Optimal Control of a Commuter Train Considering Traction and Braking Delays

OPTIMAL CONTROL OF A COMMUTER TRAIN CONSIDERING TRACTION AND BRAKING DELAYS

BY

MUZAMIL RASHID ,

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2017) McMaster University
(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE: Optimal Control of a Commuter Train Considering Traction and Braking Delays

AUTHOR: Muzamil Rashid
B.Eng.Mgt., (Electrical Engineering and Management)
McMaster University, Hamilton, Ontario, Canada

SUPERVISOR: Dr. Shahin Sirouspour

NUMBER OF PAGES: xxviii, 200

To my family

Abstract

Transit operators are increasingly interested in improving efficiency, reliability, and performance of commuter trains while reducing their operating costs. In this context, the application of optimal control theory to the problem of train control can help towards achieving some of these objectives. However, the traction and braking systems of commuter trains often exhibit significant time delays, making the control of commuter trains highly challenging. Previous literature on optimal train control ignores delays in actuation due to the inherent difficulty present in the optimal control, and in general, the control, of input-delay systems.

In this thesis, optimal control of a commuter train is presented under two cases: (i) equal, and (ii) unequal time delays in the train traction and braking commands. The solution approach uses the economic model predictive control framework, which involves formulating and solving numerical optimization problems to achieve minimum mixed energy-time optimal control in discretized spatial and time domains. The optimization problems are re-solved repeatedly along the track for the remainder of the trip, using the latest sensor measurements. This would essentially establish a feedback mechanism in the control to improve robustness to modelling errors. A key feature of the proposed methods is that they are model-based controllers, they explicitly incorporate model information, including time delays, in controller synthesis hence avoiding performance degradation and potential instability. To address the issue of input-delays, the well-established predictor approach is used to compensate for input-delays. The case of equal traction-braking delays is treated in discretized spatial domain, which uses an already developed convex approximation to the optimization problem. The use of the convex approximation allows for robust and rapid computation of the optimal control solution. The non-equal traction-braking delays scenario is formulated in time domain, leading to a nonconvex optimization problem. An alternative formulation for minimum-time optimal control problems is presented for delay-free systems that simplifies the solution of minimum-time optimal control problems compared to conventional minimum-time optimal control formulations. This formulation along with the predictor approach is used to help solve the train optimal control problem in the case of non-equal traction-braking delays. The non-equal traction-braking delay controller is compared with the equal traction-braking delay controller by insertion of an artificial delay to make the shorter delays equal to the longer delay. Results of numerical simulations demonstrate the validity and effectiveness of the proposed controllers.

Acknowledgements

The author would like to thank his supervisor, Dr. Shahin Sirouspour, for the continued guidance, support, and dedication, and would also like to thank his lab colleagues, especially Dennis Yazhemsky, for his advice, suggestions, and feedback.

The author would also like to thank Natural Sciences and Engineering Research Council (NSERC) and Thales Canada, Transportation Solutions, Inc (TCTS) for funding this research.

The author would also like to thank his family for their continuing support and encouragement, without which this thesis would not have been possible.

Acronyms

ATC	Automatic Train Control. 26–28
ATO	Automatic Train Operation. 26–28, 37
ATP	Automatic Train Protection. 26–28
ATS	Automatic Train Supervision. 26, 27
CPU	Central Processing Unit. 99
DDP	Discrete Dynamic Programming. 36
EMPC	Economic Model Predictive Control. iv, 9, 25, 26, 174
GA	Genetic Algorithm. 37
IPOPT	Interior Point Optimizer. 96, 97
KKT	Karush-Kuhn-Tucker Conditions for Optimality. 43, 45
MILP	Mixed Integer Linear Programming. 36
MIMO	Multiple-Input Multiple-Output. 22
MINLP	Mixed Integer Nonlinear Linear Programming. 56, 57
MPC	Model Predictive Control. 11, 20–22, 25, 30, 31, 35, 98

NLP Nonlinear Programming. 19, 57, 96

OS Operating System. 99

PID Proportional-Integral-Derivative Control. 28–32

PSOPT Pseudospectral Optimal Control Solver. 36

RHC Receding Horizon Control. 35

SQP Sequential Quadratic Programming. 35, 44

Notation

C	Resistance force, function of velocity. 47–50, 54, 58, 60, 62, 63, 78, 83, 84, 114, 115
C_0	Constant of Davis formula. 47, 66, 67, 100, 113, 179, 181
C_v	Coefficient of Davis formula, proportional to velocity. 47, 66, 67, 100, 113, 179, 181
C_{v^2}	Coefficient of Davis formula, proportional to the square of velocity, i.e. v^2. 47, 66, 67, 100, 113, 179, 181
Δt	Time discretization interval. 56, 57, 87
g	Component of the force of gravity due to the track grade, function of position, i.e. $g(s)$. Can also be a compound function of time since the position itself can be a function of time, i.e. $g(s(t))$. 47–50, 54, 58, 60, 62, 63, 66, 67, 78, 83, 84, 114, 179, 181
γ	Units of inverse velocity.. 65–67, 179–182
γ^+	Units of inverse velocity.. 65–67, 181
h	Time scale, used in the case of the *tau*-domain. Determines the

total trip time.. 77–82, 84–90, 114

J General objective function. 53, 57, 61, 63, 77, 79, 82, 86

k_P Constant greater than 1. Used in the approximation of the field weakening region.. 51, 113, 179, 180

m Mass of the train. 47–50, 54, 58, 60, 62, 63, 66, 67, 78, 83, 84, 99, 113, 179, 181

$\mu_{\mathbf{brk,max}}$ Maximum change in braking. 62, 63, 83, 84, 87, 88

$\mu_{\mathbf{brk,min}}$ Minimum change in braking. 62, 63, 83, 84, 87, 88, 113

$\mu_{\mathbf{max}}$ Maximum change in control input. 54, 59, 78, 80

$\mu_{\mathbf{min}}$ Minimum change in control input. 54, 59, 78, 80

$\mu_{\mathbf{trk,max}}$ Maximum change in traction. 62, 63, 83, 84, 87, 113

$\mu_{\mathbf{trk,min}}$ Minimum change in traction. 62, 63, 83, 84, 87, 88, 113

P Maximum power (constant). 51, 113, 179, 180

$p_{\mathbf{in}}$ Power input due to traction, is equal to: $p_{\text{in}}[k] = u_{\text{trk}}[k]\ v[k]$.. x, 89

$\pi_{\triangle\mathbf{ctrl}}$ Change in control, the difference between the current control and next control divided by the time-interval between them. 89, 90

r_0 Constant in quadratic polynomial that is used as a lower bound on the maximum tractive force vs speed curve. The polynomial is a function of the train velocity. Used in the convex approximation.. 66, 67, 100, 182

r_v	Coefficient of linear velocity term in quadratic polynomial that is used as a lower bound on the maximum tractive force vs speed curve. The polynomial is a function of the train velocity. Used in the convex approximation.. 66, 67, 100, 182
r_{v^2}	Coefficient of squared velocity term, v_2, in quadratic polynomial that is used as a lower bound on the maximum tractive force vs speed curve. The polynomial is a function of the train velocity. Used in the convex approximation.. 66, 67, 100, 182
ρ	Units of inverse velocity.. 65–68, 179–182
s	Position. 47–50, 53–55, 57–68, 70, 71, 77–80, 82–87, 89, 94, 99, 100, 113, 114, 178, 181
s_0	Initial position. 10, 53, 57, 59, 61, 63, 65, 78, 79
$S_{\mathbf{brk,delay}}$	Braking position delay, constant. 50, 63, 64
$S_{\mathbf{delay}}$	Position delay, constant, applies in the case of equal traction and braking delays. 49, 60, 70
$s_{\mathbf{f}}$	Final position. 53, 58, 59, 61, 63, 65–67, 71, 78, 79, 84, 86, 94, 178, 181
$s_{\mathbf{max}}$	Upper bound position constraint at a given time. 54, 62, 78, 79, 83, 84, 86
$s_{\mathbf{Milepost}}$	Milepost markers that denote the fixed distances at which the train optimal control problem is re-solved.. 86, 91, 94, 113
$s_{\mathbf{min}}$	Lower bound position constraint at a given time. 54, 62, 78, 79, 83, 84, 86

$S_{\mathbf{Next}}$ Next position discretization node greater than the current position. 71, 94

$s_{\mathbf{Predicted}}$ Predicted position. 65, 66, 71, 83, 86, 94, 178, 181

$S_{\mathbf{trk,delay}}$ Traction position delay, constant. 50, 63, 64

σ Regenerative braking recovery factor, i.e. the portion of braking energy that is recovered. 55, 59

t Time. 47–50, 53–55, 57–67, 69–71, 76–80, 82–86, 88, 91, 94, 99, 113, 178, 179, 181

t_0 Initial time. 53, 55, 57, 61, 63, 76–79

$T_{\mathbf{brk,delay}}$ Braking time delay, constant. xxii–xxviii, 48, 50, 62, 64, 91, 94, 101–112, 115–134, 136, 138–143, 145–148, 150–154, 156, 157, 159–163, 165–171

$T_{\mathbf{delay}}$ Time delay, constant, applies in the case of equal traction and braking delays. 48, 49, 60, 69–71

$t_{\mathbf{f}}$ Final time. 53, 55, 61, 76, 77

$t_{\mathbf{f,max}}$ Maximum scheduled arrival time. 53, 58, 61, 63, 78, 79, 83, 86

$t_{\mathbf{f,min}}$ Minimum scheduled arrival time. 53, 58, 61, 63, 78, 79, 83, 86

$t_{\mathbf{max}}$ Maximum scheduled arrival time for a given position. 58, 63, 65, 67, 100, 113, 178, 181

$t_{\mathbf{min}}$ Minimum scheduled arrival time for a given position. 58, 63, 65, 67, 178, 181

$T_{\mathbf{Min\text{-}Delay}}$ Minimum of the traction and braking delays. 91, 94

$t_{\mathbf{Predicted}}$ Time at the end of the predicted region. 65, 66, 71, 83, 85, 86, 91, 92, 94, 178, 181

$t_{\mathbf{qe}}$	Time at the end of all control buffer regions. 83–85, 92
$T_{\mathbf{S}}$	Control input sample time. 69, 71, 89–94, 113, 114
$t_{\mathbf{Sample\ Start}}$	Time to determine a control command using interpolation to apply to the system. 71, 94
$t_{\mathbf{Start}}$	Time at which to start model-based prediction for delay compensation.. 71, 94
$T_{\mathbf{trk,delay}}$	Traction time delay, constant. xxii–xxviii, 48, 50, 62, 63, 91, 94, 101–112, 115–134, 136, 138–143, 145–148, 150–154, 156, 157, 159–163, 165–171
τ	Pseudo-Time. 76–80, 82–85, 87–90, 92, 93, 95, 114
u	Control input. xiii, 47–49, 53–55, 57–60, 77–80, 82, 85, 90, 114, 178–180
$\bar{u}_{\mathbf{brk}}$	Braking control buffer samples, fixed.. 83, 87
u^+	Positive portion of control input, defined as: $u^+ = \max\{u, 0\}$. xiii, 178, 179
$u_{\mathbf{brk}}$	Braking control input. 48–50, 61–63, 83, 84, 86–89, 94
$u_{\mathbf{brk,max}}$	Maximum braking. 62, 63, 66, 67, 83, 84, 86, 100, 113, 179, 180, 182
$u_{\mathbf{max}}$	Maximum control input. 54, 58, 78, 80
$u_{\mathbf{min}}$	Minimum control input. 54, 58, 78, 80
$u_{\mathbf{trk}}$	Traction control input. x, 48–50, 61–63, 83–89, 94
$U_{\mathbf{trk,max}}$	Upper bound on the tractive force (constant). 51, 113, 179, 180
$u_{\mathbf{trk,max}}$	Maximum traction. 51, 62, 63, 66, 67, 83, 84, 86, 100, 182
v	Velocity. x, 47–51, 53–55, 57–63, 65–68, 71, 77–80, 82–87, 89,

94, 100, 113–115, 178–182

v_0 Initial velocity. 53, 57, 61, 63, 78, 79

$v_{\mathbf{max}}$ Maximum permitted speed at a certain position. 54, 58, 62, 63, 66, 67, 78, 79, 83, 84, 86, 100, 113, 179, 182

$v_{\mathbf{Predicted}}$ Predicted velocity. 65, 66, 71, 83, 86, 94, 178, 181

w_ρ Weight on inverse velocity term, this is used in order to get the pseudo-inverse velocity to be almost equal to to the inverse velocity (for the dynamics to hold). 65, 66, 68, 100, 107, 110, 140, 154, 169, 181

$w_{\Delta\mathbf{ctrl}}$ Weight on the change in control, used to ensure a smooth control for purposes of passenger comfort and reduce equipment wear/tear. 89, 90, 115, 125, 137, 144, 149, 159, 164

w_e Weight on energy consumption. 65, 66, 89, 100, 107, 110, 115, 125, 137, 140, 144, 149, 154, 159, 164, 169, 178, 181

w_t Weight on trip travel time. 89, 115, 125, 137, 144, 149, 159, 164, 178

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction and Problem Statement

1.1 Background

Transit rail transports millions of passengers globally, and passenger volume is expected to grow [1]. There is increasing pressure to improve the performance, reliability, and efficiency of commuter trains. Transit rail systems are facing increasing use, wear, and tear due to higher passenger volumes as evidenced in Table 9.12 in [2]. These systems are also facing pressure to reduce costs and to improve efficiency due to increasing concern regarding environmental issues. Train control systems play a role in improving performance and reliability by allowing trains to run faster and more frequently. Automated train control systems may improve safety by reducing the risk of human error. Optimal train control systems can also run in an energy and mechanical wear and tear optimal manner, thereby improving efficiency and reducing costs.

Train control systems are concerned with determining the traction and braking commands that will drive the train from station A, the origin, to station B, the destination. Train control systems must be able to manage the dynamics of the system which are nonlinear, time-varying, and have time-delays inherent in the input(s), and be able to maneuver the states of the system to follow the desired trajectory. Train control systems must also operate within the realm of safety restrictions and passenger comfort requirements. Safety restrictions include track speed limits, safe separation distance between two neighbouring trains operating on the same section of track, and safe equipment operation. Passenger comfort requirements include comfortable levels of traction and braking, smooth changes in traction and braking, smooth transitions between traction and braking, continuity of the control, and avoiding the use of discontinuous control trajectories. Safety restrictions and passenger comfort requirements serve as constraints on train motion and actuation, and must be considered when determining the control.

1.2 Motivation

There is a push to improving the performance and efficiency of transit systems, and thus train control systems. This is due to the following reasons:

- economic: high cost of fuel
- environmental: pollution and greenhouse gas emissions
- increased demand: an increasing population and a trends towards urbanization resulting in increased passenger demand

One of the ways of improving train control system performance and efficiency is by use of optimal controllers. Conventional train control systems used in industry are not optimal. These train control systems were developed at a time of scarcity of on-board computational resources. These control systems were mainly concerned with appropriately controlling the nonlinear dynamics of the train system in order to get the resultant velocity profile to follow the desired velocity profile. These controllers do not plan the entire state or control trajectory, rather these controllers are focused on determining the best control input to apply at the current time instant in order to minimize the error between the actual and the desired state trajectory, based on the current measurements. These controllers have the advantage that they require less information about the model, are simpler to implement, and require less computational and memory resources.

Optimal train control systems determine the traction and braking commands that will drive the train from station A to station B while minimizing some cost function. Common examples of such controllers are minimum-time, minimum-energy without regenerative braking, and minimum-energy with regenerative braking. This thesis considers a mixture of time and energy as the cost function to minimize. While regenerative braking is not specifically addressed here, it is rather trivial to extend the proposed minimum-energy with no regenerative braking formulation to the minimum-energy with regenerative braking case.

Currently, industry solutions that offer optimal train control systems to rail operators are driver assistance systems [3]. These systems offer advice to train drivers, suggesting to the drivers the speed to maintain, in order to improve performance or reduce costs. For example, such a system provides a suggested speed to the driver, which the driver tries to maintain, in order to minimize fuel consumption. Current industry solutions that offer optimal train control systems are not autonomous , i.e., these systems do not directly drive the train and do not remove the need for a human driver.

These optimal train control systems are unable to drive the train directly because the train dynamics considered in the optimal train control formulation may not consider all the dynamics or even all the dynamics that have a significant impact on train motion. There can be significant model mismatch between the dynamics mathematical model and the actual train dynamics. Some examples of such causes of model mismatch include: not accurately knowing the grade (force of gravity tangential to the track), failing to model the resistance forces properly, not accurately considering the maximum traction and braking limits, not considering limitations on the rate of change of traction or braking, and delays in the traction and braking actuation systems. The focus of this thesis is on a specific cause that results in significant model mismatch: the presence of delays in traction and braking actuation systems. Current optimal train control systems do not consider actuation delays in the dynamics [3]. Optimal train control schemes can be deployed in an autonomous set-up if delay compensation is implemented in the optimal train control scheme.

Time-delays in actuation are everywhere in physical systems due to the fact that there are physical delays present in the hardware and equipment. The delays in the traction and braking control inputs is due to the equipment and hardware used to implement the traction and braking systems. The equipment and hardware suffers from dead-time, reconfiguration time, and warm-up time. Implementing optimal controllers, that do not compensate for delays, in an autonomous configuration may suffer from degraded performance and reduced robustness. Implementing an optimal train controller autonomously requires consideration of the dynamics and behaviours

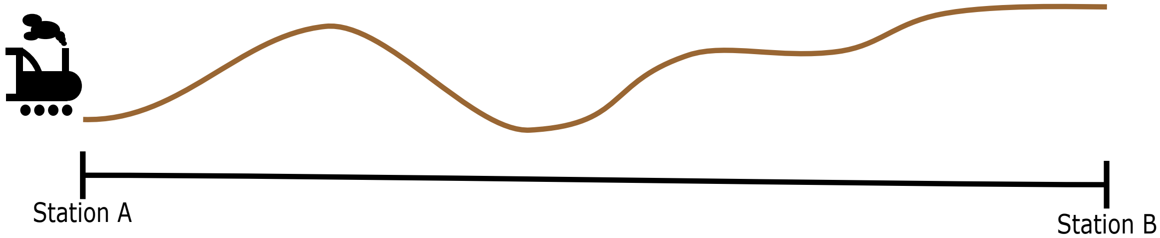

Figure 1.1: An optimal controller with delay compensation for a commuter train system.

that have a substantial effect, such as input time-delays.

1.3 Problem Statement

The problem being addressed in this thesis is the synthesis of a train optimal controller that compensates for delays. The purpose of the controller is to autonomously and optimally drive the train from station A, the origin, to station B, the destination, i.e. determine the traction and braking commands to move the train from station A to station B. This is shown in Fig. 1.1. The controller must satisfy any constraints due to safety restrictions and passenger comfort requirements. The controller must be able to adequately influence the complex dynamical behaviour of the system. The controller must also be optimal, i.e. operate the system in a manner to achieve better performance and efficiency.

The difficulty in the autonomous control of a train is compounded by the presence of delays in the traction and braking systems, and the nonlinear time-varying nature of the system. In this thesis, only the nonlinear system dynamics and the presence of time-delays in the traction and braking systems are considered. The time-varying nature of the system is not directly addressed. The presence of delays makes controller design challenging, especially optimal controller design. Delays in input actuation can degrade performance, reduce efficiency, and increase costs. The presence of delays is one of the reasons optimal controllers are not used currently to autonomously drive the train. Instead, currently optimal controllers are used to provide guidance to the driver as to the speed that should be maintained to improve performance. The problem of delays has two cases: (i) equal traction-braking delays, and (ii) nonequal traction-braking delays. Both of these problems will be addressed in this thesis.

The proposed controller must include the following features and requirements:

- optimal
- model-based
    - The controller is required to be model-based so that the controller can include all available plant and environment information, and thus achieve maximum improvement in performance, robustness, and reliability.
- ability to handle complex constraints, such as constraints on the states and inputs
- ability to handle a variety of cost functions
- ability to handle the full-range of operating conditions, e.g. maximum uphill/downhill grades, long/short travel times, long/short trips, etc.
- real-time (or close to real-time)
- robustness in determining a solution, i.e. the control values to apply to the system

1.4 Thesis Contributions

Two new optimal controllers are proposed for commuter train operation that can compensate for delays in traction and braking. The commuter train system is treated as a nonlinear system. The controllers are model-based, they explicitly incorporate the following model information: state evolution dynamics (state evolution differential equations), resistance forces, force of gravity tangential to the track, maximum traction limits, maximum braking limits, maximum rate of change of traction, minimum rate of change of traction, maximum rate of change of braking, minimum rate of change of braking, time-delays in the input.

The controllers use an economic model predictive control framework where the control problem is solved repeatedly along the track using the latest sensor measurements, as shown in Fig. 1.2. Given a train at any position along the track, the states of the train are measured, an optimal control problem is solved to determine the state and control trajectory from the current position to the end of the trip. From the given solution of the optimal control problem, the first few samples of the control are applied and the process is repeated until the train arrives at the destination. In other words, the control strategy is to solve a sequence of optimal control problems based on the latest sensor measurements. The optimal control problem, which is in continuous-time, is discretized to obtain a finite-dimensional optimization problem. The finite dimensional optimization problem is solved using optimization solvers.

Input time-delay compensation is achieved by the use of a predictor, which is a well established concept for addressing the problem of delays [4]–[11]. The predictor "predicts" the system state based on the current state and the past control history. Feedback is then based not directly upon the current state, but rather the "predicted

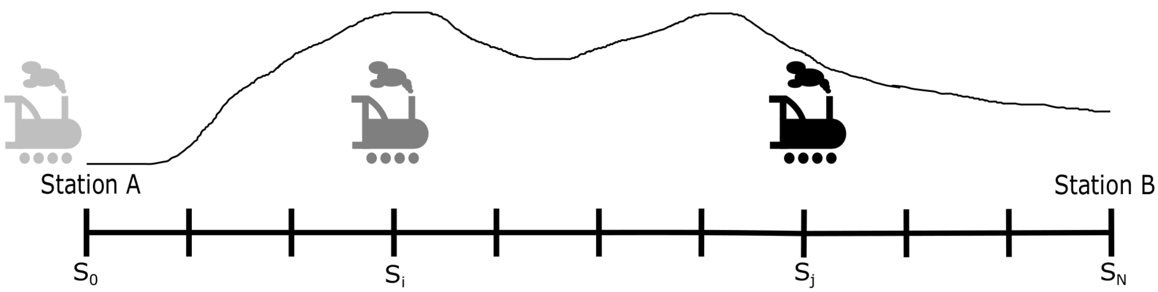

Figure 1.2: Optimal control of a train. The goal is to move the train from station A to station B while minimizing an objective function.The position, velocity, acceleration, traction effort, and braking effort profiles for the remainder of the trip are determined by formulating and solving optimal control (optimization) problems repeatedly at discrete points along the track, e.g. s_0 (starting point), s_i, s_j, etc.

state".

The controllers developed are described below.

- Controller for equal traction-braking delays that uses an existing convex approximation for robust and rapid solution of the optimization problem [1].

- Controller for non-equal traction-braking delays that results in a nonconvex optimization problem. An alternative formulation for minimum-time optimal control problems is developed that is used to help make this case tractable. Along with this, the past control history also plays a role in the optimization problem formulation.

[1]Note: the convex approximation has been developed previously, see Yazhemsky [12], and Yazhemsky *et al.* [13]. The convex approximation is not a contribution of this thesis.

1.5 Organization of the Thesis

The remainder of this thesis is organized as follows. In Chapter 2 a review of the literature on optimal control, train control, optimal train control, and model predictive control, time-delay systems, and optimization is presented. In Chapter 3 the system dynamics are presented for each of the following cases: no traction-braking delays, equal traction-braking delays, non-equal traction-braking delays. In Chapter 4 the optimal control problems are presented for each of the following cases: no traction-braking delays, equal traction-braking delays, non-equal traction-braking delays. In Chapter 5 a formulation is presented for optimal control of a commuter train system subject to equal delays in traction and braking. In Chapter 6 an alternative formulation for minimum-time optimal control problems of a commuter train with no traction-braking delays is presented. In Chapter 7 a formulation is presented for optimal control of a commuter train system subject to non-equal delays in traction and braking. In Chapter 8 the computer implementation setup is detailed and results are provided. In Chapter 9 the thesis is concluded and future work directions are highlighted.

1.6 Related Publications

1.6.1 Journal Articles

1.6.1.1 Submitted

D. Yazhemsky *et al.*, "An on-line optimal controller for a commuter train," *IEEE Intelligent Transportation Systems Transactions*, Submitted

1.6.1.2 To be submitted

M. Rashid *et al.*, "Optimal control of a commuter train with traction and braking time delays," *IEEE Intelligent Transportation Systems Transactions*, To be submitted

Chapter 2

Literature Review

2.1 Introduction

There is extensive research on automatic train control, both non-optimal and optimal train control. Many different control strategies and schemes have been proposed for automatic train operation with the aim of improving performance, reliability, and robustness.

This chapter presents a survey of the literature on control schemes for automatic train operation under the following sections:

2.2 Optimal Control

Optimal control is concerned with control of a system to achieve a desired behaviour or response in a manner that also optimizes, either maximizes or minimizes, a certain performance objective. In other words, optimal control is to select a control input of the system such that the system produces the desired behaviour or response, while also optimizing certain performance objectives.

This section provides a brief review of optimal control and the different solution approaches. The review is mainly concerned with continuous-time optimal control.

2.2.1 General Continuous Time Optimal Control Problem

The optimal control problem of a dynamical system in the continuous time domain is generally formulated as [15], [16]:

$$\min_{x(t),u(t),t} \quad J = \phi\left(x\left(t_0\right), t_0, x\left(t_f\right), t_f\right) + \int_{t_0}^{t_f} L\left(x\left(t\right), u\left(t\right), t\right) dt \tag{2.2.1}$$

$$\text{subject to:} \quad \dot{x}\left(t\right) = f\left(x\left(t\right), u\left(t\right), t\right) \tag{2.2.2}$$

$$\theta_{\text{Equality}}\left(x\left(t_0\right), t_0, x\left(t_f\right), t_f\right) = 0 \tag{2.2.3}$$

$$\theta_{\text{Inequality}}\left(x\left(t_0\right), t_0, x\left(t_f\right), t_f\right) \leq 0 \tag{2.2.4}$$

$$C_{\text{Equality}}\left(x\left(t\right), u\left(t\right), t\right) = 0 \tag{2.2.5}$$

$$C_{\text{Inequality}}\left(x\left(t\right), u\left(t\right), t\right) \leq 0 \tag{2.2.6}$$

Table 2.1: General Continuous Time Optimal Control Problem: Notation and Variables

Symbol	Quantity
t	Independent variable that is a scalar. The independent variable is usually time, especially in control and optimal control, since the dynamics of a system describe the evolution of a system with respect to time, and in reality time evolves independently. However, in general the independent variable can be any quantity.
$x(t)$	Vector of states of size n_x
$u(t)$	Vector of control inputs of size n_u

Since the initial state, $x(t_0) = x_0$, is usually known, this can also be written as:

$$\min_{x(t),u(t),t} \quad J = \phi\left(x\left(t_0\right), t_0, x\left(t_f\right), t_f\right) + \int_{t_0}^{t_f} L\left(x\left(t\right), u\left(t\right), t\right) dt \tag{2.2.7}$$

$$\text{subject to:} \quad \dot{x}\left(t\right) = f\left(x\left(t\right), u\left(t\right), t\right) \tag{2.2.8}$$

$$x\left(t_0\right) = x_0 \tag{2.2.9}$$

$$\phi_{\text{Equality}}\left(x\left(t_f\right), t_f\right) = 0 \tag{2.2.10}$$

$$\phi_{\text{Inequality}}\left(x\left(t_f\right), t_f\right) \leq 0 \tag{2.2.11}$$

$$C_{\text{Equality}}\left(x\left(t\right), u\left(t\right), t\right) = 0 \tag{2.2.12}$$

$$C_{\text{Inequality}}\left(x\left(t\right), u\left(t\right), t\right) \leq 0 \tag{2.2.13}$$

The notation used in Eqs. (2.2.7) to (2.2.13) is defined in Table 2.1.

This second optimal control formulation is now explained in more detail.

- Objective
  - The objective, Eq. (2.2.7), is a functional of the states, control, and time.
    - $L : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}$, in Eq. (2.2.7), general time-varying function of state and control at each point in time

- Constraints
  - State dynamics, Eq. (2.2.8), where $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_x}$
  - Initial state, Eq. (2.2.9), the initial state is usually known (given and fixed)
  - Final state equality constraints (boundary conditions), Eq. (2.2.10), where $\phi_{\text{Equality}} : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_{\text{Terminal-Boundary,Equality}}}$
  - Final state inequality constraints (boundary conditions), Eq. (2.2.11), where $\phi_{\text{Inequality}} : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_{\text{Terminal-Boundary,Inequality}}}$
  - State and control equality path constraints (path constraints), Eq. (2.2.12), where $C_{\text{Equality}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_{\text{Path,Equality}}}$
  - State and control inequality path constraints (path constraints), Eq. (2.2.13), where $C_{\text{Inequality}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_{\text{Path,Inequality}}}$

2.2.2 Solving an Optimal Control Problem

There are three main approaches to solving the optimal control problem detailed in Eqs. (2.2.7) to (2.2.13) as stated in [17]–[19]:

- Hamilton-Jacobi-Bellman Equation using Dynamic Programming
- Indirect Approach
- Direct Approach

2.2.2.1 Hamilton-Jacobi-Bellman Equation

The Hamilton-Jacobi-Bellman approach uses a dynamic programming approach to solve the optimal control problem. The Hamilton-Jacobi-Bellman approach involves partial differential equations, and in general is very difficult to solve analytically [19], [20].

2.2.2.2 Indirect Approach

The indirect approach involves the variational approach from the calculus of variations to determine the optimal control [21]. The indirect approach results in a multiple-point boundary-value problem that must be solved [17], [19], [21]. In general, it is very difficult to solve multiple-point boundary-value problems [19], [21]. Typically, the indirect method involves formation of a system Hamiltonian, and then application of the first-order necessary optimality conditions from the calculus of variations and Pontryagin's minimum principle to obtain a multiple-point boundary value problem that is to be solved to determine the optimal controls [17]. The indirect method approach is also known as "first optimize, then discretize" [17]–[19], since the necessary conditions of optimality are applied first, obtaining a multiple-boundary value problem, which is then discretized in order find a solution to the multiple-boundary value problem, and thus the original optimal control problem.

2.2.2.3 Direct Approach

The direct approach involves transforming the optimal control problem into an optimization problem [17], [19]. The continuous-time optimal control problem is an infinite-dimensional optimization problem [19], [22]. In this approach, it is converted to a finite-dimensional optimization problem, i.e. a finite number of decision variables [17], [19]. Note that the discrete-time optimal control problem, is a finite-dimensional optimization problem since there a finite number of decision variables [17]. The optimization problem can then be solved using any optimization solver.

Within the direct approach, there exist different methods for solving the optimal control problem. These methods, all have in common the finite parametrization of the control, but instead these methods differ on the extent the state is parametrized [17], [19]. There are two major groups of methods in the direct approach [17], [19]: (i) simultaneous, and (ii) sequential.

- In the sequential group, there exists the direct single-shooting method, which is described in [17], [19]. In this method, the control is discretized, and the state trajectory is considered to be dependent only on the initial state and the discretized control. The optimization problem has only the discretized controls as variables. The state trajectory is determined by simulation, i.e. solving an initial-value problem. Thus, the direct single-shooting method is a sequence of iterations of optimization followed by simulation.

- In the simultaneous group, two methods exist for solving optimal control problems: (i) direct multiple-shooting, and (ii) direct collocation [17], [19].

  - In the direct multiple-shooting method, the time interval is divided into subintervals. For each subinterval, the direct single-shooting method is applied. Continuity conditions are enforced for the state trajectory at the end of the i'th subinterval and at the beginning of the i+1'th subinterval, i.e. the state at the beginning of the i+1'th subinterval is constrained to be equal to the state at the end of the i'th subinterval. The values of the state at the beginning of each subinterval are considered to be variables in the optimization problem. For each subinterval, the state trajectory is considered to be dependent only on the state value at the beginning of the subinterval and the discretized control within the subinterval. The state trajectory for each subinterval is determined by simulation. The control is discretized along the entire time interval. All the discretized controls are also included in the optimization problem just like in the direct single-shooting case.

– In the direct collocation method, the states and controls along the entire time interval are discretized. Thus, the optimal control problem is transcribed into an nonlinear programming (NLP). Any continuous-time equations (constraints or objective) are converted to discrete-time. For example, (continuous-time) differential equations are converted to (discrete-time) difference equations, and integrals are converted to sums. The discretized states and controls now become decision variables in the optimization problem. A key point to mention here, is that the differential equations are only satisfied at the solution to the NLP. Also, the NLP ends up being a large-scale NLP, since now the states and control at each node are decision variables. The level of sparsity of the NLP is dependent on the transcription scheme used to convert (continuous-time) differential/integral equation constraints to (discrete-time) difference/sum equations.

The advantages of the direct approach are [17]–[19]: (1) it is easy to formulate the optimal control problem, (2) mature solvers exist for solving the resulting optimization problems, (3) constraints on state, control, or mixed state and control can easily be treated.

2.2.2.3.1 Dynamic Programming Note, it is also possible to use dynamic programming to solve the optimization problem as described in [20]. This involves discretizing the continuous-time optimal control problem to obtain a discrete-time optimal control problem. The state and control have to be quantized: instead of taking values from a continuous set, the state and control are restricted to taking values from a discrete set. The discrete-time optimal control problem can then be solved using dynamic programming. The discrete-time control input can be converted to continuous-time using zero-order hold. This approach can easily consider time-varying nonlinear systems, state constraints, control constraints, and mixed state-control constraints (i.e. constraint functions that depend on both the states and the control). The drawback of this approach is that as the discretization and quantization becomes finer, the computation and memory requirements grows rapidly and can even become intractable, this is known as the *curse of dimensionality* [20].

The direct method is also known as "first discretize, then optimize" [18], [19], since the problem is first discretized to obtain an optimization problem which is then solved to obtain the solution to the optimal control problem.

2.3 Model Predictive Control (MPC)

Model predictive control is a model-based control strategy, a mathematical model of the system being controlled is required. The model predictive control strategy is generally characterized by the following [23]:

- Use of a model of the system to predict the output of the system at future times (horizon or window).

- Calculation of the control over the future time window by formulation and solution of an optimization problem.

- A shifting optimization window, i.e. the following sequence is repeated: (i) solving an optimization problem to determine the control, (ii) application of the first few control samples, and (iii) shifting of the optimization window.

Model predictive control formulates the control problem as an optimization problem, which is then solved by an optimization solver. Model predictive control (MPC) achieves feedback by measuring the output of the system at regular intervals, and using the measured output or state in the next control iteration. The feedback law is obtained through iterative online optimizations [24]. The feedback law is actually a complex function of the current state [25].

Model predictive control can also be viewed as applying the direct method of optimal control sequentially. Both MPC and the direct method formulate the control problem as an optimization problem which is solved using optimization solvers. The difference is that the MPC strategy uses a shifting optimization window. A shifting window is characterized by continuous repetition of the following sequence: measuring the system output or state, solving the optimization (control) problem, and applying the first few control commands. The direct method is not characterized by repeatedly formulating and solving optimization problems to determine the control, whereas MPC is characterized by repeatedly formulating and solving optimization problems based on the latest system output measurements in order to determine the control.

In the direct method, the optimization problem may be formulated and solved once, and the resulting control commands applied open-loop.

Model predictive control requires the model of the system to be known relatively well, i.e. the model should describe the system dynamics relatively accurately [23], although the feedback provided by measuring the current system output or state and solving a new optimization problem to determine the control can compensate for disturbances, modelling errors, and other sources of uncertainty [26].

Some of the advantages of MPC are the following:

- easy to formulate [23],
- can control a wide range of processes and systems [23],
- can easily handle Multiple-Input Multiple-Output (MIMO) systems [23],
- can easily incorporate constraints, such as constraints on state, control, or functions of control and state [23],
- can easily incorporate nonlinear, time-varying system dynamics [23], [27], and
- can change the objective function and the system dynamics model online [28].

The disadvantages of MPC are the following [23]:

- computational and memory requirements to compute the control, and
- requirement for an appropriate model of the system or process being controlled.

2.3.1 Classical Model Predictive Control

The notation used in Eqs. (2.3.1) to (2.3.7) is defined in Table 2.2.

Table 2.2: Model Predictive Control: Notation and Variables

Symbol	Quantity
t	Time, scalar.
$x[t]$	State, vector of size n_x
$u[t]$	Control, vector of size n_u
$x[t+k \mid t]$	Refers to the predicted states at the time instant $t+k$ calculated at the current time instant t [23], of size n_x
$r_x[t]$	Reference trajectory for the states, of size n_x
$r_u[t]$	Reference trajectory for the states, of size n_u
W_x	Matrix of weights for states, of size $n_x \times n_x$
W_u	Matrix of weights for control, of size $n_u \times n_u$
$W_{\Delta u}$	Matrix of weights for control, of size $n_u \times n_u$
$l(x[t], u[t], t) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}$	Time-varying cost function of the states and the control
$f(x[t], u[t], t) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_x}$	State evolution
$C_{\text{Inequality}}(x[t], u[t], t) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_{\text{Path,Inequality}}}$	Path inequality constraints
$C_{\text{Equality}}(x[t], u[t], t) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_{\text{Path,Equality}}}$	Path equality constraints

Symbol	Quantity
$\phi_{\text{Inequality}}\left(x\left[t\right],t\right) : \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}^{n_{\text{Terminal-Boundary,Inequality}}}$	Terminal boundary inequality constraints
$\phi_{\text{Equality}}\left(x\left[t\right],t\right) : \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}^{n_{\text{Terminal-Boundary,Equality}}}$	Terminal boundary equality constraints

$$\min_{x[t+k],u[t+k],t+k,k\in 0,\ldots,N} \quad J = \sum_{k=0}^{N-1} l\left(x\left[t+k\,|t\right],u\left[t+k\right],t+k\right) \tag{2.3.1}$$

subject to:

$$x\left[k+1\right] = f\left(x\left[t+k\right],u\left[t+k\right],t+k\right) \tag{2.3.2}$$

$$\phi_{\text{Equality}}\left(x\left[t+N\right],t+N\right) = 0 \tag{2.3.3}$$

$$\phi_{\text{Inequality}}\left(x\left[t+N\right],t+N\right) \leq 0 \tag{2.3.4}$$

$$C_{\text{Equality}}\left(x\left[t+k\right],u\left[t+k\right],t+k\right) = 0 \tag{2.3.5}$$

$$C_{\text{Inequality}}\left(x\left[t+k\right],u\left[t+k\right],t+k\right) \leq 0 \tag{2.3.6}$$

- Commonly used cost functions include examples such as minimizing a quadratic function of the state (or error from a nominal state trajectory), control (or error from a nominal control trajectory), and the change in control as shown in Eq. (2.3.7). The change in control is added to maintain a smooth control that does not fluctuate rapidly, i.e. penalize discontinuous controls.

$$\begin{aligned} J = & \sum_{k=0}^{N} (x[t+k \mid t] - r_x[t+k])^T W_x (x[t+k \mid t] - r_x[t+k]) \\ & + \sum_{k=0}^{N-1} (u[t+k] - r_u[t+k])^T W_u (u[t+k] - r_u[t+k]) \\ & + \sum_{k=1}^{N-1} (u[t+k] - u[t+k-1])^T W_{\Delta u} (u[t+k] - u[t+k-1]) \end{aligned} \tag{2.3.7}$$

Classical model predictive control works as follows:

1. Measure (estimate) the current state of the system.
2. Solve the optimization problem for N steps into the future using the model of the system to predict the future states.
3. Apply the first k control commands, where $k \leq N$.
4. Repeat the procedure.

In the classical model predictive control setup, the key features are the iterative online optimizations and the moving horizon. The moving horizon nature of classical model predictive control is the reason why this control approach is also known as *receding horizon control* [24].

2.3.2 Economic Model Predictive Control (EMPC)

Economic model predictive control is a variant of model predictive control where the cost at each instant is not restricted to penalizing the distance from a set equilibrium [24]. In this variant, there is no a priori selected equilibrium that the controller attempts to stabilize or track. Instead of an a priori selection of an equilibrium trajectory to stabilize or track, the equilibrium trajectory is determined by the optimization process as a solution to the optimal control problem. The optimal control problem is defined by the dynamics, constraints, and objective function. Also, in the context of economic model predictive control (EMPC), depending on the problem, the concept of a moving horizon may no longer be suitable and instead other concepts such as a fixed horizon can instead be used [24]. In this regards, EMPC is most similar to optimal control. Specifically, the EMPC formulation is analogous to the formulation arrived at by iterative application of a simultaneous solution method from the direct solution approach framework to the optimal control problem. The key difference with EMPC and optimal control is that EMPC is a closed-loop method while optimal control may or may not result in a closed-loop control. EMPC can be considered as a closed-loop method because in EMPC, an optimization problem is periodically formulated and solved using the latest sensor feedback in order to determine the control.

2.4 Train Control

2.4.1 Introduction to Train Control

2.4.1.1 Automatic Train Control (ATC)

According to the IEEE standards manual, automatic train control (ATC) is defined as:

> The system for automatically controlling train movement, enforcing train safety, and directing train operations. ATC must include ATP and may include ATO and/or ATS. [29, p.2]

. The ATC system consists of three subsystems, ATO, ATP, and ATS, and is responsible for train safety, movement, and operation [30].

2.4.1.2 Automatic Train Supervision (ATS)

According to the IEEE standards manual, ATS is defined as: "The subsystem within the ATC system that monitors trains, adjusts the performance of individual trains to maintain schedules, and provides data to adjust service to minimize inconveniences otherwise caused by irregularities." [29, p.3].

ATS is the system responsible for managing the fleet of trains on the network and a description is provided as follows, according to [30]. ATS uses an established schedule to dispatch trains, control station dwell times, and interstation run times. ATS is also responsible for controlling access to track sections, in order to prevent collisions and ensure safe headway. The ATS system is usually located in a central location and a two-way communication system is used to communicate with the trains.

2.4.1.3 Automatic Train Operation (ATO)

According to the IEEE standards manual, ATO is defined as: "The subsystem within the ATC system that performs any or all of the functions of speed regulation, programmed stopping, door control, performance level regulation, or other functions otherwise assigned to the train operator." [29, p.3].

ATO is the system responsible for managing the operation of a single train [30]. It controls the train speed by issuing traction and braking commands. The ATO system is also responsible for accurately stopping the train at stations. This is important because the station platform may have doors that have to line up with the train doors for passengers to be able to enter and exit the train from the platform. The ATO system interacts with the ATP system.

2.4.1.4 Automatic Train Protection (ATP)

According to the IEEE standards manual, ATP is defined as: "The subsystem within the ATC system that maintains fail-safe protection against collisions, excessive speed, and other hazardous conditions through a combination of train detection, train separation, and interlocking." [29, p.3].

ATP is a safety-critical system that is responsible for the safe operation of a train [30]. The ATP system controls the emergency braking system, ensures safe train separation, monitors speed and prevents overspeeding, controls train door operation, and other safety-critical subsystems. The ATP system is designed in a fail-safe manner.

2.4.2 Non-Optimal Train Control

2.4.2.1 Proportional-Integral-Derivative Control (PID)

Given a velocity trajectory from station A to station B, the goal of Proportional-Integral-Derivative control (PID) in train control is to track a pre-specified trajectory or target speed [31], [32]. For a minimum-time objective, the target speed can be the maximum permitted speed. In practice, the target speed is close to but not exactly the maximum permitted speed, since a buffer region is necessary in order to reduce the probability that the train speed will surpass the maximum permitted speed. If the PID controller was to attempt to track the maximum permitted speed, there is a chance that the maximum permitted speed may be surpassed, thus violating the speed limit.

A reference trajectory can also be provided by another high-level controller, such as an existing optimal train control method based on a simplified model of the train. Examples of simplified train dynamics models can be found in the works on optimal train control that are based on analytical solutions to the optimal control problem using classical optimal control theory (formulation of a system Hamiltonian and then application of the first-order necessary conditions of optimality from the calculus of variations and Pontryagin's minimum principle). These simplified models ignore input-delays and limits on the change in control, and use less accurate models for the maximum traction force limit. The use of an optimal controller to generate the reference trajectory can improve operational efficiency. The PID controller attempts to follow the reference trajectory provided by the high-level controller. The challenge of dealing with the complex dynamics of the system are then transferred to the PID controller. An issue with this approach is that although attempting to follow a reference trajectory can help to improve on performance criteria, it may not be able to improve the performance objectives as much as having an optimal controller synthesized using the complex dynamics. Another issue is that if the trip specifications change partway during the trip, a new reference trajectory may have to be generated. An optimal controller synthesized using the complex dynamics has its own issues such as computational resource limitations, computation-time and memory limitations, and difficulty in finding a global optimum.

2.4.2.2 Model Predictive Control (MPC)

MPC control is similar to PID in that MPC is used to mainly track a reference trajectory. The MPC framework has been used for train control with the objective of following a reference trajectory while maintaining energy-efficiency, low-jerk, and passenger comfort [33].

In the MPC framework, as described in [33], an optimization problem is formulated and then solved at each time instant. A select number of control commands are applied. The outputs are measured, and used to formulate a new optimization problem. This process is repeated to obtain the control law. MPC train control is different from optimal control in the following ways:

- MPC requires a reference trajectory, optimal control does not require a reference trajectory [33]; and
- MPC is used as a moving (receding) horizon control scheme in train control, while optimal control has a fixed horizon [33].

Some of the advantages of MPC over PID control are the following:

- MPC follows the reference trajectory optimally, while PID does not (MPC can simultaneously optimize for tracking error, fuel consumption, passenger comfort, etc.) [33];
- MPC can explicitly consider constraints on the control and states, while PID can not [33]–[36]; and
- MPC can explicitly consider the nonlinear and time-varying behaviour, while PID can not.

Some of the advantages of PID control over MPC are the following:

- PID is not computationally intensive, while MPC is; and
- PID does not require a model of the system, while MPC does.

Since MPC also needs a reference trajectory like PID, MPC also has some of the same issues as PID. The discussion of the issues with the use of a reference trajectory, in Subsubsection 2.4.2.1, applies here as well.

2.4.2.3 Fuzzy Control

Fuzzy control was developed in order to overcome some of the deficiencies present in PID control [32], [37]–[40]. A human operator has superior performance to a PID controller. PID control was not effective at considering the different trade-offs present when choosing a control value, and was not adept at dealing with the nonlinear and time-varying behaviour present in the train dynamics. Fuzzy control is an attempt to replicate a human operating the train. A human operator would consider a number of different factors, such as desired arrival time, energy consumption, passenger comfort, proximity to speed limit, grade, frequency of past control changes, smoothness of the control, etc., and then choose an appropriate control command. Fuzzy control attempts to consider these trade-offs, in a manner similar to a human operator.

The disadvantages of using fuzzy control systems for train operation are that the fuzzy controller must be tuned, which can be a very time-consuming process [41]. Also, fuzzy control may not be adept at adapting to changes in the trip specifications partway through the trip if the rules that determine the trade-offs between various objectives, e.g. trip time, energy consumption, smoothness of control, are static, i.e.

fixed and cannot be adjusted partway through the trip. Fuzzy control is also not optimal.

2.4.2.4 Expert Systems

In [42], a two-level controller is used for automatic train control. At the low-level, the control framework in the paper actually uses a different model to characterise a specific operating region for the system. Each model is controlled with its own controller. The controller architectures used can differ greatly ranging from conventional controllers to fuzzy optimal controllers. At the high-level, there is a real-time expert system that supervises all the low-level models and their respective low-level controllers. The high-level expert system determines which model and controller are active at the current time.

The disadvantage of such an approach is the potentially large computation time compared to PID. The expert system must also be tuned and trained, which can be very time-consuming. Proper configuration of the expert system may take a considerable number of work hours on the part of a human operator or operators. Another disadvantage of such an approach is that the reference trajectory may be suboptimal when compared to optimal controller determined trajectories.

2.4.2.5 Neural Networks

In the papers, [43], [44], a train control scheme is proposed that uses fuzzy neural networks. The controller consists of two fuzzy neural networks organized in an hierarchical fashion. One fuzzy neural network determines the reference trajectory and an appropriate performance objective. Specifically, there are n reference function generators and the fuzzy neural network determines: (1) the parameter sets to be used for each reference function generator; (2) the weights to be used when summing the results of the n reference function generators. The second fuzzy neural network determines: (1) the gains to be used for a series of n PI controllers; (2) the weights to be used when summing the outputs of the n PI controllers. The lower-level controller determines the control in order to track its respective reference trajectory.

The disadvantages of such an approach are:

- The neural networks have to be trained, which can be very time-consuming [45].
- The reference trajectory may be suboptimal when compared to optimal controller determined trajectories.

2.4.3 Optimal Train Control

There is extensive literature on the optimal control of a train as can be seen in the following papers, [3], [46]–[48], and in the book by Howlett and Pudney, [49]. Specifically, the review papers, [3], [46], provide a comprehensive review of existing literature on the train optimal control problem. The papers, [3], [46]–[48], and the book, [49], provide a good overview of optimal train control.

The train optimal control problem has generally been dealt with in the position domain, and as stated in [3], it is widely accepted that it is easier to treat the train optimal control problem in the position-domain versus the time-domain. The transformation for converting an optimal control problem along a fixed path from time-domain to a coordinate system along the fixed path itself has been presented at least as early as 1985 in [50].

2.4.3.1 Calculus of Variations (Indirect Approach)

Majority of existing optimal train control literature is concerned with analytical solutions to the train optimal control problem based on classical optimal control theory. This approach involves formulating a system Hamiltonian and applying the first-order necessary conditions of optimality from the calculus of variations and Pontryagin's minimum principle. Numerical algorithms are sometimes given to solve the resulting equations. This approach provides interesting theoretical insight; however, it is limited in its ability to incorporate complex system dynamics models, state constraints, input constraints, mixed state-input constraints, and complex objectives. This is due to the difficulty in obtaining analytical solutions when incorporating more complicated objectives and constraints.

The earliest published work found by the author on the train optimal control problem is by Ichikawa in 1968, [51]. Some other early works include Kokotovic and Singh in 1972, [52], and Milroy in 1980, [53]. These works utilize classical optimal control theory, which is based on the variational approach from the calculus of variations. These works all form a system Hamiltonian, and then use the first-order necessary conditions of optimality from the calculus of variations and Pontryagin's minimum principle to determine the optimal controls. The solution to the problem of a minimum-time or minimum-energy train journey results in optimal control phases, where at each phase, one the following control strategies is applied as described in [3]:

- Maximum Power
- Speed Hold by applying traction
- Coast

- Speed Hold by applying braking
- Brake

The optimal control problem then becomes a finite-dimensional optimization problem to determine the optimal switching points between the different regions of control [3].

2.4.3.2 Direct Approach

Dynamic programing has been employed to solve the train optimal control problem [54]. However, the method suffers from the "curse of dimensionality" as detailed in [20]. The computation and memory requirements for solving optimal control problems using the dynamic programming approach can be very significant, and infeasible in some cases. The advantage of the method is that complicated constraints on control, state, functions of state and control can be considered without significantly affecting the tractability of the problem as described in [54].

In another direct approach, the train optimal control problem is transformed from continuous-time (or continuous-position) to discrete-time (or discrete-position), thus converting the infinite-dimensional optimization problem to a finite-dimensional one [54], [55]. In [54], the problem is discretized from continuous-time to discrete-time, and the problem is converted into the form of an optimization problem. The sequential quadratic programming (SQP) algorithm is then applied to solve the problem. However, the authors of [54], mention in their paper that their work is not suited to real-time control. The authors also mention some strategies to achieve real-time control, namely, receding horizon control and model predictive control. The authors also suggest the use of an extensive offline control calculation to determine the optimal operating modes for a range of conditions, which is stored in a lookup table. The lookup table is used to operate the train.

In [55], several methods under the direct approach family are used to solve the train optimal control problem. The authors use three different methods to solve the problem: (1) Gauss Pseudospectral Method to discretize the dynamics differential equation and the integral in the objective and then solve using the Pseudospectral Optimal Control Solver (PSOPT) software package; (2) convert the optimization problem into a mixed-integer linear programming (MILP) problem and solve using specially designed solvers; (3) quantize the states and the control and then use the dynamic programming algorithm by using the Discrete Dynamic Programming (DDP) software package.

A note on MILPs must be made. MILP problems are not convex because of the presence of discrete variables, i.e. variables that can only take on values in a finite set. However, these problems can be solved to global optimality in a reasonable amount of time using specially designed solvers [55], [56]. Further clarification must be made, as explained in [56], although solutions to many practical MILP problems can be obtained in a reasonable amount of time, there is no guarantee that every MILP can be solved in a reasonable amount of time. Although the solvers guarantee finding the global optimum for MILPs, it may take an unreasonably long time to solve for global optimality.

The issues with the approach used in [55] is that the optimal control approach only determines a reference trajectory. There exists another low-level controller that is intended to track the provided reference trajectory. The issue with this is that the overall system may be suboptimal because the system may not track the reference trajectory accurately all the time. Performance can be improved by re-solving a new optimal trajectory which can take into account the difference between the current system state and the previously calculated optimal trajectory, i.e. find a new optimal trajectory from the current system state which is not on the previous optimal trajectory. This will determine an optimal way of going from the current system state to the desired final system state, since the previous optimal trajectory may not be the optimal trajectory to follow to achieve the desired final system state from the current system state. Also, the authors state that the optimal controller in the paper is not suitable for real-time control.

In [57], [58], the train optimal control problem is solved using the genetic algorithm (GA). Particularly in [57], the method is used to calculate a schedule for a single train operation that can be followed by the ATO system. The schedule indicates when the train should coast in order to save energy. The schedule is calculated before the train departs. The authors use a suboptimal solution as the starting point for the GA and assume that the optimal solution is close to the suboptimal starting point. The suboptimal point is itself determined from a lookup table which provides the distance from the destination at which the train can start coasting, as a function of the supply voltage, timetabled passenger loads, and on-time operation. In [58], simulation results are provided for the GA and compared against Howlett's results, which are based on the indirect calculus of variations approach. The results in the paper show that the algorithm works reasonably well compared to the indirect calculus of variations approach. One of the issues with this approach is that the GA can only search near a suboptimal initial starting point. The optimization algorithm is, in a sense, restricted to look for a solution near the suboptimal initial starting point and is restricted in its ability to search a large space of the variables for a possibly better point.

2.5 Time-Delay Systems

Time-delays in actuation or sensing may degrade performance and even make systems unstable [4]. Systems with delays pose considerable challenge and difficulty, and optimal control of such systems can pose additional problems. For many systems, time-delays in sensing can be moved over to the actuation side for the purpose of analysis and control.

2.5.1 Continuous Time

Unlike conventional systems, continuous time time-delay systems are characterized by an infinite-dimensional state, i.e. an infinite amount of memory [4], [7]–[11], [59], [60]. This greatly adds to the complexity of time-delay systems.

For linear systems, different control approaches have been proposed such as finite spectrum assignment, [8], [61], reduction, [9], modified Smith predictor, [62], [63], and feedback stabilization, [7]. Finite spectrum assignment is detailed in [8], [61]. Linear input time-delay systems can be considered to possess an infinite spectrum, i.e. an infinite number of eigenvalues. Finite spectrum assignment is concerned with obtaining a feedback law which results in a closed-loop system consisting of only a finite number of eigenvalues which can be arbitrarily placed, and the rest of the eigenvalues are eliminated. The feedback law depends on the past control input history.

Reduction is concerned with the transformation of an infinite-dimensional system to a finite-dimensional one, the system with delay is transformed to one without delay [9]. This approach has been termed as a "predictor-like" technique because the control law is based on the past control input trajectories [64].

Feedback stabilization and the modified Smith predictor are both predictive approaches to dealing with delay [4], [7], [65]. The Smith predictor involves including feedback based on the predicted evolution of the system. The predicted evolution is obtained by the use of a mathematical model of the system. The use of a predictor converts the controller design problem from control of a input time-delay system to one of a input delay-free system. All of the above approaches for linear systems transform the problem of control of an input time-delay system to one without any input time-delay.

For nonlinear systems, the predictor approach has been presented for systems with the same delay for all of its inputs [4]–[6]. The predictor framework bases the feedback not on the current state or output, but rather a future predicted state or output. The state and output at a future time is completely determined by the current state, the past control history, and a mathematical model of the system.

2.5.2 Discrete-Time

Discrete-time systems with constant bounded time-delays in the input can be converted to a higher-order discrete-time delay-free system by augmentation of the states, see [66]–[70].

2.5.3 Time-Delays and Optimal Train Control

Existing literature on optimal control of a train does not consider delays in the actuation model.

Table 2.3: Notation and Variables

Notation	Description
x	vector of size $\mathbb{R}^n$
a_i	vector of size $\mathbb{R}^n$
b_i	scalar
c	vector of size $\mathbb{R}^n$
$f(x)$	$\mathbb{R}^n \to \mathbb{R}$
$g_i(x)$	$\mathbb{R}^n \to \mathbb{R}$
$h_i(x)$	$\mathbb{R}^n \to \mathbb{R}$

2.6 Optimization

The intent of this section is to provide a very brief review of a few concepts from optimization. The notation used in this section is defined in Table 2.3.

2.6.1 General Optimization Problem

A general nonlinear optimization problem can be formulated as Eq. (2.6.1) [71]. Equation (2.6.1.1) is the objective, Eq. (2.6.1.2) are the m inequality constraints, and Eq. (2.6.1.3) are the p equality constraints. Note that any inequality constraints of the form $g_{i,geq}(x) \geq 0$ can be converted to $g_i(x) \leq 0$ by multiplying both sides by -1, and then replacing $-g_{i,geq}(x)$ by $g_i(x)$.

$$\min_{x \in \mathbb{R}^n} \quad f(x) \tag{2.6.1.1}$$

$$\text{s.t.:} \quad g_i(x) \leq 0 \quad \text{for } i = 1, \ldots, m \tag{2.6.1.2}$$

$$h_i(x) = 0 \quad \text{for } i = 1, \ldots, p \tag{2.6.1.3}$$

2.6.2 Convex Optimization Problem

A convex optimization problem is one where the objective function is convex and the feasible set is a convex set [71], [72]. More specifically, for a minimization problem, the problem is convex, if the objective function is convex and the feasible set is convex. For a maximization problem, the problem is convex, if the objective function is concave and the feasible set is convex.

A convex optimization problem written in standard form is formulated as Eq. (2.6.2).

$$\min_{x \in \mathbb{R}^n} \quad f(x) \tag{2.6.2.1}$$

$$\text{s.t.:} \quad g_i(x) \leq 0 \quad \text{for } i = 1, \ldots, m \tag{2.6.2.2}$$

$$a_i^{\mathrm{T}} x - b_i = 0 \quad \text{for } i = 1, \ldots, p \tag{2.6.2.3}$$

where $f(x)$, and $g_i(x)$ are all convex functions. Note that affine functions, $a_i^{\mathrm{T}} x - b_i = 0$, are convex (and concave).

Some important notes about convex optimization problems:

- In general convex optimization problems can be solved for global optimality in a reasonable amount of time, i.e. convex optimization problems are computationally tractable [73], [74].
- Any local optimizer of a convex optimization problem is also a global optimizer [71], [75].

2.6.3 Linear Programming

A linear program is one where the objective and all the constraints are linear or affine functions of the variables. A linear program can be formulated as:

$$\min_{x\in\mathbb{R}^n} \quad c^{\mathrm{T}}x \tag{2.6.3.1}$$

$$\text{s.t.:} \quad a_i^{\mathrm{T}}x - b_i \leq 0 \quad \text{for } i = 1, \ldots, m \tag{2.6.3.2}$$

$$g_i^{\mathrm{T}}x - h_i = 0 \quad \text{for } i = 1, \ldots, p \tag{2.6.3.3}$$

2.6.4 Local vs. Global Optimum

A local optimum is a point that minimizes the objective function among neighbouring feasible points [71]. A mathematical description of a locally optimal point, x, is as follows [71]:

$$f(x) = \inf \{f(y) \,|g_i(y) \leq 0, i = 1, \ldots, m, h_i(y) = 0, i = 1, \ldots, p, \|y - x\|_2 \leq R\} \tag{2.6.4}$$

where $R \geq 0$.

A global optimum is a point that minimizes the objective function over all feasible points [71]. A mathematical description of a globally optimal point is as follows [71]:

$$f(x) = \inf \{f(y) \,|g_i(y) \leq 0, i = 1, \ldots, m, h_i(y) = 0, i = 1, \ldots, p\} \tag{2.6.5}$$

2.6.5 Dual Problem

The optimization problem defined in Eq. (2.6.1) is also known as the primal problem [76]. The dual problem is defined as [76]:

$$\max_{\lambda,\nu} \quad \inf_{x\in\mathbb{D}} \left(f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right) \tag{2.6.6.1}$$

$$\text{s.t.:} \quad \lambda_i(x) \geq 0 \quad \text{for } i = 1, \ldots, m \tag{2.6.6.2}$$

where $\mathbb{D} = \{x | g_i(x) \leq 0, i = 1, \ldots, m, h_i(x) = 0, i = 1, \ldots, p\}$, i.e. D is the feasible set for the primal problem. The dual problem is always a convex problem regardless of whether the primal problem is convex or not [76].

2.6.6 KKT Conditions

Assuming that the functions $f(x), g_1(x), \ldots, g_m(x), h_1(x), \ldots, h_p(x)$ are differentiable, the Karush-Kuhn-Tucker conditions for optimality (KKT) conditions for the general nonconvex optimization problem are [76]:

- Primal Feasibility

$$g_i(x^*) \leq 0 \quad \text{for } i = 1, \ldots, m \tag{2.6.7.1}$$

$$h_i(x^*) = 0 \quad \text{for } i = 1, \ldots, p \tag{2.6.7.2}$$

- Dual Feasibility

$$\lambda_i^* \geq 0 \quad \text{for } i = 1, \ldots, m \tag{2.6.7.3}$$

- Complementary Slackness

$$\lambda_i^* g_i(x^*) = 0 \quad \text{for } i = 1, \ldots, m \tag{2.6.7.4}$$

- Stationarity Condition

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{i=1}^{p} \nu_i^* \nabla h_i(x^*) = 0 \tag{2.6.7.5}$$

2.6.7 Algorithms for Nonlinear Constrained Optimization

This section provides a very brief review of a few select algorithms for nonlinear constrained optimization.

2.6.7.1 Sequential Quadratic Programming (SQP)

SQP methods are described in [75], and the description that follows is taken mainly from [75]. These methods work by solving a sequence of quadratic subproblems. There are different ways of viewing the SQP approach. One of the ways is that the nonlinear optimization problem is approximated using a quadratic program around a given point or iterate:

- The nonlinear objective is approximated using a quadratic function.
- The nonlinear inequality and equality constraints are approximated using linear functions, i.e. linearizing the nonlinear inequality and equality functions.

The quadratic program is solved, resulting in a new iterate for the optimal point. The process is repeated at the new iterate. Thus, a sequence of quadratic programs is solved, hence the name *sequential quadratic programming.*

2.6.7.2 Interior-Point Methods

Interior-point methods work by successively solving approximately the perturbed KKT conditions [75], [77], [78]. Then the perturbed KKT conditions are adjusted to more closely approximate the actual KKT conditions for the optimization problem and the process is repeated again, see [75], [77], [78] for a much more detailed discussion of interior-point methods.

There is a variant class of interior-point methods known as primal-dual interior-point methods. The primal-dual methods differ from primal interior-point methods in the following ways [75], [78]:

- Primal methods work only in the space of primal variables. Primal-dual methods work in the space of both primal and dual variables. Primal-dual methods update both primal and dual variables at each iteration.
- The search directions of the primal and primal-dual methods may be different, i.e. produce a different set of iterates.
- Primal methods require a feasible starting point while the primal-dual methods do not.

Chapter 3

Dynamics Formulation

In [79], it is shown that the equations of motion using a distributed-mass train model can be reduced to the equations of motion of a point-mass train model. Thus, only the point-mass train model will be considered and the impact of in-train forces on the train dynamics behaviour will be ignored. The equations of motion in the time domain are presented next.

3.1 Time Domain

3.1.1 Zero Traction-Braking Delays

The train dynamics model is taken from [3], [80]–[82]. The forces included in the train dynamics model are:

- Resistance force

  In [3], [83], the modelling of the resistance force is briefly described. The resistance force is modelled by the Davis formula which takes the form of a second-order polynomial in $v(t)$ where all the coefficients are non-negative: $C(v(t)) = C_0 + C_v v(t) + C_{v^2} v^2(t)$. The resistance force consists of the following components [80]:

  - Journal resistance: a constant force and only occurs if the train is moving, corresponds to the term, C_0.
  - Flange resistance: proportional to velocity, corresponds to the term, $C_v v(t)$.
  - Air resistance: proportional to the square of velocity, corresponds to the term, $C_{v^2} v^2(t)$.

- Component of force of gravity due to track grade

  This is the force of gravity tangential to the track. It is modelled as:

$$g(s(t)) = m g_{\text{gravity,acc}} \sin(\theta(t)) \tag{3.1.1}$$

  where θ is the angle between the incline and the horizontal. Since the height of the track is known for any position along the track, thus the grade and the component of gravity affecting longitudinal motion are known functions of position.

- Traction or braking force of the train itself

  The force is given by $u(t)$, where u is the control, traction or braking, force.

For the train system, the equations of motion in time-domain are:

- Position evolution:

$$\frac{ds(t)}{dt} = v(t) \tag{3.1.2.1}$$

- Velocity evolution:

$$m\frac{dv\left(t\right)}{dt}=-C\left(v\left(t\right)\right)+g\left(s\left(t\right)\right)+u\left(t\right) \tag{3.1.2.2}$$

3.1.2 Equal Traction-Braking Delays

The only difference between this case and the case in Subsection 3.1.1 is that the dynamics equation, Eq. (3.1.2.2), is replaced with Eq. (3.1.3.1).

$$m\frac{dv\left(t\right)}{dt}=-C\left(v\left(t\right)\right)+g\left(s\left(t\right)\right)+u\left(t-T_{\text{delay}}\right) \tag{3.1.3.1}$$

3.1.3 Non-Equal Traction-Braking Delays

A more complicated system model includes one in which traction and braking are separate control inputs, each with different time-delays. The difference between this case and Subsection 3.1.1 are the following: the single control input, $u\left(t\right)$, is replaced with the two control inputs, traction, $u_{\text{trk}}\left(t\right)$, and braking, $u_{\text{brk}}\left(t\right)$; and the dynamics equation, Eq. (3.1.2.2), is replaced with Eq. (3.1.4.1).

$$\begin{aligned} m\frac{dv\left(t\right)}{dt}=&-C\left(v\left(t\right)\right)+g\left(s\left(t\right)\right)+u_{\text{trk}}\left(t-T_{\text{trk,delay}}\right)\\ &-u_{\text{brk}}\left(t-T_{\text{brk,delay}}\right)\end{aligned} \tag{3.1.4.1}$$

3.2 Position-Domain

The equations of motion using position as the independent variable are detailed in [3], [47], [48], [84] for a train, and for more general vehicles in [85]. The position domain dynamics are provided since the optimal control problem for the case of equal traction-braking delays will be formulated and solved in the position domain, as will be explained in Chapter 5.

3.2.1 Zero Traction-Braking Delays

The continuous-position dynamics equations are:

- Time evolution:

$$\frac{dt(s)}{ds} = \frac{1}{v(s)} \tag{3.2.1.1}$$

- Velocity evolution:

$$mv(s)\frac{dv(s)}{ds} = -C(v(s)) + g(s) + u(s) \tag{3.2.1.2}$$

3.2.2 Equal Traction-Braking Delays

The only difference between this case and the case in Subsection 3.2.1 is that the dynamics equation, Eq. (3.2.1.2), is replaced with Eq. (3.2.2.1)

$$mv(s)\frac{dv(s)}{ds} = -C(v(s)) + g(s) + u\left(s - S_{\text{delay}}\right) \tag{3.2.2.1}$$

where $s - S_{\text{delay}}$ is the position T_{delay} seconds ago, i.e. $t\left(s - S_{\text{delay}}\right) = t(s) - T_{\text{delay}}$.

3.2.3 Non-Equal Traction-Braking Delays

For this case, the single control input, $u(s)$, is replaced with the two control inputs, traction, $u_{\text{trk}}(s)$, and braking, $u_{\text{brk}}(s)$; and the dynamics equation, Eq. (3.2.1.2), is replaced with Eq. (3.2.3.1).

$$\begin{aligned} mv(s)\frac{dv(s)}{ds} = -C(v(s)) + g(s) + u_{\text{trk}}\left(s - S_{\text{trk,delay}}\right) \\ - u_{\text{brk}}\left(s - S_{\text{brk,delay}}\right) \end{aligned} \tag{3.2.3.1}$$

Note that for Eq. (3.2.3.1), $s - S_{\text{trk,delay}}$ is the position $T_{\text{trk,delay}}$ seconds ago, i.e. $t\left(s - S_{\text{trk,delay}}\right) = t(s) - T_{\text{trk,delay}}$; and $s - S_{\text{brk,delay}}$ is the position $T_{\text{brk,delay}}$ seconds ago, i.e. $t\left(s - S_{\text{brk,delay}}\right) = t(s) - T_{\text{brk,delay}}$.

There are also a number of disadvantages with framing the train optimal control problem using position as the independent variable as described in [3]. In the position domain, the differential equations have singularities when the velocity is zero. This can be circumvented by setting the velocity at the origin and the destination to be arbitrarily small but non-zero [47]. The rate of change of time evolution approaches infinity as the velocity approaches zero (arbitrarily small but positive velocity). The rate of change of velocity evolution approaches infinity if the acceleration experienced is positive and the velocity approaches zero (arbitrarily small but positive velocity). The singularity in the velocity evolution equation can also be removed by changing state from velocity to kinetic energy as described in [3], [47].

3.3 Maximum Tractive and Braking Forces

3.3.1 Maximum Tractive Force

The maximum traction force that can be applied is considered to be a function of the current train velocity. The traction force is limited by

$$u_{\text{trk,max}} = \min\left\{U_{\text{trk,max}}, \frac{P}{v}, \frac{k_P P}{v^2}\right\} \tag{3.3.1}$$

where P is the maximum power (constant), v is train velocity, $U_{\text{trk,max}}$ is an upper bound on the tractive force (constant), and k_P is a constant greater than or equal to one. Note that the term, $\frac{k_P P}{v^2}$, is an approximation used for the motor field weakening region [33], [86].

3.3.2 Maximum Braking Force

The braking force is assumed to be a constant independent of the train velocity.

Chapter 4

Optimal Control Formulation of a Commuter Train with and without Delays in the Input

The control approach to drive a train from Station A to Station B is as follows. The control problem is solved repeatedly along the track using the latest sensor measurements. The first few control commands are applied to the system. The state or output of the system is measured, and then a new optimal control problem is solved based upon the latest sensor measurements.

Formulations are provided for the train optimal control problem in the continuous temporal and spatial domains for each of the following cases: (i) no input delay in traction and braking commands, (ii) equal delays in traction and braking commands, and (iii) non-equal delays in traction and braking commands. The formulations are provided and the intractable nature of each of them are discussed.

4.1 Zero Traction-Braking Delays

4.1.1 Continuous-Time

- Objective function:

$$\min_{s(t),v(t),u(t),t} J \tag{4.1.1.1}$$

- Subject to the following constraints:
  - Initial states
    - Initial position

$$s\left(t_0\right) = s_0 \tag{4.1.1.2}$$

    - Initial velocity

$$v\left(t_0\right) = v_0 \tag{4.1.1.3}$$

  - Final states
    - Arrive at destination

$$s\left(t_{\text{f}}\right) = s_{\text{f}} \tag{4.1.1.4}$$

    - Stop at rest

$$v\left(t_{\text{f}}\right) = 0 \tag{4.1.1.5}$$

  - Final arrival time scheduling bounds

$$t_{\text{f,min}} \leq t_{\text{f}} \leq t_{\text{f,max}} \tag{4.1.1.6}$$

  - Position dynamics

$$\frac{ds\left(t\right)}{dt} = v\left(t\right) \tag{4.1.1.7}$$

– Velocity dynamics

$$m\frac{dv(t)}{dt} = -C(v(t)) + g(s(t)) + u(t) \tag{4.1.1.8}$$

– Must either not pass a certain position limit for each value of time, or, must not be behind a certain position limit for each value of time

$$s_{\text{min}}(t) \leq s(t) \leq s_{\text{max}}(t) \tag{4.1.1.9}$$

– Speed limits

$$0 \leq v(t) \leq v_{\text{max}}(s(t)) \tag{4.1.1.10}$$

– Control/actuation limits

$$u_{\text{min}}(v(t)) \leq u(t) \leq u_{\text{max}}(v(t)) \tag{4.1.1.11}$$

– Limits on the change in control

$$\mu_{\text{min}}(t) \leq \frac{d}{dt}(u(t)) \leq \mu_{\text{max}}(t) \tag{4.1.1.12}$$

A note on the above optimal control problem: the formulation above is not tractable in the continuous time domain, but is instead tractable in the discrete time domain.

4.1.1.1 Objective Function

The two widely used objective functions according to [3], [47], are:

- Minimum-time:

$$\min_{s(t),v(t),u(t),t} t_{\mathrm{f}} \tag{4.1.1.13}$$

- Minimum-energy, which has the two following variants:
  - Minimum-energy with no regenerative braking. This occurs when there is no system to recover any of the kinetic energy during braking, i.e. all the kinetic energy is lost when the train brakes. The objective function is,

$$\min_{s(t),v(t),u(t),t} \int_{t_0}^{t_{\mathrm{f}}} \max\left(u\left(t\right),0\right) v\left(t\right) dt \tag{4.1.1.14}$$

  - Minimum-energy with regenerative braking. This occurs when the train has a system that can recover a portion of the kinetic energy while braking. The objective function is,

$$\min_{s(t),v(t),u(t),t} \int_{t_0}^{t_{\mathrm{f}}} \left\{\max\left(u\left(t\right),0\right) + \sigma \min\left(u\left(t\right),0\right)\right\} v\left(t\right) dt \tag{4.1.1.15}$$

    where σ is the regenerative braking recovery factor, i.e. the portion of braking energy that is recovered.

Only minimum-time and minimum-energy with no regenerative braking formulations are considered in this thesis. It is trivial to extend the minimum-energy with no regenerative braking formulation to the case of minimum-energy with regenerative braking.

4.1.1.2 Disadvantages of Formulating the Train Optimal Control Problem in Time Domain

There are a number of disadvantages with framing the optimal control problem of a train using time as the independent variable [3], [47], [48], [84]. This is explained in more detail below.

Speed limits are known in advance as a function of position. Different sections of the track may have different speed limits [3], [47], [48], [84].

Grade is known in advance as a function of position. Since the component of the force of gravity due to track grade may have a considerably larger magnitude than the resistance force and it may also be the largest force experienced by the train, other than the tractive and braking force of the engine and braking mechanism, respectively, it is important to accurately consider the grade while discretizing the problem in order to solve it using an optimization solver [3], [47], [48], [84].

An important real-life objective is to minimize the travel time. Minimizing travel time allows for the operation of more train trips, decreasing headway, increasing passenger capacity, and reducing passenger wait times across the rail line [3], [48], [84]. Including travel time into the objective function results in increasing the complexity of the nonlinear optimization problem as opposed to a fixed trip travel time. A more detailed elaboration on the minimum-time objective increasing problem formulation complexity is as follows:

- If the discretization intervals, Δt, are fixed, then the objective would be to minimize the number of intervals. In this case, the optimization problem becomes a mixed-integer nonlinear linear programming (MINLP). This is because the number of intervals is a discrete variable which can only take positive integer values. MINLP problems are extremely difficult to solve [87], [88].

- If the number of discretization intervals is fixed, then the objective becomes to minimize the sum of the discretization intervals Δt_i, i.e. the Δt_i are decision variables. This problem is definitely easier than the first one, since it is not an MINLP but instead an NLP. However, it increases the number of decision variables and makes the constraints more complicated as opposed to the problem in the position-domain. Specifically, it adds N more decision variables, where N is the number of discretization nodes.

4.1.2 Continuous-Position

To address the issues with using time as the independent variable, the problem is formulated in the position domain.

- Objective function:

$$\min_{t(s),v(s),u(s),s} J \tag{4.1.2.1}$$

- Subject to the following constraints:
  - Initial states
    - Initial time

$$t\left(s_0\right) = t_0 \tag{4.1.2.2}$$

    - Initial velocity

$$v\left(s_0\right) = v_0 \tag{4.1.2.3}$$

  - Final states

* Stop at rest

$$v\left(s_{\mathrm{f}}\right) = 0 \tag{4.1.2.4}$$

- Final arrival time scheduling bounds

$$t_{\mathrm{f,min}} \leq t\left(s_{\mathrm{f}}\right) \leq t_{\mathrm{f,max}} \tag{4.1.2.5}$$

- Time dynamics

$$\frac{dt\left(s\right)}{ds} = \frac{1}{v\left(s\right)} \tag{4.1.2.6}$$

- Velocity dynamics

$$mv\left(s\right)\frac{dv\left(s\right)}{ds} = -C\left(v\left(s\right)\right) + g\left(s\right) + u\left(s\right) \tag{4.1.2.7}$$

- Must either not pass a certain time limit for each value of position, or, must not be behind a certain time limit for each value of position, i.e. must be within a certain time window for each position

$$t_{\mathrm{min}}\left(s\right) \leq t\left(s\right) \leq t_{\mathrm{max}}\left(s\right) \tag{4.1.2.8}$$

- Speed limits

$$0 \leq v\left(s\right) \leq v_{\mathrm{max}}\left(s\right) \tag{4.1.2.9}$$

- Control/actuation limits

$$u_{\mathrm{min}}\left(v\left(s\right)\right) \leq u\left(s\right) \leq u_{\mathrm{max}}\left(v\left(s\right)\right) \tag{4.1.2.10}$$

– Limits on the change in control

$$\mu_{\text{min}}(s) \leq v(s) \frac{d}{ds}(u(s)) \leq \mu_{\text{max}}(s) \tag{4.1.2.11}$$

A note on the above optimal control problem: the formulation above is not tractable in the continuous position domain, but is instead tractable in the discrete position domain.

4.1.2.1 Objective Functions

The objective functions now are [3], [47]:

- Minimum-time:

$$\min_{t(s),v(s),u(s),s} t\left(s_{\text{f}}\right) \tag{4.1.2.12}$$

- Minimum-energy:
  - Minimum-energy with no regenerative braking:

$$\min_{t(s),v(s),u(s),s} \int_{s_0}^{s_{\text{f}}} \max(u(s), 0)\, ds \tag{4.1.2.13}$$

  - Minimum-energy with regenerative braking:

$$\min_{t(s),v(s),u(s),s} \int_{s_0}^{s_{\text{f}}} \{\max(u(s), 0) + \sigma \min(u(s), 0)\}\, ds \tag{4.1.2.14}$$

    where σ is the regenerative braking recovery factor, i.e. the portion of braking energy that is recovered.

4.2 Equal Traction-Braking Delays

Formulations are presented for the case of equal traction-braking delays that are not tractable as explained in each subsection below. These formulations are arrived at by using the input-delay dynamics differential equation in the optimal control problem, instead of the delay-free one. A tractable formulation of the problem will be provided in Chapter 5.

4.2.1 Time Domain

The only difference between the case of a input delay-free system and a single input-delay system is that dynamics equation Eq. (4.1.1.8) is replaced by Eq. (4.2.1.1) in the optimal control formulation.

$$m\frac{dv(t)}{dt} = -C(v(t)) + g(s(t)) + u\left(t - T_{\text{delay}}\right) \tag{4.2.1.1}$$

The current formulation of the problem is not tractable due to the following reasons: the infinite-dimensional nature of the system due to the presence of the delay, and the infinite-dimensional nature of the optimization problem in continuous-time.

4.2.2 Position Domain

The only difference between the case of a input delay-free system and a single input-delay system is that dynamics equation Eq. (4.1.2.7) is replaced by Eq. (4.2.2.1) in the optimal control formulation.

$$mv(s)\frac{dv(s)}{ds} = -C(v(s)) + g(s) + u\left(s - S_{\text{delay}}\right) \tag{4.2.2.1}$$

The current formulation of the problem is not tractable because of the infinite-dimensional nature of the system due to the presence of the delay, and the infinite-dimensional nature of the optimization problem in continuous-position.

4.3 Non-Equal Traction-Braking Delays

Formulations are presented for the case of non-equal traction-braking delays. These formulations are not tractable as explained in each subsection below. These formulations are arrived at by using the non-equal traction-braking input delay dynamics differential equation in the optimal control problem, instead of the delay-free one. A tractable formulation of the problem will be provided in Chapter 7.

4.3.1 Time Domain

$$\min_{s(t),v(t),u_{\text{trk}}(t),u_{\text{brk}}(t),t} \quad J \tag{4.3.1.1}$$

s.t.

$$s\left(t_0\right) = s_0 \tag{4.3.1.2}$$

$$v\left(t_0\right) = v_0 \tag{4.3.1.3}$$

$$s\left(t_{\text{f}}\right) = s_{\text{f}} \tag{4.3.1.4}$$

$$v\left(t_{\text{f}}\right) = 0 \tag{4.3.1.5}$$

$$t_{\text{f,min}} \leq t_{\text{f}} \leq t_{\text{f,max}} \tag{4.3.1.6}$$

$$\frac{ds\left(t\right)}{dt} = v\left(t\right) \tag{4.3.1.7}$$

$$m\frac{dv(t)}{dt} = -C(v(t)) + g(s(t)) + u_{\text{trk}}\left(t - T_{\text{trk,delay}}\right) - u_{\text{brk}}\left(t - T_{\text{brk,delay}}\right) \tag{4.3.1.8}$$

$$s_{\text{min}}(t) \leq s(t) \leq s_{\text{max}}(t) \tag{4.3.1.9}$$

$$0 \leq v(t) \leq v_{\text{max}}(s(t)) \tag{4.3.1.10}$$

$$0 \leq u_{\text{trk}}(t) \leq u_{\text{trk,max}}(v(t)) \tag{4.3.1.11}$$

$$0 \leq u_{\text{brk}}(t) \leq u_{\text{brk,max}}(v(t)) \tag{4.3.1.12}$$

$$u_{\text{trk}}(t)\, u_{\text{brk}}(t) = 0 \tag{4.3.1.13}$$

$$\mu_{\text{trk,min}}(t) \leq \frac{d}{dt}\left(u_{\text{trk}}(t)\right) \leq \mu_{\text{trk,max}}(t) \tag{4.3.1.14}$$

$$\mu_{\text{brk,min}}(t) \leq \frac{d}{dt}\left(u_{\text{brk}}(t)\right) \leq \mu_{\text{brk,max}}(t) \tag{4.3.1.15}$$

The constraint, Eq. (4.3.1.13), is added to ensure that either one of traction or braking is active at any given time. Applying both traction and braking can result in: equipment damage; increased cost of operation and reduced operating efficiency by increasing mechanical wear and tear, and reducing energy efficiency by wastage; reduced operating performance; and may also reduce passenger comfort. Equation (4.3.1.11) is a limit on the tractive force, likewise, Eq. (4.3.1.12), is a limit on the braking force. Equation (4.3.1.14) is a limit on the rate of change of the tractive force, likewise, Eq. (4.3.1.15), is a limit on the rate of change of the braking force.

The current formulation of the problem is not tractable because of the infinite-dimensional nature of the system due to the presence of the delay, and the infinite-dimensional nature of the optimization problem in the continuous domain.

4.3.2 Position Domain

$$\min_{t(s),v(s),u_{\text{trk}}(s),u_{\text{brk}}(s),s} \quad J \tag{4.3.2.1}$$

s.t.

$$t\left(s_0\right) = t_0 \tag{4.3.2.2}$$

$$v\left(s_0\right) = v_0 \tag{4.3.2.3}$$

$$t_{\text{f,min}} \leq t\left(s_{\text{f}}\right) \leq t_{\text{f,max}} \tag{4.3.2.4}$$

$$v\left(s_{\text{f}}\right) = 0 \tag{4.3.2.5}$$

$$\frac{dt\left(s\right)}{ds} = \frac{1}{v\left(s\right)} \tag{4.3.2.6}$$

$$\begin{aligned} mv\left(s\right)\frac{dv\left(s\right)}{ds} &= -C\left(v\left(s\right)\right) + g\left(s\right) \\ &\quad + u_{\text{trk}}\left(s - S_{\text{trk,delay}}\right) \\ &\quad - u_{\text{brk}}\left(s - S_{\text{brk,delay}}\right) \end{aligned} \tag{4.3.2.7}$$

$$t_{\text{min}}\left(s\right) \leq t\left(s\right) \leq t_{\text{max}}\left(s\right) \tag{4.3.2.8}$$

$$0 \leq v\left(s\right) \leq v_{\text{max}}\left(s\right) \tag{4.3.2.9}$$

$$0 \leq u_{\text{trk}}\left(s\right) \leq u_{\text{trk,max}}\left(v\left(s\right)\right) \tag{4.3.2.10}$$

$$0 \leq u_{\text{brk}}\left(s\right) \leq u_{\text{brk,max}}\left(v\left(s\right)\right) \tag{4.3.2.11}$$

$$u_{\text{trk}}\left(s\right) u_{\text{brk}}\left(s\right) = 0 \tag{4.3.2.12}$$

$$\mu_{\text{trk,min}}\left(s\right) \leq v\left(s\right)\frac{d}{ds}\left(u_{\text{trk}}\left(s\right)\right) \leq \mu_{\text{trk,max}}\left(s\right) \tag{4.3.2.13}$$

$$\mu_{\text{brk,min}}\left(s\right) \leq v\left(s\right)\frac{d}{ds}\left(u_{\text{brk}}\left(s\right)\right) \leq \mu_{\text{brk,max}}\left(s\right) \tag{4.3.2.14}$$

Note that for Eq. (4.3.2.7), $s - S_{\text{trk,delay}}$ is the position $T_{\text{trk,delay}}$ seconds ago, i.e. $t\left(s - S_{\text{trk,delay}}\right) = t\left(s\right) - T_{\text{trk,delay}}$; and for Eq. (4.3.2.7), $s - S_{\text{brk,delay}}$ is the position $T_{\text{brk,delay}}$ seconds ago, i.e. $t\left(s - S_{\text{brk,delay}}\right) = t\left(s\right) - T_{\text{brk,delay}}$.

The problem with non-equal traction-braking delays is not tractable using position as the independent variable. The problem is not tractable because $S_{\text{brk,delay}}$, and $S_{\text{trk,delay}}$ are nonlinear functions of past state trajectories, i.e. the problem is of an infinite-dimensional nature.

Chapter 5

Equal Traction-Braking Delays

The optimal control formulation, used in Algorithm 1, and based on the approximation in [13] resulting in a convex optimization problem is given below. A summary of the convex approximation is provided in Appendix A for ease of reference. The continuous position and the discrete position formulations are provided.

5.1 Continuous-Position

$$\min_{t(s),v(s),\rho(s),\gamma(s),\gamma^{+}(s)} \quad \int_{s_0}^{s_\text{f}} w_e \gamma^{+}(s)\, ds + w_\rho t\left(s_\text{f}\right) \tag{5.1.1.1}$$

s.t.

$$t\left(s_0\right) = t_\text{Predicted} \tag{5.1.1.2}$$

$$s\left(s_0\right) = s_\text{Predicted} \tag{5.1.1.3}$$

$$v\left(s_0\right) = v_\text{Predicted} \tag{5.1.1.4}$$

$$t_\text{min} \leq t\left(s_\text{f}\right) \leq t_\text{max} \tag{5.1.1.5}$$

$$v\left(s_{\mathrm{f}}\right) = 0\,\mathrm{m/s} \tag{5.1.1.6}$$

$$\frac{dt\left(s\right)}{ds} = \rho\left(s\right) \tag{5.1.1.7}$$

$$\frac{dv\left(s\right)}{ds} = g\left(s\right) - \frac{C_0}{m}\rho\left(s\right) - \frac{C_v}{m} - \frac{C_{v^2}}{m}v\left(s\right) + \gamma\left(s\right) \tag{5.1.1.8}$$

$$\left\| \begin{bmatrix} 2 \\ v\left(s\right) - \rho\left(s\right) \end{bmatrix} \right\|_2 \leq v\left(s\right) + \rho\left(s\right) \tag{5.1.1.9}$$

$$0 \leq \gamma^{+}\left(s\right) \tag{5.1.1.10}$$

$$\gamma\left(s\right) \leq \gamma^{+}\left(s\right) \tag{5.1.1.11}$$

$$0\,\mathrm{m/s} < v\left(s\right) \leq v_{\mathrm{max}}\left(s\right) \tag{5.1.1.12}$$

$$0\,\mathrm{s/m} < \rho\left(s\right) < \infty\,\mathrm{s/m} \tag{5.1.1.13}$$

$$\left(u_{\mathrm{brk,max}}\right)\rho\left(s\right) \leq \gamma\left(s\right) \tag{5.1.1.14}$$

$$\gamma\left(s\right) \leq u_{\mathrm{trk,max}}\rho\left(s\right) \tag{5.1.1.15}$$

$$\gamma\left(s\right) \leq r_0\rho\left(s\right) + r_v + r_{v^2}v\left(s\right) \tag{5.1.1.16}$$

5.2 Discrete-Position

The optimal control formulation, for a train at a position marker $s\left[q\right]$, is given by:

$$\min_{t[k],v[k],\rho[k],\gamma[k],\gamma^{+}[k],k\in\{q,\ldots,N\}} \quad \sum_{i=q}^{N-1}\left\{w_e\gamma^{+}\left[i\right]\delta s\left[i\right] + w_\rho\rho\left[i\right]\right\} \tag{5.2.1.1}$$

s.t.

$$t\left[q\right] = t_{\mathrm{Predicted}} \tag{5.2.1.2}$$

$$s\left[q\right] = s_{\mathrm{Predicted}} \tag{5.2.1.3}$$

$$v\left[q\right] = v_{\mathrm{Predicted}} \tag{5.2.1.4}$$

$$t_{\text{min}} \leq t\,[N] \leq t_{\text{max}} \tag{5.2.1.5}$$

$$s\,[N] = s_{\text{f}} \tag{5.2.1.6}$$

$$v\,[N] = 0\,\text{m/s} \tag{5.2.1.7}$$

for $k \in \{q, \ldots, N-1\}$:

$$t\,[k+1] = t\,[k] + \rho\,[k]\,\delta s\,[k] \tag{5.2.1.8}$$

$$\begin{aligned} v\,[k+1] = u_{\text{brk,max}}\,[k] + \Bigg\{ g\,[k] - \frac{C_0}{m}\rho\,[k] \\ - \frac{C_v}{m} \\ - \frac{C_{v^2}}{m} v\,[k] \\ + \gamma\,[k] \Bigg\}\,\delta s\,[k] \end{aligned} \tag{5.2.1.9}$$

for $k \in \{q, \ldots, N\}$:

$$\left\| \begin{bmatrix} 2 \\ v\,[k] - \rho\,[k] \end{bmatrix} \right\|_2 \leq v\,[k] + \rho\,[k] \tag{5.2.1.10}$$

$$0 \leq \gamma^{+}\,[k] \tag{5.2.1.11}$$

$$\gamma\,[k] \leq \gamma^{+}\,[k] \tag{5.2.1.12}$$

$$0\,\text{m/s} < v\,[k] \leq v_{\text{max}}\,[k] \tag{5.2.1.13}$$

$$0\,\text{s/m} < \rho\,[k] < \infty\,\text{s/m} \tag{5.2.1.14}$$

$$\left(u_{\text{brk,max}}\right)\rho\,[k] \leq \gamma\,[k] \tag{5.2.1.15}$$

$$\gamma\,[k] \leq \left(u_{\text{trk,max}}\right)\rho\,[k] \tag{5.2.1.16}$$

$$\gamma\,[k] \leq r_0\rho\,[k] + r_v + r_{v^2} v\,[k] \tag{5.2.1.17}$$

The objective, Eq. (5.2.1.1), includes a weighted sum of tractive force applied, for minimizing energy, and a weighted sum of $\rho[k]$, for minimizing time and more importantly to minimize the error between $\rho[k]$ and $\frac{1}{v[k]}$, i.e. $\left|\rho[k] - \frac{1}{v[k]}\right|$, so that the approximated dynamics for the convex approximation is close (approximately equal) to the actual dynamics. Minimizing only the sum of $\rho[k]$ corresponds to a minimum-time journey. A sufficiently large value for the weight on $\rho[k]$, w_ρ, is always required regardless of whether the journey is minimum-energy or minimum-time to ensure that the approximated dynamics in the convex formulation match the actual dynamics of the system.

5.3 Algorithm

The solution procedure for optimally operating or driving the train from Station A to Station B for the case of equal traction-braking delays is detailed in Algorithm 1 and a high-level overview is given as follows. The position discretization nodes, denoted by s_i in Fig. 5.8, are selected once initially from the origin to the destination before beginning the trip. The discretization nodes are not modified again. The optimal control problem is successively solved at the pre-selected discretization nodes, i.e. see Fig. 5.8. The set of remaining position discretization nodes is prepended with an additional discretization node, $s[q]$, for the predicted states, time and velocity, at the predicted position. This allows the initial conditions, time and velocity, and continuity of control, current fixed control, to be fixed at the predicted position discretization node. This additional node is later discarded when the optimal solution is returned from the solver.

At each of the remaining discretization nodes, the optimal control solution returns the control commands, and time and velocity states. The time and control values are used in an interpolation scheme to generate control command values at regular control time steps of T_S, which are then issued to the train propulsion and braking systems.

An overview of the prediction process in Algorithm 1 is presented below. Note that for equal traction-braking delays, delay-compensation is achieved using model-based prediction.

1. Using the knowledge of the previous control inputs to the system, and the train dynamics, predict the effect on the system using a model of the system, i.e. what is the expected position and velocity. This is shown in Fig. 5.5.

    - Initially it can be assumed that the past control history consists of all zero traction and braking commands. If the initial states of the system are at rest (stationary position, zero velocity, zero acceleration) and the past control history is all zero traction and braking commands. Issuing traction and braking commands of zero magnitude to a train at rest will not cause the train to move, i.e. the states of the train do not change.

2. Use the expected states (position and velocity) as the initial conditions in a new optimization problem to determine the control inputs from $t + T_{\text{delay}}$ onwards. This is shown in Fig. 5.6. In this optimization problem, ignore the input delay in the dynamics (state evolution) equations. The predictor synchronizes the system state with the control, i.e. we know the value of the states after all the control values in the buffer have had an effect. The control commands present in the buffer are commands that have been issued but have not yet had an effect on the system. Thus, the dynamics equations will be the same as in Eq. (3.1.2.2).

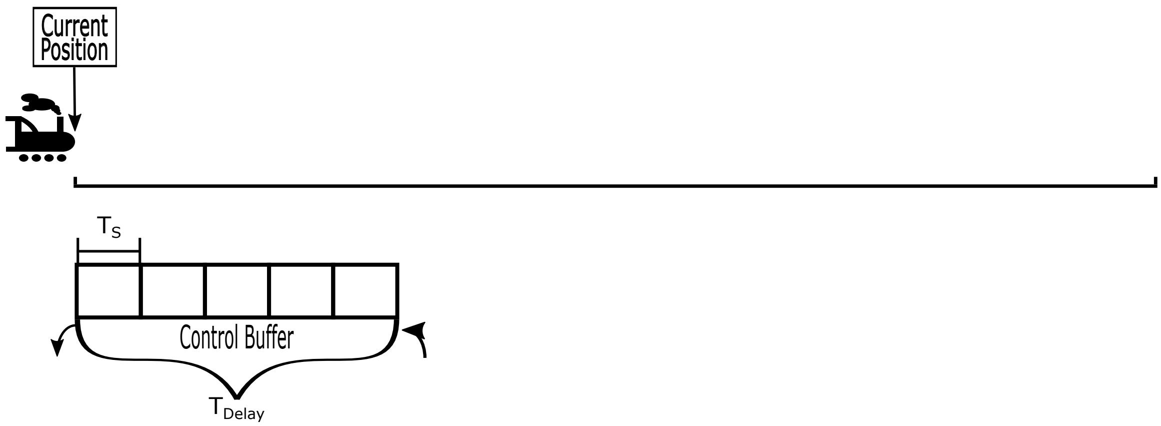

Figure 5.1: Control buffer.The arrow pointing into the control buffer shows where the control commands enter the buffer, and the arrow pointing out of the control buffer shows where the control commands exit the buffer.

- The set of discretization nodes in the optimization formulation includes all remaining discretization nodes from the predicted position till the end, with the addition of the predicted position.

3. Apply a select number of the control commands (solved for by the solution to the optimization problem) to the system, as shown in Fig. 5.7. The control commands will not take effect immediately but rather after a delay.

4. Repeat this procedure. This is shown in Fig. 5.8.

Note that the history of the control input trajectory is required for the time period starting from the current time, $t(s)$, and going back in time to $t\left(s - S_{\text{delay}}\right)$. The quantity, $t\left(s - S_{\text{delay}}\right)$, is simply equal to $t(s) - T_{\text{delay}}$, which greatly simplifies the calculation of $t\left(s - S_{\text{delay}}\right)$, since, S_{delay}, is a function of the past state trajectory.

Use of the predictor transforms the problem from control of a delayed input system to the control of a delay-free input system.

Algorithm 1: Optimal control of train with equal traction-braking delays

begin
  Initialize T_S
  Initialize s [k] for $k \in \{0, \ldots, N\}$
  $t \leftarrow 0$
  $t_{\text{Sample Start}} \leftarrow 0$
  $S_{\text{Next}} \leftarrow 0$
  Measure t, s, v
  while $s < s_f$ **do**
    if $s \geq S_{Next}$ **then**
      Function *Prediction(Guideway Data, t, s, v, $\{u_{buf}[0], \ldots, u_{buf}[j]\}$):* **is**
        $t_{\text{Start}} \leftarrow t$
        $t_{\text{Predicted}} \leftarrow t$
        while $t_{Predicted}$-$t_{Start} < T_{Delay}$ **do**
          Integrate ODE using values from the control buffer
          $t_{\text{Predicted}} \leftarrow t_{\text{Predicted}}+\text{dt}$
        return $t_{Predicted}$, $s_{Predicted}$, $v_{Predicted}$
      Function *Optimization(Guideway Data, $t_{Predicted}$, $s_{Predicted}$, $v_{Predicted}$, $\{s[q], \ldots, s[N]\}$):* **is**
        Optimize
        return ($\{t^*[q], \ldots, t^*[N]\}$, $\{v^*[q], \ldots, v^*[N]\}$, $\{u^*[q], \ldots, u^*[N]\}$)
      $S_{\text{Next}} \leftarrow$ Find next discretization node greater than S_{Next}
    if t-$t_{Sample\ Start} \geq T_S$ **then**
      $t_{\text{Sample Start}} \leftarrow t$
      $u_{\text{Current}} \leftarrow$ `Interpolate(` $t+T_{delay}$, $\{t^*[q], \ldots, t^*[N]\}$, $\{u^*[q], \ldots, u^*[N]\}$)

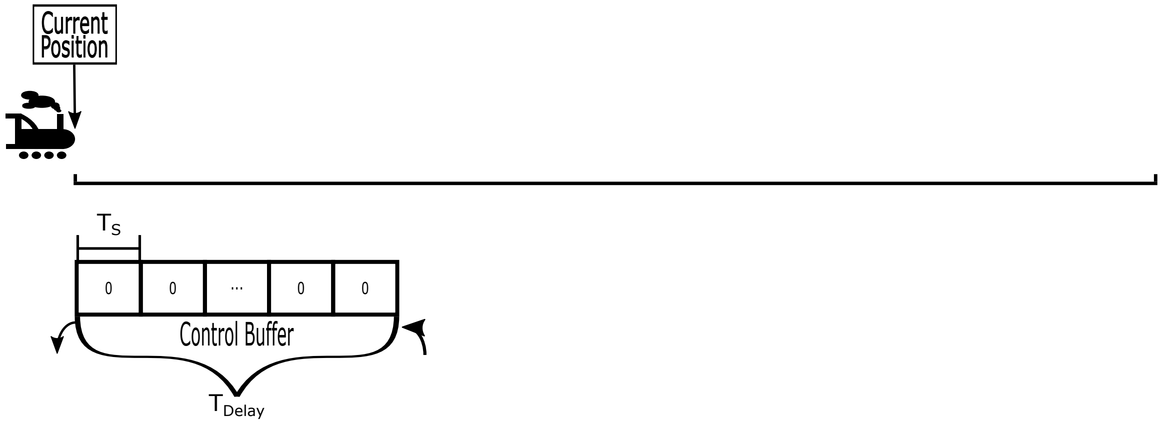

Figure 5.2: Control buffer filled with zeros. Initially, since the system is at rest, the control buffer is filled with zeros.

The results in this thesis for the case of equal traction-braking delays are derived using the convex approximation as presented in [13]. The convex approximation can be applied for fast and robust solution of the approximate train optimal control problem. If the two inputs have different delays, then an artificial delay can be inserted using hardware such that the two control inputs now have the same delay, thus allowing the convex approximation to be applied.

The problem with multiple inputs with different time-delays is not tractable using position as the independent variable. The problem becomes tractable using time as the independent variable as will be explained in Chapter 7. However, in such a case, it is also possible to artificially delay the input commands with shorter delays so that all input channels would have equal delays, and then apply the control approach presented in this section. These two approaches to handling of non-equal delays will be compared in Chapter 8.

Figure 5.3: Control buffer after the first control input is applied. The first control sample will remain in the buffer for the length of the delay after which it will exit the buffer and thereupon affect the system.

Figure 5.4: Control buffer filled.

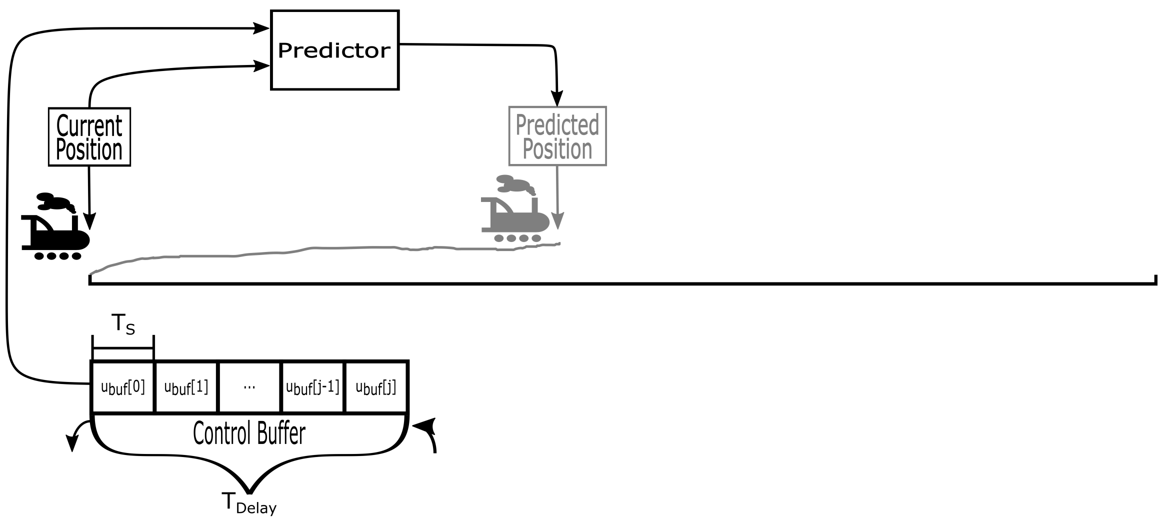

Figure 5.5: Predicted states using the samples in the control buffer.

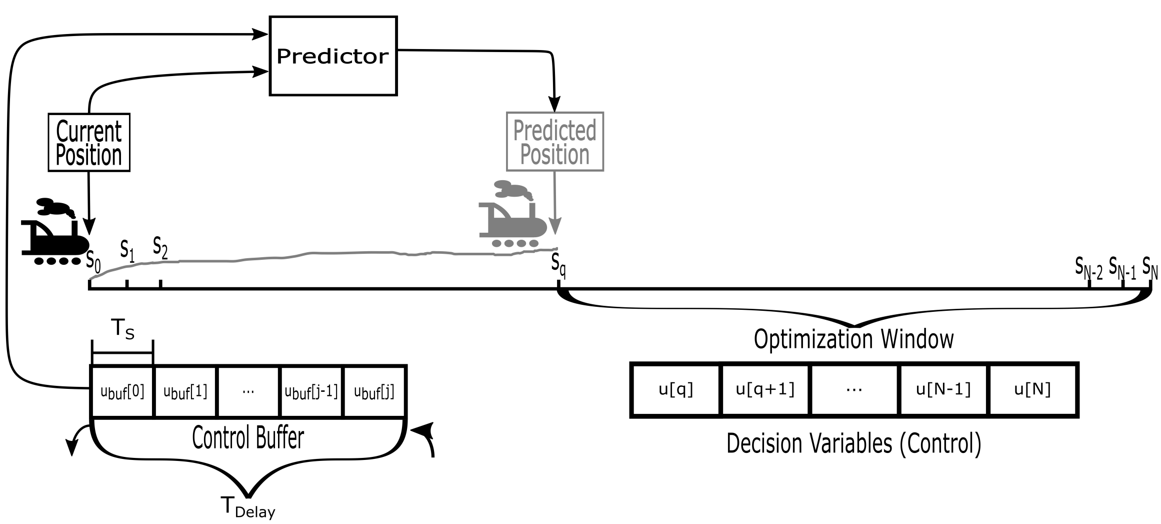

Figure 5.6: Optimization problem using the predicted states as the initial condition on the system time and velocity in the optimization problem.

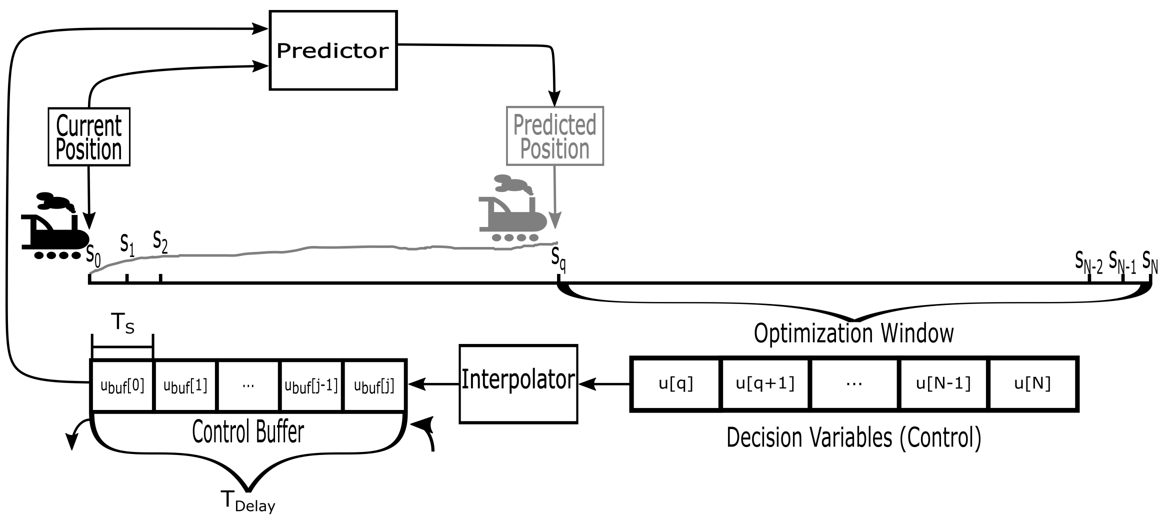

Figure 5.7: Application of the newly determined control inputs.

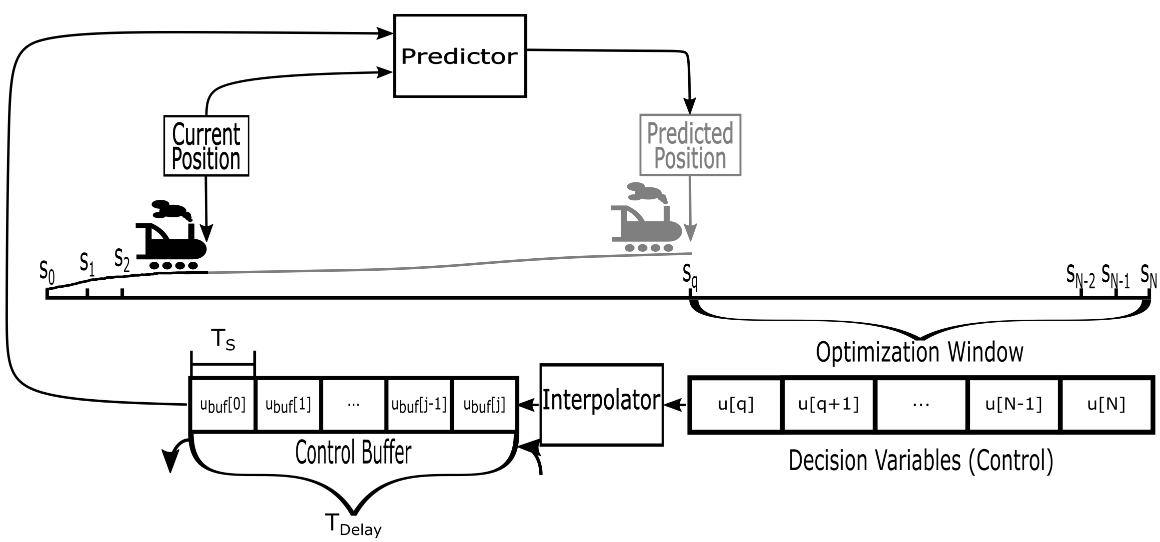

Figure 5.8: Application of the control algorithm at an arbitrary instant.

Chapter 6

Alternative Formulation to Minimum-Time Journeys

An alternative formulation of the optimal control problem is developed, using time as the independent variable, that is more suited to minimum-time problems. The formulation is initially provided for the single-input delay-free case, i.e. zero traction-braking delays.

We plan to use a change of variable from t to τ using the following transformation that maps t to τ , as defined in [15]. Note that, t takes on values in the interval:

$$t \in \left[t_0, t_{\mathrm{f}}\right] \tag{6.0.1}$$

and that, τ takes on values in the interval:

$$\tau \in [0, 1] \tag{6.0.2}$$

The transformation is as follows:

$$\tau = \frac{t - t_0}{t_\mathrm{f} - t_0} \tag{6.0.3}$$

The inverse transformation is as follows:

$$t = \left(t_\mathrm{f} - t_0\right)\tau + t_0 \tag{6.0.4}$$

This allows us to express the following derivatives as:

$$\frac{ds\left(t\right)}{dt} = \frac{ds\left(\tau\right)}{d\tau}\frac{d\tau}{dt} = \frac{ds\left(\tau\right)}{d\tau}\frac{1}{h} \tag{6.0.5}$$

$$\frac{dv\left(t\right)}{dt} = \frac{dv\left(\tau\right)}{d\tau}\frac{d\tau}{dt} = \frac{dv\left(\tau\right)}{d\tau}\frac{1}{h} \tag{6.0.6}$$

where we have denoted

$$\frac{d\tau}{dt} = \frac{1}{h} = \frac{1}{t_\mathrm{f} - t_0} \tag{6.0.7}$$

or

$$h = t_\mathrm{f} - t_0 \tag{6.0.8}$$

6.1 Continuos τ-Domain

The optimal control problem using τ as the independent variable, in the continuous domain, is given in Eq. (6.1.1).

$$\min_{h,s(\tau),v(\tau),u(\tau)} \quad J \tag{6.1.1.1}$$

s.t.

$$s(\tau = 0) = s_0 \tag{6.1.1.2}$$

$$v(\tau = 0) = v_0 \tag{6.1.1.3}$$

$$s(\tau = 1) = s_{\text{f}} \tag{6.1.1.4}$$

$$v(\tau = 1) = 0 \tag{6.1.1.5}$$

$$t(\tau) = t_0 + h\tau \tag{6.1.1.6}$$

$$t_{\text{f,min}} \leq t(\tau = 1) \leq t_{\text{f,max}} \tag{6.1.1.7}$$

$$\frac{ds(\tau)}{d\tau}\frac{1}{h} = v(\tau) \tag{6.1.1.8}$$

$$m\frac{dv(\tau)}{d\tau}\frac{1}{h} = -C(v(\tau)) + g(s(\tau)) + u(\tau) \tag{6.1.1.9}$$

$$s_{\text{min}}(\tau) \leq s(\tau) \leq s_{\text{max}}(\tau) \tag{6.1.1.10}$$

$$0 \leq v(\tau) \leq v_{\text{max}}(s(\tau)) \tag{6.1.1.11}$$

$$u_{\text{min}}(\tau) \leq u(\tau) \leq u_{\text{max}}(\tau) \tag{6.1.1.12}$$

$$\mu_{\text{min}}(\tau) \leq \frac{1}{h}\frac{d}{d\tau}(u(\tau)) \leq \mu_{\text{max}}(\tau) \tag{6.1.1.13}$$

6.1.1 Objective Functions

6.1.1.1 Minimum-Time

$$\min_{h,s(\tau),v(\tau),u(\tau)} h \tag{6.1.2}$$

6.1.1.2 Minimum-Energy

$$\min_{h,s(\tau),v(\tau),u(\tau)} \int_{\tau=0}^{\tau=1} \max(u(\tau), 0)\ v(\tau)\ h\, d\tau \tag{6.1.3}$$

6.2 Discrete τ-Domain

The optimal control problem using τ as the independent variable, in the discrete domain, is given in Eq. (6.2.1).

$$\min_{h,s[k],v[k],u[k],k\in\{0,\ldots,N\}} \quad J \tag{6.2.1.1}$$

s.t.

$$t[0] = t_0 \tag{6.2.1.2}$$

$$s[0] = s_0 \tag{6.2.1.3}$$

$$v[0] = v_0 \tag{6.2.1.4}$$

$$s[N] = s_{\mathrm{f}} \tag{6.2.1.5}$$

$$v[N] = 0 \tag{6.2.1.6}$$

$$t_{\mathrm{f,min}} \le t[N] \le t_{\mathrm{f,max}} \tag{6.2.1.7}$$

for $k \in [1,\ldots,N]$:

$$s[k] = f_1\left(h, s[0],\ldots,s[N], v[0],\ldots,v[N]\right) \tag{6.2.1.8}$$

$$v[k] = f_2\left(h, v[0],\ldots,v[N], u[0],\ldots,u[N]\right) \tag{6.2.1.9}$$

$$t[k] = t[0] + h\sum_{i=0}^{i=k-1} \delta\tau[i] \tag{6.2.1.10}$$

for $k \in [0,\ldots,N]$:

$$s_{\mathrm{min}}[k] \le s[k] \le s_{\mathrm{max}}[k] \tag{6.2.1.11}$$

$$0 \le v[k] \le v_{\mathrm{max}}[s[k]] \tag{6.2.1.12}$$

$$u_{\min}[k] \leq u[k] \quad (6.2.1.13)$$

$$u[k] \leq u_{\max}[k] \quad (6.2.1.14)$$

for $k \in [0, \ldots, N-1]$:

$$\mu_{\min}[k] \leq \frac{1}{h}\frac{\Delta u[k]}{\Delta \tau[k]} \quad (6.2.1.15)$$

$$\frac{1}{h}\frac{\Delta u[k]}{\Delta \tau[k]} \leq \mu_{\max}[k] \quad (6.2.1.16)$$

6.2.1 Objective Functions

6.2.1.1 Minimum-Time

$$\min_{h,s[k],v[k],u[k]} h \quad (6.2.2)$$

6.2.1.2 Minimum-Energy

Using the trapezoidal integration rule [89], the minimum-energy objective function is given below.

$$\min_{h,s[k],v[k],u[k]} \sum_{i=0}^{i=N-1} \frac{(\max[u[k],0]\,v[k] + \max[u[k+1],0]\,v[k+1])}{2} h\delta\tau[i] \quad (6.2.3)$$

Some explanation is now needed for the formulation in Eqs. (6.1.1) and (6.2.1). The independent variable in this case is not t, but rather the variable τ. The relationship between t and τ is an affine one. The quantity, h, which is defined in Eq. (6.0.8), is a scalar quantity that determines the trip time. This is different compared to the position-domain as is explained in the following. The path of travel is known and fixed. It is the velocity trajectory that determines the travel time. The goal is to determine the velocity profile (a function) over the path that will minimize travel time. In this formulation, the velocity profile, which is a function, is determined such that the position profile that follows from the velocity profile, satisfies the constraints of starting from the origin and arriving at the destination. The velocity profile and the position profile that follows from the velocity profile, determine the magnitude of the single variable, h, which determines the trip time. The variable, h, is used in the dynamics equations that describe the behaviour of the commuter train system.

Chapter 7

Non-Equal Traction-Braking Delays

The non-equal traction-braking delay case is dealt with the use of (i) a predictor, model-based prediction, to compensate for the "common" portion of the delays; (ii) use of the alternative formulation for minimum-time journeys, i.e. use of the τ-domain; and (iii) use of either traction and braking command buffers, whichever is longer, to serve as constraints on the traction or braking control profile, respectively. This is explained in more detail in this section. The optimal control formulation, used in Algorithm 2, is given below. The continuous time and the discrete time formulations are provided.

7.1 Continuous Time

The continuous time formulation is presented in Eq. (7.1.1).

$$\min_{h,s,v,u,t,\tau} \quad J \tag{7.1.1.1}$$

s.t.

$$s\left(t = t_{\text{Predicted}}\right) = s_{\text{Predicted}} \tag{7.1.1.2}$$

$$v\left(t = t_{\text{Predicted}}\right) = v_{\text{Predicted}} \tag{7.1.1.3}$$

for $t \in \left[t_{\text{Predicted}}, t_{\text{qe}}\right]$:

$$\frac{ds\left(t\right)}{dt} = v\left(t\right) \tag{7.1.1.4}$$

$$m\frac{dv\left(t\right)}{dt} = -C\left(v\left(t\right)\right) + g\left(s\left(t\right)\right) + u_{\text{trk}}\left(t\right) - u_{\text{brk}}\left(t\right) \tag{7.1.1.5}$$

$$u_{\text{brk}}\left(t\right) = \bar{u}_{\text{brk}}\left(t\right) \tag{7.1.1.6}$$

$$s_{\text{min}}\left(t\right) \leq s\left(t\right) \leq s_{\text{max}}\left(t\right) \tag{7.1.1.7}$$

$$0 \leq v\left(t\right) \leq v_{\text{max}}\left(s\left(t\right)\right) \tag{7.1.1.8}$$

$$0 \leq u_{\text{trk}}\left(t\right) \leq u_{\text{trk,max}}\left(v\left(t\right)\right) \tag{7.1.1.9}$$

$$0 \leq u_{\text{brk}}\left(t\right) \leq u_{\text{brk,max}}\left(v\left(t\right)\right) \tag{7.1.1.10}$$

$$u_{\text{trk}}\left(t\right) u_{\text{brk}}\left(t\right) = 0 \tag{7.1.1.11}$$

$$\frac{d}{dt}\left(u_{\text{trk}}\left(t\right)\right) \leq \mu_{\text{trk,max}}\left(t\right) \tag{7.1.1.12}$$

$$\mu_{\text{trk,min}}\left(t\right) \leq \frac{d}{dt}\left(u_{\text{trk}}\left(t\right)\right) \tag{7.1.1.13}$$

$$\frac{d}{dt}\left(u_{\text{brk}}\left(t\right)\right) \leq \mu_{\text{brk,max}}\left(t\right) \tag{7.1.1.14}$$

$$\mu_{\text{brk,min}}\left(t\right) \leq \frac{d}{dt}\left(u_{\text{brk}}\left(t\right)\right) \tag{7.1.1.15}$$

$$s\left(\tau = 0\right) = s\left(t_{\text{qe}}\right) \tag{7.1.1.16}$$

$$v\left(\tau = 0\right) = v\left(t_{\text{qe}}\right) \tag{7.1.1.17}$$

$$t_{\text{f,min}} \leq t\left(\tau = 1\right) \leq t_{\text{f,max}} \tag{7.1.1.18}$$

$$s(\tau = 1) = s_{\text{f}} \tag{7.1.1.19}$$

$$v(\tau = 1) = 0 \tag{7.1.1.20}$$

for $t \in \left(t_{\text{qe}}, t_N\right], \tau \in [0,1]$:

$$t(\tau) = t_{\text{qe}} + h\tau \tag{7.1.1.21}$$

$$\frac{ds(\tau)}{d\tau}\frac{1}{h} = v(\tau) \tag{7.1.1.22}$$

$$\begin{aligned} m\frac{dv(\tau)}{d\tau}\frac{1}{h} = &-C(v(\tau)) \\ &+ g(s(\tau)) \\ &+ u_{\text{trk}}(\tau) \\ &- u_{\text{brk}}(\tau) \end{aligned} \tag{7.1.1.23}$$

$$s_{\text{min}}(\tau) \leq s(\tau) \leq s_{\text{max}}(\tau) \tag{7.1.1.24}$$

$$0 \leq v(\tau) \leq v_{\text{max}}(s(\tau)) \tag{7.1.1.25}$$

$$0 \leq u_{\text{trk}}(\tau) \leq u_{\text{trk,max}}(v(\tau)) \tag{7.1.1.26}$$

$$0 \leq u_{\text{brk}}(\tau) \leq u_{\text{brk,max}}(v(\tau)) \tag{7.1.1.27}$$

$$u_{\text{trk}}(\tau)\, u_{\text{brk}}(\tau) = 0 \tag{7.1.1.28}$$

$$\mu_{\text{trk,min}}(\tau) \leq \frac{1}{h}\frac{d}{d\tau}\left(u_{\text{trk}}(\tau)\right) \tag{7.1.1.29}$$

$$\frac{1}{h}\frac{d}{d\tau}\left(u_{\text{trk}}(\tau)\right) \leq \mu_{\text{trk,max}}(\tau) \tag{7.1.1.30}$$

$$\mu_{\text{brk,min}}(\tau) \leq \frac{1}{h}\frac{d}{d\tau}\left(u_{\text{brk}}(\tau)\right) \tag{7.1.1.31}$$

$$\frac{1}{h}\frac{d}{d\tau}\left(u_{\text{brk}}(\tau)\right) \leq \mu_{\text{brk,max}}(\tau) \tag{7.1.1.32}$$

7.1.1 Objective Functions

7.1.1.1 Minimum-Time

$$\min_{h,s,v,u,t,\tau} h \tag{7.1.1.33}$$

7.1.1.2 Minimum-Energy

$$\min_{h,s,v,u,t,\tau} \int_{t=t_{\text{Predicted}}}^{t=t_{\text{qe}}} u_{\text{trk}}(t)\ v(t)\ dt + \int_{\tau=0}^{\tau=1} u_{\text{trk}}(\tau)\ v(\tau)\ h\, d\tau \tag{7.1.1.34}$$

Note, however, that this problem is still not tractable, since it is still infinite-dimensional. The reason is that the constraint, Eq. (7.1.1.6), requires knowledge of function values over a continuous interval which is infinite-dimensional, i.e. requires an infinite amount of memory. The problem is made tractable by recognizing that the control commands issued to the train traction and braking systems are piecewise constant. A piecewise constant function can be stored in a finite amount of memory because only a finite number of samples are necessary to be able to characterize the entire function [67].

The problem is made tractable by assuming that the inputs are piecewise constant [67], and by use of the predictor which allows for compensating the "common" portion of the delays by predicting the states using a mathematical model of the train, and traction and braking control commands in the delay buffers. The discrete-time optimal control formulation is detailed in Section 7.2.

7.2 Discrete-Time

The discrete-time formulation is presented in Eq. (7.2.1). The optimal control formulation, for a train at a position marker $s_{\text{Milepost}}[q]$, is given by:

$$\min_{h,s[k],v[k],u_{\text{trk}}[k],u_{\text{brk}}[k],k\in\{0,\ldots,N+n_{\text{cc}}\}} \quad J \tag{7.2.1.1}$$

s.t.

$$t[0] = t_{\text{Predicted}} \tag{7.2.1.2}$$

$$s[0] = s_{\text{Predicted}} \tag{7.2.1.3}$$

$$v[0] = v_{\text{Predicted}} \tag{7.2.1.4}$$

$$t_{\text{f,min}} \leq t[N+n_{\text{cc}}] \leq t_{\text{f,max}} \tag{7.2.1.5}$$

$$s[N+n_{\text{cc}}] = s_{\text{f}} \tag{7.2.1.6}$$

$$v[N+n_{\text{cc}}] = 0 \tag{7.2.1.7}$$

for $k \in [1,\ldots,N+n_{\text{cc}}]$:

$$s_{\text{min}}[k] \leq s[k] \leq s_{\text{max}}[k] \tag{7.2.1.8}$$

$$0 \leq v[k] \leq v_{\text{max}}[s[k]] \tag{7.2.1.9}$$

for $k \in [0,\ldots,N+n_{\text{cc}}]$:

$$u_{\text{trk}}[k]\, u_{\text{brk}}[k] = 0 \tag{7.2.1.10}$$

$$0 \leq u_{\text{trk}}[k] \leq u_{\text{trk,max}}[v[k]] \tag{7.2.1.11}$$

$$0 \leq u_{\text{brk}}[k] \leq u_{\text{brk,max}}[v[k]] \tag{7.2.1.12}$$

for $k \in [1,\ldots,n_{\text{cc}}]$:

$$\begin{aligned} s[k] = f_1 (&s[0],\ldots,s[n_{\text{cc}}], \\ &v[0],\ldots,v[n_{\text{cc}}]) \end{aligned} \tag{7.2.1.13}$$

$$v[k] = f_2 (v[0], \ldots, v[n_{cc}], u_{\text{trk}}[0], \ldots, u_{\text{trk}}[n_{cc} - 1], u_{\text{brk}}[0], \ldots, u_{\text{brk}}[n_{cc} - 1]) \tag{7.2.1.14}$$

$$\frac{\Delta u_{\text{trk}}[k]}{\Delta t[k]} \leq \mu_{\text{trk,max}} \tag{7.2.1.15}$$

$$-\frac{\Delta u_{\text{trk}}[k]}{\Delta t[k]} \leq -\mu_{\text{trk,min}} \tag{7.2.1.16}$$

$$\frac{\Delta u_{\text{brk}}[k]}{\Delta t[k]} \leq \mu_{\text{brk,max}} \tag{7.2.1.17}$$

$$-\frac{\Delta u_{\text{brk}}[k]}{\Delta t[k]} \leq -\mu_{\text{brk,min}} \tag{7.2.1.18}$$

for $k \in [0, \ldots, n_{cc} - 1]$:

$$u_{\text{brk}}[k] = \bar{u}_{\text{brk}}[k] \tag{7.2.1.19}$$

for $k \in [n_{cc} + 1, \ldots, N + n_{cc}]$:

$$s[k] = f_3 (h, s[n_{cc}], \ldots, s[N + n_{cc}], v[n_{cc}], \ldots, v[N + n_{cc}]) \tag{7.2.1.20}$$

$$v[k] = f_4 (h, v[n_{cc}], \ldots, v[N + n_{cc}], u_{\text{trk}}[n_{cc}], \ldots, u_{\text{trk}}[N + n_{cc}], u_{\text{brk}}[n_{cc}], \ldots, u_{\text{brk}}[N + n_{cc}]) \tag{7.2.1.21}$$

$$\frac{1}{h} \frac{\Delta u_{\text{trk}}[k]}{\Delta \tau [k - (n_{cc} + 1)]} \leq \mu_{\text{trk,max}}[k] \tag{7.2.1.22}$$

$$\frac{1}{h}\frac{\Delta u_{\text{trk}}[k]}{\Delta\tau[k-(n_{\text{cc}}+1)]} \geq \mu_{\text{trk,min}}[k] \quad (7.2.1.23)$$
$$\frac{1}{h}\frac{\Delta u_{\text{brk}}[k]}{\Delta\tau[k-(n_{\text{cc}}+1)]} \leq \mu_{\text{brk,max}}[k] \quad (7.2.1.24)$$
$$\frac{1}{h}\frac{\Delta u_{\text{brk}}[k]}{\Delta\tau[k-(n_{\text{cc}}+1)]} \geq \mu_{\text{brk,min}}[k] \quad (7.2.1.25)$$
$$t[k] = t[n_{\text{cc}}] + h \sum_{i=0}^{i=k-(n_{\text{cc}}+1)} \delta\tau[i] \quad (7.2.1.26)$$

Equations (7.2.1.13) to (7.2.1.19) are constraints in the “constraints on control” region in Fig. 7.1. Equations (7.2.1.20) to (7.2.1.25) are constraints after the “constraints on control” region and in the “optimization window” region in Fig. 7.1. Equations (7.2.1.8) to (7.2.1.12) apply for discretization nodes in both the time and τ-domain regions, i.e. “optimization window” region in Fig. 7.1.

Equations (7.2.1.13), (7.2.1.14), (7.2.1.20) and (7.2.1.21) are state, position and velocity, evolution constraints and are left general as any discretization scheme can be used. Equation (7.2.1.19) serves as constraints on the control input with the longer delay, in this case braking, as shown in the “constraints on control” region in Fig. 7.1. Equation (7.2.1.10) ensures that both traction and braking are not active at the same time as it may result in equipment damage or increased wear and tear. Equation (7.2.1.8) ensures that the train departs from the origin and arrives at the destination without overshooting. Equation (7.2.1.9) are speed limits. Equation (7.2.1.11) is maximum traction limit. Equation (7.2.1.12) is maximum braking limit. Equation (7.2.1.26) determines the time state at each discretization node in the τ-region, and Eq. (7.2.1.5) ensures that the train arrives within the scheduled arrival window. Equations (7.2.1.15) to (7.2.1.18), time-domain, and Eqs. (7.2.1.22) to (7.2.1.25), τ-domain, are limits on the change in control due to equipment limitations, equipment protection, and passenger comfort.

A select few sample objective functions are presented in Eqs. (7.2.1.27) and (7.2.1.28). Note that a zero-order hold is assumed on the control input for samples associated with T_{S} (time-domain), while a first-order hold is assumed on the control input for samples associated with $\delta\tau\,[k]$ (τ-domain).

7.2.1 Objective Functions

7.2.1.1 Minimum-Time

$$\min_{h,s[k],v[k],u_{\mathrm{trk}}[k],u_{\mathrm{brk}}[k]} h \tag{7.2.1.27}$$

7.2.1.2 Minimum-Energy

$$\begin{aligned}\min_{h,s[k],v[k],u_{\mathrm{trk}}[k],u_{\mathrm{brk}}[k]} & \sum_{k=0}^{k=n_{\mathrm{cc}}-1} p_{\mathrm{in}}\,[k]\; T_{\mathrm{S}} \\ & + \sum_{k=n_{\mathrm{cc}}}^{k=N+n_{\mathrm{cc}}-1} \big(p_{\mathrm{in}}\,[k]\;\; v\,[k] + p_{\mathrm{in}}\,[k+1]\big)\; h\,\delta\tau\,[k]\end{aligned} \tag{7.2.1.28}$$

7.2.1.3 Mixed Time-Energy-Change in Control

$$\begin{aligned}\min_{h,s[k],v[k],u_{\mathrm{trk}}[k],u_{\mathrm{brk}}[k]} & w_t h \\ & + w_e \sum_{k=0}^{k=n_{\mathrm{cc}}-1} p_{\mathrm{in}}\,[k]\; T_{\mathrm{S}} \\ & + w_e \sum_{k=n_{\mathrm{cc}}}^{k=N+n_{\mathrm{cc}}-1} \big(p_{\mathrm{in}}\,[k] + p_{\mathrm{in}}\,[k+1]\big)\; h\,\delta\tau\,[k] \\ & + w_{\Delta\mathrm{ctrl}} \sum_{k=0}^{k=N+n_{\mathrm{cc}}-1} \big|\pi_{\Delta\mathrm{ctrl}}\,[k]\big|\end{aligned} \tag{7.2.1.29}$$

where

$$\pi_{\Delta\text{ctrl}}[k] = \begin{cases} \frac{u[k+1]-u[k]}{T_\text{S}}, & k = 0, \ldots, n_\text{cc} - 1 \text{ for time-domain samples} \\ \frac{u[k+1]-u[k]}{h\delta\tau[k]}, & k = n_\text{cc}, \ldots, N + n_\text{cc} \text{ for } \tau\text{-domain samples} \end{cases} \tag{7.2.1.30}$$

The objective function now is to minimize a weighted sum of final arrival time, energy consumption, and change in control. Note that, $\sum_{j=1}^{j=k-1} w_{\Delta\text{ctrl}} \left|\pi_{\Delta\text{ctrl}}[j]\right|$, is a penalty term on the change in control that is added to smooth out the resulting optimal controls. This is done primarily for reasons of passenger comfort, to filter out solutions that consist of cycles of traction and braking.

7.3 Algorithm

The method of performing optimal control for the system with non-equal traction-braking delays is detailed in Algorithm 2, but first the problem setup and graphical overview of the algorithm, as shown in Fig. 7.1, is explained. Note that in this figure, it is assumed that the delay in the braking control input is larger than the delay in the traction control input, however, the problem formulation is still applicable if the delay in the traction control input is equal to or larger than the delay in the braking control input.

In this case, there are now two different control command buffers. One buffer is for storing the traction commands and the other is for storing braking commands. The buffers are now of different sizes. The length of the traction buffer is given by the delay, from issue to effect in traction, and the same is true for the braking buffer except the delay value used is that for the braking system. The sampling period of each element in the buffer is the rate at which the control commands are issued. The control command issue rate is assumed to be fixed and constant. This is denoted by T_{S} in Fig. 7.1 and Algorithm 1. The rate is assumed to be identical for traction and braking commands. The samples enter the buffer from one end and exit from the opposite end. In this figure, the samples enter from the right and exit from the left. The optimization window is from the end of the prediction window, $t_{\text{Predicted}}$, shown in Fig. 7.1, till the end of the trip, t_N.

Some more details regarding the optimal control procedure, Algorithm 2, are provided as follows.

The optimal control problem is solved when the train passes a pre-selected position marker, $s_{\text{Milepost}}[q]$, along the track. The position markers are chosen once at the beginning of the trip and are not modified again for the remainder of the trip. The position markers demarcate when the next optimization problem is solved; these are not the discretization nodes used in the optimization formulation.

Prediction is performed using the known applied control input (control commands present in the buffers) until $t_{\text{Predicted}}$, which is defined as: $t_{\text{Predicted}} = t + T_{\text{Min-Delay}}$, where $T_{\text{Min-Delay}} = \min\left(T_{\text{trk,delay}}, T_{\text{brk,delay}}\right)$. This is shown as the prediction window in Fig. 7.1. The prediction window has fixed time-steps that are determined by the control command input rate. Note, that for the train system at rest, the buffers can be assumed to be filled with all zeros.

An optimal control problem is solved from the end of prediction, $t_{\text{Predicted}}$, till the end of the trip, t_N. For the optimization problem, some important constraints will be highlighted as follows:

- The predicted states will be initial conditions on the system state for the optimization problem.

- For the control buffer with the longer delay, control samples from the end of prediction, $t_{\text{Predicted}}$, till the end of buffer, t_{qe}, are constraints on that specific control input. The time-steps will be fixed over this period, and the time-steps will be determined by the control sample input rate, T_{S} , as shown in Fig. 7.1.

Note, that the optimization problem will be solved in both the time-domain and the τ-domain. Part of the constraints will be in time-domain, and part of the constraints will be in τ-domain. This is shown in Fig. 7.1.

Some important notes regarding the optimization problem in Eq. (7.2.1) and the differences in the implementation of the non-equal traction-braking delay case and the equal traction-braking delay case are provided as follows.

The optimal control solution now returns at each discretization node: control (traction and braking commands), time state, position state, and velocity state. The position state is now a variable that has to be determined by the optimization solver. The position state is returned for each discretization node. The time state is only returned for nodes in the τ-region.

The number of nodes in the optimization problem is now equal to the number of the remaining position discretization nodes from the current train position till the destination, with the addition of a fixed number of nodes due to the constraints on the control due to existing values in the control buffer, as shown in Fig. 7.1. Once the train passes a position discretization node, the number of nodes in the optimization problem decreases by one. This point can also be rephrased as the following. The discretization nodes in the optimization formulation include: (i) a fixed number of nodes separated by constant T_{S} intervals corresponding to a constraint on either the traction or braking control depending on which has the longer delay, and (ii) nodes derived from converting τ from the continuous domain to the discrete domain. The τ discretization is chosen to be uniform. The number of τ discretization nodes is equal to the number of remaining position markers from the current train position until the destination. Once the train passes a position marker, the number of τ discretization nodes decreases by one. As the train progresses along the track, the set of τ discretization nodes changes because the number of nodes decreases; however, τ is always in the interval $\tau \in [0, 1]$.

Algorithm 2: Optimal control of train with non-equal traction-braking delays

```
begin
    Initialize T_S
    Initialize s_Milepost[k] for k ∈ {0, ..., L}
    T_Min-Delay ← min(T_trk,delay, T_brk,delay)
    t ← 0
    t_Sample Start ← 0
    S_Next ← 0
    Measure t, s, v
    while s < s_f do
        if s ≥ S_Next then
            Function Prediction(Guideway Data, t, s, v,
              {u_trk,buf[0], ..., u_trk,buf[j − 1]}, {u_brk,buf[0], ..., u_brk,buf[j − 1]}):
              is
                t_Start ← t
                t_Predicted ← t
                while t_Predicted - t_Start < T_min-Delay do
                    Integrate ODE using values from the traction and braking
                      control buffers
                    t_Predicted ← t_Predicted + dt
                return t_Predicted, s_Predicted, v_Predicted
            Function Optimization(Guideway Data, t_Predicted, s_Predicted,
              v_Predicted, {u_brk,buf[j], ..., u_brk,buf[j + n_cc − 1]},
              {τ[0], ..., τ[N]}, ): is
                Optimize
                return ( {t*[0], ..., t*[N + n_cc]}, {s*[0], ..., s*[N + n_cc]},
                  {v*[0], ..., v*[N + n_cc]}, {u*_trk[0], ..., u*_trk[N + n_cc]},
                  {u*_brk[0], ..., u*_brk[N + n_cc]} )
            S_Next ← Find Next Discretization Node Greater Than S_Next
        if t - t_Sample Start ≥ T_S then
            t_Sample Start ← t
            u_trk ← Interpolate( t + T_trk,delay, {t*[0], ..., t*[N + n_cc]},
              {u*_trk[0], ..., u*_trk[N + n_cc]})
            u_brk ← Interpolate( t + T_brk,delay, {t*[0], ..., t*[N + n_cc]},
              {u*_brk[0], ..., u*_brk[N + n_cc]})
```

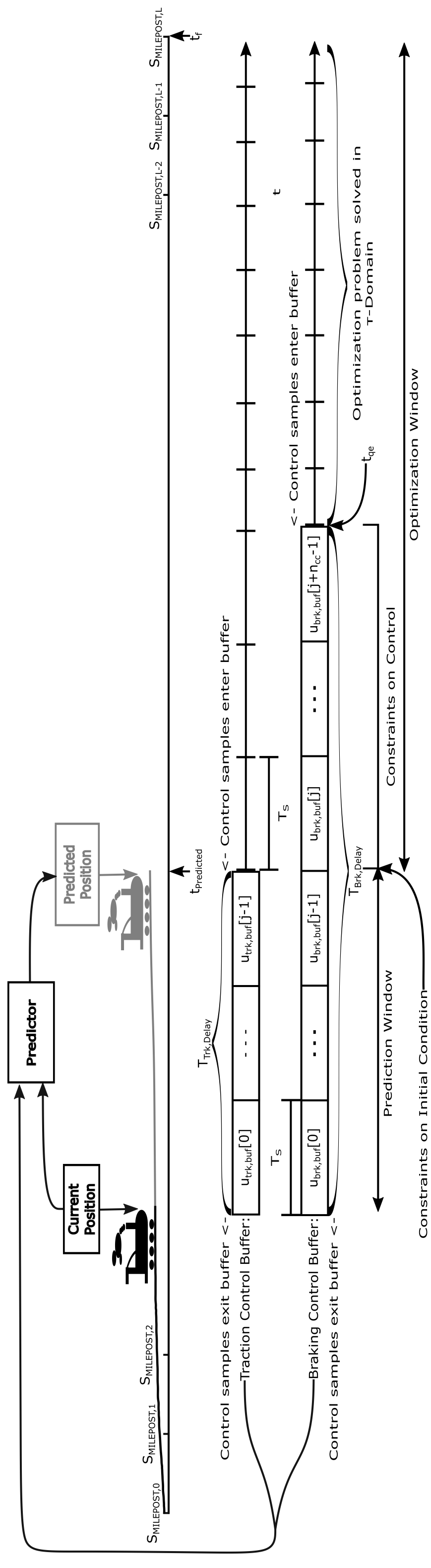

Figure 7.1: Optimal control of train system with traction and braking as separate inputs, each with its own unique actuation delay.The optimization problem for the section "constraints on control" is in time-domain while the optimization problem from the end of that segment till the end of the trip is in τ-domain.

Chapter 8

Computer Implementation and Simulation

8.1 Computer Implementation

The optimization problem in Eq. (5.2.1) is convex and can be solved for a global optimum with any solver capable of solving convex optimization problems. The optimization problem in Eq. (7.2.1) is nonconvex and nonlinear and can be solved for a local optimum using any nonlinear programming solver. The nonlinear programming solver may solve for a global optimum; however, this is not guaranteed. In general, solving a nonconvex optimization problem for a global optimum is extremely difficult and time-consuming [73].

The computer implementation results for both the convex and the nonconvex problem are produced by using Interior Point Optimizer (IPOPT) [90] interfaced to MATLAB using the OPTI Toolbox [91]. IPOPT implements the primal-dual interior-point algorithm [90].

The following information is supplied to the IPOPT software package:

- Objective function
  - Gradient of objective (first derivatives)
- Constraints
  - Jacobian of constraints (first derivatives)
  - Structure of Jacobian of constraints (sparsity pattern of the Jacobian)
- Hessian of the Lagrangian (second derivatives)
  - Structure of the Hessian of the Lagrangian (sparsity pattern of the Hessian)

The analytical expressions for each of the above were calculated manually and then implemented in MATLAB code as functions that are callable by the IPOPT solver.

The termination criteria used for IPOPT were the default ones [92]. A solver time limit of 100 seconds is imposed on IPOPT for all optimizations except the first one, as explained in the point on computation times below.

8.2 Results

The results shown are the following:

- The closed-loop trajectories for the states and controls.
- The computation times for each optimization for a single train run from station A to station B. As the train moves from station A to station B, the following

process is repeated: measuring the states of the train, solving an optimization problem to determine the future control commands, and application of a few of those control commands. For each of these optimization problems, the computation times are provided. Rephrased in another way, as the train runs from station A to station B, a sequence of optimization problems are solved at regular intervals based upon the latest sensor feedback to determine the control commands from the current train position till the end of the trip, and the computation times are provided for each of those optimization problems.

More details on the results provided are given below.

Closed-loop trajectories:

An optimization problem is solved at regular intervals, where the position at that interval is the origin and the destination remains fixed, known as a fixed horizon model predictive control strategy. A specific subset of the control commands returned from this optimization are fed into the simulator. The control commands are issued until the next optimization problem is triggered once the train passes the next position marker. This allows for feedback, because the measured time, position, and velocity are used as initial conditions for the optimization problem. This allows incorporation of the current states of the system in determining any future control commands.

Computation time:

Computation times are provided for each closed-loop optimization run. Note that for the nonconvex case, computation times can vary widely because of the nonconvexity. Also, note that for the initial optimization, the starting point is chosen at random, and for any subsequent optimization problems, warm-start is used, i.e. the solution from the previous step is used. The computation times are measured from

Table 8.1: Test System Specifications

OS Name:	Microsoft Windows 10
Processor:	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 2601 Mhz, 4 Core(s), 8 Logical Processor(s)
Installed Physical Memory (RAM):	16.0 GB

the internal computer clock. The time shown is actual time rather than the central processing unit (CPU) time. There are multiple processes running at the same time, and it is at the sole discretion of the operating system which process to schedule on the CPU at what time. The test system is not a real-time operating system (OS). Note that there is a maximum optimization time limit of 100 s for all optimizations other than the initial optimization. The initial optimization occurs before train departure and can be done offline, any subsequent optimizations are intended to be performed online.

8.2.1 Test System Specifications

The test system specifications are shown in Table 8.1.

8.2.2 Test Problem 1: Equal Traction-Braking Delays (Discrete-Position)

The convex optimization formulation presented in Eq. (5.2.1) is used here. The formulation parameters are:

- $N = 401$
- $t[0] = 0\,\mathrm{s}$
- $m = 1\,\mathrm{kg}$
- $s[0] = 0\,\mathrm{m}$

- $v[0] = 0\,\mathrm{m/s}$
- $t_{\mathrm{max}} = 1,000\,\mathrm{s}$
- $s[N] = 10,000\,\mathrm{m}$
- $v[N] = 0\,\mathrm{m/s}$
- $v_{\mathrm{max}}[k] = 100\,\mathrm{m/s}$ for $k \in \{0, \ldots, N\}$
- $u_{\mathrm{brk,max}} = -1.25\,\mathrm{m/s^2}$
- $u_{\mathrm{trk,max}} = 1.25\,\mathrm{m/s^2}$
- $r_0 = 1.327e0\,\mathrm{kg{\bullet}m/s^2}$
- $r_v = -3.663e-2\,\mathrm{kg/s}$
- $r_{v^2} = 2.528e-4\,\mathrm{kg/m}$
- $C_0 = -0.1\,\mathrm{kg{\bullet}m/s^2}$
- $C_v = -0.01\,\mathrm{kg/s}$
- $C_{v^2} = -0.001\,\mathrm{kg/m}$

The system in the convex case has equal traction-braking delays. While this case considers equal traction-braking delays, its results can also be used as reference for comparison with another case discussed later in which the traction delay is shorter than the braking delay; the nonconvex formulation will be used to solve that problem. Alternatively, addition of an artificial delay to the traction commands renders both delays equal, resulting in the case considered here with the convex formulation.

The weights in the objective function affect the trip journey since a relatively large value of w_e would put more weight on minimizing energy, and a relatively large value of w_ρ would place more weight on minimizing travel time. A sufficiently large value of w_ρ is necessary for the formulation to work as explained in Chapter 5.

8.2.2.1 Minimum-Time Optimal Control

The objective function, Eq. (7.2.1.29), parameters are: $w_\rho = 1,000$, and $w_e = 0$.

A minimum-time journey is characterized by extremal traction and braking, and minimum time spent coasting. For example, a minimum-time trip with no minimum arrival time is characterized by extremal, bang-bang, control: phases of maximum traction and braking; although, note that this may not be the only solution that results in a minimum-time trip.

8.2.2.1.1 Shorter Delays

- $T_{\text{trk,delay}} = 2.8\,\text{s}$
- $T_{\text{brk,delay}} = 2.8\,\text{s}$

The results are shown in Figs. 8.1 and 8.2.

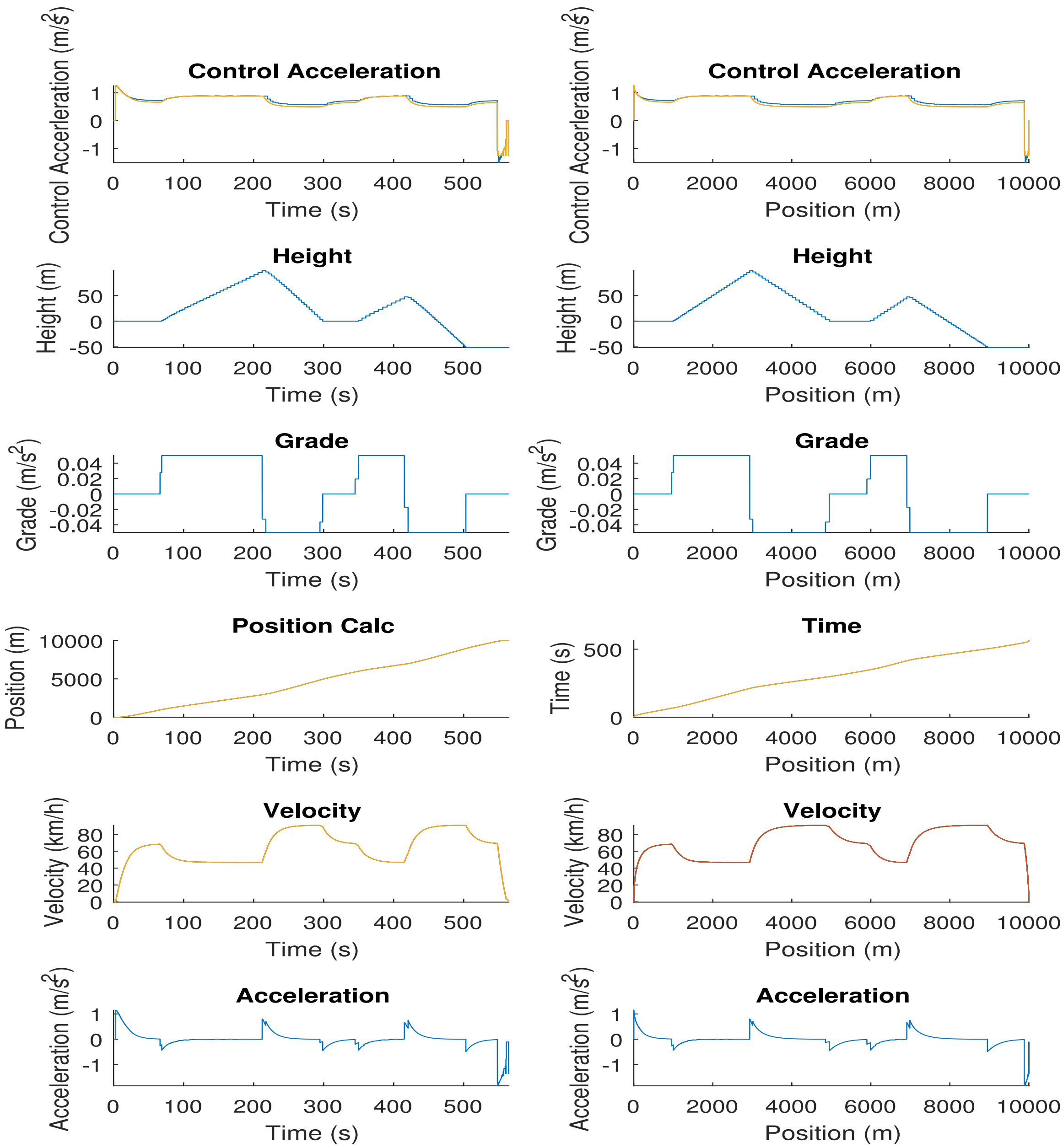

Figure 8.1: Convex formulation: minimum-time considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 2.8\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10,002.15 m. Overshoot: 2.15 m. Trip Time: 564.41 s. Trip Delay: N/A s. Energy Consumed: 6568.42 J.

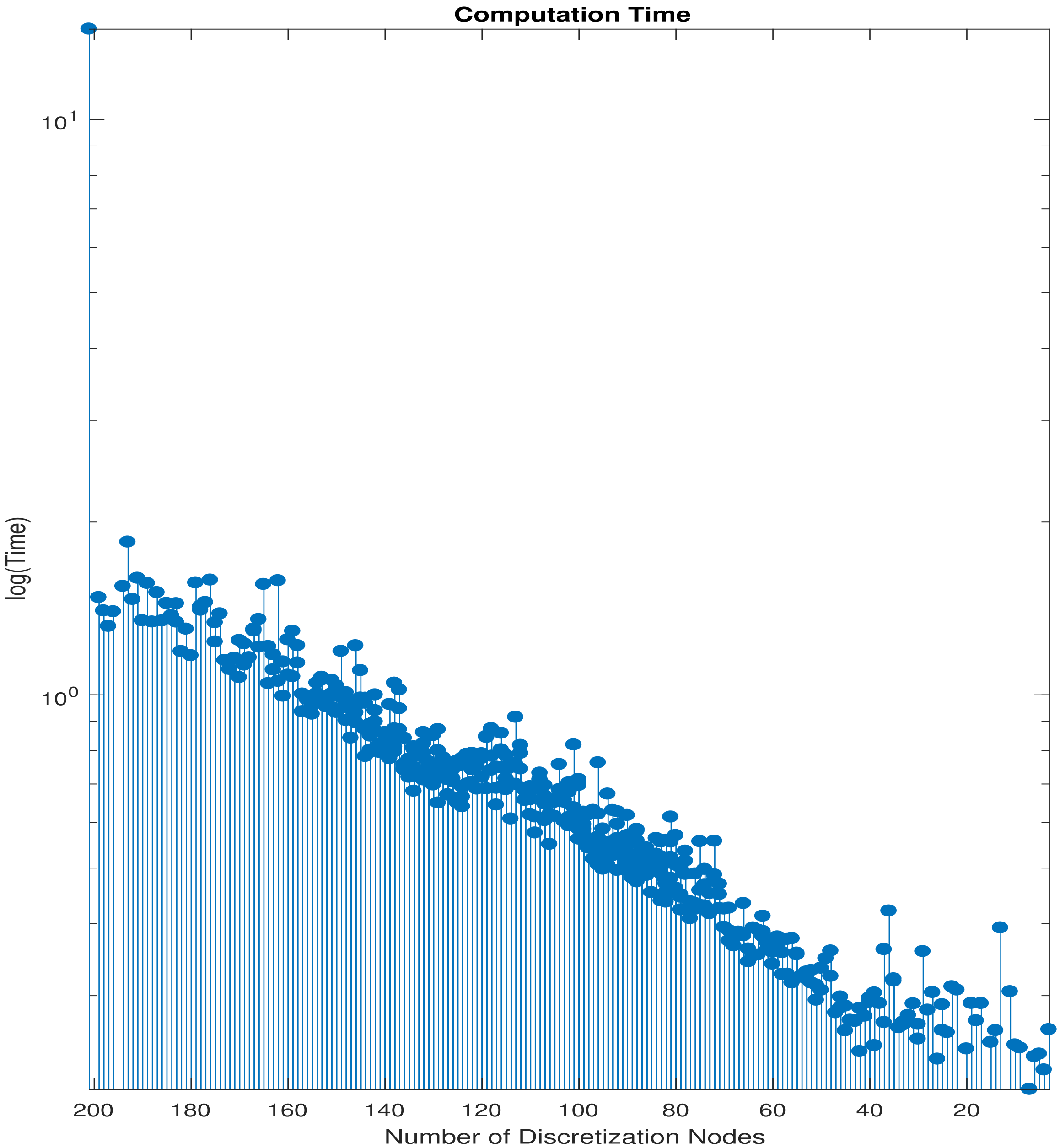

Figure 8.2: Convex formulation: minimum-time considering delays in the control input; computation times. $T_{\text{trk,delay}} = 2.8\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

8.2.2.1.2 Longer Delays

- $T_{\text{trk,delay}} = 7.0\,\text{s}$
- $T_{\text{brk,delay}} = 7.0\,\text{s}$

The results are shown in Figs. 8.3 and 8.4.

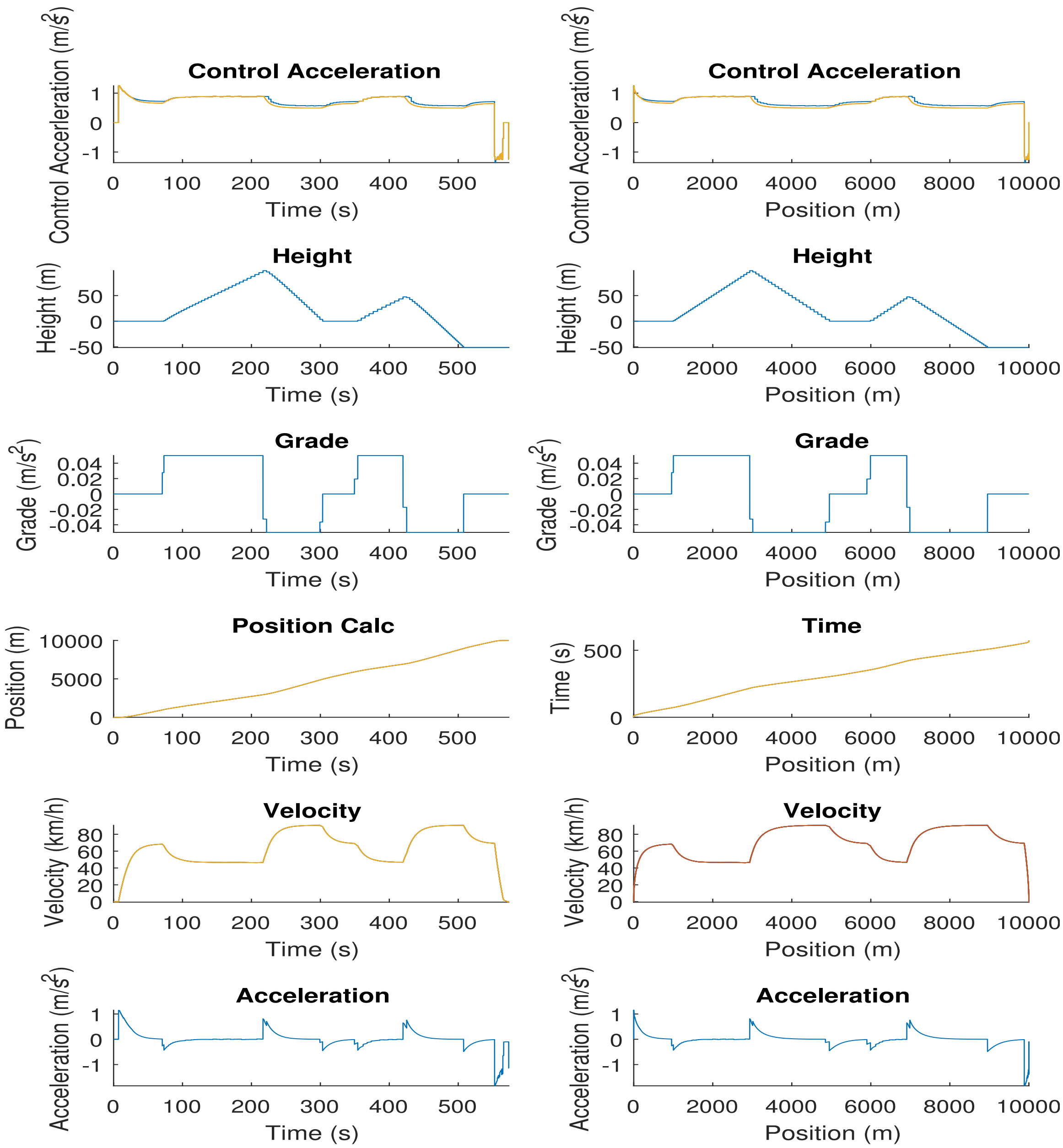

Figure 8.3: Convex formulation: minimum-time considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 7.0\,\text{s}$, $T_{\text{brk,delay}} = 7.0\,\text{s}$. Final Position: 10001.51 m. Overshoot: 1.51 m. Trip Time: 573.37 s. Trip Delay: N/A s. Energy Consumed: 6565.11 J.

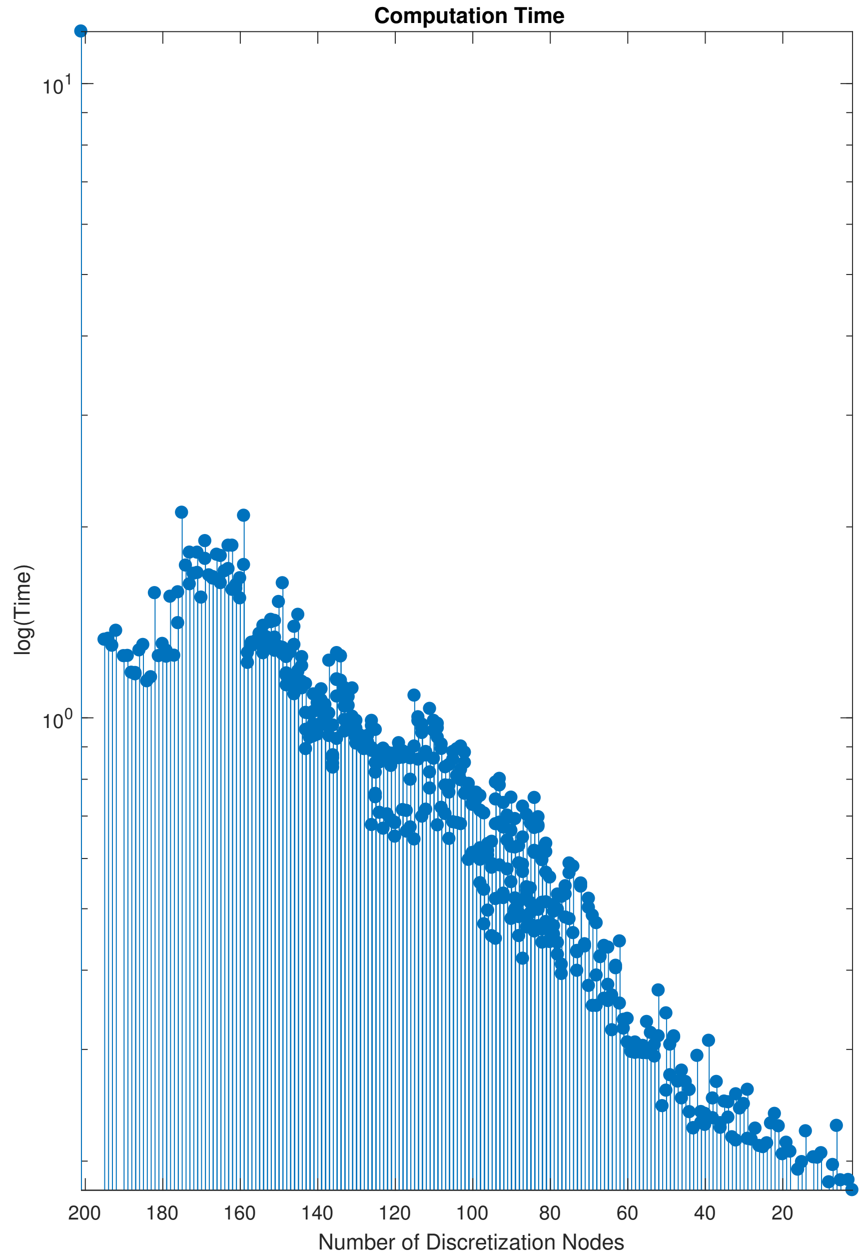

Figure 8.4: Convex formulation: minimum-time considering delays in the control input; computation times. $T_{\text{trk,delay}} = 7.0\,\text{s}$, $T_{\text{brk,delay}} = 7.0\,\text{s}$.

8.2.2.2 Minimum-Energy Optimal Control

A minimum-energy journey is characterized by more time spent coasting and less time spent braking (in the case of no regenerative braking). This is because braking results in energy loss. For example, a minimum-energy trip with no maximum arrival time is characterized by phases of traction and coasting. There is no braking because there is no time-limit and the kinetic energy of the system can be reduced to zero due to resistance losses alone.

8.2.2.2.1 Shorter Delays

- $T_{\text{trk,delay}} = 2.8\,\text{s}$
- $T_{\text{brk,delay}} = 2.8\,\text{s}$

The objective function, Eq. (7.2.1.29), parameters are: $w_\rho = 1,000$, and $w_e = 10,000$.

The results are shown in Figs. 8.5 and 8.6.

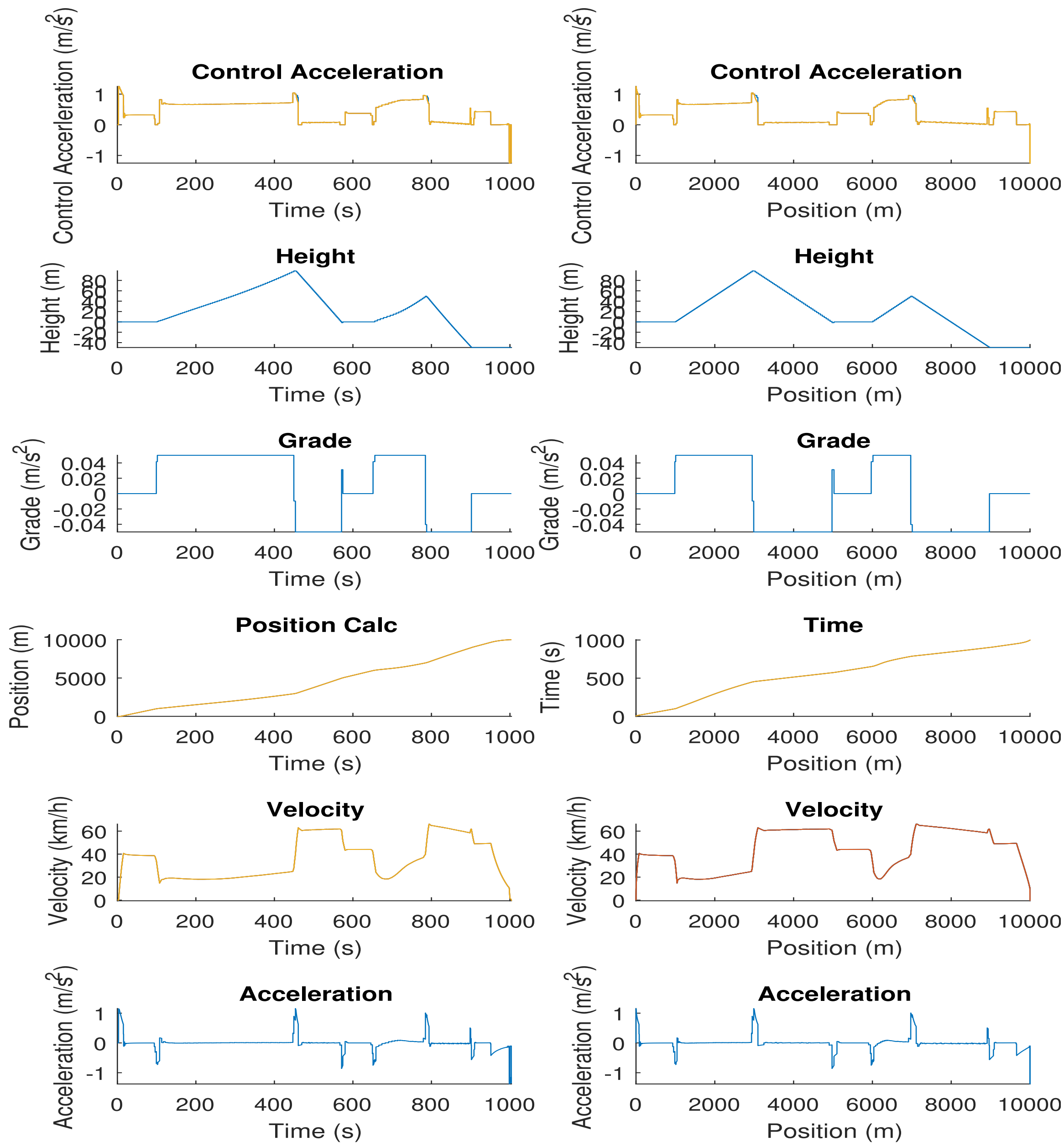

Figure 8.5: Convex formulation: minimum-energy considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 2.8\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10,000.73 m. Overshoot: 0.73 m. Trip Time: 1003.73 s. Trip Delay: 3.73 s. Energy Consumed: 3463.84 J.

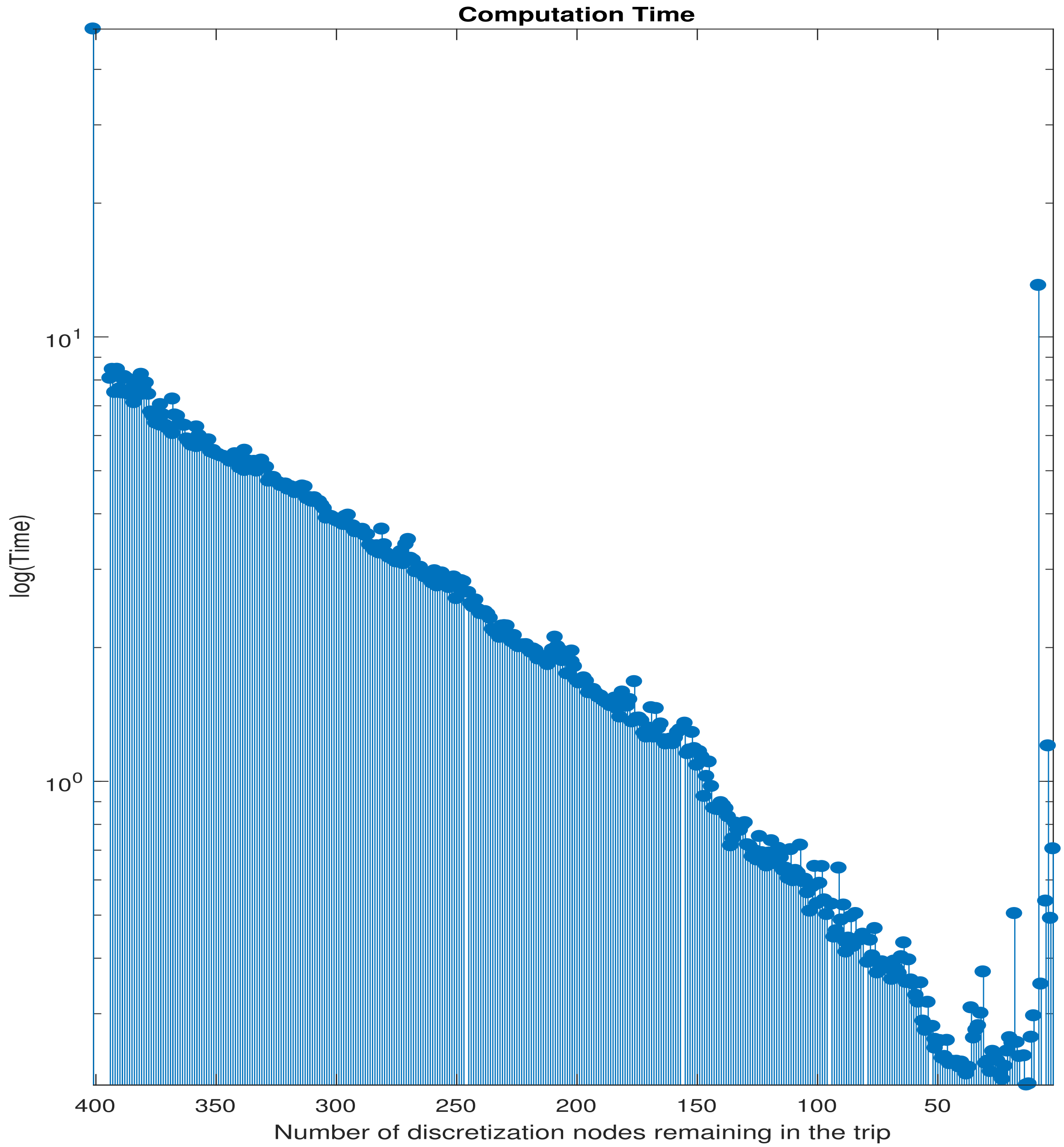

Figure 8.6: Convex formulation: minimum-energy considering delays in the control input; computation times. $T_{\text{trk,delay}} = 2.8\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

8.2.2.2.2 Longer Delays

- $T_{\text{trk,delay}} = 7.0\,\text{s}$
- $T_{\text{brk,delay}} = 7.0\,\text{s}$

The objective function, Eq. (7.2.1.29), parameters are: $w_\rho = 1,000$, and $w_e = 568.97$.

The results are shown in Figs. 8.7 and 8.8.

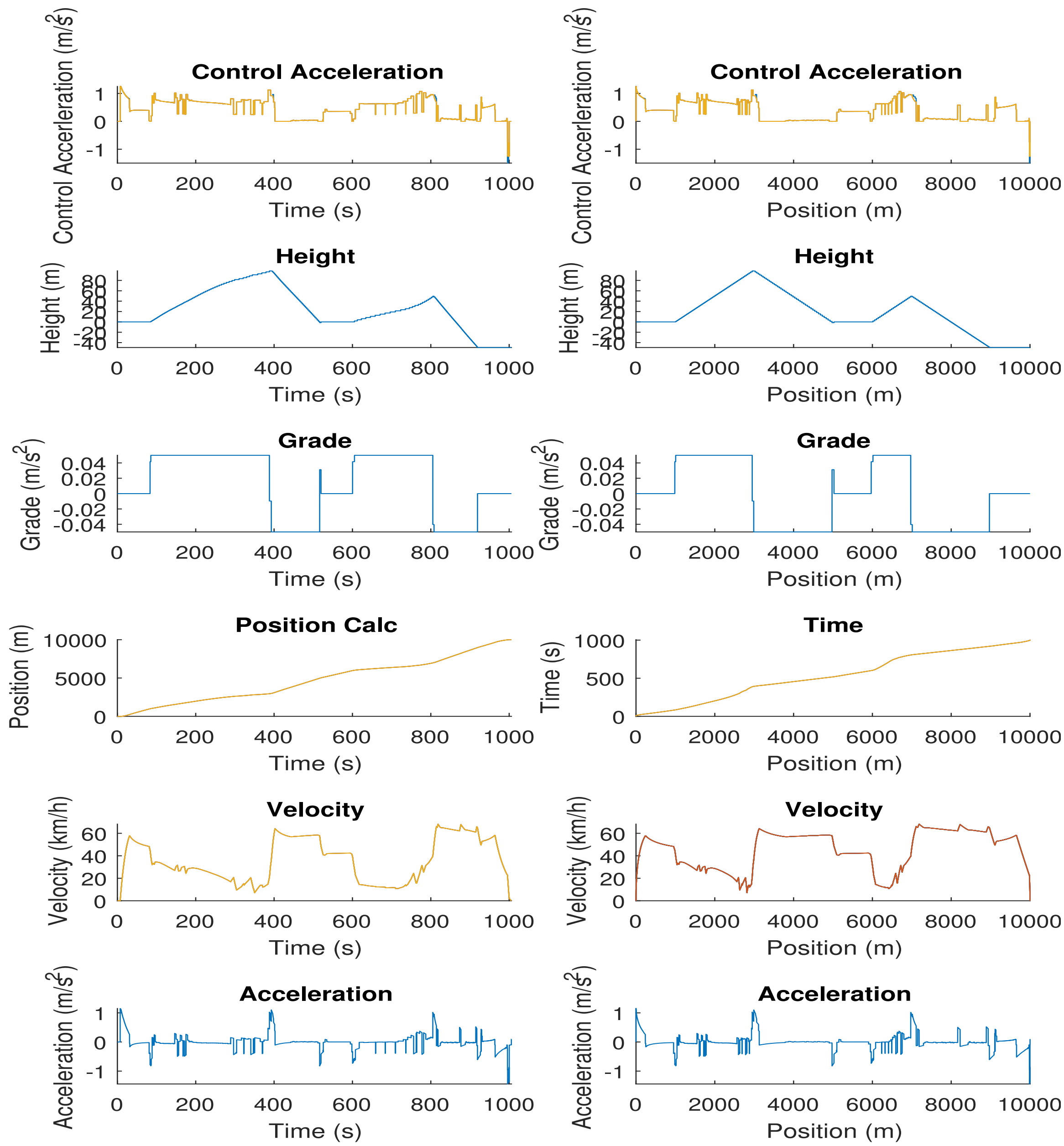

Figure 8.7: Convex formulation: minimum-energy considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 7.0\,\text{s}$, $T_{\text{brk,delay}} = 7.0\,\text{s}$. Final Position: 10001.15 m. Overshoot: 1.15 m. Trip Time: 1005.69 s. Trip Delay: 5.69 s. Energy Consumed: 3626.36 J.

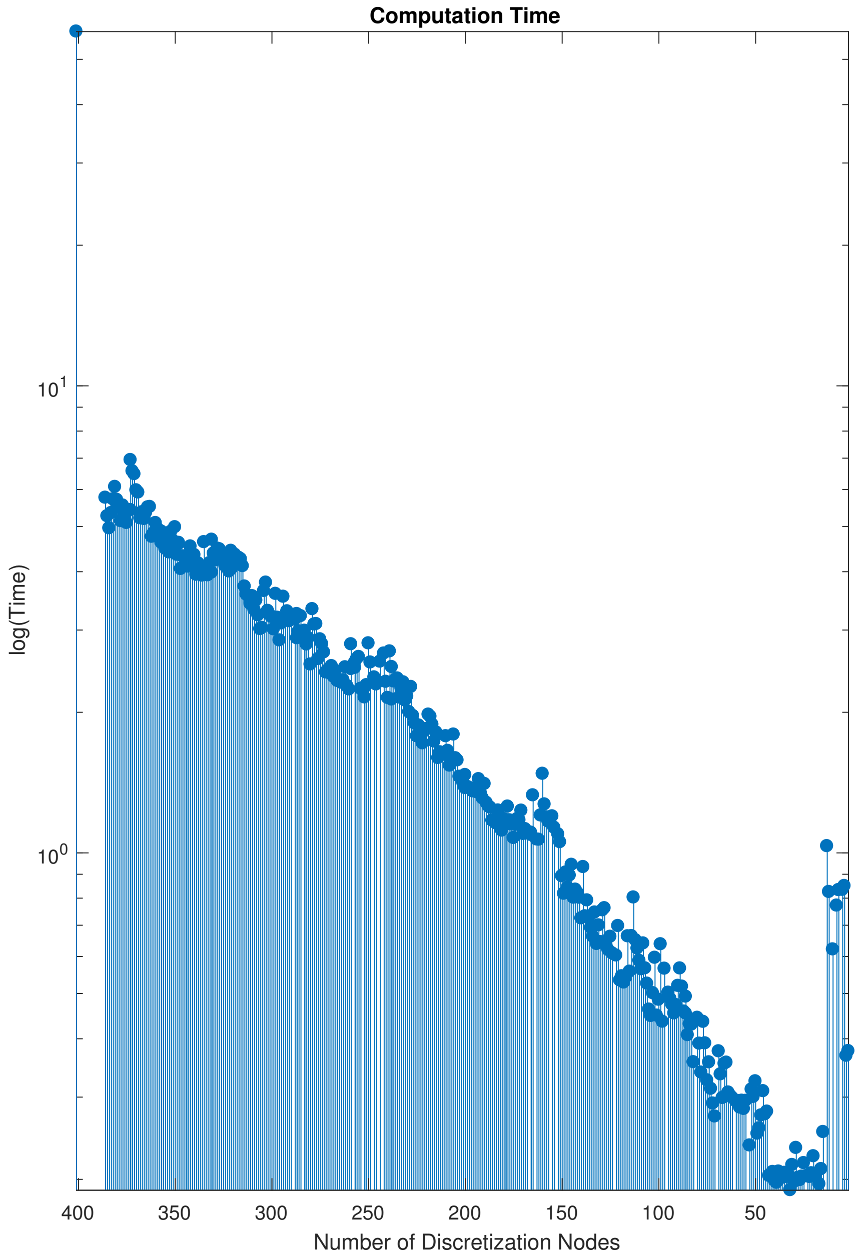

Figure 8.8: Convex formulation: minimum-energy considering delays in the control input; computation times. $T_{\text{trk,delay}} = 7.0\,\text{s}$, $T_{\text{brk,delay}} = 7.0\,\text{s}$.

8.2.3 Test Problem 2: Non-Equal Traction-Braking Delays (Discrete-Time)

The nonconvex optimization formulation presented in Eq. (7.2.1) is used to solve the optimal control problem for the case of non-equal traction-braking delays. The values of the parameters are:

- $N = 201$
- $T_{\mathrm{S}} = 0.07\,\mathrm{s}$
- $m = 1\,\mathrm{kg}$
- $t\,[0] = 0\,\mathrm{s}$
- $s\,[0] = 0\,\mathrm{m}$
- $v\,[0] = 0\,\mathrm{m/s}$
- $t_{\mathrm{max}} = 1,000\,\mathrm{s}$
- $s\,[N] = 10,000\,\mathrm{m}$
- $v\,[N] = 0\,\mathrm{m/s}$
- $v_{\mathrm{max}}\,[k] = 100\,\mathrm{m/s}$ for each $s_{\mathrm{Milepost}}\,[k]$ for $k \in \{0, \ldots, L\}$
- $u_{\mathrm{brk,max}} = -1.25\,\mathrm{m/s^2}$
- $U_{\mathrm{trk,max}} = 1.25\,\mathrm{kg{\bullet}m/s^2}$
- $P = 12.5\,\mathrm{kg{\bullet}m^2/s^3}$
- $k_P = 12.5\,\mathrm{m/s}$
- $\mu_{\mathrm{trk,min}} = -0.2\,\mathrm{kg{\bullet}m/s^3}$
- $\mu_{\mathrm{trk,max}} = 0.2\,\mathrm{kg{\bullet}m/s^3}$
- $\mu_{\mathrm{brk,min}} = -0.2\,\mathrm{kg{\bullet}m/s^3}$
- $\mu_{\mathrm{brk,min}} = 0.2\,\mathrm{kg{\bullet}m/s^3}$
- $C_0 = -0.1\,\mathrm{kg{\bullet}m/s^2}$
- $C_v = -0.01\,\mathrm{kg/s}$
- $C_{v^2} = -0.001\,\mathrm{kg/m}$

The maximum traction force is given by Eq. (3.3.1). The parameters used in that equation are given in the list above.

The following specific difference equations, Eqs. (8.2.1.1) to (8.2.1.4), are used in place of the general difference equations in Eqs. (7.2.1.13), (7.2.1.14), (7.2.1.20) and (7.2.1.21), respectively. The equations, Eqs. (8.2.1.1) and (8.2.1.2), are arrived at by assuming a zero-order hold on the control input in the time-domain region, and the equations, Eqs. (8.2.1.3) and (8.2.1.4), are arrived at by assuming a first-order hold on the input in the τ-domain region.

for $j \in \{1, \ldots, n_{\text{cc}}\}$:

$$\begin{aligned} s[j+1] = s[j] + \frac{1}{2}\big(v[j] \\ + v[j+1]\big) T_{\text{S}} \end{aligned} \tag{8.2.1.1}$$

$$\begin{aligned} v[j+1] = v[j] + \frac{1}{2}\Bigg[-\frac{C(v[j])}{m} \\ + g[j] + u[j] \\ - \frac{C(v[j+1])}{m} \\ + g[j+1] + u[j]\Bigg] T_{\text{S}} \end{aligned} \tag{8.2.1.2}$$

for $j \in \{n_{\text{cc}}, \ldots, N + n_{\text{cc}} - 1\}$:

$$\begin{aligned} s[j+1] = s[j] + \frac{1}{2}\big(v[j] \\ + v[j+1]\big) h\delta\tau[j] \end{aligned} \tag{8.2.1.3}$$

$$\begin{aligned} v[j+1] = v[j] + \frac{1}{2}\Bigg[-\frac{C(v[j])}{m} \\ + g[j] + u[j] \\ - \frac{C(v[j+1])}{m} \\ + g[j+1] + u[j+1]\Bigg] h\delta\tau[j] \end{aligned} \tag{8.2.1.4}$$

The resistance force equation is:

$$C\left(v\left[j\right]\right) = 0.1 + 0.01v\left[j\right] + 0.001v^2\left[j\right] \tag{8.2.1.5}$$

8.2.3.1 Minimum-Time Optimal Control

The objective function, Eq. (7.2.1.29), parameters are: $w_t = 1$, $w_e = 0$, and $w_{\Delta\text{ctrl}} = 0$.

8.2.3.1.1 Shorter Delays

- $T_{\text{trk,delay}} = 1.4\,\text{s}$
- $T_{\text{brk,delay}} = 2.8\,\text{s}$
- $n_{\text{cc}} = 40$ (dependent on length of longest delay)

The results are shown in Figs. 8.9 to 8.12. As can be seen in the figure captions, ignoring delays results in an overshoot of the destination, the train passes the station by 25.57 m, while considering delays results in an overshoot of -0.27 m. The trip time while ignoring delays is 567.63 s, while for considering delays is 568.54 s, a difference of around 0.16 %. The energy usage for ignoring delays is 6360.59 J, while for considering delays is 6354.93 J, a difference of around 0.09 %.

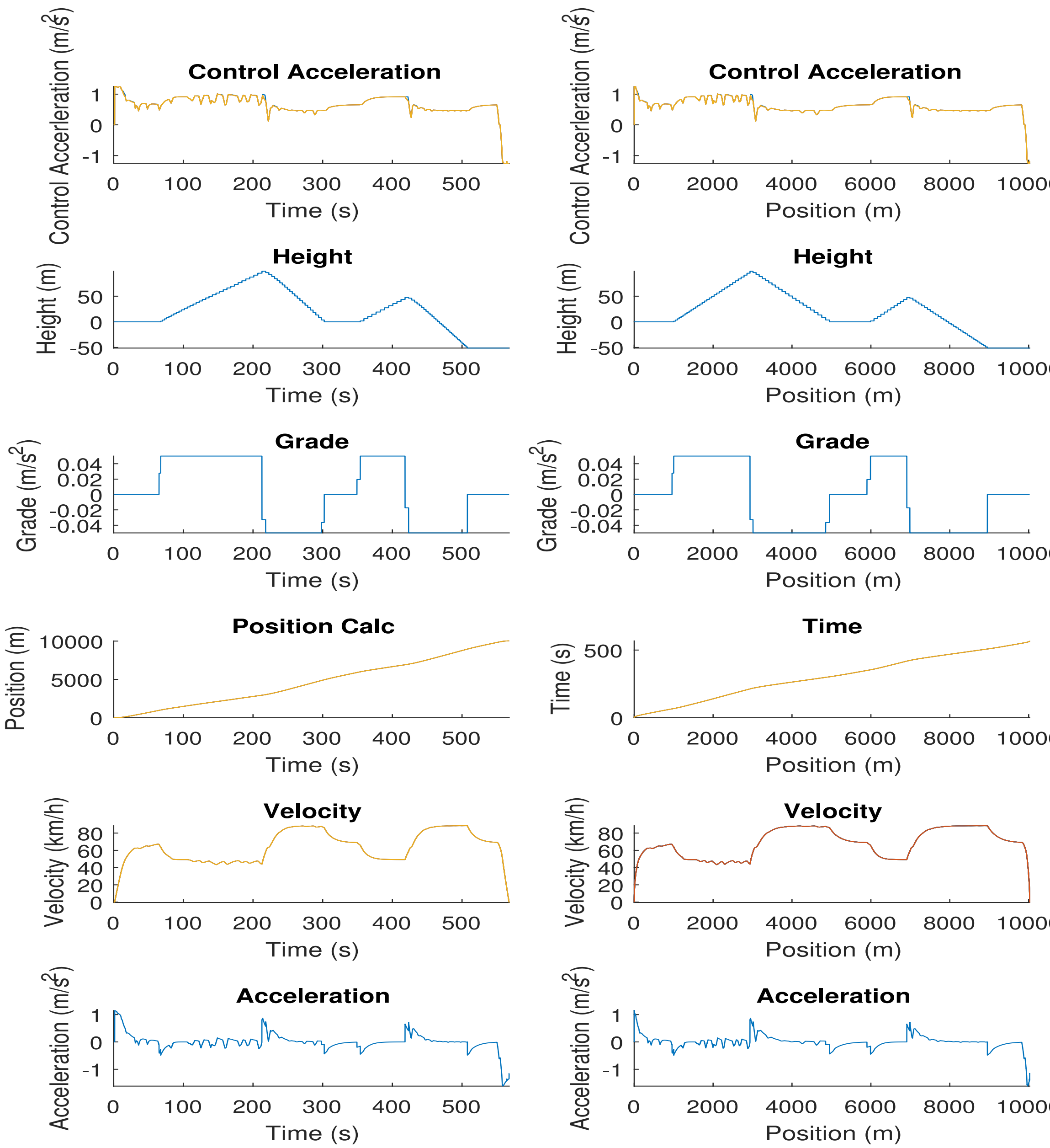

Figure 8.9: Minimum-time ignoring delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10025.57 m. Overshoot: 25.57 m. Trip Time: 567.63 s. Trip Delay: N/A s. Energy Consumed: 6360.59 J.

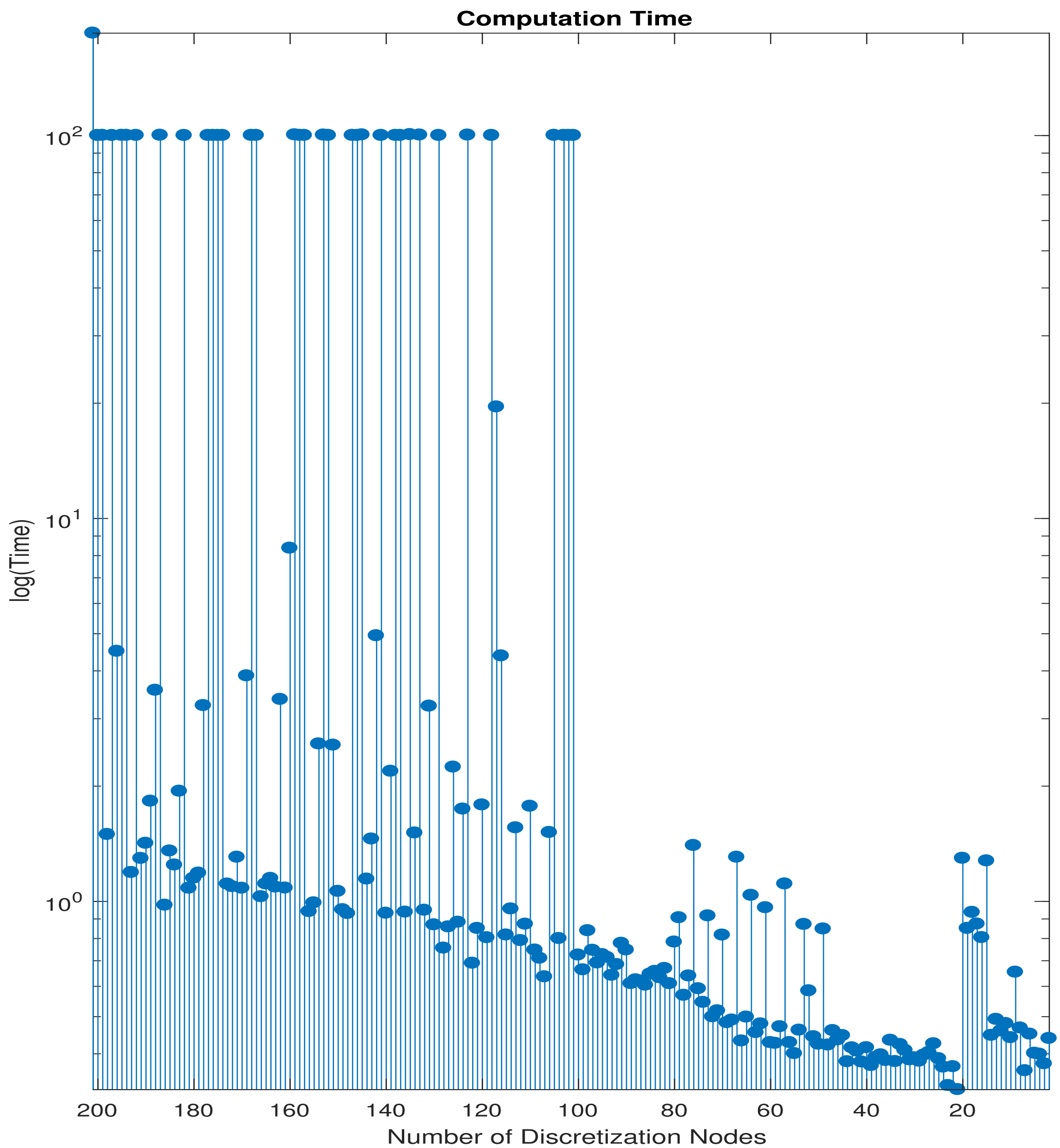

Figure 8.10: Minimum-time ignoring delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

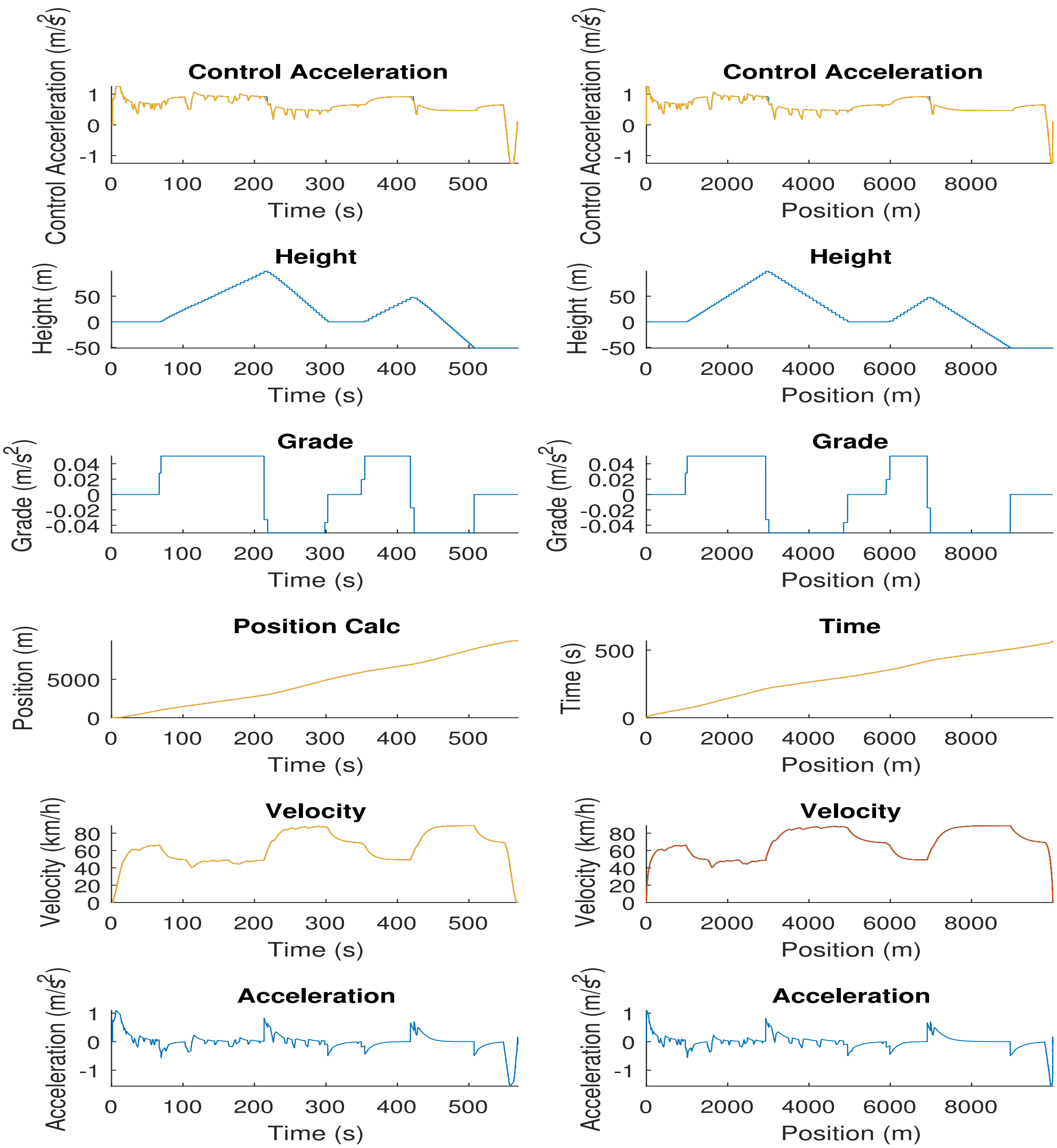

Figure 8.11: Minimum-time considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 9999.73 m. Overshoot: -0.27 m. Trip Time: 568.54 s. Trip Delay: N/A s. Energy Consumed: 6354.93 J.

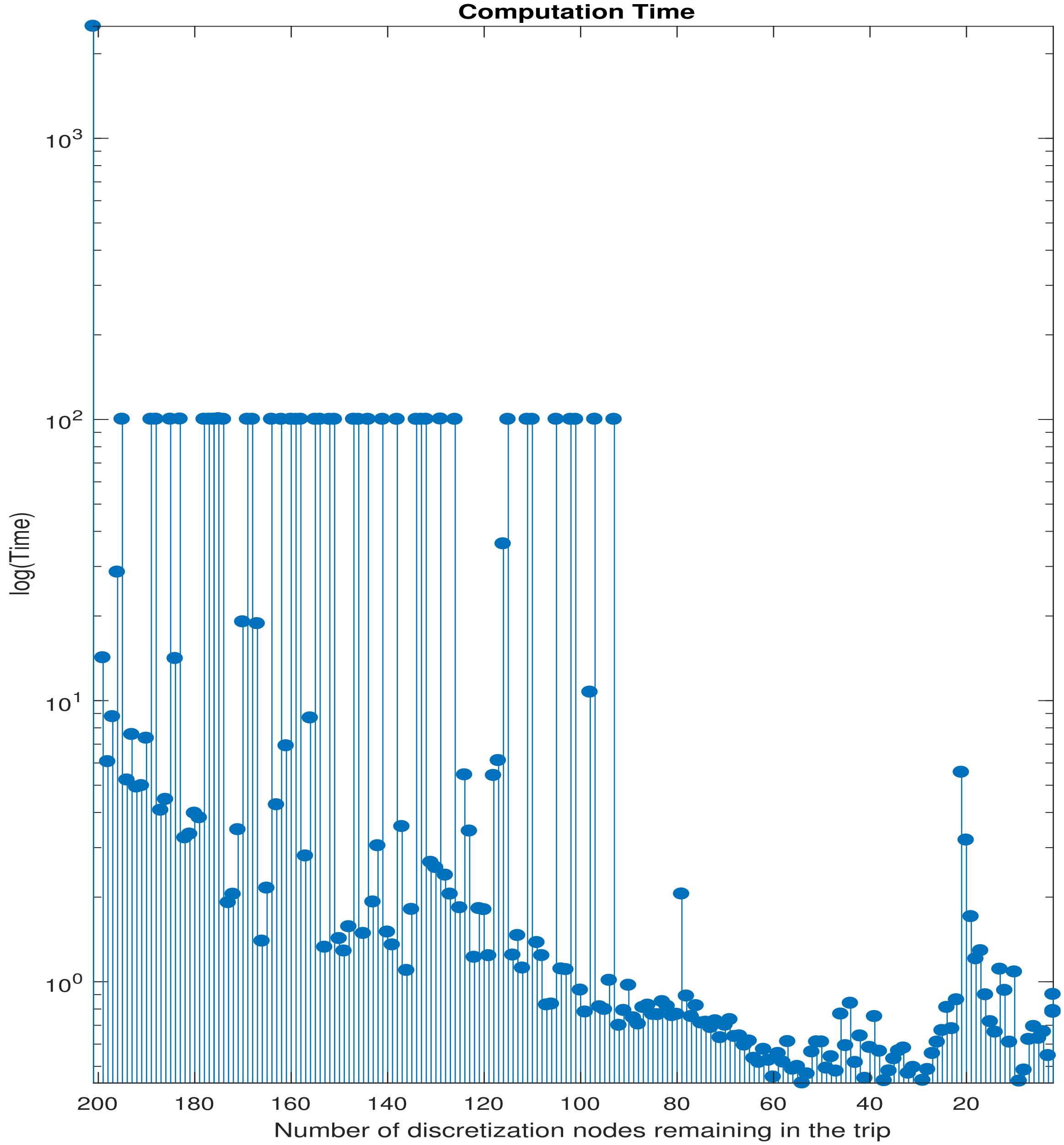

Figure 8.12: Minimum-time considering delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

8.2.3.1.2 Longer Delays

- $T_{\text{trk,delay}} = 3.5\,\text{s}$
- $T_{\text{brk,delay}} = 7.0\,\text{s}$
- $n_{\text{cc}} = 100$ (dependent on length of longest delay)

The results are shown in Figs. 8.13 to 8.16. As can be seen in the figure captions, ignoring delays results in an overshoot of the destination, the train passes the station by 16.44 m, while considering delays results in an overshoot of -0.18 m. The trip time while ignoring delays is 578.55 s, while for considering delays is 598.43 s, a difference of around 3.44 %. The energy usage for ignoring delays is 6205.31 J, while for considering delays is 6130.40 J, a difference of around 1.22 %.

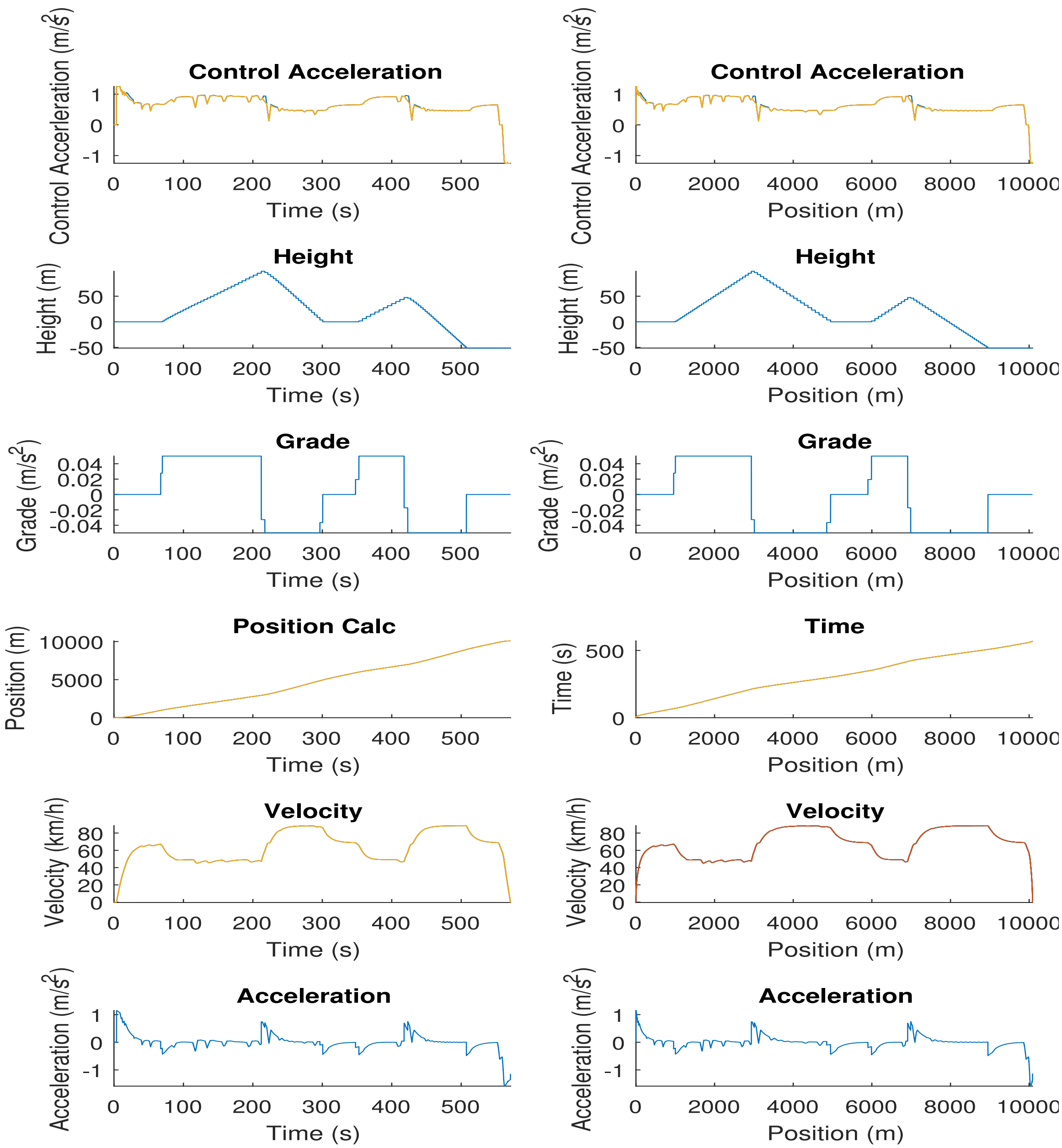

Figure 8.13: Minimum-time ignoring delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 3.5\,\text{s}$, $T_{\text{brk,delay}} = 7.0\,\text{s}$. Final Position: 10016.44 m. Overshoot: 16.44 m. Trip Time: 578.55 s. Trip Delay: N/A s. Energy Consumed: 6205.31 J.

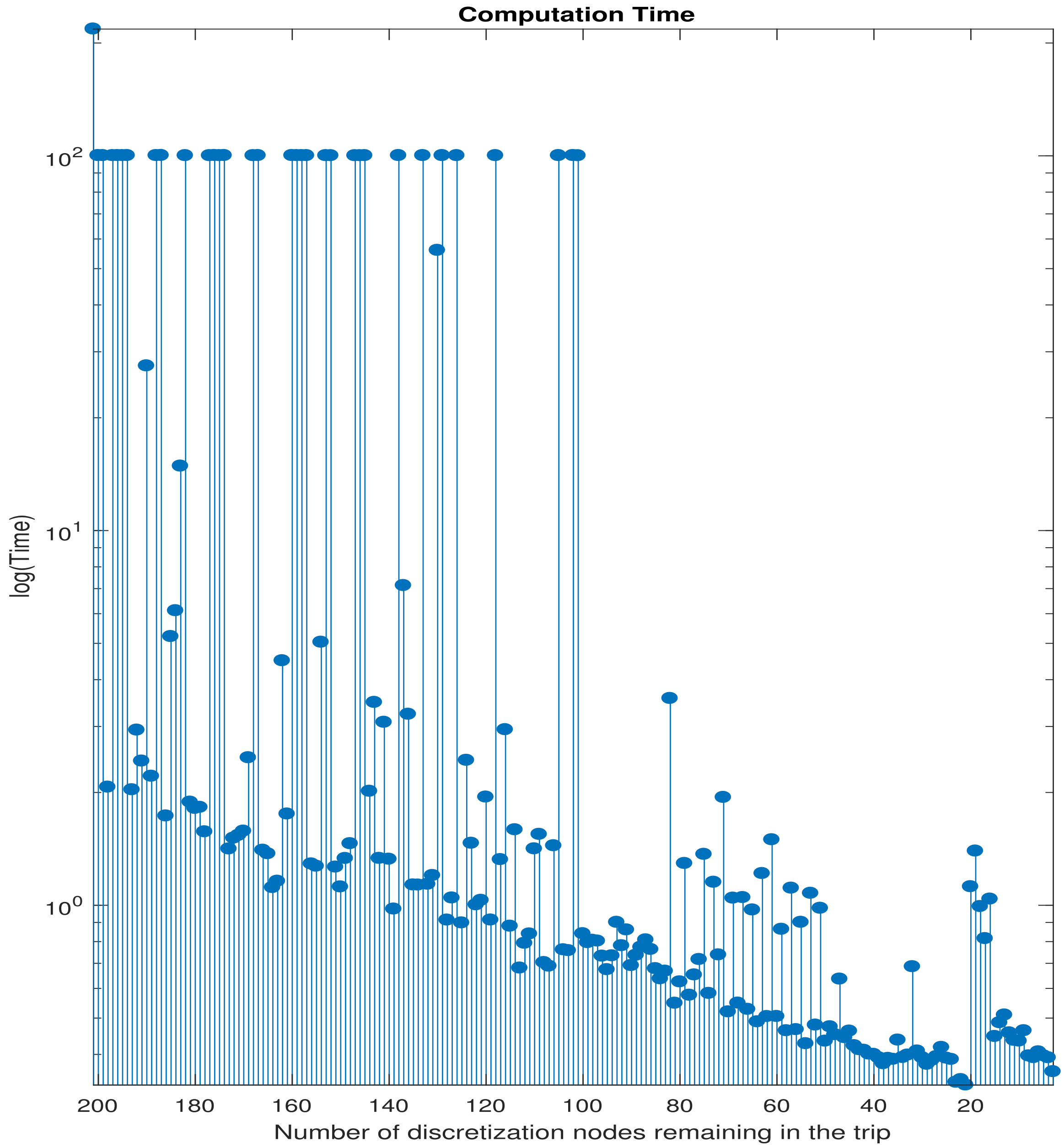

Figure 8.14: Minimum-time ignoring delays in the control input; computation times. $T_{\text{trk,delay}} = 3.5\,\text{s}$, $T_{\text{brk,delay}} = 7.0\,\text{s}$.

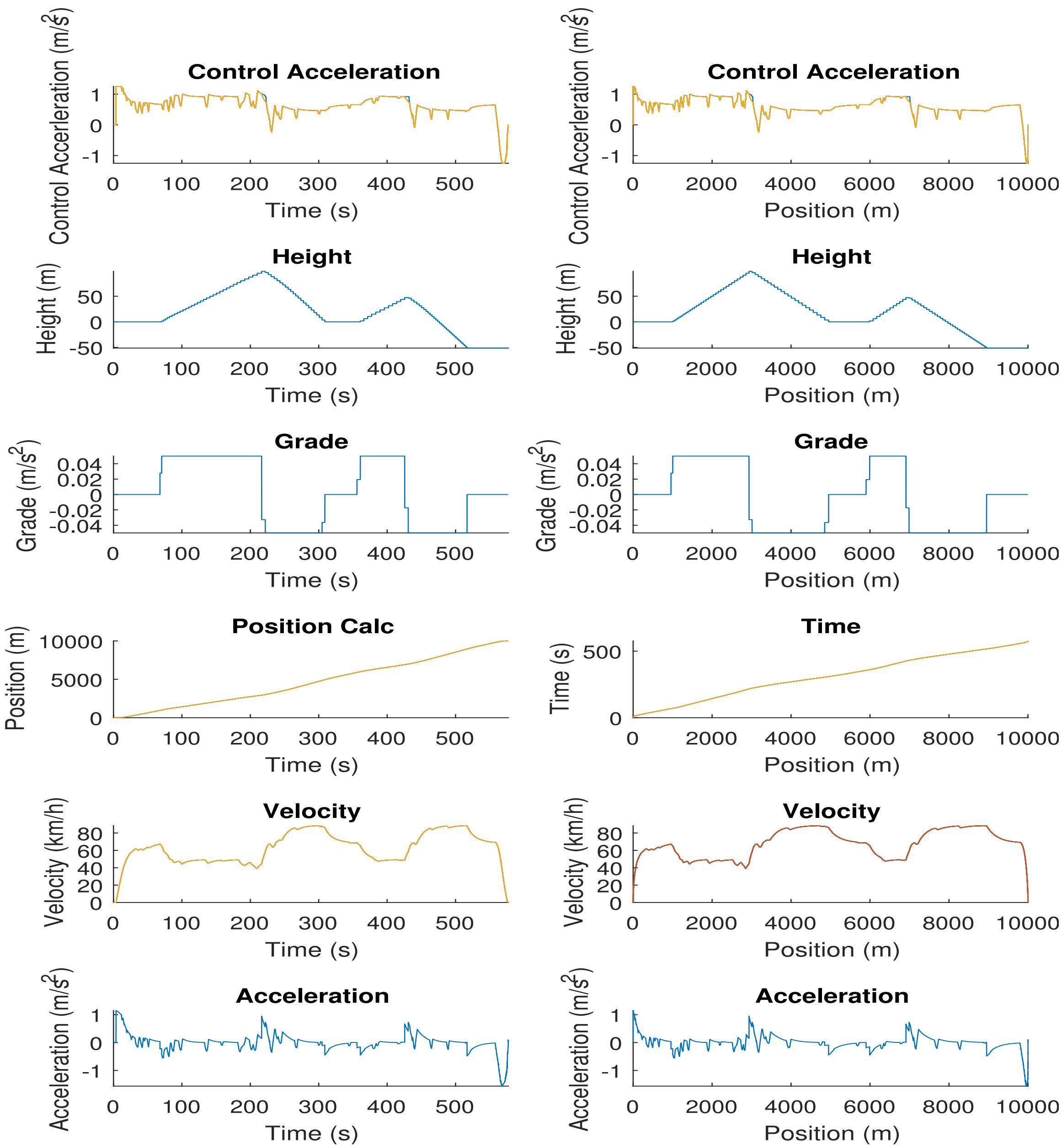

Figure 8.15: Minimum-time considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 3.5\,\text{s}$, $T_{\text{brk,delay}} = 7.0\,\text{s}$. Final Position: 9999.82 m. Overshoot: -0.18 m. Trip Time: 598.43 s. Trip Delay: N/A s. Energy Consumed: 6130.40 J.

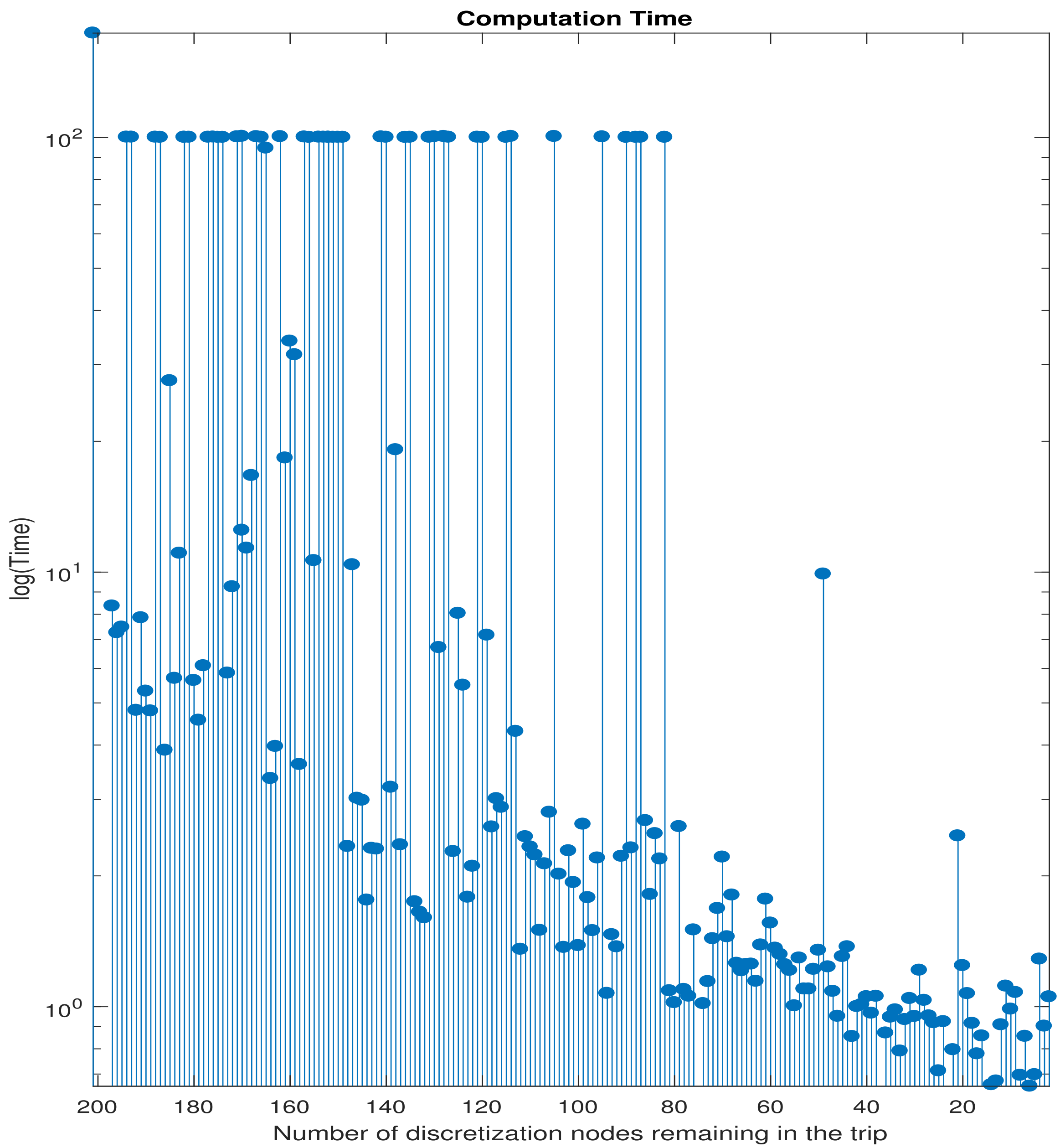

Figure 8.16: Minimum-time considering delays in the control input; computation times. $T_{\text{trk,delay}} = 3.5\,\text{s}$, $T_{\text{brk,delay}} = 7.0\,\text{s}$.

8.2.3.2 Minimum-Energy Optimal Control

The objective function, Eq. (7.2.1.29), parameters are: $w_t = 0$, $w_e = 1$, and $w_{\Delta\text{ctrl}} = 1$.

8.2.3.2.1 Shorter Delays

- $T_{\text{trk,delay}} = 1.4\,\text{s}$
- $T_{\text{brk,delay}} = 2.8\,\text{s}$
- $n_{\text{cc}} = 40$ (dependent on length of longest delay)

The results are shown in Figs. 8.17 to 8.20. As can be seen in the figure captions, ignoring delays results in an overshoot of the destination, the train passes the station by 5.89 m, while considering delays results in an overshoot of 0.07 m. The energy usage for ignoring delays is 3423.78 J, while for considering delays is 3443.09 J, a difference of around 0.56 %. The trip time while ignoring delays is 997.36 s, while for considering delays is 1002.05 s. The train is late by 2.05 s when considering delays. Minimizing energy would entail driving at lower speeds, thus increasing the travel time. This is because losses in energy occur due to resistance and braking. Resistance losses increase as the speed increases, thus increasing energy consumption.

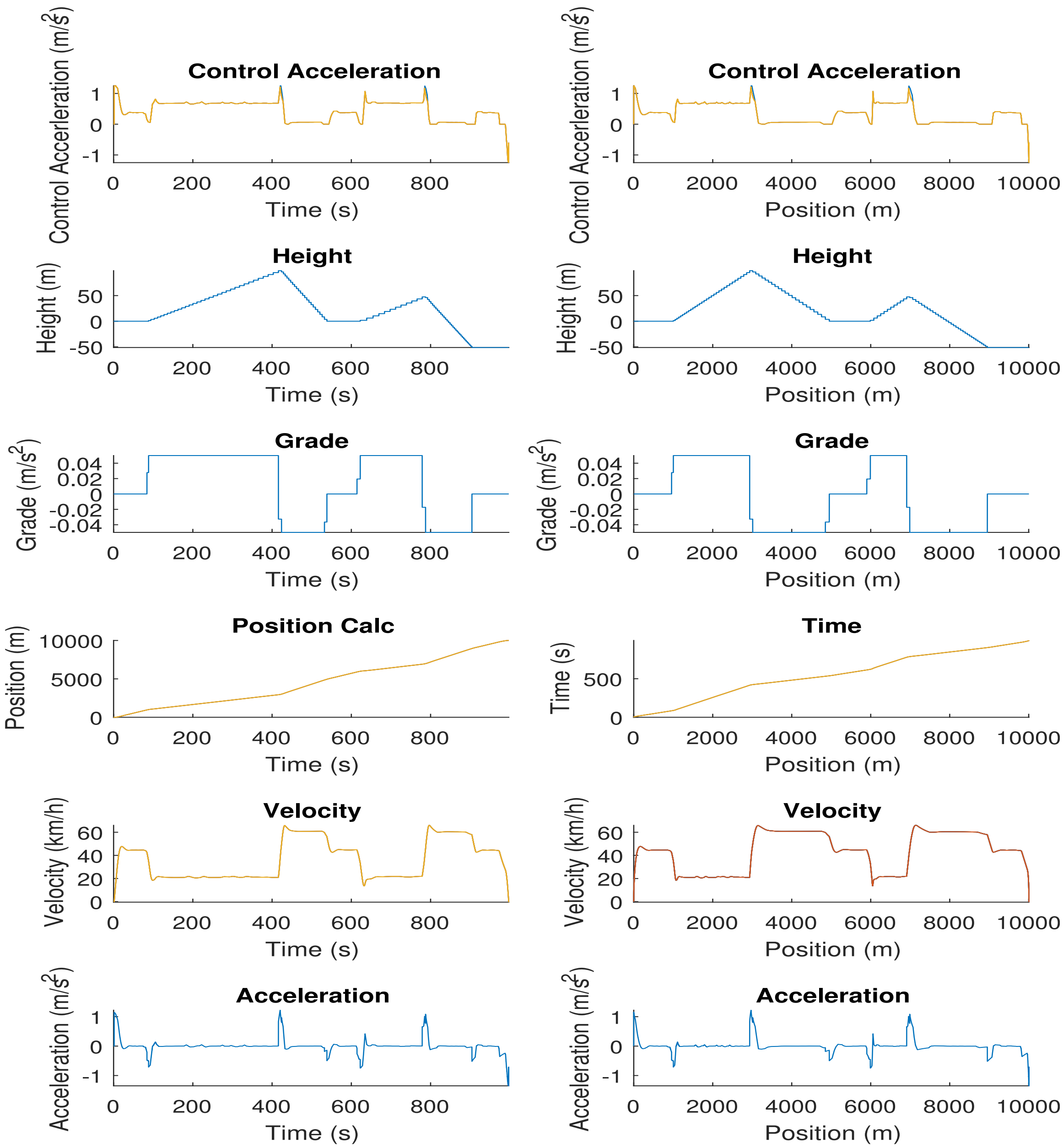

Figure 8.17: Minimum-energy ignoring delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10,005.89 m. Overshoot: 5.89 m. Trip Time: 997.36 s. Trip Delay: N/A s. Energy Consumed: 3423.78 J.

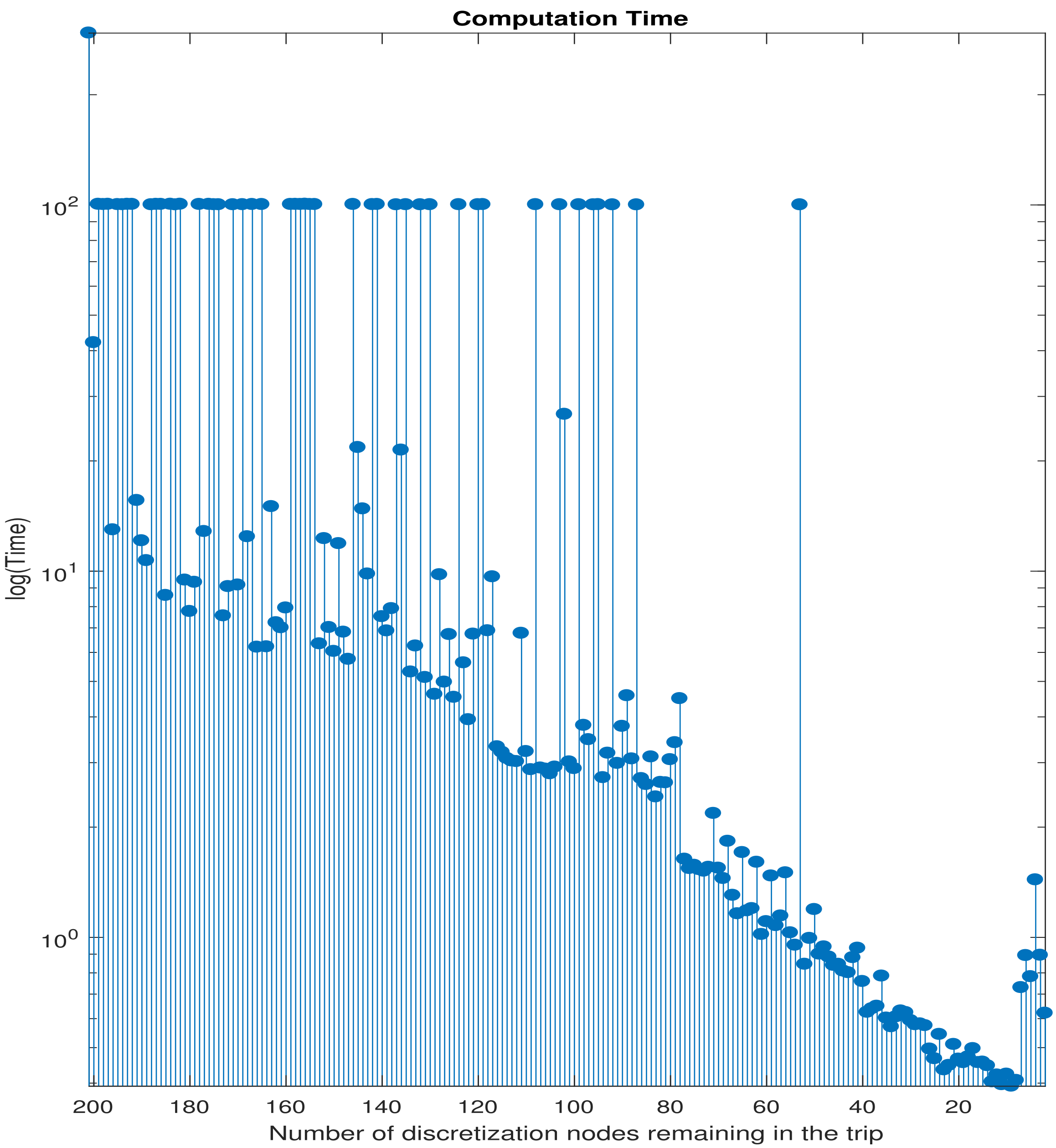

Figure 8.18: Minimum-energy ignoring delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

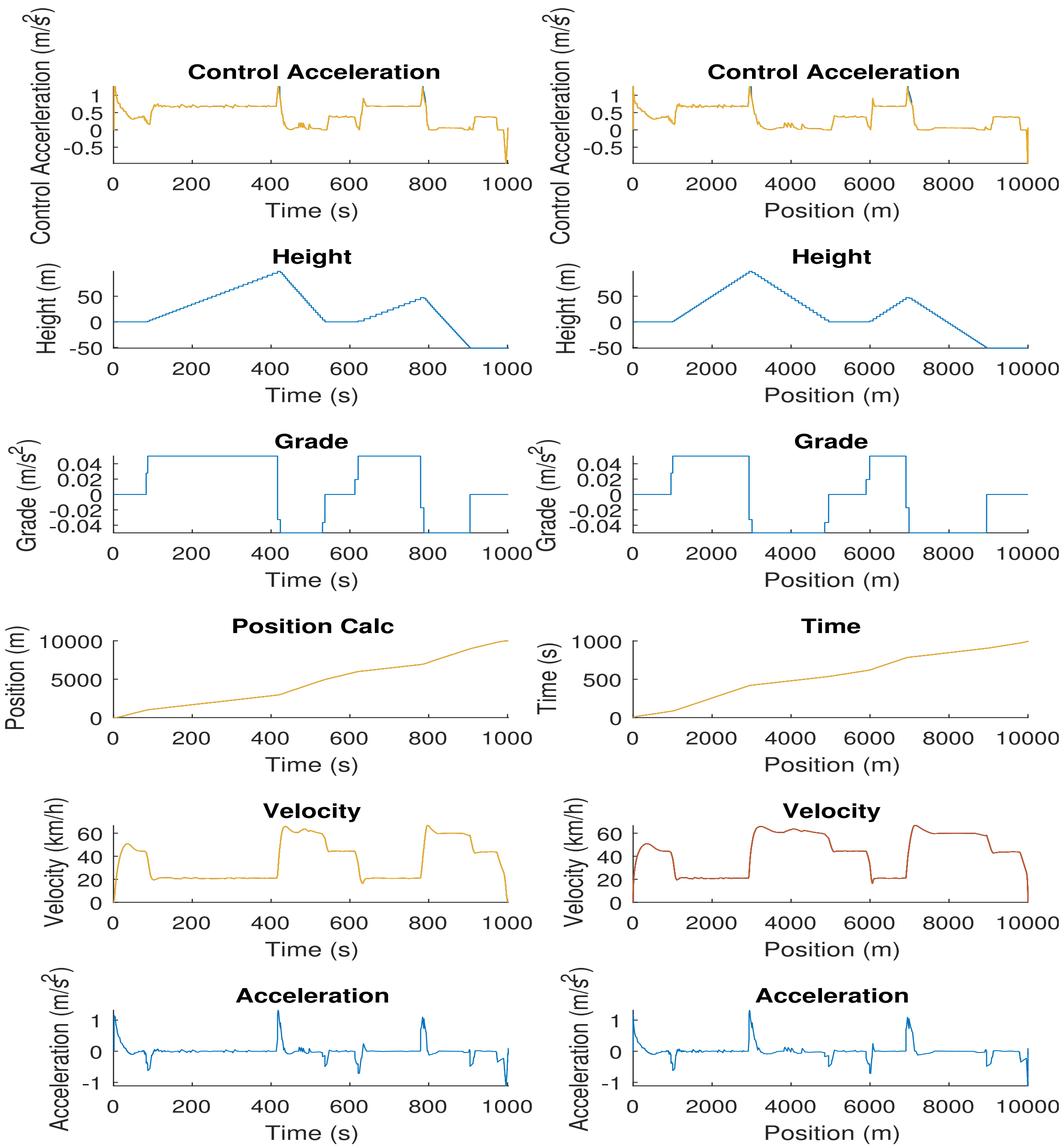

Figure 8.19: Minimum-energy considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10,000.07 m. Overshoot: 0.07 m. Trip Time: 1002.05 s. Trip Delay: 2.05 s. Energy Consumed: 3443.09 J.

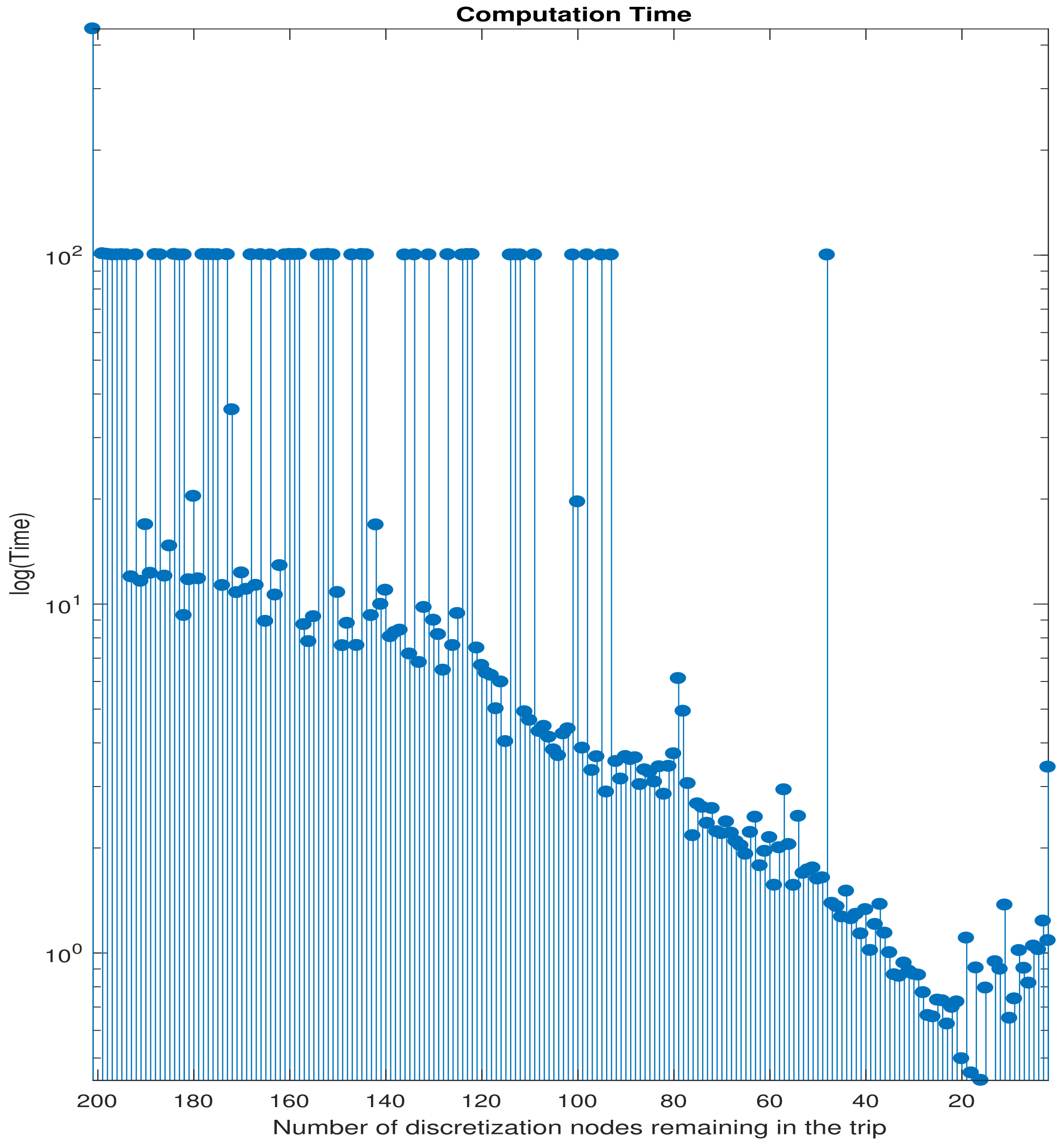

Figure 8.20: Minimum-energy considering delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

8.2.3.2.2 Longer Delays

- $T_{\text{trk,delay}} = 3.5\,\text{s}$
- $T_{\text{brk,delay}} = 7.0\,\text{s}$
- $n_{\text{cc}} = 100$ (dependent on length of longest delay)

The results are shown in Figs. 8.21 to 8.24. As can be seen in the figure captions, ignoring delays results in an overshoot of the destination, the train passes the station by 2.22 m, while considering delays results in an overshoot of 0.34 m. The energy usage for ignoring delays is 3451.66 J, while for considering delays is 3550.94 J, a difference of around 2.88 %. The trip time while ignoring delays is 997.29 s, while for considering delays is 1003.24 s. The train is late by 3.24 s when considering delays.

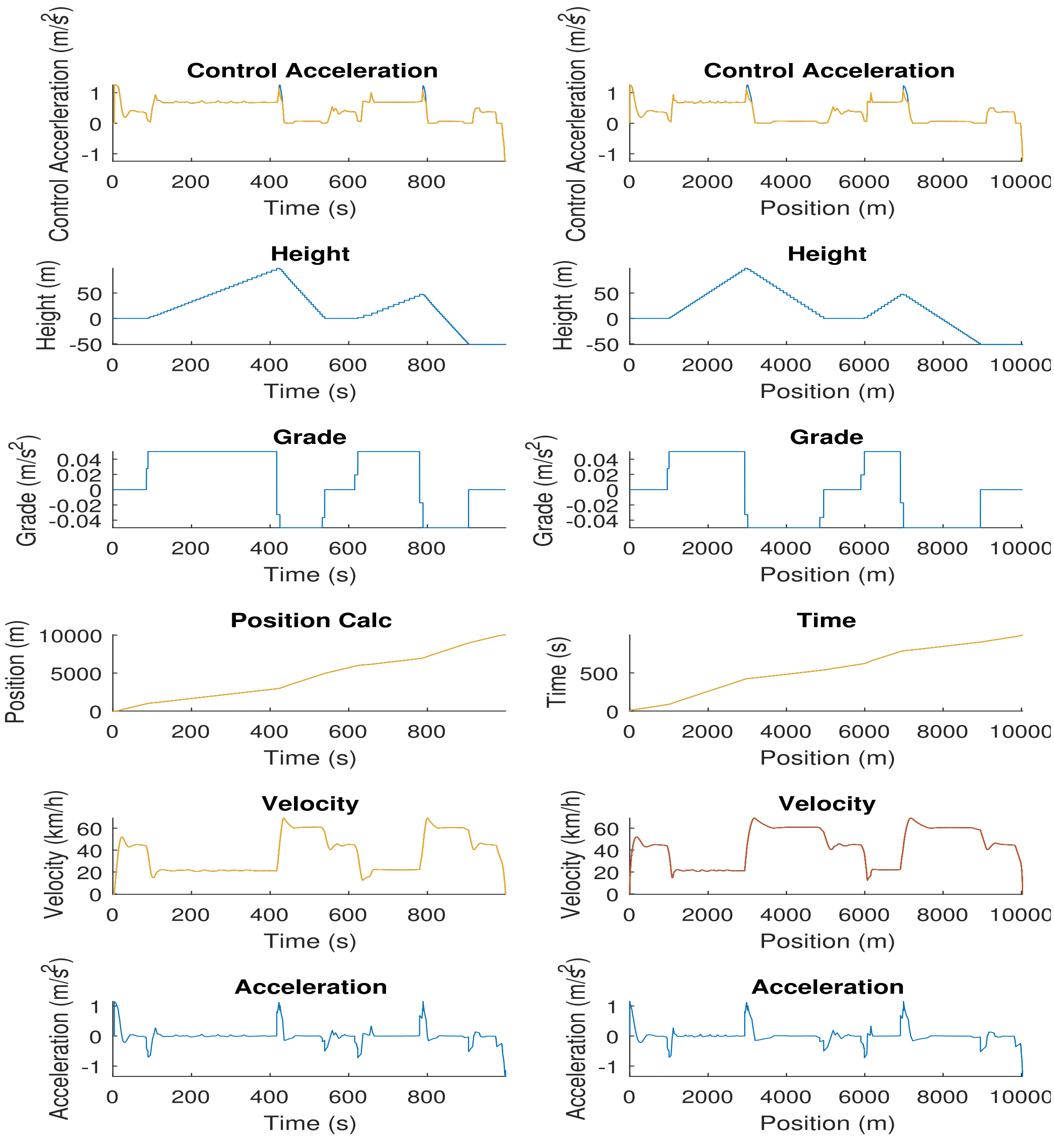

Figure 8.21: Minimum-energy ignoring delays in the control input; closed-loop response. $T_{\mathrm{trk,delay}} = 3.5\,\mathrm{s}$, $T_{\mathrm{brk,delay}} = 7.0\,\mathrm{s}$. Final Position: 10,002.22 m. Overshoot: 2.22 m. Trip Time: 997.29 s. Trip Delay: N/A s. Energy Consumed: 3451.66 J.

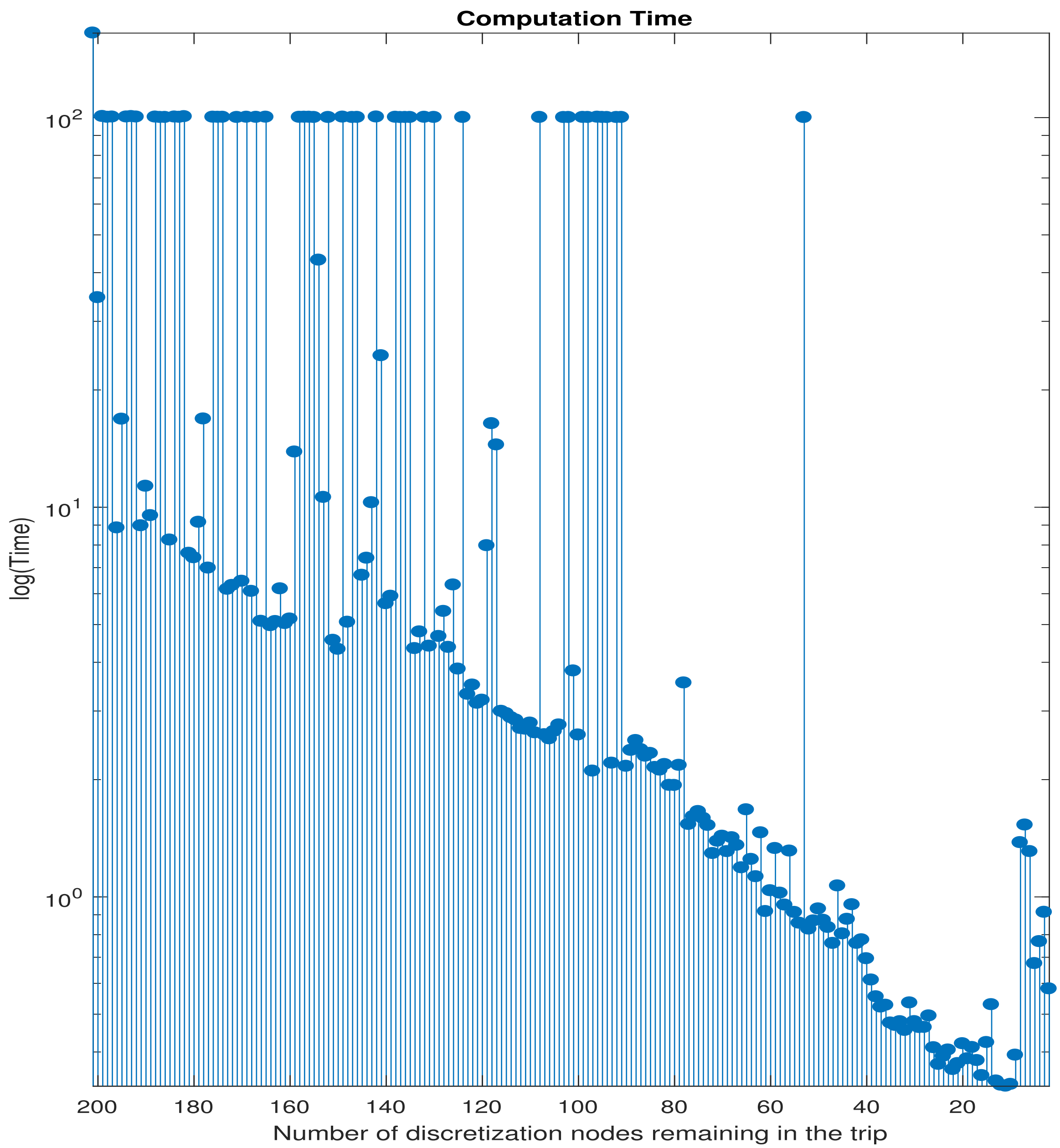

Figure 8.22: Minimum-energy ignoring delays in the control input; computation times. $T_{\text{trk,delay}} = 3.5\,\text{s}$, $T_{\text{brk,delay}} = 7.0\,\text{s}$.

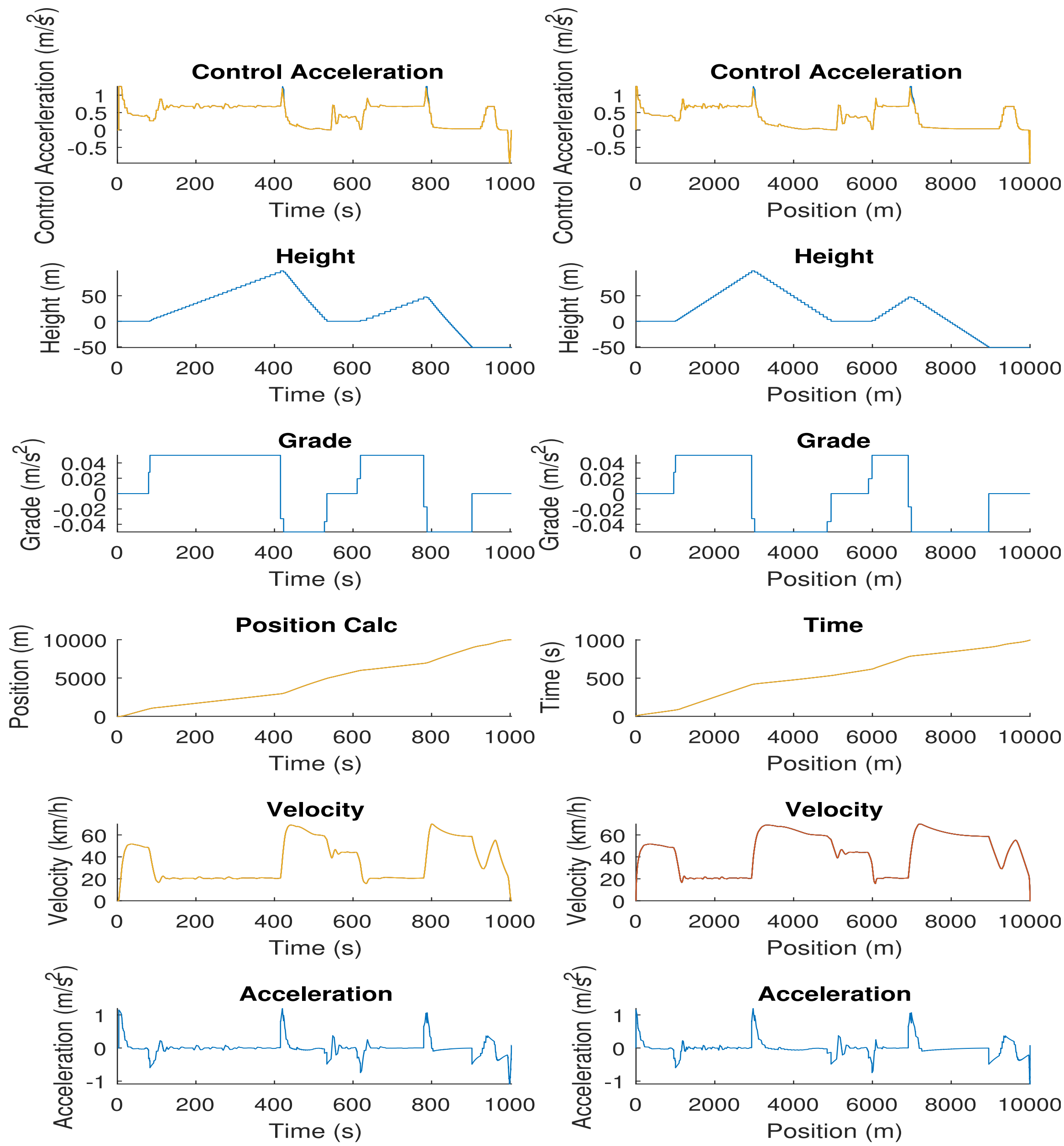

Figure 8.23: Minimum-energy considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 3.5\,\text{s}$, $T_{\text{brk,delay}} = 7.0\,\text{s}$. Final Position: 10,000.34 m. Overshoot: 0.34 m. Trip Time: 1003.24 s. Trip Delay: 3.24 s. Energy Consumed: 3550.94 J.

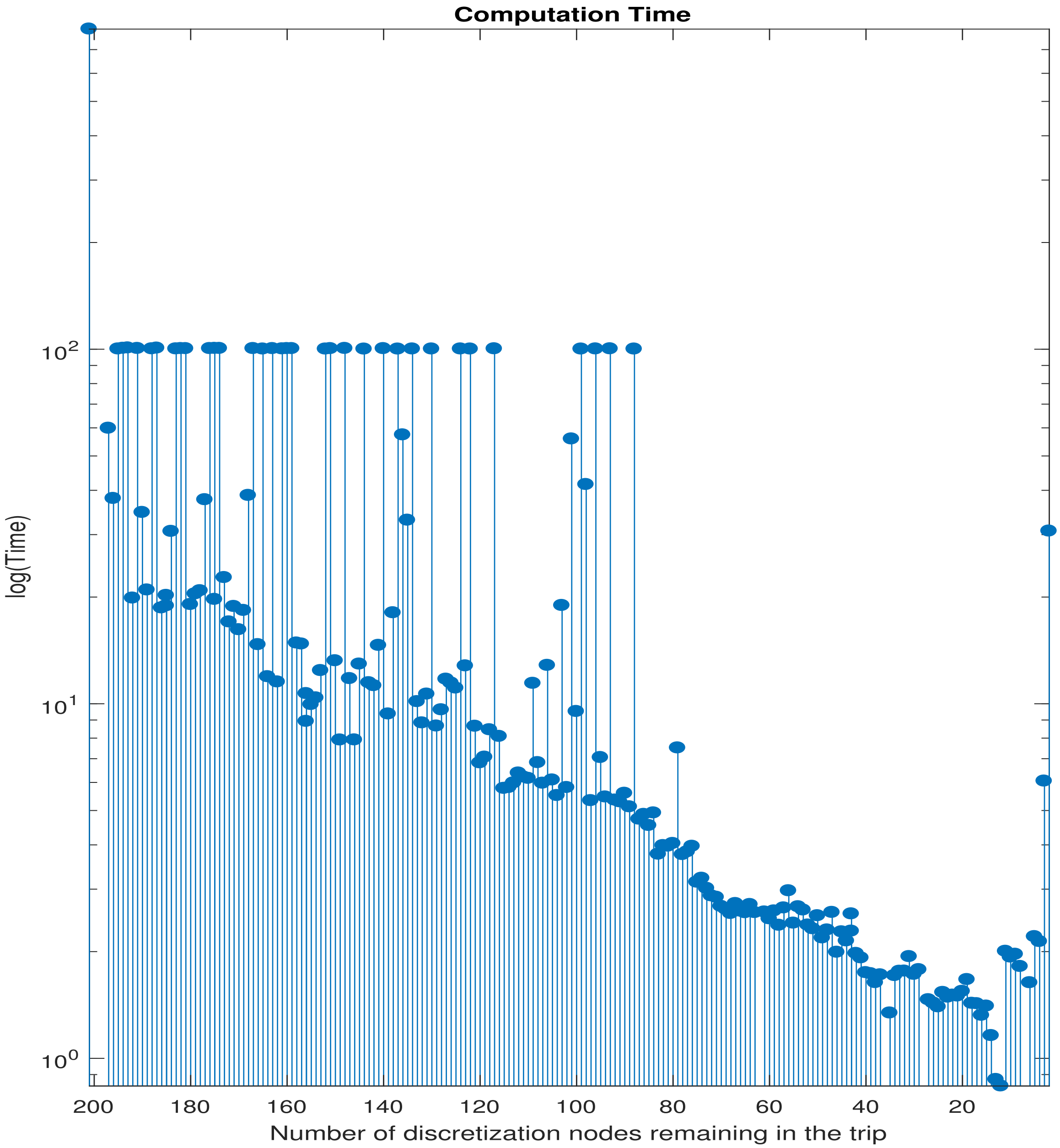

Figure 8.24: Minimum-energy considering delays in the control input; computation times. $T_{\text{trk,delay}} = 3.5\,\text{s}$, $T_{\text{brk,delay}} = 7.0\,\text{s}$.

Comparing the nonconvex, i.e. accurate dynamics, optimal control formulation vs the convex approximation optimal control formulation: for the minimum-time case the convex approximation resulted in a 4 % greater energy consumption and a trip time that is -0.73 % smaller but with an overshoot of around 2.15 m vs -0.27 m; for the minimum-energy case the convex approximation resulted in almost a 0.17 % greater energy consumption, a trip time that is 0.60 % larger, and an overshoot of 0.73 m vs 0.07 m. For the minimum-time case, the overshoot of 2.15 m is high, however, it can be dealt with another control system that activates near the destination whose objective is to stop as accurately as possible at the destination, as is done currently [31]. Switching to another controller near the end of the trip will not significantly impact the performance of the entire trip, since the non-optimal controller will be operating the train for a very small portion of the overall trip. The performance of the convex approximation is very similar to the nonconvex controller. The greatest advantage to the convex formulation is its rapid computation time, that is one to three orders of magnitude smaller than using the nonconvex formulation which uses an accurate dynamics model. Given that the convex formulation has a much shorter computation time and can be solved more robustly than the nonconvex formulation, the slight degradation in the controller performance due to the extra delay may very well be acceptable in many scenarios.

8.2.4 Test Problem 3: Equal Traction-Braking Delays with Disturbances (Discrete-Position)

In this test case, random noise is added to the sensor measurements, both position and velocity, and the process, i.e. a random acceleration is added to the system.

Position sensor noise: Random Gaussian noise is added to the position sensor measurement with the following parameters:

- Position sensor mean = 0
- Position sensor variance = 0.1

Velocity sensor noise: Random Gaussian noise is added to the velocity sensor measurement with the following parameters:

- Velocity sensor mean = 0
- Velocity sensor variance = 0.1

The position and velocity sensor noises are uncorrelated and independent. The sensor noises are independent of the process noise.

Process noise: A random uniform noise is added to the acceleration experienced by the train, i.e. the noise is multiplied by the mass to get a force that is added to the velocity dynamics. The noise has the following parameters:

- Process (uniform) noise lower bound = −0.1
- Process (uniform) noise upper bound = 0.1

The nonconvex optimization formulation presented in Eq. (7.2.1) is used to solve the optimal control problem for the case of ignoring traction-braking delays. The values of the parameters are listed in Subsection 8.2.3.

The delay values are:

- $T_{\text{trk,delay}} = 1.4\,\text{s}$
- $T_{\text{brk,delay}} = 2.8\,\text{s}$
- $n_{\text{cc}} = 40$ (dependent on length of longest delay)

8.2.4.1 Minimum-Time Optimal Control

The objective function, Eq. (7.2.1.29), parameters are: $w_t = 1$, $w_e = 0$, and $w_{\Delta\text{ctrl}} = 0$.

The results ignoring delays are shown in Figs. 8.25 and 8.26.

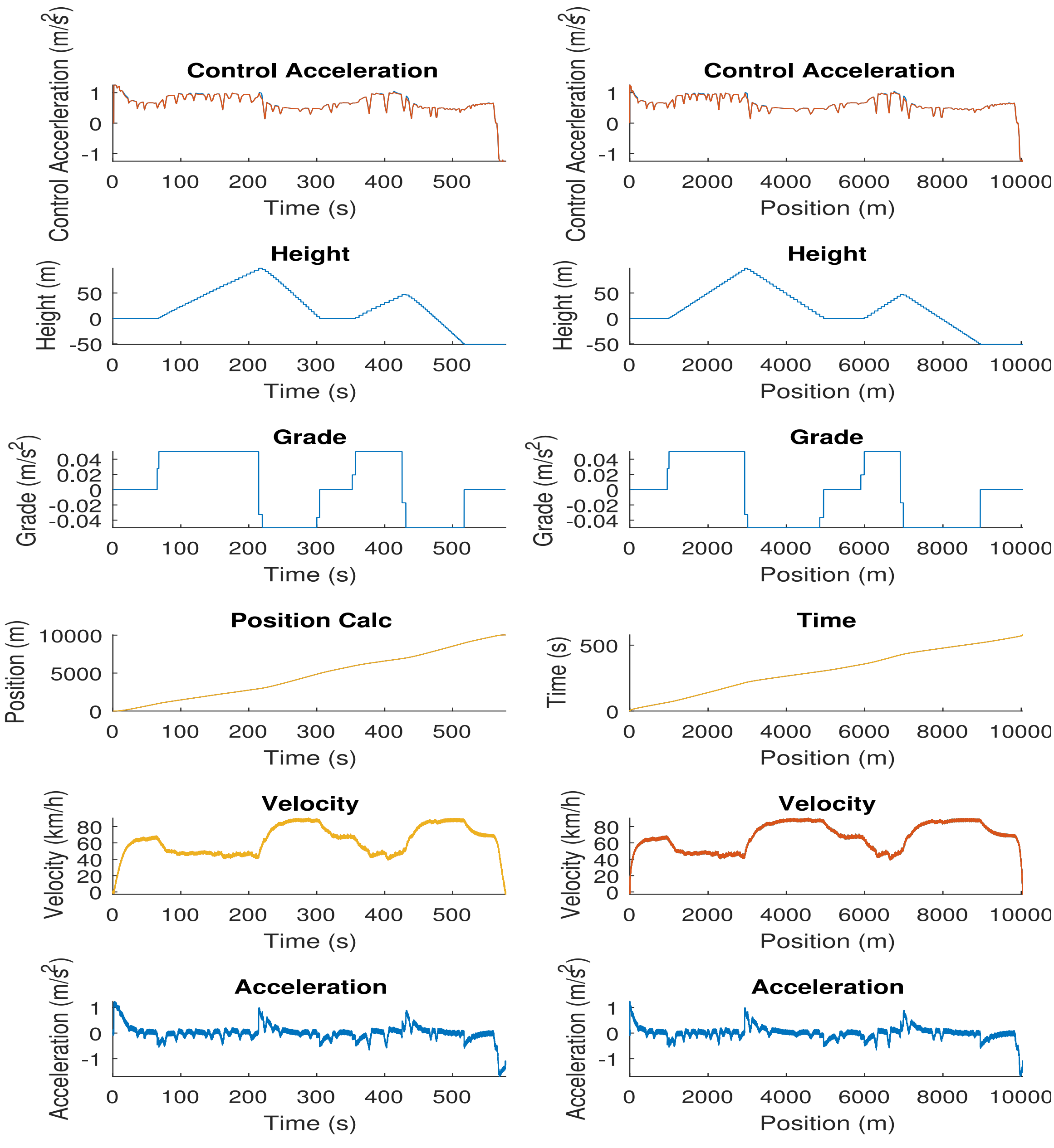

Figure 8.25: Sensor and process noise. Minimum-time ignoring delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10031.30 m. Overshoot: 31.30 m. Trip Time: 577.85 s. Trip Delay: N/A s. Energy Consumed: 6244.91 J.

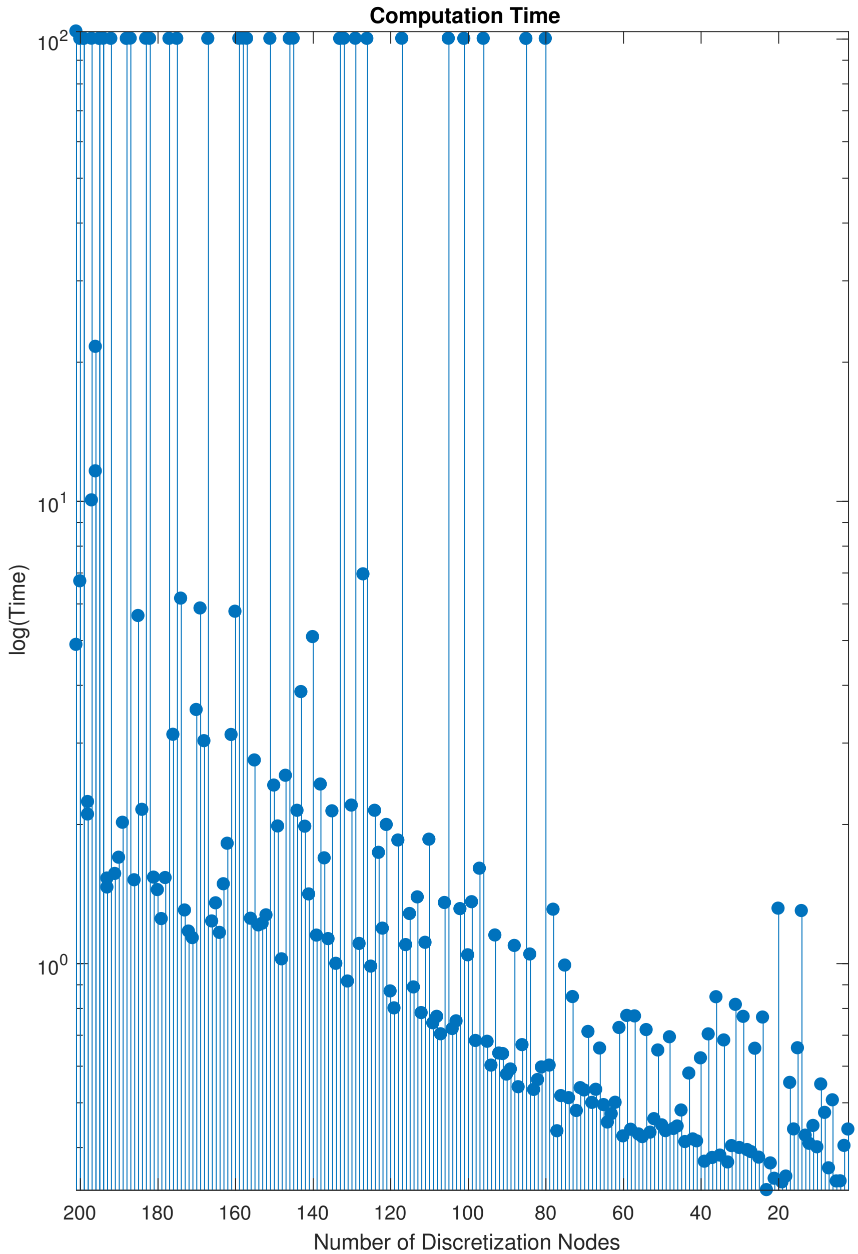

Figure 8.26: Sensor and process noise. Minimum-time ignoring delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

As can be seen from Fig. 8.25, the presence of noise has now compounded the already poor performance due to ignoring the delays and resulted in an overshoot of around 31.30 m, which is 6 m larger than the case of ignoring delays with no noise (25.57 m), as shown in Fig. 8.9.

The convex optimization formulation presented in Eq. (5.2.1) is used here to solve the problem considering equal traction-braking delays. It can be assumed that an artificial delay has been added to the traction command to make the two delays equal. The formulation parameters are listed in Subsection 8.2.2.

The delay values are:

- $T_{\text{trk,delay}} = 2.8\,\text{s}$
- $T_{\text{brk,delay}} = 2.8\,\text{s}$

The objective function, Eq. (7.2.1.29), parameters are: $w_\rho = 1,000$, and $w_e = 0$.

The results considering delays are shown in Figs. 8.27 and 8.28.

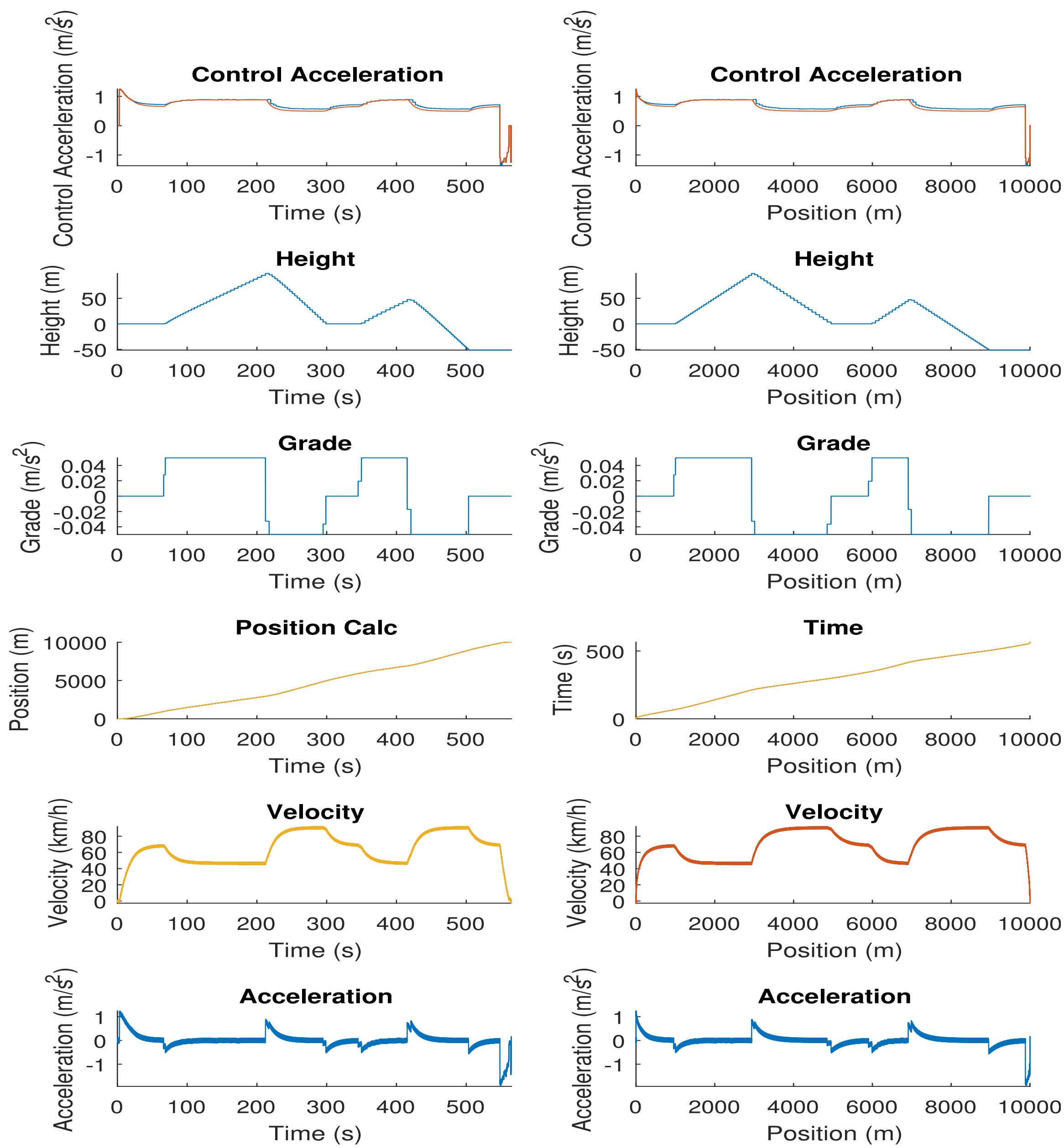

Figure 8.27: Sensor and process noise. Convex formulation: minimum-time considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 2.8\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10001.64 m. Overshoot: 1.64 m. Trip Time: 564.76 s. Trip Delay: N/A s. Energy Consumed: 6569.29 J.

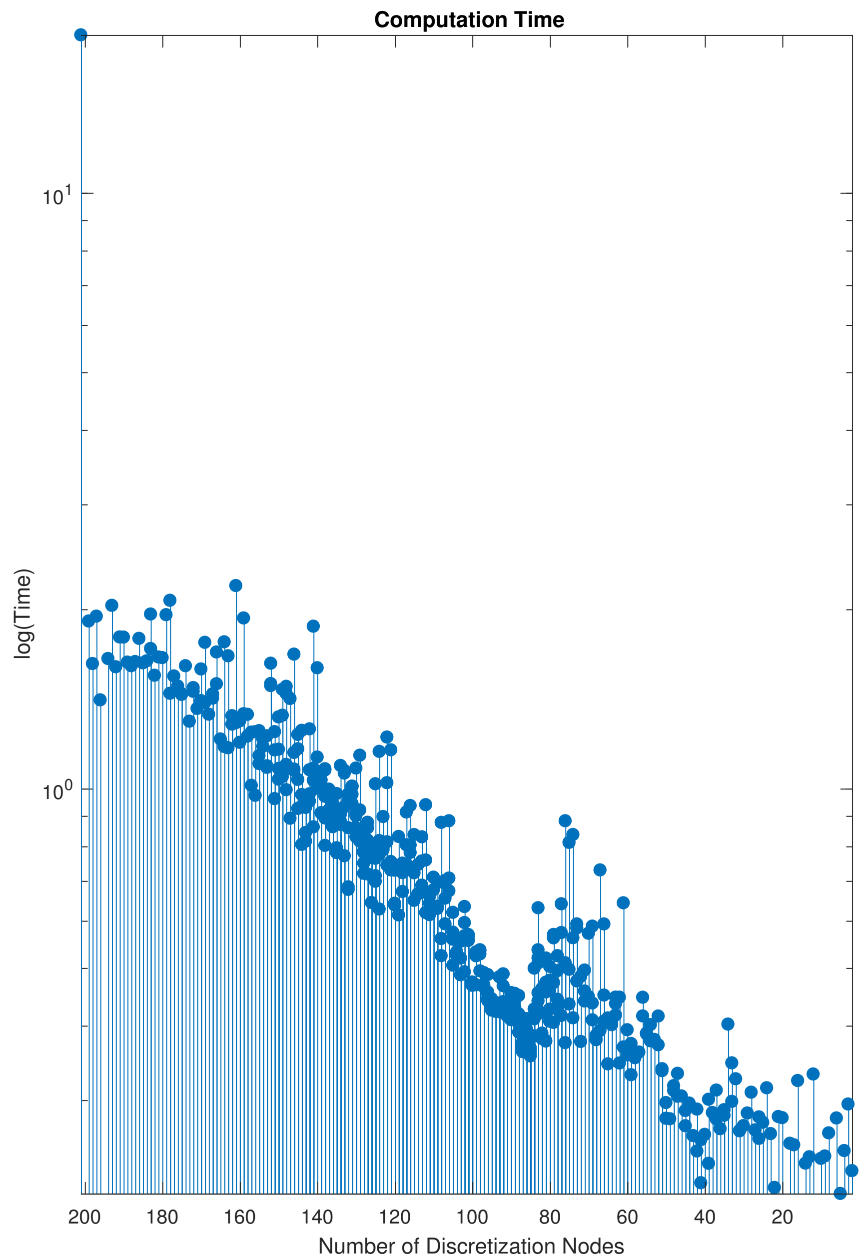

Figure 8.28: Sensor and process noise. Convex formulation: minimum-time considering delays in the control input; computation times. $T_{\text{trk,delay}} = 2.8\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

As can be seen from Fig. 8.27, the controller is quite robust when dealing with noise. The overshoot in the convex case is 1.64 m, Fig. 8.27, while the overshoot in the nonconvex case ignoring delays is 31.30 m, Fig. 8.25. The trip time in the convex case is 564.76 s, while in the nonconvex case ignoring delays is 577.85 s. The convex controller considering delays has a much smaller overshoot and a smaller trip time, which is the objective.

8.2.5 Test Problem 4: Non-Equal Traction-Braking Delays with Disturbances and Modelling Error, Underestimating Mass (Discrete-Time)

In this test case, just like in Subsection 8.2.4, random noise is added to the sensor measurements, both position and velocity, and the process, a random acceleration is added to the system. In addition, there is also model mismatch, the actual mass is 5 % greater than the estimated mass. The estimated mass is used for the purposes of control. In other words, the actual mass is 5 % greater than the mass parameter value used in the controller.

The noise parameters are the same as in Subsection 8.2.4.

The nonconvex optimization formulation presented in Eq. (7.2.1) is used to solve the optimal control problem for the case of non-equal traction-braking delays. The formulation parameters are listed in Subsection 8.2.3.

The delay values are:

- $T_{\text{trk,delay}} = 1.4\,\text{s}$
- $T_{\text{brk,delay}} = 2.8\,\text{s}$

- $n_{\text{cc}} = 40$ (dependent on length of longest delay)

8.2.5.1 Minimum-Time Optimal Control

The objective function, Eq. (7.2.1.29), parameters are: $w_t = 1$, $w_e = 0$, and $w_{\Delta\text{ctrl}} = 0$.

The results ignoring delays are shown in Figs. 8.29 and 8.30. The results considering delays are shown in Figs. 8.31 and 8.32.

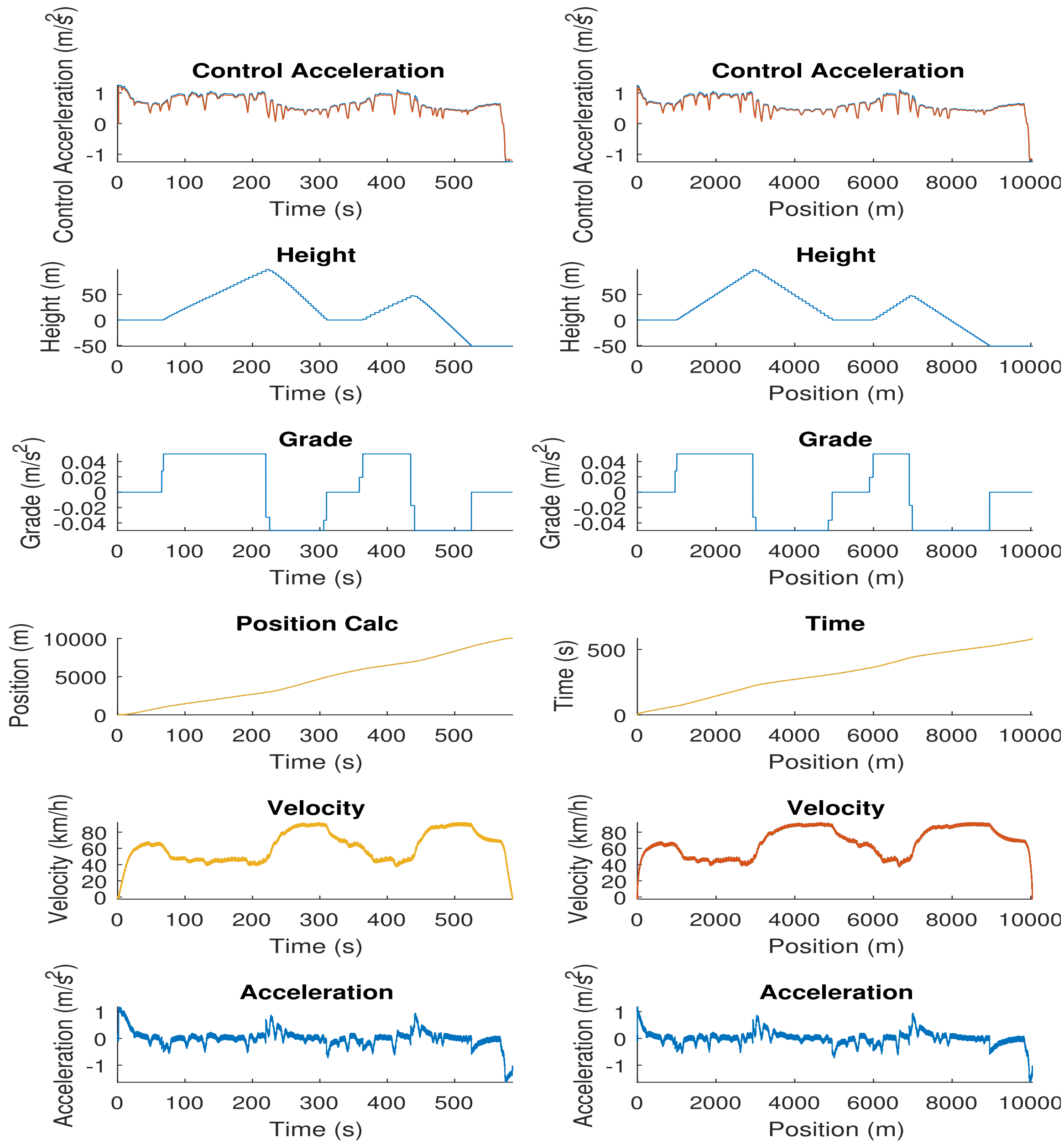

Figure 8.29: Modelling errors, underestimating the train mass, with sensor and process noise added. Minimum-time ignoring delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10037.77 m. Overshoot: 37.77 m. Trip Time: 585.76 s. Trip Delay: N/A s. Energy Consumed: 6210.42 J.

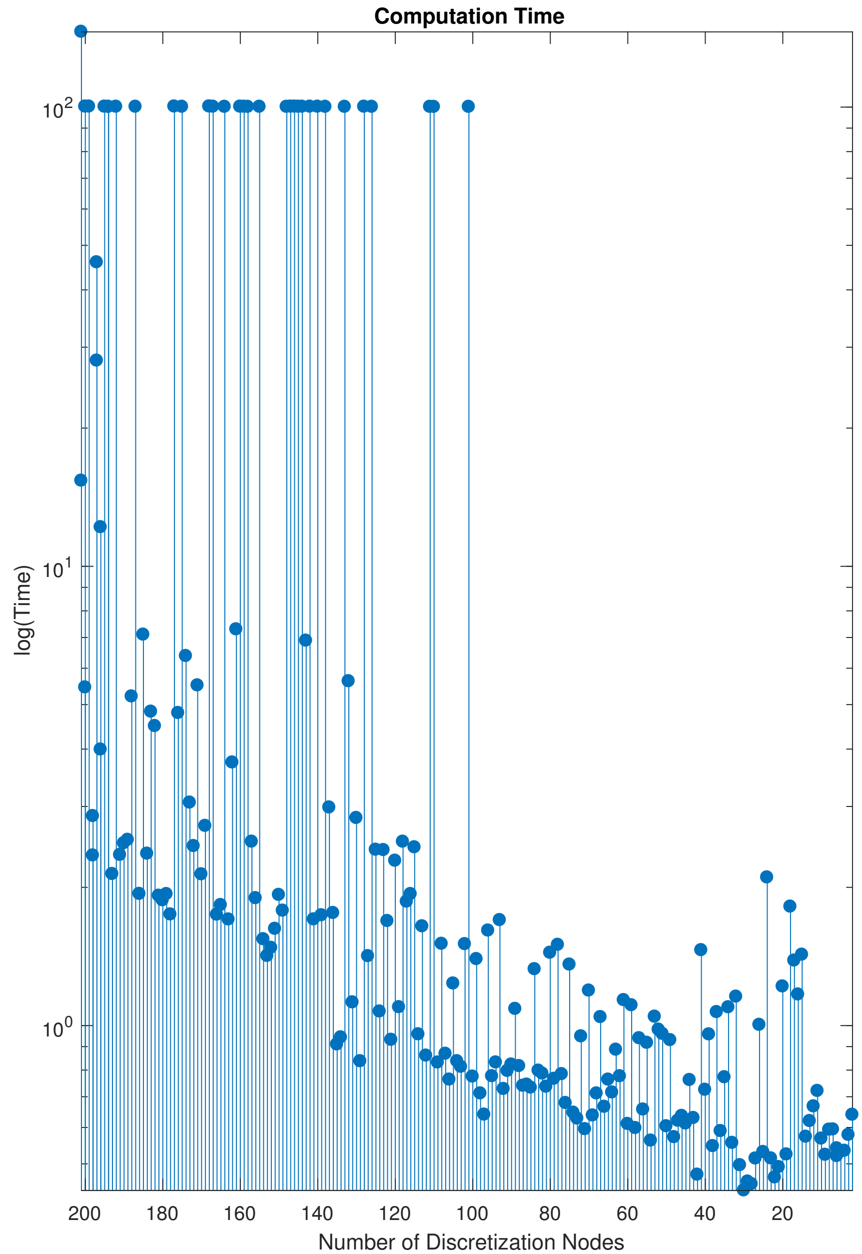

Figure 8.30: Modelling errors, underestimating the train mass, with sensor and process noise added. Minimum-time ignoring delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

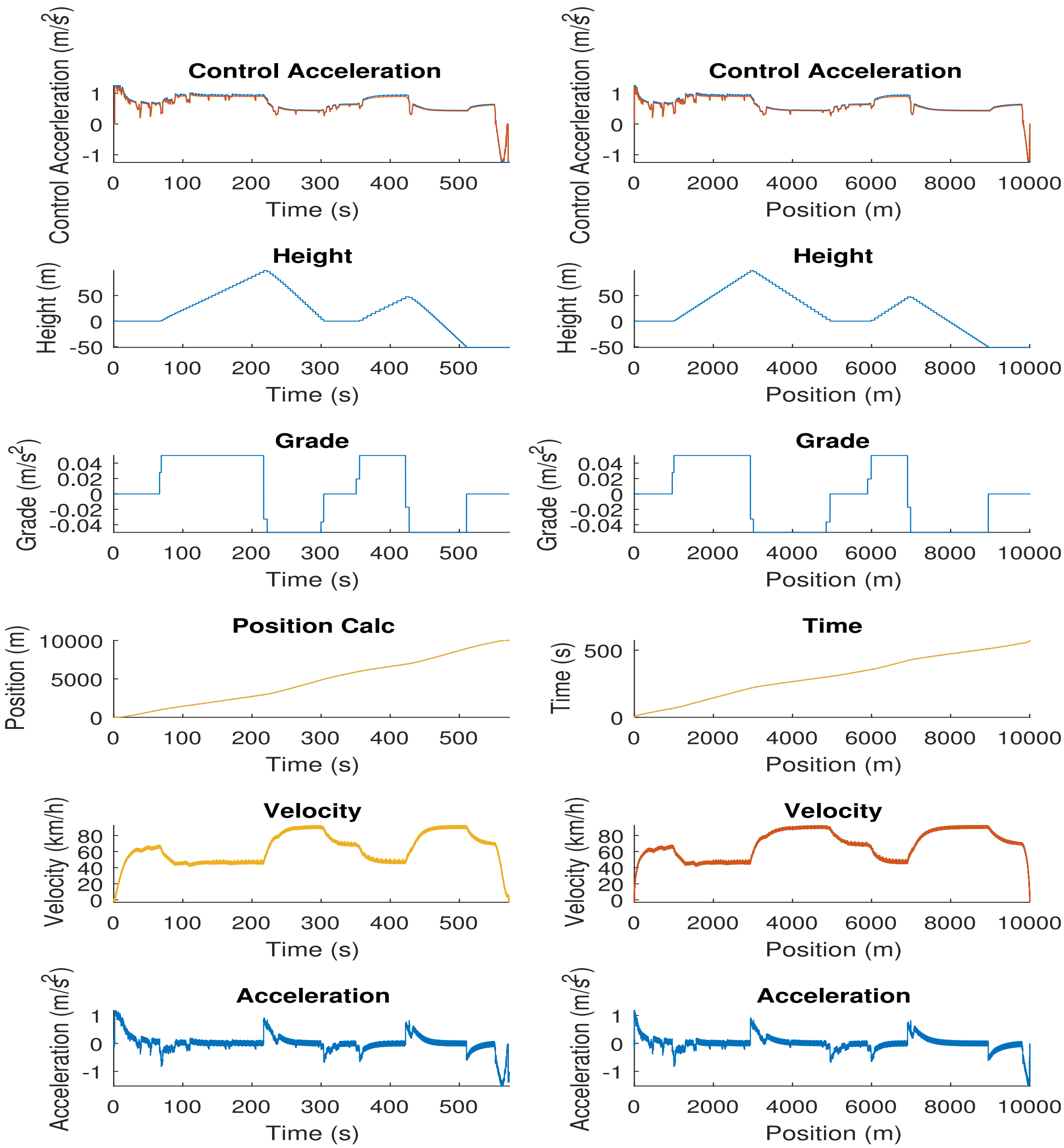

Figure 8.31: Modelling errors, underestimating the train mass, with sensor and process noise added. Minimum-time considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10005.37 m. Overshoot: 5.37 m. Trip Time: 572.18 s. Trip Delay: N/A s. Energy Consumed: 6387.85 J.

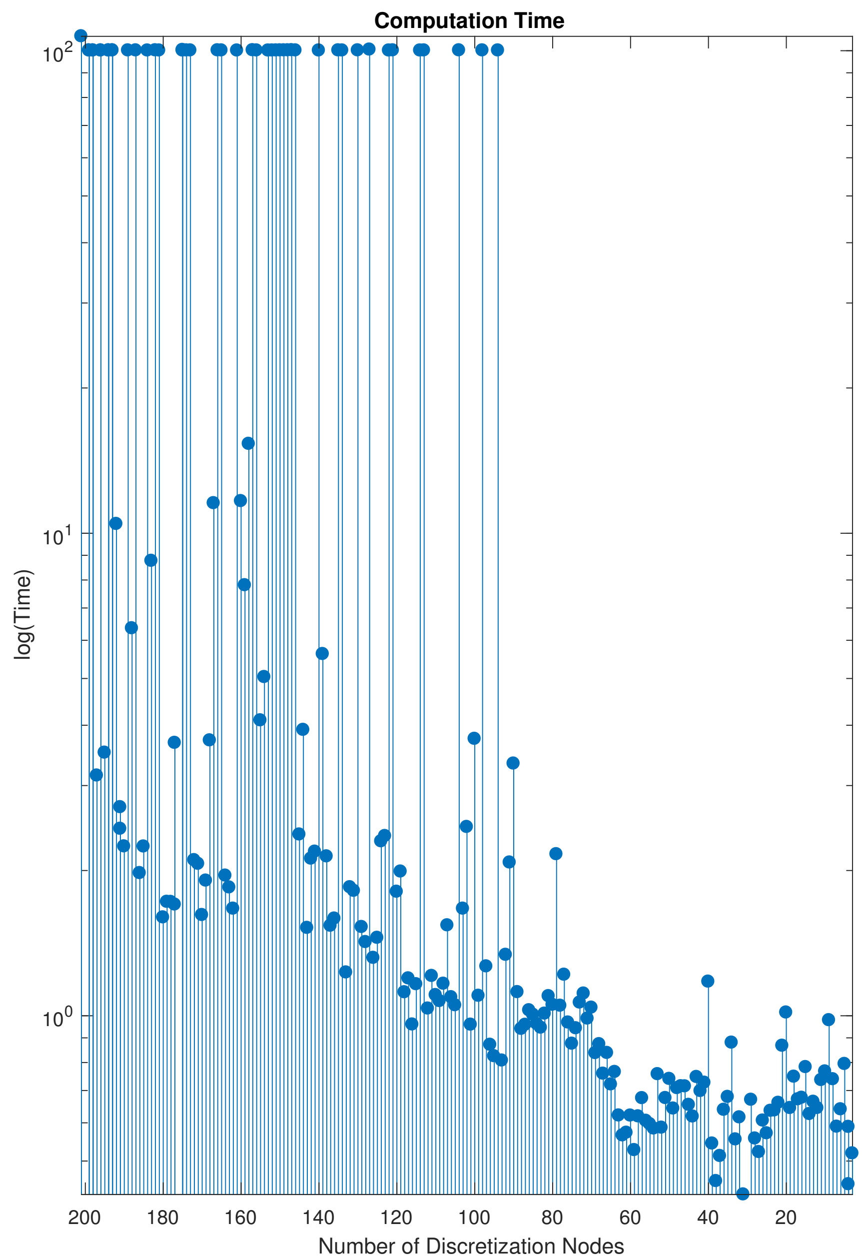

Figure 8.32: Modelling errors, underestimating the train mass, with sensor and process noise added. Minimum-time considering delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

Comparing Figs. 8.29 and 8.31, considering delays resulted in an overshoot decrease from 37.77 m to 5.37 m, and a trip time decrease from 585.76 s to 572.18 s.

8.2.5.2 Minimum-Energy Optimal Control

The objective function, Eq. (7.2.1.29), parameters are: $w_t = 0$, $w_e = 1$, and $w_{\Delta\text{ctrl}} = 1$.

The results ignoring delays are shown in Figs. 8.33 and 8.34. The results considering delays are shown in Figs. 8.35 and 8.36.

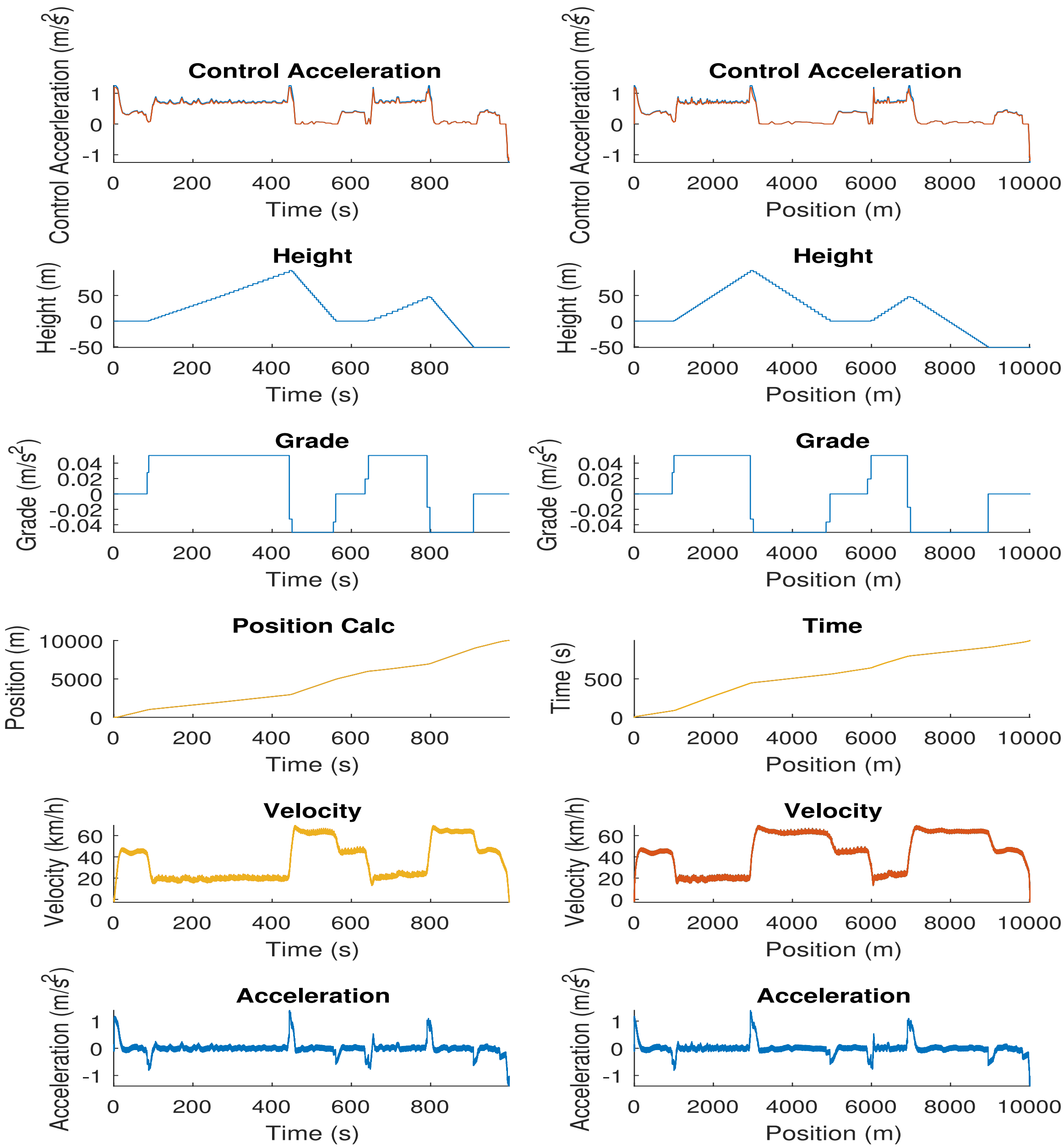

Figure 8.33: Modelling errors, underestimating the train mass, with sensor and process noise added. Minimum-energy ignoring delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10009.24 m. Overshoot: 9.24 m. Trip Time: 998.55 s. Trip Delay: N/A s. Energy Consumed: 3554.23 J.

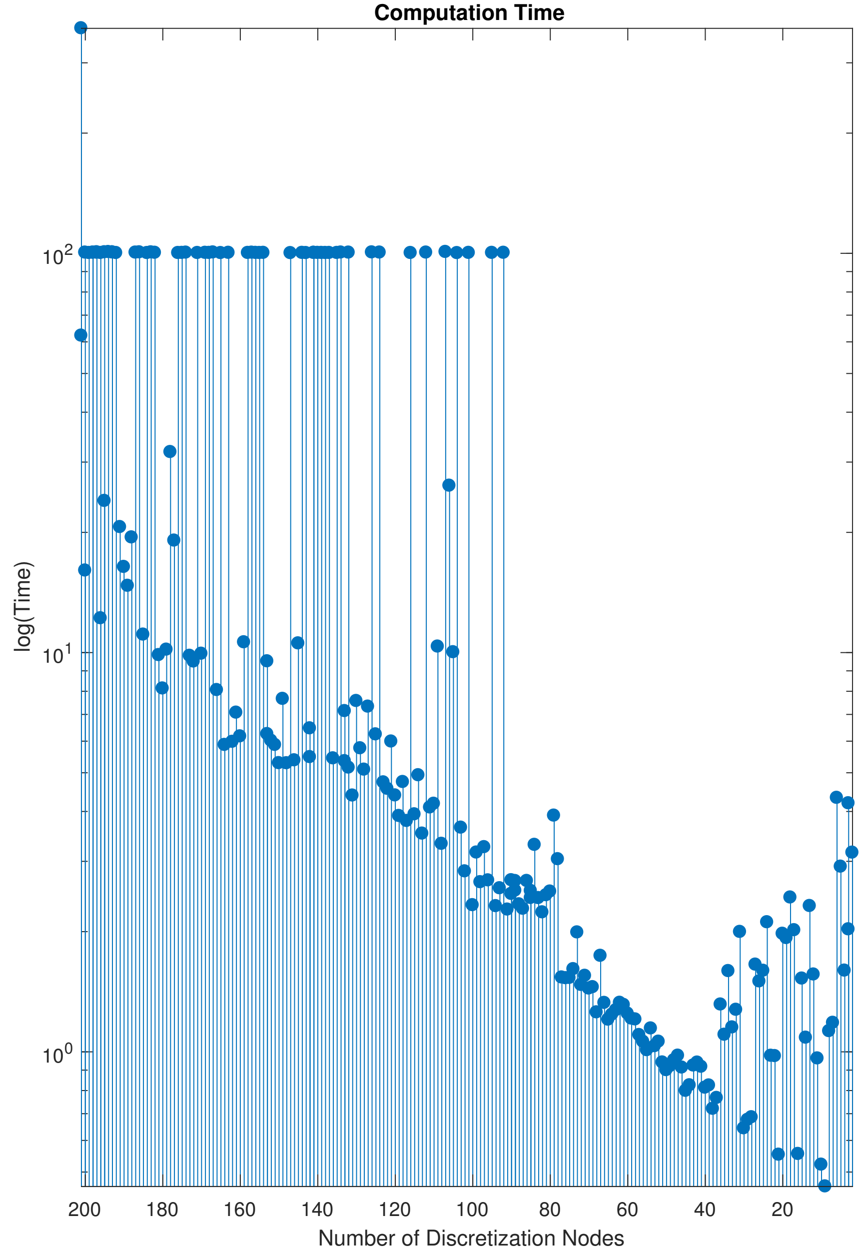

Figure 8.34: Modelling errors, underestimating the train mass, with sensor and process noise added. Minimum-energy ignoring delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

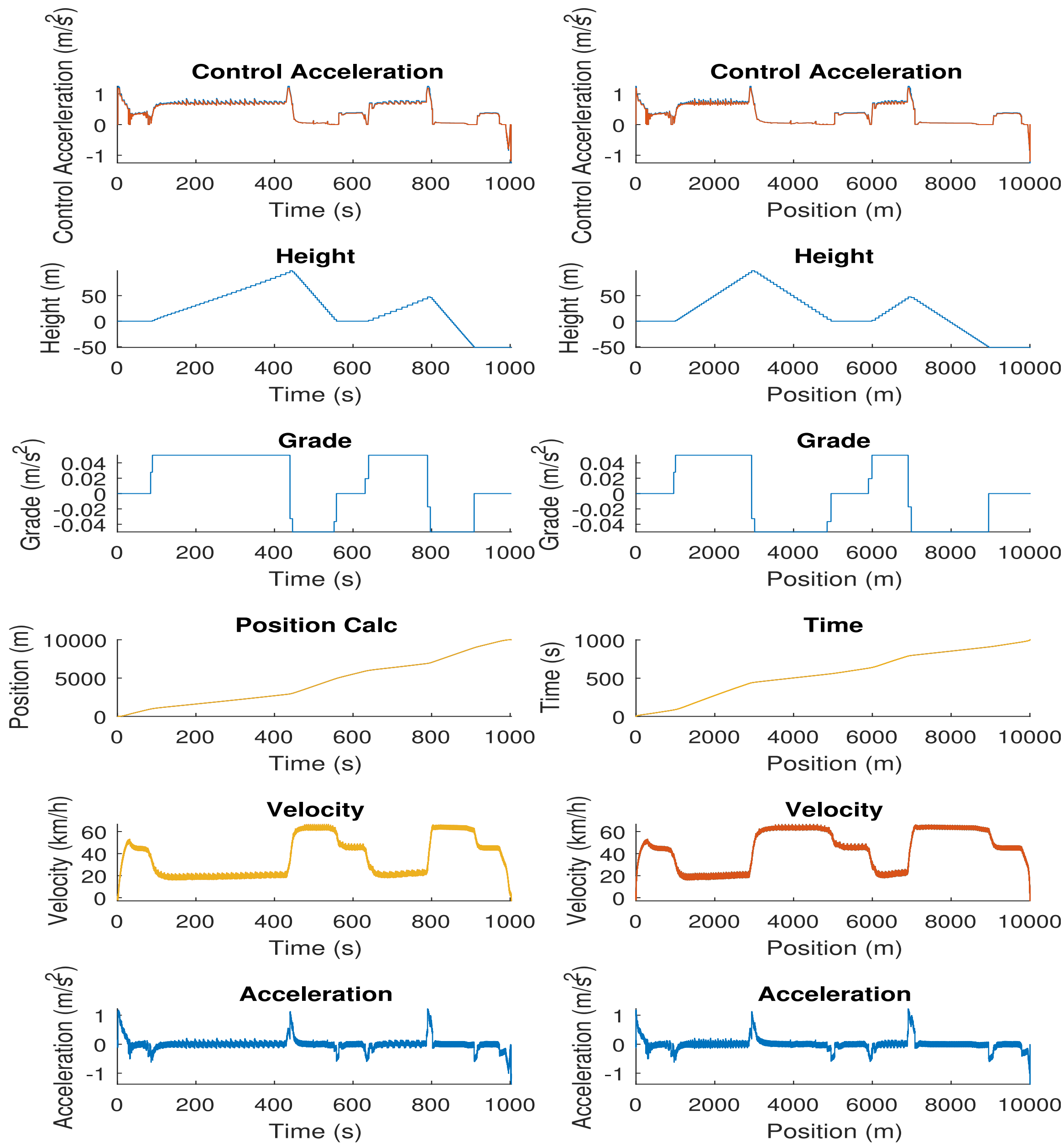

Figure 8.35: Modelling errors, underestimating the train mass, with sensor and process noise added. Minimum-energy considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10003.16 m. Overshoot: 3.16 m. Trip Time: 1003.17 s. Trip Delay: 3.17 s. Energy Consumed: 3507.59 J.

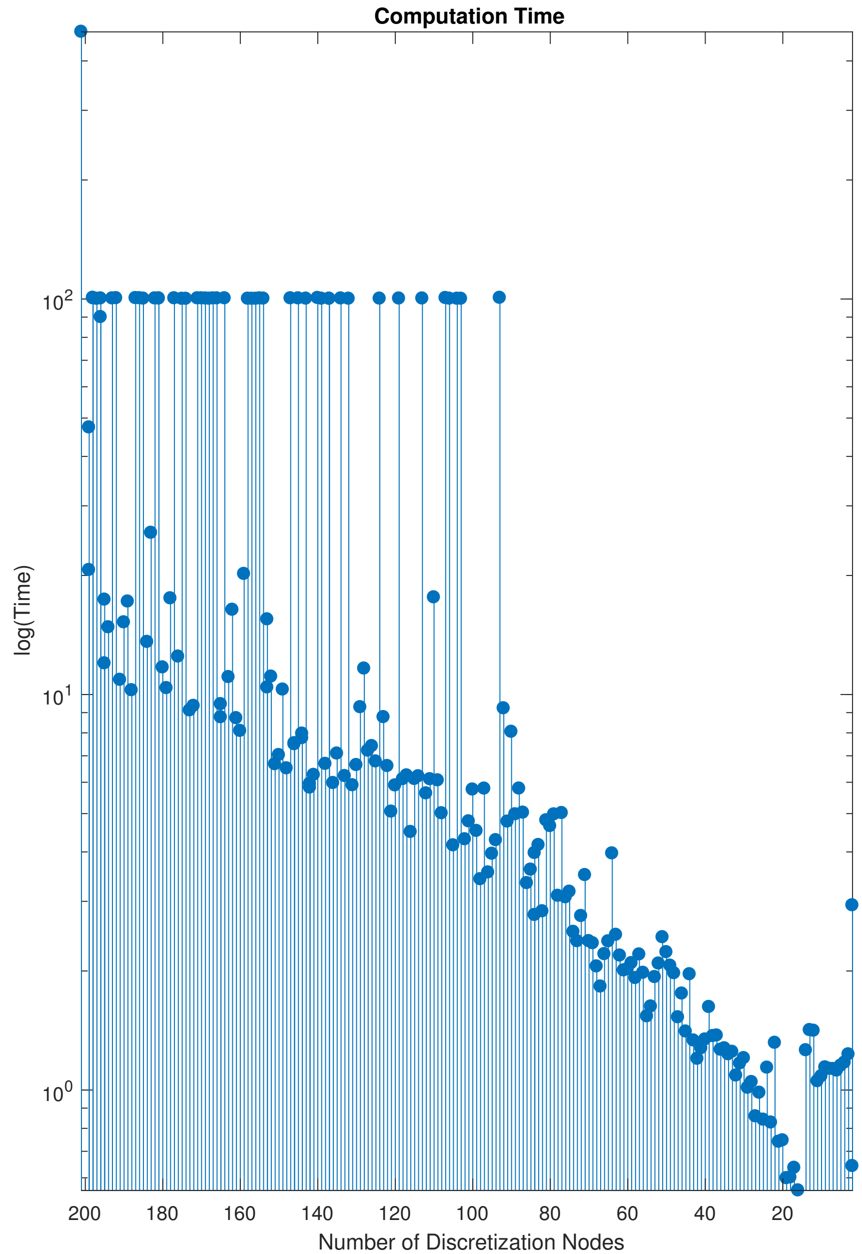

Figure 8.36: Modelling errors, underestimating the train mass, with sensor and process noise added. Minimum-energy considering delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

Comparing Figs. 8.33 and 8.35, considering delays resulted in an overshoot decrease from 9.24 m to 3.16 m, an energy consumption decrease from 3554.23 J to 3507.59 J, but a trip time increase from 998.55 s to 1003.17 s. There is a trip delay of 3.17 s for the case of considering delays.

The reason why the overshoot is worse than in the case of noise only and no modelling errors, Subsection 8.2.4, is because when the controller is configured with an underestimate of the mass parameter value, the controller requests less braking than if the controller was configured with the correct higher mass parameter value. This lower braking force request increases the stopping distance.

8.2.6 Test Problem 4-A: Equal Traction-Braking Delays with Disturbances and Modelling Error, Underestimating Mass (Discrete-Position)

The convex optimization formulation presented in Eq. (5.2.1) is used here to solve the problem considering equal traction-braking delays. It can be assumed that an artificial delay has been added to the traction command to make the two delays equal compared to Subsection 8.2.5. The formulation parameters are listed in Subsection 8.2.2.

The delay values are:

- $T_{\text{trk,delay}} = 2.8\,\text{s}$
- $T_{\text{brk,delay}} = 2.8\,\text{s}$

8.2.6.1 Minimum-Time Optimal Control

The objective function, Eq. (7.2.1.29), parameters are: $w_\rho = 1,000$, and $w_e = 0$.

The results considering delays are shown in Figs. 8.37 and 8.38.

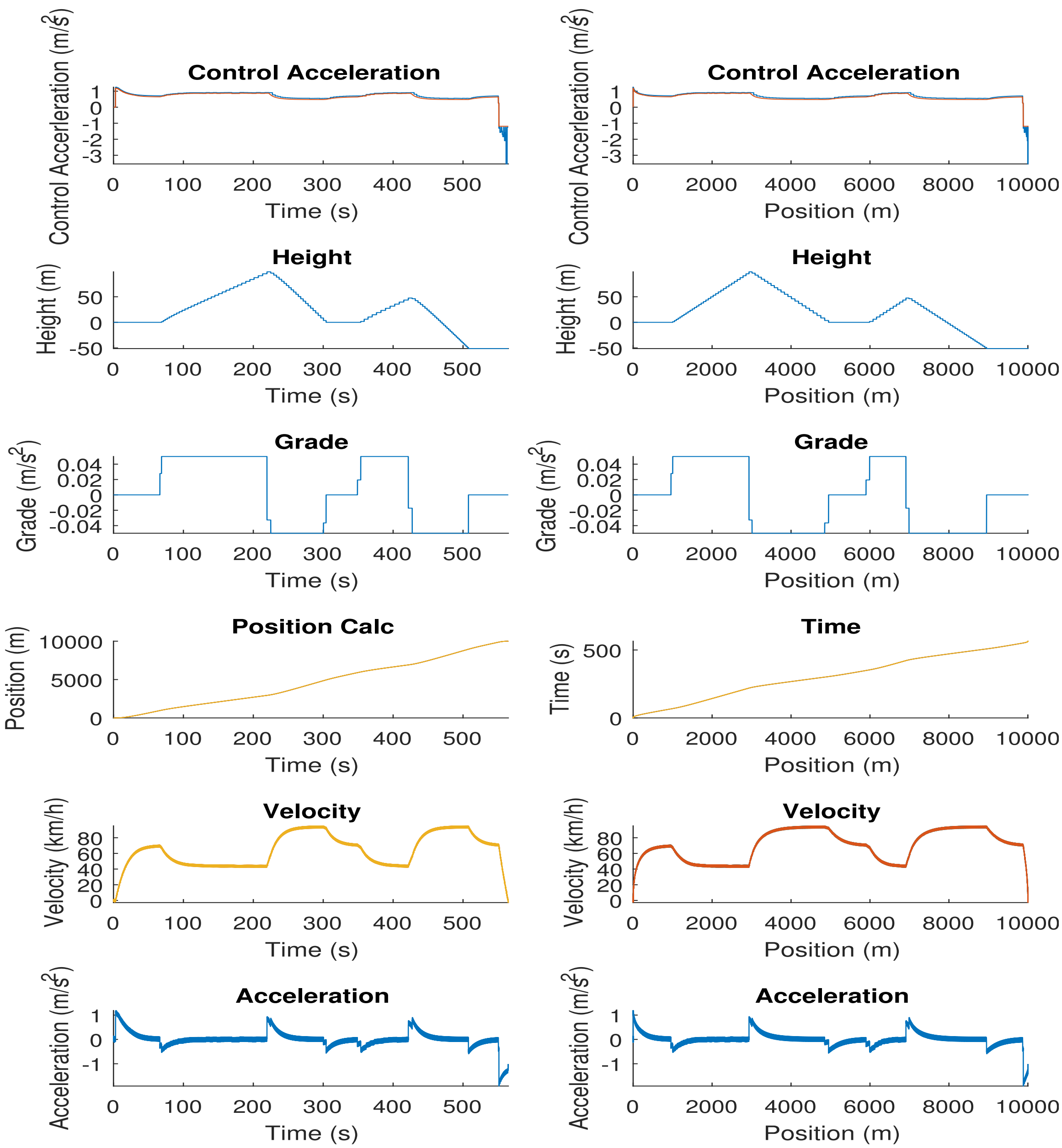

Figure 8.37: Modelling errors, underestimating the train mass, with sensor and process noise added. Convex formulation: minimum-time considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 2.8\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10002.40 m. Overshoot: 2.40 m. Trip Time: 564.76 s. Trip Delay: N/A s. Energy Consumed: 6707.40 J.

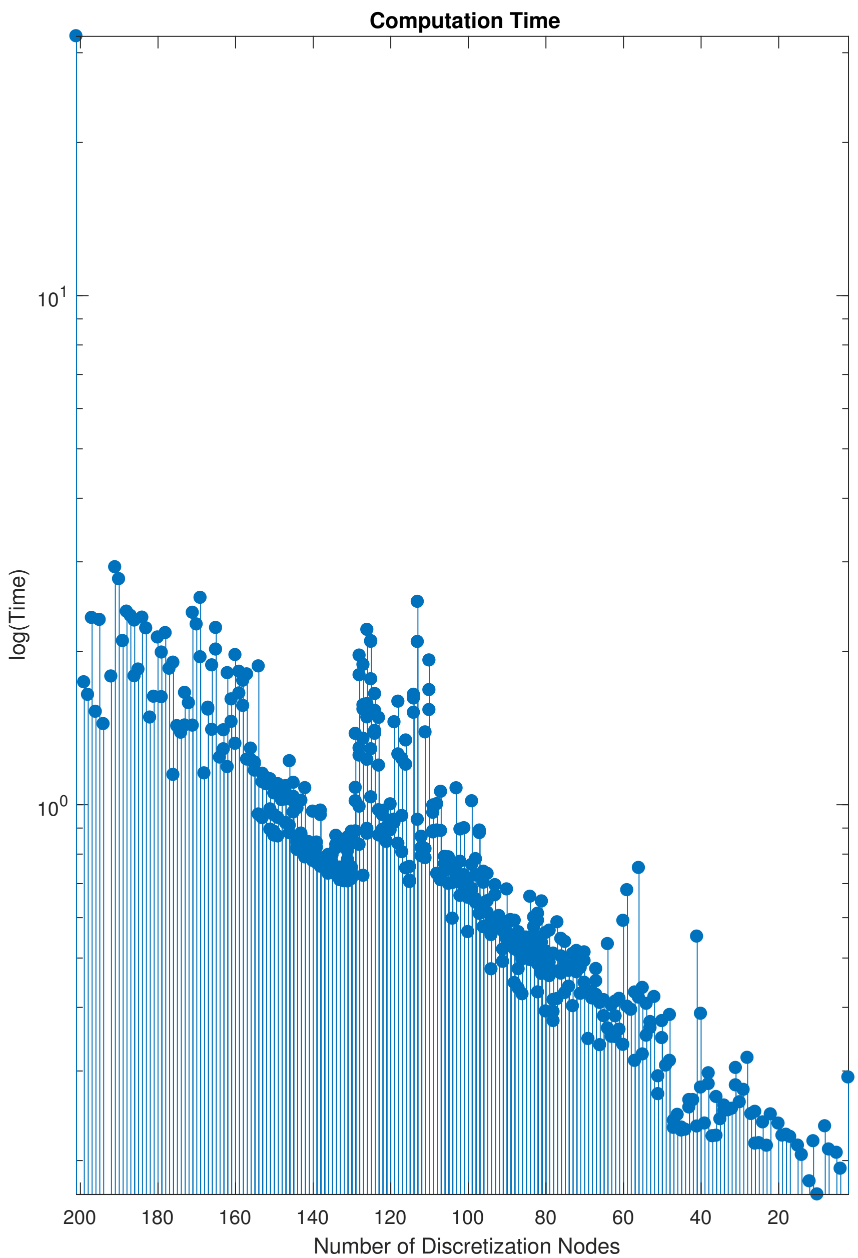

Figure 8.38: Modelling errors, underestimating the train mass, with sensor and process noise added. Convex formulation: minimum-time considering delays in the control input; computation times. $T_{\text{trk,delay}} = 2.8\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

As can be seen from Fig. 8.37, the controller is quite robust when dealing with noise and modelling errors. The overshoot in the convex case is 2.40 m, Fig. 8.37, while the overshoot in the nonconvex case (considering delays) is 5.37 m, Fig. 8.31. The trip time in the convex case is 564.76 s, while in the nonconvex case is 572.18 s. The performance is better in the convex case as opposed to the nonconvex case, but this can be explained by the increased frequency of solving optimal control problems using feedback in the convex case. For the convex case, for a single run from station A to station B, the optimal control problem is solved 401 times, for the nonconvex case, the optimal control problem is solved 201 times. Feedback occurs nearly twice as frequently in the convex case as in the nonconvex case, thus mitigating modelling errors. The increased frequency of feedback is possible for the convex case because of the ability to rapidly and robustly solve convex optimization problems.

8.2.7 Test Problem 5: Non-Equal Traction-Braking Delays with Disturbances and Modelling Error, Overestimating Mass (Discrete-Time)

In this test case, just like in Subsection 8.2.5, random noise is added to the sensor measurements, both position and velocity, and the process, a random acceleration is added to the system. In addition, there is also model mismatch, the actual mass is 5 % lower than the estimated mass. The estimated mass is used for the purposes of control. In other words, the actual mass is 5 % lower than the mass parameter value used in the controller.

The noise parameters are the same as in Subsection 8.2.4.

The nonconvex optimization formulation presented in Eq. (7.2.1) is used to solve the optimal control problem for the case of non-equal traction-braking delays. The formulation parameters are listed in Subsection 8.2.3.

The delay values are:

- $T_{\text{trk,delay}} = 1.4\,\text{s}$
- $T_{\text{brk,delay}} = 2.8\,\text{s}$
- $n_{\text{cc}} = 40$ (dependent on length of longest delay)

8.2.7.1 Minimum-Time Optimal Control

The objective function, Eq. (7.2.1.29), parameters are: $w_t = 1$, $w_e = 0$, and $w_{\Delta\text{ctrl}} = 0$.

The results ignoring delays are shown in Figs. 8.39 and 8.40. The results considering delays are shown in Figs. 8.41 and 8.42.

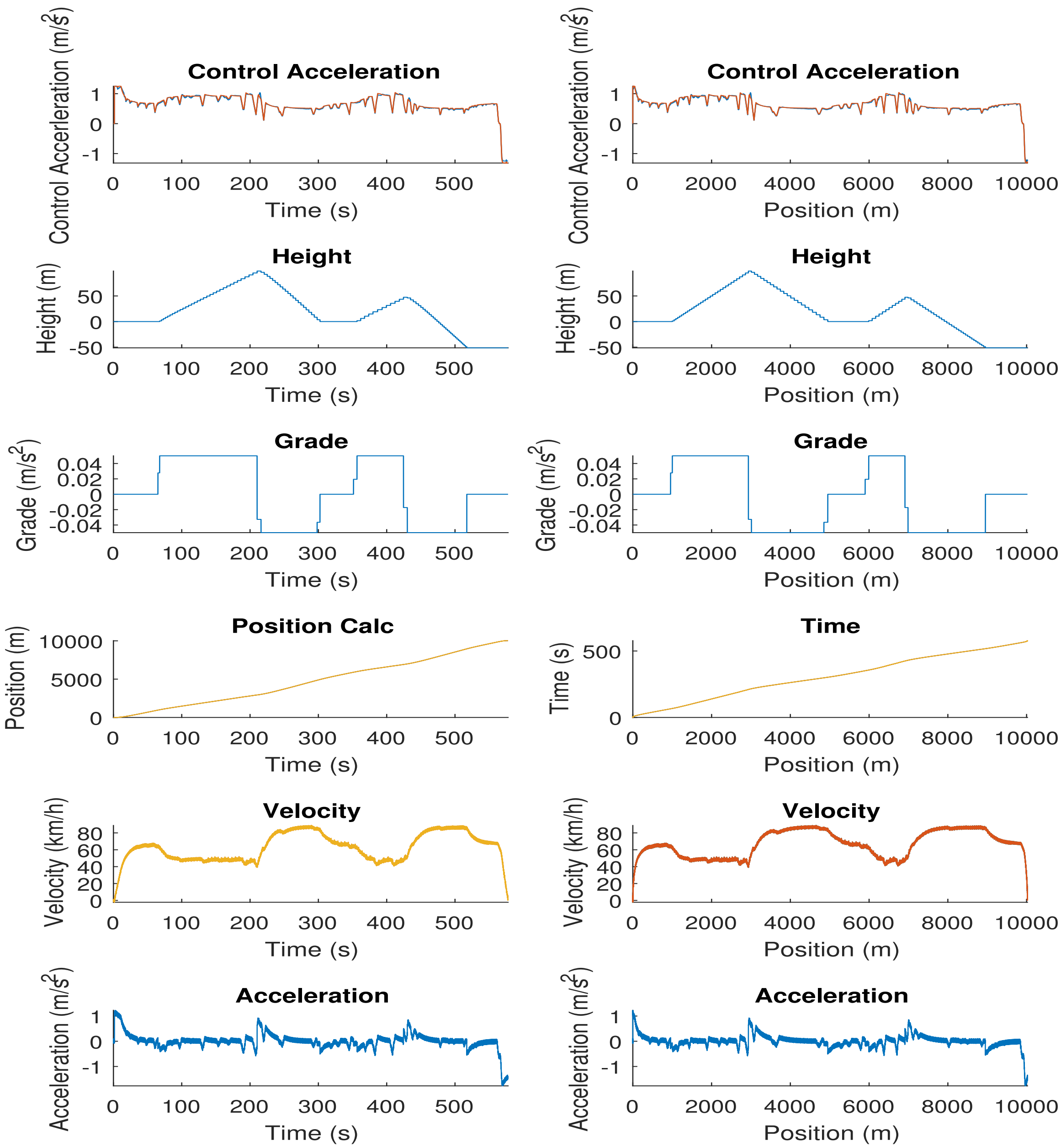

Figure 8.39: Modelling errors, overestimating the train mass, with sensor and process noise added. Minimum-time ignoring delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10021.19 m. Overshoot: 21.19 m. Trip Time: 577.64 s. Trip Delay: N/A s. Energy Consumed: 6169.69 J.

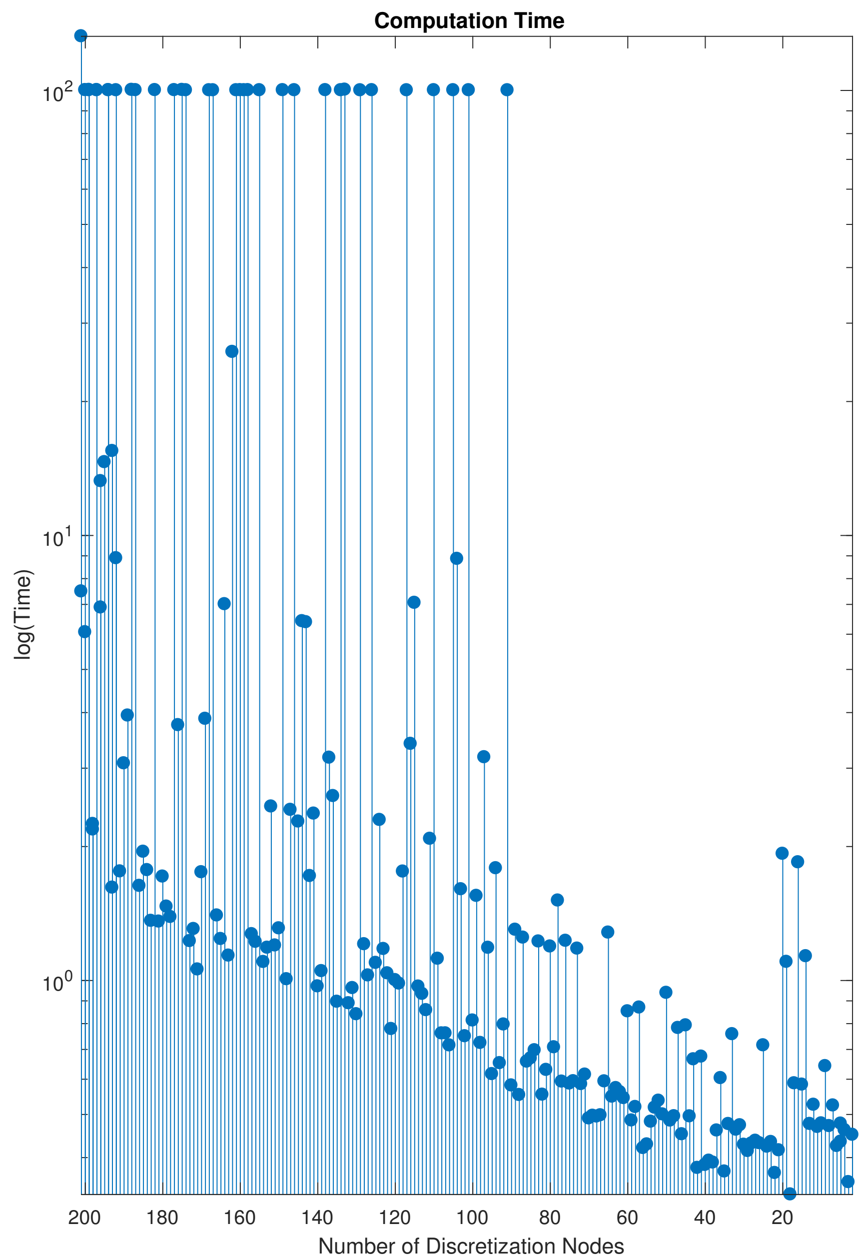

Figure 8.40: Modelling errors, overestimating the train mass, with sensor and process noise added. Minimum-time ignoring delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

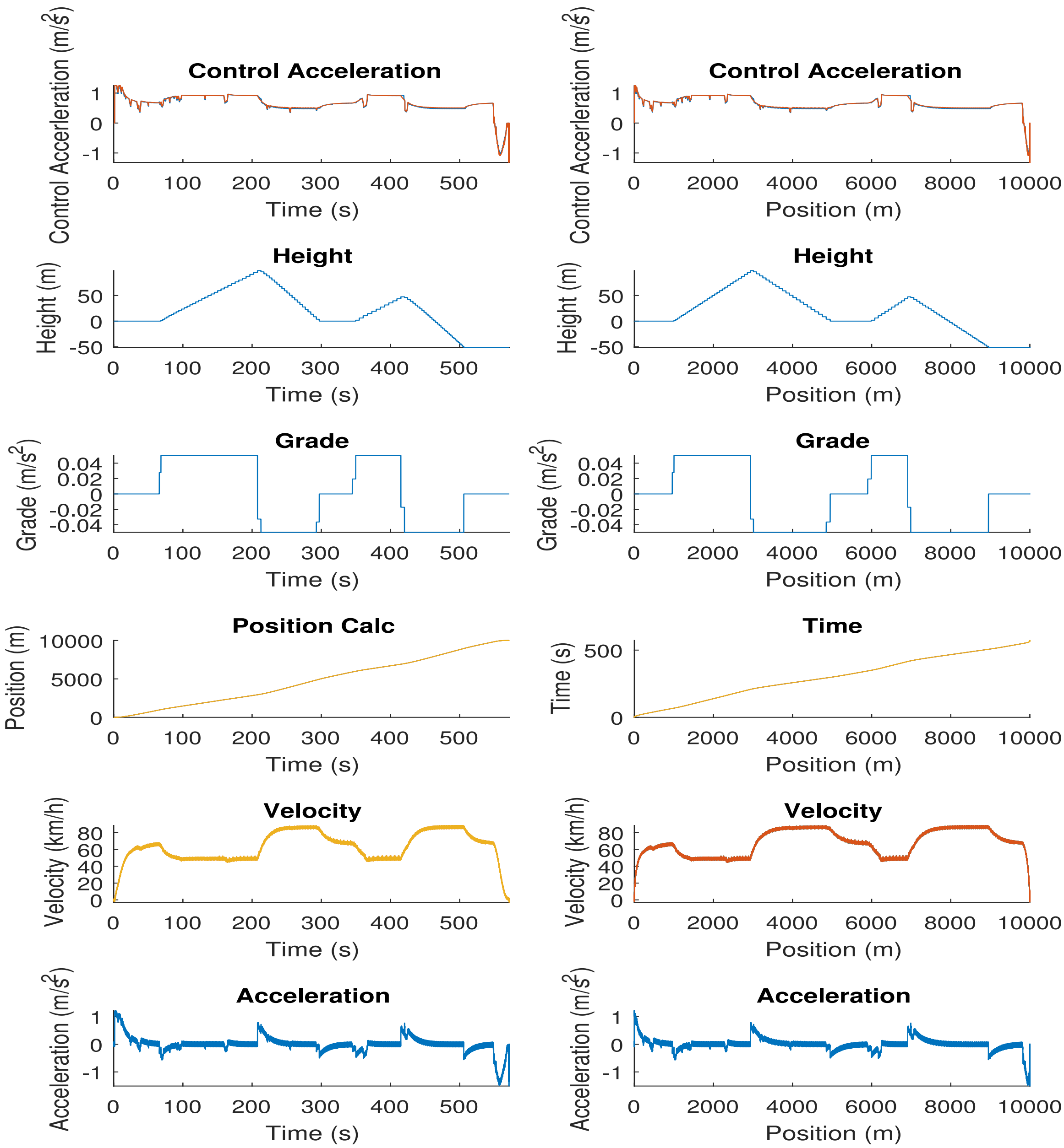

Figure 8.41: Modelling errors, overestimating the train mass, with sensor and process noise added. Minimum-time considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10001.15 m. Overshoot: 1.15 m. Trip Time: 571.62 s. Trip Delay: N/A s. Energy Consumed: 6301.26 J.

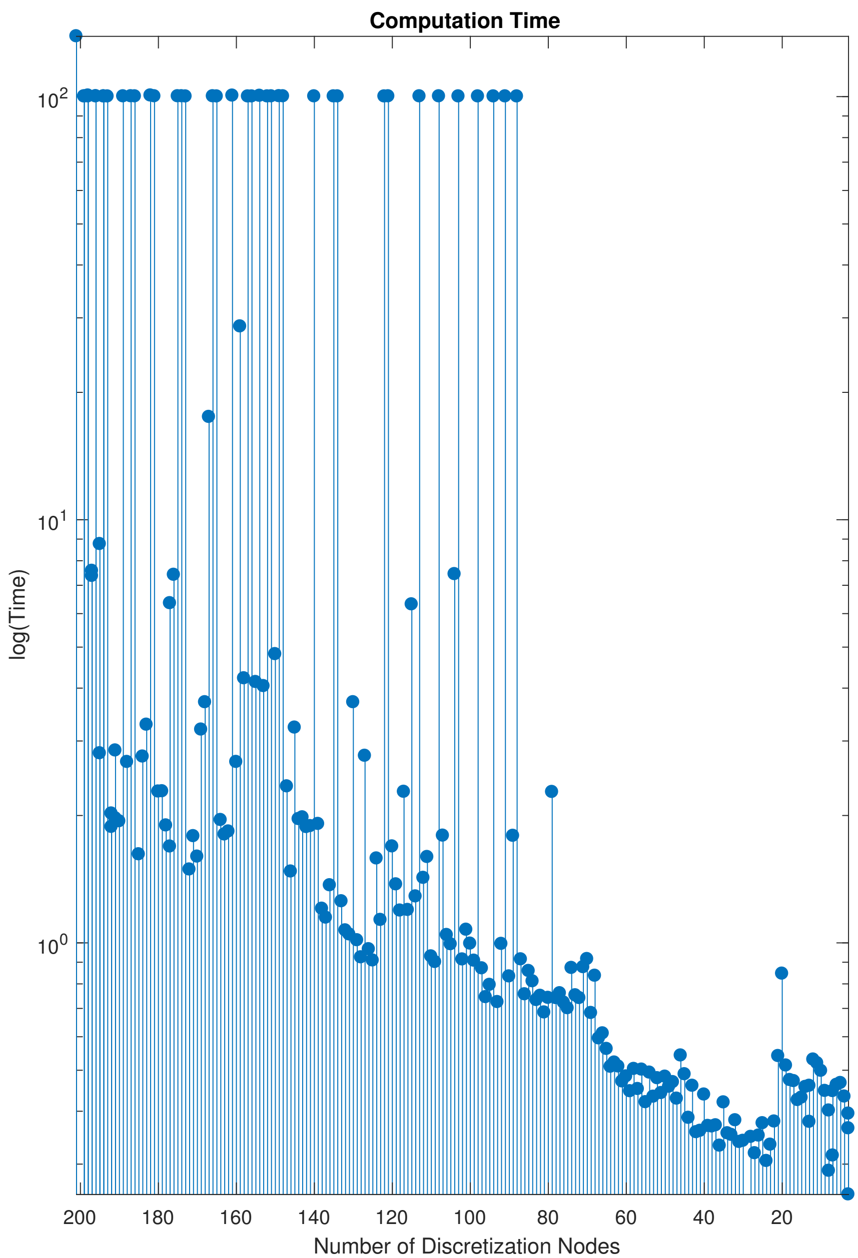

Figure 8.42: Modelling errors, overestimating the train mass, with sensor and process noise added. Minimum-time considering delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

Comparing Figs. 8.39 and 8.41, considering delays resulted in an overshoot decrease from 21.19 m to 1.15 m, and a trip time decrease from 577.64 s to 571.62 s.

8.2.7.2 Minimum-Energy Optimal Control

The objective function, Eq. (7.2.1.29), parameters are: $w_t = 0$, $w_e = 1$, and $w_{\Delta \mathrm{ctrl}} = 1$.

The results ignoring delays are shown in Figs. 8.43 and 8.44. The results considering delays are shown in Figs. 8.45 and 8.46.

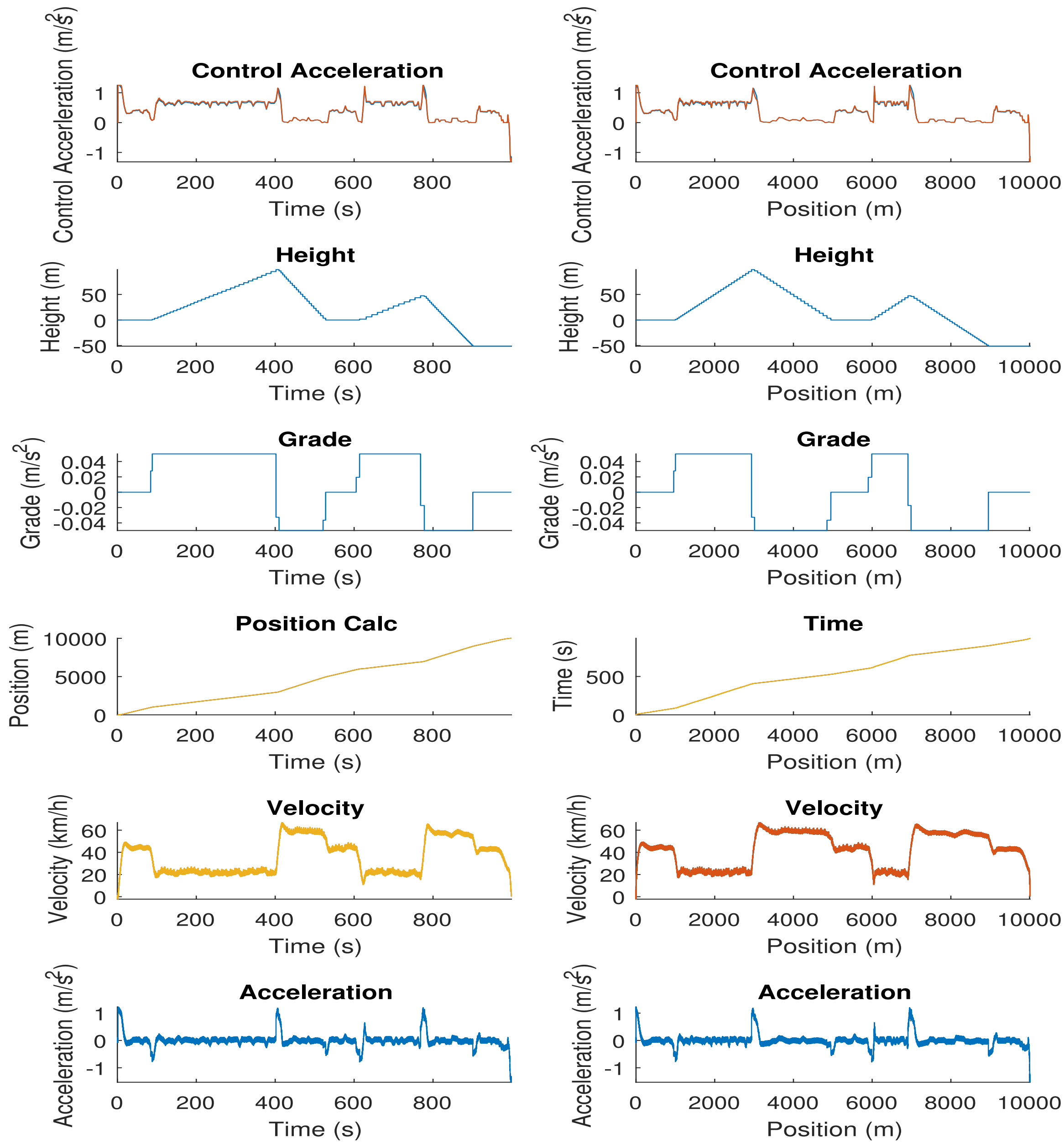

Figure 8.43: Modelling errors, overestimating the train mass, with sensor and process noise added. Minimum-energy ignoring delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10011.00 m. Overshoot: 11.00 m. Trip Time: 998.62 s. Trip Delay: N/A s. Energy Consumed: 3344.91 J.

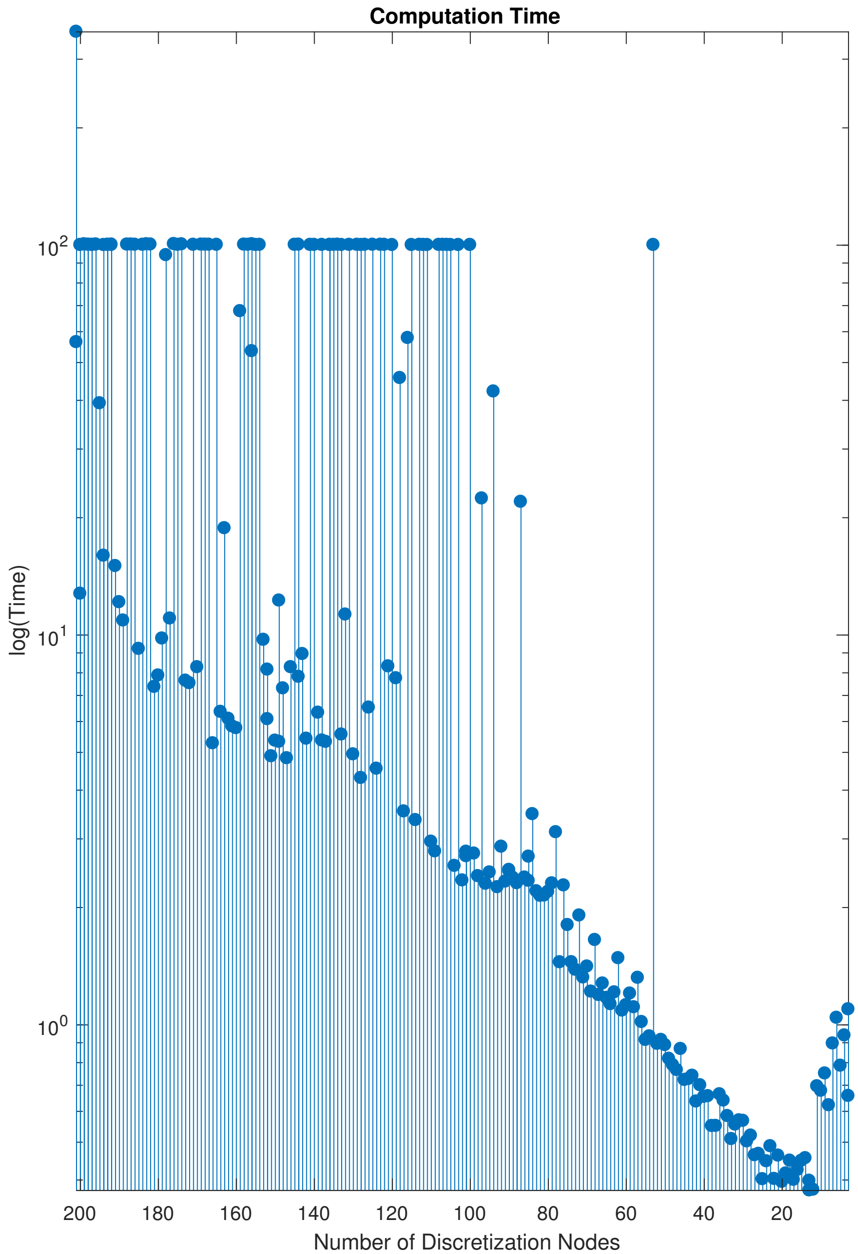

Figure 8.44: Modelling errors, overestimating the train mass, with sensor and process noise added. Minimum-energy ignoring delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

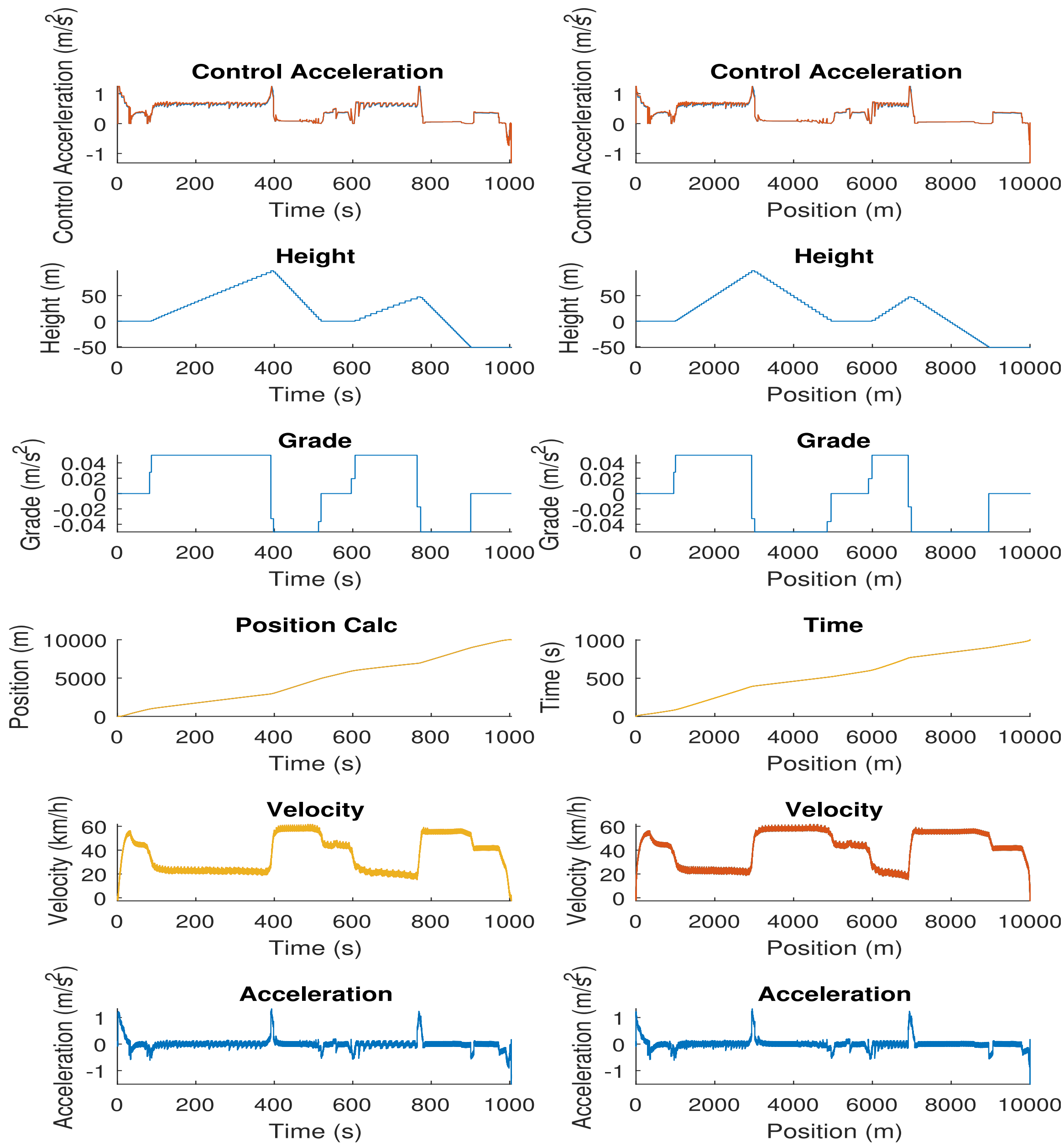

Figure 8.45: Modelling errors, overestimating the train mass, with sensor and process noise added. Minimum-energy considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10000.74 m. Overshoot: 0.74 m. Trip Time: 1004.36 s. Trip Delay: 4.36 s. Energy Consumed: 3271.07 J.

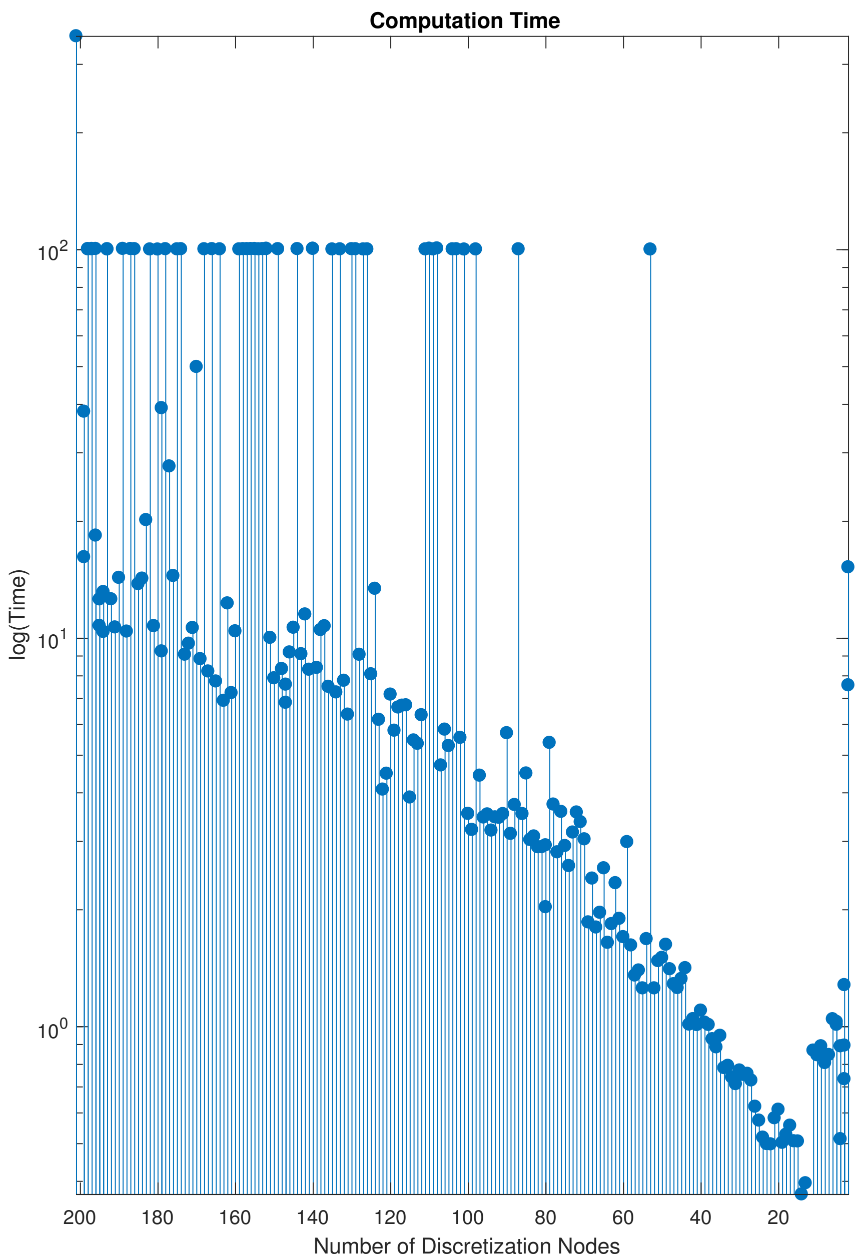

Figure 8.46: Modelling errors, overestimating the train mass, with sensor and process noise added. Minimum-energy considering delays in the control input; computation times. $T_{\text{trk,delay}} = 1.4\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

Comparing Figs. 8.43 and 8.45, considering delays resulted in an overshoot decrease from 11.00 m to 0.74 m, an energy consumption decrease from 3344.91 J to 3271.07 J, but a trip time increase from 998.62 s to 1004.36 s. There is a trip delay of 4.36 s for the case of considering delays.

The reason why the overshoot is lower in this case vs the case of underestimating the mass, Subsection 8.2.5, is because when overestimating the mass the controller assumes that the train has a greater amount of kinetic energy, and thus the controller requests more braking than if the controller was configured with the actual mass parameter value. This higher braking force request decreases the stopping distance.

8.2.8 Test Problem 5-A: Equal Traction-Braking Delays with Disturbances and Modelling Error (Discrete-Position)

The convex optimization formulation presented in Eq. (5.2.1) is used here to solve the problem considering equal traction-braking delays. It can be assumed that an artificial delay has been added to the traction command to make the two delays equal compared to Subsection 8.2.7. The formulation parameters are listed in Subsection 8.2.2.

The delay values are:

- $T_{\text{trk,delay}} = 2.8\,\text{s}$
- $T_{\text{brk,delay}} = 2.8\,\text{s}$

8.2.8.1 Minimum-Time Optimal Control

The objective function, Eq. (7.2.1.29), parameters are: $w_\rho = 1,000$, and $w_e = 0$.

The results considering delays are shown in Figs. 8.47 and 8.48.

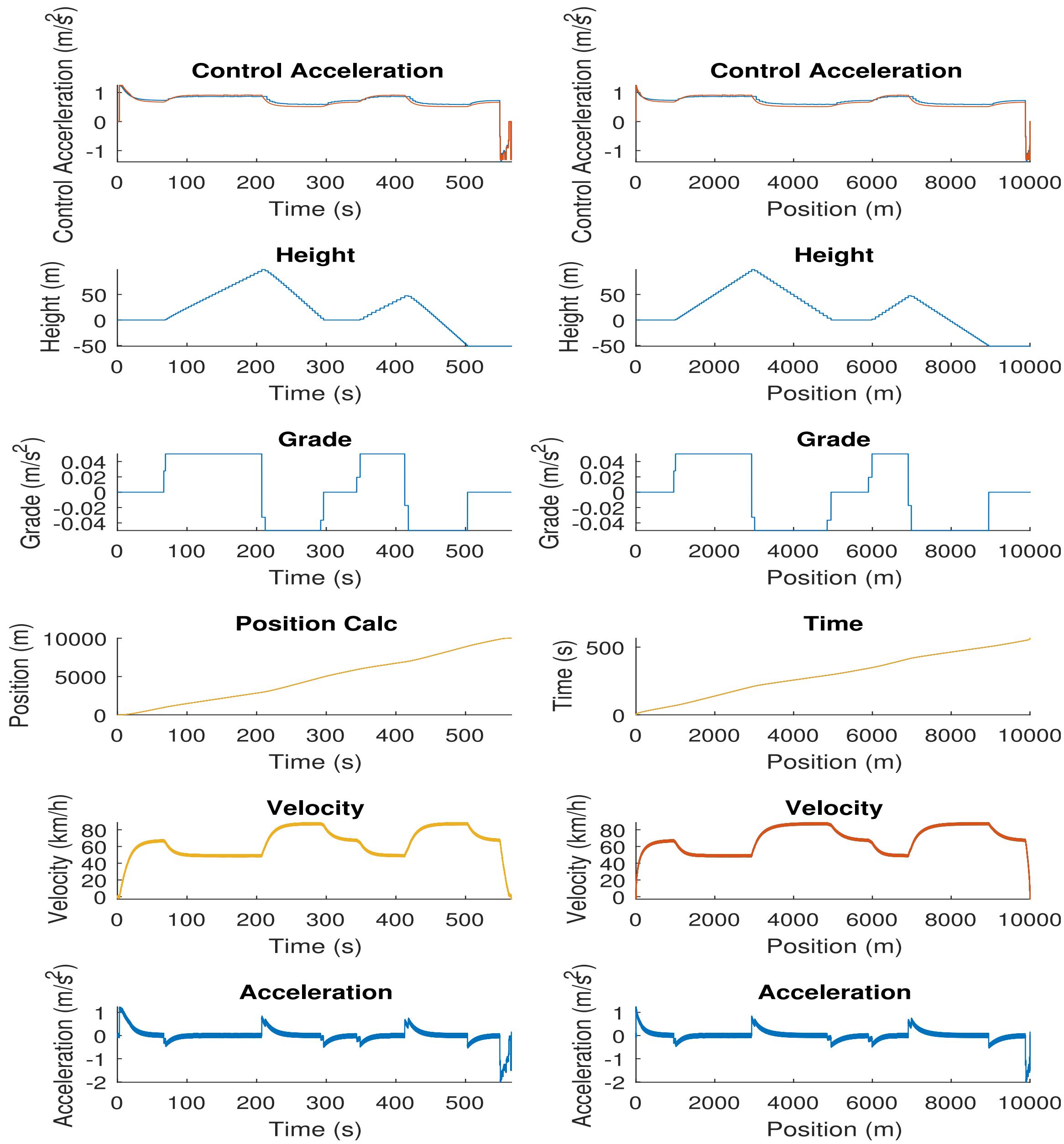

Figure 8.47: Modelling errors, overestimating the train mass, with sensor and process noise added. Convex formulation: minimum-time considering delays in the control input; closed-loop response. $T_{\text{trk,delay}} = 2.8\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$. Final Position: 10001.34 m. Overshoot: 1.34 m. Trip Time: 565.88 s. Trip Delay: N/A s. Energy Consumed: 6411.33 J.

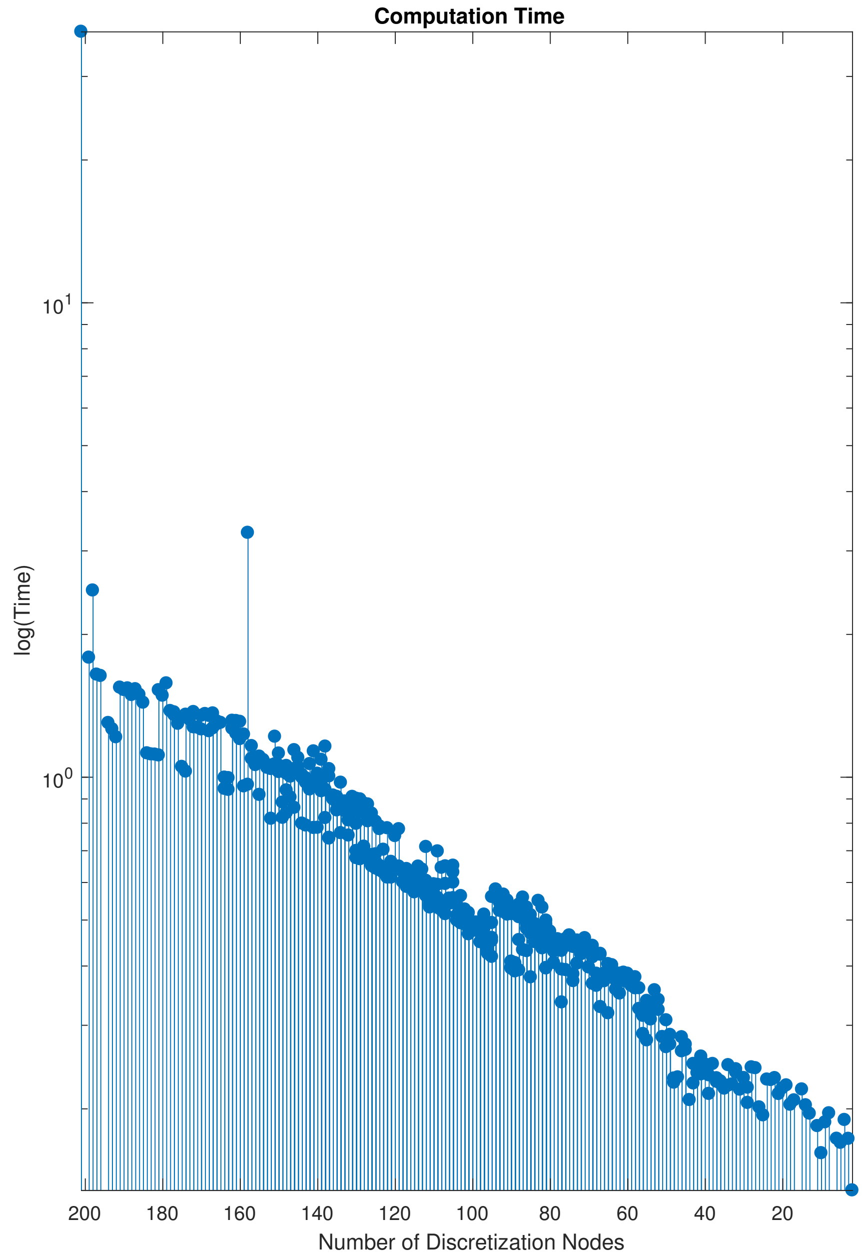

Figure 8.48: Modelling errors, overestimating the train mass, with sensor and process noise added. Convex formulation: minimum-time considering delays in the control input; computation times. $T_{\text{trk,delay}} = 2.8\,\text{s}$, $T_{\text{brk,delay}} = 2.8\,\text{s}$.

As can be seen from Fig. 8.47, the controller is quite robust when dealing with noise and modelling errors. The overshoot in the convex case is 1.34 m, Fig. 8.47, while the overshoot in the nonconvex case (considering delays) is 1.15 m, Fig. 8.41. The trip time in the convex case is 565.88 s, while in the nonconvex case is 571.62 s. The performance is better in the convex case as opposed to the nonconvex case, but this can be explained by the same reasons as in Subsection 8.2.6. The difference in the overshoot, 0.19 m, is not significant.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

Optimal control of a commuter train, a nonlinear system, under the presence of traction and braking delays was studied. Solutions were proposed for the two cases of equal and non-equal traction-braking delays. In the case of equal traction-braking delays, a convex optimization model was presented that employed a model of the system in the discrete position domain resulting in a fast global solution to the problem of mixed energy-time optimal control. For the case of non-equal traction-braking delays, a nonconvex optimization model using a model of the system in the discrete time domain was developed. The control formulations are quite general and provide a mechanism to achieve optimal mixed energy-time objectives under various user-specified operational constraints. The case of non-equal traction-braking delays is quite general and can be extended to allow for optimal control of any nonlinear system with an arbitrary number of inputs with different delays.

Two new optimal controllers are proposed that can compensate for delays in traction and braking for commuter train operation. The commuter train system is modelled as a nonlinear system. The controllers are model-based, they explicitly incorporate the model information and the trip parameters. The controllers use an EMPC framework where the control problem is solved repeatedly along the track using the latest sensor measurements. Given a train at any position along the track, the states, time, position, and velocity, of the train are measured, an optimal control problem is solved to determine the state and control trajectory from the current position to the end of the trip. From the given solution of the optimal control problem, the first few samples of the control are applied and the process is repeated until the train arrives at the destination. The optimal control problem is formulated as an optimization problem and solved using optimization solvers. Delay compensation is achieved by the use of a predictor, i.e. model-based prediction. The optimal control formulation in the case of equal traction-braking delays is formulated in the position domain using an existing convex approximation. The optimization problem being convex allows for rapid and robust computation of a global optimum. The non-equal traction-braking delay case is made tractable with the assumption that the input is piecewise constant. A predictor is used to compensate for the "common" portion of the delay. An optimization problem is solved from the end of prediction till the end of the trip. This optimization problem consists of two parts, one part where the independent variable is time, and the other part where the independent variable is an affine function of time. The affine function maps an arbitrary time interval to an interval from zero to one. Over this interval, the length of time can be expressed as an affine function of this alternative independent variable. For the control input with the longer delay, the control samples at the end of prediction till the end of the issued control samples serve as constraints on that respective control input.

Results and simulations are presented demonstrating the feasibility of the proposed formulation. Results of numerical simulations demonstrated that the proposed control methodologies are quite effective in moving the train from origin to destination according to the operator requirements. The main advantage of the convex formulation is that it can be quickly solved to find a globally optimal solution to the problem, whereas the nonconvex formulation has a much longer computation time and may not necessarily produce a globally optimal solution. However, the results of the simulations also revealed that adding artificial delay to one of the control channels so that the convex formulation can be employed can be a very effective strategy for solving the non-equal delay problem, when real-time computation time is of concern.

9.2 Future Work

Some possible future directions for research to improve upon the proposed controllers are:

- Parameter estimation: Parameter estimation can be used to allow for the controller to automatically update the estimates of the model parameters in real-time. This will allow for more accurate estimation of the model parameters, thus reducing errors due to the use of inaccurate parameter values in the mathematical model.

  This can also assist when parameter values change over periods of time, due to changing environmental conditions, wear & tear, aging, etc. Some examples of conditions that result in parameter value changes are provided below.

  - Train mass can change from station to station as passengers board and disembark from the train.
  - Train resistance forces can change as track conditions change.
    * Aerodynamic resistance forces vary when the train is travelling inside a tunnel versus travelling outside a tunnel.
    * Rain and temperature can change the rolling resistance experienced by the train due to changing rail track conditions.

  Parameter estimation can also include estimation of input-delays. The controller can be extended to automatically estimate delays in traction and braking. Estimation of the delays can result in more accurate delay values, thus reducing errors from the use of inaccurate parameter values.

  Adjusting to different parameter values can make the controller adaptive. The controller can become more portable as it can be installed on different trains with less configuration and tuning required. The controller will automatically estimate the parameters and utilize them for the purposes of control.

- State Estimation: Another research direction would be to use more accurate sensor models, including delays and uncertainties in sensor measurements, in the optimal control problem. Currently, it is assumed, that the exact state values can be measured; however, this is usually not the case in real-life. In real-life, the full state measurement is usually not available, sensor measurements are not free of errors, and sensors can be subject to delays. An optimal estimator can be designed to provide improved state estimates. The optimal estimator can be coupled with an optimal controller to enable both optimal observation and optimal action based upon that observation.

- Robust Control: Robust control can be used to allow for increased robustness to disturbances and noise. For example, the grade may not be known exactly, and there may also be errors in model parameters such as resistance coefficients and train mass. This can also include robustness to using an inaccurate value of the delay in the system model. Delays in actuation can severely degrade performance and may also cause instability. The proposed controllers in this thesis have not been tested for robustness to inaccurate traction or braking delay values.

- State-varying delays in traction and braking: A challenging future research direction is extending the input-delay model in the optimal control formulation from constant delays in input to state-varying delays in input. The delays in traction and braking would now become functions of the state. This is a more realistic model, because the train actuation mechanism can have different delays in traction and braking when the train is in different modes of operation: rest mode, traction mode, and braking mode.

The above list is by no means complete or comprehensive, but rather a limited list of possible related future research directions that are of interest.

Appendix A

Convex Formulation of Optimal Mixed Energy-Time Train Control

This appendix provides a brief overview of the convex approximation detailed in [13]. The original train optimal control problem formulated as an optimization problem in the position domain, i.e. before the convex approximation is applied, is as follows.

$$\min_{t[k],v[k],u[k],k\in\{q,\ldots,N\}} w_e \left\{ \sum_{i=q}^{N-1} u^+ [i]\, \delta s [i] \right\} + w_t t [N] \tag{A.0.1.1}$$

s.t.

$$t [q] = t_{\text{Predicted}} \tag{A.0.1.2}$$

$$s [q] = s_{\text{Predicted}} \tag{A.0.1.3}$$

$$v [q] = v_{\text{Predicted}} \tag{A.0.1.4}$$

$$t_{\min} \leq t [N] \leq t_{\max} \tag{A.0.1.5}$$

$$s [N] = s_{\text{f}} \tag{A.0.1.6}$$

$$v[N] = 0\,\mathrm{m/s} \tag{A.0.1.7}$$

for $k \in \{q, \ldots, N-1\}$:

$$t[k+1] = t[k] + \frac{1}{v[k]}\delta s[k] \tag{A.0.1.8}$$

$$v[k+1] = v[k] + \left\{ \frac{(g[k] - C_0)}{mv[k]} - \frac{C_v}{m} - \frac{C_{v^2}}{m} v[k] + \frac{u[k]}{v[k]} \right\} \delta s[k] \tag{A.0.1.9}$$

for $k \in \{q, \ldots, N\}$:

$$0 \leq u^+[k] \tag{A.0.1.10}$$

$$u[k] \leq u^+[k] \tag{A.0.1.11}$$

$$0\,\mathrm{m/s} < v[k] \leq v_{\mathrm{max}}[k] \tag{A.0.1.12}$$

$$u_{\mathrm{brk,max}} \leq u[k] \tag{A.0.1.13}$$

$$u[k] \leq \min\left\{ U_{\mathrm{trk,max}}, \frac{P}{v}, \frac{k_P P}{v^2} \right\} \tag{A.0.1.14}$$

With a substitution of variable of $\rho[k] = \frac{1}{v[k]}$, and $\gamma[k] = \frac{u[k]}{v[k]}$, we arrive at the following dynamics equations:

$$t[k+1] = t[k] + \rho[k]\,\delta s[k] \tag{A.0.2}$$

$$v[k+1] = v[k] + \left\{ \frac{(g[k] - C_0)\,\rho[k]}{m} - \frac{C_v}{m} - \frac{C_{v^2}}{m} v[k] + \gamma[k] \right\} \delta s[k] \tag{A.0.3}$$

And we also arrive at the following algebraic equation:

$$\frac{1}{v[k]} = \rho[k] \tag{A.0.4}$$

Note that, $\rho \geq 0$, $v \geq 0$, and that the control variable to be selected is now $\gamma[k]$ instead of $u[k]$.

Equation (A.0.4) is relaxed from the equality constraint to the inequality constraint:

$$\frac{1}{v[k]} \leq \rho[k] \tag{A.0.5}$$

Equation (A.0.5) is also known as an hyperbolic constraint, and in its current form is nonconvex, but can be transformed into the following constraint that is convex [93]:

$$\left\| \begin{bmatrix} 2 \\ v[k] - \rho[k] \end{bmatrix} \right\|_2 \leq v[k] + \rho[k] \tag{A.0.6}$$

To ensure that $\rho[k] \approx \frac{1}{v[k]}$, the term $\rho[k]$ is included in the objective function with a significant enough weight to ensure that the inequality is active. The optimal control problem is now transformed to the disciplined convex form [71], [94]–[97], thus ensuring convexity.

The dynamics are now characterized by the difference equations, Eqs. (A.0.2) and (A.0.3), and the algebraic equation, Eq. (A.0.6). Note, that this approximation only matches the real dynamics if Eq. (A.0.6) is active.

$$u_{\text{brk,max}} \leq u \leq \min\left\{U_{\text{trk,max}}, \frac{P}{v}, \frac{k_P P}{v^2}\right\} \tag{A.0.7}$$

The constraints on the control, Eq. (A.0.7), must be approximated by convex functions so that the optimization problem is convex, see [13]. Thus, the resulting convex optimization formulation is now:

$$\min_{t[k],v[k],\rho[k],\gamma[k],\gamma^+[k],k\in\{q,\ldots,N\}} \quad \sum_{i=q}^{N-1} \left\{ w_e \gamma^+ [i]\, \delta s [i] + w_\rho \rho [i] \right\} \tag{A.0.8.1}$$

s.t.

$$t[q] = t_{\text{Predicted}} \tag{A.0.8.2}$$

$$s[q] = s_{\text{Predicted}} \tag{A.0.8.3}$$

$$v[q] = v_{\text{Predicted}} \tag{A.0.8.4}$$

$$t_{\text{min}} \leq t[N] \leq t_{\text{max}} \tag{A.0.8.5}$$

$$s[N] = s_{\text{f}} \tag{A.0.8.6}$$

$$v[N] = 0\,\text{m/s} \tag{A.0.8.7}$$

for $k \in \{q, \ldots, N-1\}$:

$$t[k+1] = t[k] + \rho[k]\, \delta s[k] \tag{A.0.8.8}$$

$$v[k+1] = v[k] + \left\{ g[k] - \frac{C_0}{m} \rho[k] - \frac{C_v}{m} - \frac{C_{v^2}}{m} v[k] + \gamma[k] \right\} \delta s[k] \tag{A.0.8.9}$$

for $k \in \{q, \ldots, N\}$:

$$\left\| \begin{bmatrix} 2 \\ v[k] - \rho[k] \end{bmatrix} \right\|_2 \leq v[k] + \rho[k] \tag{A.0.8.10}$$

$$0 \leq \gamma^+[k] \tag{A.0.8.11}$$

$$\gamma[k] \leq \gamma^+[k] \tag{A.0.8.12}$$

$$0\,\mathrm{m/s} < v\,[k] \leq v_{\max}\,[k] \tag{A.0.8.13}$$

$$0\,\mathrm{s/m} < \rho\,[k] < \infty\,\mathrm{s/m} \tag{A.0.8.14}$$

$$\left(u_{\mathrm{brk,max}}\right)\rho\,[k] \leq \gamma\,[k] \tag{A.0.8.15}$$

$$\gamma\,[k] \leq \left(u_{\mathrm{trk,max}}\right)\rho\,[k] \tag{A.0.8.16}$$

$$\gamma\,[k] \leq r_0\rho\,[k] + r_v + r_{v^2}v\,[k] \tag{A.0.8.17}$$

The optimal control problem is now transformed to the disciplined convex form [71], [94]–[97], thus ensuring convexity.

Bibliography

[1] ITF, "Itf transport outlook 2015," Paris, 2015. DOI: http://dx.doi.org/10.1787/9789282107782-en. [Online]. Available: /content/book/9789282107782-en.

[2] S. D. S.C. Davis and R. Boundy, *Transportation Energy Data Book*, 35th ed. Oak Ridge, TN: Oak Ridge National Laboratory, Oct. 31, 2016. DOI: ORNL-6992. [Online]. Available: http://cta.ornl.gov/data/tedb35/Edition35_Full_Doc.pdf.

[3] A. Albrecht, P. Howlett, P. Pudney, X. Vu, and P. Zhou, "The key principles of optimal train control—part 1: Formulation of the model, strategies of optimal type, evolutionary lines, location of optimal switching points," *Transportation Research Part B: Methodological*, vol. 94, pp. 482–508, 2016, ISSN: 0191-2615. DOI: http://dx.doi.org/10.1016/j.trb.2015.07.023. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0191261515002064.

[4] M. Krstic, "Introduction," in *Delay Compensation for Nonlinear, Adaptive, and PDE Systems*. Boston: Birkhäuser Boston, 2009, pp. 1–13, ISBN: 978-0-8176-4877-0. DOI: 10.1007/978-0-8176-4877-0_1. [Online]. Available: http://dx.doi.org/10.1007/978-0-8176-4877-0_1.

[5] I. Karafyllis and M. Krstic, "Preview of predictor feedback and delay compensation," in *Predictor Feedback for Delay Systems: Implementations and Approximations*. Cham: Springer International Publishing, 2017, pp. 1–16, ISBN: 978-3-319-42378-4. DOI: 10.1007/978-3-319-42378-4_1. [Online]. Available: http://doi.org/10.1007/978-3-319-42378-4_1.

[6] B. Zhou, "Stabilization of linear time-delay systems by higher-order tpf," in *Truncated Predictor Feedback for Time-Delay Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 179–198, ISBN: 978-3-642-54206-0. DOI: 10.1007/978-3-642-54206-0_7. [Online]. Available: http://doi.org/10.1007/978-3-642-54206-0_7.

[7] W. Kwon and A. Pearson, "Feedback stabilization of linear systems with delayed control," *IEEE Transactions on Automatic Control*, vol. 25, no. 2, pp. 266–269, Apr. 1980, ISSN: 0018-9286. DOI: 10.1109/TAC.1980.1102288.

[8] A. Manitius and A. Olbrot, "Finite spectrum assignment problem for systems with delays," *IEEE Transactions on Automatic Control*, vol. 24, no. 4, pp. 541–552, Aug. 1979, ISSN: 0018-9286. DOI: 10.1109/TAC.1979.1102124.

[9] Z. Artstein, "Linear systems with delayed controls: A reduction," *IEEE Transactions on Automatic Control*, vol. 27, no. 4, pp. 869–879, Aug. 1982, ISSN: 0018-9286. DOI: 10.1109/TAC.1982.1103023.

[10] M. Krstic, "Input delay compensation for forward complete and strict-feedforward nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 287–303, Feb. 2010, ISSN: 0018-9286. DOI: 10.1109/TAC.2009.2034923.

[11] M. Krstic and D. Bresch-Pietri, "Delay-adaptive full-state predictor feedback for systems with unknown long actuator delay," in *2009 American Control Conference*, Jun. 2009, pp. 4500–4505. DOI: 10.1109/ACC.2009.5159839.

[12] D. Yazhemsky, "A real-time capable adaptive optimal controller for a commuter train," May 19, 2017. [Online]. Available: http://hdl.handle.net/11375/21469.

[13] D. Yazhemsky, M. Rashid, and S. Sirouspour, "An on-line optimal controller for a commuter train," *IEEE Intelligent Transportation Systems Transactions*, Submitted.

[14] M. Rashid, D. Yazhemsky, and S. Sirouspour, "Optimal control of a commuter train with traction and braking time delays," *IEEE Intelligent Transportation Systems Transactions*, To be submitted.

[15] G. T. Huntington and A. V. Rao, "A comparison between global and local orthogonal collocation methods for solving optimal control problems," in *2007 American Control Conference*, Jul. 2007, pp. 1950–1957. DOI: 10.1109/ACC.2007.4282393.

[16] I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: From theory to flight," *Annual Reviews in Control*, vol. 36, no. 2, pp. 182–197, 2012, ISSN: 1367-5788. DOI: http://dx.doi.org/10.1016/j.arcontrol.2012.09.

002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367578812000375.

[17] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.

[18] W. Li, J. Lu, H. Zhou, and S.-N. Chow, "Method of evolving junctions: A new approach to optimal control with constraints," *Automatica*, vol. 78, pp. 72–78, 2017, ISSN: 0005-1098. DOI: http://doi.org/10.1016/j.automatica.2016.12.023. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109816305313.

[19] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast Motions in Biomechanics and Robotics, Optimization and Feedback Control*, ser. Lecture Notes in Control and Information Sciences, vol. 340, Heidelberg, Allemagne: Springer Berlin Heidelberg, 2006, pp. 65–93, ISBN: 9783540361183. DOI: 10.1007/978-3-540-36119-0. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-36119-0.

[20] F. L. Lewis, D. Vrabie, and V. L. Syrmos, "Dynamic programming," in *Optimal Control*, 3rd ed., ser. A Wiley-interscience publication. Wiley, Jan. 2012, ch. 6, pp. 260–286, ISBN: 9780470633496.

[21] ——, "Optimal control of continuous-time systems," in *Optimal Control*, 3rd ed., ser. A Wiley-interscience publication. Wiley, Jan. 2012, ch. 3, pp. 110–176, ISBN: 9780470633496.

[22] J. T. Betts, "3. optimal control preliminaries," in *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial &

Applied Mathematics, 2010, pp. 91–121. DOI: 10.1137/1.9780898718577.ch3. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9780898718577.ch3. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9780898718577.ch3.

[23] E. F. Camacho and C. Bordons, "Introduction to model predictive control," in *Model Predictive control*. London: Springer London, 2007, pp. 1–11, ISBN: 978-0-85729-398-5. DOI: 10.1007/978-0-85729-398-5_1. [Online]. Available: http://dx.doi.org/10.1007/978-0-85729-398-5_1.

[24] L. Grüne and J. Pannek, "Introduction," in *Nonlinear Model Predictive Control: Theory and Algorithms*. Cham: Springer International Publishing, 2017, pp. 1–11, ISBN: 978-3-319-46024-6. DOI: 10.1007/978-3-319-46024-6_1. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46024-6_1.

[25] D. J. Griffin, M. A. Grover, Y. Kawajiri, and R. W. Rousseau, "Data-driven modeling and dynamic programming applied to batch cooling crystallization," *Industrial & Engineering Chemistry Research*, vol. 55, no. 5, pp. 1361–1372, 2016. DOI: 10.1021/acs.iecr.5b03635. eprint: http://dx.doi.org/10.1021/acs.iecr.5b03635. [Online]. Available: http://dx.doi.org/10.1021/acs.iecr.5b03635.

[26] M. Ellis, J. Liu, and P. D. Christofides, "Introduction," in *Economic Model Predictive Control: Theory, Formulations and Chemical Process Applications*. Cham: Springer International Publishing, 2017, pp. 1–19, ISBN: 978-3-319-41108-8. DOI: 10.1007/978-3-319-41108-8_1. [Online]. Available: https://doi.org/10.1007/978-3-319-41108-8_1.

[27] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000, ISSN: 0005-1098. DOI: http://dx.doi.org/10.1016/S0005-1098(99)00214-9. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109899002149.

[28] "Introduction," in *Receding Horizon Control: Model Predictive Control for State Models*. London: Springer London, 2005, pp. 1–15, ISBN: 978-1-84628-017-7. DOI: 10.1007/1-84628-017-6_1. [Online]. Available: http://doi.org/10.1007/1-84628-017-6_1.

[29] "Ieee standard for communications-based train control (cbtc) performance and functional requirements," *IEEE Std 1474.1-2004 (Revision of IEEE Std 1474.1-1999)*, pp. 1–45, 2004. DOI: 10.1109/IEEESTD.2004.95746.

[30] J. Greenway and R. Sheldon, "Automatic train control and communications for washington metro," *Communications Society*, vol. 12, no. 6, pp. 14–21, Nov. 1974, ISSN: 0094-5579. DOI: 10.1109/MCOMD.1974.1088881.

[31] J. Yin, D. Chen, and Y. Li, "Smart train operation algorithms based on expert knowledge and ensemble cart for the electric locomotive," *Knowledge-Based Systems*, vol. 92, pp. 78–91, 2016, ISSN: 0950-7051. DOI: http://dx.doi.org/10.1016/j.knosys.2015.10.016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705115004037.

[32] H. Oshima, S. Yasunobu, and S. i. Sekino, "Automatic train operation system based on predictive fuzzy control," in *Proceedings of the International Workshop on Artificial Intelligence for Industrial Applications*, May 1988, pp. 485–489. DOI: 10.1109/AIIA.1988.13336.

[33] L. Zhang and X. Zhuan, “Optimal operation of heavy-haul trains equipped with electronically controlled pneumatic brake systems using model predictive control methodology,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 1, pp. 13–22, Jan. 2014, ISSN: 1063-6536. DOI: 10.1109/TCST.2013.2238235.

[34] ——, “Development of an optimal operation approach in the mpc framework for heavy-haul trains,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1391–1400, Jun. 2015, ISSN: 1524-9050. DOI: 10.1109/TITS.2014.2364178.

[35] S. Li, B. D. Schutter, L. Yang, and Z. Gao, “Robust model predictive control for train regulation in underground railway transportation,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 1075–1083, May 2016, ISSN: 1063-6536. DOI: 10.1109/TCST.2015.2480839.

[36] Y. Luo, D. Li, and Y. Xi, “Model predictive control with swithing strategy for a train system,” in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, Dec. 2014, pp. 913–918. DOI: 10.1109/ICARCV.2014.7064426.

[37] C. S. Chang, D. Y. Xu, and H. B. Quek, “Pareto-optimal set based multiobjective tuning of fuzzy automatic train operation for mass transit system,” *IEE Proceedings - Electric Power Applications*, vol. 146, no. 5, pp. 577–583, Sep. 1999, ISSN: 1350-2352. DOI: 10.1049/ip-epa:19990481.

[38] C. S. Chang and D. Y. Xu, “Differential evolution based tuning of fuzzy automatic train operation for mass rapid transit system,” *IEE Proceedings - Electric*

Power Applications, vol. 147, no. 3, pp. 206–212, May 2000, ISSN: 1350-2352. DOI: 10.1049/ip-epa:20000329.

[39] S. Yasunobu, S. Miyamoto, and H. Ihara, "Fuzzy control for automatic train operation system," *IFAC Proceedings Volumes*, vol. 16, no. 4, pp. 33–39, 1983, 4th IFAC/IFIP/IFORS Conference on Control in Transportation Systems , Baden-Baden, FRG, 20-22 April 1983, ISSN: 1474-6670. DOI: http://dx.doi.org/10.1016/S1474-6670(17)62539-4. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667017625394.

[40] Y. Bai, T. K. Ho, B. Mao, Y. Ding, and S. Chen, "Energy-efficient locomotive operation for chinese mainline railways by fuzzy predictive control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 938–948, Jun. 2014, ISSN: 1524-9050. DOI: 10.1109/TITS.2013.2292712.

[41] Y. Xu, X. Zhao, L. Wang, X. Liu, and Q. Zhang, "Optimal control of automatic train operation based on multi-scale dynamic programming," in *Proceedings of the 33rd Chinese Control Conference*, Jul. 2014, pp. 3429–3433. DOI: 10.1109/ChiCC.2014.6895507.

[42] L.-M. Jia, X.-D. Zhang, and Z.-T. Xie, "Automatic train control-an intelligent approach," in *TENCON '93. Proceedings. Computer, Communication, Control and Power Engineering.1993 IEEE Region 10 Conference on*, vol. 4, Oct. 1993, 338–342 vol.4. DOI: 10.1109/TENCON.1993.320501.

[43] S. Sekine, N. Imasaki, and T. Endo, "Application of fuzzy neural network control to automatic train operation and tuning of its control rules," in *Proceedings of 1995 IEEE International Conference on Fuzzy Systems.*, vol. 4, Mar. 1995, 1741–1746 vol.4. DOI: 10.1109/FUZZY.1995.409917.

[44] S. Sekine and M. Nishimura, "Application of fuzzy neural network control to automatic train operation," in *Proceedings of 1995 IEEE International Conference on Fuzzy Systems.*, vol. 5, Mar. 1995, 39–40 vol.5. DOI: 10.1109/FUZZY.1995.410032.

[45] J. H. Holmes, "Chapter 7 - knowledge discovery in biomedical data: Theory and methods," in *Methods in Biomedical Informatics*, I. N. Sarkar, Ed., Oxford: Academic Press, 2014, pp. 179–240, ISBN: 978-0-12-401678-1. DOI: https://doi.org/10.1016/B978-0-12-401678-1.00007-5. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780124016781000075.

[46] A. Albrecht, P. Howlett, P. Pudney, X. Vu, and P. Zhou, "The key principles of optimal train control—part 2: Existence of an optimal strategy, the local energy minimization principle, uniqueness, computational techniques," *Transportation Research Part B: Methodological*, vol. 94, pp. 509–538, 2016, ISSN: 0191-2615. DOI: http://dx.doi.org/10.1016/j.trb.2015.07.024. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0191261515002076.

[47] E. Khmelnitsky, "On an optimal control problem of train operation," *IEEE Transactions on Automatic Control*, vol. 45, no. 7, pp. 1257–1266, Jul. 2000, ISSN: 0018-9286. DOI: 10.1109/9.867018.

[48] R. (Liu and I. M. Golovitcher, "Energy-efficient operation of rail vehicles," *Transportation Research Part A: Policy and Practice*, vol. 37, no. 10, pp. 917–932, 2003, ISSN: 0965-8564. DOI: http://dx.doi.org/10.1016/j.tra.2003.07.001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965856403000752.

[49] P. G. Howlett and P. J. Pudney, *Energy-Efficient Train Control*, First.: Springer-Verlag London, 1995. DOI: 10.1007/978-1-4471-3084-0. eprint: http://dx.doi.org/10.1007/978-1-4471-3084-0. [Online]. Available: http://dx.doi.org/10.1007/978-1-4471-3084-0.

[50] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985. DOI: 10.1177/027836498500400301. eprint: http://dx.doi.org/10.1177/027836498500400301. [Online]. Available: http://dx.doi.org/10.1177/027836498500400301.

[51] K. ICHIKAWA, "Application of optimization theory for bounded state variable problems to the operation of train," *Bulletin of JSME*, vol. 11, no. 47, pp. 857–865, 1968. DOI: 10.1299/jsme1958.11.857.

[52] P. Kokotovic and G. Singh, "Minimum-energy control of a traction motor," *IEEE Transactions on Automatic Control*, vol. 17, no. 1, pp. 92–95, Feb. 1972, ISSN: 0018-9286. DOI: 10.1109/TAC.1972.1099870.

[53] I. P. Milroy, "Aspects of automatic train control," Jul. 1980. [Online]. Available: http://dspace.lboro.ac.uk/2134/10613.

[54] M. Miyatake and H. Ko, "Optimization of train speed profile for minimum energy consumption," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 5, no. 3, pp. 263–269, 2010, ISSN: 1931-4981. DOI: 10.1002/tee.20528. [Online]. Available: http://dx.doi.org/10.1002/tee.20528.

[55] Y. Wang, B. D. Schutter, T. J. van den Boom, and B. Ning, "Optimal trajectory planning for trains — a pseudospectral method and a mixed integer linear programming approach," *Transportation Research Part C: Emerging Technologies*, vol. 29, pp. 97–114, 2013, ISSN: 0968-090X. DOI: http://doi.org/10.1016/j.trc.2013.01.007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X13000193.

[56] M. Grant and S. Boyd, *CVX,: Matlab software for disciplined convex programming, version 2.1*, 2012. [Online]. Available: http://web.cvxr.com/cvx/doc/intro.html#mixed-integer-problems.

[57] H. Dong, B. Ning, B. Cai, and Z. Hou, "Automatic train control system development and simulation for high-speed railways," *IEEE Circuits and Systems Magazine*, vol. 10, no. 2, pp. 6–18, May 2010, ISSN: 1531-636X. DOI: 10.1109/MCAS.2010.936782.

[58] S. H. Han, Y. S. Byen, J. H. Baek, T. K. An, S. G. Lee, and H. J. Park, "An optimal automatic train operation (ato) control using genetic algorithms (ga)," in *TENCON 99. Proceedings of the IEEE Region 10 Conference*, vol. 1, 1999, 360–362 vol.1. DOI: 10.1109/TENCON.1999.818425.

[59] I. Karafyllis and M. Krstic, "Part iii: Extensions of predictor feedback: Introduction to part iii of the book," in *Predictor Feedback for Delay Systems: Implementations and Approximations*. Cham: Springer International Publishing, 2017, pp. 227–228, ISBN: 978-3-319-42378-4. DOI: 10.1007/978-3-319-42378-4_1. [Online]. Available: http://link.springer.com/content/pdf/bfm%3A978-3-319-42378-4/3/1.pdf.

[60] B. Zhou, "Introduction," in *Truncated Predictor Feedback for Time-Delay Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1–8, ISBN: 978-3-642-54206-0. DOI: 10.1007/978-3-642-54206-0_1. [Online]. Available: https://doi.org/10.1007/978-3-642-54206-0_1.

[61] S. Mondie and W. Michiels, "Finite spectrum assignment of unstable time-delay systems with a safe implementation," *IEEE Transactions on Automatic Control*, vol. 48, no. 12, pp. 2207–2212, Dec. 2003, ISSN: 0018-9286. DOI: 10.1109/TAC.2003.820147.

[62] M. R. Matausek and A. D. Micic, "A modified smith predictor for controlling a process with an integrator and long dead-time," *IEEE Transactions on Automatic Control*, vol. 41, no. 8, pp. 1199–1203, Aug. 1996, ISSN: 0018-9286. DOI: 10.1109/9.533684.

[63] ——, "On the modified smith predictor for controlling a process with an integrator and long dead-time," *IEEE Transactions on Automatic Control*, vol. 44, no. 8, pp. 1603–1606, Aug. 1999, ISSN: 0018-9286. DOI: 10.1109/9.780433.

[64] J.-P. Richard, "Time-delay systems: An overview of some recent advances and open problems," *Automatica*, vol. 39, no. 10, pp. 1667–1694, 2003, ISSN: 0005-1098. DOI: http://doi.org/10.1016/S0005-1098(03)00167-5. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109803001675.

[65] M. Krstic, "Lyapunov stability of linear predictor feedback for time-varying input delay," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 554–559, Feb. 2010, ISSN: 0018-9286. DOI: 10.1109/TAC.2009.2038196.

[66] L. L. Chung, C. C. Lin, and K. H. Lu, "Time-delay control of structures," *Earthquake Engineering & Structural Dynamics*, vol. 24, no. 5, pp. 687–701, 1995, ISSN: 1096-9845. DOI: 10.1002/eqe.4290240506. [Online]. Available: http://dx.doi.org/10.1002/eqe.4290240506.

[67] K. J. Åström and B. Wittenmark, "Discrete-time systems," in *Computer-Controlled Systems: Theory and Design*, 3rd ed. Dover Publications, Dec. 2011, ch. 2, pp. 30–76, ISBN: 978-0-486-48613-0.

[68] M. S. Mahmoud, "Time-delay systems: Recent progress," in *Switched Time-Delay Systems: Stability and Control*. Boston, MA: Springer US, 2010, pp. 35–74, ISBN: 978-1-4419-6394-9. DOI: 10.1007/978-1-4419-6394-9_3. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-6394-9_3.

[69] L. Xie and H. Zhang, "Optimal control for systems with input/output delays," in *Control and Estimation of Systems with Input/Output Delays*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 27–51, ISBN: 978-3-540-71119-3. DOI: 10.1007/978-3-540-71119-3_3. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-71119-3_3.

[70] E. Fridman, "Discrete-time delay systems," in *Introduction to Time-Delay Systems: Analysis and Control*. Cham: Springer International Publishing, 2014, pp. 243–272, ISBN: 978-3-319-09393-2. DOI: 10.1007/978-3-319-09393-2_6. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-09393-2_6.

[71] S. Boyd and L. Vandenberghe, "Convex optimization problems," in *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004, pp. 127–214, ISBN: 978-0-521-83378-3.

[72] A. Beck, *Introduction to Nonlinear Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2014. DOI: 10.1137/1.9781611973655. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9781611973655. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611973655.

[73] S. Boyd and L. Vandenberghe, "Introduction," in *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004, pp. 1–18, ISBN: 978-0-521-83378-3.

[74] A. Ben-Tal and A. Nemirovski, "Front matter," in *Lectures on Modern Convex Optimization*, pp. i–xvi. DOI: 10.1137/1.9780898718829.fm. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9780898718829.fm. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9780898718829.fm.

[75] "Theory of constrained optimization," in *Numerical Optimization*. New York, NY: Springer New York, 2006, pp. 304–354, ISBN: 978-0-387-40065-5. DOI: 10.1007/978-0-387-40065-5_12. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-40065-5_12.

[76] S. Boyd and L. Vandenberghe, "Duality," in *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004, pp. 215–288, ISBN: 978-0-521-83378-3.

[77] A. Wä chter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006, ISSN: 1436-4646. DOI: 10.1007/s10107-004-0559-y. [Online]. Available: http://dx.doi.org/10.1007/s10107-004-0559-y.

[78] S. Boyd and L. Vandenberghe, "Interior-point methods," in *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004, pp. 561–630, ISBN: 978-0-521-83378-3.

[79] P. Howlett, I. Milroy, and P. Pudney, "Energy-efficient train control," *Control Engineering Practice*, vol. 2, no. 2, pp. 193–200, 1994, ISSN: 0967-0661. DOI: http://dx.doi.org/10.1016/0967-0661(94)90198-8. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0967066194901988.

[80] S. Iwnicki, "Longitudinal train dynamics," pp. 239–278, 2006. [Online]. Available: http://libaccess.mcmaster.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.ftcquniv.oai.acquire.cqu.edu.au.cqu.3906&site=eds-live&scope=site.

[81] K. Popp and W. Schiehlen, "Longitudinal motions," in *Ground Vehicle Dynamics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 263–275, ISBN: 978-3-540-68553-1. DOI: 10.1007/978-3-540-68553-1_8. [Online]. Available: https://doi.org/10.1007/978-3-540-68553-1_8.

[82] V. K. Garg and R. V. Dukkipati, "Chapter 9 - train dynamics," in *Dynamics of Railway Vehicle Systems*, V. K. Garg and R. V. Dukkipati, Eds., Academic Press, 1984, pp. 332–366, ISBN: 978-0-12-275950-5. DOI: https://doi.org/10.1016/B978-0-12-275950-5.50014-6. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780122759505500146.

[83] B. Jaekel and T. Albrecht, "Comparative analysis of algorithms and models for train running simulation," *Journal of Rail Transport Planning & Management*, vol. 4, no. 1-2, pp. 14–27, 2014, ISSN: 2210-9706. DOI: http://doi.

org/10.1016/j.jrtpm.2014.06.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210970614000316.

[84] P. Howlett, "The optimal control of a train," *Annals of Operations Research*, vol. 98, no. 1, pp. 65–87, 2000, ISSN: 1572-9338. DOI: 10.1023/A:1019235819716. [Online]. Available: http://dx.doi.org/10.1023/A:1019235819716.

[85] T. Lipp and S. Boyd, "Minimum-time speed optimisation over a fixed path," *International Journal of Control*, vol. 87, no. 6, pp. 1297–1311, 2014, ISSN: 0020-7179. DOI: http://dx.doi.org/10.1080/00207179.2013.875224. [Online]. Available: http://resolver.scholarsportal.info/resolve/00207179/v87i0006/1297_msooafp.

[86] R. J. Hill, "Electric railway traction. i. electric traction and dc traction motor drives," *Power Engineering Journal*, vol. 8, no. 1, pp. 47–56, Feb. 1994, ISSN: 0950-3366. DOI: 10.1049/pe:19940105.

[87] P. Bonami, M. Kilinç, and J. Linderoth, "Algorithms and software for convex mixed integer nonlinear programs," in *Mixed Integer Nonlinear Programming*, J. Lee and S. Leyffer, Eds. New York, NY: Springer New York, 2012, pp. 1–39, ISBN: 978-1-4614-1927-3. DOI: 10.1007/978-1-4614-1927-3_1. [Online]. Available: http://doi.org/10.1007/978-1-4614-1927-3_1.

[88] C. Floudas, "Mixed-integer nonlinear optimization," in *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*, ser. Topics in Chemical Engineering. Oxford University Press, 1995, pp. 109–224, ISBN: 978-0-195-35655-7. DOI: 10.1109/TAC.1980.1102288. [Online]. Available: http://books.google.ca/books?id%20=%20OhTfOjSkq18C.

[89] A. Quarteroni, R. Sacco, and F. Saleri, "Numerical integration," in *Numerical Mathematics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 379–422, ISBN: 978-3-540-49809-4. DOI: 10.1007/978-3-540-49809-4_9. [Online]. Available: https://doi.org/10.1007/978-3-540-49809-4_9.

[90] COIN-OR. (). Ipopt, COIN-OR, [Online]. Available: http://projects.coin-or.org/Ipopt.

[91] J. Currie and D. I. Wilson, "OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User," in *Foundations of Computer-Aided Process Operations,*, N. Sahinidis and J. Pinto, Eds., 8–11 January, Savannah, Georgia, USA, Jan. 2012.

[92] COIN-OR. (). Ipopt documentation, Termination, COIN-OR, [Online]. Available: http://www.coin-or.org/Ipopt/documentation/node42.html.

[93] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, no. 1, pp. 193–228, 1998, International Linear Algebra Society (ILAS) Symposium on Fast Algorithms for Control, Signals and Image Processing, ISSN: 0024-3795. DOI: http://dx.doi.org/10.1016/S0024-3795(98)10032-0. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379598100320.

[94] M. Grant, S. Boyd, and Y. Ye, "Disciplined convex programming," in *Global Optimization: From Theory to Implementation*, L. Liberti and N. Maculan, Eds. Boston, MA: Springer US, 2006, pp. 155–210, ISBN: 978-0-387-30528-8. DOI: 10.1007/0-387-30528-9_7. [Online]. Available: https://doi.org/10.1007/0-387-30528-9_7.

[95] M. Grant and S. Boyd, *CVX,: Matlab software for disciplined convex programming, version 2.1*, 2012. [Online]. Available: http://cvxr.com/cvx/doc/intro.html#what-is-disciplined-convex-programming.

[96] ——, *CVX,: Matlab software for disciplined convex programming, version 2.1*, 2012. [Online]. Available: http://cvxr.com/cvx/doc/dcp.html.

[97] M. C. Grant, "Disciplined convex programming," Dec. 2004. [Online]. Available: http://stanford.edu/~boyd/papers/pdf/mcg_thesis.pdf.