Fuel-minimizing vehicle trajectory specification in the presence of traffic lights with certain or stochastic switching times

Dynamic Systems and Simulation Laboratory

Department of Production Engineering and Management

Technical University of Crete

Diploma Thesis

Supervisor:
Prof. Markos Papageorgiou

Author:
Kalogianni Ioanna

Abstract

Driving style of road vehicles has a significant impact on the fuel economy, hence the recent term eco-driving to denote a driving style that reduces fuel consumption. In the last decades, autonomous driving, but also vehicular communications are becoming more and more important. One application of vehicle connectivity is to receive information about the next signal switching time when vehicles approach a traffic light. Based on this information, appropriately developed systems (or apps), known as GLOSA (Green Light Optimal Speed Adaptation), compute a fuel-efficient velocity profile for the vehicle to cross the traffic lights, e.g. without stopping. The main purpose of this thesis is to generate optimal (fuel-minimising) trajectories for vehicles crossing a signalized junction, with traffic signals operated in either fixed-time or real-time mode. In the case of fixed-time signals, the next switching time is known beforehand; in real-time signals, the next switching time is decided in real time based on the prevailing traffic conditions and is therefore uncertain in advance. This thesis approaches the problem by using traffic lights' information and calculating a trajectory and a velocity profile for the vehicle, based on the vehicle's initial state (position and speed) and a fixed final destination state (downstream of the junction). In the case of fixed signals, an appropriate optimal control problem is formulated and solved analytically via the Pontryagin's Minimum Principle (PMP). In the case of real-time signals, availability of a time-window of possible signal switching times, along with the corresponding probability distribution, is assumed, and the problem is cast in the format of a stochastic optimal control problem and is solved numerically using Stochastic Dynamic Programming techniques.

Acknowledgements

Herewith, I would like to thank Dr. Mountakis Kyriakos and Typaldos Panagiotis for their guidance and contribution to the implementation of the research and results presented in this paper. Also, I'd like to express my gratitude to Professor Markos Papageorgiou and Professor Papamichail Ioannis for the opportunity they gave me, to work on such an interesting subject. Finally, I would like to thank my parents, my friends and especially my sister and my partner Antonis for the support they have provided me both during my studies and in the completion of this thesis.

Ioanna Kalogianni, Chania, June 2018

Table of Contents

List of Figures

List of Tables

1 Introduction

Transportation fuel consumption is a critical problem for our society. Driving style and traffic conditions have been proved to have a significant impact on the fuel economy and eco-driving. Both car manufacturers and transportation researchers are trying to use new technologies, such as cylinder deactivation, start-stop system, in order to reduce fuel consumption. The increasing need for traffic efficiency led to the advent of cooperative (ITS), which is one of the effective solutions to critical transportation problems. ITS includes telematics and all kinds of communication in vehicles, between vehicles and between infrastructure and vehicles [1].

Traffic lights (TL) coordinate the traffic for most of the city intersections. A busy intersection in a typical urban area might coordinate the movement of thousands of vehicles a day. In this context, the accumulated energy waste due to stopping at red light and the corresponding generated emissions become crucial. Therefore, being able to fluently cross the Traffic Lights is of great value. A Traffic Light Assistant (TLA) would therefore contribute to improve the overall transportation system energy efficiency and, for the case of a fully electric vehicle, to the all-electric range extension.

Greenlight optimal speed advisory (GLOSA) is one of the important applications to reduce traffic delay and fuel consumption in ITS. The GLOSA application provides the driver with accurate speed advice about the approaching intersection by taking advantage of traffic light timing and vehicle positions. The aim of this application is to guide the vehicle going through signaled intersections with a more appropriate speed in order to reduce stopped time and save fuel. Drivers need to adjust their speed in order to pass the intersection without stopping as they are approaching the intersection.

The known GLOSA implementations use data from statically behaved traffic lights where the traffic lights' times are constant. Nowadays most signals are dynamically managed, which means that the green and red times are influenced by the corresponding demands at the traffic lights. The green and red times always have a minimum and maximum value. When the traffic intensity increases, most traffic lights are specified such that the green times will increase and the system can serve more vehicles.

This thesis aims to implement a GLOSA system in order to approach both statically and dynamically managed traffic lights, by using traffic lights' information in order to calculate a trajectory and a velocity profile, based on the initial and the desired final destination point. The optimal velocity profile is advised to the driver or can be used directly in autonomous vehicles. Therefore, for static information, a solution to the Optimal Control Problem, based on the Pontryagin's maximum principle (PMP) is developed, and in order to approach dynamically managed traffic lights, the algorithm of Stochastic Dynamic Programming has been implemented.

1.1 Vehicle Connectivity

The rapid advancement of vehicle technology is dramatically altering transportation models around the world. From early stage consumer infotainment features, to ride sharing and on-demand mobility services, to fully autonomous vehicles in the future, connectivity in the car has been the driving force behind recent automotive technology advancements. As a result, vehicles have morphed into much more than just a way to get from one place to another, but extensions of consumer digital lifestyles and a catalyst for significant change in the way society will experience future mobility.

There are 5 ways a vehicle can be connected to its surroundings and communicate with them:

- (V2I) "Vehicle to Infrastructure": The technology captures data generated by the vehicle and provides information about the infrastructure to the driver. The V2I technology communicates information about safety, mobility or environment-related conditions. [2]
- (V2V) "Vehicle to Vehicle": The technology communicates information about speed and position of surrounding vehicles through a wireless exchange of information. The goal is to avoid accidents, ease traffic congestion and have a positive impact on the environment. [3]
- (V2C) "Vehicle to Cloud": The technology exchanges information about and for applications of the vehicle with a cloud system. This allows the vehicle to use information from other, though the cloud connected industries like energy, transportation and smart homes and make use of IoT.[4]

- (V2P) "Vehicle to Pedestrian": The technology senses information about its environment and communicates it to other vehicles, infrastructure and personal mobile devices. This enables the vehicle to communicate with pedestrians and is intended to improve safety and mobility on the road. [5]

- (V2X) "Vehicle to Everything": The technology interconnects all types of vehicles and infrastructure systems with another. This connectivity includes cars, highways, ships, trains and airplanes. [6]

1.2 Green Light Optimal Speed Advisory GLOSA

Intelligent traffic lights are believed to play an important role in tomorrow's transportation system as they are a major factor in the optimization of traffic flows. Dynamic traffic light programs can adapt to current traffic in order to lower waiting times and increase traffic throughput [7]. Equipped with communication devices, traffic lights could also inform approaching vehicles of the current traffic light program to further reduce the amount of stops and starts in order to decrease CO_2 emissions and fuel consumption [1]. As shown in Figure 1, this information can be used by an approaching driver to alter his original trajectory through the use of certain driving maneuvers to avoid having to stop at the red light but arrive shortly after the signal turns green. [8]

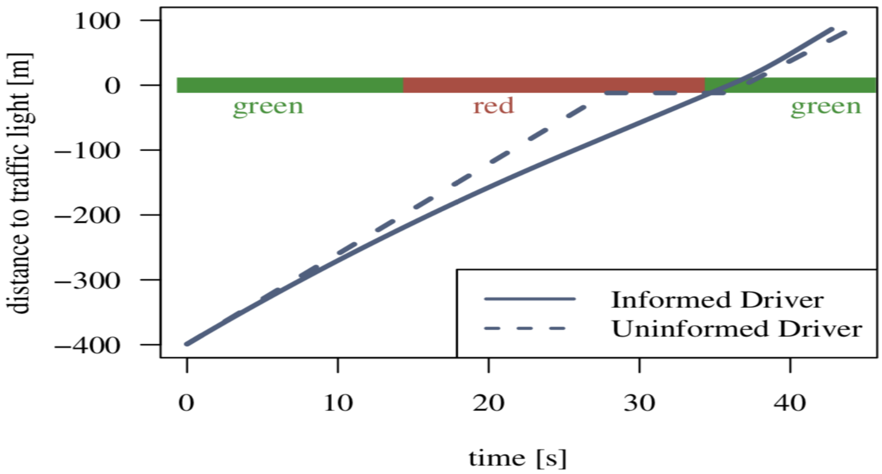


Figure 1 Trajectories of informed and uninformed drivers. The uninformed driver has to stop at the red light, while the informed driver arrives when the signal turns green.

GLOSA is a system that uses data in order to calculate an optimal speed at which the vehicle could probably pass the next traffic light without stopping. When a vehicle approaches a traffic light, it receives information regarding the location of the intersection and the signal phase and timing. With this information and its own position the vehicle can calculate a speed advice. The vehicle could either notify the driver of the optimal speed or notify the autonomous car system directly.

The aim of this system is to prevent a stop-and-go situation, reduce the fuel consumption and CO2 emission and thus lead to a more efficient infrastructure.Research shows that it can save up to 20 % of fuel and 17% of stop time [1]. In addition to lower fuel consumption, GLOSA systems can provide higher travel comfort, have lower environmental impact, and the traffic flow is smoothed and increased.Furthermore, obtaining more and more information about handling intersection can, finaly, lead to better road safety.

The recent years there have been several successful realizations of GLOSA systems in a real world environment. Most of these projects were implemented as proofs of concept, usually to demonstrate abilities of the cooperative ITS technology and its potential for the future. Most of them are also part of ongoing research activities of major car manufacturers.

Audi Travolution is a collaboration between Audi and many of its partners. One part of this project is a GLOSA system deployed at 25 intersections in Ingolstadt, Germany. Fifteen of these intersections use car-to-infrastructure wireless local area networks (WLAN) allowing for a direct communication between cars and the intersections. The remaining intersections send data to a back end server located in Ingolstadt city centre. Besides displaying a recommended speed to the driver, two of the test cars have been equipped with an adaptive cruise control connected to the GLOSA system. Another large GLOSA implementation has been carried out by Swarco and Audi in Berlin and it involves over 800 intersections. However, they do not use any form of a direct V2I communication. An interesting additional feature of this implementation is a connection with the start-stop system in Audi vehicles. When the time to green is below a predefined threshold, the engine is not stopped. Another project that offers GLOSA capabilities is Signal Guru, a smart phone app of the Massachusetts Institute of Technology (MIT). The smart phone needs to be positioned directly behind the windshield and the app uses the smart phone camera to collect data of passed traffic lights. This data will be saved in a database and is used to determine the traffic light signal phases. When the system has collected sufficient data the app can display the remaining red time and an optimal speed to pass green lights.

2 Optimal Control Theory

Optimal Control theory is an extension of the calculus of variations, and is a mathematical optimization method for deriving control policies. The method is largely due to the work of Lev Pontryagin and Richard Bellman in the 1950s, after contributions to calculus of variations by Edward J. McShane. Optimal control theory deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved. A control problem includes a cost functional that is a function of state and control variables. An optimal control is a set of differential equations describing the paths of the control variables that minimize the cost function. The optimal control can be derived using (PMP), or by solving the Hamilton–Jacobi–Bellman equation as a sufficient condition.

2.1 Continuous-time Dynamic Systems

A continuous-time dynamic system is described by an n-dimensional state vector $x(t)$ at time t. Choice of an m-dimensional control vector $u(t)$ determines the time rate of change of the state through the relations

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t) \tag{2.1.1}$$

A general optimization problem for such a system is to find the optimal control vector $\boldsymbol{u}(t)$ for $t_0 \leq t \leq t_e$, in order to minimize a performance index in the following form:

$$J = \theta(\boldsymbol{x}(t_e), t_e) + \int_{t_0}^{t_e} \phi[\boldsymbol{x}(t), \boldsymbol{u}(t), t]dt \tag{2.1.2}$$

where $[t_0, t_e]$ is the time interval of interest. The final weighting function $\theta(x(t_e), t_e)$ depends on the final state and final time, and the weighting function $\phi[x(t), u(t), t]$ depends on the state and input at intermediate times in $[t_0, t_e]$.

Satisfying the boundary conditions given by:

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{2.1.3}$$

$$g[\boldsymbol{x}(t_e), t_e] = 0 \tag{2.1.4}$$

Optimality Conditions

For the continuous time optimal control problem we define the Hamiltonian function as: [9]

$$H(\boldsymbol{x}, \boldsymbol{u}, t) = L(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\lambda}^T f(\boldsymbol{x}, \boldsymbol{u}, t) \tag{2.1.5}$$

where $\boldsymbol{\lambda} \in R^n$ are Lagrange multipliers, also known as co-states.
Then, the Pontryagin's principle provides the following 1^{st} order necessary conditions for optimality [9]:

$$\dot{\boldsymbol{x}} = \frac{\vartheta H(\boldsymbol{x}, \boldsymbol{u}, t)}{\vartheta \boldsymbol{\lambda}} \tag{2.1.6}$$

$$\dot{\boldsymbol{\lambda}} = \frac{\vartheta H(\boldsymbol{x}, \boldsymbol{u}, t)}{\vartheta \boldsymbol{x}} \tag{2.1.7}$$

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} H(\boldsymbol{x}, \boldsymbol{u}, t) \tag{2.1.8}$$

Furthermore, the following boundary and transversality conditions must be met:

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{2.1.9}$$

$$g[\boldsymbol{x}(t_e), t_e] = 0 \tag{2.1.10}$$

$$\boldsymbol{\lambda}(t_e) = \theta_{\boldsymbol{x}(t_e)} + {g_{\boldsymbol{x}(t_e)}}^T \nu \tag{2.1.11}$$

2.1.1 Free Final Time

In the case with a free final time t_e, we wish to find the optimal control vector $u(t)$ for $t_0 \leq t \leq t_e$ and the final time t_e, in order to minimize the following performance index:

$$J = \theta(\boldsymbol{x}(t_e), t_e) + \int_{t_0}^{t_e} \phi[\boldsymbol{x}(t), \boldsymbol{u}(t), t] dt \tag{2.1.12}$$

The new element is the dependence of the performance index, the terminal constraints and the dynamic equation on the final time t_e. The following necessary condition is called transversality condition and determines the value of the final time t_e.

$$H[\boldsymbol{x}(t_e), \boldsymbol{u}(t_e), \boldsymbol{\lambda}(t_e), t_e] + \theta_{t_e} = 0 \tag{2.1.13}$$

Eq. 2.1.13 with the initial and final conditions give us the following boundary conditions, which can be used to solve the two-point boundary-value problem and can be used to calculate the free end time t_e.

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{2.1.14}$$

$$g[\boldsymbol{x}(t_e), t_e] = 0 \tag{2.1.15}$$

$$\boldsymbol{\lambda}(t_e) = \theta_{\boldsymbol{x}(t_e)} + g^T_{\boldsymbol{x}(t_e)}\nu \tag{2.1.16}$$

$$H(\boldsymbol{x}(t_e), \boldsymbol{u}(t_e), (t_e), t_e) + \theta_{t_e} = 0 \tag{2.1.17}$$

2.1.2 Interior Point Constraints

In this problem it is required that position satisfies equality constraints at some point t_1, where $t_0 \leq t_1 \leq t_e$. This is known as interior point constraint and can be expressed as:

$$N[x(t_1)] = 0 \tag{2.1.18}$$

where t_1 is an intermediate time and N is a q-component vector function. We now have a three-point boundary-value problem instead of two-point boundary-value problem. Now eq. 2.1.18 represents a set of terminal constraints for $t \in [t_0, t_1]$
If we let t_{1^-} signify just before t_1, and t_{1^+} just after, we can derive the following:

$$\lambda^T(t_1^-) = \lambda^T(t_1^+) + \pi^T \frac{\vartheta N}{\vartheta x(t_1)} \tag{2.1.19}$$

$$H(t_1^-) = H(t_1^+) + \pi^T \frac{\vartheta N}{\vartheta t_1} \tag{2.1.20}$$

We then have the following boundary conditions, which can be used to solve the three-point boundary-value problem.
For $t \in [0, t_1]$:

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{2.1.21}$$

$$N[x(t_1)] = 0 \tag{2.1.22}$$

$$\boldsymbol{\lambda}(t_1) = \theta_{\boldsymbol{x}(t_1)} + N^T_{\boldsymbol{x}(t_1)}\pi \tag{2.1.23}$$

For $t \in [t_1, t_e]$

$$N[x(t_1)] = 0 \tag{2.1.24}$$

$$g[\boldsymbol{x}(t_e)] = 0 \tag{2.1.25}$$

$$\boldsymbol{\lambda}(t_e) = \theta_{\boldsymbol{x}(t_e)} + g^T_{\boldsymbol{x}(t_e)}\nu \tag{2.1.26}$$

2.2 Discrete-time Dynamic Systems

A discrete-time system is described by an n-dimensional state vector $x(k)$ at step k. Choice of an m-dimensional control vector $u(k)$ determines a transition of the system to state $x(k+1)$ through the relation:

$$\boldsymbol{x}(k+1) = f[\boldsymbol{x}(k), \boldsymbol{u}(k), k], k = 0, ..., K-1 \tag{2.2.1}$$

where $\theta, \phi \in R^q$ are twice continuous differential functions.The problem consists of minimizing the discrete-time cost function [**?**]

$$J[\boldsymbol{x}(k), \boldsymbol{u}(k), k] = \theta(\boldsymbol{x}(K)) + \sum_{k=0}^{K-1} \phi[\boldsymbol{x}(k), \boldsymbol{u}(k), k] \tag{2.2.2}$$

The final state may be free or may be required to satisfy a final condition:

$$g[\boldsymbol{x}(K)] = 0 \tag{2.2.3}$$

Although expressing a dynamic physical procedure, the above formulated problem is, from a mathematical point of view, a static optimization problem due to the discrete-time nature of the involved process model. To see this, define the vectors:

$$\boldsymbol{X} = [\boldsymbol{x}(1)^T \boldsymbol{x}(2)^T ... \boldsymbol{x}(K)^T]^T \tag{2.2.4}$$

$$\boldsymbol{U} = [\boldsymbol{u}(0)^T \boldsymbol{u}(1)^T ... \boldsymbol{u}(K-1)^T]^T \tag{2.2.5}$$

The discrete optimal control may then be expressed as follows:

$$\text{Minimize } \Phi(\boldsymbol{X}, \boldsymbol{U})$$
$$\text{subject to } F(\boldsymbol{X}, \boldsymbol{U}) = 0$$

where Φ expresses the discrete-time cost function, and F the state equations for all $k \in [0, K-1]$ and the terminal condition.

Optimality Conditions

In order to derive the necessary conditions of optimality for the discrete-time optimal control problem we use the following Lagrangian function.

$$\begin{aligned} L[\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\lambda}(k), k] &= \Phi(\boldsymbol{X}, \boldsymbol{U}) + \boldsymbol{\Lambda}^T F(\boldsymbol{X}, \boldsymbol{U}) = \\ &= \theta[\boldsymbol{x}(K)] + \phi[\boldsymbol{x}(k_1)] + \sum_{k=0}^{K-1} \phi[\boldsymbol{x}(k), \boldsymbol{u}(k), k] + \\ &+ \sum_{k=0}^{K-1} \{\boldsymbol{\lambda}(k+1)^T [f[\boldsymbol{x}(k), \boldsymbol{u}(k), k] - \boldsymbol{x}(k+1)]\} + \nu^T g[\boldsymbol{x}(K)] \end{aligned}$$

where $\boldsymbol{\lambda}(k+1) \in R^n$, k=0,...,K-1, is Lagrange multiplier for the equality conditions. The multipliers ν, π are assigned to the final and the intermediate condition respectively.

Applying the necessary conditions of optimality [3]:

$$\frac{dL}{d\boldsymbol{X}} = 0 \Rightarrow \boldsymbol{\lambda}(k) = \phi_{\boldsymbol{x}(k)} + f^T_{\boldsymbol{x}(k)} \boldsymbol{\lambda}(k+1) \tag{2.2.6}$$

$$\frac{dL}{d\boldsymbol{U}} = 0 \Rightarrow \phi_{\boldsymbol{u}(k)} + f^T_{\boldsymbol{u}(k)} \lambda(k+1) = 0 \tag{2.2.7}$$

$$\frac{dL}{d\boldsymbol{\Lambda}} = 0 \Rightarrow \boldsymbol{x}(k+1) = f[\boldsymbol{x}(k), \boldsymbol{u}(k), k] \tag{2.2.8}$$

We derive the necessary conditions of optimality for the discrete-time optimal control problem. These conditions are expressed in terms of the discrete-time Hamiltonian function that is defined as follows.

$$H[\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\lambda}(k+1), k] = \phi[\boldsymbol{x}(k), \boldsymbol{u}(k), k] + \boldsymbol{\lambda}(k+1)^T f[\boldsymbol{x}(k), u(k), k] \tag{2.2.9}$$

We, then, have the following necessary conditions of optimality for the discrete-time optimal control problem:

$$\boldsymbol{\lambda}(k) = H_{\boldsymbol{x(k)}} = \phi_{\boldsymbol{x}(k)} + f_{\boldsymbol{x}(k)}^T \boldsymbol{\lambda}(k+1) \tag{2.2.10}$$

$$H_{\boldsymbol{x}(k)} = \phi_{u(k)} + f_{\boldsymbol{u}(k)}^T \boldsymbol{\lambda}(k+1) = 0 \tag{2.2.11}$$

$$\boldsymbol{x}(k+1) = f[\boldsymbol{x}(k), \boldsymbol{u}(k), k] \tag{2.2.12}$$

Moreover the following boundary and transversality conditions must be satisfied.

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{2.2.13}$$

$$\boldsymbol{\lambda}(K) = \theta_{\boldsymbol{x}(K)} + g_{\boldsymbol{x}(K)}^T \nu \tag{2.2.14}$$

$$g[\boldsymbol{x}(K)] = 0 \tag{2.2.15}$$

2.2.1 Free Final Time

This extension of the optimal control problem, where the final time is free, is simply to regard the final time as a parameter to be optimized, in addition to control. We aim to find the control vector sequence u(k), k=0,...,K-1 and the final free time t_e to minimize:

$$J[\boldsymbol{x}(k), \boldsymbol{u}(k), k] = \theta(\boldsymbol{x}(K)) + \sum_{k=0}^{K-1} \phi[\boldsymbol{x}(k), \boldsymbol{u}(k), k] \tag{2.2.16}$$

where the number of steps K is specified and

$$t_e = K\Delta t \tag{2.2.17}$$

subject to the constraints:

$$g[\boldsymbol{x}(K), t_e] = 0 \tag{2.2.18}$$

$$\boldsymbol{x}(k+1) = f[\boldsymbol{x}(k), \boldsymbol{u}(k), k, \Delta t] \tag{2.2.19}$$

$$\boldsymbol{x}(x_0) = \boldsymbol{x}_0 \tag{2.2.20}$$

where Δt is the timestep.

Optimality conditions

The necessary conditions can be obtained by the extension of the Lagrangian function [10]

$$L[\boldsymbol{x}(k),\boldsymbol{u}(k),\boldsymbol{\lambda}(k),k] = \Phi(\boldsymbol{X},U) + \boldsymbol{\Lambda}^T F(\boldsymbol{X},U) = \sum_{k=0}^{K-1}\phi[\boldsymbol{x}(k),u(k),k,\Delta t]+$$
$$+\sum_{k=0}^{K-1}\{\boldsymbol{\lambda}(k+1)^T[f[\boldsymbol{x}(k),\boldsymbol{u}(k),k,\Delta t]-\boldsymbol{x}(k+1)]\}+\nu^T g[\boldsymbol{x}(K)]$$

Using the Lagrangian function we derive the following necessary conditions of optimality for the discrete-time optimal control problem with final free time.

$$\boldsymbol{\lambda}(k) = H_{\boldsymbol{x}(k)} = \phi_{x(k)} + f^T_{\boldsymbol{x}(k)}\boldsymbol{\lambda}(k+1) \tag{2.2.21}$$
$$H_{u(k)} = \phi_{\boldsymbol{u}(k)} + f^T_{\boldsymbol{u}(k)}\boldsymbol{\lambda}(k+1) = 0 \tag{2.2.22}$$
$$\boldsymbol{x}(k+1) = f[\boldsymbol{x}(k),\boldsymbol{u}(k),k,\Delta t] \tag{2.2.23}$$

Moreover the following boundary and transversality conditions must be satisfied

$$\Theta_{\Delta t} + \sum_{k=0}^{K-1} H_{\Delta t}(k) = 0 \tag{2.2.24}$$
$$\boldsymbol{\lambda}(K) = \theta_{\boldsymbol{x}(K)} + g^T_{\boldsymbol{x}(K)}\nu \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{2.2.25}$$

$$\text{where} \quad \Theta = \theta[\boldsymbol{x}(K),t_e] + \nu^T g[\boldsymbol{x}(K),t_e] \tag{2.2.26}$$

is new necessary condition is a transversality condition that determines the optimal time step Δt.

2.2.2 Interior Point Constraints

In this problem it is required that the states variables atisfies equality constraints at some point k_1 (the position of the traffic light), where $0 \leq k_1 \leq K$. This is known as interior point constraint and can be expressed as:

$$N[x(k_1)] = 0 \tag{2.2.27}$$

where k_1 is an intermediate point and N is a q-component vector function. We now have a three-point boundary-value problem instead of two-point boundary-value problem. Now eq. 2.2.27 represents a set of terminal constraints for $k \in [0, k_1]$. The discrete-time cost function is transformed as follows:

$$\begin{aligned} J[\boldsymbol{x}(k), \boldsymbol{u}(k), k] = \theta(\boldsymbol{x}(K)) + \phi[\boldsymbol{x}(k_1)] + \\ + \sum_{k=0}^{K-1} \phi[\boldsymbol{x}(k), \boldsymbol{u}(k), k] \end{aligned} \tag{2.2.28}$$

In order to derive the necessary conditions of optimality for the discrete-time optimal control problem we formulate the Lagrangian function

$$\begin{aligned} L[\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\lambda}(k), k] = \Phi(\boldsymbol{X}, \boldsymbol{U}) + \boldsymbol{\Lambda}^T F(\boldsymbol{X}, \boldsymbol{U}) = \theta[\boldsymbol{x}(K)] + \phi[\boldsymbol{x}(k_1)] + \\ + \sum_{k=0}^{K-1} \phi[\boldsymbol{x}(k), \boldsymbol{u}(k), k] + \sum_{k=0}^{K-1} \{\boldsymbol{\lambda}(k+1)^T [f[\boldsymbol{x}(k), \boldsymbol{u}(k), k] - \boldsymbol{x}(k+1)]\} + \\ \nu^T g[\boldsymbol{x}(K)] + \pi^T N[x(k_1)] \end{aligned} \tag{2.2.29}$$

where $\boldsymbol{\lambda}(k+1) \in R^n$, k=0,...,K-1, is Lagrange multiplier for the equality conditions. The multipliers ν, π are assigned to the final and the intermediate condition respectively.

Applying the necessary conditions of optimality:

$$\frac{dL}{d\boldsymbol{X}} = 0 \Rightarrow \boldsymbol{\lambda}(k) = \phi_{\boldsymbol{x}(k)} + f^T_{\boldsymbol{x}(k)}\boldsymbol{\lambda}(k+1) \tag{2.2.30}$$

$$\frac{dL}{d\boldsymbol{U}} = 0 \Rightarrow \phi_{\boldsymbol{u}(k)} + f^T_{\boldsymbol{u}(k)}\lambda(k+1) = 0 \tag{2.2.31}$$

$$\frac{dL}{d\boldsymbol{\Lambda}} = 0 \Rightarrow \boldsymbol{x}(k+1) = f[\boldsymbol{x}(k), \boldsymbol{u}(k), k] \tag{2.2.32}$$

We derive the necessary conditions of optimality for the discrete-time optimal control problem. These conditions are expressed in terms of the discrete-time Hamiltonian function that is defined as follows.

$$H[\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{\lambda}(k+1), k] = \phi[\boldsymbol{x}(k), \boldsymbol{u}(k), k] + \boldsymbol{\lambda}(k+1)^T f[\boldsymbol{x}(k), \boldsymbol{u}(k), k] \tag{2.2.33}$$

We, then, have the following necessary conditions of optimality for the discrete-time optimal control problem:

$$\boldsymbol{\lambda}(k) = H_{\boldsymbol{x}(k)} = \phi_{\boldsymbol{x}(k)} + f^T_{\boldsymbol{x}(k)}\boldsymbol{\lambda}(k+1) \tag{2.2.34}$$

$$H_{\boldsymbol{x}(k)} = \phi_{\boldsymbol{u}(k)} + f^T_{\boldsymbol{u}(k)}\boldsymbol{\lambda}(k+1) = 0 \tag{2.2.35}$$

$$\boldsymbol{x}(k+1) = f[\boldsymbol{x}(k), \boldsymbol{u}(k), k] \tag{2.2.36}$$

Moreover the following boundary and transversality conditions must be satisfied.

For $k \in [0, k_1]$:

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{2.2.37}$$

$$N[x(k_1)] = 0 \tag{2.2.38}$$

$$\lambda(k_1) = \vartheta_{x(k_1)} + N^T_{x(k_1)}\pi \tag{2.2.39}$$

For $k \in [k_1, K]$

$$N[x(k_1)] = 0 \tag{2.2.40}$$

$$\boldsymbol{\lambda}(K) = \vartheta_{\boldsymbol{x}(K)} + g^T_{\boldsymbol{x}(K)}\nu \tag{2.2.41}$$

$$g[\boldsymbol{x}(K)] = 0 \tag{2.2.42}$$

The above three-point boundary-value problem can be extended as a problem with multiple interior point constraints. Then the general form of the Lagrangian function can be expressed as:

$$L[\boldsymbol{x}(k), u(k), \boldsymbol{\lambda}(k), k] = \Phi(\boldsymbol{X}, \boldsymbol{U}) + \boldsymbol{\Lambda}^T F(\boldsymbol{X}, \boldsymbol{U}) = \sum_{k=0}^{K-1} \phi[\boldsymbol{x}(k), \boldsymbol{u}(k), k] +$$
$$+ \sum_{k=0}^{K-1} \{\boldsymbol{\lambda}(k+1)^T [f[\boldsymbol{x}(k), \boldsymbol{u}(k), k] - \boldsymbol{x}(k+1)]\} + \nu^T g[\boldsymbol{x}(K)] + \sum_{i=1}^{M} \pi^T N[x(k_i)]$$

where $g[\boldsymbol{x}(K)]$ is the terminal boundary condition, $N_i[x(k_i)]$ is the i^{th} interior point constraint, and M in the total number of interior point constraints.

3 Dynamic Programming

Dynamic programming (DP) is a very different approach to solve optimal control problems than the ones presented previously. The methodology was developed in 1950s, most prominently by Richard Bellman who also coined the term dynamic programming. In his work, Bellman writes "An optimal policy has the property that whatever the initial state and the initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision". This is known as Bellman's Principle and is the central idea to dynamic programming. Interestingly, dynamic programming is easiest to apply to systems with discrete state and control spaces. When DP is applied to discrete time systems with continuous state spaces, some approximations have to be made, usually by discretization. Generally, this discretization leads to exponential growth of computational cost with respect to the dimension n_x of the state space, what Bellman called the "curse of dimensionality". In the continuous time case, DP is formulated as a partial differential equation in the state space, the Hamilton-Jacobi-Bellman (HJB) equation, suffering from the same limitation. On the positive side, DP can easily deal with all kinds of hybrid systems or non-differential dynamics, and it even allows us to treat stochastic optimal control with recourse. This chapter is an attempt to present a brief overview of the most important concepts and ideas in the Bellman theory of DP. Our emphasis is to review the basic concepts and definitions in DP and to show the many sources of difficulty in obtaining the optimal solution.

3.1 Dynamic Programming Fundamentals

A discrete-time system is described by an n-dimensional state vector x(k) at step k. Choice of an m-dimensional control vector u(k) determines a transition of the system to state x(k+1) through the relation:

$$\boldsymbol{x}(k+1) = f[\boldsymbol{x}(k), \boldsymbol{u}(k), k], k = 0, ..., K-1 \tag{3.1.1}$$

The problem consists of minimizing the following discrete-time cost function [11].

$$J[\boldsymbol{x}(k), \boldsymbol{u}(k), k] = \theta(x(K)) + \sum_{k=0}^{K-1} \phi[\boldsymbol{x}(k), \boldsymbol{u}(k), k] \tag{3.1.2}$$

The final states may be free or may be required to satisfy a final condition

$$g[\boldsymbol{x}(K)] = 0 \tag{3.1.3}$$

The permissible control area is defined as:

$$\boldsymbol{u}(k) \in \boldsymbol{U}[\boldsymbol{x}(k), k] = \{\boldsymbol{u}(k) | h^*[\boldsymbol{x}(k), \boldsymbol{u}(k), k] \leq 0\} \tag{3.1.4}$$

For a given problem, the minimum transfer cost is $J_k^* = \min J_k$ (considering all boundary conditions), depends exclusively from state $x(k)$ and time k. We call this minimum cost $V[x(k), k]$ and we obtain the following equation:

$$V[\boldsymbol{x}(k), k] = \min J_k = \min \{\phi[\boldsymbol{x}(k), u(k), k], J_{k+1}\} \tag{3.1.5}$$

where J_{k+1} is called the cost-to-go from state x_{k+1}, since it is encapsulates the remaining cost of the trajectory, and $\theta, \phi \in R^q$ are twice continuous difference functions.

By applying the principle of optimality we obtain:

$$V[\boldsymbol{x}(k), k] = \min \{\phi[\boldsymbol{x}(k), \boldsymbol{u}(k), k] + V[\boldsymbol{x}(k), \boldsymbol{u}(k), k), k+1]\} \tag{3.1.6}$$

Eq. 3.1.6 is known as Bellman's Equation and is the basis of dynamic programming. Dynamic programming finds the optimal cost-to-go for a system by using Bellman's Equation backwards on a trajectory. If the terminal costs J_K^* are known, then J_{K-1}^* can be found by using Bellman's Equation. This process can be repeated until J_0^* is known. The advantage of dynamic programming is that it computes the optimal trajectory for every possible state.

3.1.1 Discrete Dynamic Programming

The general numerical resolution of a Discrete-time Optimal control problem is possible if we introduce a distinct set of points in the permitted state region $X(k)$

and in the permissible control area $U(k)$ [11].
The discretization intervals X(k) and U(k) are chosen appropriately, depending on the problem and the desired solution accuracy.
If we apply to a discrete $x^i(k)$ state, all discrete controls $u^j(k)$, we then have a finite number of transitions in the next k + 1 stage with the corresponding costs. Applying this process to all discrete states, we end up with a discrete system of multi-tiered decisions.
The application of a discrete control value $u^j(k)$ to a discrete state $x^i(k)$ leads to a state:

$$\boldsymbol{x}(k+1) = \phi[\boldsymbol{x}^i(k), \boldsymbol{u}^j(k), k] \tag{3.1.7}$$

Overall, we get a discrete environment to perform the multilevel dynamic programming procedure, which is why this solution method is called discrete dynamic programming. As a consequence of the discretization, for every point $x^i(k)$ there are finitely many transitions $u^j(k)$, which lead to the next step $k+1$. The one-stage optimization can thus be carried out by direct comparison of these transitional lengths. The procedure for DP is to first consider the one-stage (instantaneous) cost for being in the final state, for example at $k = N$. Moving backward in time one step to $k = N - 1$ a control decision in each state will define the state at the next time step ($k = N$), and so the one-stage cost of being in each state is added to the optimal cost from the resulting state onward. By following this procedure a table can be constructed containing the optimal cost of moving from any state x at time k to the end of the problem, where non-optimal control trajectories are ignored completely. The multi-stage optimization procedure thus assumes the following form under the new circumstances, which can easily be converted into a generally applicable computer program [11]:

Stage K - 1: For all grid points $\boldsymbol{x}^i(K-1) \in \boldsymbol{X}(K-1)$, the corresponding discrete control values $u^j(K-1) \in \boldsymbol{U}(x^i(K-1), K-1)$ should be determined, which minimizes:

$$J_{K-1} = \theta(\boldsymbol{x}(K)) + \phi(\boldsymbol{x}^i(K-1), \boldsymbol{u}^j(K-1), K-1)$$

In this case applies:

$$\boldsymbol{x}(K) = f(\boldsymbol{x}^i(K-1), \boldsymbol{u}^j(K-1), K-1).$$

Stage K - 2: For all grid points $\boldsymbol{x}^i(K-2) \in X(K-2)$, the corresponding discrete control values $\boldsymbol{u}^j(K-2) \in U(\boldsymbol{x}^i(K-2), K-2)$ should be determined, which minimizes:

$$J_{K-2} = V(\boldsymbol{x}(K-1), K-1) + \phi(\boldsymbol{x}^i(K2), \boldsymbol{u}^j(K2), K2)$$

In this case applies:

$$\boldsymbol{x}(K-1) = f(\boldsymbol{x}^i(K-2), \boldsymbol{u}^j(K-2), K-2).$$

Stage 0: For $\boldsymbol{x}^i(0)$ the associated discrete control value $\boldsymbol{u}^j(0) \in \boldsymbol{U}(\boldsymbol{x}^i(0), 0)$ is determined as:

$$J = V(\boldsymbol{x}(1), 1) + (\boldsymbol{x}^i(0), \boldsymbol{u}^j(0), 0)$$

In this case applies:

$$\boldsymbol{x}(1) = f(\boldsymbol{x}^i(0), \boldsymbol{u}^j(0), 0).$$

In this way, DP allows us to solve the optimal control problem up to global optimality, but with a different complexity than simple enumeration. To assess its complexity, let us remark that the most cost intensive step is the generation of the cost-to-go functions J_k. Each recursion step needs to go through all n_X states x. For each state it needs to test n_U controls u by evaluating once the system $f(x, u)$ and stage cost $J(x, u)$, which by definition costs one computational unit. Thus, the overall computational complexity is $O(N_n X_n U)$.

3.2 Stochastic Dynamic Programming

Stochastic Dynamic Programming, which has originally introduced by Richard E. Bellman in 1957, is a technique for modelling and solving problems of decision making under uncertainty. Stochastic dynamic programming represents the problem under scrutiny in the form of a Bellman equation. The aim is to compute a policy prescribing how to act optimally in the face of uncertainty.
We consider the discrete-time dynamic system:

$$\boldsymbol{x}(k+1) = \phi[\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{z}(k), k] \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \quad k = 0, ..., K-1 \tag{3.2.1}$$

with the state vector $\boldsymbol{x}(k) \in X(k) \subset R^n$, the control vector $\boldsymbol{u}(k) \in U(k) \subset R^m$ and the stochastic disturbances vector $z(k) \in Z(k) \subset R^p$. The stochastic disturbances vector $z(k)$ can contain arbitrary values of $Z(k)$ with the time, state, and control-dependent probability distribution $P(z|x(k), u(k), k)$. Here, the disturbance values z(k) become independent of all previous values $z(k-1), z(k-2)$,..., provided. We first define the remaining expected costs (cost-to-go) J_k as follows:

$$J_k = E\{\theta[\boldsymbol{x}(K)] + \sum_{k=k}^{K-1} \phi[\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{z}(k), k]\} \tag{3.2.2}$$

For a given problem, the optimal cost-to go $J^* = \min J_k$ depends exclusively from x(k) and k. We denote this minimal cost (optimal cost-to-go) with the V-function, and we obtain:

$$V(\boldsymbol{x}(k), k) = \min J_k = \min E\{\phi(\boldsymbol{x}(k), \boldsymbol{u}(k), z(k), k) + J_{k+1}\} \tag{3.2.3}$$

By applying the optimality principle in eq. 3.2.3 we obtain the stochastic version of Bellman's recursion formula:

$$V(\boldsymbol{x}(k), k) = \min E\{\phi(\boldsymbol{x}(k), \boldsymbol{u}(k), z(k), k) + V(f(\boldsymbol{x}(k), \boldsymbol{u}(k), z(k), k), k+1)\} \tag{3.2.4}$$

With the following boundary condition:

$$V[\boldsymbol{x}(K), K] = \theta[\boldsymbol{x}(K)] \tag{3.2.5}$$

3.3 Complexity of Dynamic Programming

The DP algorithm discussed in the previous section requires a minimization of cost function for each time step, over the space of control inputs. This minimization can result in a major difficulty known as curse of dimensionality. A numerical solution of each minimization in DP is prone to curse of dimensionality, because it has an exponential complexity with respect the dimensions of the state and the control signal. To further explain the curse of dimensionality, one can observe that in each iteration of the Bellman recursion, an expected value operation and a minimization operation needs to take place. The multi-stage procedure also leads to a global minimum of the entire problem, since the dynamic programming indirectly investigates all possible combinations of transitions. This global minimum, of course, refers to the discretized problem definition and will thus represent an approximation of the solution of the original discrete-time (but value-continuous) control task. However, for sufficiently short discretization intervals $\Delta\boldsymbol{x}(k), \Delta\boldsymbol{u}(k)$, this approximation can be made as precise as desired. The result of the discrete dynamic programming is an optimal control law, which is present in tabular form: for each grid point $x^i(k)$, the associated discrete control value $u^l(i)(k)$ is known which optimally in the next time point $k+1$ leads. Let $a^i(k), i = 1, ., , , n$, be the number of grid points for the individual components of the state vector $\boldsymbol{x}(k)$ and $\beta^j(k), j = 1, ..., m$, the number of discrete control values for the individual components of the control vector $\boldsymbol{u}(k)$ [9].

Then the state grid includes a total of

$$\textstyle\sum_{k=0}^{K} \prod_{i=1}^{n} a_i(k)$$

points, and the number of transfers for each of these points is:

$$\textstyle\prod_{j=1}^{m} \beta_j(k)$$

Assuming that the number of transitions for each of these points is, for simplification, $a^i(k) = a$ and $\beta^j(k) = \beta$ for all i, all j, and all k, the computation time required is:

$$\tau \sim K a^n \beta^m.$$

The memory required for the storage of the tabular control law is $\boldsymbol{x}^i(k)m$ values. This results in

$$m \textstyle\sum_{k=0}^{K} \prod_{i=1}^{n} a_i(k)$$

total values to be stored. The above equations show that the computational effort required when using discrete dynamic programming increases exponentially with the problem dimensions n, m.

4 Development of analytic solution to trajectory optimization for statically managed traffic lights

4.1 Optimal Control Problem Formulation

In this chapter we are focusing in the development an optimal trajectory for the vehicle considering the existence of fixed-time traffic lights. In this case, an appropriate optimal control problem is formulated and solved analytically via the Pontryagin's Minimum Principle (PMP) for four different case studies. For the formulation of the Optimal Control Problem, the following are considered:

1. Only one vehicle and a single lane are considered.
2. The red traffic lights are detected at t_0.
3. The traffic lights' switching time is fixed and perfectly known.
4. The system is described by the following state variables where x is the position, ν is the speed and u is the acceleration as control variable.

$$\dot{x} = \nu \qquad \dot{\nu} = u \tag{4.1.1}$$

5. The initial and final conditions of the vehicle, are denoted as:

$$x(t_0) = x_0 \tag{4.1.2}$$

$$\nu(t_0) = \nu_0 \tag{4.1.3}$$

$$x(t_e) = x_e \tag{4.1.4}$$

$$\nu(t_e) = \nu_e \tag{4.1.5}$$

4.2 Case 1: Analytic Solution with fixed final time & no traffic light.

We consider the case of defining an optimal trajectory, which minimizes the required acceleration, starting from a known initial point to a fixed final destination point. In Case 1, we assume that there is no traffic light interfering the vehicle's trajectory and the final time is fixed and known beforehand.
The objective is to bring the system from the initial condition $x_0 = [x_0, \nu_0]^T$ to the final condition $x_e = [x_e, \nu_e]^T$ by time t_e while minimizing the fuel-consumption cost criterion:[12]

$$J = \frac{1}{2} \int_{t_0}^{t_e} u(t)^2 dt \tag{4.2.1}$$

Using the optimality condition (eq. 2.2.30) we define the Hamiltonian function as:

$$H(t) = \frac{1}{2} u^2 + \lambda_1 \nu + \lambda_2 u \tag{4.2.2}$$

Where λ_1, λ_2 are co-state variables.Using the Pontryagin's principle, we derive the following 1^{st} order necessary conditions for optimality:

$$\dot{x} = \frac{\vartheta H}{\vartheta \lambda_1} = \nu \tag{4.2.3}$$

$$\dot{\nu} = \frac{\vartheta H}{\vartheta \lambda_2} = u \tag{4.2.4}$$

$$\dot{\lambda_1} = \frac{-\vartheta H}{\vartheta x} = 0 \tag{4.2.5}$$

$$\dot{\lambda_2} = \frac{-\vartheta H}{\vartheta \nu} = -\lambda_1 \tag{4.2.6}$$

$$H_u = u + \lambda_2 = 0 \tag{4.2.7}$$

Solving the ODE system (eq. 4.2.3 - 4.2.5) we obtain that:

$$\lambda_1(t) = c_1 \tag{4.2.8}$$

$$\lambda_2(t) = -c_1 t - c_2 \tag{4.2.9}$$

and

$$u(t) = -\lambda_2 = c_1 t + c_2 \tag{4.2.10}$$

$$\nu(t) = \frac{1}{2}c_1 t^2 + c_2 t + c_3 \tag{4.2.11}$$

$$x(t) = \frac{1}{6}c_1 t^3 + \frac{1}{2}c_2 t^2 + c_3 t + c_4 \tag{4.2.12}$$

Where $c_1 - c_4$ are constants to be computed. In order to calculate these four constants the initial and the final conditions (eq. 6.2.8-6.2.7) of the problem are used, resulting in the derivation of the following system of equations.

$$x_0 = \frac{1}{6}c_1 {t_0}^3 + \frac{1}{2}c_2 {t_0}^2 + c_3 t_0 + c_4 \tag{4.2.13}$$

$$\nu_0 = \frac{1}{2}c_1 {t_0}^2 + c_2 t_0 + c_3 \tag{4.2.14}$$

$$x_e = \frac{1}{6}c_1 {t_e}^3 + \frac{1}{2}c_2 {t_e}^2 + c_3 t_e + c_4 \tag{4.2.15}$$

$$\nu_e = \frac{1}{2}c_1 {t_e}^2 + c_2 t_e + c_3 \tag{4.2.16}$$

Solving the above system we obtain four constants $c_1 - c_4$ which depend on the initial and finally conditions and can easily computed using differential computing (e.g Mathematica).

$$c_1 = \frac{6(t_0\nu_0 - t_e\nu_0 + t_0 v_e - t_e\nu_e - 2x_0 + 2x_e)}{(t_0 - t_e)^3} \tag{4.2.17}$$

$$c_2 = -\frac{2t_0^2\nu_0 - t_e\nu_0 t_0 - 2t_e^2\nu_0 + 2t_0^2\nu_e - t_0 t_e v_e - t_e^2 v_e - 3t_0 x_0 - 3t_e x_0 + 3t_0 x_e + 3t_e x_e}{(t_0 - t_e)^3} \tag{4.2.18}$$

$$c_3 = -\frac{-2t_0^2 v_0 t_e + t_e^2 v_0 t_0 + t_e^3 v_0 - t_0^3 v_e - t_0^2 t_e v_e - t_e^2 v_e + 2t_0 t_e^2 v_e + 6t_e x_0 t_0 - 6t_0 x_e t_e}{(t_0 - t_e)^3} \tag{4.2.19}$$

$$c_4 = -\frac{t_0^2 v_0 t_e^2 - t_e^3 v_0 t_0 + t_0^3 v_e t_e - t_0^2 t_e^2 v_e - 3t_0 t_e^2 x_0 + t_e^3 x_0 - t_0^3 x_e + 3t_e x_e t_0^2}{(t_0 - t_e)^3} \tag{4.2.20}$$

Hence, the derived four constants lead to an analytic solution of the state equations, which due to the complexity of the equations cannot be presented.

4.3 Case 2: Analytic Solution with free final time & no traffic light.

In this section, we propose an extension of Case 1. Particularly, we consider the minimization of vehicle's acceleration, but with free final time (t_e).
The objective is to bring the system from the initial condition $x_0 = [x_0, \nu_0]^T$ to the final condition $x_e = [x_e, \nu_e]^T$ by time t_e while minimizing the cost criterion:

$$J = \frac{1}{2}\int_{t_0}^{t_e} u(t)^2 dt + \frac{1}{2} w t_e \tag{4.3.1}$$

where w is the weighting factor of the final free time.
Using the optimality conditions (eq. 2.2.30) we define the Hamiltonian function as:

$$H(x, \nu, u, t) = \frac{1}{2}u^2 + \lambda_1 \nu + \lambda_2 u \tag{4.3.2}$$

where λ_1, λ_2 are co-state variables. Using the Pontryagin's principle, we derive the following 1^{st} order necessary conditions for optimality.

$$\dot{x} = \frac{\vartheta H}{\vartheta \lambda_1} = \nu \tag{4.3.3}$$

$$\dot{\nu} = \frac{\vartheta H}{\vartheta \lambda_2} = u \tag{4.3.4}$$

$$\dot{\lambda_1} = \frac{-\vartheta H}{\vartheta x} = 0 \tag{4.3.5}$$

$$\dot{\lambda_2} = \frac{-\vartheta H}{\vartheta \nu} = -\lambda_1 \tag{4.3.6}$$

$$H_u = u + \lambda_2 = 0 \tag{4.3.7}$$

Solving the ODE system (eq. 4.3.3 - 4.3.5) we obtain that:

$$\lambda_1(t) = c_1 \tag{4.3.8}$$

$$\lambda_2(t) = -c_1 t - c_2 \tag{4.3.9}$$

and

$$u(t) = -\lambda_2 = c_1 t + c_2 \tag{4.3.10}$$

$$\nu(t) = \frac{1}{2} c_1 t^2 + c_2 t + c_3 \tag{4.3.11}$$

and

$$x(t) = \frac{1}{6} c_1 t^3 + \frac{1}{2} c_2 t^2 + c_3 t + c_4 \tag{4.3.12}$$

For the final free time the following transversality condition must be satisfied.

$$H(t_e) + \frac{1}{2} w = 0 \tag{4.3.13}$$

In order to calculate the for constants $c_1 - c_4$ the the initial and the final conditions o the trajectory are used, which lead to the following system of equations.

$$x_0 = \frac{1}{6} c_1 {t_0}^3 + \frac{1}{2} c_2 {t_0}^2 + c_3 t_0 + c_4 \tag{4.3.14}$$

$$\nu_0 = \frac{1}{2} c_1 {t_0}^2 + c_2 t_0 + c_3 \tag{4.3.15}$$

$$x_e = \frac{1}{6} c_1 {t_e}^3 + \frac{1}{2} c_2 {t_e}^2 + c_3 t_e + c_4 \tag{4.3.16}$$

$$\nu_e = \frac{1}{2} c_1 {t_e}^2 + c_2 t_e + c_3 \tag{4.3.17}$$

$$H(t_e) + \frac{1}{2} w = 0 \Rightarrow -c_2^2 + 2 c_1 c_3 + w = 0 \tag{4.3.18}$$

Solving the above system we obtain the following four constants $c_1 - c_4$.

$$c_1 = \frac{6(t_0\nu_0 - t_e\nu_0 + t_0 v_e - t_e\nu_e - 2x_0 + 2x_e)}{(t_0 - t_e)^3} \tag{4.3.19}$$

$$c_2 = -\frac{2t_0^2\nu_0 - t_e\nu_0 t_0 - 2t_e^2\nu_0 + 2t_0^2\nu_e - t_0 t_e v_e - t_e^2 v_e - 3t_0 x_0 - 3t_e x_0 + 3t_0 x_e + 3t_e x_e}{(t_0 - t_e)^3} \tag{4.3.20}$$

$$c_3 = -\frac{-2t_0^2 v_0 t_e + t_e^2 v_0 t_0 + t_e^3 v_0 - t_0^3 v_e - t_0^2 t_e v_e - t_e^2 v_e + 2t_0 t_e^2 v_e + 6t_e x_0 t_0 - 6t_0 x_e t_e}{(t_0 - t_e)^3} \tag{4.3.21}$$

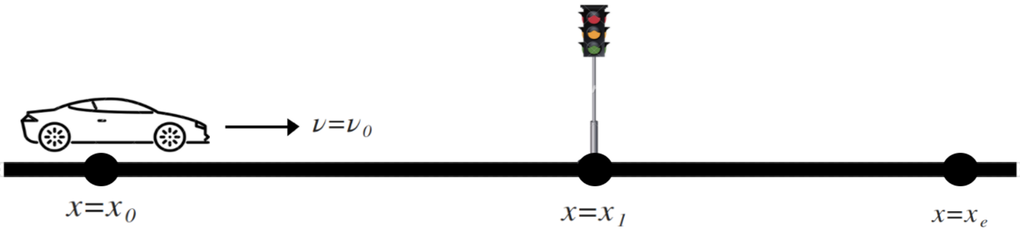
$$c_4 = -\frac{t_0^2 v_0 t_e^2 - t_e^3 v_0 t_0 + t_0^3 v_e t_e - t_0^2 t_e^2 v_e - 3t_0 t_e^2 x_0 + t_e^3 x_0 - t_0^3 x_e + 3t_e x_e t_0^2}{(t_0 - t_e)^3} \tag{4.3.22}$$

$$-c_2^2 + 2c_1 c_3 + w = 0 \tag{4.3.23}$$

Hence, these conditions lead to an analytic solution of the $c_1 - c_4$ and the final free time. Therefore we can derive the state equations, which due to complexity of the equations can not be presented.
Using Mathematica, there are two options in order to derive the state equations.We can either use the solution of the four constants from Case 1 and then add the equation for the final free time (eq.4.3.13), or solve the system of the five equations directly. Both options have the same results, but the second option needs 108.54 seconds in order to derive the equations of motion and the first option needs 8.26 seconds. For this reason, the first option is preferred as for the aspect of computational time.

4.4 Case 3: Analytic Solution with fixed final time & a traffic light.

In case 3, we consider the minimization of the vehicles trajectory, starting from a known initial point to a final destination, but with a traffic light interfering the vehicle's trajectory. The traffic light is detected at red phase. For the case with traffic light, extra consideration is needed for the calculation of the optimal velocity profile. The problem can be divided into two parts, the first part is the region until the vehicle reaches where the traffic light is located and the second part is the region from the traffic light to the destination point.



For the formulation of the Optimal Control problem the following are considered:

1. The red traffic light is detected at t_0.
2. The traffic light's switching time is known and is defined as T
3. The traffic light's position is known and is defined as x_1
4. We define as t_1 the moment that our vehicle passes the red traffic light.
5. In order to avoid creating a trajectory where vehicle passes through the red traffic light, in addition to the initial and final conditions, we set the following point constraint which allows us to specify the position of the vehicle at moment t_1.

$$x(t_1) = x_1 \tag{4.4.1}$$

6. We define the following inequality constraint, which ensures that the Optimal time that the vehicle passes through the traffic light (t_1) is equal or greater than the traffic light's switching time.

$$h[x(t), \nu(t), u(t), t] = T - t_1 \leq 0 \tag{4.4.2}$$

For case 3 we assume that the final time is fixed.
Using the optimality conditions (2.2.30) we define the Hamiltonian function:

$$H(x, \nu, u, t) = \frac{1}{2}u^2 + \lambda_1 \nu + \lambda_2 u \tag{4.4.3}$$

where λ_1, λ_2 are co-state variables.
Using the Pontryagin's principle, we derive the following $1st$ order necessary conditions for optimality.

$$\dot{x} = \frac{\vartheta H}{\vartheta \lambda_1} = \nu \tag{4.4.4}$$

$$\dot{\nu} = \frac{\vartheta H}{\vartheta \lambda_2} = u \tag{4.4.5}$$

$$\dot{\lambda_1} = \frac{-\vartheta H}{\vartheta x} = 0 \tag{4.4.6}$$

$$\dot{\lambda_2} = \frac{-\vartheta H}{\vartheta \nu} = -\lambda_1 \tag{4.4.7}$$

$$H_u = u + \lambda_2 = 0 \tag{4.4.8}$$

Solving the ODE system (eq. 4.4.4 - 4.4.7) we obtain that:

$$\lambda_1(t) = c_1 \tag{4.4.9}$$

$$\lambda_2(t) = -c_1 t - c_2 \tag{4.4.10}$$

and

$$u(t) = -\lambda_2 = c_1 t + c_2 \tag{4.4.11}$$

$$\nu(t) = \frac{1}{2} c_1 t^2 + c_2 t + c_3 \tag{4.4.12}$$

Now eq. (4.4.2) represents a set of terminal constraints for the point of the path $t = t_0$ to $t = t_1$. If we let t_1^- signify just before t_1 and t_1^+ just after t_1, we may interpret the influence functions and the Hamiltonian function at t_1 as:

$$\lambda_1(t_1^+) = \lambda_1(t_1^-) + \pi^\tau \frac{\vartheta N}{\vartheta x(t_1)} \quad \text{(discontinuity)} \tag{4.4.13}$$

$$\lambda_2(t_1^+) = \lambda_2(t_1^-) \quad \text{(continuity)} \tag{4.4.14}$$

$$H(t_1^-) = H(t_1^+) \quad \text{(continuity)} \tag{4.4.15}$$

where π is a q-component vector of constant Lagrange multipliers, determined so that the q conditions are satisfied. According to the derived equations there's continuity of speed and position at t_1.

Applying the necessary conditions of optimality as in the previous section, the constraint eq. (4.4.2) leads into two linear accelerations and therefore the equations of motion are transformed as follows:

$$u(t) = \begin{cases} c_1 t + c_2 & t_0 \le t \le t_1^- \\ c_1 t + c_2 + c_3(t - t_1) & t_1^+ \le t \le t_e \end{cases} \tag{4.4.16}$$

$$\nu(t) = \begin{cases} \frac{1}{2}c_1 t^2 + c_2 t + c_3 & t_0 \le t \le t_1^- \\ (c_1 t_1 + c_2)(t - t_1) + \frac{1}{2}c_5(t - t_1)^2 + c_6 & t_1^+ \le t \le t_e \end{cases} \tag{4.4.17}$$

$$x(t) = \begin{cases} \frac{1}{6}c_1 t^3 + \frac{1}{2}c_2 t^2 + c_3 t + c_4 & t_0 \le t \le t_1^- \\ \frac{1}{2}(c_1 t_1 + c_2)(t - t_1)^2 + \frac{1}{6}c_5(t - t_1)^3 + c_6(t - t_1) + c_7 & t_1^+ \le t \le t_e \end{cases} \tag{4.4.18}$$

For simplification reasons we transform the above equations as:

For $0 \le t \le t_1^-$ we have the following equations:

$$u_1(t) = c_1 t + c_2 \tag{4.4.19}$$

$$\nu_1(t) = \frac{1}{2}c_1 t^2 + c_2 t + c_3 \tag{4.4.20}$$

$$x_1(t) = \frac{1}{6}c_1 t^3 + \frac{1}{2}c_2 t^2 + c_3 t + c_4 \tag{4.4.21}$$

For $t_1^+ \leq t \leq t_e$ we have the following equations:

$$x_2(t) = \frac{1}{2}(c_1 t_1 + c_2)(t - t_1)^2 + \frac{1}{6}c_5(t - t_1)^3 + c_6(t - t_1) + c_7 \tag{4.4.22}$$

$$\nu_2(t) = (c_1 t_1 + c_2)(t - t_1) + \frac{1}{2}c_5(t - t_1)^2 + c_6 \tag{4.4.23}$$

$$u_2(t) = c_1 t_1 + c_2 + c_5(t - t_1) \tag{4.4.24}$$

The objective of this approach is to bring the system from the initial condition to the final state by a fixed final time (t_e) while minimizing the following extended cost criterion: [13]

$$J = \frac{1}{2}\int_{t_0}^{t_1} u_1^2 dt + \frac{1}{2}\int_{t_1}^{te} u_2^2 dt \tag{4.4.25}$$

Obeying the following initial and final conditions:

$$x_1(t_0) = x_0 \quad \nu_1(t_0) = \nu_0 \tag{4.4.26}$$

$$x_2(t_e) = x_e \quad \nu_2(t_e) = \nu_e \tag{4.4.27}$$

The resulting value of t_1, is derived from the following applied KKT condition. (4.4.2).

$$\mu(T - t_1) = 0 \tag{4.4.28}$$

More explicitly there we distinguish an active and an inactive inequality constraint as follows:

$$\mu = \begin{cases} > 0 & h = 0 \quad \text{(active)} \\ \\ = 0 & h < 0 \quad \text{(inactive)} \end{cases} \tag{4.4.29}$$

In order to derive time t_1 which leads to the optimal cost criterion (J^*) we formulate the following equations, which consist of the gradient of J^* and $h = (T - t_1)$ Finally, from eq. (4.4.29) we obtain:

For $\mu > 0$:

$$T = t_1 \tag{4.4.30}$$

$$\frac{\vartheta J^*}{\vartheta t_1} + \frac{\vartheta[\mu(T - t_1)]}{\vartheta t_1} = 0 \tag{4.4.31}$$

For $\mu = 0$:

$$\mu(T - t_1) = 0 \tag{4.4.32}$$

$$\frac{\vartheta J^*}{\vartheta t_1} = 0 \tag{4.4.33}$$

In the event that the obtained time t_1 is before the traffic lights switching time (T) then the inequality constraint eq. (4.4.2) is inactive and the value of t_1 becomes equal to the traffic lights switching time (T).

In this case, 8 constants ($c_1 - c_7$) and KKT multiplier μ have to be defined. Using the initial and final condition, the point constraint eq. (4.4.1), and the continuity of speed and position at t_1, we derive the following system of equations.

$$x_1(t_0) = \frac{1}{6}c_1{t_0}^3 + \frac{1}{2}c_2{t_0}^2 + c_3t_0 + c_4 \tag{4.4.34}$$

$$\nu_1(t_0) = \frac{1}{2}c_1{t_0}^2 + c_2t_0 + c_3 \tag{4.4.35}$$

$$x_2(t_e) = \frac{1}{2}(c_1t_1 + c_2)(t_e - t_1)^2 + \frac{1}{6}c_5(t_e - t_1)^3 + c_6(t_e - t_1) + c_7 \tag{4.4.36}$$

$$\nu_2(t_e) = (c_1t_1 + c_2)(t_e - t_1) + \frac{1}{2}c_5(t_e - t_1)^2 + c_6 \tag{4.4.37}$$

$$x_1(t_1) = x_1 \Rightarrow x_1 = \frac{1}{6}c_1{t_1}^3 + \frac{1}{2}c_2{t_1}^2 + c_3t_1 + c_4 \tag{4.4.38}$$

$$x_1(t_1) = x_2(t_1) \Rightarrow \frac{1}{6}c_1{t_1}^3 + \frac{1}{2}c_2{t_1}^2 + c_3t_1 + c_4 = c_7 \tag{4.4.39}$$

$$\nu_1(t_1) = \nu_1(t_1) \Rightarrow \frac{1}{2}c_1{t_1}^2 + c_2t_1 + c_3 = c_6 \tag{4.4.40}$$

By solving the above system, using symbolic differentiation tools (e.g Mathematica), we can calculate the eight constants $c_1 - c_7$ and the KKT multiplier μ.

Therefore we can obtain the equations of motion, which due to complexity of the equations can not be presented.

4.5 Case 4: Analytic Solution with free final time & a traffic light

In this approach, we consider an extension of all the previous approaches. The purpose of this approach is the minimization of the vehicles acceleration, starting from a known initial state to a final destination, with free final time and a traffic light interfering the vehicle's trajectory. Moreover the final time (t_e)is free.

Using the optimality conditions (see section 2.2.30) we define the Hamiltonian function:

$$H(x, \nu, u, t) = \frac{1}{2}u^2 + \lambda_1 \nu + \lambda_2 u \tag{4.5.1}$$

where λ_1, λ_2 are co-state variables.
Using the Pontryagin's principle , we derive the following 1_{st} order necessary conditions for optimality.

$$\dot{x} = \frac{\vartheta H}{\vartheta \lambda_1} = \nu \tag{4.5.2}$$

$$\dot{\nu} = \frac{\vartheta H}{\vartheta \lambda_2} = u \tag{4.5.3}$$

$$\dot{\lambda_1} = \frac{-\vartheta H}{\vartheta x} = 0 \tag{4.5.4}$$

$$\dot{\lambda_2} = \frac{-\vartheta H}{\vartheta \nu} = -\lambda_1 \tag{4.5.5}$$

$$H_u = u + \lambda_2 = 0 \tag{4.5.6}$$

Solving the ODE system (eq. 4.5.2 - 4.5.5) we obtain that:

$$\lambda_1(t) = c_1 \tag{4.5.7}$$

$$\lambda_2(t) = -c_1 t - c_2 \tag{4.5.8}$$

and

$$u(t) = -\lambda_2 = c_1 t + c_2 \tag{4.5.9}$$

$$\nu(t) = \frac{1}{2}c_1 t^2 + c_2 t + c_3 \tag{4.5.10}$$

Now eq. 4.4.2 represents a set of terminal constraints for the point of the path $t = t_0$ to $t = t_1$. If we let t_1^- signify just before t_1 and t_1^+ just after t_1, we may interpret the influence functions and the Hamiltonian function at t_1^+ as:

$$\lambda_1(t_1^+) = \lambda_1(t_1^-) + \pi^\tau \frac{\vartheta N}{\vartheta x(t_1)} \quad \text{(discontinuity)} \tag{4.5.11}$$

$$\lambda_2(t_1^+) = \lambda_2(t_1^-) \quad \text{(continuity)} \tag{4.5.12}$$

$$H(t_1^-) = H(t_1^+) \quad \text{(continuity)} \tag{4.5.13}$$

where π is a q-component vector of constant Lagrange multipliers, determined so that the q conditions are satisfied. According to the derived equations there's continuity of speed and position at t_1.

Applying the necessary conditions of optimality as in the previous section, the constraint eq. (4.4.2) leads into two linear accelerations and therefore the equations of motion are transformed as follows:

$$u(t) = \begin{cases} c_1 t + c_2 & t_0 \le t \le t_1^- \\ c_1 t + c_2 + c_3(t - t_1) & t_1^+ \le t \le t_e \end{cases} \tag{4.5.14}$$

$$\nu(t) = \begin{cases} \frac{1}{2}c_1 t^2 + c_2 t + c_3 & t_0 \le t \le t_1^- \\ (c_1 t_1 + c_2)(t - t_1) + \frac{1}{2}c_5 (t - t_1)^2 + c_6 & t_1^+ \le t \le t_e \end{cases} \tag{4.5.15}$$

$$x(t) = \begin{cases} \frac{1}{6}c_1 t^3 + \frac{1}{2}c_2 t^2 + c_3 t + c_4 & t_0 \le t \le t_1^- \\ \frac{1}{2}(c_1 t_1 + c_2)(t - t_1)^2 + \frac{1}{6}c_5 (t - t_1)^3 + c_6 (t - t_1) + c_7 & t_1^+ \le t \le t_e \end{cases} \tag{4.5.16}$$

For simplification reasons we transform the above equations as:

For $0 \le t \le t_1^-$ we have the following equations:

$$u_1(t) = c_1 t + c_2 \tag{4.5.17}$$

$$\nu_1(t) = \frac{1}{2}c_1 t^2 + c_2 t + c_3 \tag{4.5.18}$$

$$x_1(t) = \frac{1}{6}c_1 t^3 + \frac{1}{2}c_2 t^2 + c_3 t + c_4 \tag{4.5.19}$$

For $t_1^+ \le t \le t_e$ we have the following equations:

$$x_2(t) = \frac{1}{2}(c_1 t_1 + c_2)(t - t_1)^2 + \frac{1}{6}c_5(t - t_1)^3 + c_6(t - t_1) + c_7 \tag{4.5.20}$$

$$\nu_2(t) = (c_1 t_1 + c_2)(t - t_1) + \frac{1}{2}c_5(t - t_1)^2 + c_6 \tag{4.5.21}$$

$$u_2(t) = c_1 t_1 + c_2 + c_5(t - t_1) \tag{4.5.22}$$

The objective of this approach is to bring the system from the initial condition to the final state by a fixed final time (t_e) while minimizing the following extended cost criterion:

$$J = \frac{1}{2}\int_{t_0}^{t_1} u_1^2 dt + \frac{1}{2}\int_{t_1}^{te} u_2^2 dt + \frac{1}{2}wt_e \tag{4.5.23}$$

With the following initial and final conditions:

$$x_1(t_0) = x_0 \quad \nu_1(t_0) = \nu_0 \tag{4.5.24}$$

$$x_2(t_e) = x_e \quad \nu_2(t_e) = \nu_e \tag{4.5.25}$$

The resulting value of t_1, is derived from the following applied KKT condition. [13]

$$\mu(T - t_1) = 0 \tag{4.5.26}$$

More explicitly there we distinguish an active and an inactive inequality constraint as follows:

$$\mu = \begin{cases} > 0 & h = 0 \quad \text{(active)} \\ \\ = 0 & h < 0 \quad \text{(inactive)} \end{cases} \tag{4.5.27}$$

In order to derive time t_1 which leads to the optimal cost criterion (J^*) we formulate the following equations, which consist of the gradient of J^* and $h = (T - t_1)$ Finally, from eq. (4.5.27) we obtain:

For $\mu > 0$:

$$T = t_1 \tag{4.5.28}$$

$$\frac{\vartheta J^*}{\vartheta t_1} + \frac{\vartheta[\mu(T - t_1)]}{\vartheta t_1} = 0 \tag{4.5.29}$$

For $\mu = 0$:

$$\mu(T - t_1) = 0 \tag{4.5.30}$$

$$\frac{\vartheta J^*}{\vartheta t_1} = 0 \tag{4.5.31}$$

In the event that the obtained time t_1 is before the traffic lights switching time (T) then the inequality constraint eq. 4.4.2 is inactive and the value of t_1 becomes equal to the traffic lights switching time (T).

In this case, 8 constants ($c_1 - c_7$), KKT multiplier μ and final free time have to be defined. Using the initial and final condition, the point constraint eq. (4.4.2), and the continuity of speed and position at t_1, we derive the following system of equations.

$$x_1(t_0) = \frac{1}{6}c_1{t_o}^3 + \frac{1}{2}c_2{t_o}^2 + c_3t_o + c_4 \tag{4.5.32}$$

$$\nu_1(t_0) = \frac{1}{2}c_1{t_0}^2 + c_2t_0 + c_3 \tag{4.5.33}$$

$$x_2(t_e) = \frac{1}{2}(c_1t_1 + c_2)(t_e - t_1)^2 + \frac{1}{6}c_5(t_e - t_1)^3 + c_6(t_e - t_1) + c_7 \tag{4.5.34}$$

$$\nu_2(t_e) = (c_1t_1 + c_2)(t_e - t_1) + \frac{1}{2}c_5(t_e - t_1)^2 + c_6 \tag{4.5.35}$$

$$x_1(t_1) = x_1 \Rightarrow x_1 = \frac{1}{6}c_1{t_1}^3 + \frac{1}{2}c_2{t_1}^2 + c_3t_1 + c_4 \tag{4.5.36}$$

$$x_1(t_1) = x_2(t_1) \Rightarrow \frac{1}{6}c_1{t_1}^3 + \frac{1}{2}c_2{t_1}^2 + c_3t_1 + c_4 = c_7 \tag{4.5.37}$$

$$\nu_1(t_1) = \nu_1(t_1) \Rightarrow \frac{1}{2}c_1{t_1}^2 + c_2t_1 + c_3 = c_6 \tag{4.5.38}$$

$$H(t_e) + \frac{1}{2}w = 0 \Rightarrow -c_2^2 + 2c_1c_3 + w = 0 \tag{4.5.39}$$

The solution of the system gives us several values for the final free time, from which some of them are infeasible and from the feasible ones we choose the one that leads to the minimum value of the cost function.
Therefore we can obtain the equations of motion, which due to the complexity of the equations cannot be presented.

5 Development of numerical solution to trajectory optimization for dynamically managed traffic lights.

5.1 Problem Statement

In previous chapters of this thesis the generation of optimal velocity profile, based on the information about the traffic and the route, has been discussed. Furthermore, traffic light related information may not be static and might be subject of change. For example in high density traffic the operation sequence of traffic lights can change, according to the traffic flow rate. In the case of real-time signals, availability of a time-window of possible signal switching times, along with the corresponding probability distribution, is assumed, and the problem is cast in the format of a stochastic optimal control problem and is solved numerically using SDP techniques For the formulation of the Problem the following are considered:

1. Only one vehicle and a single lane are considered.

2. The position of the traffic light is known by the vehicle.

3. The traffic light is detected in red phase.

4. The state variables should obey the following linear discrete-time state equations:

$$x(k+1) = x(k) + \nu(k)\tau + \frac{\tau^2}{2}u(k) \tag{5.1.1}$$

$$\nu(k+1) = \nu(k) + u(k)\tau \tag{5.1.2}$$

5. The traffic light's switching time T is not perfectly known by the vehicle. However, the switching time is bounded between a known minimum and a known maximum value, where:

$$T_{min} \leq T \leq T_{max} \tag{5.1.3}$$

6. We also enforce the following hard inequality constraints:

$$\nu_{\min} \leq \nu_k \leq \nu_{\max} \tag{5.1.4}$$

$$u_{\min} \leq u_k \leq u_{\max} \tag{5.1.5}$$

where $\nu_{\min}$ and $\nu_{\max}$ are the road speed limits and $u_{\min}$, $u_{\max}$ are the feasible bounds for deceleration and acceleration.

5.2 Probability density function

In the stochastic case we define a probability density function $p(z|k)$ where:

$$z(k) = \begin{cases} 0, & \text{If the traffic lights switch at } (k+1)\tau \\ 1, & \text{else} \end{cases} \tag{5.2.1}$$

In particular, the chance that the traffic light will switch between the interval $[k\tau, (k+1)\tau]$, under the condition that it has not switched up to k,can be calculated for various distributions, and can be cropped and scaled in every time step according to the knowledge we have the traffic light's switching time. In the event that the traffic light will switch at time step $(k+1)\tau$ the vehicle follows the trajectory from the previous optimal control problem with cost J^*_{opt}. Different distributions have been formulated and are presented in sections 5.2.1 and 5.2.2.

5.2.1 Uniform probability distribution

In probability theory and statistics, the discrete uniform distribution is a symmetric probability distribution whereby a finite number of values are equally likely to be observed; every one of n values has equal probability 1/n. For p being uniform between the interval $[T_{min}, T_{max}]$ we obtain that:

$$p(z=0|k) = \frac{1}{n} \quad \text{where} \quad n = (T_{\max} - T_{\min}) + 1 \tag{5.2.2}$$

where the probabilities are cropped and scaled in every stage that the traffic light does not switch into green [14].

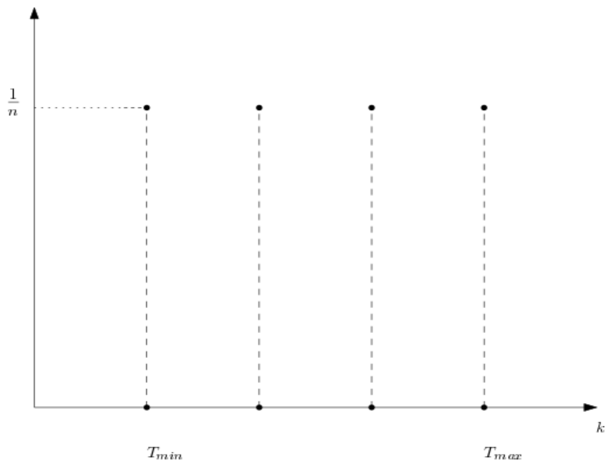


Figure 2 Uniform probability distribution.

5.2.2 Triangular probability distribution

In probability theory and statistics, the triangular distribution is a continuous probability distribution with lower limit a, upper limit b and mode c, where $a < b$ and $a \leq c \leq b$. In our problem we define the lower limit as $T_{\min}$, the upper limit as $T_{\max}$ and the mode probability as $p[c] = \frac{2}{T_{max}-T_{min}}$. The triangular probability density function formulated in the SDP, is denoted as:

$$p[z=0|k] = \begin{cases} \frac{2(k-T_{min})}{(T_{max}-T_{min})(c-T_{min})} & if \quad T_{min} \leq k < c \\ \\ \frac{2}{T_{min}-T_{max}}, & if \quad k = c \\ \\ \frac{2(T_{max}-k)}{(T_{max}-T_{min})(T_{max}-c)} & if \quad c < k \leq T_{max} \end{cases} \tag{5.2.3}$$

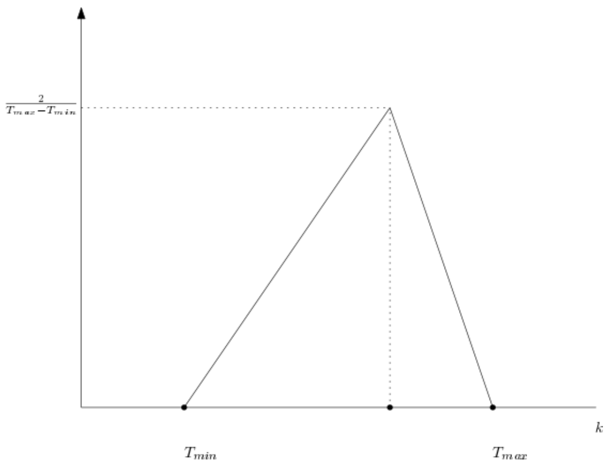


Figure 3 Triangular Probability distribution.

5.3 The Hamilton-Jacobi-Bellman equation

The cost function consists of minimizing the cost of the action $\phi[x(k), u(k), z(k)]$, and the expected cost of the situation of the next time step. That leads to the following modified Bellman's equation:

$$V[x(k), \widetilde{x}(k), k] = \min\{\phi[x(k), u(k), z(k)] + \mathbb{P}(z = 0|k)J^*_{opt}+ \tag{5.3.1}$$

$$+ (1 - \mathbb{P}(z = 0|k))V[x(k+1), z(k), k+1)]\}$$

5.4 Stochastic Dynamic Programming Algorithm

In order to find an optimal trajectory, the algorithm of discrete-time stochastic dynamic programming has been used. An illustration of the grid-based solution can be seen in Fig. 4.

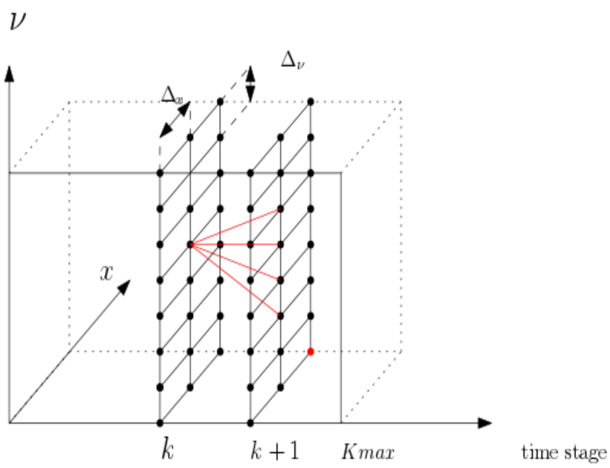


Figure 4 Illustration of the 3D grid of time and state space.

The grid is representing the dimension of time in the interval $[0, T_{\max}]$ and the two dimensional state space x in $[0, x_{\max}]$ and ν in $[0, \nu_{\max}]$. Each cell is a discrete point k with a state $\boldsymbol{x}(k)$.
The presented algorithm of stochastic dynamic programming (SDP) has been implemented. We define as checkpoint-states, the states between $T_{\min}$ and $T_{\max}$.

For these states the optimal cost function consists of minimizing the actual cost for the action $\phi[x(k), u(k), z(k)]$ with a probability $\mathbb{P}(z = 0|k)$ and the cost of the optimal control problem with a probability $1 - \mathbb{P}(z = 0|k)$ (the traffic light will not switch to green).

If the state is not a checkpoint-state then the probability of switching $\mathbb{P}(z = 0|k)$ is zero and the cost function consists of minimizing only the actual cost for the action $\phi[x(k), u(k), z(k)]$ and the cost-to-go.
For a constant input u_k the cost for the action is:

$$\phi(\boldsymbol{x}(k), u(k), z(k) := \frac{1}{2} u(k)^2 \tag{5.4.1}$$

Where, $\mathbb{P}(z = 0|k)$ indicates the event of observing the traffic light switch to green at time $k\tau$. Moreover,

$$u^*(k) := \arg\min J_k(x(k), u(k)) : u(k) \in \mathbb{U}_k \tag{5.4.2}$$

Algorithm 1 Stochastic Dynamic Programming

$J^*_K(x(K)) \leftarrow J_{opt}(x(K), K)\,, \forall x(K) \in \mathbb{X}_K$
for each $k = K-1, K-2, \ldots, 0$ **do**
  for each state $x(k) \in \mathbb{X}_k$ **do**
    for each control $u(k) \in \mathbb{U}_k$ **do**
      construct $x(k+1)$, $v(k+1)$
      $J_k(x(k), u(k)) \leftarrow \phi(x(k), u(k)) + J^*_{k+1}(x(k+1))$
    end for
    if k is a checkpoint-state **then**
      $J^*_k(x(k)) \leftarrow \mathbb{P}(z = 0|k) J_{opt}(x(k), k) + (1 - \mathbb{P}(z = 0|k) J_k(x(k), u^*(k))$
    else
      $J^*_k(x(k)) \leftarrow J_k(x(k), u^*(k))$
    end if
  end for
end for

5.4.1 Discretization factor

The value of discretization factor not only has a significant impact on the computational time and the memory requirements but it is important for the level of accuracy of the resulting trajectory. More particularly the discretization factor of acceleration plays an important role in the minimization of fuel consumption, as higher values can lead to sudden changes in speed and severe driving. Using eq. (6.2.7), we can derive that the relation between speed and acceleration is linear, and therefore, for $\tau = 1s$, we can set the discretization factor of speed equal to the discretization factor of acceleration ($\Delta V = \Delta U$). Likewise, in order to obtain an suitable discretization factor of position, we assume that if x, ν, u are discrete points in the grid then the following equations apply.

$$x = n\Delta X \tag{5.4.3}$$

$$\nu = m\Delta U = m\Delta V \tag{5.4.4}$$

$$u = l\Delta U \tag{5.4.5}$$

Where n, m, l are integer numbers.
Using the following equation:

$$\Delta x = \frac{1}{2}\Delta u \Delta t^2 \tag{5.4.6}$$

and eq.(5.4.3-5.4.5) it can be justified, as follows, that if u, x, v are discrete points in the grid, then x(k+1) and $\nu(k+1)$ are discrete points too.

$$\begin{aligned} x(k+1) = n\Delta x + m\Delta U\tau + \frac{1}{2}l\Delta U\tau^2 = \\ \frac{1}{2}\Delta U\tau^2 + m\Delta U\tau + \frac{1}{2}l\Delta U\tau^2 = \\ \frac{1}{2}\Delta U\tau^2(n + \frac{2}{\tau}m + l) = \\ \Delta X(n + \frac{2}{\tau}m + l) \end{aligned} \tag{5.4.7}$$

Where n, m, l, τ are integer numbers.

5.4.2 Computational effort

The computational effort of an optimization method is often a significant factor that determines whether a method is being applied in practice for a given problem or not. Therefore, not only the accuracy of a solution is relevant, but also the corresponding computational effort. The number of model-function evaluations for the basic DP with an equally spaced grid, as presented in section 3.3 is given by:

$$K^{DP} = K_{\boldsymbol{x}} K_u KMAX \tag{5.4.8}$$

The variable $K_{\boldsymbol{x}}$ represents the number of grid points for the state space, K_u for the control signal, and KMAX the total number of stages.

6 Results

6.1 Results to Velocity Profile Optimization Problem Based on Pontryagin's Minimum Principle

Using the derived equations of motion via the solution of the formulated optimal control problem, several scenarios have been tested for each approach presented in section 4. The following figures present the vehicle's speed, position, and acceleration over time.
We assune that we have two scenarios where in scenario 1.1 the vehicle starts from immobility and then accelerates until it reaches the final destination point with the given final conditions. In Scenario 2.2 the vehicle starts with an initial velocity until it reaches immobility at the final destination point. Using the formulated problem from case 2, the same scenarios have been tested, but assuming that the final time is free. The derived trajectories for each scenario, seem to have the same behavior with case 1.
In Case 3 (fixed final time and a traffic light) and Case 4 (free final time and a traffic light), four different scenarios have been tested. Scenario 3.1 and 4.1 examine a resulting optimal trajectory where time T is before the optimal time (t_1), which allows the vehicle to pass through the traffic light at an optimal time and minimize the cost criterion. In contrast to the previous examination, in scenarios 3.2 and 4.2 time T is after the resulting optimal time t_1 and therefore the inequality constraint applied in the optimal control problem, force the vehicle to pass through the traffic lights at time $t_1 = T$.

6.1.1 Case 1: Analytic Solution with fixed final time & no traffic light

For the case with fixed final time and no traffic light different scenarios have been tested with the following initial and final conditions. Furthermore the weighting factor is set as w = 0.1.

Scenario	t_0	x_0	v_0	t_e	x_e	v_e
Scenario 1.1	0	0	0	50	500	16
Scenario 1.2	0	0	16	50	500	0

Table 1 Case 1 with fixed final time and no traffic light

The following figures represent the resulting optimal trajectory derived from the formulation of optimal control problem in section 4 for each scenario tested.

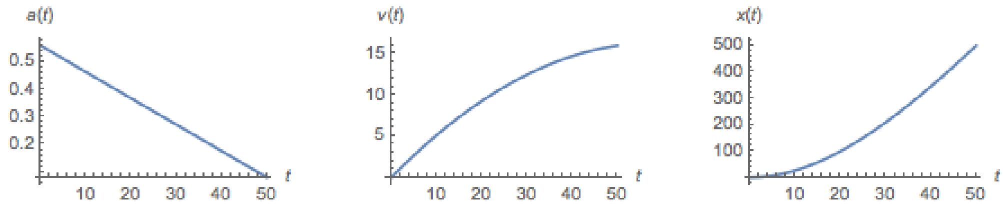


Figure 5 Performance of the Optimal Control Problem for scenario 1.1 with cost = 3.04

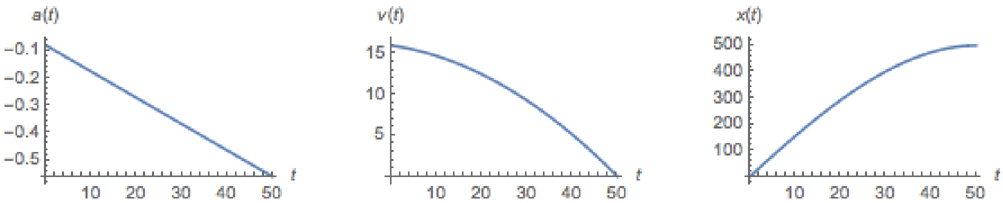


Figure 6 Performance of the Optimal Control Problem for scenario 1.2 with cost = 5.14

6.1.2 Case 2: Analytic Solution with free final time & no traffic light.

For the case with free final time and no traffic light the following scenarios have been tested with the presented initial and final conditions. Furthermore the weighting factor is set as w = 0.1

Scenario	t_0	x_0	v_0	x_e	v_e
Scenario 2.1	0	0	0	500	16
Scenario 2.2	0	0	16	500	0

Table 2 Case 2 with free final time and no traffic light

The following figures represent the resulting optimal trajectory derived from the formulation of optimal control problem in section 4 for each scenario tested.

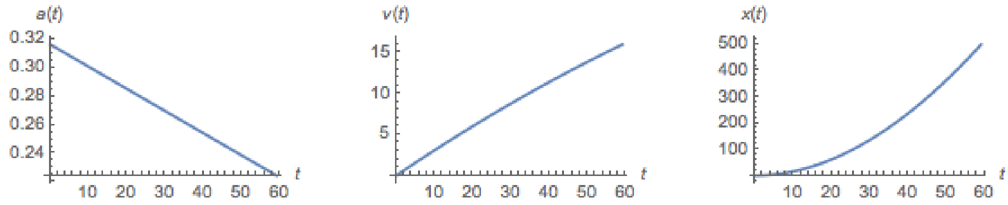


Figure 7 Performance of the Optimal Control Problem for scenario 2.1 with cost= 5.14 and $t_e = 59.1$

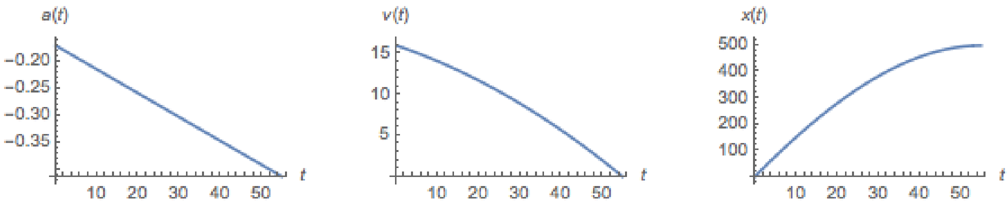


Figure 8 Performance of the Optimal Control Problem for scenario 2.2 with cost=5.2 and $t_e = 54.8$

6.1.3 The choice of weighting factors

The choice of weighting factors plays an important role for the shape of the resulting optimal trajectories. Using scenario 1.1 in section 6.1.2 for different weighting factors but identical boundary values, as the following figure shows, if as the weighting factor w increases, the final free time t_e has the opposite behaviour.

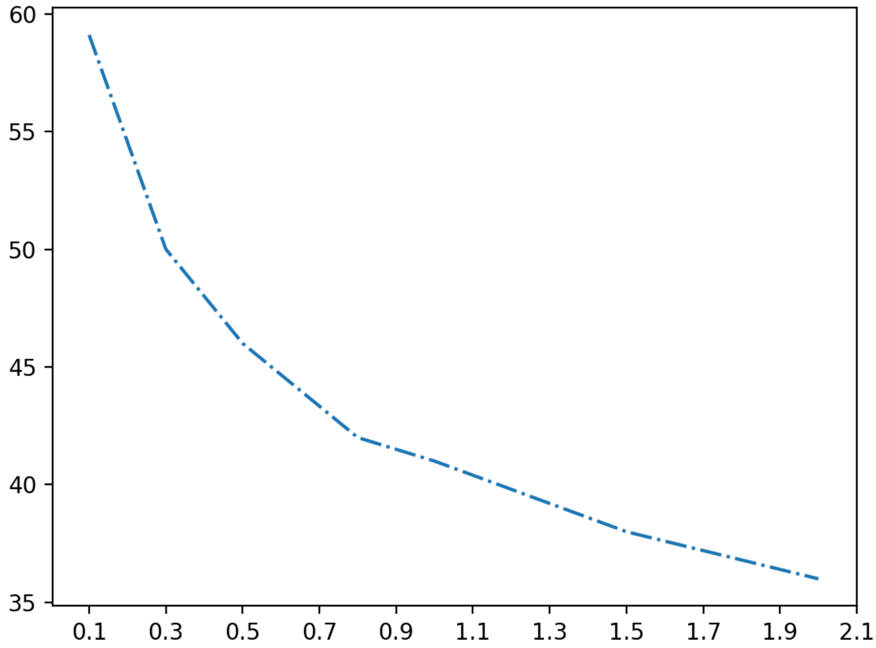


Figure 9 Graphical representation of the relationship between final time and weighting factor

6.1.4 Case 3: Analytic Solution with fixed final time & traffic light

The following table represents the scenarios that have been used in order to derive the vehicle's trajectory and speed profile for Case 3.

Scenario	t_0	x_0	v_0	x_e	v_e	x_{red}	T	t_e
Scenario 3.1	0	0	0	500	16	150	15	50
Scenario 3.2	0	0	10	500	0	150	15	50

Table 3 Case 3 with fixed final time & traffic light

The following figures represent the resulting optimal trajectory derived from the formulation of optimal control problem in section 6.1 for each scenario tested.

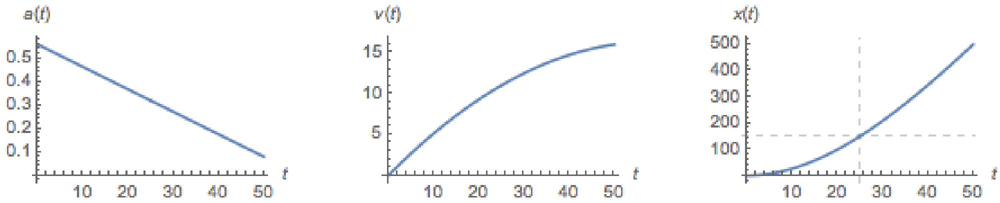


Figure 10 Performance of the Optimal Control Problem for scenario 3.1 with cost=3.04 and $t_1 = 25.0$

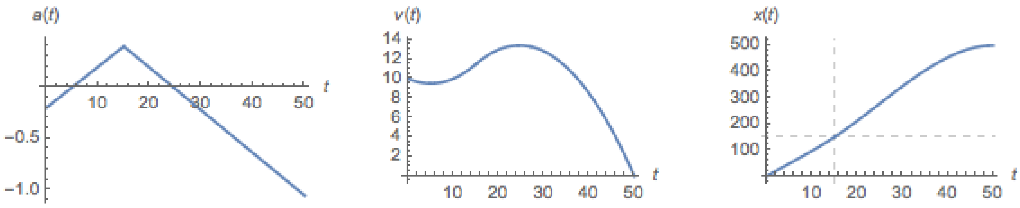


Figure 11 Performance of the Optimal Control Problem for scenario 3.2 with cost $= 5.28$ and $t_1 = 15$

6.1.5 Case 4: Analytic Solution with free final time & traffic light

The following table represents the scenarios that have been used in order to derive the vehicle's trajectory and speed profile for case 4.

Scenario	t_0	x_0	v_0	x_e	v_e	x_{red}	T
Scenario 4.1	0	0	0	500	16	150	15
Scenario 4.2	0	0	10	500	0	150	15

Table 4 Case 4 with fixed final time and a traffic light

The following figures represent the resulting optimal trajectory derived from the formulation of optimal control problem in section 6.1 for each scenario tested.

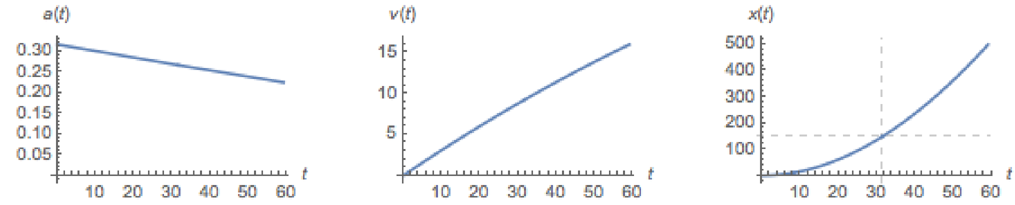


Figure 12 Performance of the Optimal Control Problem for scenario 4.1 with cost $= 5.14$, $t_e = 59.16s$ and $t_1 = 31.62s$

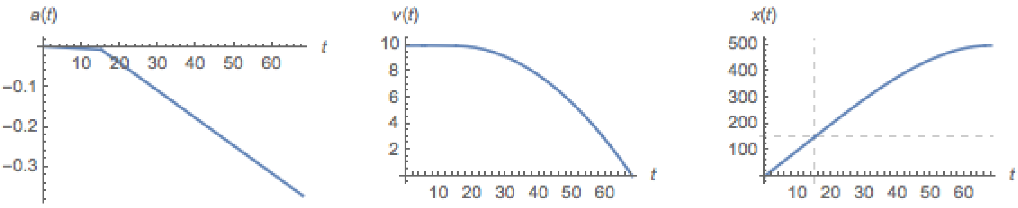


Figure 13 Performance of the Optimal Control Problem scenario 4.2 with cost $= 4.63$, $t_e = 68s$ and $t_1 = 15s$

6.1.6 Comparison of cost derived using free and fixed final time.

In case 1 and 3 the final time is fixed and the cost criterion is:

$$J = \frac{1}{2}\int_{t_0}^{t_e} u^2 dt \tag{6.1.1}$$

which consists of the integral of acceleration from the initial to final time. In case 2 and 4, where the final time is free the cost criterion is:

$$J = \frac{1}{2}\int_{t_0}^{t_e} u^2 dt + \frac{1}{2} w t_e \tag{6.1.2}$$

The first term consists of the integral of acceleration from the initial to final time and the second term consists of penalizing the derived final free time with a weighting factor w. A comparison is made using only the first term of the cost criterion for the case studies with free final time. The resulting costs from case with free final time are less, as the formulated optimal control problem derives the optimal time t_e. The following table presents the resulting costs for each scenario presented in section 6.1.

Scenarios	Cost	Scenarios	Cost	Scenarios	Cost	Scenarios	Cost
Scenario 1.1	3.04	Scenario 1.2	3.04	Scenario 3.1	3.04	Scenario 3.2	5.28
Scenario 2.1	2.18	Scenario 2.2	2.46	Scenario 4.1	2.18	Scenario 4.2	1.23

Table 5 Cost comparison for fixed and free final time

6.2 Simulation Results for dynamically managed traffic lights

6.2.1 Implementation of the Solver

The presented algorithm of Stochastic Dynamic Programming (STP) in section 5.4 has been implemented, in order to approach dynamically managed traffic lights.

The main parameters of the problem are assumed the following:

- The maximum distance to the traffic lights are:

$$x_{max} = 300m \tag{6.2.1}$$

- The maximum speed of the vehicle is:

$$\nu_{max} = 11m/s \tag{6.2.2}$$

- The input space is $U = [u_{\min}, u_{\max}]$, where:

$$u_{\max} = -u_{\min} = 3m/s^2 \tag{6.2.3}$$

- The final position of the vehicle is denoted as:

$$x_e = 350m \qquad \nu_e = 11m/s \tag{6.2.4}$$

- The minimum and the maximum traffic light's switching time is:

$$T_{\min} = 5s \qquad T_{\max} = 15s \tag{6.2.5}$$

- The state variables should obey the following linear discrete-time state equations:

$$x(k+1) = x(k) + \nu(k)\tau + \frac{\tau^2}{2}u(k) \tag{6.2.6}$$

$$\nu(k+1) = \nu(k) + u(k)\tau \tag{6.2.7}$$

- Furthermore, time has been discretized in:

$$\tau = 1s. \tag{6.2.8}$$

6.2.2 Comparative relation of discretization factor in relation to cost and computational effort

In addition to the parameters presented in section 6.2.1 the following initial conditions are assumed:

$$x_0 = 0m \qquad v_0 = 10m/s \tag{6.2.9}$$

In this section, for every scenario tested, uniform probability distribution has been used. Table 6 presents the resulting cost and the overall computing time of both SDP and optimal control problem solutions, in relation to the discretization factor that has been tested.

ΔU	ΔV	ΔX	Cost	Execution time (s)	DP complexity (10^-6)
0.5	0.5	0.25	0.93	113.85	6.0
1.0	1.0	0.5	3.98	32.21	0.8
1.5	1.5	0.75	5.80	15.55	0.28
2.0	2.0	1.0	9.12	8.75	0.11

Table 6 Comparative relation of discretization rates in relation to cost and computational effort

Figure 14 presents the derived values of cost and computing time in relation to the presented scenarios in table 6. As the discretization factor becomes higher, the cost has an upward movement, in contrast to computational time which has the opposite behavior. Taking into consideration that the solution is computed once and stored for each intersection, then an accurate solution is critical. Consequently, an efficient discretization factor of speed, acceleration and position is $\Delta U = 0.5m/s^2$, $\Delta V = 0.5m/s$, and $\Delta X = 0.25m$ respectively. In this section, for every scenario tested, uniform probability distribution has been used. Table 6 presents the resulting cost and the overall computing time of both SDP and optimal control problem solutions, in relation to the discretization factor that has been examined.

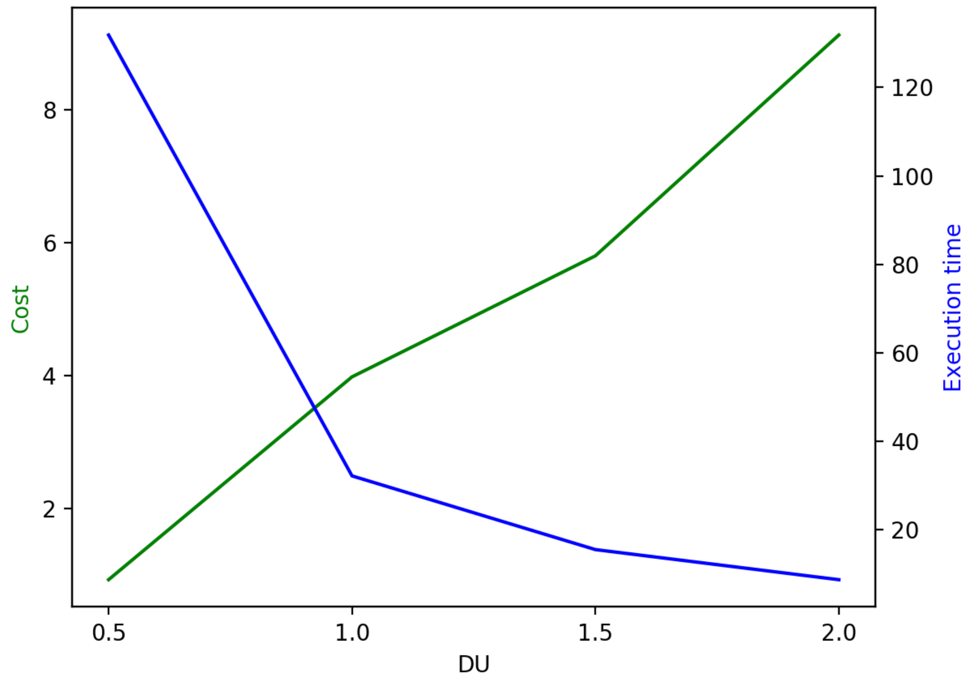


Figure 14 Cost and execution time in relation to ΔX, ΔV and ΔU.

6.2.3 Comparative relation among costs derived via Dynamic programming techniques.

In this section, a comparison is made among the costs derived from dynamic programming techniques. For each scenario tested, uniform probability distribution has been applied.

Using the implemented SDP algorithm, we can derive the expected cost (Ecost), which is resulted from the combination of all possible checkpoint times (T) with the corresponding probabilities. Furthermore from SDP algorithm, for a particular scenario, we can obtain the vehicle's trajectory cost (VTC), which consists of the amount of transition cost for every stage and then from optimal control cost until the final destination point. From the same deterministic version of the problem, considering that time T is perfectly known by the vehicle, we derive the vehicle's trajectory cost (DVTC) and a cost calculated using the resulting trajectory from the DDP algorithm, but with the corresponding probabilities in every possible time T (hybrid).

The optimal control problem (Jopt) becomes smaller as the vehicle approaches the final destination point. Due to this observation, when the traffic light changes to green in greater stages, the VTC cost is less than Ecost. In the event, that the traffic light changes to green earlier, Ecost and VTC have the opposite behavior. The cost acquired from DDP in relation to the SDP cost is smaller as the vehicle has the leverage of full knowledge, in addition to SDP which combines the minimization of all possible trajectories.
The optimal performance of SDP can be verified from the comparison of Ecost and hybrid cost. The Ecost from the implemented algorithm of SDP is smaller than Hybrid in every scenario examined, due to the fact that SDP forms only one trajectory for every possible checkpoint time.

Scenarios	x_0	v_0	$T_{\min}$	$T_{\max}$	Checkpoint time	Ecost	VTC	DVTC	Hybrid
Scenario 1	0.0	10.0	5.0	15.0	15.0	2.18	0.93	0.08	4.30
Scenario 2	0.0	7.5	5.0	15.0	15.0	2.18	3.34	3.25	3.25
Scenario 3	0.0	10.0	5.0	15.0	5.0	2.18	3.34	3.25	3.5
Scenario 4	0.0	7.5	5.0	15.0	5.0	1.52	2.60	2.31	2.32

Table 7 Comparative relation among costs derived by Dynamic programming techniques

Figures (17 & 18) represent the resulting optimal trajectories from SDP and DDP. The optimal trajectory of the same deterministic problem is more directed into reaching the position of a state at the known checkpoint time, which has the lowest optimal control cost, in addition to the trajectory derived from SDP, which combines the minimization of cost for all potential checkpoint times.

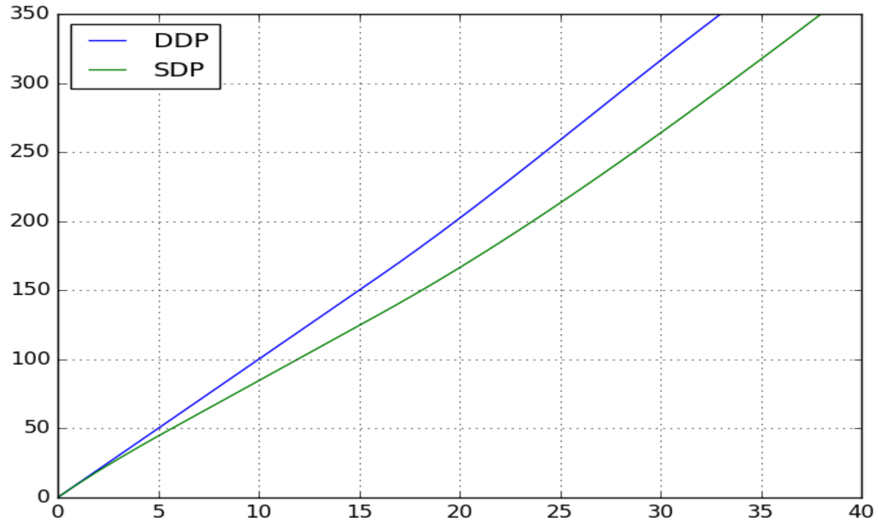


Figure 15 Optimal derived position trajectories in relation to time for scenario 1

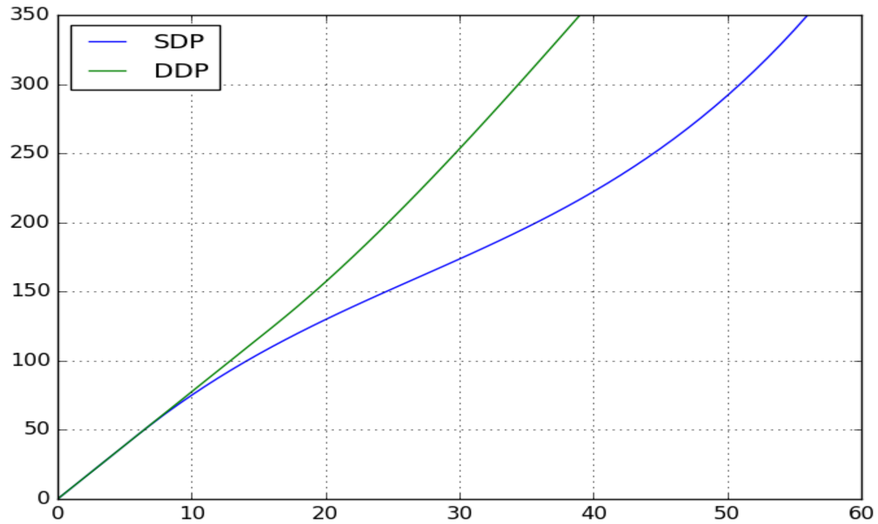


Figure 16 Optimal derived position trajectories in relation to time for scenario 2

6.2.4 Comparison of optimal trajectories for different probability distributions

In this section some notable scenarios are examined in the implemented algorithm of SDP for different probability distributions, based on the derived equations of section 5.2. The scenarios examined are presented in table 8

Scenarios	x_0	v_0	$T_{\min}$	$T_{\max}$	Checkpoint time
Scenario 1	0.0	9.25	5.0	15.0	15.0
Scenario 2	0.0	11.0	5.0	15.0	15.0

Table 8 Scenarios tested for several probability distributions

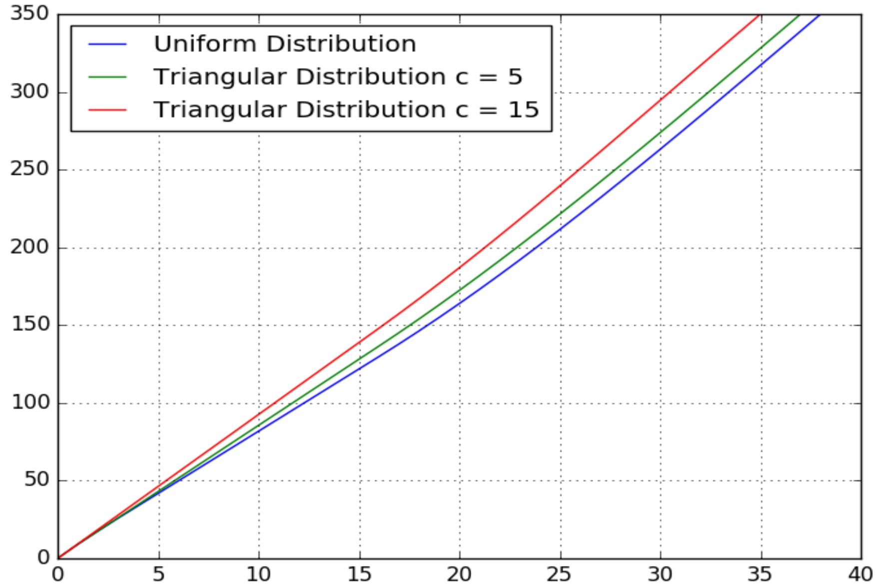


Figure 17 Optimal derived position trajectory in relation to time with different probability distributions for scenario 1

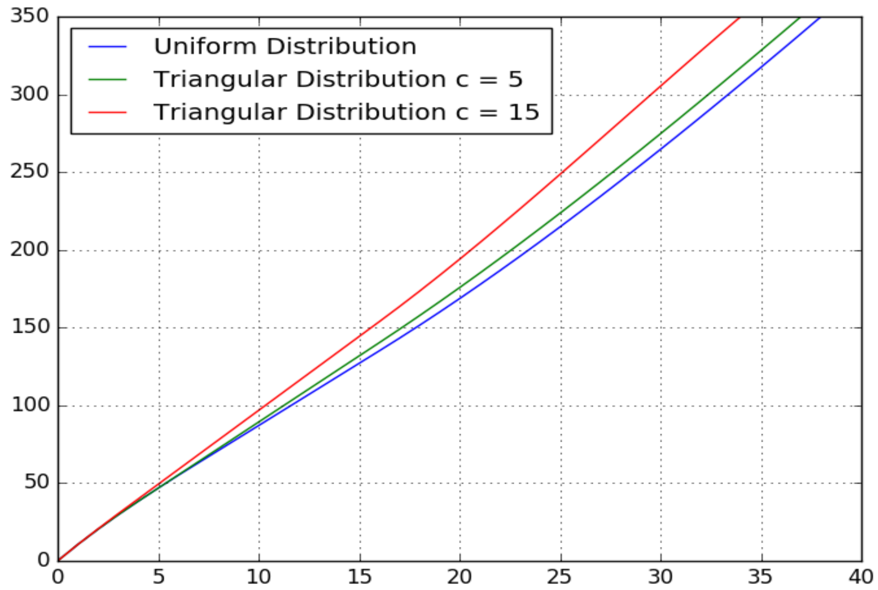


Figure 18 Optimal derived position trajectory in relation to time with different probability distributions for scenario 2

The previous figures present the resulting vehicle's trajectory applying uniform and triangular probability distribution. From the initial point until the traffic light's switching time (T) the trajectory is derived from the implemented algorithm of SDP, and after that until the final destination point the formulated optimal control problem, presented in section 4, has been used. Applying uniform probability distribution, where every stage between $T_{\min}$ and $T_{\max}$ has the same probability of switching to green, the vehicle develops an optimal strategy in order to combine the minimization of cost criterion for each possible checkpoint T. Regarding triangular distribution, the resulting trajectory has different behavior for various modes (c). Considering that the mode time is c=5, the vehicle tries to create a trajectory, in order to pass the traffic light at the given mode value. More specifically, in the event that the given mode value is c=15 the resulting trajectory is more determined into achieving the state with the lowest optimal control cost in T=15. In overall,the simulation results clearly demonstrate that a full-horizon SDP solution has a smooth trajectory, in addition to an uninformed driver, who would stop and start more often at traffic lights.

7 Contributions and future work

This thesis presents the solutions to the vehicle's velocity profile in order to decrease the fuel consumption in different driving situations. In section 4 different cases with fixed and free final time traffic lights are presented. An appropriate optimal control problem is formulated for different traffic scenarios and solved analytically via PMP. In section 5 we propose and implement and efficient and accurate algorithm of SDP for the case of the existence of dynamically managed traffic lights.

As a future work, our goal is to extend the estimation of real time traffic information, in the presence of accident, road work or any kind of traffic incidence, then to incorporate these real time information with the static information obtained from the traffic light and generate a more comprehensive optimal velocity profile. Currently, generated optimal velocity profile is only suggested to the driver. In that framework, the driver is indeed the controller that follows the suggested speed. In the future the advised velocity profile can be fed to the advanced cruise control (ACC) system for better following of the optimal speed profile.

Another approach would be to extend the use of speed advisory system (SAS) for real time information traffic conditions. For example, when there is an accident on the road the driver would not be able to follow the advised optimal speed. Therefore the system needs to track the driven velocity profile and in the case when the driver can not follow the optimal speed, it should be re-calculated. The re-calculation can be achieved by the vehicle, using SDP techniques with more efficient complexity (e.g Differential Dynamic Programming).

References

[1] Konstantinos Vasileios Katsaros, Ralf Kernchen, Mehrdad Dianati, and David Rieck. Performance study of a green light optimized speed advisory (glosa) application using an integrated cooperative its simulation platform. *2011 7th International Wireless Communications and Mobile Computing Conference*, pages 918–923, 2011.

[2] Joint Program Office Intelligent Transportation Systems. Intelligent Transportation Systems - Vehicle to Infrastructure (V2I) Deployment Guidance and Resources.

[3] Volpe center. Vehicle-to-Vehicle Communication.

[4] ABI Research. Connected Vehicle Cloud Platforms.

[5] CAAT Staff. Automated and connected vehicles.

[6] Siemens. Future of Infrastructure: Vehicle-to-X (V2X) communication technology.

[7] Marco Wiering, Jacqueline A. C. Vreeken, J. van Veenen, and Arne Koopman. Simulation and optimization of traffic in a city. *IEEE Intelligent Vehicles Symposium, 2004*, pages 453–458, 2004.

[8] Victor Gradinescu, Cristian Gorgorin, Raluca Diaconescu, Valentin Cristea, and Liviu Iftode. Adaptive traffic lights using car-to-car communication. *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, pages 21–25, 2007.

[9] Markos Papageorgiou. *Optimierung.* Springer, 1991.

[10] Arthur Earl Bryson. *Applied optimal control: optimization, estimation and control.* CRC Press, 1975.

[11] Marinaki M. Typaldos P. Makantasis K. Papageorgiou, M. A feasible direction algorithm for the numerical solution of optimal control problems- extended version. 1995.

[12] Ioannis Ntousakis, Ioannis Nikolos, and Markos Papageorgiou. Optimal vehicle trajectory planning in the context of cooperative merging on highways. 71, 10 2016.

[13] Papamichail I. Papageorigiou M. Typaldos, P. Minimization of fuel consumption for vehicle trajectories. In *Proceedings of the 97th Annual Meeting of the Transportation Research Board (TRB), Washington, D.C., USA, paper no. 18-03953.*

[14] Andreas Lawitzky, Dirk Wollherr, and Martin Buss. Energy optimal control to approach traffic lights. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4382–4387. IEEE, 2013.

[15] Ciprian Dobre, Adriana Szekeres, Florin Pop, and Valentin Cristea. Intelligent traffic lights to reduce vehicle emissions. 05 2012.

[16] R Braun, Fritz Busch, Carsten Kemper, R Hildebrandt, F Weichenmeier, Cornelius Menig, I Paulus, and R Presslein-Lehle. Travolution-netzweite optimierung der lichtsignalsteuerung und lsa-fahrzeug-kommunikation. 53:365–374, 01 2009.

[17] Steven Lee Fassino. Optimal control of differential equations with pure state constraints. 2013.

[18] Engin Ozatay. *Driving Profile Optimization for Everyday Driving*. PhD thesis, The Ohio State University, 2014.

[19] Bart De Schutter and Bart De Moor. Optimal traffic light control for a single intersection. *European Journal of Control*, 4(3):260–276, 1998.

[20] Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2012.

[21] Emmanouil Koukoumidis, Li-Shiuan Peh, and Margaret Rose Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 127–140. ACM, 2011.

[22] David Eckhoff, Bastian Halmos, and Reinhard German. Potentials and limitations of green light optimal speed advisory systems. In *Vehicular Networking Conference (VNC), 2013 IEEE*, pages 103–110. IEEE, 2013.

[23] Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.

[24] José Javier Anaya, Pierre Merdrignac, Oyunchimeg Shagdar, Fawzi Nashashibi, and José E Naranjo. Vehicle to pedestrian communications for protection of vulnerable road users. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 1037–1042. IEEE, 2014.

[25] Matlab optimization toolbox, 2017.

[26] Python 3.6.5 documentation, 2018.

8 Appendix A

This appendix includes the figures that represent the vehicle's position, speed and acceleration in each time step for each scenario tested in section 6.2.4.

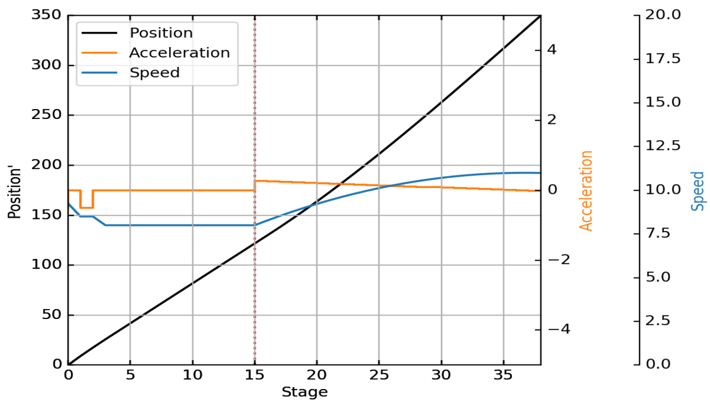


Figure 19 Optimal derived trajectory in relation to time using uniform probability distribution for scenario 1.

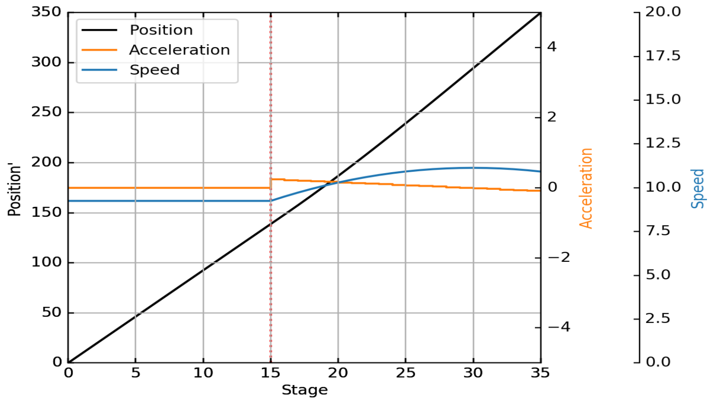


Figure 20 Optimal derived trajectory in relation to time using triangular probability distribution with c=5 for scenario 1.

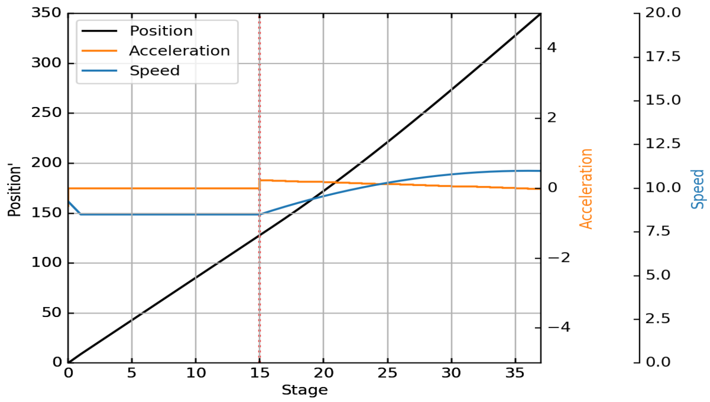


Figure 21 Optimal derived trajectory in relation to time using triangular probability distribution with c=15 for scenario 1.

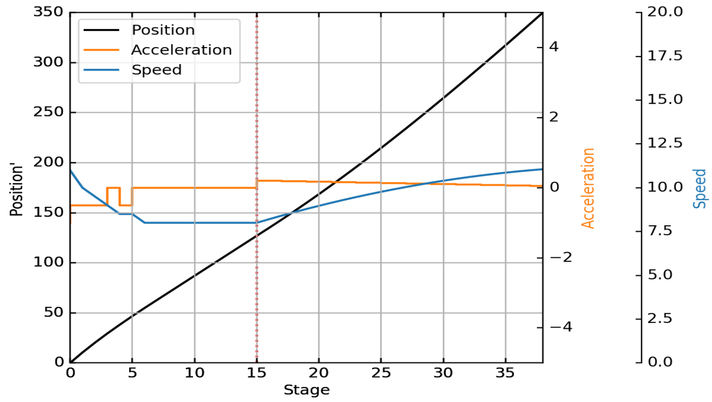


Figure 22 Optimal derived trajectory in relation to time using uniform probability distribution for scenario 2.

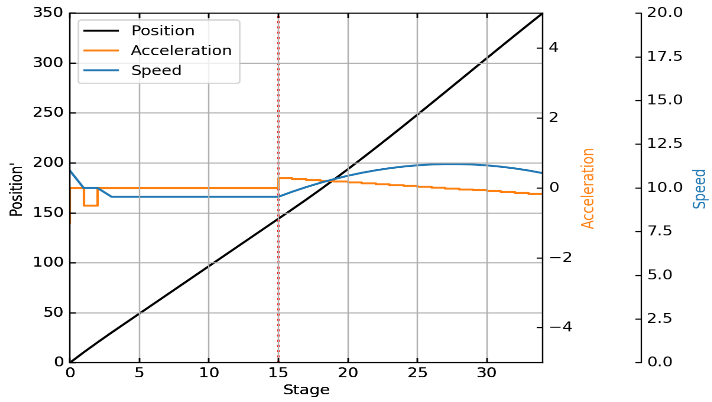


Figure 23 Optimal derived trajectory in relation to time using triangular probability distribution with c=5 for scenario 2.

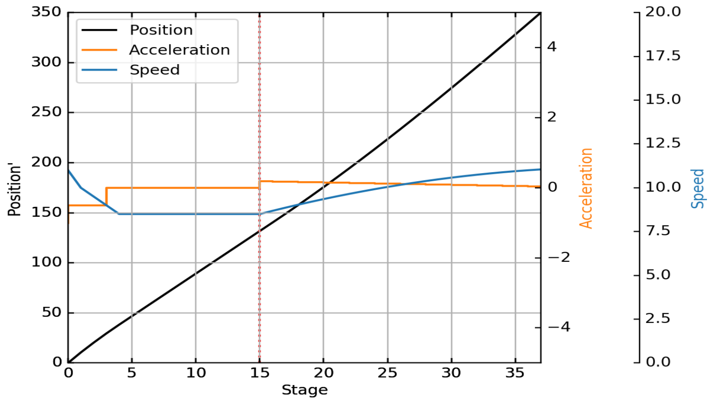


Figure 24 Optimal derived trajectory in relation to time using triangular probability distribution with c=15 for scenario 2.