/article15511624.ece>. Citado na página 11.

8 MERRIKH-BAYAT, F.; JAMSHIDI, A. Comparing the performance of optimal pid and optimal fractional-order pid controllers applied to the nonlinear boost converter. *arXiv preprint arXiv:1312.7517*, 2013. Citado na página 11.

9 TRIBER, M.; KESTING, A. *Traffic Flow Dynamics - Data, Models and Simulation*. [S.l.]: Springer, 2013. ISBN 978-3-642-32459-8. Citado 3 vezes nas páginas 13, 20, and 24.

10 GERLOUGH, D. L.; HUBER, M. J. *Traffic Flow Theory (Special Report ; No. 165)*. [S.l.]: Transportation Research Board, 1975. ISBN 0309024595. Citado na página 13.

11 HELBING, D. et al. Micro-and macro-simulation of freeway traffic. *Mathematical and computer modelling*, Elsevier, v. 35, n. 5-6, p. 517–547, 2002. Citado na página 19.

12 TREIBER, M.; HELBING, D. Explanation of observed features of self-organization in traffic flow. *arXiv preprint cond-mat/9901239*, 1999. Citado na página 20.

13 BRACKSTONE, M.; MCDONALD, M. Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*, Elsevier, v. 2, n. 4, p. 181–196, 1999. Citado na página 21.

14 DAGANZO, C. F. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, Elsevier, v. 28, n. 4, p. 269–287, 1994. Citado 2 vezes nas páginas 21 and 23.

15 DAGANZO, C. F. The cell transmission model, part ii: network traffic. *Transportation Research Part B: Methodological*, Elsevier, v. 29, n. 2, p. 79–93, 1995. Citado na página 23.

16 CATALIN, D.; DAUPHIN-TANGUY, G.; POPESCU, D. Macroscopic modeling of road traffic by using hydrodynamic flow models. In: IEEE. *Control & Automation (MED), 2012 20th Mediterranean Conference on.* [S.l.], 2012. p. 42–47. Citado na página 23.

17 BOYD, S.; VANDENBERGHE, L. *Convex optimization.* [S.l.]: Cambridge university press, 2004. Citado na página 26.

18 FLOUDAS, C. A. *Nonlinear and mixed-integer optimization: fundamentals and applications.* [S.l.]: Oxford University Press, 1995. Citado na página 26.

19 HINDMARSH, A. C. et al. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw.*, ACM, v. 31, n. 3, p. 363–396, set. 2005. ISSN 0098-3500. Disponível em: <http://dx.doi.org/10.1145/1089014.1089020>. Citado na página 32.

20 KHALIL, H. K. Noninear systems. *Prentice-Hall, New Jersey*, v. 2, n. 5, p. 5–1, 1996. Citado na página 34.

21 CAMACHO, E. F.; ALBA, C. B. *Model predictive control.* [S.l.]: Springer Science & Business Media, 2013. Citado na página 35.

22 LYAPUNOV, A. M. The general problem of the stability of motion. *International Journal of Control*, Taylor & Francis, v. 55, n. 3, p. 531–534, 1992. Citado na página 36.

23 AGUIAR, M. A. An augmented lagrangian method for optimal control of continuous time dae systems. *Federal University of Santa Catarina*, 2016. Citado na página 38.

24 KIRK, D. E. *Optimal control theory: an introduction.* [S.l.]: Courier Corporation, 2012. Citado na página 38.

25 RAO, A. V. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, Univelt, Inc., v. 135, n. 1, p. 497–528, 2009. Citado na página 38.

26 FREGO, M. Numerical methods for optimal control problems with application to autonomous vehicles. University of Trento, 1998. Citado na página 40.

27 CAO, Y.; LI, S.; PETZOLD, L. Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software. *Journal of computational and applied mathematics*, Elsevier, v. 149, n. 1, p. 171–191, 2002. Citado na página 42.

28 MODELS of freeway traffic and control. Citado na página 46.

29 DAGANZO, C. F. Requiem for second-order fluid approximations of traffic flow. *Transportation Research Part B: Methodological*, Elsevier, v. 29, n. 4, p. 277–286, 1995. Citado na página 47.

30 Spack, Mike. *Numbers Every Traffic Engineer Should Know.* Citado na página 55.

31 BELLEMANS, T.; De Schutter, B.; De Moor, B. Models for traffic control. *Journal A*, v. 43, n. 3–4, p. 13–22, 2002. Citado na página 55.

32 ROSSUM, G. van; (EDS), F. D. Python reference manual. 2001. Disponível em: <http://www.python.org>. Citado na página 56.

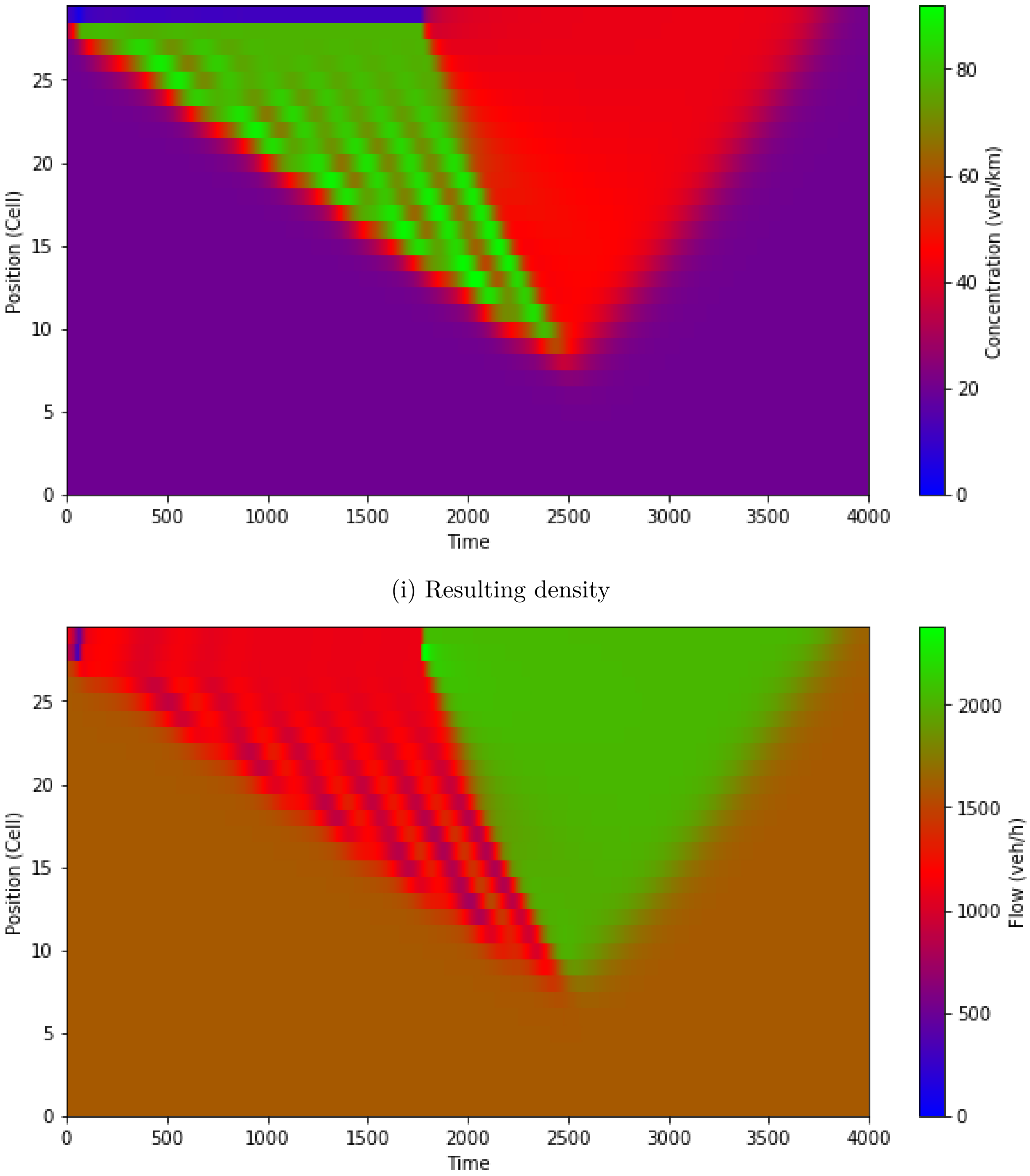
33 ANDERSSON, J. *A General-Purpose Software Framework for Dynamic Optimization.* Tese (PhD thesis) — Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, October 2013. Citado na página 56.

34 WÄCHTER, A.; BIEGLER, L. T. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, v. 106, n. 1, p. 25–57, 2006. Citado na página 56.

35 HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. Citado na página 56.

Appendix

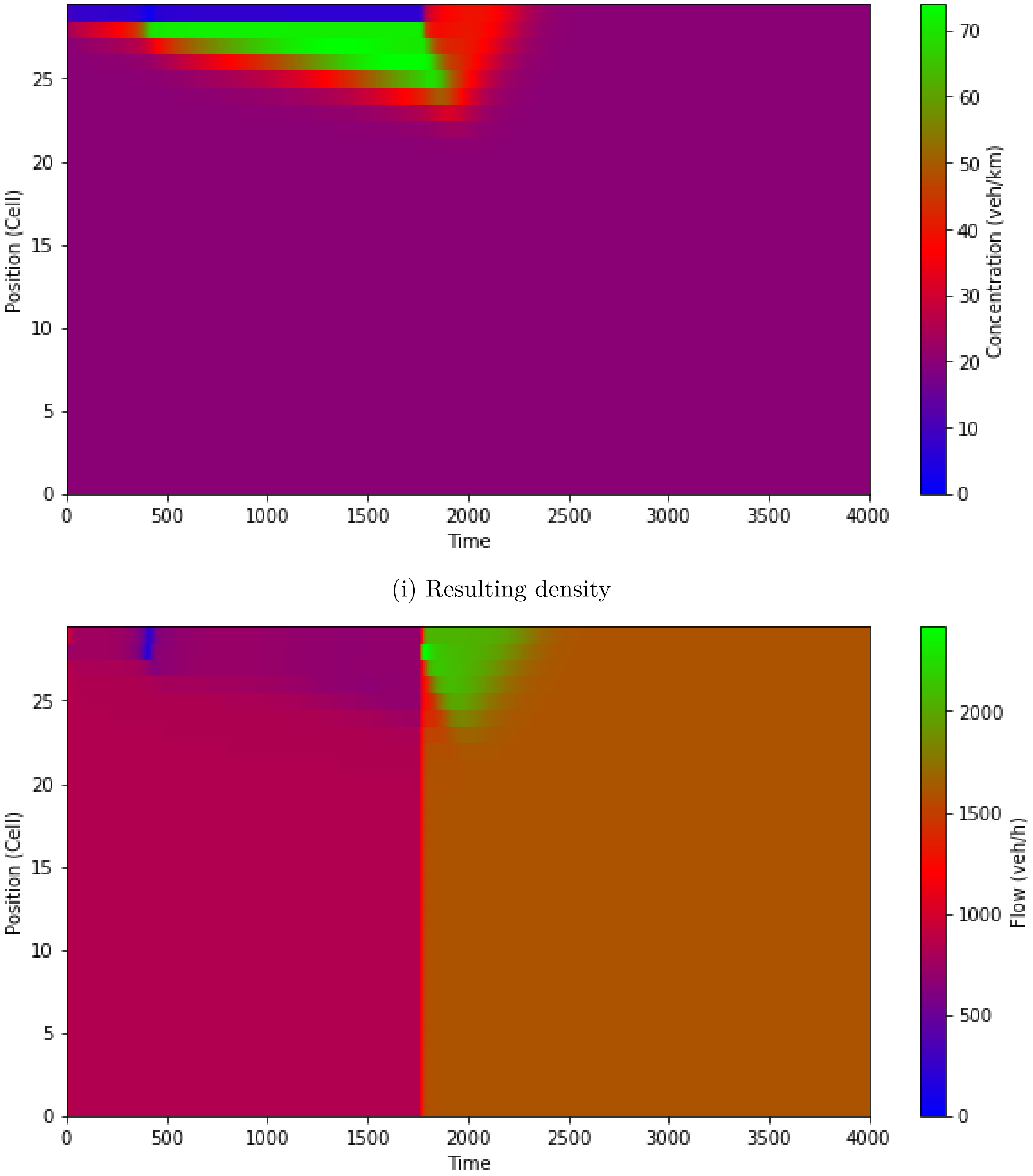
APPENDIX A – Further Simulation Figures



(i) Resulting density

(ii) Resulting flow

Figure 23 – Simulation without control



(i) Resulting density

(ii) Resulting flow

Figure 24 – Simulation with control

APPENDIX B – Implementation Codes

Listing B.1 – Controlled Simulation

```
N = 30  # number of cells
tau = 10  # time response
V_ss = 80  # regular speed
Rho_ss = 20  # regular density
Rho_max_param = 6 * Rho_ss  # maximum density
V_free = 115.2  # free speed
delta_x = 1000  # cell size
t_final = 4000  # simulation final time
delta_t = 5  # simulation step
ACCIDENT_CELL = N-2  # cell which accident occurs
N_CONTROL = 3  # number of controls

model = SystemModel(n_x=2*N, n_p=2*N, n_u=N_CONTROL)
v = model.x_sym[:N]
rho = model.x_sym[N:]
V_0 = model.p_sym[:N]
Rho_max = model.p_sym[N:]
u = model.u_sym

ode = [None]*(2*N)
for i in range(N):
    control_var = u[0]
    if i == 0:
        rho_i_m_1 = rho[i]
        v_i_m_1 = v[i]
    else:
        rho_i_m_1 = rho[i - 1]
        v_i_m_1 = v[i - 1]

    if i == N - 2:
        control_var = u[1]

    if i == N - 1:
        rho_i_p_1 = rho[i]
        rho_max_i_p_1 = Rho_max[i]
        control_var = u[2]
    else:
        rho_i_p_1 = rho[i + 1]
        rho_max_i_p_1 = Rho_max[i + 1]



    v_f_rho = control_var*V_0[i]*(((1-(rho[i]/Rho_max[i]))**2)**0.5)*
               (((1-(rho_i_p_1/rho_max_i_p_1))**2)**0.5)
    pressure_term = (40/(2*tau*rho[i]))*(rho_i_m_1-rho[i])/delta_x
    acceleration = (v_f_rho-v[i])/tau + pressure_term

    delta_Q = (rho_i_m_1*v_i_m_1 - rho[i]*v[i])
    d_rho = (delta_Q/delta_x)

    ode[i] = acceleration
    ode[i+N] = d_rho

model.include_system_equations(ode=ode)

v_ant = [V_ss]*N
rho_ant = [Rho_ss]*N
v_f = []
rho_f = []
sim_p = []
v_rho = []
u_sim = []

L = -sum([v[i] for i in range(N)])
problem = OptimalControlProblem(model,
x_0=vertcat(v_ant, rho_ant), t_f=t_final, L=L)

problem.u_max = [1.0]*N_CONTROL
problem.u_min = [0.0]*N_CONTROL
problem.delta_u_max = [1.0]*N_CONTROL
problem.delta_u_min = [-1.0]*N_CONTROL
problem.x_max = vertcat([V_free]*N, [Rho_max_param]*N)
problem.x_min = vertcat([0]*N, [0]*N)
```

```
solution_method = DirectMethod(problem, finite_elements=20,
                           discretization_scheme='collocation')

last_result = []
for i in range(100):
    last_result += [V_ss]*N
    last_result += [Rho_ss]*N
    last_result += [0]
last_result += [1.0]*N_CONTROL*20

for i in range((t_final/delta_t) + 1):
    CURRENT_TIME = i*delta_t
    print(CURRENT_TIME)
    vel_des = [V_free]*N
    den_max = [Rho_max_param]*N
    if (CURRENT_TIME > 20) and (CURRENT_TIME < 1800):
        vel_des[ACCIDENT_CELL] = V_free*0.5
        den_max[ACCIDENT_CELL] = Rho_max_param*0.5

    result = solution_method.solve(initial_guess=last_result,
    p=vertcat(vel_des, den_max), x_0=vertcat(v_ant, rho_ant))
    last_result = result.raw_decision_variables
    u = result.u_interpolation_data['values'][0][0]
    sim = model.simulate(x_0=vertcat(v_ant, rho_ant),
                       t_f=(i + 1) * delta_t, t_0=i * delta_t,
                       p=vertcat(vel_des, den_max, u))

    v_ant = sim['xf'][:N]
    rho_ant = sim['xf'][N:]

    v_f.append(sim['xf'][:N])
    rho_f.append(sim['xf'][N:])
    sim_p.append(vertcat(vel_des, den_max))
    u_sim.append(u)
```