May 2021

Velocity Optimization in Connected Autonomous Vehicles and its Impact on Surrounding Traffic

Sumedh Sathe
Clemson University, sumedh.sathe1995@gmail.com

Velocity Optimization in Connected Autonomous Vehicles and its Impact on Surrounding Traffic

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

by
Sumedh Prashant Sathe
May 2021

Accepted by:
Dr. Mohammad Naghnaeian, Committee Chair
Dr. Ardalan Vahidi
Dr. Yue Wang

Abstract

Connected Autonomous Vehicles are equipped with the capabilities of autonomous navigation, Vehicle to Vehicle, and Vehicle to Infrastructure communication, which have the potential to improve fuel and/ or energy efficiency. Velocity optimization is a driving technique that aims to follow a velocity profile that minimizes fuel consumption, energy consumption, idling at traffic lights, and overall trip time. Velocity optimization can be implemented in CAVs by utilizing V2I and V2V capabilities, and optimal control techniques. As CAVs become more ubiquitous, they are likely to interact closely with human driven cars. In such a scenario, it is important to find the right trade-off between safety and efficiency, as safety constraints may restrict efficient actions and vice-versa. Vehicle control systems that are heavily biased towards efficiency, may result in conservativeness and rear-ending effects in CAVs, rendering their behavior unpredictable for human drivers, which may result in collisions, compromise safety and obstruct the surrounding traffic. Through this research, we have proposed a velocity optimization strategy that optimizes the velocity profile for fuel consumption, without significantly compromising safety and affecting the traffic flow. A Model Predictive Controller is designed to compute the optimal velocity profile based on fuel consumption and impact to the surrounding traffic. A mathematical control parameter is introduced for deterministic control of impact on traffic flow. An iterative convex optimization approach is adopted for online solution of the optimal control problem. A simulation case study is presented to demonstrate fuel saving capability and reduced impact on the surrounding traffic flow, of the proposed control system.

Dedication

This thesis is dedicated to my parents Prashant S. Sathe and Suhasini P. Sathe, for their never ending love and support throughout my life, and their continued encouragement to pursue the goals I am striving to achieve. I would also like to dedicate my work to my extended family and friends, who supported me through tough times. Lastly, I would like to dedicate this thesis to my advisor, Dr. Mohammad Naghnaeian, for his guidance and support throughout my graduate studies, committee members Dr. Ardalan Vahidi and Dr. Yue Wang for their teachings and help with research.

Acknowledgments

I would like to thank my advisor, Dr. Mohammad Naghnaeian for providing guidance, learning materials for research. I would also like to thank Dr. Ardalan Vahidi for sharing Signal Phase and Timing data from the City of Greenville, SC, for performing simulations.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Aggressive behaviour, characterized by higher levels of acceleration and braking, high speed driving causes an increase in fuel consumption much as 22% in conventional vehicles and 32% in hybrid electric vehicles, according to a study published by Oak Ridge National Laboratory [22]. Traffic congestion further worsens fuel-consumption in vehicles due to excessive idling at traffic lights, and stimulating aggressive driving to make up for the lost time. The Texas A&M Transportation Institute attributes 3.3 billion gallons of extra fuel consumed, 8.8 billion extra hours of travel, $1.79 billion worth of wasted time and fuel in 2017 to congestion [17]. These statistics are expected to rise to 3.6 billion gallons, 10 billion hours and $237 billion respectively in 2025 [17].

Although active traffic management technologies such as Real-Time Adaptive Signal Control [1], could potentially counter the congestion problem to some degree [13], they have been deployed only in limited numbers in a few cities [20]. Furthermore, the capital costs for large scale implementation of these technologies could be as high as $303 million, and operating and maintenance cost as much as $ 425 million annually according to [15]. Hence there is a need to plan the driving speed at the vehicular level to save fuel wasted in idling at traffic lights and aggressive driving.

Connected Automated Vehicle (CAV) technology has the potential to significantly improve fuel economy along with safety and convenience, with the capabilities of autonomous navigation, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication [21]. Velocity Optimization, also known as Optimal Eco-driving is a driving style that aims to follow a velocity trajectory that minimizes fuel consumption and/or energy under time and distance constraints. Eco-driving is realizable in CAVs by utilizing V2V/ V2I abilities and optimal control techniques [21].

The factors affecting the optimal velocity trajectory are traffic flow, traffic lights, road grade, road curvature and stop signs. Considering all of these factors makes velocity optimization a very complex problem that requires modeling these factors and integrating them into the vehicle's dynamic model to determine vehicle response and performance. To optimize the vehicle's trajectory, algorithms that solve optimal control problem based on minimization of fuel consumption, energy consumption and/or overall trip time have been developed. Optimal control theory is the science of calculating control inputs to a dynamic system that satisfy one or more optimality criteria. For velocity optimization in vehicles, the optimality criteria may be minimization of time, fuel consumption, energy and/or emissions. The control inputs to the system may be engine torque or traction force, braking torque or braking force, and gear ratio. The inputs are subject to the vehicle dynamic constraints, limiting constraints and terminal time constraints.

As autonomous vehicles become more ubiquitous, they are likely to interact with human driven vehicles. Hence, it will become extremely important to address safety while driving efficiently. There is a basic trade-off between safety and efficiency, as safety constraints can restrict actions that are efficient, and vice-versa. Autonomous vehicle control systems that are biased towards efficiency are more likely to drive conservatively as compared to the human driven cars that are driving alongside, making them unpredictable for human drivers. This can result confuse or agitate human drivers, increasing the risk of rear-end collisions, seriously compromise safety and obstruct the surrounding traffic flow. Since 2014, 295 autonomous vehicle collisions are reported by Department of Motor Vehicles, State of California [6], majority of them being rear-end collisions involving human drivers. Therefore, it is required to design optimal velocity control systems that can optimize between safe driving and efficient driving.

Through this research, we have proposed a velocity optimization strategy that optimizes the velocity profile for fuel consumption, without significantly compromising safety and affecting the traffic flow. An online Model Predictive Controller is designed to compute the optimal velocity profile based on fuel consumption and impact to the surrounding traffic. A mathematical parameter is introduced for deterministic control of impact on traffic flow. An iterative convex optimization approach is adopted for online solution of the optimal control problem. A simulation case study followed by a parametric study is presented to demonstrate fuel saving capability and reduced impact on the surrounding traffic flow, of the proposed control system.

1.1 Velocity Optimization

A significant amount of research has been carried out to develop optimal driving algorithms that maximize fuel efficiency, minimize energy consumption, minimize total trip time etc. A cloud-based optimal velocity planning approach is proposed in [16]. A dynamic programming algorithm is implemented to calculate optimal velocity trajectories using vehicle and fuel consumption models. The test results reflected significant fuel economy improvement without significantly affecting travel time. A distance based two-stage eco-driving strategy is designed in [14]. The two stage heirarchy is composed of a long term and short term velocity planning that optimizes fuel consumption and a short term planning for safe traffic following. A predictive cruise control system is proposed by Asadi and Vahidi [2] to control the velocity trajectory by optimizing timely arrival at traffic lights in their green phase, while minimizing deviations from the set speed and braking. A dynamic programming based robust optimal control strategy is proposed by [21] to optimize fuel consumption considering uncertain traffic signal timings. In [9] topographic information is used to determine the fuel optimal velocity profile. In this study, we have proposed a velocity optimization algorithm that minimizes fuel consumption in consideration of future state of traffic flow and traffic lights, road grade, longitudinal vehicle dynamics, without significantly compromising safety and causing obstruction to the traffic flow, as a result of conservative driving. A macroscopic traffic flow model has been formulated to predict admissible speeds over an urban route consisting of a series of signalized intersections. The predicted admissible speeds are used as a reference to calculate optimal velocities that reduce fuel consumption without significantly impacting the flow of traffic using Model Predictive Control.

1.2 Model Predictive Control

Model predictive control (MPC) is an advanced control method that can control a dynamic system while satisfying an optimality criterion and a set of constraints. MPC can handle dynamic systems with multiple inputs and outputs and having interactions between inputs and outputs, and constraints on input and output variables. With the MPC, it is also possible to optimize current states, while keeping future states in account [8]. This is achieved by optimizing over a finite horizon, implement-

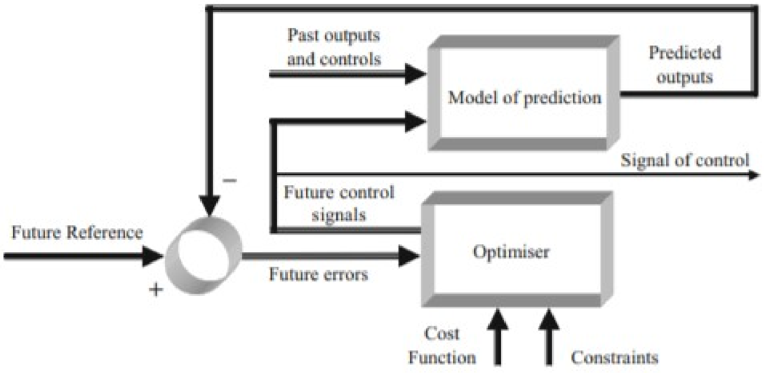


Figure 1.1: General Scheme of Model Predictive Control [3]

ing the current states, receding the horizon, repeating the process [8]. MPC is also capable of predicting a series of future states and planning control actions accordingly. The other conventional controllers, such as Proportional-Integral-Derivative control, do not have the ability. PID controllers can only plan control actions based on the current state.

Due to the above stated advantages, we have selected MPC for solving our research problem. The MPC is formulated as an online optimization that iterates between the traffic model and a convex optimization problem. Once admissible velocities are calculated from the traffic model, they are used to calculate optimal velocities for a subject vehicle present in the traffic flow, which are incorporated into the traffic model using a set of coupled equations. The admissible velocities are recalculated and the process is repeated.

A general schematic diagram of an MPC controller is shown in Figure 1.1. A dynamic model is used to predict future states and control actions from past states and control actions. The optimiser compares the predicted states with a reference and calculates control actions that satisfy constraints and minimize a cost function.

Chapter 2

Dynamic Models and Equations

To formulate our research problem, a mathematical model to characterize, simulate and visualize the behavior of traffic flow is required. The longitudinal motion dynamics and fuel consumption characteristics of the subject vehicle on which the proposed control system is implemented are also required to be known. Coupled equations are required to be formulated to locate the position of the subject vehicle in the traffic flow. In this chapter, the required vehicle, fuel consumption and traffic models and coupled equations are discussed in detail.

2.1 Vehicle Model

The subject vehicle is assumed to be a passenger vehicle equipped with a gasoline Internal Combustion Engine (ICE). Since the objective is to find optimal longitudinal velocity trajectory, only the longitudinal dynamic equations were considered, and the lateral dynamics are disregarded. The longitudinal dynamic equations are determined by acceleration, velocity and position in the longitudinal direction. The acceleration is determined using The longitudinal position and velocity are determined

Table 2.1: Vehicle Parameters and Coefficients

Parameter	Value	Unit
Vehicle Mass m	1707	kg
Gravitational Acceleration g	9.81	m/s^2
Rolling Resistance C_r	0.05	
Aerodynamic Drag Coefficient C_d	0.29	
Air Density ρ_a	1.18	kg/m^3
Vehicle Frontal Area A	2.5	m^2

by integrating the acceleration. The equations are written as:

$$\frac{dX}{dt} = V(t) \tag{2.1}$$

$$m\frac{dV(t)}{dt} = F_t(t) - F_r(t) - F_b(t) \tag{2.2}$$

where m is the vehicle mass, F_t is the traction force, F_b is the braking force, and F_r is the resistance force, calculated using the equation:

$$F_r(t) = mg(cos(\theta)C_r - sin(\theta)) - \frac{1}{2}C_d\rho AV^2(t) \tag{2.3}$$

where g is acceleration due to gravity, θ is the road grade, C_d is the aerodynamic drag coefficient, C_r is the rolling resistance, ρ is the air density and $V(t)$ is the instantaneous longitudinal speed of the subject vehicle. The resistance force is the sum total of the rolling resistance force, road grade resistance and aerodynamic drag force. The vehicle parameter values are depicted in table 1.

2.2 Fuel Consumption Model

The fuel consumption in a gasoline ICE is a characteristic function of the engine torque and engine speed. The engine torque is responsible for acceleration of the vehicle. The engine speed is also proportional to the vehicle's longitudinal speed V by a factor equal to the ratio of wheel radius and gear reduction ratio. Hence, some papers express fuel consumption as a function of vehicle acceleration and the longitudinal vehicle speed V. The fuel consumption model developed by [12] approximates the fuel rate as a non-linear polynomial function of speed and acceleration as:

$$f_v = b_0 + b_1 V + b_2 V^2 + b_3 V^3 + a(c_0 + c_1 V + c_2 V^2) \tag{2.4}$$

where a is the acceleration of the vehicle. The coefficients $b_0 = 0.1569,\ b_1 = 2.45 \times 10^{-2},\ b_2 = -7.415 \times 10^{-4},\ b_3 = -7.415 \times 10^{-4}$ and $c_0 = 0.07224,\ c_1 = 9.681 \times 10^{-2},\ c_2 = 1.075 \times 10^{-3}$ are scalar constants.

2.3 Traffic Model

Traffic flow simulation models are an important tool used by traffic engineers to characterise the behavior of complex traffic flow systems [10]. Applications of traffic flow models are found in assessing traffic management systems, design and testing of transport facilities, and optimization of traffic operations [10]. According to the level of detail with which they describe the traffic flow, traffic models are categorized as macroscopic and microscopic models. Microscopic models represent the spatio-temporal behavior and interactions between individual vehicles [10]. The key characteristics of a microscopic model are - acceleration of each vehicle, relative position of each vehicle with respect to a neighbouring vehicle, and relative velocity of each vehicle with respect to the neighbouring vehicle. Macroscopic models on the other hand, represent the traffic flow by considering aggregate behavior of a volume of vehicles. The important properties of a macroscopic model are - flow rate, density and velocity. Macroscopic models approximate a queue of vehicles as a continuum. Owing to its low complexity, ease of formulation and computational inexpensiveness, macroscopic modeling approach was used to simulate the flow of traffic for our research problem.

To model the flow of traffic on an urban route comprised of a sequence of traffic lights at fixed locations, a discretized numerical formulation of the macroscopic model developed by Lighthill and Whitman and Richards (LWR) [4] is developed. It manifests itself as a one-dimensional conservation equation subject to fixed initial and boundary conditions:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0,\ x \in R_{\geq 0},\ \ t > 0 \tag{2.5}$$

Where $\rho \in [0,\ \ \rho_{max}]$ is the density of vehicles, and q is the traffic flow characterized by the fundamental diagram $q = f(\rho)$, x is any location in the positive 1D space and t is any instant in time.

2.3.1 Finite Difference Approximation

Solving the traffic model described by equation (2.5) using a finite difference scheme was a significant challenge, as the initial and boundary conditions are discontinuous. Carlos F. Daganzo [5] has presented a finite difference scheme that results in a stable solution in the presence of shocks (disturbances) resulting out of discontinuous boundary conditions and/ or initial conditions. The continuous equation is discretized over the time interval Δt and spacial interval Δx. The descretized equa-

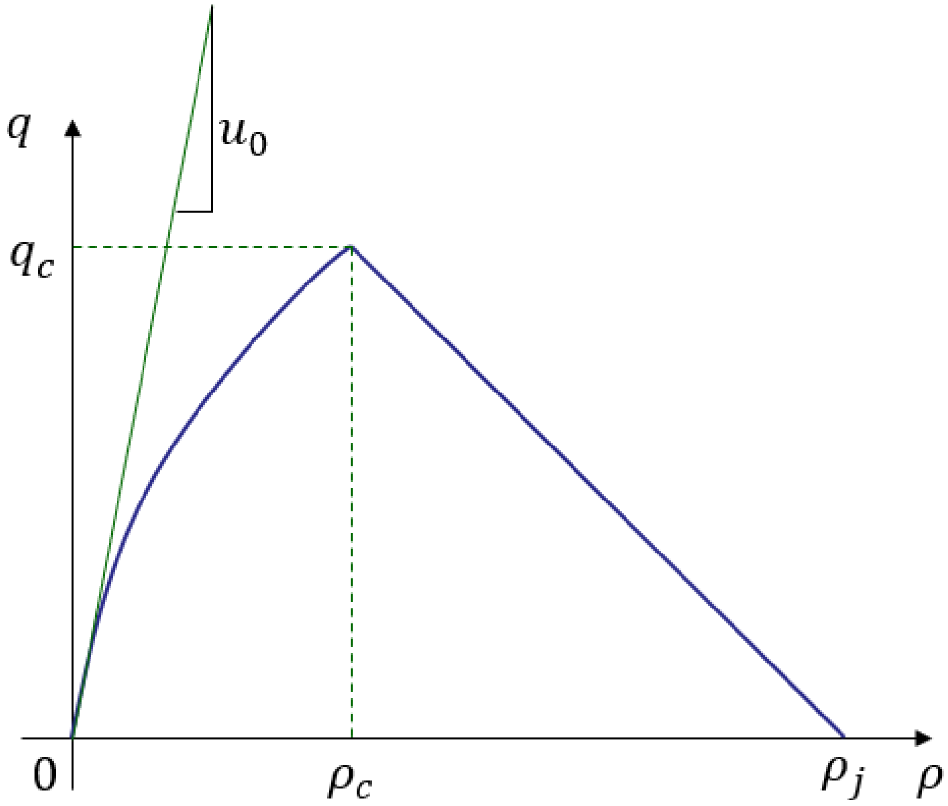


Figure 2.1: Fundamental Diagram

tion reads as

$$\rho^i(k+1) = \rho^i(k) + \frac{\Delta t}{\Delta x}(q^i(k) - q^{i+1}(k)) \tag{2.6}$$

where $k = 1, 2, 3, ...$ and $i = 1, 2, 3, ...$ are the integer counts corresponding to the current discrete instant in time and discrete position in space respectively.

The vehicle flow is an explicit function of the vehicle expressed by the fundamental diagram equation

$$q^i(k) = \begin{cases} u_0\ \rho^i(k)(1 - \frac{\rho^i(k)}{\rho_j}) & \text{for } \rho \in [0,\ \ \rho_c] \\ u_0\ \rho_c(1 - \frac{\rho^i(k)}{\rho_j}) & \text{for } \rho \in (\rho_c,\ \ \rho_j] \end{cases} \tag{2.7}$$

where u_0 is the speed limit or maximum speed, ρ_c is the critical density when the flow is maximum, and ρ_j is the maximum density. The flow at traffic signal locations I during the red phase of the traffic lights T_{red} is forced to zero. During the green phase T_{green}, it is equal to the flow as calculated by the fundamental diagram function f, described by equation (2.7).

$$q^I(k) = \begin{cases} 0 & \text{for} k \in T_{red} \\ f(\rho^I(k)) & \text{for } k \in T_{green} \end{cases} \tag{2.8}$$

The admissible speed is calculated from the flow and density as

$$v^i(k) = minimum\ (\ \frac{q^i(k)}{\rho^i(k)}\ ,\ Speed\ Limit(i)) \tag{2.9}$$

When the vehicle density nears zero at a any location, the admissible speed for a vehicle is equal to the speed limit at that location.

For Finite Difference Models (FDE) to yield a stable feasible results, small disturbances due to initial conditions or boundary conditions, must propagate forward when traffic is light, backward when traffic is heavy and never faster than the vehicles causing the disturbance [5]. To capture this phenomenon correctly, a modification is made to the fundamental diagram equation (2.7). Daganzo [5] proposes to approximate the flow diagram by

$$q^i(k) = \min(T(\rho^i(k))\ ,\ R(\rho_j - \rho^{i+1}(k))) \tag{2.10}$$

where

$$T(\rho) = \begin{cases} q_c & \text{for } \ \rho \in [0\ ,\ \rho_c] \\ u_0\rho_c(1 - \frac{\rho}{\rho_j}) & \text{for } \rho \in (\rho_c\ ,\ \rho_j] \end{cases} \tag{2.11}$$

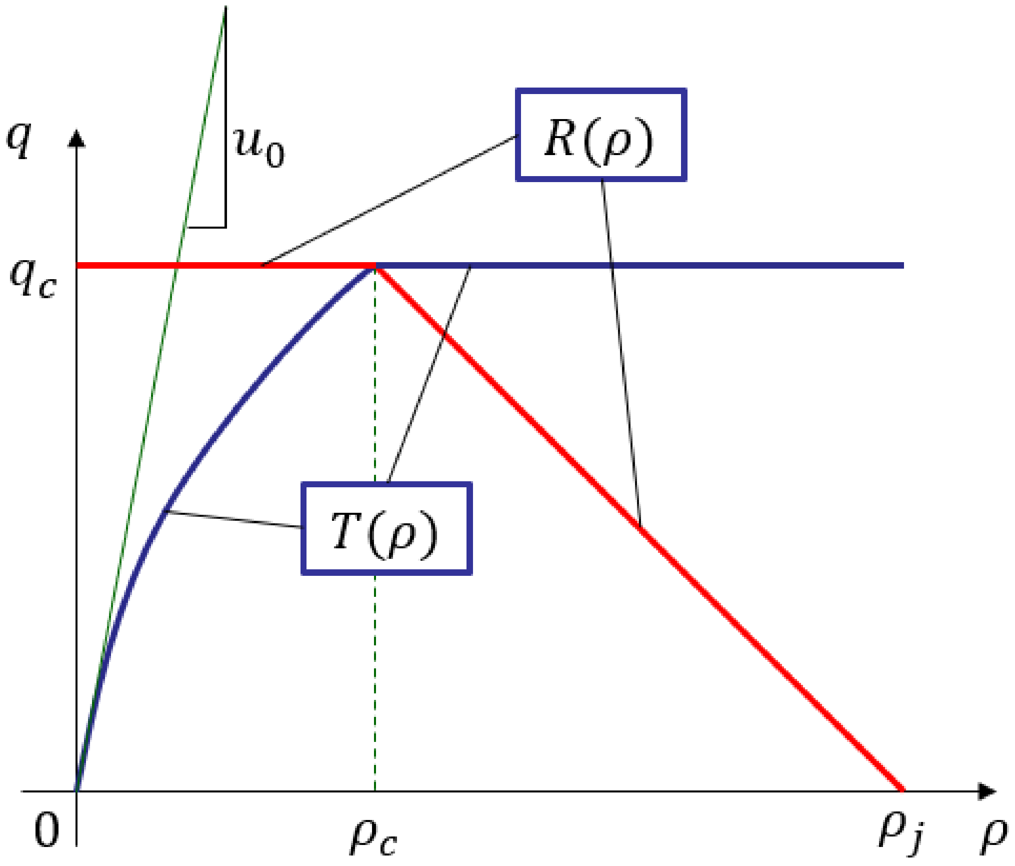


Figure 2.2: Modified Fundamental Diagram

$$R(\rho) = \begin{cases} u_0\rho(1-\frac{\rho}{\rho_j}) & \text{for } \ \rho \in [0\ ,\ \rho_c] \\ q_c & \text{for } \rho \in (\rho_c\ ,\ \rho_j] \end{cases} \tag{2.12}$$

The modified fundamental diagram is shown in Figure 2.2. The cell size condition for stability [5] is given by

$$\frac{\Delta x}{\Delta t} \geq u_0 \tag{2.13}$$

2.4 Coupled Equations

The subject vehicle described by (2.1)-(2.3) is travelling with a different (optimal) speed as compared to the other vehicles in the traffic flow, as its velocity is determined by the proposed control system. To describe the distinct behavior of the subject vehicle, the vehicle dynamic equations (2.1)-(2.3) are explicitly coupled with the vehicle dynamics. This is achieved by equating the flow rate at the subject vehicle's spatial position is equal to the product of density at that location and speed of the subject vehicle. The coupled equation for traffic flow on a single lane road is given by:

$$v^I(k) = \rho^I(k)V(k) \tag{2.14}$$

where I is the position index of the subject vehicle.

The coupled equations for traffic flow on a multi-lane road is given by

$$v^I(k) = \frac{\rho^I(k)V(k)}{n} + \frac{(n-1)q^I(k)}{n} \tag{2.15}$$

where, n is the number of lanes.

Upon solving the traffic model, we get a space-time plot of densities (refer Figure), flow rates and admissible velocities. The plot is depicted in the form of a 2D color plot. The color bar is a representation of the magnitude of density. Yellow zones indicate queuing of vehicles at signalized intersections during the red phase of the traffic lights.

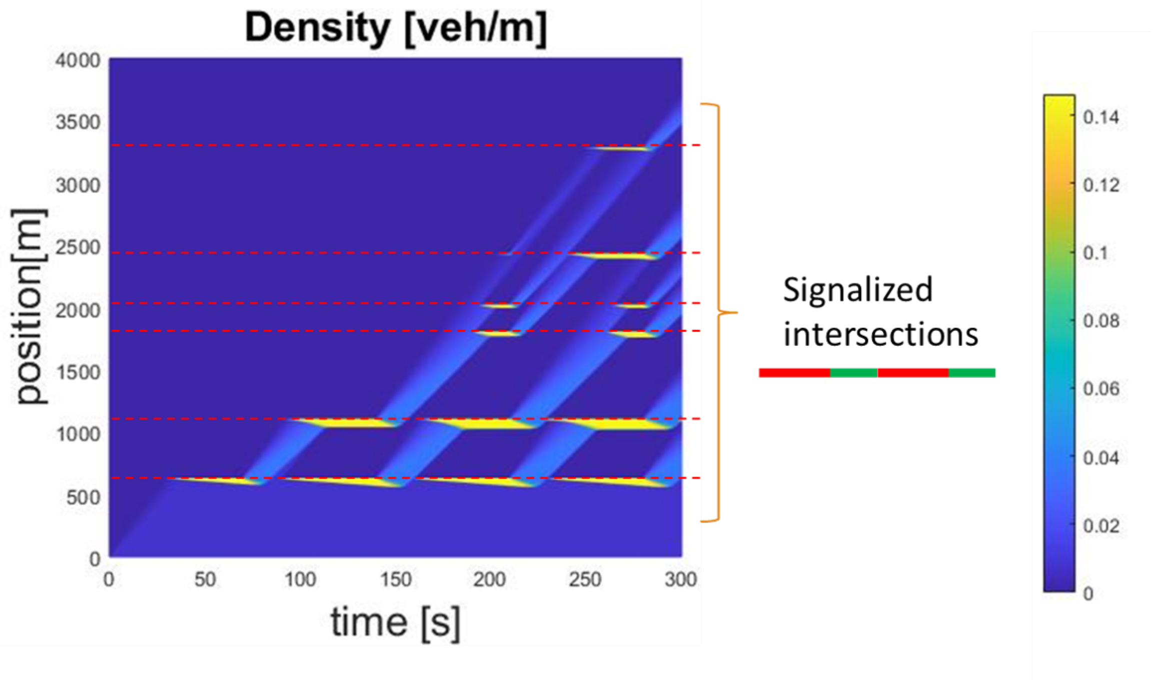


Figure 2.3: Space-time plot of vehicle density

Chapter 3

Problem Formulation and Methods

3.1 Optimal Control Formulation

Once, the dynamic equations and coupled equations are formulated, an optimal control problem is formulated to calculate optimal control inputs and generate optimal velocity trajectory. The optimal control problem minimizes an objective (cost) function subject to a set of constraints, and computes optimal variables. Our objective is to minimize the total fuel consumption of the subject vehicle, and the impact on the surrounding traffic. The impact on the surrounding traffic will be minimum if, the subject vehicle drives as close as possible to the admissible speed i.e. the speed of the flow of traffic. The impact is measured by the difference between admissible speed and the subject vehicle speed. If the optimal controller is biased towards fuel consumption, then the impact is expected to be greater and result in less fuel consumption. On the other hand, if the controller is biased towards the impact term, then it is expected to result in a relatively greater fuel consumption, but also relatively less impact. The objective function is represented by the equation:

$$J = \begin{aligned} & \sum_{k=1}^{k=N} [\, \dot{m}_f(F_t(k)\,, V(k))\,]\, \Delta t \\ + \quad & W\, ||\, C(k)\, || \;\; + \;\; W_1\, ||(F_t(k+1) - F_b(k+1)) - (F_t(k) - F_b(k))|| \\ + \quad & W_2\;\, ||\, V^j(k) - V^{j-1}(k)\, || \end{aligned} \tag{3.1}$$

where $k = 1, 2, 3, ...$, $C(k)$ is the difference between the instantaneous speed of the subject vehicle, $V(k)$ and the admissible speed $v^i(k)$, expressed in equation (2.9). $||.||$ is the Euclidean norm. The norm of $C(k)$ is the impact term. It quantifies the impact on surrounding traffic. W is the penalty weight of the impact term, which will be referred to as "Impact Factor". W_1 and W_2 are penalty weights associated with the comfort cost. The comfort cost is related to passenger discomfort due to the G-forces from acceleration and deceleration of the vehicle.

The optimization is subject to the vehicle dynamic constraints

$$X(k+1) = X(k) + \frac{\Delta t}{2}(V(k) + V(k+1)) \quad \text{for } k = 1, 2, .., N-1 \tag{3.2}$$

$$\begin{aligned} V(k+1) = V(k) + \frac{\Delta t}{m}(F_t(k) - F_r(k) - F_b(k)) \\ \text{for } \; k = 1, 2, ..., N-1 \end{aligned} \tag{3.3}$$

$$F_r(k) = mg(cos(\theta)\; C_r - sin(\theta)) - \frac{1}{2} C_d\, \rho_a\, A\, V^2(k) \tag{3.4}$$

Limits are enforce on the traction force F_t, braking force F_b and the vehicle speed V through the linear constraints

$$0 \ \leq \ F_t(k) \ \leq \ m\, a_{max} \tag{3.5}$$

$$m\, a_{min} \ \leq \ F_b(k) \ \leq 0 \tag{3.6}$$

$$V(k) + C(k) \ = \ v^i(k) \tag{3.7}$$

$$\begin{aligned} V(k) &\geq 0 \\ C(k) &\geq 0 \end{aligned} \tag{3.8}$$

The subject vehicle is set to reach a desired destination at distance d from its current position X_0 with a terminal velocity equal to the admissible speed at that location.

The initial and terminal constraints are written as:

$$X(1) = X_0 \tag{3.9}$$

$$V(1) = v^0(1) \tag{3.10}$$

$$X(N) = X_0 + d \tag{3.11}$$

$$V(N) = v^i(N) \tag{3.12}$$

The objective function is also subject to the traffic constraints (2.1)-(2.3) and coupled equation contraints (2.14)-(2.15).

Sections 3.2 contains detailed discussion of the proposed solution method. The possible approaches for solving the proposed problem consist of dynamic programming and iterative convex optimization. Since trajectory optimization problems are often non-convex, dynamic programming approach is used in many papers [2,16]. Although dynamic programming can provide a global solution to the eco-driving problem, long computation times make it difficult to implement in real-time [14]. Hence, other approaches are explored by researchers to solve non-linear optimization problems. Sequential Quadratic Programming approach is demonstrated by [14] for solving a two-stage optimization for a distance-based ecological driving scheme. Sequential convex optimization approach provides approximate solutions to non-linear problems by sequentially forming convex sub-problems and converging to a local minimum [11]. Sequential Convex Programming has been used in eco-driving problems as well trajectory optimization in robots [11, 18].

In this study, we have used iterative convex optimization approach to solve the problem defined by equations (3.1)-(3.12).

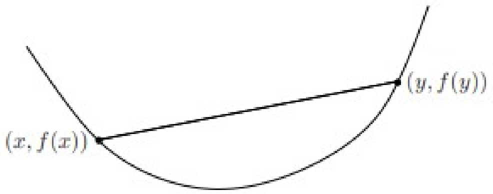


Figure 3.1: Graph of a convex function. The chord (i.e., line segment) between any two points on the graph lies above the graph [19]

3.2 Iterative Convex Optimization Solution

3.2.1 Convex Optimization

Convex optimization is a commonly used approach for solving optimization problems in the areas of automatic control systems, estimation and signal processing, communications and networks, design of electronic circuits, data analysis and modeling, statistics and finance [19]. It is a special class of problems of the form:

$$\textit{Minimize} \quad f_0(x) \tag{3.13}$$

$$\textit{Subject to} \quad f_i(x) \leq b_i \quad i = 1,\ 2,\ ...,\ m \tag{3.14}$$

where, $f_0,\ f_1,\ ...,\ f_m : \mathbb{R}^n \ \rightarrow \ \mathbb{R}$ are convex, or in other words satisfy the condition

$$f_i(\alpha x + \beta y) \leq \alpha x + \beta y \tag{3.15}$$

for all $x, y \in \mathbb{R}$ and all $\alpha, \beta \in \mathbb{R}$ with $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$. Interior point method is most commonly used to solve these problems [19].

3.2.2 Global vs Local Optimization

In global optimization, the aim is to find the optimal variable x which minimizes the objective over all feasible points [19]. Local optimization on the other hand, seeks a solution that is only locally optimal, i.e. it minimizes the objective function among the neighbouring points, however, not guaranteed to a lower objective as compared to all the feasible points [19]. Local optimization problems require an initial guess, that greatly affect the value of the local solution, and are very sensitive to algorithm parameters. Global optimization becomes exponentially more complex as the size of the problem increases [19]. Local optimization problems solve relatively faster. Thus, global optimization is suitable for problems where computation time is not critical, and a global solution is indispensable [19]. Trajectory optimization problems require to solve in real time, making computation time a more critical factor as compared to finding a global solution. Hence a local optimization approach is adopted.

3.2.3 Convex Approximation and Iterative Approach

Solution to non-convex problems can be determined by finding an exact solution to an approximate convex problem. This point is then used as the starting point for a local optimization method, applied to the original non-convex problem. For solving our problem we propose using an iterative convex approach. To generate an initial guess, the subject vehicle is assumed to follow the admissible speed trajectory i.e. the subject vehicle follows the speed of the traffic flow in front of it.

$$V_{guess}(k) = v^i(k) \quad k = 1, 2, 3...N \tag{3.16}$$

$$X_{guess}(k) = \begin{cases} 0 & \text{for } k = 1 \\ X_{guess}(k-1) + V_{guess}(k-1)\Delta t & \text{for } k = 2, 3, .., N \end{cases} \tag{3.17}$$

The objective of the first iteration is:

$$J_1 = \begin{matrix} W \; ||V(k) - V_{guess}(k)|| \; + \\ W_1 \; ||(F_t(k+1) - F_b(k+1)) - (F_t(k) - F_b(k))|| \end{matrix} \tag{3.18}$$

The non-linear vehicle dynamic constraint (2.3) is linearized, shown in the equation below:

$$F_r(k) = mg(cos(\theta)\; C_r - sin(\theta)) - \frac{1}{2} C_d \, \rho_a \, A \, (2VV_{guess} - V_{guess}^2(k)) \tag{3.19}$$

The optimal values obtained are passed as a guess to the next iteration. In general, the optimal values from iteration $j-1$ are passed as guesses to iteration j.

$$X_{guess}^{j+1}(k) = X_{optimal}^{j}(k) \quad \text{where } j = 1, 2, 3, ...M \tag{3.20}$$

$$V_{guess}^{j+1}(k) = V_{optimal}^{j}(k) \quad \text{where } j = 1, 2, 3, ...M \tag{3.21}$$

The process illustrated by equations (3.18)-(3.21) is repeated M times. The objective function for subsequent iterations $j = 2, 3, ..M$ is represented by equation

$$J_j = \begin{array}{ll} & \sum_{k=1}^{k=N} [\, \dot{m}_f(F_t(k)\,, V(k))\,]\, \Delta t \\ + & W \,||\, C(k)\, || \;+\; W_1\, ||(F_t(k+1) - F_b(k+1)) - (F_t(k) - F_b(k))|| \\ + & W_2 \;||\, V^j(k) - V_{guess}(k)\, || \end{array} \tag{3.22}$$

The last term ensures a stable convergent solution to the iterative convex optimization. After M iterations, the traffic velocities are computed again using the coupled equation

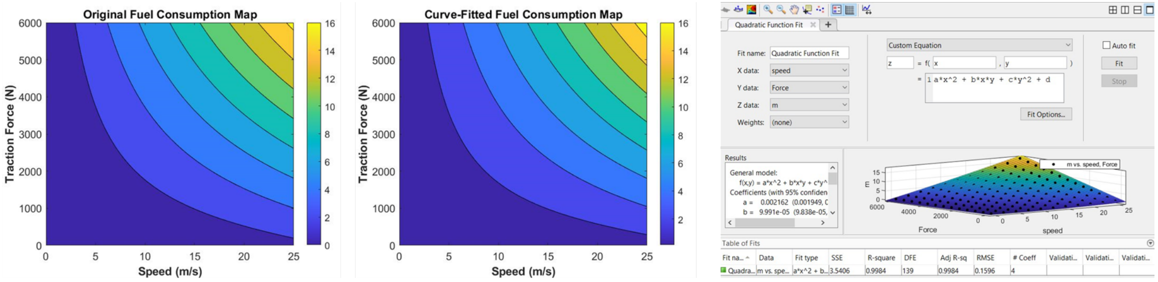
$$v^I(k) = \frac{\rho^I(k) V^M_{optimal}(k)}{n} + \frac{(n-1)q^I(k)}{n} \tag{3.23}$$

The flow rates and densities are recalculated using equations (2.5)-(2.15). The process (3.18)-(3.23) is repeated multiple times. The section illustrates the overview of the algorithm used.

3.2.4 Quadratic Cost Modification

The fuel consumption model in equation (2.4) is based on a non-convex function. To adapt it to our problem, we have approximated the model by a convex quadratic Function.This is achieved using curve fitting. 150,000 data points are calculated using equation (2.4), and a quadratic curve represented by equation (3.24) is fitted using MATLAB curve fitting tool. The coefficient of determination of the fit is $R^2 \approx 0.94$

$$\dot{m}_f(t) = \alpha_1\, V^2(t) + \alpha_2\, V(t)\, F_t(t) + \alpha_3\, F_t^2(t) + \alpha_0 \tag{3.24}$$



Fit name: Quadratic Function Fit
X data: speed
Y data: Force
Z data: m
Weights: (none)

Custom Equation

z = f(x , y)

= 1 a*x^2 + b*x*y + c*y^2 + d

General model:
f(x,y) = a*x^2 + b*x*y + c*y^
Coefficients (with 95% confiden
a = 0.002162 (0.001949, C
b = 9.991e-05 (9.838e-05,

Fit na...	Data	Fit type	SSE	R-square	DFE	Adj R-sq	RMSE	# Coeff	Validati...	Validati...	Validati...
Quadra...	m vs. spe...	a*x^2 + b...	3.5406	0.9984	139	0.9984	0.1596	4			

Figure 3.2: Fuel Consumption Map: The original fuel consumption map (left) is curve-fitted into a quadratic function (center) using MATLAB curve fitting tool (right)

where $\dot{m}_f(t)$ is the instantaneous fuel consumption rate, and $\alpha_0 = 0.0078$, $\alpha_1 = 4.5 \times 10^{-5}$, $\alpha_2 = 9.98 \times 10^{-8}$, and $\alpha_3 = 0.1569$ are scalar constants. The curve fitting process is demonstrated in Figure 3.2.

$$\sum_{k=1}^{k=N} [\dot{m}_f(F_t(k), V(k))]\Delta t = Z^T Q Z \tag{3.25}$$

where Z is a vector of velocities and traction forces at all times

$Z = [V(1), V(2), ..., V(N), F_t(1), F_t(2), ..., F_t(N)]$

$$Q = \begin{bmatrix} \alpha_1 & 0 & ... & 0 & \alpha_2 & 0 & ... & 0 \\ 0 & \alpha_1 & ... & 0 & 0 & \alpha_2 & ... & 0 \\ . & . & ... & 0 & . & . & ... & 0 \\ . & & . & \alpha_1 & . & & . & \alpha_2 \\ \alpha_2 & 0 & ... & 0 & \alpha_3 & 0 & ... & 0 \\ 0 & \alpha_2 & ... & 0 & 0 & \alpha_3 & ... & 0 \\ . & . & ... & 0 & . & . & ... & 0 \\ . & & . & \alpha_2 & . & & . & \alpha_3 \end{bmatrix}$$

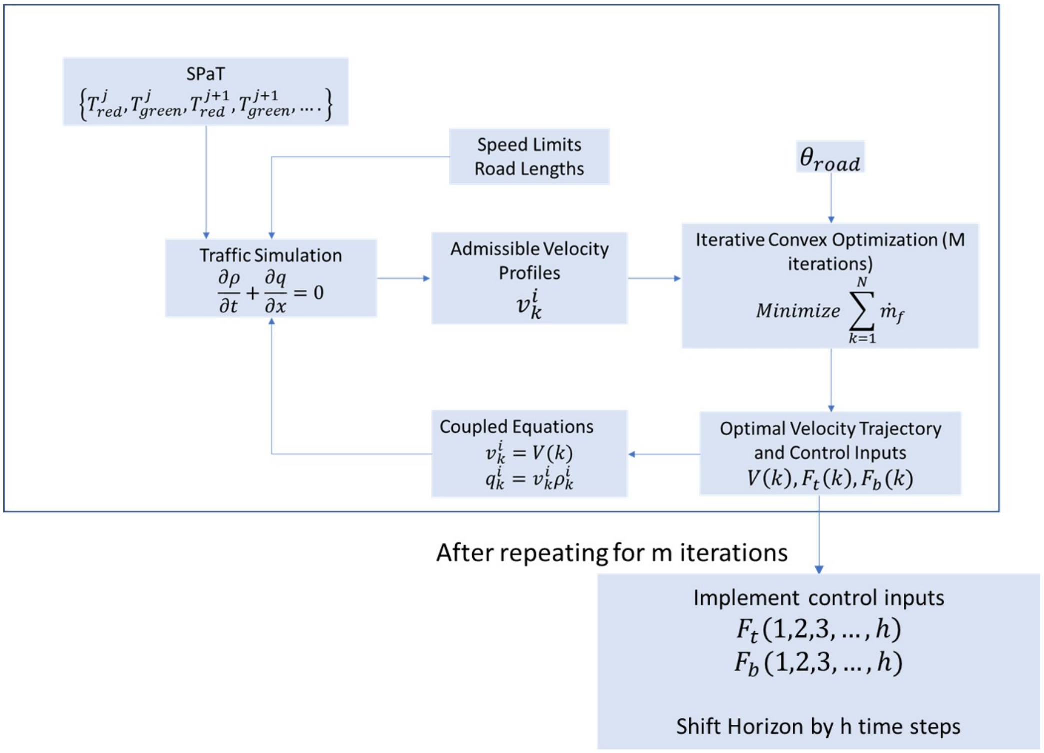

SPaT
$\{T^j_{red}, T^j_{green}, T^{j+1}_{red}, T^{j+1}_{green}, \ldots\}$
Speed Limits
Road Lengths
θ_{road}
Traffic Simulation
$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0$
Admissible Velocity Profiles
v^i_k
Iterative Convex Optimization (M iterations)
$Minimize \sum_{k=1}^{N} \dot{m}_f$
Coupled Equations
$v^i_k = V(k)$
$q^i_k = v^i_k \rho^i_k$
Optimal Velocity Trajectory and Control Inputs
$V(k), F_t(k), F_b(k)$
After repeating for m iterations
Implement control inputs
$F_t(1,2,3,\ldots,h)$
$F_b(1,2,3,\ldots,h)$
Shift Horizon by h time steps

Figure 3.3: Overview of the algorithm process flow

The coefficients α_0, α_1, α_2 and α_3 were selected such that the matrix Q is positive definite, such that the objective function (3.22) remains convex.

3.3 Model Predictive Control and Algorithm Overview

After the optimal velocities are determined for a finite horizon of T time steps, MPC was used to implement first h control inputs are implemented. The horizon is shifted by h steps and the optimization process is repeated. The algorithm overview is depicted in Fig. 3.3.

Chapter 4

Simulation and Results

4.1 Simulation Setup

To demonstrate the capability of the proposed control strategy for fuel saving and reduced impact on the surrounding traffic, a simulation case study was performed. A stretch of Pleasantburg Drive located in the city of Greenville, SC was selected for simulation, and traffic signal phase and timing (SPAT) data was acquired. Road grade data was calculated from elevation data gathered from Google Application Programming Interface. The speed limit was assumed to be 45 miles/hour i.e. 20

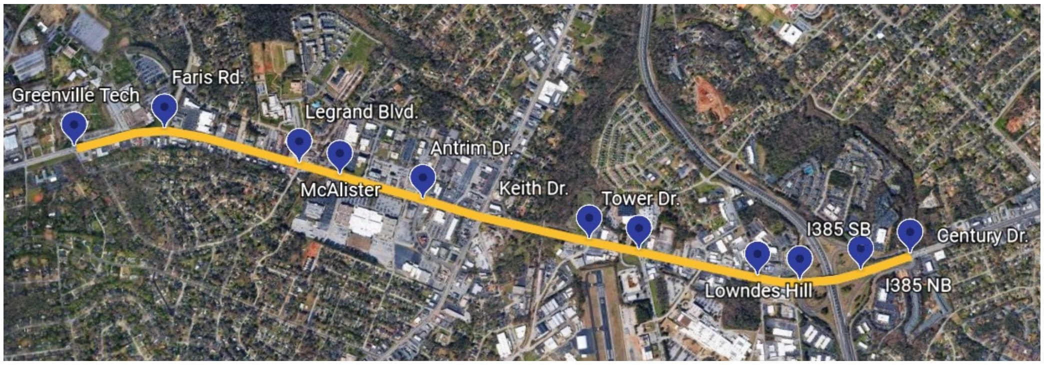


Figure 4.1: Google Earth view of the selected route

m/s. The initial flowrates and densities were assumed to be zero. The flowrate at starting position $X = 0$ was assumed to be 0.25 veh/s. The Google Earth view of the selected route is shown in Figure. The traffic light locations are shown using blue markers. The simulation was setup in MATLAB software. A processor of the configuration six core Intel I-7 10750H, 2.6 GHz base frequency, 12 MB cache and 16 GB RAM. For a total simulation time of 400 seconds, the CPU processing time recorded was 45 seconds. CVX, [7] a MATLAB-based modeling system for convex optimization, was used to formulate optimal control in MATLAB.

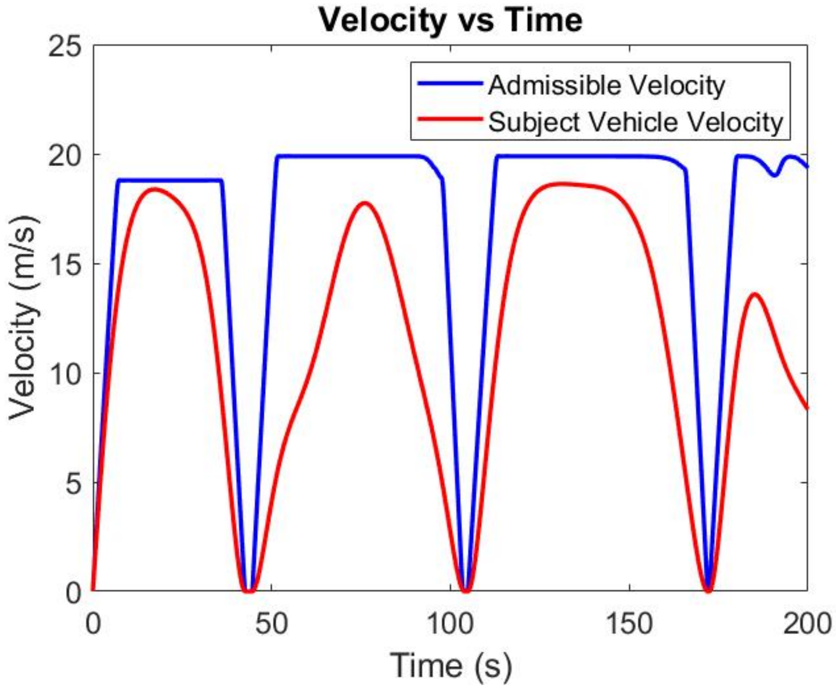


Figure 4.2: Optimal Speed vs Time

4.2 Simulation Results

Upon solving the simulation, the optimal speed trajectory was plotted against time and distance, depicted in Figure and Figure respectively.. The optimal control inputs i.e. traction force and braking force are shown in Figure. The maximum fuel saving was found to be 12.38%.

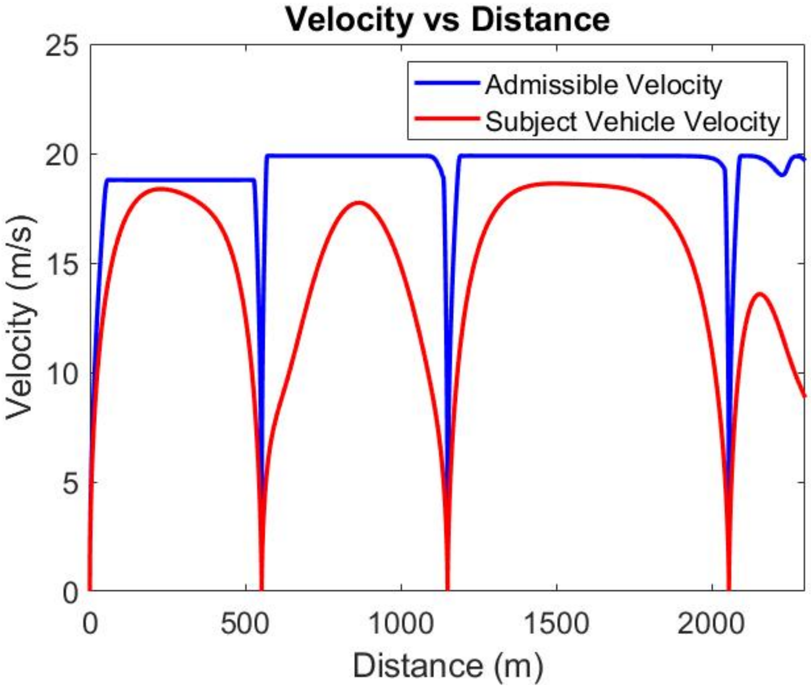


Figure 4.3: Optimal Speed vs Distance

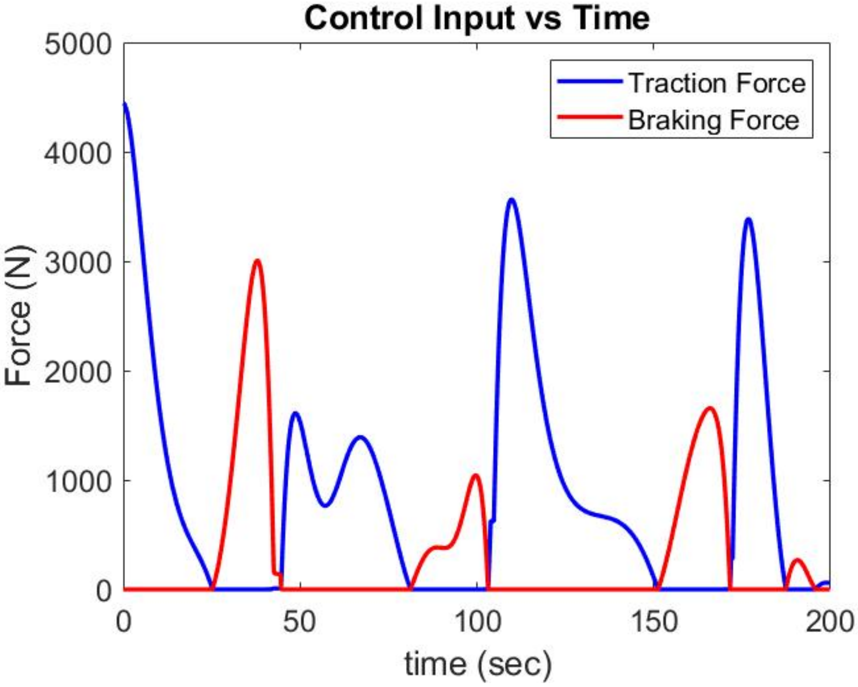


Figure 4.4: Control inputs vs Time

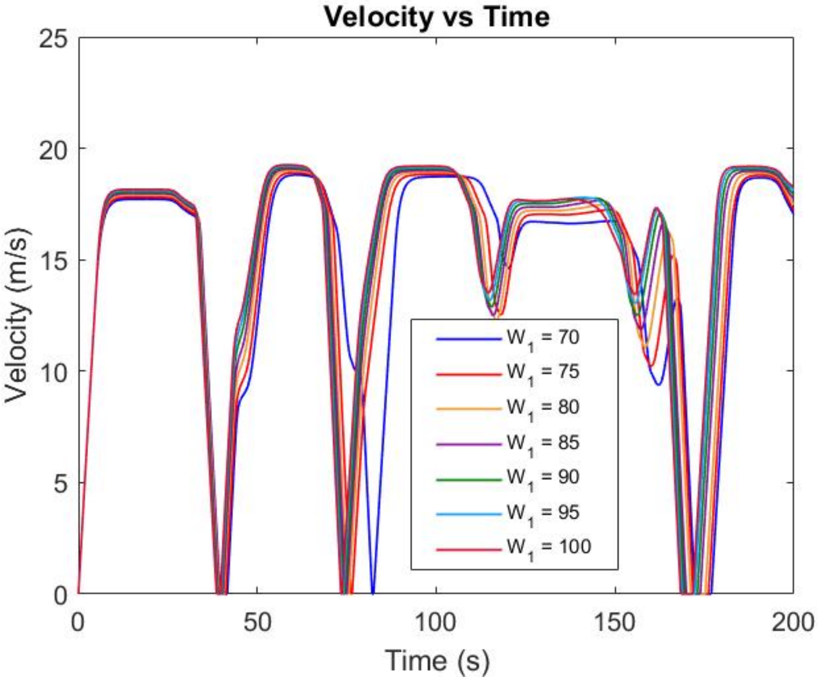


Figure 4.5: Optimal Speed vs Time for different impact factors

4.3 Parametric Study

To be able to assess the impact on surrounding traffic, fuel consumption was calculated for different values of impact factor. Euclidean norm of the difference between admissible Speed and optimal speed was selected as a measure of impact. The plots of velocity profile with respect to distance and time are depicted in Figure and Figure.

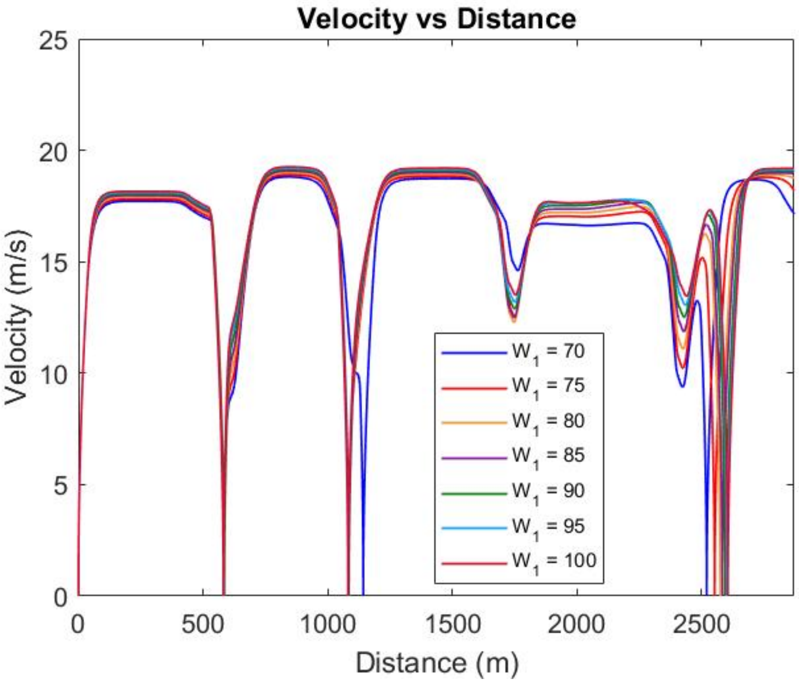


Figure 4.6: Optimal Speed vs Distance for different impact factors

Table shows statistics of impact, fuel consumption and fuel saving for different impact factors. The amount of fuel saved and impact are plotted against impact factor in Figure. As anticipated it was found that as the impact factor increases, the amount of fuel saved decreases, absolute fuel consumption increases and the impact on surrounding traffic decreases.

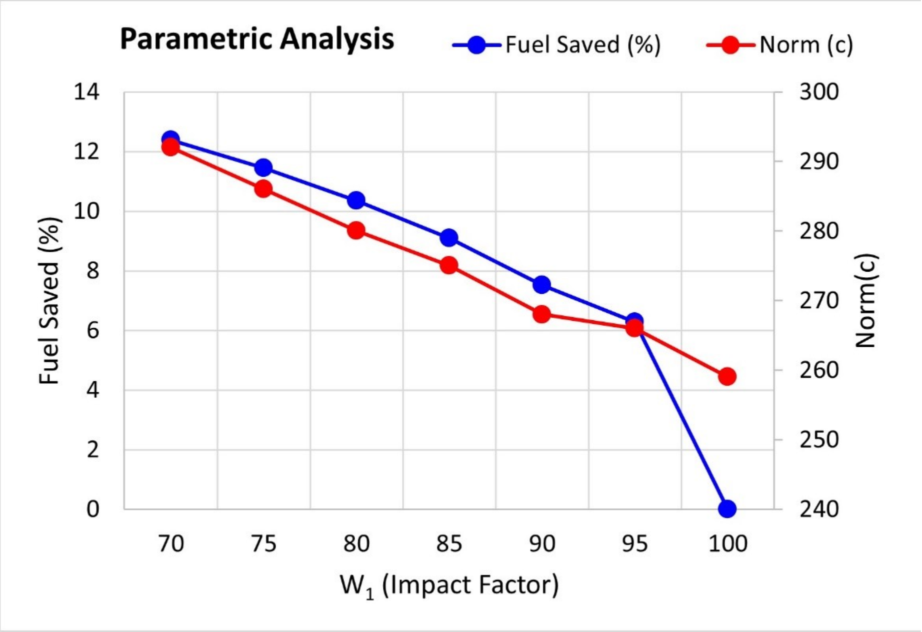


Figure 4.7: Impact Factor Parametric Analysis

Sr.No.	***Impact Factor*** (W_1)	***Fuel (ml)***	***Fuel Saved (%)***	***Norm (C(k))***
1	70	559	12.38	292
2	75	565	11.44	286
3	80	572	10.35	280
4	85	580	9.09	275
5	90	590	7.52	268
6	95	598	6.27	266
6	100	638	0	259

Table 4.1: Parametric Study Statistics

Chapter 5

Conclusions and Discussion

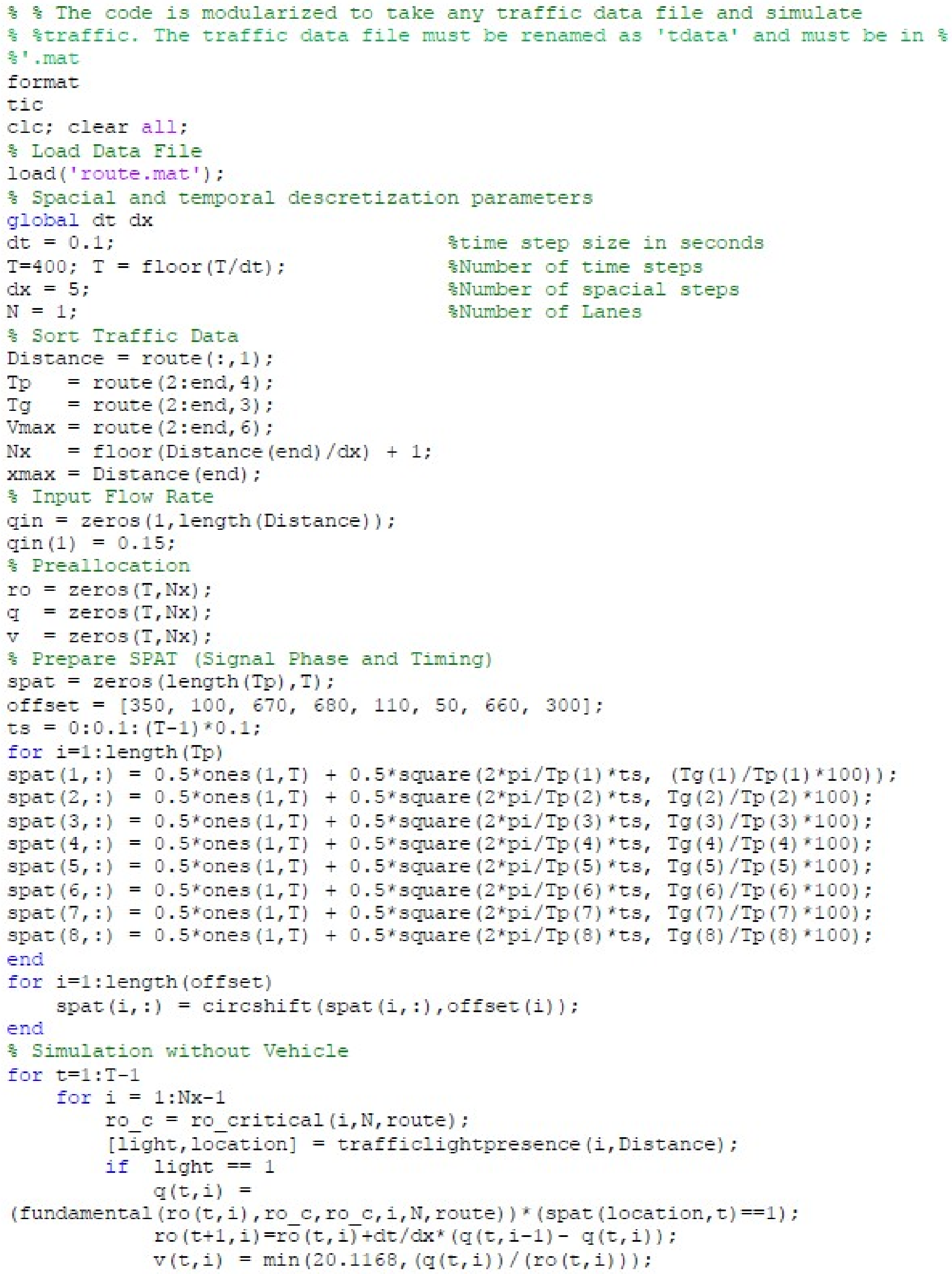
This study proposes a model predictive control strategy for velocity optimization in connected autonomous vehicle to reduce fuel consumption with minimal impact on the surrounding traffic. The iterative convex optimization approach provides a rapid, but robust solution to the optimal controller, rendering it potentially feasible for real-time implementation. From the results of the simulation case study and parametric analysis, it was ascertained that fuel consumption can be reduced by a significant amount while remaining congruous with the surrounding traffic flow consisting of human driven vehicles. By modulating the impact factor parameter it is possible to adjust the behavior of the subject vehicle for fuel efficiency and safety.

Our future work will include simulating multiple traffic scenarios using commercial mobility simulation software. The current traffic model assumes fixed phases and timings of traffic lights. If the traffic lights are actuated, the timings will not remain fixed. Integrating actuated traffic light behavior into the traffic model could be a potential topic for future studies. Modeling and analysis of behavior of multiple CAVs equipped with optimal velocity controllers could also be an interesting study.

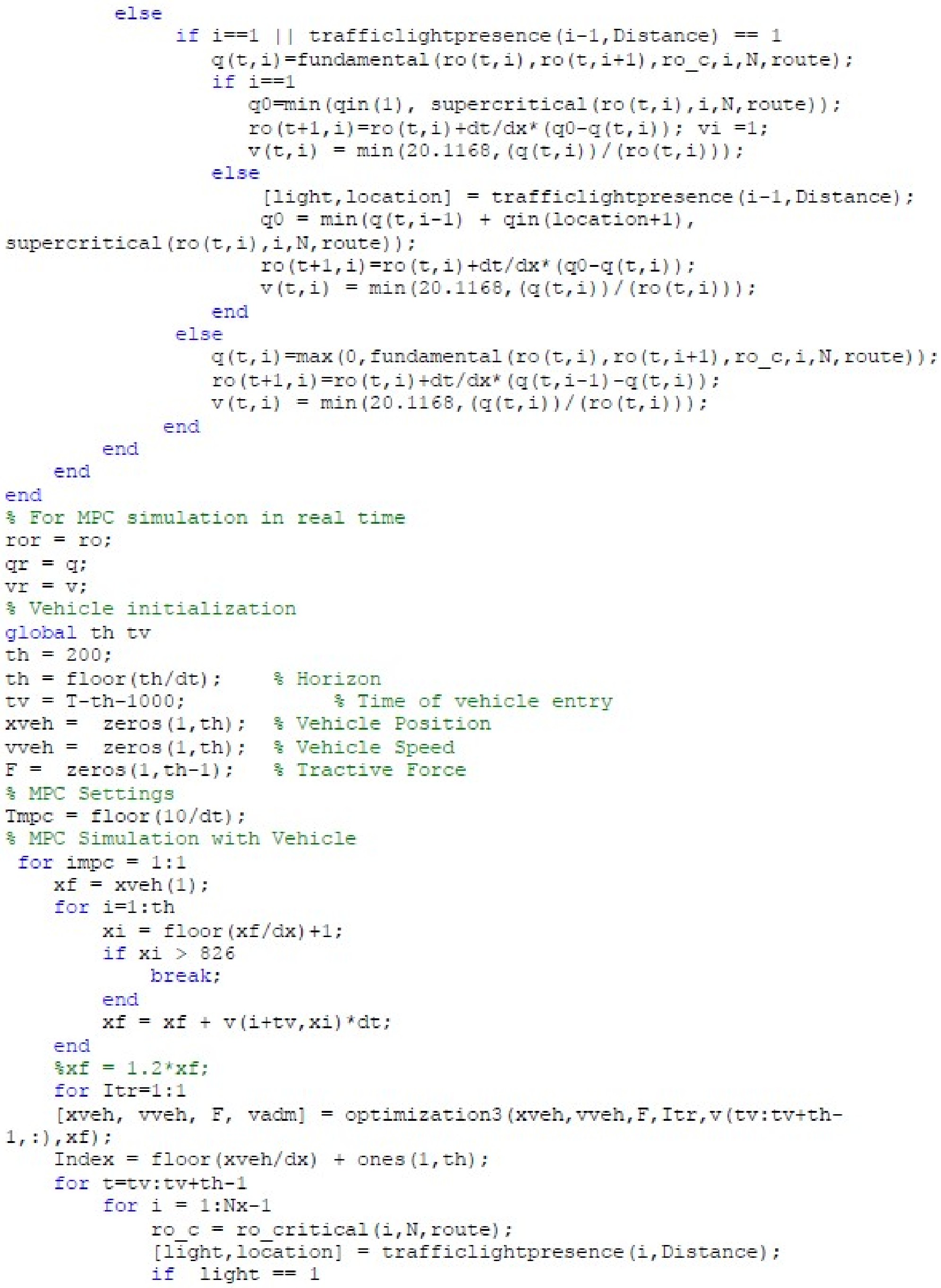
Appendices

Appendix A MATLAB Codes

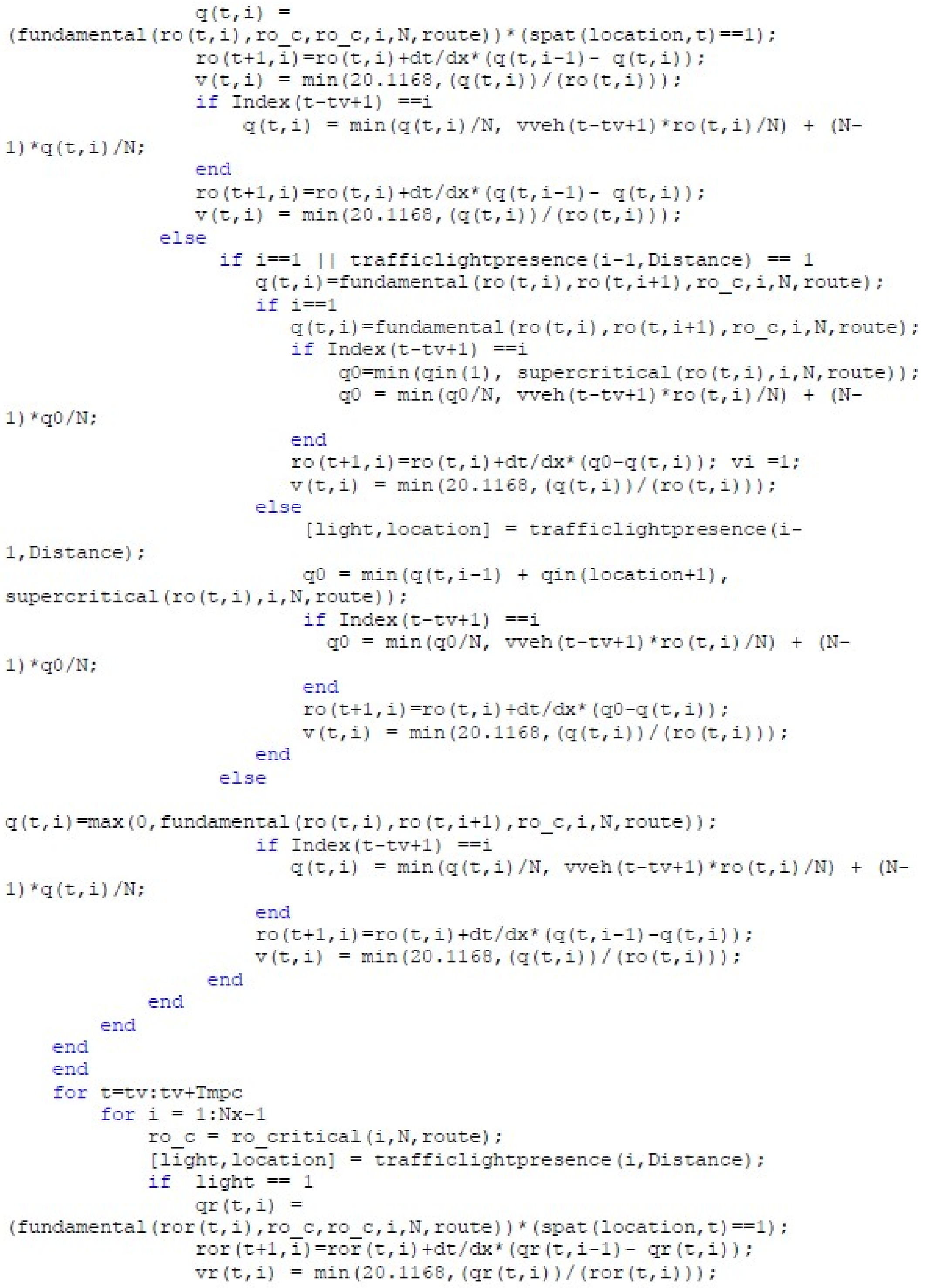
A.1 Main Code



```
% % The code is modularized to take any traffic data file and simulate
% %traffic. The traffic data file must be renamed as 'tdata' and must be in %
%'.mat
format
tic
clc; clear all;
% Load Data File
load('route.mat');
% Spacial and temporal descretization parameters
global dt dx
dt = 0.1;                            %time step size in seconds
T=400; T = floor(T/dt);              %Number of time steps
dx = 5;                              %Number of spacial steps
N = 1;                               %Number of Lanes
% Sort Traffic Data
Distance = route(:,1);
Tp   = route(2:end,4);
Tg   = route(2:end,3);
Vmax = route(2:end,6);
Nx   = floor(Distance(end)/dx) + 1;
xmax = Distance(end);
% Input Flow Rate
qin = zeros(1,length(Distance));
qin(1) = 0.15;
% Preallocation
ro = zeros(T,Nx);
q  = zeros(T,Nx);
v  = zeros(T,Nx);
% Prepare SPAT (Signal Phase and Timing)
spat = zeros(length(Tp),T);
offset = [350, 100, 670, 680, 110, 50, 660, 300];
ts = 0:0.1:(T-1)*0.1;
for i=1:length(Tp)
spat(1,:) = 0.5*ones(1,T) + 0.5*square(2*pi/Tp(1)*ts, (Tg(1)/Tp(1)*100));
spat(2,:) = 0.5*ones(1,T) + 0.5*square(2*pi/Tp(2)*ts, Tg(2)/Tp(2)*100);
spat(3,:) = 0.5*ones(1,T) + 0.5*square(2*pi/Tp(3)*ts, Tg(3)/Tp(3)*100);
spat(4,:) = 0.5*ones(1,T) + 0.5*square(2*pi/Tp(4)*ts, Tg(4)/Tp(4)*100);
spat(5,:) = 0.5*ones(1,T) + 0.5*square(2*pi/Tp(5)*ts, Tg(5)/Tp(5)*100);
spat(6,:) = 0.5*ones(1,T) + 0.5*square(2*pi/Tp(6)*ts, Tg(6)/Tp(6)*100);
spat(7,:) = 0.5*ones(1,T) + 0.5*square(2*pi/Tp(7)*ts, Tg(7)/Tp(7)*100);
spat(8,:) = 0.5*ones(1,T) + 0.5*square(2*pi/Tp(8)*ts, Tg(8)/Tp(8)*100);
end
for i=1:length(offset)
    spat(i,:) = circshift(spat(i,:),offset(i));
end
% Simulation without Vehicle
for t=1:T-1
    for i = 1:Nx-1
        ro_c = ro_critical(i,N,route);
        [light,location] = trafficlightpresence(i,Distance);
        if  light == 1
            q(t,i) =
(fundamental(ro(t,i),ro_c,ro_c,i,N,route))*(spat(location,t)==1);
            ro(t+1,i)=ro(t,i)+dt/dx*(q(t,i-1)- q(t,i));
            v(t,i) = min(20.1168,(q(t,i))/(ro(t,i)));
```

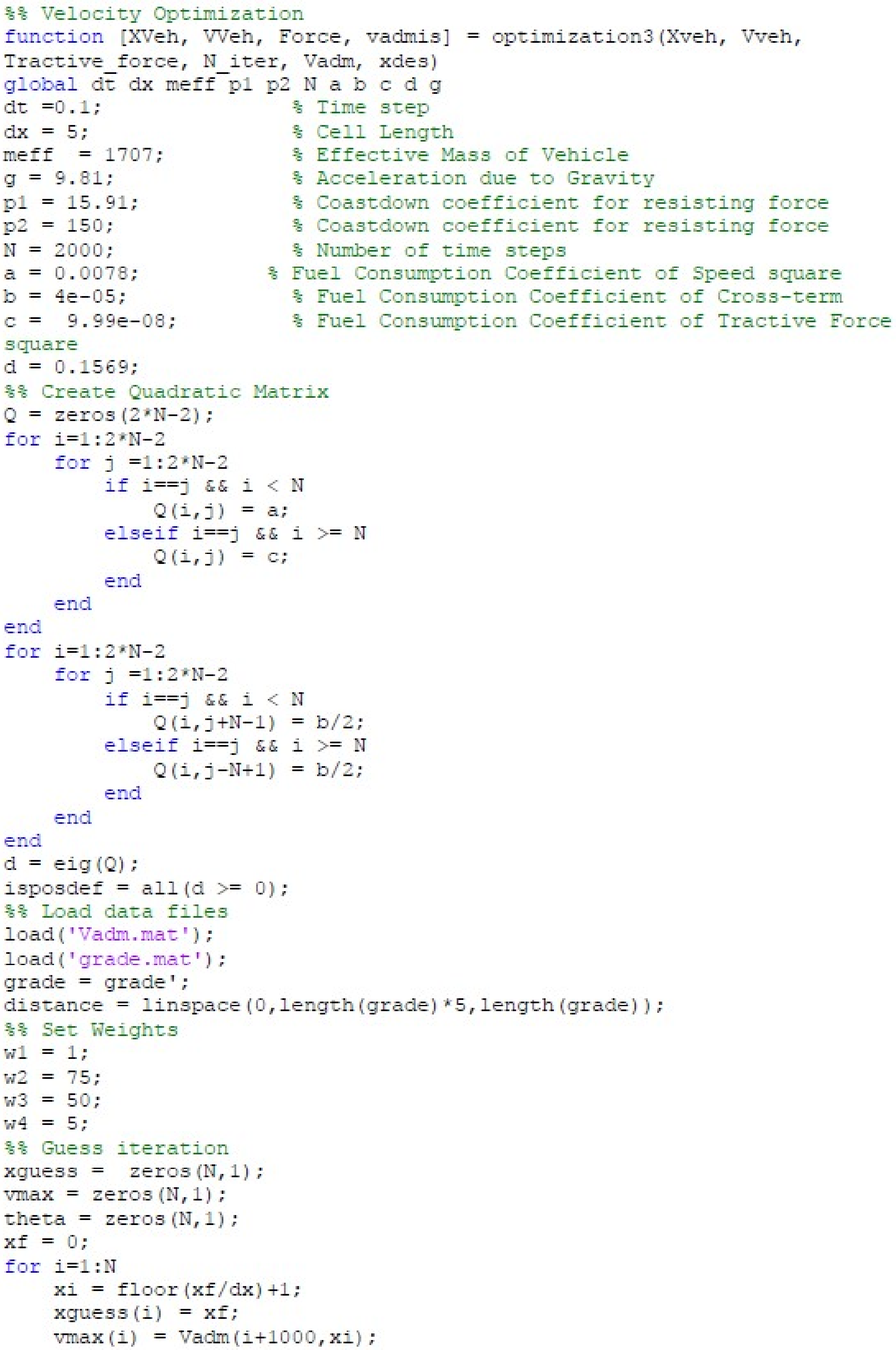


```
         else
              if i==1 || trafficlightpresence(i-1,Distance) == 1
                 q(t,i)=fundamental(ro(t,i),ro(t,i+1),ro_c,i,N,route);
                 if i==1
                    q0=min(qin(1), supercritical(ro(t,i),i,N,route));
                    ro(t+1,i)=ro(t,i)+dt/dx*(q0-q(t,i)); vi =1;
                    v(t,i) = min(20.1168,(q(t,i))/(ro(t,i)));
                 else
                     [light,location] = trafficlightpresence(i-1,Distance);
                     q0 = min(q(t,i-1) + qin(location+1),
supercritical(ro(t,i),i,N,route));
                     ro(t+1,i)=ro(t,i)+dt/dx*(q0-q(t,i));
                     v(t,i) = min(20.1168,(q(t,i))/(ro(t,i)));
                 end
              else
                 q(t,i)=max(0,fundamental(ro(t,i),ro(t,i+1),ro_c,i,N,route));
                 ro(t+1,i)=ro(t,i)+dt/dx*(q(t,i-1)-q(t,i));
                 v(t,i) = min(20.1168,(q(t,i))/(ro(t,i)));
             end
        end
    end
end
% For MPC simulation in real time
ror = ro;
qr = q;
vr = v;
% Vehicle initialization
global th tv
th = 200;
th = floor(th/dt);    % Horizon
tv = T-th-1000;            % Time of vehicle entry
xveh =  zeros(1,th);  % Vehicle Position
vveh =  zeros(1,th);  % Vehicle Speed
F =  zeros(1,th-1);   % Tractive Force
% MPC Settings
Tmpc = floor(10/dt);
% MPC Simulation with Vehicle
 for impc = 1:1
    xf = xveh(1);
    for i=1:th
        xi = floor(xf/dx)+1;
        if xi > 826
            break;
        end
        xf = xf + v(i+tv,xi)*dt;
    end
    %xf = 1.2*xf;
    for Itr=1:1
    [xveh, vveh, F, vadm] = optimization3(xveh,vveh,F,Itr,v(tv:tv+th-
1,:),xf);
    Index = floor(xveh/dx) + ones(1,th);
    for t=tv:tv+th-1
        for i = 1:Nx-1
            ro_c = ro_critical(i,N,route);
            [light,location] = trafficlightpresence(i,Distance);
            if  light == 1
```

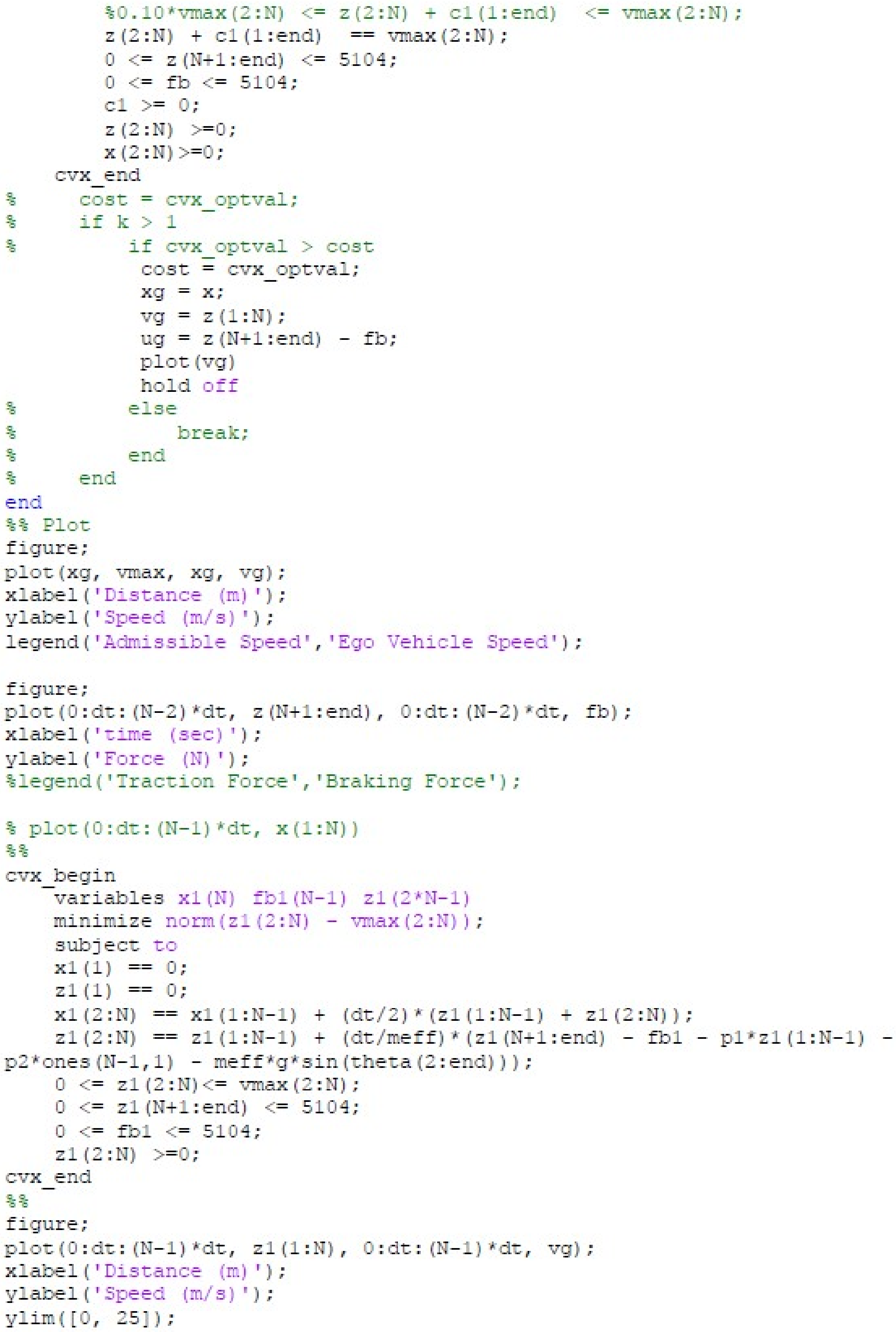


```
                q(t,i) =
(fundamental(ro(t,i),ro_c,ro_c,i,N,route))*(spat(location,t)==1);
                ro(t+1,i)=ro(t,i)+dt/dx*(q(t,i-1)- q(t,i));
                v(t,i) = min(20.1168,(q(t,i))/(ro(t,i)));
                if Index(t-tv+1) ==i
                    q(t,i) = min(q(t,i)/N, vveh(t-tv+1)*ro(t,i)/N) + (N-
1)*q(t,i)/N;
                end
                ro(t+1,i)=ro(t,i)+dt/dx*(q(t,i-1)- q(t,i));
                v(t,i) = min(20.1168,(q(t,i))/(ro(t,i)));
             else
                  if i==1 || trafficlightpresence(i-1,Distance) == 1
                     q(t,i)=fundamental(ro(t,i),ro(t,i+1),ro_c,i,N,route);
                     if i==1
                        q(t,i)=fundamental(ro(t,i),ro(t,i+1),ro_c,i,N,route);
                        if Index(t-tv+1) ==i
                            q0=min(qin(1), supercritical(ro(t,i),i,N,route));
                            q0 = min(q0/N, vveh(t-tv+1)*ro(t,i)/N) + (N-
1)*q0/N;
                        end
                        ro(t+1,i)=ro(t,i)+dt/dx*(q0-q(t,i)); vi =1;
                        v(t,i) = min(20.1168,(q(t,i))/(ro(t,i)));
                     else
                         [light,location] = trafficlightpresence(i-
1,Distance);
                         q0 = min(q(t,i-1) + qin(location+1),
supercritical(ro(t,i),i,N,route));
                         if Index(t-tv+1) ==i
                           q0 = min(q0/N, vveh(t-tv+1)*ro(t,i)/N) + (N-
1)*q0/N;
                         end
                         ro(t+1,i)=ro(t,i)+dt/dx*(q0-q(t,i));
                         v(t,i) = min(20.1168,(q(t,i))/(ro(t,i)));
                     end
                  else

q(t,i)=max(0,fundamental(ro(t,i),ro(t,i+1),ro_c,i,N,route));
                     if Index(t-tv+1) ==i
                        q(t,i) = min(q(t,i)/N, vveh(t-tv+1)*ro(t,i)/N) + (N-
1)*q(t,i)/N;
                     end
                     ro(t+1,i)=ro(t,i)+dt/dx*(q(t,i-1)-q(t,i));
                     v(t,i) = min(20.1168,(q(t,i))/(ro(t,i)));
                 end
            end
        end
    end
    end
    for t=tv:tv+Tmpc
        for i = 1:Nx-1
            ro_c = ro_critical(i,N,route);
            [light,location] = trafficlightpresence(i,Distance);
            if  light == 1
                qr(t,i) =
(fundamental(ror(t,i),ro_c,ro_c,i,N,route))*(spat(location,t)==1);
                ror(t+1,i)=ror(t,i)+dt/dx*(qr(t,i-1)- qr(t,i));
                vr(t,i) = min(20.1168,(qr(t,i))/(ror(t,i)));
```

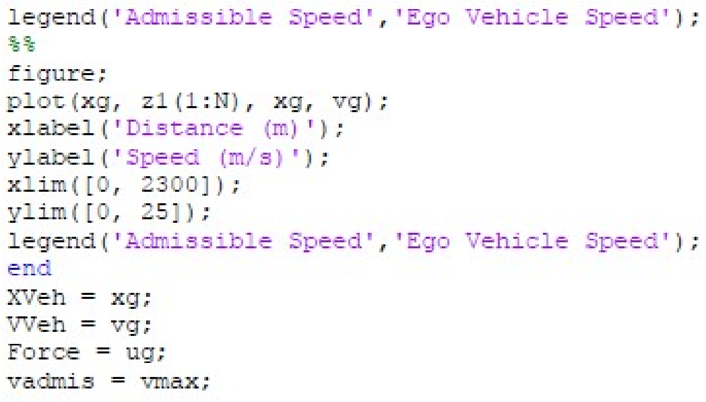
A.2 Velocity Optimization



```
%% Velocity Optimization
function [XVeh, VVeh, Force, vadmis] = optimization3(Xveh, Vveh,
Tractive_force, N_iter, Vadm, xdes)
global dt dx meff p1 p2 N a b c d g
dt =0.1;               % Time step
dx = 5;                % Cell Length
meff  = 1707;          % Effective Mass of Vehicle
g = 9.81;              % Acceleration due to Gravity
p1 = 15.91;            % Coastdown coefficient for resisting force
p2 = 150;              % Coastdown coefficient for resisting force
N = 2000;              % Number of time steps
a = 0.0078;          % Fuel Consumption Coefficient of Speed square
b = 4e-05;             % Fuel Consumption Coefficient of Cross-term
c =  9.99e-08;         % Fuel Consumption Coefficient of Tractive Force
square
d = 0.1569;
%% Create Quadratic Matrix
Q = zeros(2*N-2);
for i=1:2*N-2
    for j =1:2*N-2
        if i==j && i < N
            Q(i,j) = a;
        elseif i==j && i >= N
            Q(i,j) = c;
        end
    end
end
for i=1:2*N-2
    for j =1:2*N-2
        if i==j && i < N
            Q(i,j+N-1) = b/2;
        elseif i==j && i >= N
            Q(i,j-N+1) = b/2;
        end
    end
end
d = eig(Q);
isposdef = all(d >= 0);
%% Load data files
load('Vadm.mat');
load('grade.mat');
grade = grade';
distance = linspace(0,length(grade)*5,length(grade));
%% Set Weights
w1 = 1;
w2 = 75;
w3 = 50;
w4 = 5;
%% Guess iteration
xguess =  zeros(N,1);
vmax = zeros(N,1);
theta = zeros(N,1);
xf = 0;
for i=1:N
    xi = floor(xf/dx)+1;
    xguess(i) = xf;
    vmax(i) = Vadm(i+1000,xi);
    if xi > 826
       break;
    end
    xf = xf + vmax(i)*dt;
end
xdes = 0.8*xf;
for j =1:N
    xi = xguess(j);
    p = find(distance>=xi,1);
    theta(j) = grade(p);
end
cvx_begin
    variables x(N) fb(N-1) z(2*N-1) c1(N-1)
    minimize w1*quad_form(z(2:end), Q) + w2*norm(c1);
    subject to
    x(1) == 0;
    z(1) == 0;
    x(N) >= xdes;
    x(2:N) == x(1:N-1) + (dt/2)*(z(1:N-1) + z(2:N));
    z(2:N) == z(1:N-1) + (dt/meff)*(z(N+1:end) - fb - p1*z(1:N-1) -
p2*ones(N-1,1) - meff*g*sin(theta(2:end)));
    %0.90*vmax(2:N) <= z(2:N) + c1(1:end) <= vmax(2:N);
    z(2:N) + c1(1:end) == vmax(2:N);
    0 <= z(N+1:end) <= 5104;
    0 <= fb <= 5104;
    c1 >= 0;
    z(2:N) >=0;
cvx_end
xg = x;
vg = z(1:N);
ug = z(N+1:end) - fb;
%% Sequential Optimization
vmax = zeros(N,1);
theta = zeros(N,1);
for k=1:50
    for j = 1:N
        xi = floor((xg(j))/dx)+1;
        vmax(j) = Vadm(j + 1000,xi);
    end
    plot(vmax)
    hold on
    for j =1:N
        xi = xg(j);
        p = find(distance>=xi,1);
        theta(j) = grade(p);
    end
    cvx_begin
        variables x(N) fb(N-1) z(2*N-1) c1(N-1)
        minimize w1*quad_form(z(2:end), Q) + w2*norm(c1)  + w3*norm(z(1:N) -
vg) + w4*norm(z(N+2:end) - fb(2:end) - z(N+1:end-1) + fb(1:end-1));
        subject to
        x(1) == 0;
        z(1) == 0;
        x(N) >= xdes;
        x(2:N) == x(1:N-1) + (dt/2)*(z(1:N-1) + z(2:N));
        z(2:N) == z(1:N-1) + (dt/meff)*(z(N+1:end) - fb - p1*z(1:N-1) -
p2*ones(N-1,1) - meff*g*sin(theta(2:end)));
```

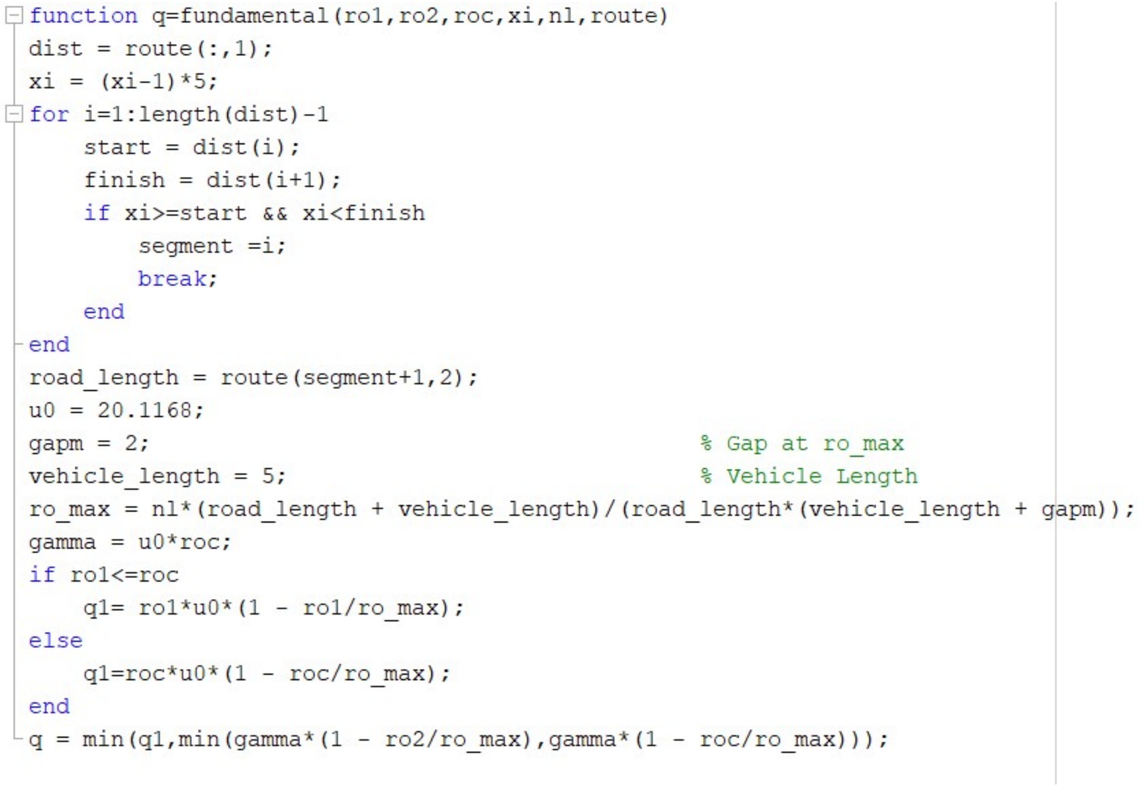


```
        %0.10*vmax(2:N) <= z(2:N) + c1(1:end)  <= vmax(2:N);
        z(2:N) + c1(1:end)  == vmax(2:N);
        0 <= z(N+1:end) <= 5104;
        0 <= fb <= 5104;
        c1 >= 0;
        z(2:N) >=0;
        x(2:N)>=0;
    cvx_end
%     cost = cvx_optval;
%     if k > 1
%         if cvx_optval > cost
           cost = cvx_optval;
           xg = x;
           vg = z(1:N);
           ug = z(N+1:end) - fb;
           plot(vg)
           hold off
%         else
%             break;
%         end
%     end
end
%% Plot
figure;
plot(xg, vmax, xg, vg);
xlabel('Distance (m)');
ylabel('Speed (m/s)');
legend('Admissible Speed','Ego Vehicle Speed');

figure;
plot(0:dt:(N-2)*dt, z(N+1:end), 0:dt:(N-2)*dt, fb);
xlabel('time (sec)');
ylabel('Force (N)');
%legend('Traction Force','Braking Force');

% plot(0:dt:(N-1)*dt, x(1:N))
%%
cvx_begin
    variables x1(N) fb1(N-1) z1(2*N-1)
    minimize norm(z1(2:N) - vmax(2:N));
    subject to
    x1(1) == 0;
    z1(1) == 0;
    x1(2:N) == x1(1:N-1) + (dt/2)*(z1(1:N-1) + z1(2:N));
    z1(2:N) == z1(1:N-1) + (dt/meff)*(z1(N+1:end) - fb1 - p1*z1(1:N-1) -
p2*ones(N-1,1) - meff*g*sin(theta(2:end)));
    0 <= z1(2:N)<= vmax(2:N);
    0 <= z1(N+1:end) <= 5104;
    0 <= fb1 <= 5104;
    z1(2:N) >=0;
cvx_end
%%
figure;
plot(0:dt:(N-1)*dt, z1(1:N), 0:dt:(N-1)*dt, vg);
xlabel('Distance (m)');
ylabel('Speed (m/s)');
ylim([0, 25]);
```



```
legend('Admissible Speed','Ego Vehicle Speed');
%%
figure;
plot(xg, z1(1:N), xg, vg);
xlabel('Distance (m)');
ylabel('Speed (m/s)');
xlim([0, 2300]);
ylim([0, 25]);
legend('Admissible Speed','Ego Vehicle Speed');
end
XVeh = xg;
VVeh = vg;
Force = ug;
vadmis = vmax;
```

A.3 Fundamental Diagram



```
function q=fundamental(ro1,ro2,roc,xi,nl,route)
dist = route(:,1);
xi = (xi-1)*5;
for i=1:length(dist)-1
    start = dist(i);
    finish = dist(i+1);
    if xi>=start && xi<finish
        segment =i;
        break;
    end
end
road_length = route(segment+1,2);
u0 = 20.1168;
gapm = 2;                                         % Gap at ro_max
vehicle_length = 5;                               % Vehicle Length
ro_max = nl*(road_length + vehicle_length)/(road_length*(vehicle_length + gapm));
gamma = u0*roc;
if ro1<=roc
    q1= ro1*u0*(1 - ro1/ro_max);
else
    q1=roc*u0*(1 - roc/ro_max);
end
q = min(q1,min(gamma*(1 - ro2/ro_max),gamma*(1 - roc/ro_max)));
```

Bibliography

[1] Federal Highway Administration Adm. Adaptive Signal Control Technology.

[2] Behrang Asadi and Ardalan Vahidi. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE Transactions on Control Systems Technology*, 19(3):707–714, 2011.

[3] V. Bagyaveereswaran, Tushar D. Mathur, Sukrit Gupta, and P. Arulmozhivarman. Performance comparison of next generation controller and mpc in real time for a siso process with low cost daq unit. *Alexandria Engineering Journal*, 55(3):2515–2524, 2016.

[4] Michael Burger, Simone Göttlich, and Thomas Jung. Derivation of a first order traffic flow model of Lighthill-Whitham-Richards type. *IFAC-PapersOnLine*, 51(9):49–54, 2018.

[5] Carlos F. Daganzo. Requiem for second-order fluid approximations of traffic flow. *Transportation Research Part B*, 29(4):277–286, 1995.

[6] State of California Department of Motor Vehicles. Autonomous Vehicle Collision Reports. https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/autonomous-vehicle-collision-reports/.

[7] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[8] Wenjian Hao. TigerPrints Data-Driven Control with Learned Dynamics Data Driven Control with Learned Dynamics. (August), 2020.

[9] Erik Hellström, Maria Ivarsson, Jan Åslund, and Lars Nielsen. Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control Engineering Practice*, 17(2):245–254, 2009.

[10] Serge Hoogendoorn and P. Bovy. State-of-the-art of vehicular traffic flow modeling. *J. Syst. Cont. Eng.*, 215:283–303, 06 2001.

[11] Xianan Huang and Huei Peng. Speed trajectory planning at signalized intersections using sequential convex optimization. *arXiv*, pages 2992–2997, 2017.

[12] M. A.S. Kamal, Masakazu Mukai, Junichi Murata, and Taketoshi Kawabe. Ecological vehicle control on roads with up-down slopes. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):783–794, 2011.

[13] J. Lardoux, R. Martinez, C. White, N. Gross, N. Patel, and R. Meyer. Adaptive Traffic Signal Control for Tarrytown Road in White Plains. (14), 2014.

[14] Hansang Lim, Wencong Su, and Chunting Chris Mi. Distance-Based Ecological Driving Scheme Using a Two-Stage Hierarchy for Long-Term Optimization and Short-Term Adaptation. *IEEE Transactions on Vehicular Technology*, 66(3):1940–1949, 2017.

[15] National Transportation Operations Coalition and National Traffic Operations Coalition. 2012 National Traffic Signal Report Card Technical Report. *National Transportation Operations Coalition (NTOC)*, 2012.

[16] Engin Ozatay, Simona Onori, James Wollaeger, Umit Ozguner, Giorgio Rizzoni, Dimitar Filev, John Michelini, and Stefano Di Cairano. Cloud-based velocity profile optimization for everyday driving: A dynamic-programming-based solution. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2491–2505, 2014.

[17] David Schrank, Bill Eisele, and Tim Lomax. 2019 Urban Mobility Report. Technical report, Texas AM Transportation Institute, 2019.

[18] John Schulman, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization. 2016.

[19] Lieven Vandenberghe Stephen Boyd. *Convex Optimization.* 2004.

[20] Eric Stump and Darren Myer. Adaptive Traffic Signals Reduce Delay, Increase Safety, and Improve Public Satisfaction, 2017.

[21] Chao Sun, Jacopo Guanetti, Francesco Borrelli, and Scott J. Moura. Optimal Eco-Driving Control of Connected and Autonomous Vehicles Through Signalized Intersections. *IEEE Internet of Things Journal*, 7(5):3759–3773, may 2020.

[22] John Thomas, Shean Huff, Brian West, and Paul Chambon. Fuel Consumption Sensitivity of Conventional and Hybrid Electric Light-Duty Gasoline Vehicles to Driving Style. *SAE International Journal of Fuels and Lubricants*, 10(3):672–689, 2017.