A Real-time Capable Adaptive Optimal Controller for a Commuter Train

A REAL-TIME CAPABLE ADAPTIVE OPTIMAL CONTROLLER FOR A COMMUTER TRAIN

BY

DENNIS ION YAZHEMSKY, B.Eng.

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:	A Real-time Capable Adaptive Optimal Controller for a Commuter Train
AUTHOR:	Dennis Ion Yazhemsky B.Eng., (Electrical and Biomedical Engineering) McMaster University, Hamilton, Ontario, Canada
SUPERVISOR:	Dr. Shahin Sirouspour
NUMBER OF PAGES:	xv, 124

Abstract

This research formulates and implements a novel closed-loop optimal control system that drives a train between two stations in an optimal time, energy efficient, or mixed objective manner. The optimal controller uses sensor feedback from the train and in real-time computes the most efficient control decision for the train to follow given knowledge of the track profile ahead of the train, speed restrictions and required arrival time windows. The control problem is solved both on an open track and while safely driving no closer than a fixed distance behind another locomotive. In contrast to other research in the field, this thesis achieves a real-time capable and embeddable closed-loop optimization with advanced modeling and numerical solving techniques with a non-linear optimal control problem.

This controller is first formulated as a non-convex control problem and then converted to an advanced convex second-order cone problem with the intent of using a simple numerical solver, ensuring global optimality, and improving control robustness. Convex and non-convex numerical methods of solving the control problem are investigated and closed-loop performance results with a simulated vehicle are presented under realistic modeling conditions on advanced tracks both on desktop and embedded computer architectures. It is observed that the controller is capable of robust vehicle driving in cases both with and without modeling uncertainty. The benefits of

pairing the optimal controller with a parameter estimator are demonstrated for cases where very large mismatches exists between the controller model and the simulated vehicle. Stopping performance is consistently within $\pm 25cm$ of target stations, and the worst case closed-loop optimization time was within $100ms$ for the computation of a 1000 point control horizon on an i7-6700 machine.

Acknowledgements

I would like to express my deepest appreciation to all those who have made this research possible. Many thanks to Dr. Shahin Sirouspour for his efforts in supervising this work. His support, commitment and expertise have been an invaluable asset during this research.

I am also grateful to my colleague Rashid Muzamil, as through our collaboration we developed the non-convex optimal control formulation during the early stages of this research.

A very special thanks to Thales Canada Transport Solutions (TCTS) and the Natural Sciences and Engineering Research Council of Canada (NSERC) for funding this research.

Notation and Abbreviations

SOCP	Second Order Cone Program
ATO	Automatic Train Operation
ATS	Automatic Train Supervision
ATC	Automatic Train Control
DAS	Driver Assited System
ECOS	Embedded COnic Solver
SLSQP	Sequential Least SQuares Programming
IPOPT	Interior Point OPTimizer
BFGS	Broyden-Fletcher-Goldfarb-Shanno Hessian approximation algorithm
KF	Kalman Filter
API	Application Program Interface
IP	Interior Point
PID	Proportional-Integral-Derivative controller
ω^*	The optimal value of the associated variable (the optimal solution for ω)

Table 1: Trip Parameters

$K \in \mathbb{Z} \geq 0$	Discretization size
$V_{max} \in \mathbb{R}^K \geq 0$	Max allowable speed limit vector (m/s)
$V_0 \in \mathbb{R}$	Current train speed (m/s)
$X_0 \in \mathbb{R}$	Current vehicle position (m)
$A_0 \in \mathbb{R}$	Current vehicle acceleration (m)
$V_f \in \mathbb{R}$	Speed at the arrival point ($0m/s$ for station a stop).
$G(x)$	Continuous grade acceleration (m/s^2)
$G \in \mathbb{R}^{K-1}$	Discrete grade acceleration (m/s^2)
$g = 9.8$	Gravitational Constant (m/s^2)
$X \in \mathbb{R}^K$	Discrete position vector (m)
$\Delta_x \in \mathbb{R}^{K-1} \geq 0$	Diff of the discrete position vector
$w_t, w_e \in \mathbb{R} \geq 0$	Time and energy weights for the mixed objective
$w_{brk}, w_{os} \in \mathbb{R} \geq 0$	Overbraking and overspeeding objective weights
$T_{min}, T_{max} \in \mathbb{R} \geq 0$	Min/Max limits on arrival time into a station
$T \in \mathbb{R}$	Current time into trip
$T_{start} \in \mathbb{R}$	Minimum start time in optimization
$T_{buff} \in \mathbb{R}^{K-1}$	Discrete shifted leading train time profile
$t_{lead}(x)$	Continuous leading train time profile
$X_{sep} \in \mathbb{R}$	Minimum separation distance between successive trains (m)
$\theta(x)$	Track inclination (rad)
f_{ctl}	Closed loop control frequency (Hz)
$\epsilon \in \mathbb{R} > 0$	Very small positive number
$P^* \in \mathbb{R}$	Optimal primal solution ($P^* = f(z^*)$)
$D^* \in \mathbb{R}$	Optimal dual solution
$\mathbb{K}$	Proper cone
$\mathbb{K}_{++}$	Non-negative orthant cone
$\mathbb{K}_{so}$	Second order cone
$\mathbb{K}_{exp}$	Exponential cone

Table 2: Train Parameters and Optimization Variables

$A(v, \Gamma)$	Traction and braking function (m/s^2)
$C(v)$	Motion resistance function (m/s^2)
$F_{min} \in \mathbb{R} \leq 0$	Maximum braking force
$F_{max} \in \mathbb{R} \geq 0$	Maximum propulsion force
$c_0, c_v, c_{v2} \in \mathbb{R}$	Davis equation coefficients
$r_0, r_v, r_{v2} \in \mathbb{R}$	Tractive loss quadratic equation coefficients
$\Gamma \in \mathbb{R}^{K-1}$	Control input force (N)
$u \in \mathbb{R}^{K-1}$	Control input acceleration $(u = \Gamma/m)$ (m/s^2)
$v \in \mathbb{R}^K$	Speed (m/s)
$t \in \mathbb{R}^K$	Time (m/s)
$\Delta_t \in \mathbb{R}$	Discrete time step (s)
$\rho \in \mathbb{R}^{K-1}$	Inverse speed (s/m)
$a \in \mathbb{R}^{K-1}$	Total train acceleration (m/s^2)
$\alpha \in \mathbb{R}^{K-1}$	Position domain acceleration $(1/s)$
$\gamma, \gamma+ \in \mathbb{R}^{K-1}$	Γ/v , and $\max(\Gamma/v, 0)$ respectively $(1/s)$
$c_t, c_e \in \mathbb{R}$	Arrival time cost, trip energy cost
$c_{brk}, c_{os} \in \mathbb{R}$	Brake reserve cost, overspeed cost
$f(z) \in \mathbb{R}$	Minimization Objective
$m \in \mathbb{R}$	Vehicle mass
$B_{os} \in \mathbb{R}$	Overspeed buffer size
$B_{brk} \in \mathbb{R}$	Brake reserve buffer (m/s^2)
$b \in \mathbb{R}^{K-1}$	Additional overspeed violation variable
$h \in \mathbb{R}^{K-1}$	Additional brake reserve violation variable
$z \in \mathbb{R}^{7K-6}$	Optimization variable vector
$w \in \mathbb{R}^{7K-6}$	Standard form objective weight vector
$U \in \mathbb{R}^{2K+2,7K-6}$	Standard form equality constraint matrix
$y \in \mathbb{R}^{2K+2}$	Standard form equality vector
$Q \in \mathbb{R}^{12(K-1)+3(K-1),7K-6}$	Standard form inequality matrix
$Q_{++} \in \mathbb{R}^{12(K-1),7K-6}$	Non-negative orthant matrix part of Q
$Q_{so} \in \mathbb{R}^{3(K-1),7K-6}$	Second order cone orthant matrix part of Q
$r \in \mathbb{R}^{12(K-1)+3(K-1)}$	Standard form inequality vector
$r_{++} \in \mathbb{R}^{12(K-1)}$	Non-negative orthant vector part of r
$r_{so} \in \mathbb{R}^{3(K-1)}$	Second order cone vector part of r
$s \in \mathbb{R}^{12(K-1)+3(K-1)}$	Primal slack variables
λ, μ	Dual variables in the conic optimization problem
E_f	Simulated energy used for completing a trip (J/kg)
X_s	Simulated final stopping point of the vehicle (m)
T_f	Simulated arrival time at a station (s)

Table 3: Uncertainty and Parameter Estimation

$Y \in \mathbb{R}$	Gaussian modeling uncertainty
$j \in \mathbb{Z}$	Kalman Filter estimation index (time-domain)
$n \in \mathbb{Z}$	Number of total KF estimates up to the stopping station
$E \in \mathbb{R}^n$	Priori and posteriori error covariance (KF)
$H \in \mathbb{R}$	Measurement uncertainty (KF)
$R \in \mathbb{R}$	Modeling uncertainty (KF)
$e \in \mathbb{R}$	Inverse mass (KF)
$\hat{e} \in \mathbb{R}^n$	Optimal inverse mass (KF)
$l \in \mathbb{R}^n$	Measured traction acceleration (KF)
$\tilde{y} \in \mathbb{R}^n$	Measurement residual (KF)
$S \in \mathbb{R}^n$	Residual covariance (KF)
$\chi \in \mathbb{R}^n$	Kalman Filter gain (KF)
$\hat{m} \in \mathbb{R}^n$	Optimal mass estimate (KF)
$\bar{X}$	Measured vehicle position m
$\bar{V}$	Measured vehicle velocity (from speedometer) (m/s)
$\bar{A}$	Measured vehicle acceleration (from accelerometer) (m/s^2)
σ_X^2	Position measurement noise variance (m)
σ_V^2	Velocity measurement noise variance (m/s)
σ_A^2	Acceleration measurement noise variance (m/s^2)
σ_a^2	Acceleration modelling variance (m/s^2)

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Global commuter train networks drive an increasing ridership every year and represent a rapidly growing commercial industry. In the US alone there has been a 27% ridership increase over the ten year period from 1997-2007 (U.S. Department of Transportation, 2009) and in China the government is presently spending a record 1.25% percent of its GDP on investments in new rail infrastructure (Renner *et al.*, 2015). In total, the global light rail market is estimated at a value of $180.78 billion as of 2016 with an expected growth of 2.3% by 2020 (Leenen and Wolf, 2016). Rail transportation is so prolific that the railway transportation industry has accounted for 2% of the total energy consumption in Europe alone (European Commission: Transport Statistics, 2014).

Given the ever-present and increasing need for rail commuter transit, there is great motivation for reliable, energy efficient and safe on-board control systems for passenger locomotives. Efficient train control can significantly benefit not only total energy

consumption but greatly improve timetable adherence, reduce overall vehicle CO_2 emission, and lessen wear-and-tear on vehicle drive trains (European Commission: Transport Statistics, 2014; Hofestadt, 1995).

There has been a large global push towards automating or modernizing existing light urban rail infrastructure, thus bringing forth the need for automatic train operation systems (ATO). ATO systems, contrary to manually controlled rail vehicles, allow for the added benefits of more accurate timetable adherence, safe and robust system failure detection and fast subsequent response, and the ability to minimize the energy expenditure for a given trip. There is a spectrum of ATO automation levels ranging from semi-automatic operation (STO) to fully unattended train operation (UTO) (Schifers and Hans, 2000).

A commuter train, contrary to popular belief, is quite a complicated piece of engineering that has, in practice, been shown to be a challenge to control (Novak *et al.*, 2015). For fully automated (UTO) control, an automatic train operation (ATO) controller is charged with driving the train between stations through a selection of braking and motoring commands. It must do this while remaining below a civil speed limit and ensuring arrival within a required time window at a station. Above the ATO system resides an automatic train supervision system (ATS) which monitors the ATO, computational hardware, and the vehicle itself (Schifers and Hans, 2000). Effectively, the ATS emergency stops the train when and if the vehicle strays out of safe operating conditions in both manual and driverless train operation. Together, these two systems ensure that the vehicle adheres to a timetable in a safe and efficient manner.

Conventional ATO control techniques in the railway industry mainly rely on PID

based strategies, where the ATO controller attempts to follow a prespecified speed trajectory between a set of stations (Zhang and Zhuan, 2015). Due to the lack of any process knowledge and a simple constant gain multiplier feedback arrangement, the PID is a reactive controller as opposed to a planning controller and is a compromise (Reibig, 2011). As such, this type of control inherently presents some issues and shortcomings. In recent years, there has been an increasing interest in moving away from classical control strategies to more advanced control architectures due to the need to resolve increasingly complicated control objectives (LaValle, 2006) and due to the need for safer operation (Tomlin *et al.*, 1998). This resulted in the general move away from classical control strategy and towards optimal strategies for rail vehicle control.

As compared to classical control, optimal control techniques work on the altogether different principle of actively and intelligently re-designing the entire trip (for an entire control horizon) during every application cycle, whereas classical control strategies largely rely on combinations of well tuned ad-hoc strategies for decent feedback control of simple control objectives (LaValle, 2006). Therefore, for an optimal controller the present and all subsequent control decisions are the most efficient possible given a set cost criteria and problem constraints, whereas a classical controller is only as good as its tuning and is far from optimal for advanced control objectives. Additionally, the added benefits of an optimal control strategy also allows for advanced constraining of state and input limits for an entire control horizon (Al-Gherwi *et al.*, 2013). This is especially interesting for the control of commuter trains since it allows for the possibility of defining safety constraints and enforcing an optimal control strategy that resides within them. It should also be noted that optimal control

techniques would have full knowledge of a locomotive and would therefore be able to smartly control the train. PID based strategies often require tuning to perform well within operating bands whereas, assuming accurate model knowledge, a model based controller such as an optimal controller simply needs to be configured with the known vehicle model and thus no further tuning would be required (LaValle, 2006).

While a PID can be used as a paired unit following an optimally generated speed profile, it does not have the flexibility of an optimal controller and is both unusable for more advanced control objective and cannot actively redesign the speed profile when encountering large disturbances, modeling mismatch, or on-line changes in trip constraints (LaValle, 2006).

Research efforts in the field have focused on optimal control of commuter trains as either a numerical or analytical optimization problem (Novak *et al.*, 2015; Wang *et al.*, 2011). Analytical methods allow for a closed-form solution to the optimization that is very fast to compute yet suffers from only being able to support simplistic vehicle models and control objectives. With realistic models, input constraints and complex trip objectives, such as are required for advanced train control, traditional analytical optimization methods become unsolvable. Numerical optimization strategies, on the other hand, show great promise in their ability to handle more realistic control problems but traditionally suffer from being slow and computationally expensive in computing optimal solutions. Additionally, the optimal solution uniqueness together with global optimality have largely not been addressed with numerical train control problems (Wang *et al.*, 2011).

Numerical optimal control is usually prevalent in processes with very slow cycles, such as engineering processes where control decision can take minutes to hours

to compute without negatively affecting closed-loop performance (Al-Gherwi *et al.*, 2013). Trains, on the other hand, require quite fast control frequencies in order respond immediately to disturbances and maintain an energy efficient control profile that guarantees a timely arrival. The inherent complexity of the vehicle model, track and problem constraints make the optimal control problem for a train a challenge. This, combined with the requirement for a full guideway and high resolution control horizon all while being real-time capable has largely meant that this control problem has not been resolved for realistic models in a real-time manner as an on-board system (Novak *et al.*, 2015; Wang *et al.*, 2011; Liu and Golovitcher, 2003).

1.2 Problem Statement

The objective of this research is to design a fully deployable optimal control system for a commuter train that is capable of reliably controlling a rail vehicle under all feasible real world objectives and track configurations. It must do this while being fully configurable between time or energy efficient driving strategies while ensuring speeds stay within allowed dynamic track limits, arrival is within an allowed time window, and with a high final stopping accuracy. All of this must be done while adhering to very strict computational time constraints, something that has not been possible with existing numerical optimization strategies, on minimal computing hardware.

This thesis investigates formulating a train control problem as a non-convex control law followed by a convex transformation of the original non-linear control problem with the subsequent formulations implemented and the practicality of both systems as a deployable control system analyzed.

1.3 Research Overview

The objective of an optimal controller for a rail vehicle is simply to drive a train between two stations while adhering to strict speed limits with both starting and finishing at a stopped state. It must do all of this while keeping in consideration the physical properties of the vehicle, the track conditions ahead and in some cases the additional constraint of driving no closer than a set distance to another vehicle sharing the track. To achieve the highest accuracy control trajectory it is most useful to optimize for an entire guideway and not just for a small window as is usually the case with simpler model predictive control strategies (Al-Gherwi *et al.*, 2013). This presents a significant challenge as the guideway must be accurately discretized, and longer guideways would result in a higher dimensional control problem that would inherently be slower and more computationally expensive to numerically solve. Furthermore, since railed vehicles are capable of high speed operation, the maximum allowable controller response time is quite tightly constrained to no more than 100ms per cycle.

This thesis formulates and implements a controller that is capable of all of the above. First, vehicle dynamics models are developed in both time and position domain. With the models developed, a control law is proposed which takes into account the true position domain vehicle model with the intent of using it to completely plan out all control decisions based on vehicle performance for an entire control horizon. Beyond this, there is a large number of possible constraint variations that can specify any given trip such as: both lower and upper bounds on arrival times at a station; speed limits that the vehicle cannot violate which may vary in different track regions due safety or civil specifications; and strict requirements for high accuracy stopping

when arriving at a station. All of these are implemented as constraints in the optimization. Since there are, in essence, infinite feasible control laws that satisfy these constraints for any given trip, the next phase is to define a sensible and applicable objective function to uniquely define what is the most optimal course of control action for the ATO to take. Total trip time and energy are chosen for the minimization criteria of a mixed objective cost function where the weights attached to these criteria can allow for vast customizability of the objective function.

With the initial formulation being in a non-convex form, the next phase of this research was to investigate approaches for numerically solving this problem and the robustness, real-time viability and quality of the resultant solution. Two optimization packages, NLOPT and IPOT, are compared in this regard. To ensure that the most realistic timing information is observed for both solvers, the optimization is hard-coded into their C APIs. It is observed that, while both can solve the aforementioned control law, neither scales well for larger problem sizes. Because of this, neither is real-time capable for realistic control horizon dimensions. IPOPT shows substantially better performance than NLOPT, but both greatly suffer from infeasibility issues when run in closed-loop with a simulated vehicle model. These problems, along with the fact that a global optimum would require a multi-start optimization strategy, under which it may still be unattainable (Pál, 2013), make these numerical techniques unsuitable for vehicle control and leads to the motivation in reformulating the control law to a convex representation.

Through significant restructuring of the problem by epigraph and hyperbolic means, the problem is transformed into a second-order cone problem that is fully convex. In this form, extensions to the control law are presented as they are quite

simple to implement and provide significant performance benefits. An additional soft speed limit, brake reservation constraints and the ability to safely follow another vehicle are presented as a mix of new constraints and a new augmented objective function. In a convex form, effective numerical computational methods can be used to solve the problem that satisfy necessary and sufficient conditions for optimality of a global optimum (Boyd and Vandenberghe, 2004; Hiriart-Urruty and Lemaréchal, 2013).

To assess the quality of the solutions to the convex problem and to select the best numerical solver for further development a simple Disciplined Convex Programming (DCP) environment, CVXpy, is used (Diamond and Boyd, 2016). The convex numerical solvers MOSEK, ECOS, SCS and CVXOPT are compared and all show the capability to solve the convex control problem with a varying degree of performance and quality. The convex problem shows very impressive scalability and real-time capable performance with both MOSEK and ECOS for very large horizon dimensions. All solvers but SCS are robustly capable of solving the control problem to a high precision. Through significant testing, and the fact that it is open source (Domahidi *et al.*, 2013), ECOS is ultimately selected for further development into an embedded controller.

The control problem is implemented directly into ECOSs C-API in order to eliminate the large computational overhead of CVXpy and the subsequent controller is interfaced to an advanced vehicle model written Lua. The performance of control is then assessed with closed-loop tests under a wide spectrum of possible control scenarios that range from a simple trip on a flat grade all of the way up to long trip with highly dynamic grade and speed limit profiles subject to safely following another

vehicle. Good performance is observed in all tested situations. Subsequently, performance is assessed under modeling uncertainty and plant/model mismatch. A simple parameter estimator is formulated and tested to improve performance under the presence of very large plant/model mismatch- showing promising control improvements when operating in conjunction with the controller. Real-time results are presented on both desktop and embedded hardware demonstrating that the controller is shown to be greatly capable in real-time on-board applications.

1.4 Thesis Outline

- *Ch. 2* presents an outline of existing literature in the field of train control.
- *Ch. 3* develops the train vehicle model.
- *Ch. 4* develops a non-convex numerical control strategy.
- *Ch. 5* assesses methods of solving the non-convex problem.
- *Ch. 6* reformulates the optimal control problem into a convex form and formulates useful extensions to the problem.
- *Ch. 7* compares numerical techniques for solving the convex problem.
- *Ch. 8* implements the convex numerical control problem as an embedded closed-loop controller.
- *Ch. 9* demonstrates performance results for the optimal convex control system.

- *Ch. 10* demonstrates performance results under uncertainty conditions and demonstrates the effectiveness of pairing the optimal controller to a parameter estimator.
- *Ch. 11* concludes this thesis and presents possible future directions for this research.

1.5 Research Contributions over Existing work

- A novel non-convex numerical optimization with the position domain model of a train. The formulation for this was done in collaboration with Muzamil Rashid (McMaster University, Department of Electrical and Computer Engineering).
- Comparison testing on the capability of numerical solvers in solving the non-convex train control problem implementation.
- A novel advanced numerical optimization closed-loop controller as a convex second-order cone problem that is real-time capable.
- Closed-loop performance tests with the controller both with and without system uncertainty.
- Globally optimal solution for a numerically optimized train controller.
- Comparison of numerical solvers for the purpose of solving the convex train control problem.
- Embeddable numerical problem and solver interface for deployment on minimal hardware.

- Closed-loop optimal control of a train while maintaining a safe distance behind another train.

1.6 Related Publication

Submitting to IEEE Transactions on Intelligent Transportation Systems : "A Convex Real-time Capable Optimal Controller For a Commuter Train".

Status: pending industrial review by Thales Canada Transportation Solutions (TCTS).

Chapter 2

Literature Review

Research in train speed control methods can be separated into non-optimal and optimal methods that can, to a various degree, drive a railed vehicle. In general, optimal control methods, especially numerical strategies, allow for the most advanced control capabilities as is evident in the reviewed research below (Wang *et al.*, 2011; Novak *et al.*, 2015).

2.1 Non-optimal Control Strategies

Classical control systems, as mentioned previously, involve a collection of ad-hoc strategies that are tuned until performing as required. Allotta *et al.* (2013) devised a switching PID system where good driving and stopping performance was demonstrated in following a reference speed profile (Allotta *et al.*, 2013). The design featured three seperately tuned PID systems; two of which tracked the feedback error between a reference speed signal and the measured velocity for increasing and decreasing speed regions, respectively; and a stopping PID that was switched to when close to a stop,

which would be tuned to minimize the stopping error (Allotta *et al.*, 2013). Since their control approach was not model based and required trip speed profiles to follow, their approach was neither adaptive nor optimal and suffered from being unable to operate with more advanced trip objectives and constraints, while requiring significant tuning for three separate PID systems.

2.2 Optimal Control Strategies

Research into the optimal control of commuter trains is largely distributed between numerical and analytical optimization techniques. Numerical techniques, such as the implementations presented in this research, usually allow for more realistic vehicle and trip models but generally suffer from unreliable global minima and computationally do not usually allow for a real-time capable control platform (Wang *et al.*, 2011).

2.2.1 Analytical Optimal Control Strategies

In the analytical realm of control, Ichikawa (1968) was the first to solve the optimal control problem on a simplified train model through Pontryagin's maximization principle (Ichikawa, 1968). Since then, analytical methods have greatly advanced but still generally suffer from the inability to find a solution when subjected to more advanced nonlinear vehicle models, track models and trip constraints (Ko *et al.*, 2004).

Albrecht *et al.* (2016a) formulated a theoretical optimal energy control strategy that was based on local optimization around grade transitions followed by global optimization to determine the optimal control switching points along a track (Albrecht *et al.*, 2016a). Previously, Howlett and Pudney (1995) have proven, through an

analytical optimization problem, that an energy optimal control profile for a railed vehicle on a flat grade with a constant speed limit is to MOTOR-SPEEDHOLD-COAST-BRAKE (Howlett and Pudney, 1995; Albrecht *et al.*, 2016a), as can be seen in *Fig. 2.1.* Albrecht *et al.* (2016a) along with numerous other research have focused on treating the train control problem as time optimal or energy optimal switching problems where the transition points for a MOTOR-SPEEDHOLD-COAST-BRAKE strategy are presented as an analytical or non-convex numerical optimization problem (Novak *et al.*, 2015). The MOTOR phase represents a region of maximum motoring power to reach the maximum allowed speed under a maximum allowed acceleration. In the SPEEDHOLD region, limited motoring is used to counteract resistance forces and keep the vehicle's speed close to the maximum allowable speed limit. Afterwards the vehicle coasts for a period in the COAST phase and then brakes with maximum force in the BRAKING region. In cases where there is a track grade, such a control strategy does not hold.

Albrecht *et al.* (2016a) formulated a local optimization to determine local switching points for each transition in grade segments and proposed optimal driving modes for uphill as well downhill segments. The local optimization result was then used to compute the master switching problem along the entire track. They were able to demonstrate good off-line optimization results for a relatively advanced vehicle model. Solutions to their problem required a complicated analytical solving technique paired with sub-optimal ad-hoc strategies and numerical optimization methods with a both poorly modeled grade profile and piece-wise constant track model (Albrecht *et al.*, 2016b). The controller was not presented in a closed-loop online form and required several ad-hoc re-computations of the optimal profile to come up with

a feasible solution (Albrecht *et al.*, 2016b,a). As such, their claim for the existence of a unique optimal solution is overshadowed by the fact that it is for a sub-optimally modeled system, and no claim is made on whether or not it is always attainable using their perturbation methods. In addition, no computational aspects were presented and their solution did not allow for continuous control input nor did it account for dynamic speed limit profiles (Albrecht *et al.*, 2016b).

Albrecht *et al.* (2015) formulated an energy optimal control strategy for cases where there are two trains driving in sequence one after another on a flat grade profile in such a way as to maintain a safe distance buffer between them. Their research focused on a two stage approach. First, the leading train was optimized for followed by the following in an energy optimal profile. As with the related research conducted by Albrecht *et al.* (2016a), the problem was developed as a switching problem. The following train problem enforces the separation for the two vehicles by constraining the arrival time at each of the signals along the track for the following train, such that the following vehicle cannot arrive at these positions earlier than a set time offset. Their problem formulation was not intended to be run on-line. Methods of analytically or numerically solving this optimal control problem were not addressed (Albrecht *et al.*, 2015).

2.2.2 Numerical Optimal Control Strategies

In the realm of numerical train control strategies, Matsuura and Miyatake (2014) implemented a dynamic programming approach to solving the optimal speed profile for a train, in both a single-threaded and parallel approach. The vehicle model that

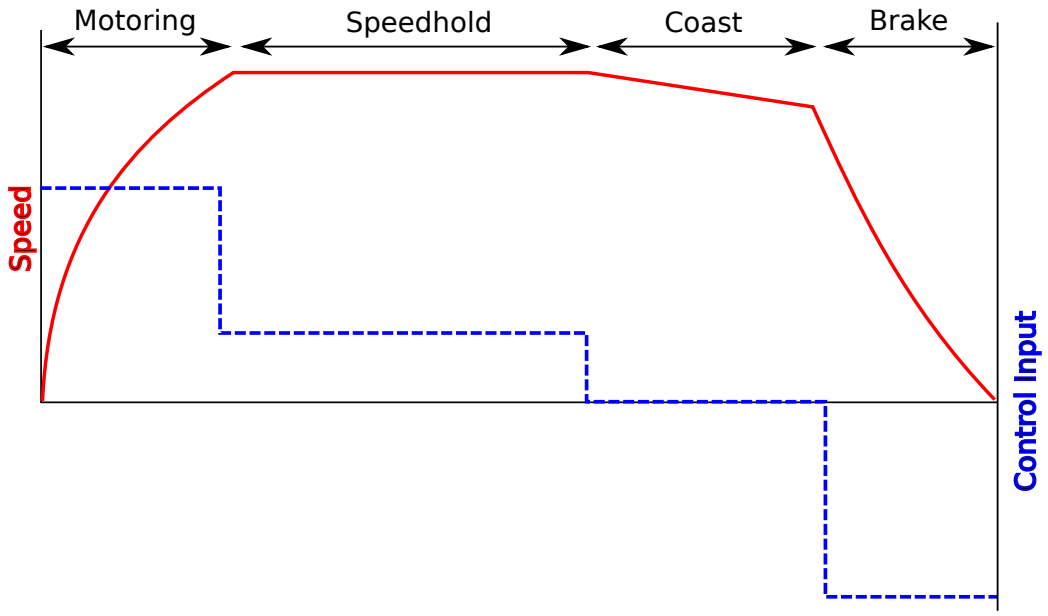


Figure 2.1: Energy Optimal Control Strategy

they used was significantly advanced involving complicated speed limits, grades, regenerative braking and even advanced dynamic modeling for cross-car forces. They were largely interested in controlling freight trains which are known to have significantly more advanced models than commuter trains (Matsuura and Miyatake, 2014). Although they were able to demonstrate a significant computational improvement when parallelizing their algorithms, their solution was far from real-time capable as a 1km journey took hundreds of seconds to compute on a parallel cluster (Matsuura and Miyatake, 2014).

Other numerical research has yielded similar limitations to the work by Matsuura and Miyatake (2014) and as such has proposed developing optimal control lookup tables of required trip trajectories based on varying train and trip configurations that can be loaded in real-time once a train is in motion. Needless to say, this is by no means an optimal closed-loop strategy and is incredibly computationally expensive,

completely non-adaptive to changes in trip conditions, and impractical for day-to-day operation (Vašak *et al.*, 2009; Novak *et al.*, 2015).

Since real-time considerations have generally not been overcome with neither advanced analytical not numerical optimization techniques, other research has focused alternatively on the optimal tuning or the optimal design of a suboptimal real-time capable control strategy. Chang and Xu (2000) demonstrated the effectiveness of pseudo-stochastic numerical optimization strategies (genetic algorithms and differential evolution) to optimally select the membership functions for a fuzzy ATO control strategy (Chang and Xu, 2000). They were able to demonstrate good performance under advanced grade profiles and non-constant speed limits (Chang and Xu, 2000). Their tuning objective function was multi-objective and thus the fuzzy ATO controller could be retuned for different performance criteria. Regardless of this, their control did not account for the future impact of fuzzy control decision and as such cannot guarantee arrival times nor smartly select control based on future track conditions (Chang and Xu, 2000; Welch and Venayagamoorthy, 2006).

Huang *et al.* (2016) researched into more advanced neural network methods for controlling the braking system on a train. They used a multi-layer neural network that was trained via a modified back-propagation algorithm on normal train driving data with the goal of it learning when to switch to braking for the vehicle. They demonstrated good performance for transitioning to braking on their testing data (Huang *et al.*, 2016). As with many other neural network control systems, nothing was addressed in regards to the reliability of the system (Huang *et al.*, 2016; Kosiński and Kozłowski, 1998). The reliability of neural networks relies on many factors such as the size and structure of the network and the reliability of each discrete neuron and

the training problem. Additionally, the explicit algorithm of operation for a neural network is not known, and thus not all possible reactions to all possible inputs can be reliably known (Kosiński and Kozłowski, 1998). Furthermore, at best the controller will learn from the performance in the training data and thus will only be as energy or time efficient as the training data is itself. This concern paired with the control algorithm only being usable for braking with only speed limit adherence and accurate stopping in mind make this control architecture unsuited to real-world train control applications.

2.3 Optimal Driver Assisted Systems

Since real-time optimal control has presented a substantial challenge to research, research focusing on industrial applications of such systems has instead reverted to DAS (driver assisted system) type systems. For these systems, real-time considerations are to a lesser degree a concern as these systems are responsible with supplying the driver with information as to when optimal control mode switching is encouraged. The driver then chooses when and if to follow the DAS output. Thus, these are not truly closed-loop control architectures. The systems work by receiving timetable, grade and speed limit information while the train is stationary at a starting station and optimally computing the required switching points once, then optionally (rarely) recomputing them again while the train is in motion (Novak *et al.*, 2015; Wang *et al.*, 2011). As such, this is not an ATO system and suffers from the inability of being able to rapidly and reliably compensate for unforeseen disturbances. Despite this, Siemens, Bombardier, and TTG Transportation Systems have demonstrated a notable reduction (in some cases up to 23%) in driving energy and have shown an

added benefit of reduced wear on the tested vehicles (Hofestadt, 1995; Bombardier, 2008; Albrecht *et al.*, 2015).

The TTG Transportation Systems team have developed their own numerical optimization algorithms for their Energymiser product based on the research by Albrecht *et al.* (2016a) and claim to be capable of actively recomputing the optimization problem in several seconds on mainstream hardware. No performance testing or evidence is supplied to support this claim (Albrecht *et al.*, 2015). Such performance would again mean that such a system is more suited to a closed-loop DAS system more than it is to true ATO operation.

2.4 Contrasting with Existing Research

Truly optimal train control methods are often too complex and too slow to solve online and as such research has largely targeted optimal planning techniques as either off-line systems or DAS type models (Wang *et al.*, 2011; Novak *et al.*, 2015). This thesis structures the optimal control problem as an on-board online closed-loop control system in contrast to the more conventionally researched optimal off-line planning problems.

This thesis presents a novel globally-optimal convex derivation and implementation of a commuter train optimal control system that is capable of real-time performance. Contrary to other numerical optimal control techniques in the field, this thesis does so with an advanced and real-world accurate train model, realistic trip and track designs (speed limits, arrival times, grade profiles, etc.), both with and without sharing the track with another vehicle. For several vehicles sharing the same track this research employs a similar strategy to that of Albrecht *et al.* (2015), but

does so for every signal point along the track and not just at signaling points. In addition, Albrecht *et al.* (2015) formulated their optimization for a flat grade profile, whereas the controller developed in this thesis can do this on any arbitrary grade in real-time.

The convexity of the presented approach assures global optimality— something uncommon with most numerical optimal control in the field— thus supporting the notion of a unique control solution (Boyd and Vandenberghe, 2004). Moreover, it does this by using the entire trip distance (from stop to stop) as the control horizon as opposed to using only a small subset of the trip, and does not solve for only switching points but allows for far more dynamic behavior along the track. As the problem is resolved optimally at every control cycle, any changes in the trip timetable or speed profiles can be optimized for immediately. The controller presents the ability of using an advanced mixed optimal objective which can be used to tune the optimal solution between a minimum energy, minimum time, minimum braking above a soft maximum, and a soft buffer under the speed limit. As such, the control is highly user configurable allowing for fine-tuning of the vehicle, safety margins, commuter comfort and target arrival to allow for the best possible closed-loop performance.

In contrast to other research discussed above, this thesis develops a controller with the idea of embedded applications in mind. The controller is implemented with a direct interface to a robust second-order cone problem solver (ECOS) and shows real-time capability with an advanced train model, sensor and model uncertainty on non-trivial guideways and trip configurations on both x86-64 machines and embedded ARM computers.

Chapter 3

Model Formulation

The control objective for normal commuter train driving involves balancing traction and braking effort subject to hardware limitations, knowledge of the track and trip constraints. For efficient optimal control trajectory planning, accurate knowledge of the dynamics of the locomotive vehicle are required, and as such are formulated below.

3.1 Time Domain

The equations of motion of a locomotive can be described with the position ($x(t_i)(|m|)$), speed ($v(t_i)|m/s|$) and acceleration ($a(t_i)|m/s^2|$) states at discrete points in time, t_i (Xu and Wang, 2014; Zhang and Zhuan, 2015). Thus the physics of a train are formulated as a discrete time point mass system that has, at time t_i (for $i = 1 \dots K$), an applied propulsion or braking command force Γ_i, motion resistance $C(v_i)$, and track grade acceleration forces $G(x_i)$ all in N (Xu and Wang, 2014; Zhang and Zhuan, 2015; Rochard and Schmid, 2000; Boschetti and Mariscotti, 2012). The model can

be described by the ordinary differential Newtonian equations of motion numerically integrated via the Euler method as below:

$$x_{i+1} = x_i + v_i \Delta_t \tag{3.1a}$$

$$v_{i+1} = v_i + a_i \Delta_t \tag{3.1b}$$

$$a_i = A(v_i, \Gamma_i) + C(v_i) + G(x_i) \tag{3.1c}$$

where the grade acceleration $G(x_i)$ is a function of the vertical track inclination $(\theta(x_i))$ as follows:

$$G(x_i) = g \sin(\theta(x_i)) \tag{3.1d}$$

For convenience, this thesis also refers to the input acceleration u which is represented as: $u = A(v_i, \Gamma_i)$. This, as seen in the above equations, represents the acceleration due to motoring or braking input forces that is felt by the vehicle.

Davis Equation

The motion resistance model $C(v_i)$ can be expressed as the Davis equation with constant coefficients c_0, c_v, c_{v2} (Rochard and Schmid, 2000). The coefficients of this model represent the rolling friction, air resistance, mechanical resistance and other forms all of which can be captured through a proper selection of c_0, c_v, c_{v2} (Rochard and Schmid, 2000), either through the optimal parameter estimation of vehicle test data or semi-empirically through equations such as the Armstrong and Swiff resistance model (Boschetti and Mariscotti, 2012). The model is widely represented in modern train literature as:

$$C(v_i) = c_0 + c_v v_i + c_{v2} v_i^2 \tag{3.2}$$

Traction Loss

Traction motors supply the tractive effort needed to move a rail vehicle. They range from DC series motors and three phase induction motors to hybrid diesel electric power-trains with complex transmission mechanisms. They may also have advanced on-board control systems to track provided reference input, limit slip-slide and limit wear and tear on the motors. Regardless of their mode of operation, rail motoring systems take in a reference effort as input that must be within their operating range and output the requested effort on the driven system (Novak *et al.*, 2015; Eugene A. Avallone, 2006).

It is common for rail electric motors to exhibit tractive effort loss behavior and have maximum propulsion limits of the form of *Fig. 3.1.* As a result of the behavior of electric motors at high speed, it can be expected that when moving fast a train's motor would only be able to supply a fraction of the traction force that it would have otherwise been able to supply when moving at a crawl due to constant maximum power constraints. In literature, there is general agreement over a two region traction model. In the first, low speed, region the motor is capable of supplying up to a constant tractive acceleration force (F_{max}). In the higher speed region, the motor is now in constant power mode where the tractive effort decays along a curve $R(v)$ (Eugene A. Avallone, 2006). Within this maximum limit, the input to output force mapping is linear. As a complete model, given that the mass of the vehicle is m and the maximum braking force is F_{min}, the traction acceleration can be represented as:

$$A(v_i, \Gamma_i) = \frac{\min(\Gamma_i, R(v_i))}{m} \qquad \text{for} \qquad F_{min} \leq \Gamma \leq F_{max} \tag{3.3}$$

On a flat grade profile, the hypothetical maximum speed limit occurs along the intersection of $A(v_i, \Gamma_i)$ and the resistance curve $C(v_i)$ (Eugene A. Avallone, 2006).

In railway literature there are many presented models for the traction curve $R(v_i)$, and a very common representation is as a polynomial of v_i. For this vehicle model, the form below is used:

$$R(v_i) = r_0 + r_v v_i + r_{v2} v_i^2 \tag{3.4}$$

where r_0, r_v, r_{v2} are constant polynomial coefficients provided by a vehicle manufacturer or through parameter estimation (least-squares) of provided motoring force curves.

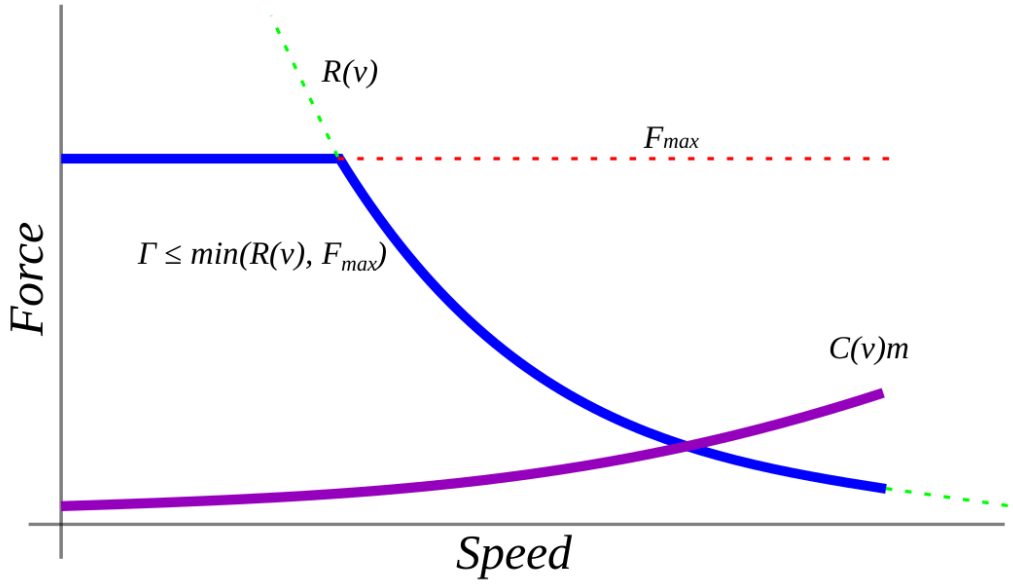


Figure 3.1: Traction Loss and Vehicle Resistance

3.2 Position Domain

In the time domain, the train control optimization horizon is the discretized time vector t for $t = 0 \ldots t_K$ with a fixed sampling interval of Δ_t (Novak *et al.*, 2015). The time domain formulation, although mathematically simple, is very difficult to

use in an optimal control problem due to the inherent non-linearities presented by the friction, traction and grade models. To address these difficulties, we propose a reformulation into the position domain.

It has been shown by Howlett and Pudney (1995) that the advanced distributed vehicle model for a multicar train can be in total represented as a point-mass model in the position domain (Howlett and Pudney, 1995). Since $G(t)$ is a non-smooth, non-differentiable function of t_i that varies from trip to trip, the train physics can be reformulated in a way that makes the grade effectively a lookup table, with x as the independent variable.

The control horizon in the position domain is now along the train position vector X (Ouyang and Dam, 2011). Therefore the entire distance between two stations can be discretized into a constant vector X of length K, with a finite differences vector Δ_x of length $K-1$. The trip grade can be discretized into the vector G of length $K-1$, where each $G[i]$ is the grade acceleration at the corresponding $X[i]$. This allows for a substantially more useful representation of the train trip since, for any given trip, the station distance is a fixed known whereas arrival time can and will vary given different train models, trip constraints, grade, and other design criteria. Energy is also simpler to calculate in this domain as it is trivially $\Gamma^T \Delta_x$ - an affine function of control input. Therefore a simple convex mixed optimization objective of both time and energy is possible.

In this domain, the equivalent to acceleration is the position domain acceleration (the partial derivative of velocity with respect to position) $\alpha(1/\text{s})$. Additionally, the position domain also allows to change the train physics model into a distributed mass model given only a change in how the constant G vector is calculated (Ouyang and

Dam, 2011; Novak *et al.*, 2015; Xu and Wang, 2014). The dynamics are transformed to the following:

$$t_{i+1} = t_i + \frac{\Delta_x}{v_i} \tag{3.5a}$$

$$v_{i+1} = v_i + \alpha_i \Delta_x \tag{3.5b}$$

$$\alpha_i = \frac{\partial v_i}{\partial x} = \frac{a_i}{v_i} = \frac{A(v_i, \Gamma_i) + C(v_i) + G[i]}{v_i} \tag{3.5c}$$

These equations are equivalent to the time domain representation of the vehicle motion *Eqn. 3.1.*

3.2.1 Resolving Control Singularities

The position domain representation of the dynamics does suffer from a singularity when the vehicle velocity (v) approaches zero. This in turn causes α to approach infinity which is a substantial problem in control. This numerical issue can be eliminated by constraining the vehicle velocity to remain larger than an arbitrarily small value (ϵ) as:

$$v \geq \epsilon \tag{3.6}$$

where ϵ is small enough to effectively mean that the train is stopped at $v = \epsilon$.

3.2.2 Higher Precision Control

Since both v_i and t_i are evaluated as difference equations, a large Δ_x when v_i is small will result in a very large jump from t_i to t_{i+1} resulting in very poor modeling accuracy. This behavior can be effectively minimized by selecting a large K, such that Δ_x is finely sampled enough for this to not be a concern. Alternatively, compared to

selecting the Δ_x profile as a constant, it can be intelligently selected as a non-uniform profile such that the elements of Δ_x are small for track regions where the velocity is expected to be low and larger otherwise. In this way a shorter control horizon dimension K can be used without any penalties in low-speed modeling accuracy, thus resulting in a more computationally efficient optimal control solution.

In practice, a trapeziodal Δ_x profile — as demonstrated in *Fig. 3.2* — was empirically determined to be sufficient for most normal vehicle driving strategies. This profile contains increasing $\Delta_x[i]$ for regions close to the starting station and conversely decreasing $\Delta_x[i]$ for regions approaching the stopping station as these regions represent large increases and decreases in velocity, respectively. The minimum value of Δ_x is chosen to be a small nominal non-zero value.

Other, more optimal selections for the Δ_x profile may exist but are outside the scope of this research and are a potential subject for future research. For this research the trapezoidal Δ_x profile is used in all tests conducted on the optimal controllers presented.

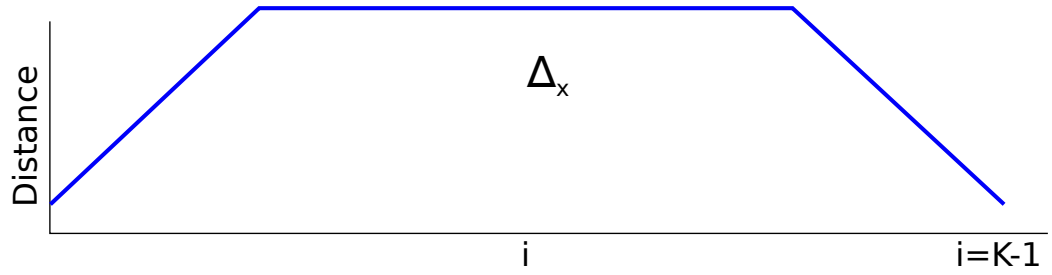


Figure 3.2: Trapeziodal Δ_x Profile

Chapter 4

Non-Convex Position Domain Optimization Formulation

4.1 Non-Convex Formulation

The purpose of the controllers formulated in this research is to simply drive a commuter train from some station A to a station B in the most efficient and safe way possible. This is done by optimizing for the entire optimal control input profile Γ^* for the entire distance between the present location of the train and the target station and subsequently sending the first control element to the vehicle (Γ_1^*). This process is repeated until the train safely arrives at the target. This is not a trivial task by any means due to both advanced train models and complex guideway constraints.

This thesis assumes that no reference velocity profile is provided to the closed-loop controller and that it is up to the optimal control system to generate the control profile for the entire remaining travel distance given only knowledge of the train model, trip specifications (speed limits, grade profile, arrival time window, etc.), and

current states of the train. The entire closed-loop control problem is formulated as follows:

$$\underset{t,v\in\mathbb{R}^K \quad \Gamma,\Gamma^+\in\mathbb{R}^{K-1}}{\text{minimize}} \qquad f(z) = w_t t_K + w_e \frac{\Delta_x^T \Gamma^+}{m} \tag{4.1a}$$

subject to:

$$0 \leq \Gamma^+ \qquad \Gamma \leq \Gamma^+ \tag{4.1b}$$

$$\epsilon \leq v_i \leq V_{max}[i] \qquad i = 2, \ldots, K-1 \tag{4.1c}$$

$$v_1 = V_0 \qquad v_K = V_f = 0 \tag{4.1d}$$

$$t_{i+1} = t_i + \frac{\Delta_x[i]}{v_i} \qquad i = 1, \ldots, K-1 \tag{4.1e}$$

$$t_1 = T_{start} \tag{4.1f}$$

$$T_{min} \leq t_K \leq T_{max} \tag{4.1g}$$

$$F_{min} \leq \Gamma \leq F_{max} \tag{4.1h}$$

$$\Gamma_i \leq r_0 + r_v v_i + r_{v2} v_i^2 \qquad i = 1, \ldots, K-1 \tag{4.1i}$$

$$v_{i+1} = v_i + (c_0 + G[i] + \frac{\Gamma_i}{m} + c_v v_i + c_{v2} v_i^2) \frac{\Delta_x[i]}{v_i} \qquad \text{for} \quad i = 1, \ldots, K-1 \tag{4.1j}$$

The objective of the control problem is a mixed cost one with weight w_t on arrival time and weight w_e on non-regenerative trip energy. This allows for simple tuning for a spectrum of control objectives simply by the selection of weights to achieve a pure time, energy or mixed objective. This allows the controller, for example, with a mixed objective to minimize the energy used when there is an active constraint of T_{min} on the arrival time at a station. Thus, since the time cannot be minimized below T_{min}, the energy will be minimized instead.

An additional auxiliary variable Γ^+ is added to extract only the positive control applied to the train for use in the energy calculation in the objective.

The total trip energy per unit mass can then be evaluated as $(\Delta_x^T \Gamma^+)/m\ |J|$. The unit energy is more useful than the actual trip energy since the objective function should not place more emphasis in energy minimization for a heavier vehicle than it would for a light vehicle regardless of their passenger loading levels. This can equivalently be rewritten as $(\Delta_x^T max(\Gamma, 0))/m$ with the auxiliary variable removed, but the former is more useful as it is at least twice continuously differentiable and is already in an affine formulation. The time objective of the cost function minimizes the arrival time t_K.

The entire mixed control objective is defined as (*Eqn. 4.1a*):

$$\underset{t,v \in \mathbb{R}^K \quad \Gamma, \Gamma^+ \in \mathbb{R}^{K-1}}{\text{minimize}} \qquad f(z) = w_t t_K + w_e \frac{\Delta_x^T \Gamma^+}{m}$$

where Γ^+ is defined as the following epigraph transformation and is equivalent to the motoring input applied on the vehicle (*Eqn. 4.1b*):

$$0 \leq \Gamma^+ \qquad \Gamma \leq \Gamma^+$$

The vehicle's velocity is constrained to be positive and within a safe required civil or track speed limit. It is constrained from below by the small positive number ϵ to stop control singularities in the optimization and disallow driving in reverse. Therefore, the speed is constrained as follows (*Eqn. 4.1c)*:

$$\epsilon \leq v_i \leq V_{max}[i] \qquad i = 2, \ldots, K-1$$

The initial velocity is then constrained as the current measured velocity of the train (V_0) — equaling ϵ on the first optimization cycle when the train has not yet left the starting station. The velocity at the target station at $X[K]$, is set in such a way as to force the train to arrive stopped at the station. It is not necessary to set the stopping speed to ϵ since it cannot cause a singularity and as such the final stopping

speed is constrained to zero. Therefore, the terminal velocity constraints are set to be (*Eqn. 4.1d*):

$$v_1 = V_0 \qquad v_K = V_f = 0;$$

where ϵ is sufficiently small such that the train has to continue braking for an insignificant fraction of a second more after reaching $X[K]$ at V_f. This ensures that the final stopping distance error is inconsequentially affected.

The time at each segment along the track can be evaluated as follows (the same as *Eqn. 3.6a*) (*Eqn. 4.1e*):

$$t_{i+1} = t_i + \frac{\Delta_x[i]}{v_i} \qquad i = 1, \ldots, K-1$$

with the initial time being zero when at the starting point and the current runtime of the trip when the train is in motion. It is constrained as (*Eqn. 4.1f*):

$$t_1 = T_{start}$$

Thus, the arrival time at $X[K]$ can be constrained from above and below by predefined minimum and maximum allowable trip traversal time limits (*Eqn. 4.1g*):

$$T_{min} \leq t_K \leq T_{max}$$

The braking limit and constant traction region of the traction loss motoring model ($A(v_i, \Gamma_i)$) are defined as the inequalities (*Eqn. 4.1h*):

$$F_{min} \leq \Gamma \leq F_{max}$$

The nonlinear region of the traction loss model ($R(v)$) is also introduced as an inequality constraint (*Eqn. 4.1i*):

$$\Gamma_i \leq r_0 + r_v v_i + r_{v2} v_i^2 \qquad i = 1, \ldots, K-1$$

Finally, the dynamic model for the vehicle velocity (*Eqn. 3.6c*) is constrained, resulting in the recursive equality equation (*Eqn. 4.1j*):

$$v_{i+1} = v_i + (c_0 + G[i] + \frac{\Gamma_i}{m} + c_v v_i + c_{v2} v_i^2)\frac{\Delta_x[i]}{v_i} \qquad \text{for} \quad i = 1, \ldots, K-1$$

4.2 The Closed Loop Problem

The full control problem can be defined as a closed-loop version of *Eqn. 4.1*. In this way a horizon x is discretized along the entire track spanning from the start station to the stop station, resulting in a known X, Δ_x, V_{max}, and G. The closed-loop control behavior can then be determined via solving *Eqn. 4.1* for the optimal Γ^* that satisfies all of the constraints and results in the optimal cost $f(z^*)$. The first element of Γ (Γ_1) is then extracted and applied to the vehicle plant. In the next feedback cycles, X_0, T_{start} and V_0 would be measured from the vehicle and fed back resulting in the receding horizon shortening X, Δ_x, V_{max}, and G. The shrinking of these vectors can be accomplished by the simple algorithm:

The current location on the guideway is used to search for the index (j) of the closest element of X to the current position X_0, such that:

$$X[j] \leq X_0 \qquad X[j+1] \geq X_0 \tag{4.2a}$$

V_{max} and G then have their initial access pointer moved to the j'th index, having the effect of popping off used up elements of theses vectors.

$$V_{max} = [V_{max}[j], V_{max}[j+1], \ldots, V_{max}[K]] \tag{4.2b}$$

$$G = [G[j], G[j+1], \ldots, G[K-1]] \tag{4.2c}$$

The same is done with X and Δ_x, but initial point of X ($X[1]$) is then updated to

the current X_0 and $\Delta_x[1]$ is the first finite difference of X.

$$X = [X_0, X[j+1], \ldots, X[K]] \tag{4.2d}$$

$$\Delta_x = [X[2] - X[1], \Delta_x[j+1], \ldots, \Delta_x[K-1]] \tag{4.2e}$$

This is a very computationally efficient implementation as it removes the need for re-allocating and re-discretizing the problem vectors.

The full control architecture can be seen in *Fig. 4.1.* Although the optimization problem is implemented in the position domain, the closed loop control law is implemented in the time domain resulting in a consistent response time in control updates. In this structure, *Eqn. 4.1* is resolved at resolved each $\frac{1}{f_{ctl}}$ seconds until the train arrives and stops at the target. In this way, the closed-loop re-optimization would allow for resolving modeling inaccuracies and external disturbances all while maintaining optimal driving behavior and rapidly adapting to changing problem specifications.

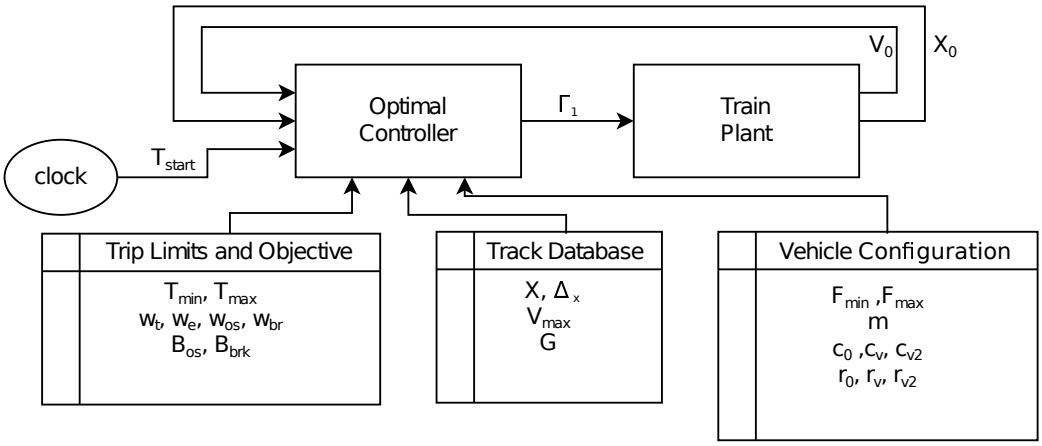


Figure 4.1: Controller Architecture

The track, model parameters and trip objective weights are all loaded from databases at each optimization cycle. Therefore, as the track horizon recedes, the controller pulls up the remaining speed and grade profiles (for the area ahead of the train) from the

database. Should the vehicle model change during operation, the updated model parameters can be supplied to the optimal controller. This is useful in situations where there is poor knowledge of the vehicle model in which case a real-time parameter estimator can be used to actively re-estimate the vehicle model and supply increasingly more accurate parameters to the optimal controller (Vukov *et al.*, 2015). The trip limits can also be changed at any control cycle. For example: considering a case where an optimally driven vehicle had driven a significant fraction of the distance to its target station when it, with no prior warning, receives an instruction from the ATS timetable scheduler that it must delay its minimum arrival time to the target station by a fixed amount of time. In this case a new T_{min} can be supplied to the optimal controller and the control system will immediately adapt the optimal driving profile given this new information, thus assuring safe adherence to the ATS limits.

For any given trip, running with an optimal controller in feedback can assure that the vehicle always stays within allowed trip constraints through optimal closed loop corrections (Vukov *et al.*, 2015). In addition, recomputing the optimal driving profiles allows the controller to take advantage of non-constraint violating feedback error and respond in an efficient way without unnecessarily expending energy or causing passenger discomfort by altering the entire future control trajectory instead of immediately and aggressively reacting to potentially favorable local disturbances. For instance: a disturbance that causes a higher than expected increase in vehicle velocity will allow the optimal controller to re-plan a trajectory that immediately corrects if a constraint violation is immediately imminent, or using the advantage gained by this increase and adjusting the future control strategy to be even more time or energy optimal that it was in the previous cycle.

Altogether, closed-loop re-optimization with updating vehicle states paired with the ability to change the trip and vehicle model in real-time allow for a highly adaptive and robust control architecture.

Chapter 5

Numerical Solutions to the Non-Convex Control Problem

As a non-convex problem, the optimal control profile can be optimized by a variety of robust non-convex numerical solvers. The motivation in doing so is to test the real-time capability, optimality and reliability of using a non-convex numerical optimizer with the non-convex optimization formulation (*Eqn. 4.1*) for the closed-loop on-board control system. Other research in the field has largely noted that real-time and robust performance is not usually apparent with non-convex numerical control (Wang *et al.*, 2011). To see if this claim is shared with the novel non-convex representation formulated in this paper, the early stages of this research focused on determining the practicality with which *Eqn. 4.1* can be solved for numerically using the non-convex NLOPT SLSQP solver package as well as IPOPT environment.

5.1 NLOPT

NLOPT is a free and open source MIT licensed solver package (although some of the internal solvers are licensed with the less liberal GNU GPLv3 license) that includes a wide collection of local and global optimization algorithms for solving unconstrained, equality constrained or inequality constrained non-convex optimization problems (Johnson, 2014). The package has bindings to numerous programming languages such as: C/C++, Python, Matlab, Julia, R and other mathematically oriented environments. Since the computational efficiency of numerically solving the control problem is of utmost importance, the lowest level and fastest implementation was used and thus the control problem was written in C and used the NLOPT native C interface.

Due to the specific need for both equality and inequality constraints to define the non-convex train control problem, NLOPT's SLSQP (Sequential Least Squares Programming) solver was used as it was the only provided solver suitable for the required problem structure. NLOPT has, in addition, some pseudo-stochastic solvers that could be used to solve the problem but these were not of notable interest, due to the poor computational performance and reliability usually attributed to solvers employing variations on the Genetic Algorithm method (Datta, 2012).

The implementation of the SLSQP method used in NLOPT is of the form proposed and implemented by Kraft (1994) and is a Quasi-Newton method that uses a dense matrix BFGS methods for Hessian matrix approximations using the supplied problem gradient information (Kraft, 1994). Therefore it is not advertised as being highly scalable and has $\mathcal{O}(n^2)$ storage and $\mathcal{O}(n^3)$ time scaling for an n dimensional

problem, and is thus only suited for problems with generally less than 1000 optimization variables (Kraft, 1994; Malouf, 2002).

The non-convex control law was implemented in C and interfaced to NLOPT's SLSQP method with supplied gradient information and BFGS approximated Hessian. A simple flat grade and constant speed limit problem was then solved only once as an initial optimization problem (with $V_0 = 0$, and $T_{start} = 0$), with the results given in *Tab. 5.1 and Fig. 5.1*

	$K = 30$	60	90
Time Optimal Objective	28.7	245.0	992.5
Energy Optimal Objective	280.0	2.0e3	129.0e3

Table 5.1: NLOPT Initial Horizon Optimization Times (ms) (AMD FX6300)

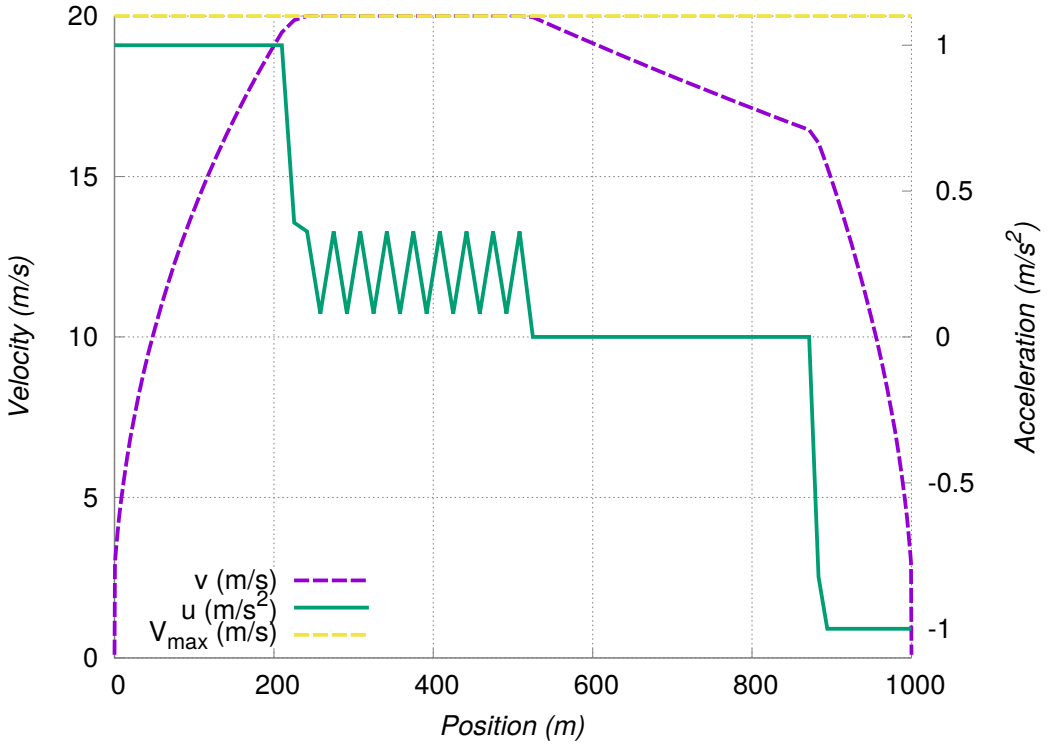


Figure 5.1: NLOPT Mixed Objective, Flat Grade (Open Loop) for K=90

As evidenced by *Tab. 5.1*, NLOPT's SLSQP solver does scale very poorly and can really only be usable in closed-loop control for $K = 30$ due to its slowness. Performance was seen to widely vary given a different selection of control objectives. As can also be seen from *Fig. 5.1* the problem is solvable with the SLSQP method to an optimal control profile similar to the expected flat grade optimum of the form of *Fig. 2.1*. Regardless of this, the small value of K results in a rather unstable control profile in the speedhold region. The slowness of SQP methods with quasi-Newton Hessian approximations, like SLSQP, in solving large or badly-scaled problems has been noted in other research (Schittkowski, 1982; Morales *et al.*, 2011; Malouf, 2002). Realistically, such a small horizon dimension cannot provide accurate vehicle modeling for control on any reasonable track length and thus SLSQP is an unsuitable non-convex optimizer for vehicle control.

5.2 IPOPT

IPOPT is a package for non-linear optimization using a primal-dual method with a filter line search algorithm developed by the COIN-OR project team (Wächter and Biegler, 2006). Like NLOPT, it too is available with bindings to other mathematical programming languages, but only the C implementation is of interest for this research. It is an open source project, but in contrast to NLOPT it does allow for using commercial subsolver routines to accelerate optimizations and allow for more robust problem solutions. In the implementation paper by Wächter and Biegler (2006), IPOPT showed favorable results in solution time, iteration count and function evaluations as compared to the commercial LOQO and KNITRO solvers (Wächter and Biegler, 2006). The algorithm contains significant heuristic correctional features and

thus claims to be robust at solving large-scale optimization problems.

As mentioned previously, the solver is capable of using a large variety of sparse symmetric indefinite matrix subsolvers in its algorithm to accelerate the problem computation such as: MUMPS (MUltifrontal Massively Parallel Sparse direct Solver) (Amestoy *et al.*, 2015), WSMP (Watson Sparse Matrix Package) (Gupta, 2000) or numerous COIN-OR HSL routines (Wächter and Biegler, 2006). It should be noted that of the three, only MUMPS is open-source and is therefore the prime candidate for use in this research. Although MUMPS is capable of multi-threaded math operations, the IPOPT interface to MUMPS does not support parallelization (Wächter and Biegler, 2006). The non-convex formulation was realized and tested with the IPOPT solver compiled with the MUMPS linear subsolver (Wächter and Biegler, 2006) and compiled under Linux with the efficient and paralleliseable openBLAS linear algebra routine directly through its C API. To further improve solver robustness and speed, the gradient and Hessian information for the objective, equality and inequality constraints were specified. Thus, no Hessian approximation is performed by IPOPT. It should also be noted that the problem is highly sparse, and as such all matrices were supplied to IPOPT sparsely to take full computational advantage.

	Linear Sub-Solver	$K = 30$	90	150	300	600	3000
Min. time	MUMPS	<1	15	38	1,399	18,192	6,883
	WSMP	12	20	28	36	85	392
	HSL MA57	8	16	4	31	2,406	8,231
Min. energy	MUMPS	12	24	40	116	286	1802
	WSMP	15	24	40	60	189	1573
	HSL MA57	6	12	16	35	56	503

Table 5.2: IPOPT Initial Horizon Optimization Times (ms) (AMD FX6300)

In *Tab. 5.2* the only free and open-source solver was MUMPS whose general

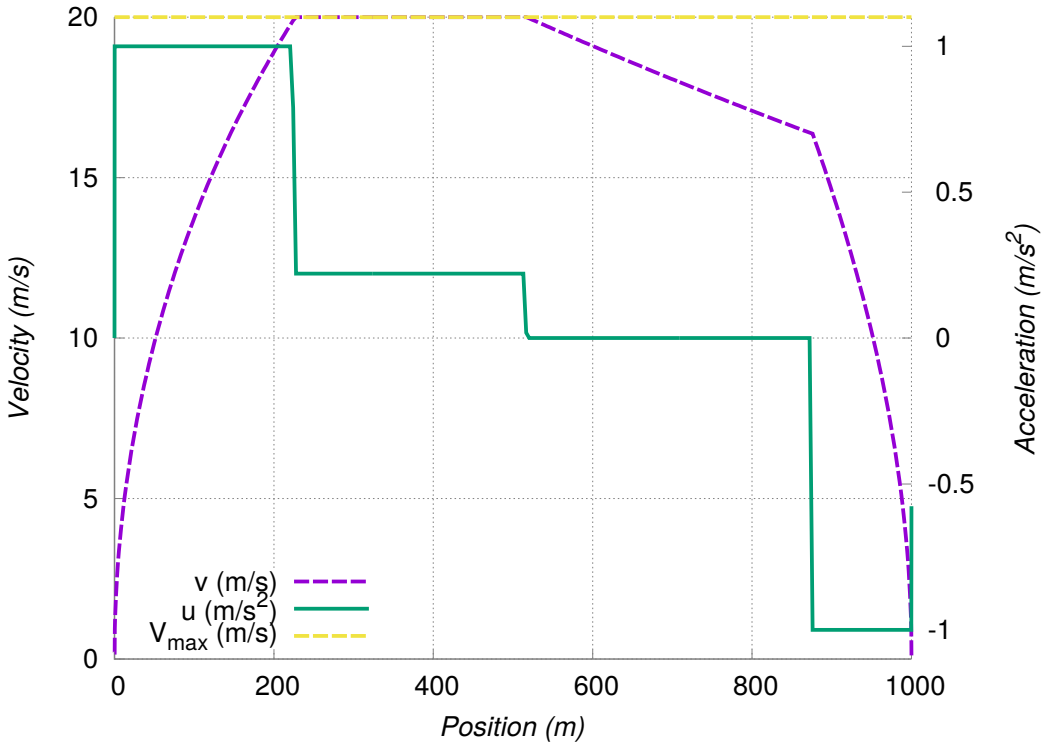


Figure 5.2: IPOPT Mixed Objective, Flat Grade (Open Loop) for K=300

performance was overall worse as compared to the commercial solvers (WSMP and HSL). As was evident with NLOPT, the optimization time is largely inconsistent between time optimal and energy optimal control regimes and the problem scaling is quite poor. The unpredictable nature of non-convex problem computation can be witnessed in the anomalous behavior with MUMPS for $K = 600$ and $K = 3000$, as the solver converges faster for the larger horizon size. This may be due to the solver attempting to solve a nearly infeasible problem for that particular value of K, memory alignment issues in MUMPS's linear algebra routines causing stochastic behavior for large problem sizes, or alternatively due to the chosen starting point resulting in a longer search for an optimal solution (Wächter and Biegler, 2006; Amestoy *et al.*, 2015).

The mixed objective optimization (*Fig. 5.2*), as was evident with NLOPT, yields the expected optimal control profile as established by Howlett and Pudney (1995).

In practical initial open loop testing, the IPOPT solver could solve small scale problems with a coarse horizon to a small duality gap and relatively quickly. Regardless, when converted to a closed-loop architecture the performance was not reliable in terms of the optimization time, solution quality and solver feasibility (the ability to find a feasible point that satisfies all problem constraints). In many cases the solver would terminate sub-optimally (with a very large predicted duality gap) in the allowed optimization time or with an infeasible solution regardless of the subsolvers used. This causes quite aggressive closed-loop behavior and the complete controller performance was quire poor, unreliable, and very slow in energy efficient modes. Therefore, since reliable real-time control is a primary design criteria of the proposed closed-loop control architecture, there is great motivation in abandoning IPOPT in favor of more robust convex routines. Before this is possible, *Eqn. 4.1* needs to be reformulated as a convex optimization problem.

Chapter 6

Convex Optimization Formulation

6.1 Convex SOCP

Eqn. 4.1 is a non-convex optimization problem and therefore suffers from the inherent issues associated with these classes of problems. Firstly, given the definition of convexity, the non-convex control formulation may not be able to attain the global optimum at each optimization cycle, whereas any local optima for a convex problem (given strong duality), by definition, is a global minima (Hiriart-Urruty and Lemaréchal, 2013). The way that this issue can be addressed for a non-convex problem, to a degree, is via a multi-start approach given a random or pseudo-random selection of the initial problem guess. Due to the real-time sensitive and robustness requirements of an efficient and safe train optimal controller, multi-starting a sufficient number of starting points such as to, with a sufficient probability, approach the global optimum is not realizable (Martí *et al.*, 2013). This problem is not present with convex optimization strategies. On a practical consideration, non-convex solvers are fewer in number, less efficient and inherently slower than their convex counterparts

(Hiriart-Urruty and Lemaréchal, 2013).

Convex problems largely benefit from their highly scalable nature (Domahidi *et al.*, 2013; MOSEK ApS, 2015), allowing for fast solutions to vastly larger problems that, for the case of this control problem, allow for a more accurate determination of the optimal trip profile for a train. Therefore, an optimal train controller would be more robust if it were to be reformulated as a convex problem.

Eqn. 4.1 can be modified into a convex realization and structured as a second-order cone problem with a minimal reformulation into a relaxed problem, allowing not only for an approach that does not suffer from the same issues as the non-convex formulation but also allows for interesting and invaluable extensions to the problem. This is done by first replacing the control force Γ with the auxiliary variable γ, such that $\gamma = \Gamma/v$. Furthermore, an additional optimization variable ρ is introduced such that $\rho = 1/v$. Thus, the convex control problem will look as follows:

$$\underset{\rho,\gamma,\gamma^+\in\mathbb{R}^{K-1}}{\underset{v,t\in\mathbb{R}^K}{\text{minimize}}} f(z) = c_t + c_e = w_t t_K + \frac{\Delta_x^T(w_e\gamma^+)}{m} \tag{6.1a}$$

subject to:

$$\epsilon \leq v_i \leq V_{max}[i] \qquad i = 2,\ldots,K-1 \tag{6.1b}$$

$$v_1 = V_0 \qquad v_K = V_f = 0 \tag{6.1c}$$

$$t_1 = T_{start} \tag{6.1d}$$

$$T_{min} \leq t_K \leq T_{max} \tag{6.1e}$$

$$t_{i+1} = t_i + \rho_i\Delta_x[i] \qquad i = 1,\ldots,K-1 \tag{6.1f}$$

$$\left\| \begin{matrix} 2 \\ v_i - \rho_i \end{matrix} \right\|_2 \leq v_i + \rho_i \qquad i = 1,\ldots,K-1 \tag{6.1g}$$

$$F_{min}\rho \leq \gamma \leq F_{max}\rho \tag{6.1h}$$

$$0 \leq \gamma^{+} \qquad \gamma \leq \gamma^{+} \tag{6.1i}$$

$$\gamma_i \leq r_0\rho_i + r_v + r_{v2}v_i \qquad i = 1, \ldots, K-1 \tag{6.1j}$$

$$v_{i+1} = v_i + ((c_0 + G[i])\rho_i + \frac{\gamma_i}{m} + c_v + c_{v2}v_i)\Delta_x[i] \qquad i = 1, \ldots, K-1 \tag{6.1k}$$

The objective function remains unchanged from *Eqn. 4.1a* for the minimum time cost. Since Γ is no longer an optimization variable in the convex reformulation, an energy efficient similar objective of $c_e = (\Delta_x^T w_e \gamma^+)/m$ is presented instead, where γ^+ is the non-negative of γ ($\gamma^+ = \max(0, \gamma)$). Physically, this new objective represents represents a momentum minimization. Therefore, as with the non-convex case, through adjusting the ratios between w_t and w_e the switching point between the speedhold, coasting and braking regions can be tweaked. For active T_{min} and T_{max}, the energy objective weight w_e allows for effectively minimizing total energy expenditure as is demonstrated in the results section of this paper.

Thus mixed time and energy objective is combined to *Eqn. 6.1a*:

$$\underset{\rho,\gamma,\gamma^+ \in \mathbb{R}^{K-1}}{\underset{v,t \in \mathbb{R}^K}{\text{minimize}}} f(z) = c_t + c_e = w_t t_K + \Delta_x^T (w_e \frac{\gamma^+}{m})$$

Although the convex energy objective term is physically a momentum minimization, it can empirically and experimentally be shown that it behaves identically to the true non-convex energy strategy. *Fig. 6.1* compares the optimal solutions for a mixed objective given trip with $w_e/w_t = 0.01 \ldots 100$ where the non-convex problem was run through a multi-start strategy to increase the probability of converging to the globally optimal solution. The non-convex case was solved with the IPOPT solver, whereas the convex problem was solved with ECOS — an interior-point solver that is used and detailed extensively throughout the rest of this research. It can be seen that both the

convex and non-convex objectives converge to the same optimal solution $f(z^*)$. Due to the globally optimal nature of the convex formulation in some regions of *Fig. 6.1* the convex problem is more optimal than the non-convex objective.

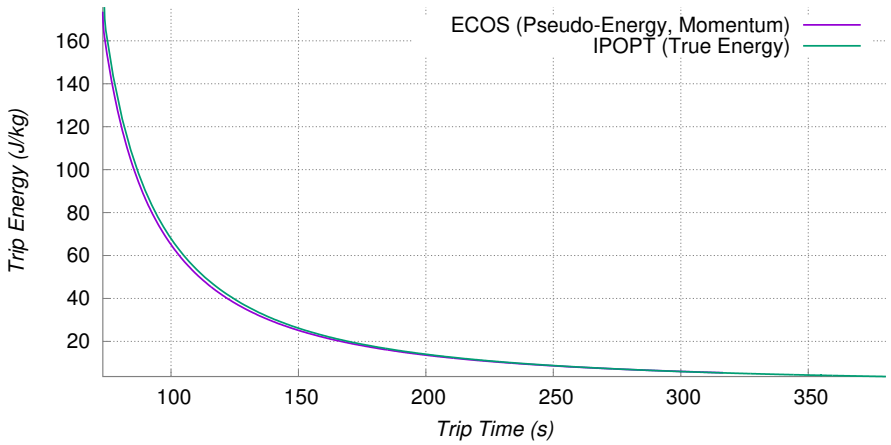


Figure 6.1: Optimal Solutions with Convex and Non-Convex Objectives

As a consequence of the different physical meanings and scaling of the energy minimization objective terms in the convex and non-convex case, it is expected that a different w_e/w_t ratio is required in the convex case for converging to the same optimal point as the non-convex case. This can be seen in *Fig. 6.2, and 6.3* .

These figures can be used to select the ratio of w_e/w_t such that the required balance between trip time and energy can be achieved for a problem with a loosely constrained final arrival time for both the convex and non-convex case.

The rest of the problem remains almost identical to the non-convex case, but with the new auxiliary variables substituted. The following constraints remain unchanged from *Eqn. 4.1* (*Eqn. 6.1b-e*):

$$\epsilon \leq v_i \leq V_{max}[i] \qquad i = 2, \ldots, K-1$$

$$v_1 = V_0 \qquad v_K = V_f = 0$$

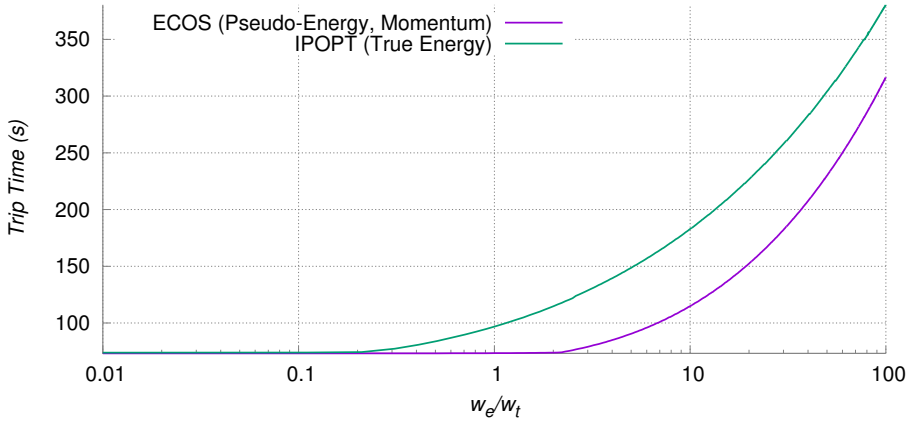


Figure 6.2: Optimal Solutions with Convex and Non-Convex Objectives: Arrival Time

$$t_1 = T_{start}$$

$$T_{min} \leq t_K \leq T_{max}$$

ρ is substituted into the recursive time equation *Eqn. 4.1e* to result in (*Eqn. 6.1f*):

$$t_{i+1} = t_i + \rho_i \Delta_x[i] \qquad i = 1, \ldots, K-1$$

$\rho = 1/v$ being a strictly non-convex constraint is first relaxed to the inequality $\rho \geq 1/v$ and then reformulated as an equivalent hyperbolic constraint. Thus it becomes a second-order cone constraint of the form (*Eqn. 6.1g*):

$$\left\| \begin{matrix} 2 \\ v_i - \rho_i \end{matrix} \right\|_2 \leq v_i + \rho_i \qquad i = 1, \ldots, K-1$$

With downwards pressure on ρ from the minimum time objective ($c_t = w_t t_K$) and no other interfering constraints on ρ, it can empirically be seen that for any given feasible problem where $c_t \gg 0$, $\rho^* = 1/v^*$ holds at optimum. It should also be noted that through empirical testing, downwards pressure from the minimum energy objective (c_e) can also reduce the $\rho^* - 1/v^*$ gap to a small margin (but not true equality) such

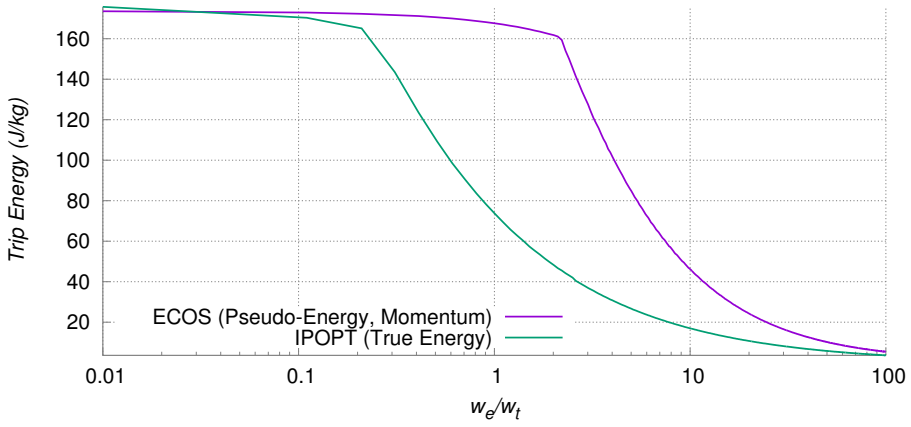


Figure 6.3: Optimal Solutions with Convex and Non-Convex Objectives: Trip Energy

that normal (but suboptimal) driving is apparent.

The additional benefit of a relaxation such as this is that even given infeasible control problems (e.g. the maximum arrival time T_{max} is below the minimum possible t_K) the controller will violate the activity of $\rho^* = 1/v^*$ resulting in a situation where $\rho^* > 1/v^*$, but maintaining a feasible control problem. In reality, this will be detected by the ATS and the problem specification can be altered to result in a feasible control problem. In this event the controller will continue functioning sub-optimally but safely until feasible trip specifications are supplied. The physical representation of this slack is a violation of the vehicle control input constraints through *Eqn. 6.1h*, and as such this equation would be relaxed to allow the controller to demand more propulsion or braking than is available. This in turn keeps the problem feasible in cases where the provided problem constraints result in a physically infeasible optimization problem. Since all problems provided to the optimal controller are strictly feasible this will not happen and $\rho^* = 1/v^*$ will be true at optimum.

Given the large number of complex constraints and great variability in vehicles,

trips and driving objectives an analytical proof for the stationarity at optimality of the hyperbolic constraint is far beyond the scope of this implementation oriented research and may be sufficiently difficult as to be practically not realizable.

Given that $\rho = 1/v$ and $\gamma = \Gamma/v$, *Eqn. 4.1h* can trivially be converted to the equivalent set of inequalities (*Eqn. 6.1h*):

$$F_{min}\rho \leq \gamma \leq F_{max}\rho$$

Since the non-negative control input is required for the energy efficient objective function it must be isolated. The following is equivalent to *Eqn. 4.1b* due to the fact that v is strictly positive as a result of *Eqn. 6.1b* (*Eqn. 6.1i*):

$$0 \leq \gamma^{+} \qquad \gamma \leq \gamma^{+}$$

The non-linear traction loss region model (*Eqn. 4.1i*) is converted to convex form through the substitution of γ and ρ (*Eqn. 6.1j*):

$$\gamma_i \leq r_0\rho_i + r_v + r_{v2}v_i \qquad\qquad i = 1, \ldots, K-1$$

The vehicle dynamics (*Eqn. 4.1j*) is converted to the affine representation below through the same substitution of γ and ρ (*Eqn. 6.1k*):

$$v_{i+1} = v_i + ((c_0 + G[i])\rho_i + \frac{\gamma_i}{m} + c_v + c_{v2}v_i)\Delta_x[i] \qquad\qquad i = 1, \ldots, K-1$$

The entire above formulation (*Eqn. 6.1*) is therefore in convex second-order conic problem form. This can be verified by comparing the problem structure to the SOCP review paper by Lobo *et al.* (1998). Therefore this problem can be numerically optimized by solvers capable of working with conic optimization problems.

6.2 Advanced Convex Controller Model

Although *Eqn. 6.1* is sufficient for closed-loop control, the convex formulation allows for additional controller features and extensions that can improve controller performance under extreme conditions such as:

- Highly restrictive speed profiles that force the optimal controller to drive as close as possible to the speed limit when running in time optimal mode or with a restrictive T_{max}.
- Cases where there is a substantial brake model or mass mismatch, resulting in an infeasible stop at a station due to the optimal controller planning an unachievable high brake rate in the braking region.
- Safely driving a train that has another train in front of it along the same route, and ensuring safety and optimality throughout the entire process.

Therefore, to achieve all of the above, the convex optimization is extended to address the mentioned issues with a substantial change for the control objective function $f(z)$ and the introduction of additional system constraints.

6.2.1 Overspeed Compensation

Although controller performance was acceptable in closed loop simulated tests, there is a significant issue in control which becomes apparent when a train is riding at or near the maximum permitted guideway speed V_{max} in the SPEEDHOLD region. As a result, any noise or model mismatch that forces the train's velocity to be registered as above the speed limit (whether it is in fact or the sensor noise makes the controller

think that this is the case) will force the optimal controller to react with an agressive response.

Large overspeeding can also cause an infeasible V_0 to be provided to the controller, if the speed is sufficiently above V_{max} such that the controller cannot feasibly reduce the velocity before the next discretization point along X. This is a potentially serious issue because the controller will not be able to optimize the trip without disabling the activity $\rho \geq 1/v$ set of constraints. Furthermore, of more serious concern is that any violation of V_{max} will cause the ATS to deploy the emergency brake.

A solution is presented via additional soft constraints on the speed limit and increasing objective penalties when the train velocity is within a soft limit of the allowed speed limit (V_{max}). Therefore a speed buffer (B_{os}), which represents a buffer below V_{max}, is introduced with the intended effect of adding cost penalties when exceeded and none when operating below, thus favoring control trajectories staying out of the buffer. Additionally, when a disturbance forces the vehicle's measured speed to violate this buffer, the resulting situation remains a feasible problem that the controller can cope with a preferably minimal amount of undesired braking.

A soft speed buffer was formulated in the controller through an additional cost term of the form:

$$c_{os} = \Delta_x^T w_{os} \max(v - (V_{max} - B_{os}), 0) \tag{6.3}$$

where B_{os} is the range of the penalty buffer (m/s). This additional objective evaluates to zero for $v \leq V_{max} - B_{os}$ or becomes a linear function of the amount of buffer violation otherwise. It can be equivalently represented in the standard form for a linear objective by introducing an auxiliary optimization variable b such that, through

an epigraph transformation:

$$c_{os} = w_{os}\Delta_x^T b \tag{6.4}$$

with the additional convex inequalities:

$$b \geq 0 \tag{6.5a}$$

$$b_i \geq v_i - (V_{max}[i] - B_{os}) \qquad i = 1, \ldots, K-1 \tag{6.5b}$$

The overspeed objective tuning weight (w_{os}) can then be empirically determined based on the required level of enforcement and required aggressiveness of the control response if the vehicle is in the buffer.

6.2.2 Reserve Braking

In normal train operation, the maximum braking is usually constrained to a fraction of the full emergency braking capabilities of vehicle (Wang *et al.*, 2011). Since *Eqn. 6.1a* does not penalize braking (although it does rigidly limit the maximum allowable braking), when a train approaches an edge where V_{max} decreases or the train is approaching a stop, the controller will hold off braking until the latest possible point (due to a minimum energy or minimum time control behavior). This can cause an overspeed if there is a significant modeling mismatch in the braking capabilities of the train or if the position estimation on the train is poor. Furthermore, very large braking can be quite uncomfortable to passengers. Therefore there is motivation in reserving some brake effort for the event of such an occurrence in a way that the controller can still demand extra braking if required, thus ensuring the control problem remains feasible.

By adding additional soft constraints on braking and adding a weight c_{brk} to the

cost, excessive braking can be minimized. In a similar way as overspeed compensation was implemented above, an over-brake penalty is introduced in the form of:

$$c_{brk} = \Delta_x^T w_{brk} \max(\gamma^+/m - \gamma/m - B_{brk}\rho, 0) \tag{6.6}$$

where B_{brk} is the desired maximum brake deceleration level (m/s^2) such that $B_{brk} < -F_{min}/m$. Therefore, $F_{min}/m - B_{brk}$ is the amount of braking that is reserved for the event of an emergency. For example: assuming that $B_{brk} = -0.8F_{min}/m$, the controller will design input trajectories that stay within 80% of the maximum possible brake level (for $w_{brk} \gg 0$). Thus when the vehicle approaches a pooled brake region the controller can demand up to an additional 20% braking effort if it is required. As in the overspeed case, an empirical selection of w_{brk} is required such as to force the optimal control solution vector (Γ^*) to be one where future brake effort is no greater in acceleration magnitude than B_{brk}.

As was required in the case of the overspeeding compensation, this objective is reformulated to a convex form in the same way as previously. In this formulation, an auxiliary variable h is introduced and c_{brk} is reformulated as:

$$c_{brk} = \Delta_x^T h \tag{6.7}$$

with the additional convex constraints on h of:

$$h \geq 0 \tag{6.8a}$$

$$h \geq \frac{\gamma^+}{m} - \frac{\gamma}{m} - B_{brk}\rho \tag{6.8b}$$

6.2.3 Successive Trains Simultaneously Sharing the Same Guideway

When maximizing headway (minimizing arrival time) a train scheduler will force a maximum number of trains onto the guideway. Thus, there may be instances where several trains are required to traverse an area between two stations one after another, effectively sharing the track (Cordeau *et al.*, 1998). This represents some safety concerns for an optimal control regime as an ideal optimal controller must be able to maintain some reasonable safety buffer between the vehicles to ensure safety and remove any chance of collision.

In the position domain it is quite straightforward to formulate a leading/following train model given the following assumptions:

- There can be any number of trains driving one after another.
- The motion of a following train (rear) is constrained by the motion of the leading train (front), where leading and following trains are as demonstrated in *Fig. 6.4* and *Fig. 6.5*.
- Leading trains may be optimally controlled, but this is not necessary for the optimal control of a following vehicle. Although this is the case, it is necessary for the following train to be able to predict the entire trajectory of its leading vehicle. Therefore, for this research, it is assumed that the leading train driving profile ($T_{lead}(x)$) is known or can be estimated to a reasonable accuracy.
- Every following train has feasible trip constraints (i.e. T_{min}, T_{max} such that following a leading train will not violate them).

- A required safety buffer distance (X_{sep}) is known, and can be either a constant scalar or a sampled vector of a distance separation function ($X_{sep}(x)$).

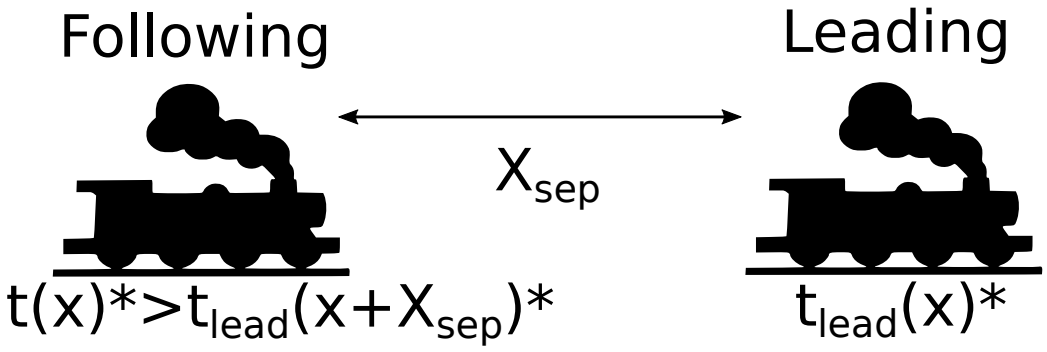


Figure 6.4: Leading and Following Train Model

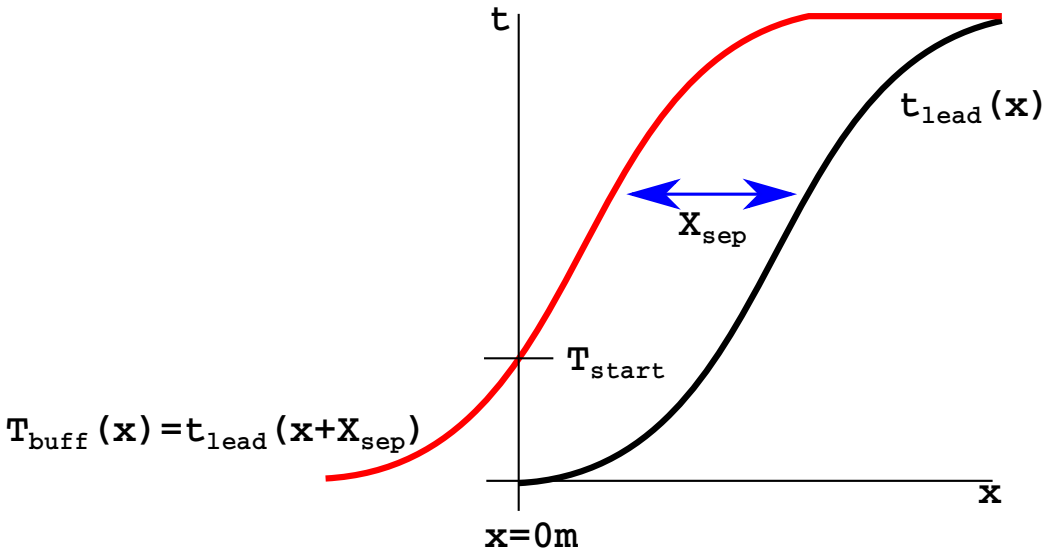


Figure 6.5: Following Train Time Constraint

Fig. 6.4 and Fig. 6.5 demonstrate how a leading train can be used to constrain a following train in the position domain. Since the optimal time profile ($t_{lead}(x)^*$) is known, it can be shifted back by the required municipal, customer or safety separation distance (X_{sep}) and redescretized along the position vector X of the following train

such that: $t \geq T_{buff}$ for $T_{buff} = t_{lead}(x + x_{sep})$. In order to accommodate more than two trains on a given track segment, simply each and every train must be running the optimal ATO and have a communication line with their immediate leading train. The leading train can then communicate back its $T(x)^*$ profile and the following trains will constrain themselves to avoid collisions with their leader. In this way the control strategy would propagate back to each and every train on the line. If the guideway is circular then the last train on the line will close the loop and become a leading train constraint for the first train on the guideway.

The constraints for the following train can be formulated as the convex inequality:

$$t_i \geq T_{buff}[i] \qquad i = 1, \ldots, K-1 \tag{6.9}$$

In this way the following train can drive optimally behind a constraining vehicle.

Complete Closed Loop Optimization with Advanced Objective

The complete closed-loop control objective with time (c_t), energy (c_e), overspeed buffer (c_{os}), and reserve braking (c_{br}) is:

$$\underset{\rho,\gamma,\gamma^+,b,h\in\mathbb{R}^{K-1}}{\underset{v,t\in\mathbb{R}^K}{\text{minimize}}} f(z) = w_t t_K + \Delta_x^T(w_e\frac{\gamma^+}{m} + w_{br}h + w_{os}b) \tag{6.10}$$

subject to: *Equations 6.1b-6.1k, 6.5, 6.8 and 6.9*

Eqn. 6.10 is henceforth referred to as the convex optimization formulation and the convex controller interchangeably and will be throughly tested and implemented in the subsequent chapters of this thesis.

Chapter 7

Numerical Solutions to the Convex Control Problem

The convex representation, as presented in the previous chapter, is both convex and in fact in disciplined convex programming form (DCP)(Diamond and Boyd, 2016). This is quite useful as it allows for the use of CVXpy, a Python disciplined convex programming modeling library, based on the popular CVX package for Matlab, used to easily interface a variety of convex problem structures to a wide array of potential numerical solvers. This allows for a rapid prototype of a convex program and easy testing under a wide variety of numerical solvers, where changing the program code to run with a different solver is as simple as changing one code parameter.

Although this is the case, CVXpy is also quite slow and computationally expensive since is does all of solver interfacing on its own thus providing the user with a simplified and intuitive rapid prototyping platform. It also does this from a high level language (Python in this case) thus allowing for a very simple and user friendly code structure. Therefore, in this research CVXpy is used to determine the most robust, fastest and

most scalable (preferably open source) solver to directly and efficiently interface and transform into an embedded closed-loop controller for a real commuter train. Of all of the solvers available for use in CVXpy only ECOS, MOSEK, SCS and CVXOPT are capable of solving the convex control problem since they all support conic problems (Diamond and Boyd, 2016). Since it is based on the matlab CVX package but is not a direct copy, some conic solvers native to CVX are not wrapped for CVXpy (the inverse is also true) (Diamond and Boyd, 2016) and as such were not tested in this thesis.

ECOS (Embedded Conic Solver) is a small footprint SOCP solver that uses a standard primal-dual Mehrotra predictor-corrector method with Nesterov-Todd scaling and self-dual embedding (Domahidi *et al.*, 2013). It is also fully open source and implements sparse martix routines for accelerated computation.

Mosek is a well known large scale commercial solver for solving LP, QP, SOCP, SDP and MIP problems (MOSEK ApS, 2015). The Mosek package contains both advanced interior point and sophisticated simplex methods for solving convex problems, but conic problems are solved with its own implementation of homogeneous and self-dual interior point algorithm (MOSEK ApS, 2015). Mosek utilizes optimized BLAS (Basic Linear Algebra Subroutines) for rapid computation of large scale problems.

SCS is a modestly accurate fist order conic solver and is the only one in the above list that is not an interior point algorithm. It uses its own operator splitting and homogeneous self-dual embedding to solve Symmetric, SOCP, SDP and exponential cone problems (ODonoghue *et al.*, 2016).

CVXOPT in contrast to all of the other solvers mentioned is implemented completely in Python. This means that embedding a convex controller paired with CVXOPT is not an option. It supports a wide range of convex optimization problems, but its conic solver is implemented purely in Python and it too uses an interior point method (M. S. Andersen and Vandenberghe, 2013).

7.1 Comparing Solvers in CVXpy

Solvers were compared based on their optimization time, scalability, and precision when solving the convex control problem. Each of CVXpy's SOCP capable solvers were configured to terminate when a duality gap of $|P^* - D^*| < 1 \times 10^{-8}$ is achieved, with all other solver settings left at default. The reader may note that this duality gap is far tighter than is necessary, but it should be noted that it takes the solvers an insignificant number of extra iterations to drop the duality gap from approximately 10^{-2} to 10^{-8}. All solvers were then run with the same problem to compute the open-loop initial optimization at $X_0 = 0m$ and with $K = 300$ where the resultant quality of the solution, optimal cost and computation time were compared. The trip and vehicle were configured as:

$$c_0 = -0.01\frac{m}{s^2},\ c_v = -0.001\frac{1}{s},\ c_{v2} = -0.0005\frac{1}{m}$$

$$r_0 = 2.23 \times 10^6 N,\ r_v = 0.08 \times 10^6\frac{kg}{s},\ r_{v2} = 0\frac{kg}{m}$$

$$F_{min} = -1 \times 10^6 N,\ F_{max} = 1 \times 10^6 N,\ T_{min} = 0s$$

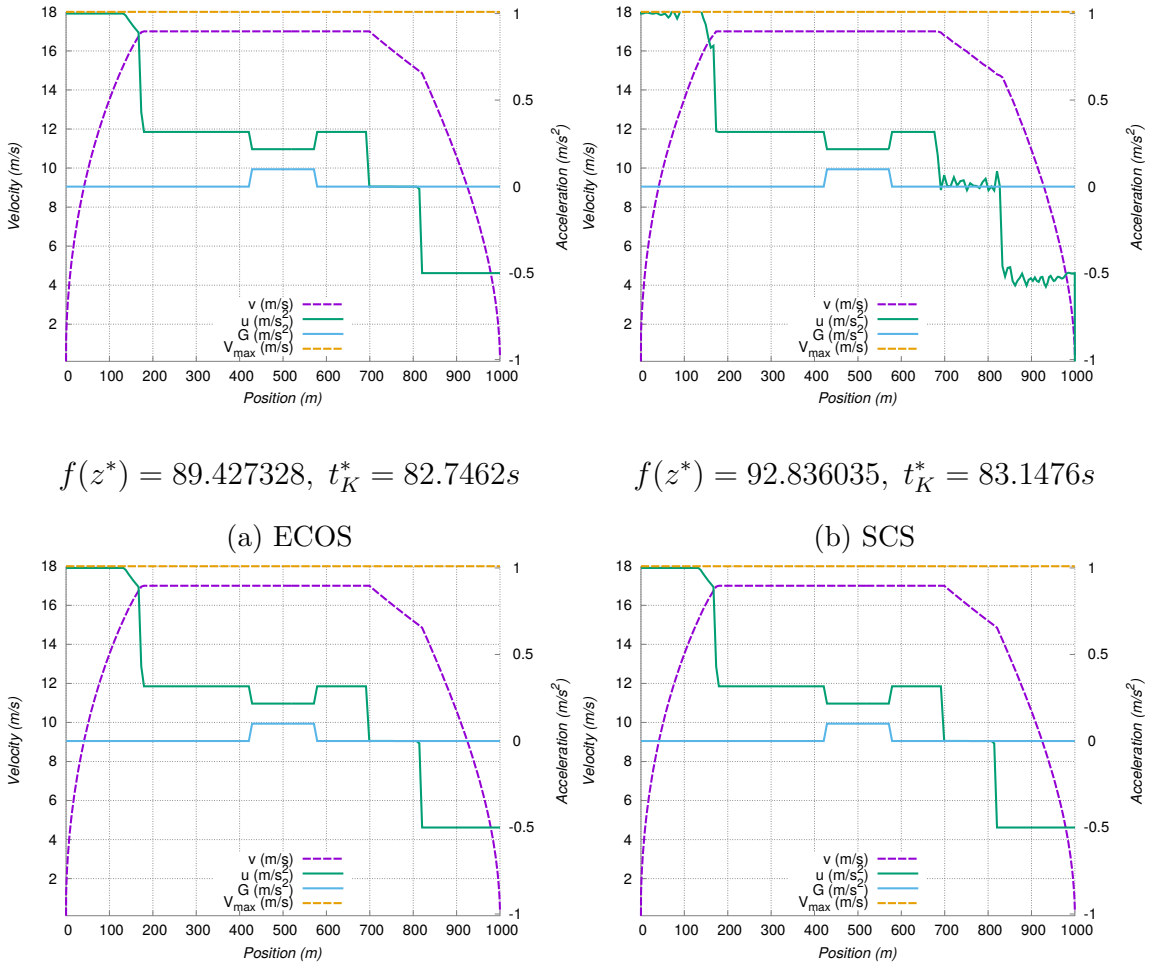
$$B_{os} = 1(m/s),\ B_{brk} = 0.5(m/s^2),\ m = 1 \times 10^6 kg$$

$$w_t = 1,\ w_e = 1,\ w_{os} = 0.02,\ w_{br} = 1$$

$$x_f = 1000m, T_{max} = 100s$$

$$g = \begin{cases} 0\frac{m}{s^2}\ for\ 0 \leq x \leq 420m \\ 0.1\frac{m}{s^2}\ for\ 420 \leq x \leq 480m \\ 0\frac{m}{s^2}\ for\ 480 \leq x \leq 1000m \end{cases}$$

$$V_{max} = \begin{cases} 18\frac{m}{s}\ for\ 0 \leq x \leq 1000m \end{cases}$$



$f(z^*) = 89.427328,\ t_K^* = 82.7462s$

(a) ECOS

$f(z^*) = 92.836035,\ t_K^* = 83.1476s$

(b) SCS

$f(z^*) = 89.427329,\ t_K^* = 82.7462s$

(c) MOSEK

$f(z^*) = 89.427329,\ t_K^* = 82.7462s$

(d) CVXOPT

Figure 7.1: Numeric Solvers

Fig. 7.1 all demonstrate an optimal solution of the MOTOR-SPEEDHOLD-COAST-BRAKE form due to a mixed time and energy optimization criterion that is equivalent to the results displayed in other research (Albrecht *et al.*, 2016a). It should be noted that there is a grade change in the trip and controller reduces motoring for the region to prevent an overspeed and to conserve energy. This, combined with there being a

significant COAST phase, supports the claim that the pseudo-energy objective term (c_e) promotes energy efficient driving behavior. Since w_e is substantially smaller than w_t in these tests the MOTOR phase is significantly longer than the Coast phase and the arrival times are all significantly lower than T_{max}. The overspeeding and over-braking constraints also work as expected as the vehicle drives no closer than $1m/s$ below the maximum allowed speed limit and reserves 50% of its maximum allowed braking.

It can be seen that ECOS, MOSEK and CVXOPT all converge to the same $f(z^*)$ (to a constraint violation of no more than 1×10^{-8}), whereas SCS returns similar yet very coarse results. Given the description of SCS as being an "experimental solver designed to solve larger problems than other CVX solvers typically handle, but to more modest levels of accuracy" (ODonoghue *et al.*, 2016) it does seem that the solver lacks the precision needed for solving the train control problem to a satisfactory level. This can also be seen in its resulting optimal objective value as it is substantially different from that of the other solvers.

Under all of the tested solvers the optimal solution was one in which $\rho \geq 1/v$ was active with a maximum violation of $|\rho_i^* - 1/v_i^*| \leq 10^{-8} \quad for \quad i = 0 \dots K-1$, thus supporting the claim for this constraint being active at optimum.

7.2 Comparing Real-time Aspects of Solving the Convex Problem

Fig. 7.1 only demonstrates the solution to the convex control problem but it does not portray the computation time required in obtaining the solution. Thus, computation

time for solving the problem in *Fig. 7.1* is presented in *Fig. 7.2, 7.3, and 7.4* .

In *Fig. 7.2 and Fig. 7.3* it can be seen that CVXOPT is the slowest solver tested — most likely due to being written in a high level language (Python) (M. S. Andersen and Vandenberghe, 2013). It is so much slower than the other solvers that it is not viable for use as a feedback controller.

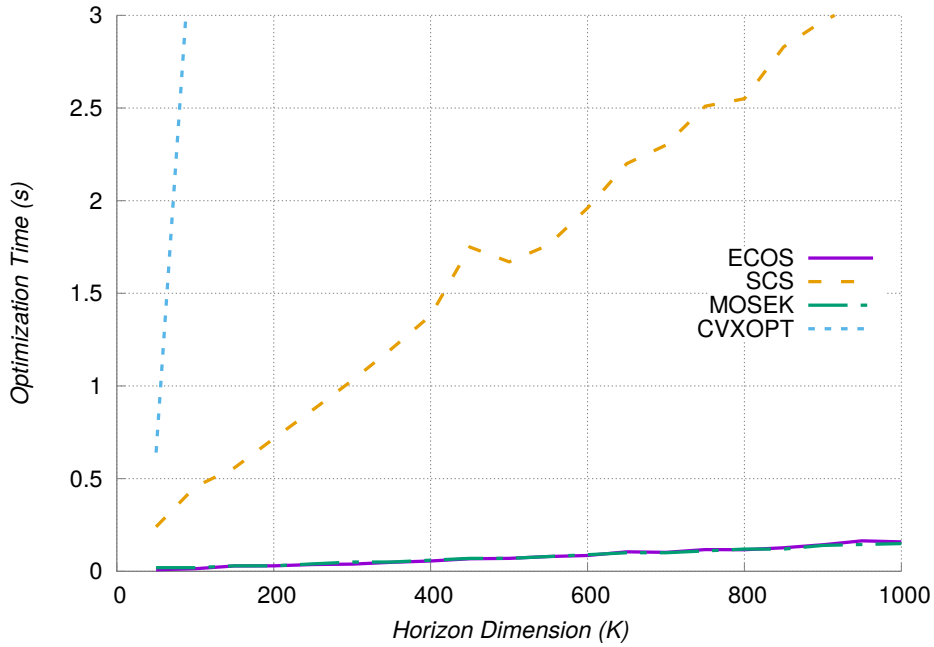


Figure 7.2: CVXpy Solver Time

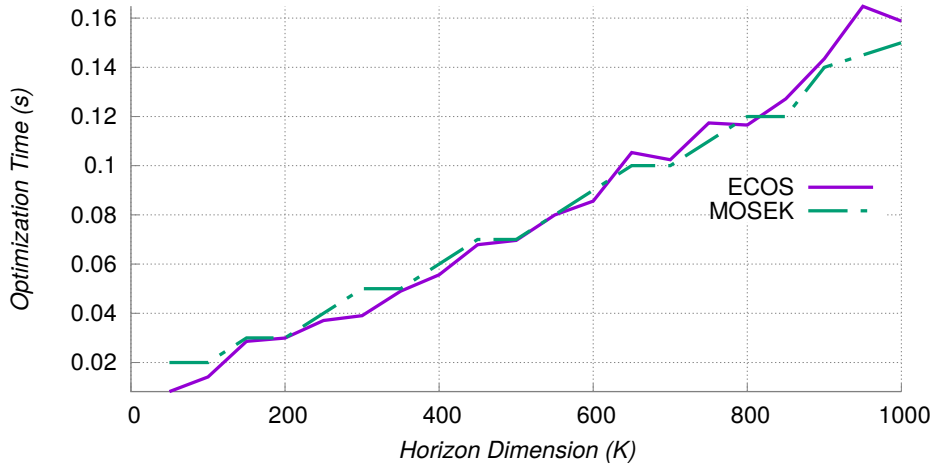


Figure 7.3: CVXpy Comparison of ECOS and MOSEK

ECOS and MOSEK are seen to be the fastest of all of the solvers tested and are very comparable in both primal solution and solve time. Both exhibited nearly linear scaling for the horizon dimensions tested in *Fig. 7.3 and Fig. 7.4*. For larger problem sizes, MOSEK was observed to be notably faster than ECOS. These observations are comparable to the results of the tests performed by Domahidi *et al.* (2013), the authors of the ECOS project (Domahidi *et al.*, 2013). Hardware optimized linear algebra routines (BLAS and LAPACK) in MOSEK as compared to the minimal CSPARSE implementation in ECOS may have attributed to the faster MOSEK performance for large problem horizons.

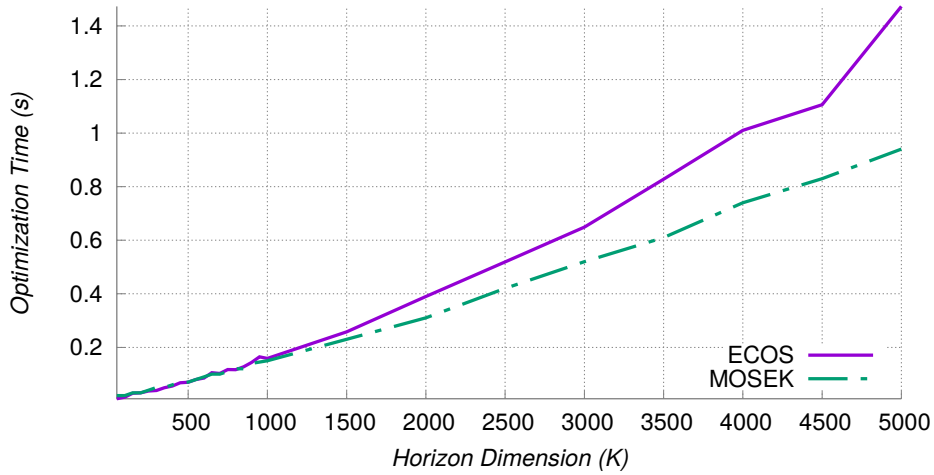


Figure 7.4: CVXpy Comparison of ECOS and MOSEK for Large Horizons

For the latter part of this research, ECOS was ultimately selected for development into a closed-loop control system. This was due to several significant reasons, staring with the fact that it is open-source (GNU GPL v3 licensed) whereas MOSEK is a closed source commercial product (MOSEK ApS, 2015). Additionally ECOS was designed from the ground up to be fully embeddable and is a sparse dedicated second-order-cone and exponential-cone solver comprised of less than 750 lines of C code (Domahidi *et al.*, 2013). Due to the open-source and library free nature of the solver, it is quite straightforward to compile or cross-compile the solver and the entire controller onto embedded platforms for field deployment. Furthermore, it uses a very efficient C library for sparse math (CSPARSE) routines including LDL factorization for the determination of search direction and does not require external dependencies, such as the BLAS and LAPACK dependencies for MOSEK (Domahidi *et al.*, 2013; MOSEK ApS, 2015). If required, the algebra subroutines in ECOS can be swapped to hardware optimized routines to allow for better or comparable performance to MOSEK for large problem (Domahidi *et al.*, 2013).

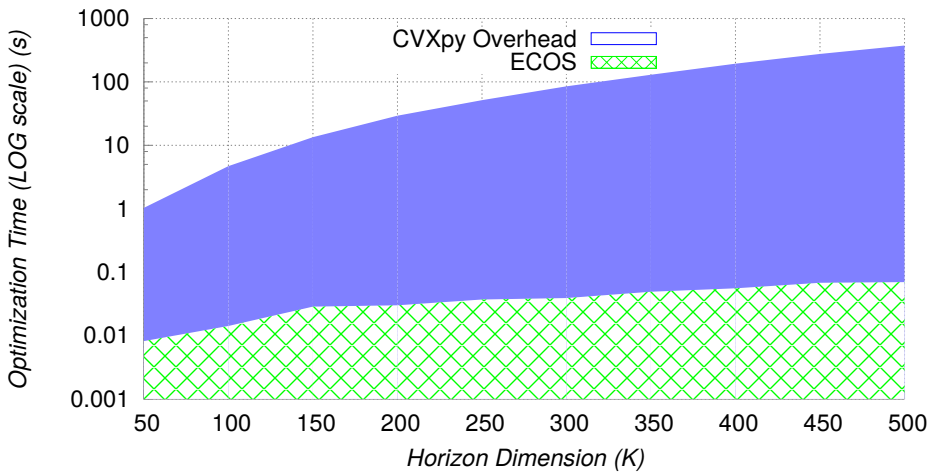


Figure 7.5: CVXpy Overhead

Fig. 7.5 demonstrates how much slower the CVXpy implementation of ECOS is as compared to the isolated ECOS solver. Since CVXPY takes in the optimization problem in Disciplined Convex Programming form, if takes a significant amount of time and computational resources for it to compute the true inputs to ECOS. In fact, the actual overhead time associated with CVXPY converting the DCP problem into the format used by ECOS was measured to be consistently no less than 100 times slower than the amount of time it takes ECOS to solve the problem. This is by no means a criticism of the CVXpy environment as much as it as an observation of the limitations associated with using a DCP environment for realtime optimization-something it was never intended to do in the first place. Therefore, if the problem can be supplied to ECOS in the exact form that the solver requires as input the huge overhead can be greatly minimized or eliminated altogether. Furthermore, interfacing the problem directly in C will further speed up both problem setup and solve time as the overhead due to CVXpy using the Python wrapped variant of the ECOS C library will be eliminated (Diamond and Boyd, 2016).

In order to interface the control problem to ECOS directly, the optimization must first be translated to general conic form, where the objectives and constraint can be supplied as matrices and vectors to the solver.

Chapter 8

Implementation of Optimal Control

8.1 SOCP In Standard form

To interface the control problem directly to ECOS's C API - so as to essentially eliminate problem setup time - the problem must be converted into a form that is used by the solver. In this case, the convex formulation must be converted into conic form with equality and inequality constraints (*Eqn. 8.1*) for cone $\mathbb{K}$ being either a non-negative orthant cone $\mathbb{K}_{++}$, second-order cone $\mathbb{K}_{so}$, or exponential cone $\mathbb{K}_{exp}$ (for geometric programming) (Domahidi *et al.*, 2013; Lobo *et al.*, 1998). In this problem $\mathbb{K}_{so}$ is used to define the hyperbolic constraint $\rho > 1/v$ through the matrix Q_{so}. The dimensionality of each conic constraint in Q_{so} is 3, thus each successive 3 rows of Q_{so} represent a Q_{so_i} that is equivalent to an individual $\rho_i \geq 1/v_i$ constraint.

The complete optimization problem is represented as the SOCP:

$$\begin{aligned} \underset{z,s}{\text{minimize}} \quad & w^T z \\ \text{subject to} \quad & Uz = y \\ & Qz + s = r, \qquad s \in \mathbb{K} \end{aligned} \tag{8.1}$$

Where:

$$z = [v_1, t_1, p_1, \gamma_1, \gamma_1^+, b_1, h_1, \quad \ldots, \\ v_{K-1}, t_{K-1}, p_{K-1}, \gamma_{K-1}, \gamma_{K-1}^+, b_{K-1}, h_{K-1}, \quad v_K, t_K] \tag{8.2}$$

$$w = [0, 0, 0, 0, \Delta_x[1]w_e, \Delta_x[1]w_{os}, \Delta_x[1]w_{br}, \quad \ldots, \\ 0, 0, 0, 0, \Delta_x[K-1]w_e, \Delta_x[K-1]w_{os}, \Delta_x[K-1]w_{br}, \quad 0, w_t] \tag{8.3}$$

$$Q = \begin{bmatrix} Q_{++} \\ Q_{so} \end{bmatrix} \qquad r = \begin{bmatrix} r_{++} \\ r_{so} \end{bmatrix} \tag{8.4}$$

Thus U and y represent the equalities *Eqn. 6.1c, d, f and k*; and Q and r represent the inequality *Eqn. 6.1b,e,g,h,i and j*, *Eqn. 6.4a-b*, *Eqn. 6.8a-b* and *Eqn. 6.9.* z denotes the primal variables and s denotes the conic slack variables (Domahidi *et al.*, 2013). Since all of the mentioned equality and inequality constraints are mostly vectorized along X at each i for $i = 0 \ldots K-1$ the matrices U and Q are largely block diagonal. U and Q are also quite sparse with approximately 0.046% and 0.025% fill, respectively.

8.2 Numerically Solving with ECOS

ECOS is configured to terminate an optimization once a sufficiently small duality gap is achieved ($|P^* - D^*| \leq 1 \times 10^{-8}$), returning the optimal primal variables z^*. As

a certificate of feasibility and optimality the dual variables λ^* and μ^* are returned thus validating that the required Karush-Kuhn-Tucker optimally conditions hold and that the resultant solution is optimal (Kuhn, 2014; Domahidi *et al.*, 2013). The dual problem to *Eqn. 8.1* is the following:

$$\begin{aligned} \underset{\lambda,\mu}{\text{maximize}} \quad & -y^T\lambda - r^T\mu \\ \text{subject to} \quad & Q^T\mu + U^T\lambda + w = 0 \\ & \mu \in \mathbb{K} \end{aligned} \tag{8.5}$$

Weak duality ($P^* \geq D^*$) is always assured for an SOCP problem. Strong duality only requires for either the primal or the dual to be strictly feasible (Domahidi *et al.*, 2013; Lobo *et al.*, 1998; Boyd and Vandenberghe, 2004). Since the convex formulation in this Thesis has a strictly feasible constraints set (for feasible trip conditions) with strictly feasible $Uz = y$ and relaxed inequality variables, both the primal and dual are attainable. In theory, a very tight arrival time window ($T_{max} - T_{min}$) combined with a poorly sampled control horizon (small K) may result a not strictly feasible $Uz = y$ constraint. As such, reasonable arrival time window limits and a large K should be provided to the controller to ensure that strong duality holds. Therefore, a globally optimal solution to the convex problem should be attainable for any realistic feasible trip scenario, as is evident by the implementation results and their associated certificates of optimality as provided by the ECOS output.

8.3 Implementation of Embedded Closed Loop Control

The complete closed-loop control law is implemented through *Alg. 1* which resides in the controller/vehicle model architecture as shown in *Fig. 4.1*. As was noted previously, U and Q are very sparse and thus take advantage of ECOS's sparse computation through the efficient CSPARSE routines (Domahidi *et al.*, 2013). A key concept to note in *Alg. 1* is the fact that at each optimization cycle the current position of the vehicle (X_0) is used to trim down X, G and V_{max}, thus resulting in the shrinking horizon as per *Eqn. 4.2*.

The optimal solution at each control cycle (z^*) is not used to warm-start the next optimization cycle. Although, in concept, this may seem to be inefficient, ECOS does not support warm starting. This is mainly because substantial benefit has not been demonstrated in warm-starting primal-dual interior-point methods due to the fact that these methods first work by finding a central path and then following that path to the solution. Since the optimal solution of the previous optimization is not on the central path for the new optimization simply starting from a random initial guess would have the same or better effect than would be reusing the old solution (John and Yıldırım, 2008). Although this is the case, other specialized IP methods have been able to demonstrate marginal reduction in computation time or solver iterations through specific variations on the interior-point method specifically for warm-starting (Wächter and Biegler, 2006; MOSEK ApS, 2015).

At the successful termination of the optimization for the current horizon, only the first immediate control decision is applied on the vehicle and must first be resolved

(as Γ is not an optimization variable in the convex form problem) by evaluating:

$$\Gamma_1 = \gamma_1^* v_1^* = z_1^* z_4^* \tag{8.6}$$

In terms of computation time, *Alg. 1* is implemented in such as a way to make the overhead time associated with solving the control problem a minimum. Unlike *Fig. 7.5* where the overhead in setting up the control problem was 100 times that of what it took for ECOS to solve the problem this implementation does not suffer from the same deficiency. In fact, no plots are presented for overhead time analysis of the C implementation simply because the setup is so fast that it cannot effectively be timed under Linux kernel even when using the high precision CLOCK_MONOTONIC timer in C's clock_gettime(3) method (Matagawa and Shudo, 2016). Therefore, there is sufficient evidence to claim that execution is sub-millisecond even for very large control horizon sizes. As such, the configuration time is considered negligible. Thus all subsequent timing results are presented as configuration time plus the ECOS solve time.

The fast problem setup can be attributed to several efficient strategies taken into consideration when designing and implementing *Alg. 1*. Firstly, the algorithm is implemented in C and compiled under the highest GCC optimization level (-03), naturally eliminating any language implementation slowdowns that where apparent when using Python. Furthermore, the problem is interfaced in the exact sparse form as required by ECOS's C API. Lastly, no further memory allocation or reallocation is performed after the initilization step, and whatever arrays and matrices that can be preserved between successive optimizations are preserved and have their access pointers moved forward as the control horizon truncates.

Algorithm 1 Optimal Controller

{Problem Initialization}
$X[i] = \sum_{j=1}^{i} \Delta_x[j] \qquad for \quad i = 1 \dots K$
$V_{max} \leftarrow [V_{max}(X[1]), V_{max}(X[2]), \dots, V_{max}(X[K])]$
$G \leftarrow [G(X[1]), G(X[2]), \dots, G(X[K-1])]$
$\Delta_t = 0.01$
sparse allocate: w, Q, r, U, y, z
$T = 0$
while $X_0 \leq X[K]$ OR $V_0 \geq 0$ **do**
  $V_0, X_0 \leftarrow$ read from vehicle sensors
  {Search for locations along discretization in X}
  for $i = 0; i \leq K; i++$ **do**
    if $X_0 > X[i]$ **then**
      BREAK
    end if
  end for
  {Recede the control horizon}
  ${}^*X[0] = {}^*X[i]; \qquad X[0] = X_0$
  ${}^*\Delta_x[0] = {}^*\Delta_x[i]; \qquad \Delta_x[0] = X[1] - X[0]$
  ${}^*V_{max}[0] = {}^*V_{max}[i]$
  ${}^*G[0] = {}^*G[i]$
  $T_{start} = T$
  evaluate w, Q, r, U, y
  $z \leftarrow 0$
  $F \leftarrow$ initialize ECOS with w, Q, r, U, y, z
  $z^*, f(z^*) \leftarrow$ solve F
  $\Gamma_1 \leftarrow z^*[1]z^*[4]$
  send Γ_1 to vehicle
  $T+ = \Delta_t$
end while

8.4 Vehicle Simulation

The convex controller was implemented in C with a direct interface to ECOS's C API with a simple wrapper then written in Lua to allow for easy access to the powerful Torch7 computation environment for simulation and analytics. All of the problem matrices were implemented as sparse triplet pairs using the CSPARSE sparse matrix data type. The entire formulation was then interfaced to a train model written in Lua and simulated as the ODE:

$$\frac{\partial x}{\partial t} = v \tag{8.7}$$

$$\frac{\partial v}{\partial t} = \frac{min(\Gamma_1, R(v))}{m} + c_0 + c_v v + c_{v2} v^2 + G(x) \tag{8.8}$$

where the ODE was simulated with a high frequency Euler integration (1kHz) rate. This assures high modeling accuracy. Since the optimal controller discretizes the track in position to a substantially lower precision some significant modeling uncertainty becomes inherently presented. Zero-order holding is performed on the control input Γ_1^* in simulation regions between successive $1/f_{ctl}$ loops. To allow for easier to follow plot, the model input control acceleration (u) is plotted in all future plots, where $u = \frac{min(\Gamma_1, R(v))}{m}$ and is the actual input acceleration experienced by the vehicle due to Γ_1.

As is evident in the next chapter, the controller was fully capable of smooth, high quality control even with this model mismatch.

Chapter 9

Controller Simulation Results

This chapter demonstrates the closed-loop performance of the convex optimal controller as implemented in *Alg. 1.* All of the tests below were conducted with $K = 500$ (unless explicitly stated otherwise) and a receding distance horizon X given X_0, V_0, and T_{start} as sensor measurements being fed back from the train model every 100ms with re-optimization at the same rate. The vehicle and trip were configured as:

$$c_0 = -0.01\frac{m}{s^2},\ c_v = -0.001\frac{1}{s},\ c_{v2} = -0.0005\frac{1}{m}$$

$$r_0 = 1.3 \times 10^6 N,\ r_v = 0.01 \times 10^6 \frac{kg}{s},\ r_{v2} = 0.002 \times 10^6 \frac{kg}{m}$$

$$F_{min} = -1 \times 10^6 N,\ F_{max} = 1 \times 10^6 N,\ T_{min} = 0s$$

$$B_{os} = 1\frac{m}{s},\ B_{brk} = 0.8\frac{m}{s^2},\ m = 1 \times 10^6 kg$$

Thus, the full friction and traction loss models are active and their affects should be apparent during higher speed driving. There is a soft buffer of $1m/s$ below V_{max} and 20% ($0.2m/s^2$) of the maximum braking capability of the vehicle is reserved for the event of an emergency.

9.1 Simple Closed Loop Case

The vehicle is first run on a simple and flat track with a constant speed limit. This is done to validate that the optimal controller works in closed-loop form and that the optimal driving strategy is representative of the optimal behavior noted by other research. The trip is configured as:

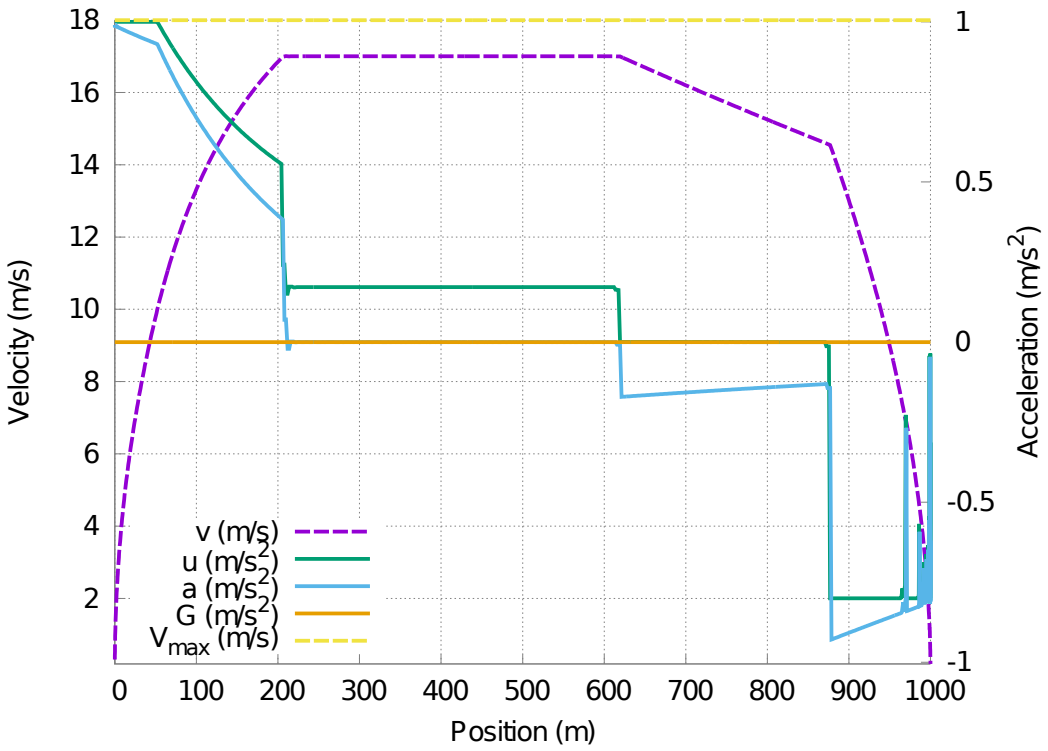
$$x_f = 1,000m,\ T_{max} = 200s$$

$$g = \begin{cases} 0m/s^2\ for\ 0 \leq x \leq 1,000m \end{cases}$$

$$V_{max} = \begin{cases} 18m/s\ for\ 0 \leq x \leq 1,000m \end{cases}$$

Under a zero grade trip, with a slack T_{max} and a constant V_{max} profile Howlett and Pudney (1995) theoretically proved that the energy optimal control strategy would be that of the form of MOTORING-SPEEDHOLD-COAST-BRAKE (Albrecht *et al.*, 2016a; Howlett and Pudney, 1995). The convex closed-loop controller also behaves as such given these same conditions as is apparent in *Fig. 9.1.* This also matches the results observed in both the convex and non-convex open-loop experiments. Thus, it can be claimed that similar performance as seen by Albrecht *et al.* (2016a), and Howlett and Pudney (1995) is observed without needing to structure a control problem to explicitly solve for optimal switching points (Albrecht *et al.*, 2016a; Howlett and Pudney, 1995).

The positive effect of the closed-loop reoptimization, although happening throughout the entire trip, can be seen very well as the train approaches the station stop. In the last 50m, the controller finely adjusts the braking. This results in high accuracy driving and a stop that is only 11cm ($X_s = 1000.11m$) past the target.



$w_t = 1,\ w_e = 1,\ w_{os} = 0.02,\ w_{br} = 1$
$T_f = 77.99s,\ E_f = 236.72J/kg,\ X_s = 1,000.11m$

Figure 9.1: Mixed Objective Trip on a Flat Grade

As modeling knowledge was accurate and no sensor uncertainty was present, the controller did not have to request any of the additional 20% braking effort that was reserved by the overbraking constraints and neither did it drive any closer than B_{os} to the speed limit.

9.2 Advanced Closed Loop Case

Since *Fig. 9.1* showed that the controller works as expected on a simple trip, a more advanced trip is configured to test how the optimal controller copes with aggressive changes in steep grade profiles while driving under a dynamic V_{max} profile for a

significantly long trip. This trip is configured as:

$$x_f = 1000m,\ T_{max} = 200m$$

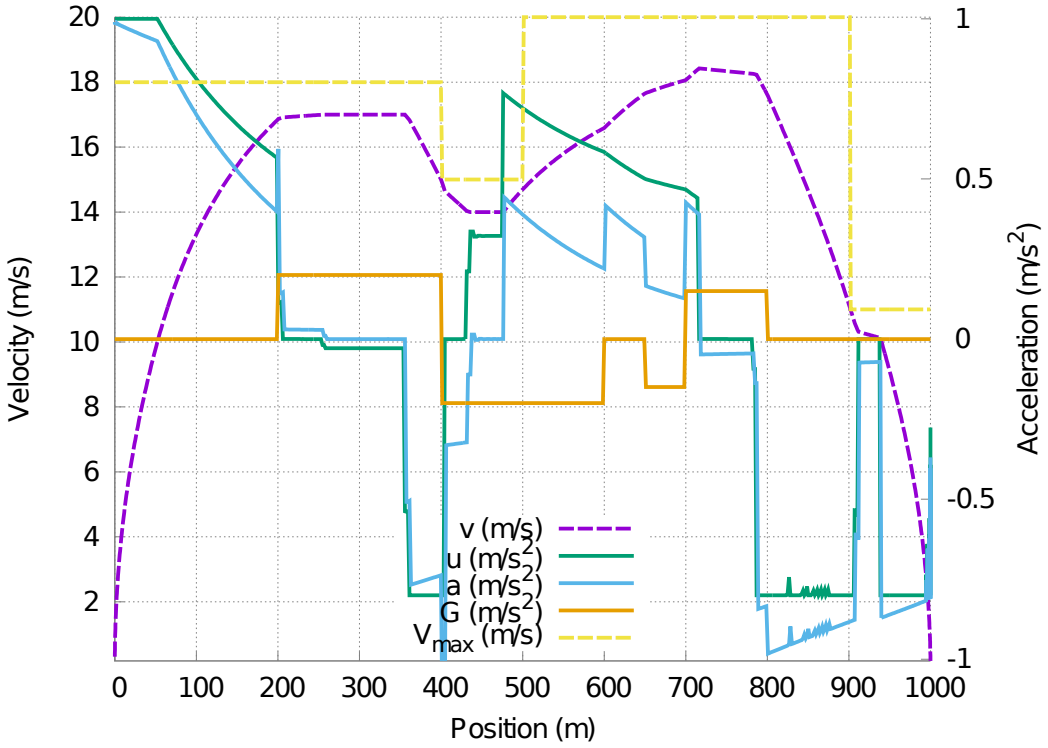
$$g = \begin{cases} 0m/s^2\ for\ 0 \leq x \leq 200m \\ 0.2m/s^2\ for\ 200 < x \leq 400m \\ -0.2m/s^2\ for\ 400 < x \leq 600m \\ 0m/s^2\ for\ 600 < x \leq 650m \\ -0.15m/s^2\ for\ 650 < x \leq 700m \\ 0.15m/s^2\ for\ 700 < x \leq 800m \\ 0m/s^2\ for\ 800 < x \leq 1,000m \end{cases}$$

$$V_{max} = \begin{cases} 18m/s\ for\ 0 \leq x \leq 400m \\ 15m/s\ for\ 400 < x \leq 500m \\ 20m/s\ for\ 500 < x \leq 900m \\ 11m/s\ for\ 900 < x \leq 1,000m \end{cases}$$

On an advanced track with a non-constant V_{max} profile, the MOTORING-SPEEDHOLD-COAST-BRAKE is no longer optimal as demonstrated by *Fig. 9.2, Fig. 9.3, Fig. 9.4 and Fig. 9.5*. It should be noted that the controller plans its course of action with a full image of the future track grade in mind. Thus it fully utilizes the acceleration and deceleration from the grade segments to further minimize the objective term to expend less energy and plans its actions in such a way as to ensure no overspeed is caused on steep downhill segments.

Fig. 9.2 demonstrates a time efficient control objective with a large weight on time and a small weight on energy with additional overspeed and brake reserve objective weights. It can be seen that the controller is largely driving as fast as possible within V_{max}, while holding outside the soft buffer region ($B_{os} = 1m/s$) as much as possible

and reserving 20% of braking for emergency situations.

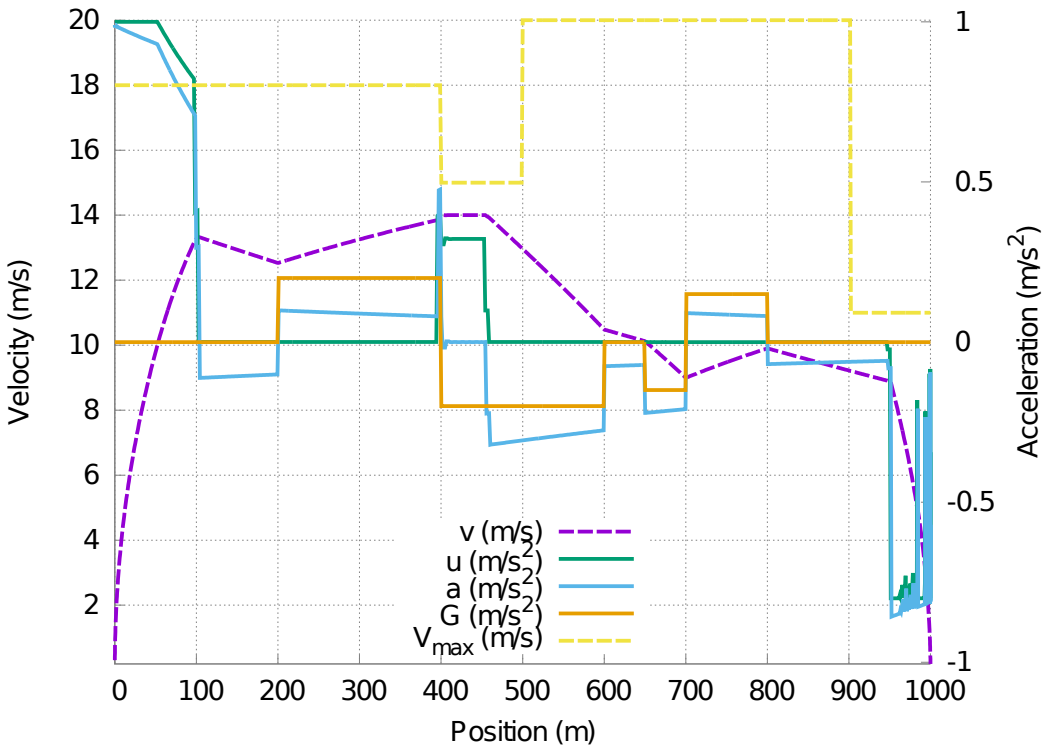


$w_t = 10,\ w_e = 1,\ w_{os} = 0.02,\ w_{br} = 1$
$T_f = 79.31s,\ E_f = 331.52J/kg,\ X_s = 999.94m$

Figure 9.2: Time Efficient Control for an Advanced Trip

Fig. 9.3, on the other hand, demonstrates more of an energy efficient objective. As such, it can be seen that the controller prefers extended coasting regions, and smartly relies on the grade profile for additional energy efficient speed-up in downhill regions. The energy expenditure is substantially reduced as compared to *Fig. 9.2* (115.34 J/kg as compared to 331.52 J/kg). As a result the arrival time is extended to 100.87s with the energy efficient objective as compared to 79.31s when in time efficient operating mode.

Fig. 9.4 demonstrates a case when equal weights are assigned to the energy and time objectives. As a result the final arrival time and total trip energy fall between



$w_t = 1,\ w_e = 10,\ w_{os} = 0.02,\ w_{br} = 1$
$T_f = 100.87s,\ E_f = 115.34J/kg,\ X_s = 999.97m$

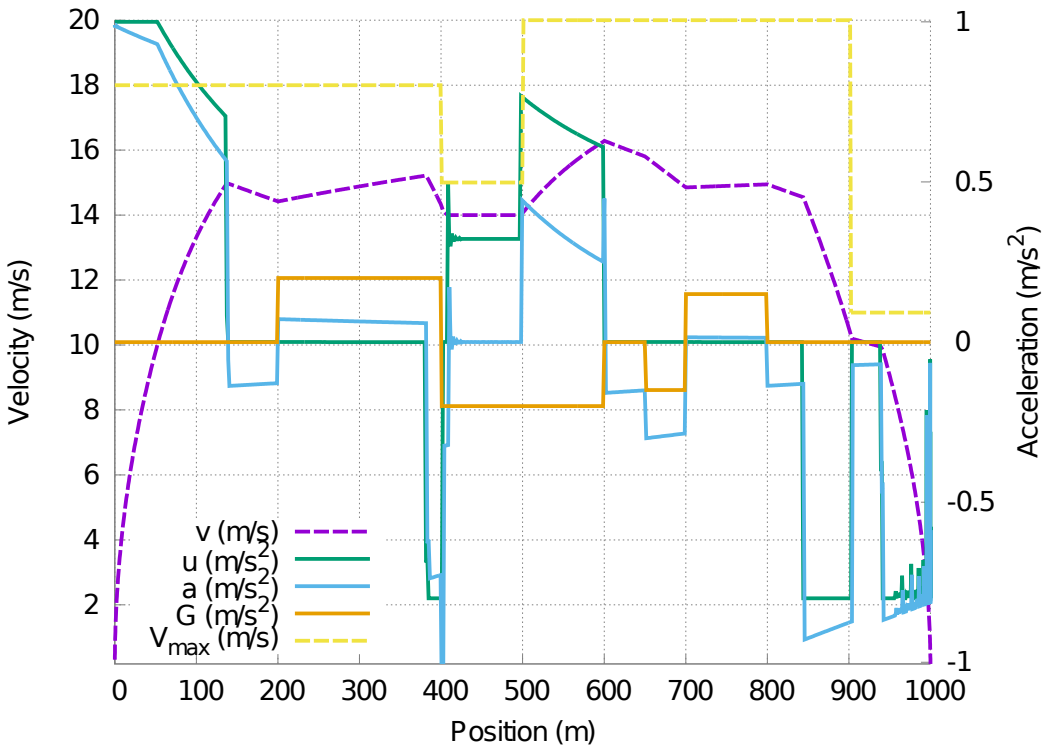
Figure 9.3: Energy Efficient Control for an Advanced Trip

those noted in *Fig. 9.2 and Fig. 9.3.*

All three of the above test cases (*Fig. 9.2, Fig. 9.3, and Fig. 9.4*) demonstrated very good stopping accuracy and complete adherence to the trip constraints.

9.3 Several Trains on a Single Guideway

To demonstrate the effectiveness of the controller in driving safely behind a leading train it is first necessary to launch a leading train running the optimal control system and then spawn a following train behind it. In order for the leading train to significantly impede the following (thus showing the full scale of the controller's ability to



$w_t = 1,\ w_e = 1,\ w_{os} = 0.02,\ w_{br} = 1$
$T_f = 84.92s,\ E_f = 222.12J/kg,\ X_s = 1,000.02m$

Figure 9.4: Mixed Objective Control for an Advanced Trip

handle such a case), the leader is put in a mixed time and energy efficient control regime (equal w_e and w_t), resulting in the closed-loop driving behavior as was evident in *Fig. 9.4*. Once the leading train has left the station, the following train is placed on the guideway. The following train is exactly the same in performance parameters as the leading but is set to run in time efficient mode, configured with the same objective as *Fig. 9.2*. This induces it to ride right up to the separation buffer as specified by the leading train.

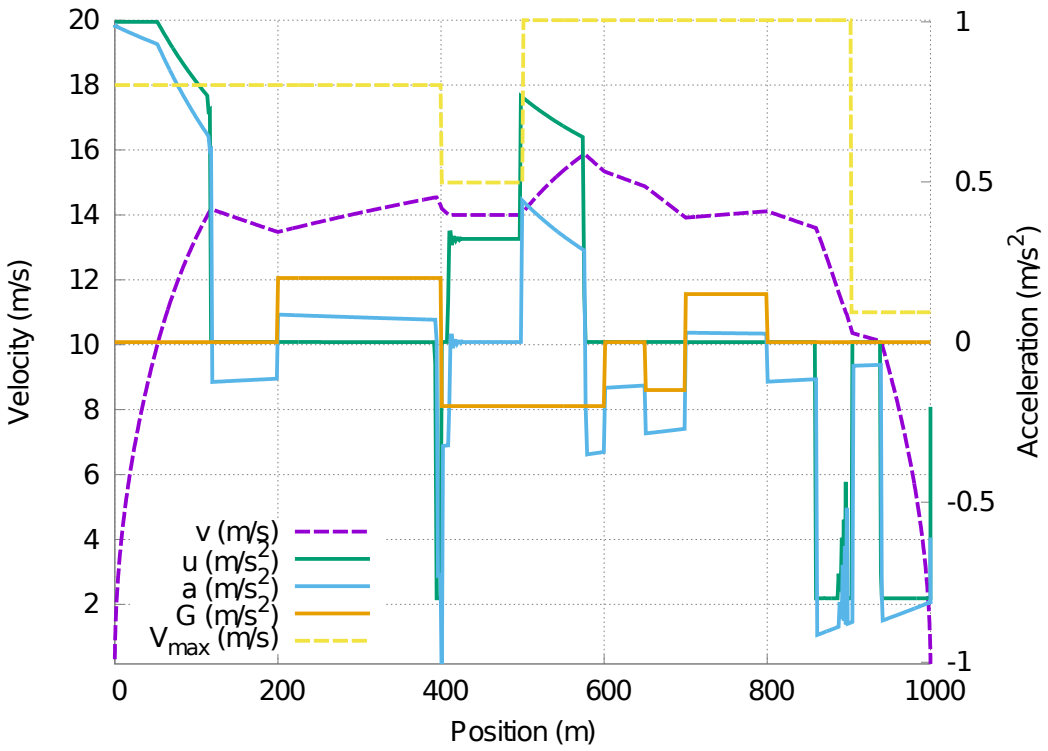
Since the two trains are identical in every way, had the separation gap constraints not been in place the two trains would have crashed as the front train takes 84.92s to complete the trip whereas the following would have taken 79.31s to complete the

same trip under the same circumstances due to their different objective functions. But this is not the case, as *Fig. 9.6* shows, since the following train does not cross over the T_{lead} buffer line and never approaches closer than 100m behind the leading, as required.

First, the following train waits until if can set off as it must wait until as least $T_{lead}[1]$ seconds. It should also be noted that only the arrival time t_K is being optimized for due to the objective in *Eqn. 6.10* and not times at any other $X[i]$ for $i = 1 \dots K-1$. Therefore there is no incentive for the following train to ride close to the buffer as its arrival up to at least 900m is constrained by the leading train. Therefore, since it had a mixed objective cost the controller, with no external intervention, minimizes the energy cost (c_e) since it cannot bring the arrival time down any more than $T_{lead}[K]$. This behavior can also be seen in *Fig. 9.5* as, despite having a larger time weight (w_t) than the leading, the following's resultant control profile looks like the leading's with an emphasis on bang-off-bang control strategies and thus it does not expend energy unnecessarily. Both leading and following vehicles drive optimally given their trip designs, and the following train uses knowledge of the leading vehicle's expected behavior to optimally plan and implement a safe control strategy.

9.4 Tight Arrival Time Constraints

The optimal control approach to the train control problem allows for substantially finer regulation of the vehicle's arrival time at a target station than does manual control (Novak *et al.*, 2015). As such, this criteria is implemented into the controller as *Eqn. 6.1e*. A closed-loop test is therefore ran to test this feature. *Fig. 9.7* demonstrates performance under a very tight 1 second arrival window of 79 to 80 seconds.



$w_t = 10,\ w_e = 1,\ w_{os} = 0.02,\ w_{br} = 1$
$T_f = 101.53s,\ E_f = 199.90J/kg,\ X_s = 999.87m$

Figure 9.5: Time Efficient Following Train

The controller is otherwise configured identically to that of *Fig. 9.4*. If not for the constraint from $T_{max} = 80s$ the vehicle would have arrived in 84.92s given its objective. This is evidently not the case as is visible in *Fig. 9.7* where the train arrives in 80.21s — an acceptable margin or error. This time is just outside of the target arrival time due to the fact that the mixed objective prefers an optimal solution where $t_K^* = T_{max}$ and consequently disturbances and modelling inaccuracies cause the vehicle to overshoot slightly. This is evidenced by the problem becoming infeasable at $x = 992.4m$. Regardless of the problem becoming infeasable, the hyperbolic constraint $p = 1/v$ relaxes so that $p^* > 1/v^* + \epsilon$ and control is still maintained with high stopping accuracy, resulting only in an insignificantly late arrival of less than a

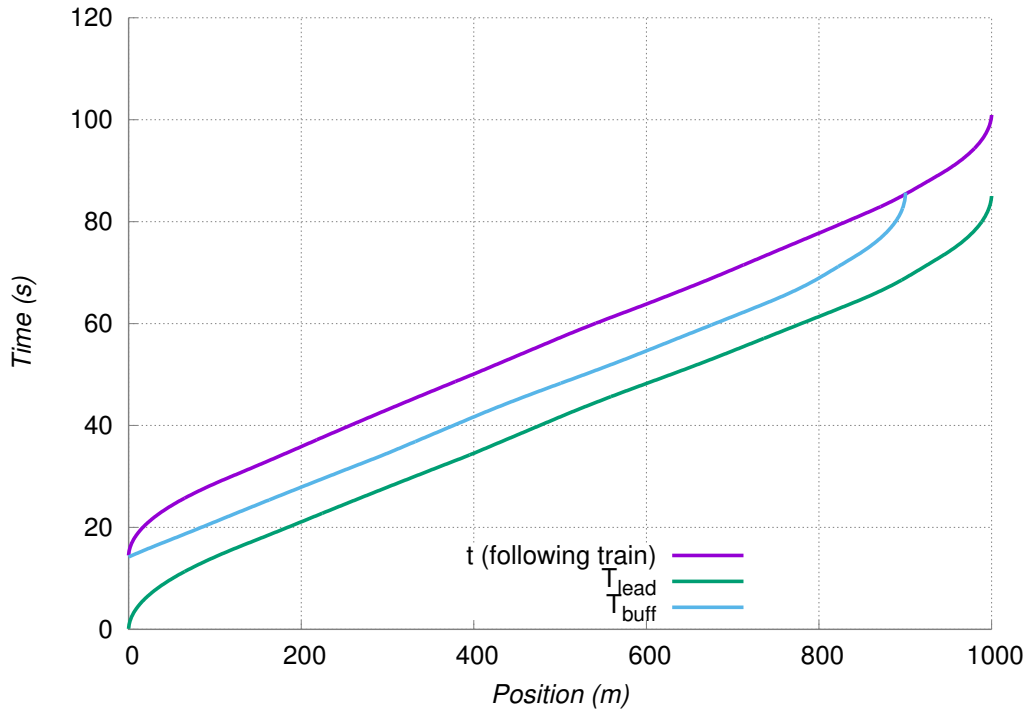
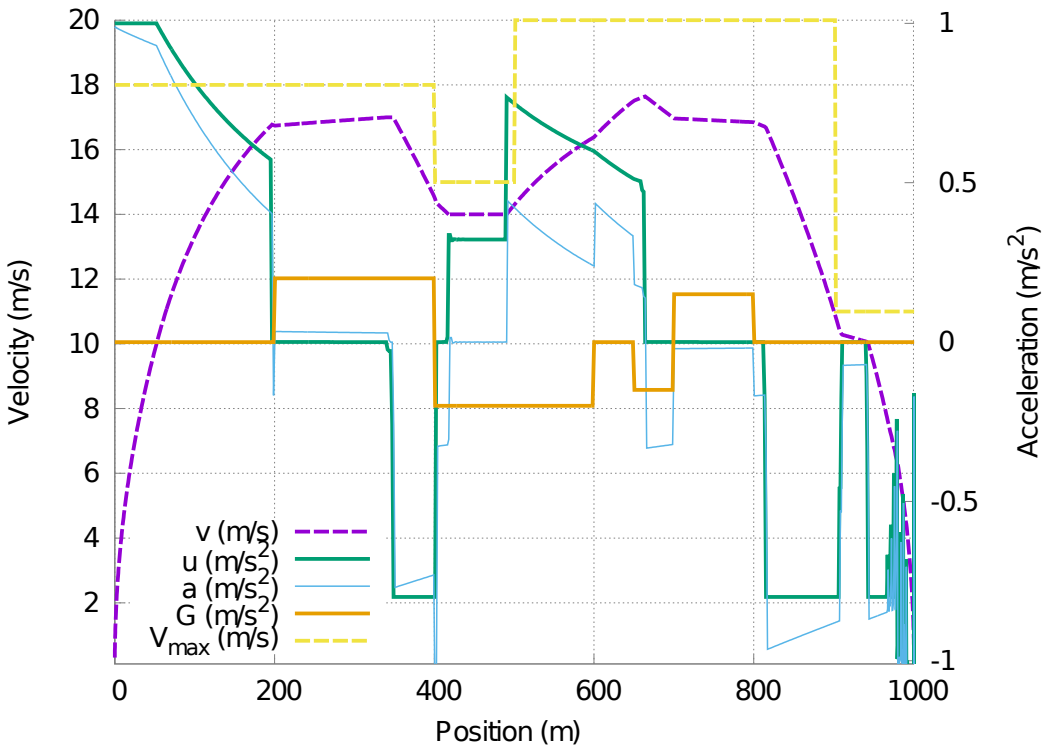


Figure 9.6: Both Leading and Following Trains on the Same Track

quarter of a second.

9.5 Long-Range Rail

As demonstrated previously, the convexity of the optimal controller allows for huge scalability for the control algorithm. Therefore, there is motivation in assessing controller performance when stations are very far apart. *Fig. 9.8* is a closed-loop example for a 10km trip with $K = 1,000$. It can be seen that control performance does not degrade when the controller is working with a control horizon of 10km as energy and time efficiency is maintained along with stopping accuracy. Regardless of the highly dynamic speed limit and grade profiles, the vehicle never overspeeds, control is fairly



$x_f = 1000m,\ T_{max} = 80s$

$w_t = 1,\ w_e = 1,\ w_{os} = 0.02,\ w_{br} = 1$

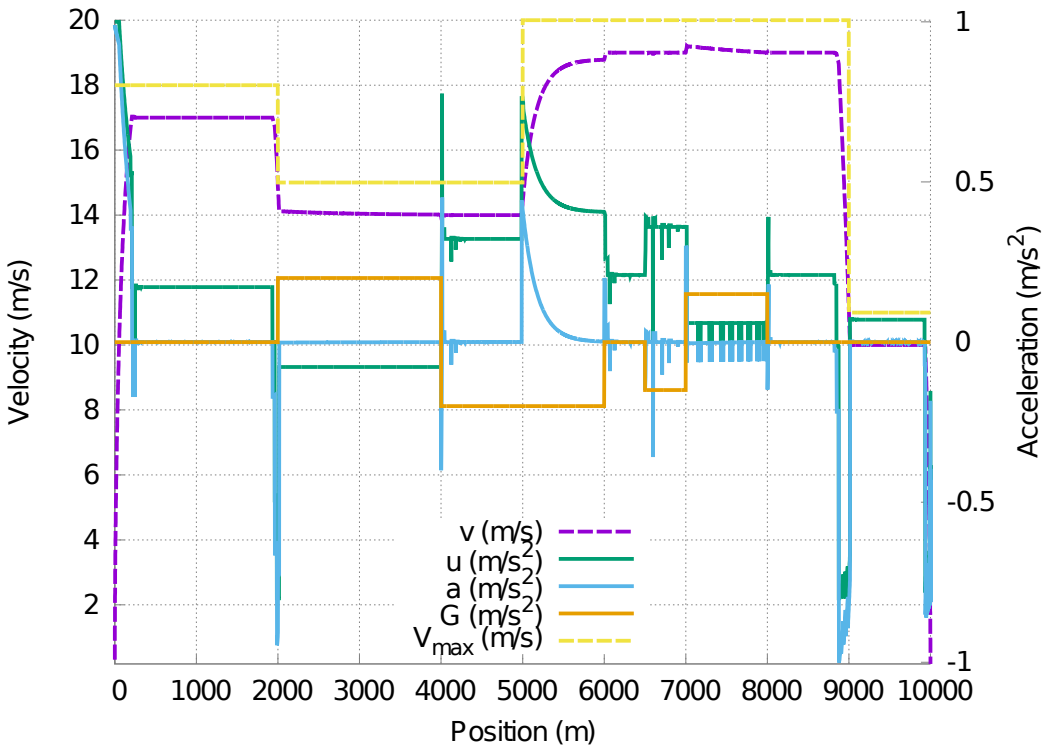
$T_f = 80.21s,\ E_f = 294.45J/kg,\ X_s = 1,000.24m$

Figure 9.7: Mixed Objective Control with Tight Arrival Time Constraints

smooth, and the final stop is within 9cm of error at the target station.

Fig. 9.9 demonstrates an even longer trip of 100km, again with a vehicle being controlled in closed-loop with a receding horizon spanning all of the way to the stopping station. Performance is again noted to be great, as less than 25cm of stopping error is witnessed after driving for 100km even with a fairly small K of 1000.

It should be noted that for very long trips, in cases where the arrival time is not constrained ($T_{min} = 0$, $T_{max} = \infty$), the controller does not necessarily need to optimize for the entire distance between the stations in one go. When this is the case the controller can work more similarly to rolling horizon window type control



$$x_f = 10km,\ T_{max} = 2,000s$$
$$w_t = 1,\ w_e = 0.1,\ w_{os} = 0.02,\ w_{br} = 1$$
$$T_f = 662.9s,\ E_f = 1,851.83J/kg,\ X_s = 9,999.91m$$

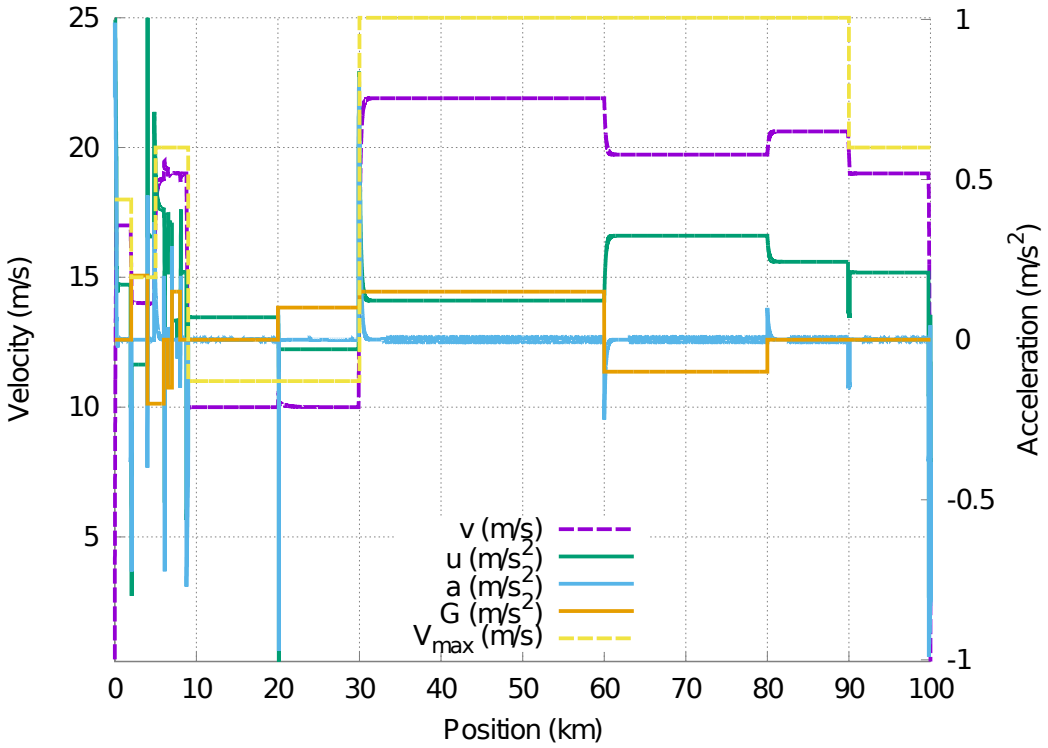
Figure 9.8: Time Efficient Long Range Trip (10km)

systems where the controller only plans for the next few kilometers ahead (smaller window distance than the full trip distance), with the terminal stopping constraint (*Eqn. 6.1c*) modified only for when the final stop is outside of the window, to the following:

$$v_1 = V_0 \qquad v_K \leq V_{max}[K] \tag{9.1}$$

This would unconstrain the final velocity of the vehicle to $V_{max}[K]$ at the end of the window allowing it to remain at high speed when the end of the window is still far from a stop. Once the rolling window becomes a receding window (when the station

stop overlaps with the end of the window) the above equation should be reverted back to *Eqn. 6.1c*, forcing a stop at the station. In this case, the time objective can be used to tune the unconstrained arrival time.



$$x_f = 100km,\ T_{max} = 20,000s$$
$$w_t = 1,\ w_e = 0.1,\ w_{os} = 0.02,\ w_{br} = 1$$
$$T_f = 6,076.07s,\ E_f = 17,216.93J/kg,\ X_s = 99,999.78m$$

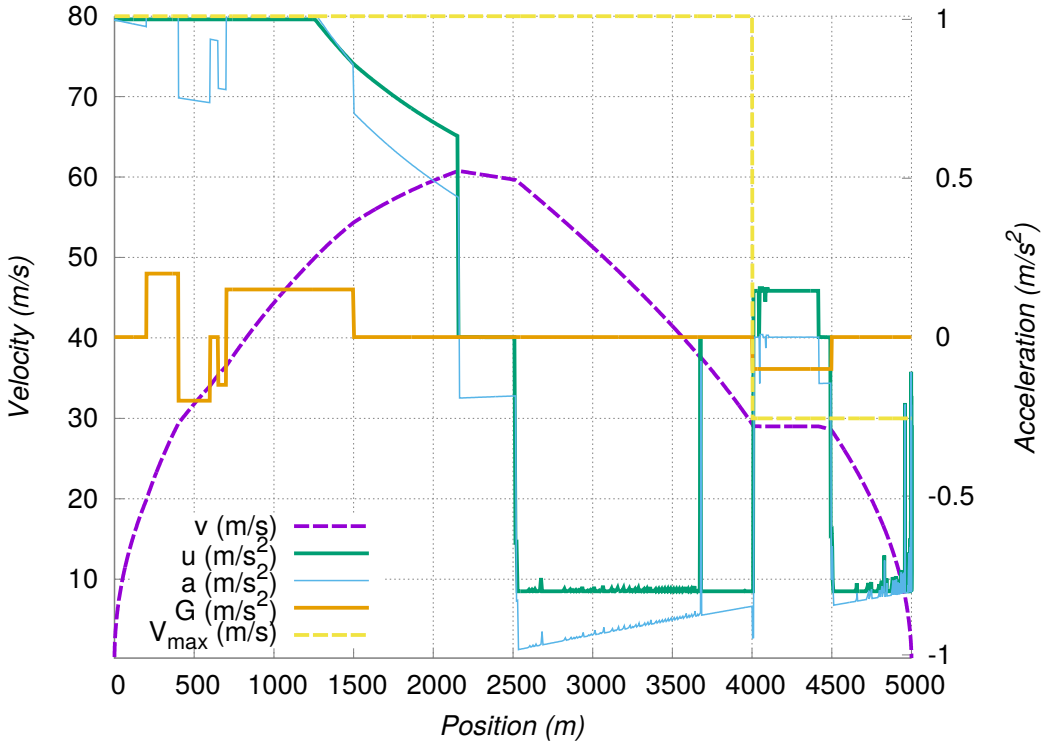
Figure 9.9: Time Efficient Long Range Trip (100km)

9.6 High Speed Rail

The optimal controller represented in thesis can also work with high speed rail vehicles equally as well as with light rail. *Fig. 9.10* represents a closed-loop control example of the performance on a vehicle with substantially lower drag and higher propulsion

capabilities than what was used in the previous tests. The vehicle for *Fig. 9.10* is configured as follows:

$$c_0 = -0.001\frac{m}{s^2},\ c_v = -1\times10^{-4}\frac{1}{s},\ c_{v2} = -5\times10^{-5}\frac{1}{m}$$
$$r_0 = 1.8\times10^6 N,\ r_v = 0.001\frac{kg}{s},\ r_{v2} = 3\times10^2\frac{kg}{m}$$
$$F_{min} = -1\times10^6 N,\ F_{max} = 1\times10^6 N,\ T_{min} = 0s$$
$$B_{os} = 1\frac{m}{s},\ B_{brk} = 0.8\frac{m}{s^2},\ m = 1\times10^6 kg$$



$$x_f = 5km,\ T_{max} = 300s$$
$$w_t = 1,\ w_e = 0.1,\ w_{os} = 0.02,\ w_{br} = 1$$
$$T_f = 159.02s,\ E_f = 2,091.4762J/kg,\ X_s = 4,999.958m$$

Figure 9.10: Time Efficient High Speed Rail Vehicle

Reaching a top speed of over 210km/hr - the vehicle in *Fig. 9.10* demonstrates the effectiveness of optimal control at high speeds. As with all of the previous test

cases, controller performance is very good and high accuracy stopping is observed.

9.7 Real-time Capability

Regardless of the large number of variables being optimized for ($z \in \mathbb{R}^{7k-6}$), the controller is real-time capable on both desktop and embedded processors as is shown in *Fig. 9.11, Fig. 9.12 and Fig. 9.13*. These figures demonstrate the computation time in solving a problem setup with the same parameters and trip as *Fig. 9.2* for various sizes of K in closed-loop form. Also demonstrated is how the optimization time decreases as the train approaches the target station due to the receding control horizon and the shrinking value of K. The non-linearities at small and large closed-loop iterations can largely be attributed to the chosen selection of the Δ_x profile being of the trapeziodal form as in *Fig. 3.2*. The noise in the plots can be attributed to the lack of any hard or soft real-time enforcement on the testing hardware. Regardless, controller reoptimization is still quite consistent in timing due the robustness and consistency of the ECOS solver.

9.7.1 Main-stream Desktop Processor

The two tested desktop CPUs were the AMD FX6300 and the Intel i7-6300. The AMD FX6300 is a 6-core CPU whereas the Intel i7-6300 was a 4-core, but this variation between the two processors is of little consequence as the entire implementation of the controller and the ECOS library are all in a single threaded mode of operation.

On a mid-high range Intel i7-6700, the worst case closed-loop optimization time is less than 0.25s for control horizons of $K = 2,000$. This represents a performance

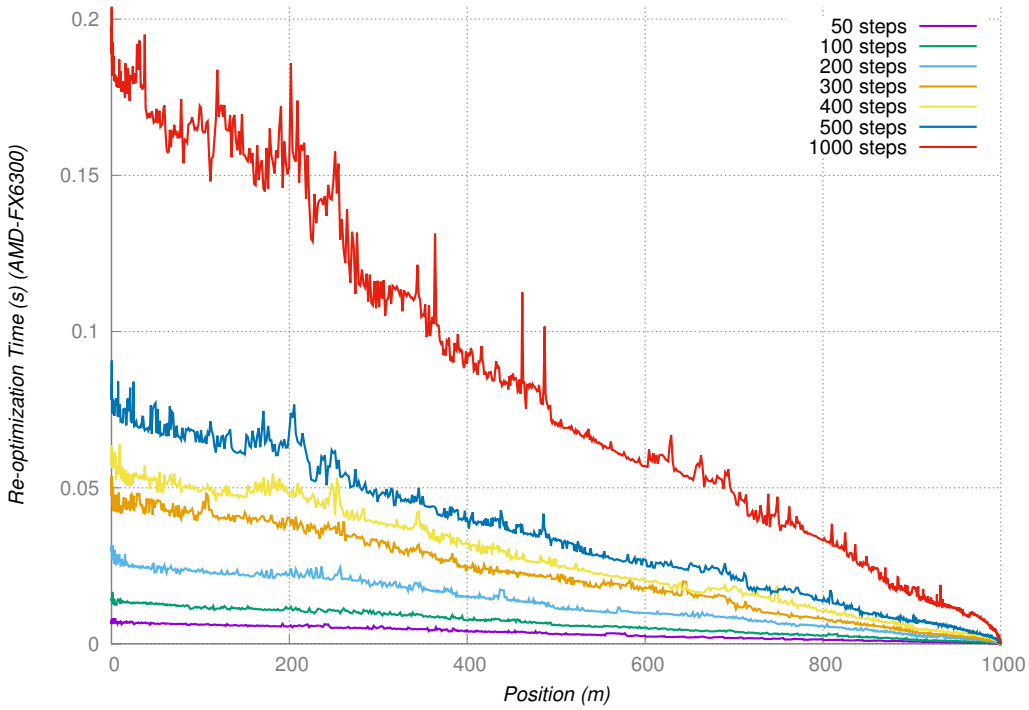


Figure 9.11: Cpu Performance - AMD FX6300

of about twice faster than the lower end AMD FX6300 for this horizon size. This large K is substantially larger than what is necessary to maintain robust control of a passenger locomotive, as was demonstrated in the above 100km driving test.

9.7.2 Embedded Applications

It can be shown that a full desktop i686 or amd64 architecture machine is not even required for robust control. Real-time performance tests were conducted on an embedded ARM architecture based Hardkernel Odroid XU4, which is similar in size and performance to the more familiar Raspberry Pi 3. As can be seen in *Fig. 9.13*, the controller is fully capable of real-time execution on a low power system (sub 15w) and dimensionally compact (83x58x22mm) package for significantly large control horizon

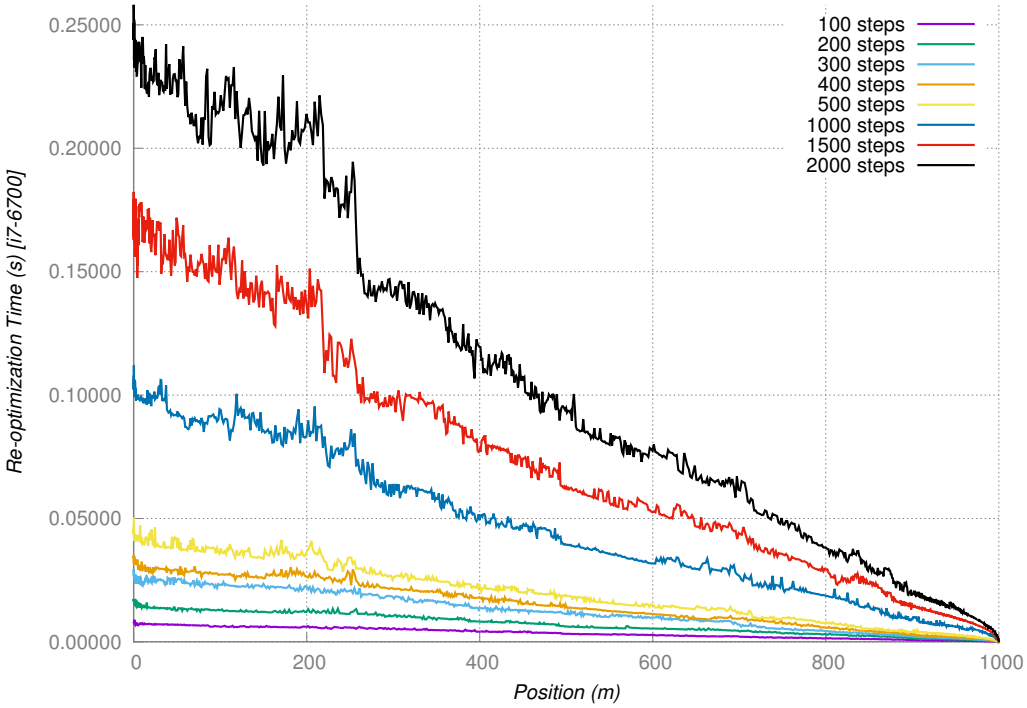


Figure 9.12: Cpu Performance - Intel i7 6700

dimensions (K) (Hardkernel, 2017). As discussed previously, a more optimal selection of the Δ_x profile may be able to significantly improve control accuracy with smaller K, allowing for even finer control on an embedded machine such as the Odroid.

The Odroid tests were performed while running under ARMv7 Linux kernel (Debian Jessie). Since ECOS and the entire implementation of *Alg. 1* are completely library free and platform independent, the convex optimal control can in practice be executed on true real-time hardware or soft real-time capable operating systems (RTOS) for real-world controller deployment on a vehicle.

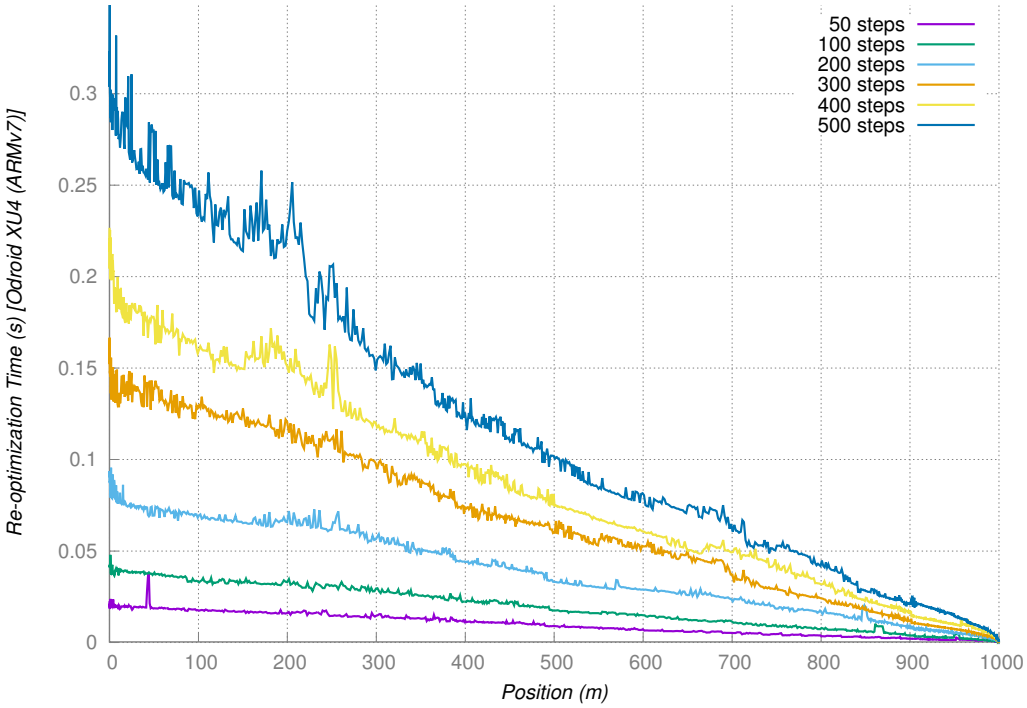


Figure 9.13: Cpu Performance - Hardkernel Odroid XU4

Chapter 10

System Performance in the Presence of Uncertainty

Since the whole purpose of a controller is to maintain control over a process under the presence of disturbance, there is great motivation in testing how well the optimal controller copes with various disturbances. Largely of interest here is to test performance under modeling and sensor uncertainty, where the modeling uncertainty can be either stochastic or a constant error between the model in the optimal controller and the actual vehicle.

10.1 Disturbance and Measurement Noise

To fully demonstrate the effectiveness of coping with disturbances the controller is tested in the vehicle with stochastic uncertainty models on both modeling and measurement uncertainty. The vehicle acceleration is simulated as an ODE as before, but

with the additional Gaussian process (white noise) of the form:

$$\frac{\partial v}{\partial t} = \frac{\min(\Gamma_1, R(v))}{m} + c_0 + c_v v + c_{v2} v^2 + G(x) + Y \tag{10.1}$$

where:

$$Y \sim \mathcal{N}(0, \sigma_a^2) \tag{10.2}$$

Additionally, a measurement noise is induced on the feedback speed reading ($\bar{V}$) and the position estimate ($\bar{X}$) that are also white noise processes:

$$\bar{V} \sim \mathcal{N}(V_0, \sigma_V^2) \tag{10.3}$$

$$\bar{X} \sim \mathcal{N}(X_0, \sigma_X^2) \tag{10.4}$$

The complete simulated system can be seen in *Fig. 10.1*, where $\bar{V}$ and $\bar{X}$ are used as feedback into the optimal controller as opposed to X_0 and V_0. This replicates having the controller reading system states from poor sensors and therefore getting a poor estimate of vehicle's speed and position.

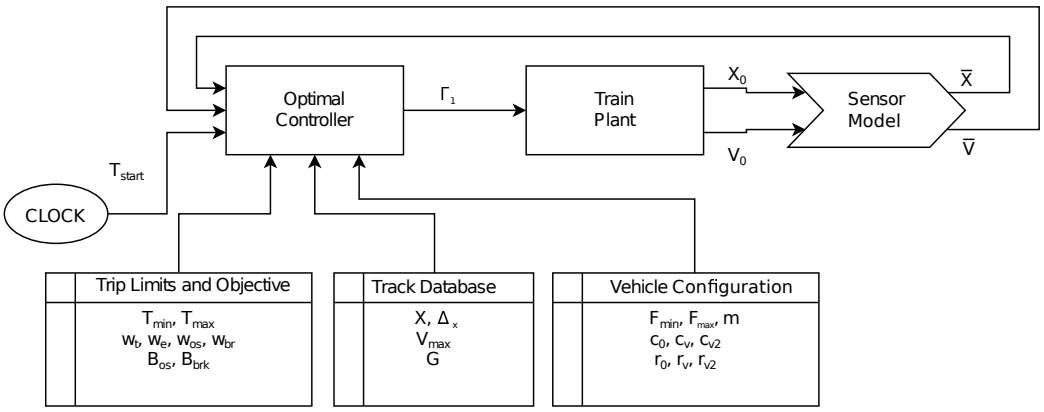
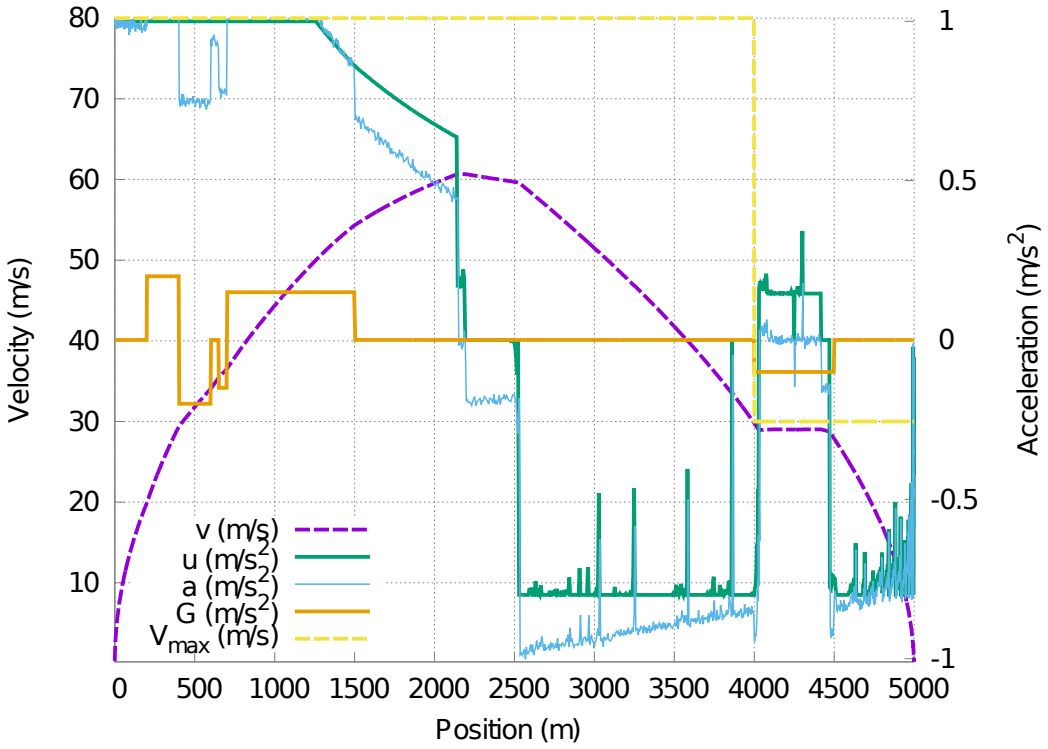


Figure 10.1: Simulation Model with Sensor Noise

For testing purposes the variances are initialized to the significantly large values as follows:

$$\sigma_a^2 = 0.05 m/s^2 \qquad \sigma_V^2 = 0.01 m/s \qquad \sigma_X^2 = 0.01 m$$

The acceleration modeling uncertainty is updated at a slow frequency of 1Hz. In this way, the effects of the acceleration uncertainty are not integrated out by the vehicle when evaluating the speed state. This represents physical uncertainty causing the vehicle's dynamics to be less predictable and thus introducing a stochastic plant/model mismatch. The speed and position measurement update rate is, on the other hand, set to the same frequency as the control feedback rate of 10Hz thus inducing a noise more similar to that of a hardware sensor.



$$x_f = 5km,\ T_{max} = 300s$$
$$w_t = 1,\ w_e = 0.1,\ w_{os} = 0.02,\ w_{br} = 1$$
$$T_f = 161.74s,\ E_f = 2,027.07J/kg,\ X_s = 4,999.87m$$

Figure 10.2: Large Gaussian System Noise

Fig. 10.2 demonstrates the convex optimal controller running in closed-loop form with the Gaussian vehicle uncertainty model as described above. As can be seen from

Fig. 10.2 that the uncertainty model does not lead to any overspeeding or final target overshoots even with the excessively noisy acceleration profile. The control output is also quite smooth and it can be seen that the low quality velocity estimation also does not cause the controller to output any undesirably aggressive changes in commands.

Under these test conditions stability is maintained and accurate stopping is observed. Thus there is some evidence that the closed-loop reoptimization strategy seems to be capable in suppressing stochastic modeling and measurement disturbance.

10.2 Model Parameter Mismatch

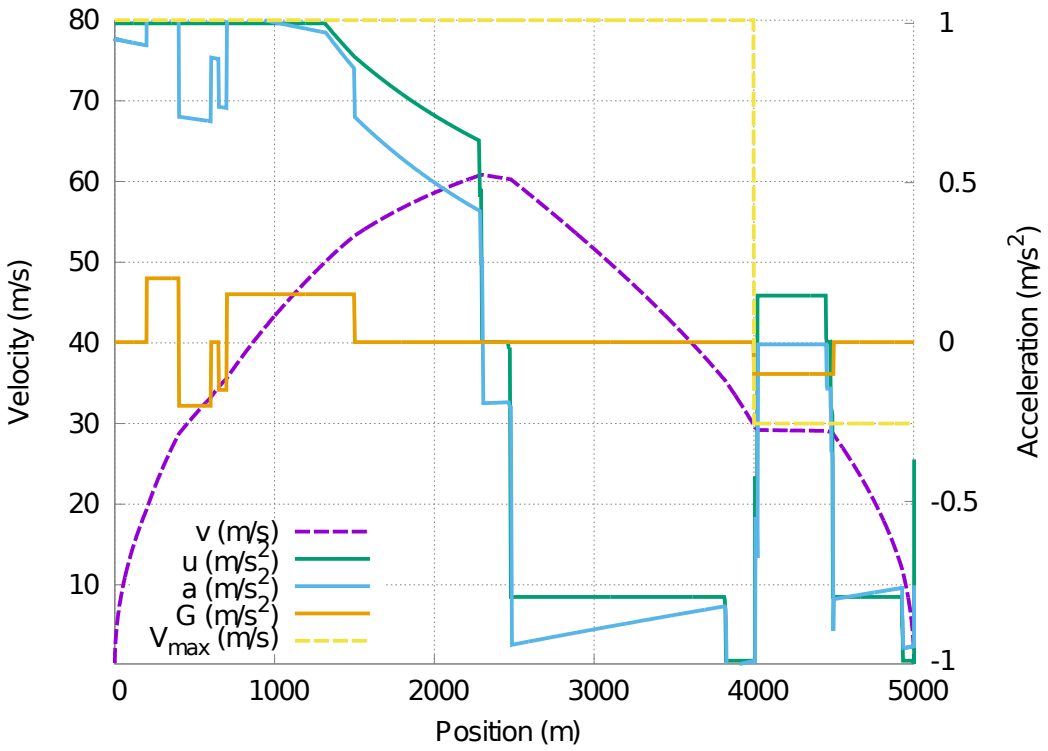
Model based control relies on the accurate knowledge of the vehicle or process being controlled. Although this is a requirement there may be instances where some model parameters may be hard to determine on a vehicle or may have changed since the last system identification. Take for instance the following scenario: Assuming that a vehicle model parameter is accurately determined during the systems identification phase, the mass may vary from trip to trip as the passenger loading levels change between different station runs.

In *Fig. 10.3* the mass provided to the controller is 5% lower than the real vehicle's mass (the controller thinks that the mass is $1 \times 10^6 kg$ whereas it is really $1.05 \times 10^6 kg$). This can represent an unmeasured change in passenger loading between different station runs (there are more passengers on-board than the controller is configured for). Subsequently, the controller's F_{min}/m and F_{max}/m result in control acceleration limits that are 5% higher than what the true vehicle is capable of. Thus every control action sent by the controller to the vehicle has a 5% lower impact on the vehicle

acceleration than was predicted.

Fig. 10.3 demonstrates the above example. It can be expected that the modeling uncertainty would degrade the control quality for an optimal controller, especially when the vehicle approaches decreasing edges in V_{max} or approaches the stopping point. This is not the case, as seen in *Fig. 10.3*. As can be observed, the overbraking constraints are helpful in this case as the controller starts commanding braking in the reserved region to make the heavier train stop with minimal overshoot. This is most visible on *Fig. 10.3* near 4km and once again near 5km as full braking is demanded near these regions. Therefore the overshoot for the final stop is minimized to less than half a meter (where it would have had 6.27m of overshoot without an active brake reserve objective) and generally good driving quality is maintained. Thus it can be seen that in closed-loop form, re-optimizing for the receding horizon can greatly help in handling modeling disturbance.

When a very large modeling mismatch is present, the controller is no longer able to accurately drive the train. As seen in *Fig. 10.4*, when subject to a 20% mass error (the controller thinks that the mass is $1 \times 10^6 kg$ whereas it is really $1.2 \times 10^6 kg$) the controller predicts incorrect optimal trajectories and the closed-loop nature of the controller is unable to compensate sufficiently. Thus the train violates the speed limit at the transition region and stops 35.74m past the target. This sort of behavior is common with model based controllers as their performance degrades when there is significant plant/model mismatch. Likewise, PID based strategies can also suffer in this way if operating outside of tuning bands (Jahanshahi and Skogestad, 2013).



$x_f = 5km,\ T_{max} = 300s$

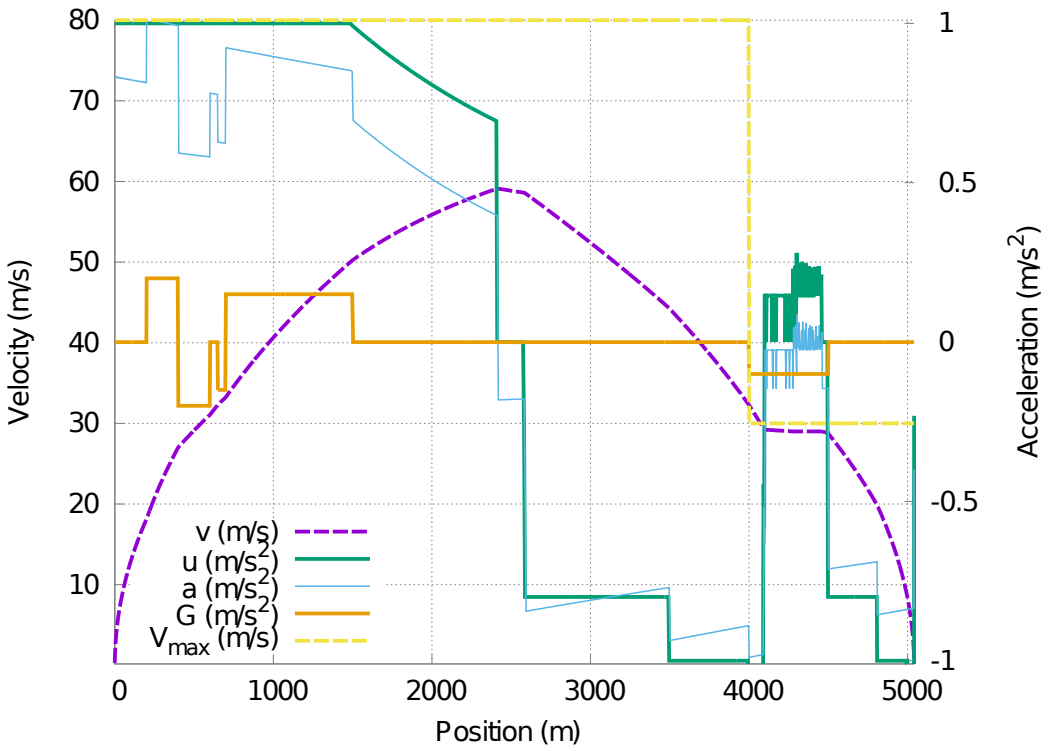
$w_t = 1,\ w_e = 0.1,\ w_{os} = 0.02,\ w_{br} = 1$

$T_f = 157.19s,\ E_f = 2,048.48J/kg,\ X_s = 5,000.484m$

Figure 10.3: 5% Mass error

10.3 Pairing the Optimal Controller With a Parameter Estimator

Since the optimal controller takes in the vehicle parameters as constants and reoptimizes a new trajectory for every closed-loop cycle, the controller can be paired with some form of parameter estimator and the constant parameters can be adjusted at each control cycle and used as inputs along with the driving measurements. Working together, the parameter estimator can feed in the latest optimal parameter estimates to the optimal controller which will subsequently use them to predict the next optimal



$$x_f = 5km,\ T_{max} = 300$$
$$w_t = 1,\ w_e = 0.1,\ w_{os} = 0.02,\ w_{br} = 1$$
$$T_f = 157.19s,\ E_f = 2,048.48J/kg,\ X_s = 5,035.74m$$

Figure 10.4: 20% Mass error

control strategy (Γ^*). Once the optimal estimator converges to the true estimate for the required parameters Γ^* will be the true optimal solution for the given vehicle.

To demonstrate the effectiveness and benefits of such an approach, the optimal controller will be paired with a simple Kalman Filter configured to act as a mass estimator. With many control strategies Kalman Filters have become the de-facto standard for both parameter and state estimation due to their simplicity, robustness, simple computation and optimality under Gaussian modeling and measurement uncertainty conditions (Wolpert and Ghahramani, 2000).

A Kalman filter is introduced to estimate the mass every $1/f_{ctl}$ seconds (every control cycle). The entire closed-loop simulation interface is altered to that in *Fig. 10.5*, where a Kalman filter takes $\bar{A}$, $\bar{V}$, $\bar{X}$ and Γ_1 as input at every control cycle and uses this information to estimate $\hat{m}$ (the optimal mass esstimate for the vehicle), which is then used as an input to the optimal controller as the m parameter. The new accelerometer sensor $\bar{A}$ is configured in the same way as were the other two sensors and is set to be a white noise process of the form: $\bar{A} \sim \mathcal{N}(A_0, \sigma_A^2)$, where A_0 is the true vehicle acceleration at a cycle. Thus, it is expected that as the Kalman filter converges to an optimal estimate of $\hat{m}$ the controller performance will improve and then become equivalent to the test cases where the mass was fully known.

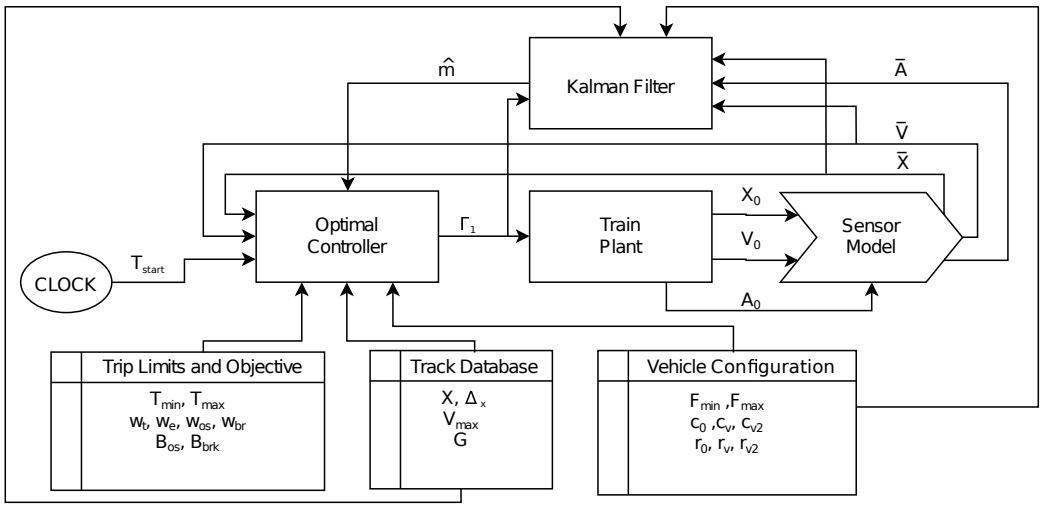


Figure 10.5: Simulation Model with Sensor Noise and KF Mass Estimator

10.3.1 Mass Estimation Model

A simple linear model is proposed for estimating mass based on *Eqn. 3.1c*— the dynamic model of the train. It should be noted that the index j for $j = 0 \ldots n$

corresponds to the optimization cycle count in the time domain, increasing at each successive re-optimization. Since *Eqn. 3.1c* is actually linear in $1/m$ and not m, a new variable e is introduced as:

$$e = \frac{1}{m} \tag{10.5}$$

This allows to treat the estimation problem linearly, and thus a linear KF can be used to estimate the inverse mass which can then be inverted to provide $\hat{m}$ to the optimal controller. For an optimal inverse mass estimate ($\hat{e}$) and the observed/estimated traction force (l) a model can be represented by the following state-space form which is linearly related to inverse mass as:

$$e_j = M_A e_{j-1} + M_B d + w_R \tag{10.6a}$$

$$l_j = M_{Cj} e_j + w_H \tag{10.6b}$$

where d is an arbitrary external input on the mass state-space equation. Since the mass is expected to be constant throughout a trip and no external input can adjust the mass, M_A and M_B are trivially selected as:

$$M_A = 1 \qquad M_B = 0 \tag{10.6c}$$

where M_{Cj} is:

$$M_{Cj} = \min(\Gamma_1^*, r_0 + r_v \bar{V} + r_{v2} \bar{V}^2) \tag{10.6d}$$

It should be noted here that using sensor measurements in M_{Cj} is a suboptimal strategy, and thus the Kalman Filter implementation is not truly optimal for this noise profile. Regardless of this, testing results with this model showed very good convergence of the mass estimate to the true mass. The objective of the KF is to minimize the error between the theoretical traction acceleration ($M_{Cj} e_j$) and the

measured traction acceleration (l_j), where the l_j can be evaluated through combining sensor outputs to create the virtual observation:

$$l_j = \bar{A} - G(\bar{X}) - c_0 - c_v\bar{V} - c_{v^2}\bar{V}^2 \tag{10.6e}$$

$\bar{X}$,$\bar{V}$ and $\bar{A}$ are measurements of present vehicle states and Γ_1^* is the current cycle's optimal control decision that was sent to the vehicle. Here, $w_R \sim \mathcal{N}(0, R)$ is the process noise and $w_H \sim \mathcal{N}(0, H)$ is the measurement noise.

10.3.2 Scalar Kalman Filter Model

With the above mass estimation model, a scalar KF can be formulated with the estimation model equations (Wolpert and Ghahramani, 2000):

The a priori prediction equations for the estimate and its covariance:

$$\hat{e}_{j|j-1} = M_A\hat{e}_{j-1|j-1} + M_B d_j \tag{10.7a}$$

$$E_{j|j-1} = M_A^2 E_{j-1|j-1} + H \tag{10.7b}$$

For the measurement residual (the difference between the theoretical and actual control acceleration virtual measurements):

$$\tilde{y}_j = l_j - M_{Cj}\hat{e}_{j|j-1} \tag{10.7c}$$

and its residual Covariance:

$$S_j = M_{Cj}^2 E_{j|j-1} + R \tag{10.7d}$$

The optimal Kalman gain to apply on the next estimate is computed as:

$$\chi_j = \frac{E_{j|j-1}M_{Cj}}{S_j} \tag{10.7e}$$

The updated (a posteriori) inverse mass estimate - the best estimate for the inverse vehicle mass thus far - is computed by combining an optimally weighted sum of the

previous best inverse mass estimate and the measurement residual:

$$\hat{e}_{j|j} = \hat{e}_{j|j-1} + \chi_j \tilde{y}_j \tag{10.7f}$$

The optimal mass estimate can then be evaluated by inverting $\hat{e}_j$:

$$\hat{m}_j = \frac{1}{\hat{e}_{j|j}} \tag{10.7g}$$

The new (a posteriori) estimate covariance is also updated for use during the next KF cycle:

$$E_{j|j} = (1 - \chi_j M_{Cj}) E_{j|j-1} \tag{10.7h}$$

The above equations must be successively resolved at each control cycle to provide an accurate estimate for the inverse vehicle mass. Thus, $\hat{e}_{j|j}$ and $E_{j|j}$ need to be stored and carried over into the $j+1$ time cycle.

10.3.3 Kalman Filter Implementation

Substituting the inverse mass estimation model (*Eqn. 10.5*) into the above scalar kalman filter formulation (*Eqn. 10.6*) results in the following simplified equations: The priori covariance is evaluated as:

$$E_{j|j-1} = E_{j-1|j-1} + H \tag{10.8a}$$

The scalar Kalman gain (χ) evaluates to:

$$\chi_j = \frac{\min(\Gamma_1^*, r_0 + r_v \bar{V} + r_{v2} \bar{V}^2) E_{j|j-1}}{\min(\Gamma_1^*, r_0 + r_v \bar{V} + r_{v2} \bar{V}^2)^2 E_{j|j-1} + R} \tag{10.8b}$$

This results in the posteriori estimate and covariance:

$$\hat{e}_{j|j} = \hat{e}_{j|j-1} + \chi((\bar{A} - G(\bar{X}) - c_0 - c_v \bar{V} - c_{v2} \bar{V}^2) - \min(\Gamma_1^*, r_0 + r_v \bar{V} + r_{v2} \bar{V}_0^2) \hat{e}_{j|j-1}) \tag{10.8c}$$

$$E_{j|j} = E_{j|j-1}(1 - \min(\Gamma_1^*, r_0 + r_v\bar{V} + r_{v2}\bar{V}^2)\chi_j) \tag{10.8d}$$

And finally, inverting the estimate to resolve the mass:

$$\hat{m}_j = \frac{1}{\hat{e}_{j|j}} \tag{10.8e}$$

$E_{0|0}$, R and H are initialized and tuned to affect the KF convergence rate, and $\hat{e}_{0|0}$ is set to be the inverse of the initial mass guess.

10.3.4 Mass Estimator Performance with No Measurement Noise

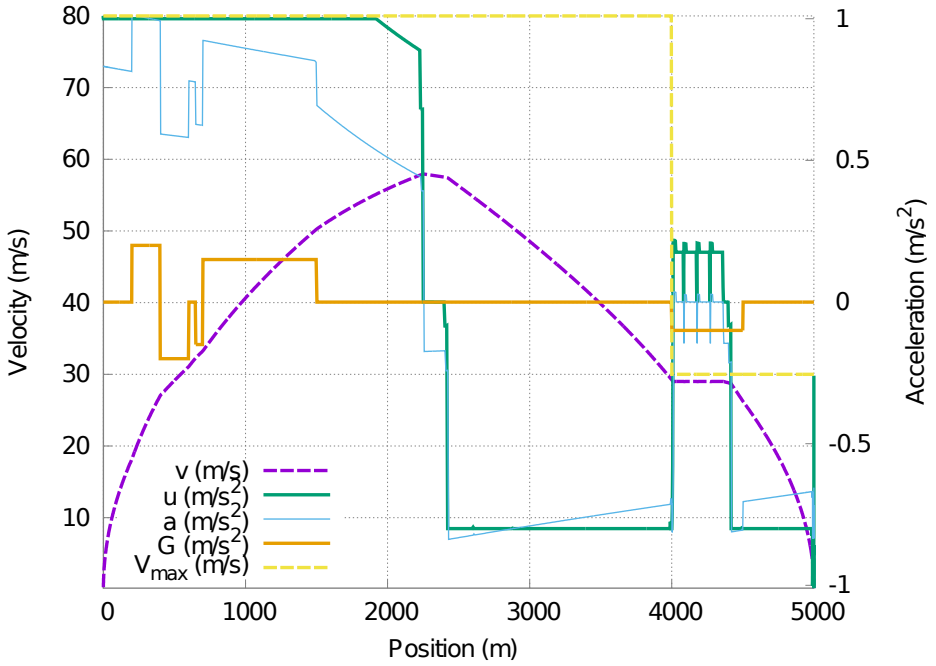
Fig. 10.6 demonstrates the effect of including the above KF formulation with the optimal controller for $E_{0|0} = 0.01$,$R = 0.01$ and $H = 10$. The initial mass guess is set to $\hat{m}_0 = 1 \times 10^6 kg$, with the actual simulated mass being $1.2 \times 10^6 kg$. At each cycle the new mass estimate is fed back into the controller as in *Fig. 10.5*. In *Fig. 10.7* it can be seen that $\hat{m}$ converges to within 1% error of the true vehicle mass in 1551m with the chosen selection of $E_{0|0}$,R and H. This is a case with no uncertainty and as such the noise model is configured as:

$$\sigma_a^2 = 0 \qquad \sigma_V^2 = 0 \qquad \sigma_X^2 = 0 \qquad \sigma_A^2 = 0$$

With the Kalman Filter in the loop acting as a suboptimal recursive least-squares estimator (Wolpert and Ghahramani, 2000), the live improvements to the parameter estimation are sufficient to remove over-speeding and the stopping overshoot is reduced from 35.74m to 0.1m as compared to the case where no mass estimation was performed (*Fig. (10.4)*).

This KF implementation is just a simple example of the significant benefits that can be gained when pairing control with a parameter estimator. In practice, a more

advanced implementation of optimal estimation should be used - one that can estimate all observable parameters and not just the mass.



$$x_f = 5km,\ T_{max} = 300s$$
$$w_t = 1,\ w_e = 0.1,\ w_{os} = 0.02,\ w_{br} = 1$$
$$T_f = 168.19s,\ E_f = 1,835.50J/kg,\ X_s = 5,000.101m$$

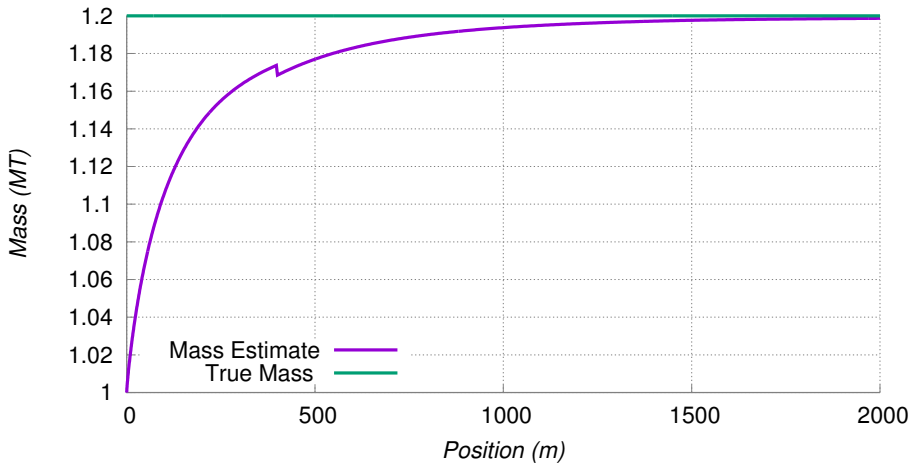
Figure 10.6: 20% Mass Error With Mass Estimator

10.3.5 Mass Estimator Performance with Measurement Noise

Since a KF is optimal under a white noise model and process noise, a test is conducted where the filter is run with sensor error. This example tests both the optimal controller's and the KF's performance when operating under uncertainty conditions. For the following test (*Fig. 10.8 and Fig. 10.9*) the sensor model is configured as:

$$\sigma_a^2 = 0.05m/s^2 \qquad \sigma_V^2 = 0.01m/s \qquad \sigma_X^2 = 0.01m \qquad \sigma_A^2 = 0.01m/s^2$$

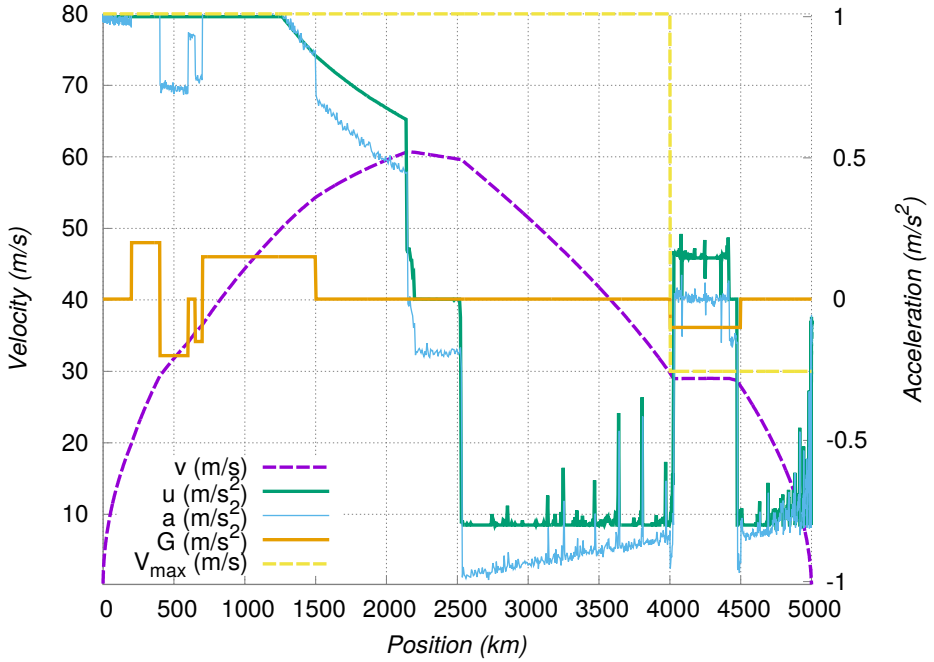
Figure 10.7: Kalman Filter Convergence



The KF is configured with the initial conditions: $E_{0|0} = 0.01$,$R = 0.01$ and $H = 10$. The initial mass guess is set to $\hat{m}_0 = 1.2 \times 10^6 kg$, with the actual simulated mass being $1 \times 10^6 kg$ (the reverse of the previous example).

As with the no noise case it can be seen that the KF accurately converges to the true mass in approximately 1km (*Fig. 10.9*). Driving performance is again quite good as can be seen in *Fig. 10.8*. Thus, it can be seen that, in practice, a suboptimal KF formulation that used noisy measurements in its parameter evolution equations still resulted in good overall driving performance.

These examples successfully demonstrate the benefits and potentials of pairing an optimal controller with an optimal parameter estimator for removing constant modeling uncertainty. Even a trivial scalar KF implementation was enough to not only eliminate unsafe driving when significant modeling error was presented, but was



$$x_f = 5km,\ T_{max} = 300s$$
$$w_t = 1,\ w_e = 0.1,\ w_{os} = 0.02,\ w_{br} = 1$$
$$T_f = 164.19s,\ E_f = 1,984.67J/kg,\ X_s = 4,999.782m$$

Figure 10.8: 20% Mass Error with Sensor Noise and Mass Estimator

also sufficient to reinstate optimal driving behavior. Thus a more advanced optimal parameter estimator, one that can estimate more than a single parameter - may be a good future research direction for optimal train control.

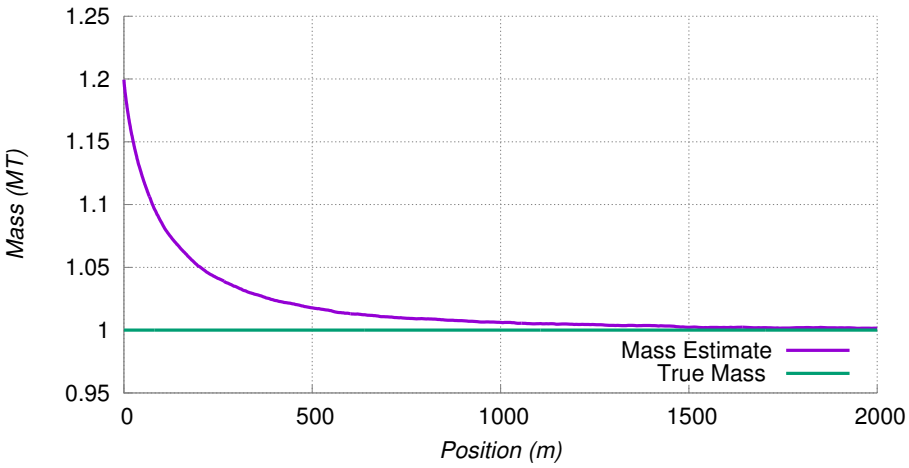


Figure 10.9: Kalman Filter Convergence

Chapter 11

Conclusion and Future Work

11.1 Conclusion

With the increasing computational capabilities of modern computer hardware, optimal control is rapidly becoming a very interesting prospective as more realistic vehicle models and more useful control problems can be solved in a reasonable amount of time. This thesis has focused on formulating and implementing an optimal controller that is usable for real-world conditions and trip objectives while being capable of in real-time controlling a rail vehicle in motion.

The controller is first formulated and tested as a non-convex optimal control problem, with IPOPT and NLOPT being used as numerical solvers. Although both solvers are capable of solving the optimal control problem to high degree of accuracy neither is well suited towards real-time control applications. As observed by other research in the field, numerical optimization strategies often suffer from computational complexity and are thus not readily real-time implementable, and thus other research has

largely focused on problem simplification, thus achieving optimal results for supoptimal problem formulations (Novak *et al.*, 2015). This is not apparent with the convex formulation in this thesis as the numerical optimization of the presented optimal control strategy is real-time capable due to the convexity, problem structure and efficient solving routines combined with a low-level embedded implementation of the control algorithm.

As compared to the analytical research with advanced models conducted by Albrecht *et al.* (2016a), the convexity of the optimal control problem allowed for solving the problem uniquely, robustly and quickly with a numerical solver in a single iteration, whereas Albrecht *et al.* (2016a) could not demonstrate real-time performance and employed numerous ad-hoc strategies (Albrecht *et al.*, 2016a). The research by Albrecht *et al.* (2015) was similar the research of Albrecht *et al.* (2016a) but was adapted to controlling two vehicles sharing the same track as a non-convex optimal control problem, but had an unspecified solution strategy (Albrecht *et al.*, 2016a, 2015). This research introduced the multi-vehicle control problem in a similar structure to Albrecht *et al.* (2015), but the convexity of the approach allowed for a fully real-time capable and robustly solvable control problem.

As compared to existing numerical control methods this research utilized advanced modeling to allow the controller to optimally work with highly complex trip configurations as is common in other numerical research methods. In contrast with other numerical methods this was done robustly, on-line and was very fast to compute; deployable simply via configuring the on-board model without requiring training as was the case with numerous numerical strategies (Matsuura and Miyatake, 2014; Vašak *et al.*, 2009; Chang and Xu, 2000).

Therefore, since analytical approaches to the train control problem have been known to be efficiently solvable yet lacking the necessary complexity for robust control (Ichikawa, 1968), the convex controller allows for an optimal control strategy that incorporates the best attributes of researched analytical and numerical train control methods: the computational efficiency and real-time capabilty of analytical train control research paired with high modeling accuracy and advanced trip objectives as seen in advanced numerical train control optimization research.

Given the extensive testing of the controller formulation as presented in this thesis in the convex case, the practical robustness of the method was demonstrated on a simulated model of a train system. This research strived to address as many variations of trip conditions while modeling the vehicle as accurately as possible. Great performance was demonstrated for simple and advanced trips alike. Performance for high speed rail and long range rail was likewise great. Demonstrated also was a case where several vehicles are sharing the track simultaneously with a safety buffer constraint between them. The controller proceeded to drive both trains optimally and safely. When subjected to disturbances, the controller performance did not degrade until very large plant/model mismatch was applied. For very large modeling errors it was paired with a parameter estimator, a combination that led to significantly improved performance resulting in the controller subsequently being able to stop the train with insignificantly small error.

Computational performance on both desktop processors and embedded ARM systems was shown to be very fast even with large control horizons. This opens up an interesting application for the controller in an embedded role. A highly robust system with a significant amount of redundancy can be cheaply and efficiently implemented

on a live vehicle. In this case, multiple ARM Linux machines - like the Hardkernel Odroid XU4 PCs - can be deployed, all of which run the same optimal controller having either the same objective or different objective weights. Therefore, in the event of a system failure the ATS module can select another physical instance of the ATO to drive the train, thus allowing for robust hard or soft failure recovery.

The same qualities could not be extended to the non-convex implementation as developed in the early stages of this research as it would not scale well and performance was largely unsuitable for real-world applications with both IPOPT and NLOPT:SLSQP implementations.

In conclusion, this research represents the viability of advanced optimal control for real-time applications through convex numerical optimization strategies. This is possible not only through efficient problem formulations but due to advancements in computational hardware and developments in evolved and efficient numerical convex solvers. It is hoped that this research assists others in realizing advanced and safety critical control problems as real-world implementable numerical optimization problems.

11.2 Future Work

The successful implementation of a convex optimal controller does allow for a significant amount of interesting research that would extend and improve the controller's capabilities and allow for faster adaptive control to a higher precision.

This research demonstrates the effectiveness of the proposed optimal control strategy in regulating a simulated rail vehicle as an embedded system. The next logical step is to install the embedded controller on one of Thales Canada Transport Solutions

vehicles for real-world performance testing. In this way, the controller performance can be further validated under real-world test and track conditions.

Also demonstrated were the gained benefits in driving performance when pairing the optimal controller to a simplistic real-time parameter estimator. A useful future research direction would be to focus on a more advanced parameter estimation algorithm - one capable of resolving most if not all of the model parameters at once. This can be a significant challenge due to the complexity of the vehicle dynamics model and the potential unobservability of some parameters under normal driving conditions. Additionally of a challenge is ensuring that no overfit in parameter estimation is apparent.

Optimal estimation paired with optimal control is prevalent for systems such as MPC and MHE paired systems where real-time estimation has been demonstrated for large-scale Quadratic Programming (QP) estimation problems (Vukov *et al.*, 2015). As was demonstrated in this thesis, a convex numerical optimization formulation proved to be well suited for real-time control. Perhaps a convex numerical optimization can be formulated for advanced parameter estimation that is equally robust, is an unbiased optimal estimator and allows for real-time parameter updates.

Another interesting future research direction is further improving the computational speed of ECOS for larger optimization problems. As was shown in *Fig. 7.4* and Domahidi *et al.* (2013), MOSEK outperforms ECOS for very large optimization horizons ($K > 1000$) (Domahidi *et al.*, 2013). As mentioned previously, this may largely be due to MOSEK using efficient hardware optimized BLAS and LAPACK routines that speed up large-scale linear algebra computation (MOSEK ApS, 2015). Therefore, modifying ECOS's rather minimal source to use something like cuSPARSE

instead of CSPARSE, may result in significant acceleration for large K due to GPU evaluations of high level matrix routines under the NVIDIA CUDA environment (Liu and Schmidt, 2015). In this manner, the precision for very long trips can be further improved with a decrease in computation time.

In conjunction, significant performance benefits may be gained through smarter selection of the Δ_x profile. In this thesis, the profile was naively set as a suboptimal trapeziodal profile so as to improve performace at station start and stop regions. A future direction can be to develop an optimal method of Δ_x selection based on the G and V_{max} profiles. This can allow for either a smaller control horizon dimension size (K) or a finer and more accurate sampling of X. In either case, better control performance may be apparent per millisecond of computation time than for a trapeziodal profile.

The scope of this research did not focus on cases where there may be a significant delay in controller actuation. Large delay has been extensively noted to be especially harmful to control in feedback systems (Brown and Coombs, 1991). Therefore a future research direction may be to incorporate controller delay straight into the optimal control strategy such that the controller can smartly handle large delays without adverse driving behavior.

The control problem, as implemented in this research, can also be extended to other types of vehicles with similar driving applications and objectives. The control objectives of automotive automatic cruise control systems come to mind and they require many of the control objectives demonstrated in this research. This optimal convex controller requires minimal modifications to fulfill the role of a cruise control system on a driver-less car, so long as there is a higher level control system that

is responsible for route mapping, and an additional system responsible for steering control.

Bibliography

Al-Gherwi, W., Budman, H., and Elkamel, A. (2013). A robust distributed model predictive control based on a dual-mode approach. *Computers and Chemical Engineering*, **50**, 130 – 138.

Albrecht, A., Howlett, P., Pudney, P., Vu, X., and Zhou, P. (2015). Optimal driving strategies for two successive trains on level track subject to a safe separation condition. In *Proceedings of the American Control Conference (ACC), 2015*, pages 2924–2929. IEEE.

Albrecht, A., Howlett, P., Pudney, P., Vu, X., and Zhou, P. (2016a). 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*, **94**, 482 – 508.

Albrecht, A., Howlett, P., Pudney, P., Vu, X., and Zhou, P. (2016b). 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*, **94**, 509–538.

Allotta, B., Chisci, L., D'Adamio, P., Papini, S., and Pugi, L. (2013). Design of

an automatic train operation (ato) system based on cbtc for the management of driverless suburban railways. In *Proceedings of the 12th IMEKO TC10 Workshop on Technical Diagnostics: New Perspective in Measurements, Tools and Techniques for Industrial Applications, Florence, Italy*, pages 84–89.

Amestoy, P., Ashcraft, C., Boiteau, O., Buttari, A., L'Excellent, J.-Y., and Weisbecker, C. (2015). Improving multifrontal methods by means of block low-rank representations. *SIAM Journal on Scientific Computing*, **37**(3), A1451–A1474.

Bombardier (2008). Ebi drive 50 driver assistance system. *Bombardier Energy Saving Technologies and their Applications*.

Boschetti, G. and Mariscotti, A. (2012). The parameters of motion mechanical equations as a source of uncertainty for traction systems simulation. In *Proceedings of the 20th Imeko World Congress, Busan South Korea*.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Brown, C. M. and Coombs, D. J. (1991). *Notes on control with delay*. University of Rochester [Department of] Computer Science.

Chang, C. and Xu, D. (2000). Differential evolution based tuning of fuzzy automatic train operation for mass rapid transit system. *IEEE Proceedings-Electric Power Applications*, **147**(3), 206–212.

Cordeau, J.-F., Toth, P., and Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation science*, **32**(4), 380–404.

Datta, S. (2012). Efficient genetic algorithm on linear programming problem for fittest chromosomes. *Journal of Global Research in Computer Science*, **3**(6), 1–7.

Diamond, S. and Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, **17**(83), 1–5.

Domahidi, A., Chu, E., and Boyd, S. (2013). ECOS: An SOCP solver for embedded systems. In *Proceedings of the European Control Conference (ECC)*, pages 3071–3076.

Eugene A. Avallone, T. B. (2006). *Marks Standard Handbook for Mechanical Engineers: pg. 166*, volume 11. McGraw-Hill.

European Commission: Transport Statistics (2014). *EU Transport in figures: Statistical pocketbook 2014.* Publications Office of the European Union.

Gupta, A. (2000). Wsmp: Watson sparse matrix package. Technical report, IBM.

Hardkernel (2017). *Hardkernel Odroid XU4 User Manual: rev. 20170310*. Found at www.hardkernel.com/main/products/.

Hiriart-Urruty, J.-B. and Lemaréchal, C. (2013). *Convex analysis and minimization algorithms I: fundamentals*, volume 305. Springer.

Hofestadt, H. (1995). Gsm-r: global system for mobile radio communications for railways. In *Proceedings of the International Conference on Electric Railways in a United Europe, 1995.*, pages 111–115. IET.

Howlett, P. G. and Pudney, P. J. (1995). *Energy-efficient train control*. Springer.

Huang, Y., Tan, L., Chen, L., and Tang, T. (2016). A neural network driving curve generation method for the heavy-haul train. *Advances in Mechanical Engineering*, **8**(5), 1687814016647883.

Ichikawa, K. (1968). Application of optimization theory for bounded state variable problems to the operation of train. *Bulletin of the Japanese Society of Mechanical Engineers*, **11**(47), 857–865.

Jahanshahi, E. and Skogestad, S. (2013). Closed-loop model identification and PID/PI tuning for robust anti-slug control. *IFAC Proceedings Volumes*, **46**(32), 233–240.

John, E. and Yıldırım, E. A. (2008). Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension. *Computational Optimization and Applications*, **41**(2), 151–183.

Johnson, S. G. (2014). *The NLopt nonlinear-optimization package*. Available at http://ab-initio.mit.edu/nlopt.

Ko, H., Koseki, T., and Miyatake, M. (2004). Application of dynamic programming to the optimization of the running profile of a train. *WIT Transactions on The Built Environment*, **74**.

Kosiński, R. A. and Kozłowski, C. (1998). Artificial neural networks–modern systems for safety control. *International Journal of Occupational Safety and Ergonomics*, **4**(3), 317–332.

Kraft, D. (1994). Algorithm 733: TOMP–Fortran modules for optimal control calculations. *ACM Transactions on Mathematical Software*, **20**(3), 262–281.

Kuhn, H. W. (2014). Nonlinear programming: a historical view. In *Traces and Emergence of Nonlinear Programming*, pages 393–414. Springer.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge University Press.

Leenen, M. and Wolf, A. (2016). Global rail market growth set to slow. *International Railway Journal*.

Liu, R. R. and Golovitcher, I. M. (2003). Energy-efficient operation of rail vehicles. *Transportation Research Part A: Policy and Practice*, **37**(10), 917 – 932.

Liu, Y. and Schmidt, B. (2015). LightSpMV: Faster CSR-based sparse matrix-vector multiplication on CUDA-enabled GPUs. In *Proceedings of the 2015 IEEE 26th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pages 82–89. IEEE.

Lobo, M. S., Vandenberghe, L., Boyd, S., and Lebret, H. (1998). Applications of second-order cone programming. *Linear algebra and its applications*, **284**(1-3), 193–228.

M. S. Andersen, J. D. and Vandenberghe, L. (2013). *CVXOPT: A python package for convex optimization, version 1.1.6*. Available at cvxopt.com.

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural language learning*, volume 20, pages 1–7. Association for Computational Linguistics.

Martí, R., Resende, M. G., and Ribeiro, C. C. (2013). Multi-start methods for combinatorial optimization. *European Journal of Operational Research*, **226**(1), 1–8.

Matagawa, N. and Shudo, K. (2016). Breakdown of a benchmark score without internal analysis of benchmarking program. *arXIV preprint:1610.06307*.

Matsuura, G. and Miyatake, M. (2014). Optimal train speed profiles by dynamic programming with parallel computing and the fine-tuning of mesh. *WIT Transactions on The Built Environment*, **135**, 767–777.

Morales, J. L., Nocedal, J., and Wu, Y. (2011). A sequential quadratic programming algorithm with an additional equality constrained phase. *IMA Journal of Numerical Analysis*, **32**, 553–579.

MOSEK ApS (2015). *The MOSEK Python optimizer API manual - Version 7.0 (Revision 141)*. Available at http://docs.mosek.com/7.0/pythonapi/index.html.

Novak, H., Vašak, M., and Lešić, V. (2015). State of the art control methods for energy efficient train operation in a railway traffic system. In *Proceedings of the 35th Conference on Transportation Systems with International Participation Automation in Transportation*.

Ouyang, P. and Dam, T. (2011). Position domain PD control: Stability and comparison. In *Proceedings of the 2011 IEEE International Conference on Information and Automation (ICIA)*, pages 8–13. IEEE.

ODonoghue, B., Chu, E., Parikh, N., and Boyd, S. (2016). Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, **169**(3), 1042–1068.

Pál, L. (2013). Comparison of multistart global optimization algorithms on the BBOB

noiseless testbed. In *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 1153–1160. Association for Computing Machinery.

Reibig, G. (2011). Computing abstractions of nonlinear systems. *IEEE Transactions on Automatic Control*, **56**(11), 2583–2598.

Renner, M., Gardner, G., and Alliance, A. (2015). *Global competitiveness in the rail and transit industry*. Worldwatch Institute.

Rochard, B. P. and Schmid, F. (2000). A review of methods to measure and calculate train resistances. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, **214**(4), 185–199.

Schifers, C. and Hans, G. (2000). IEEE standard for communications-based train control (CBTC) performance and functional requirements. In *Proceedings of the Vehicular Technology Conference*, pages 1581–1585.

Schittkowski, K. (1982). The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function. *Numerische Mathematik*, **38**(1), 83–114.

Tomlin, C., Pappas, G. J., and Sastry, S. (1998). Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, **43**(4), 509–521.

U.S. Department of Transportation (2009). *Preliminary National Rail Plan*. Federal Railroad Administration.

Vašak, M., Baotić, M., Perić, N., and Bago, M. (2009). Optimal rail route energy management under constraints and fixed arrival time. In *Proceedings of the 2009 European Control Conference*, pages 2972–2977. IEEE.

Vukov, M., Gros, S., Horn, G., Frison, G., Geebelen, K., Jørgensen, J. B., Swevers, J., and Diehl, M. (2015). Real-time nonlinear MPC and MHE for a large-scale mechatronic application. *Control Engineering Practice*, **45**, 64–78.

Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, **106**(1), 25–57.

Wang, Y., Ning, B., Cao, F., De Schutter, B., and van den Boom, T. J. (2011). A survey on optimal trajectory planning for train operations. In *Proceedings of the IEEE International Conference on Service Operations, Logistics, and Informatics (SOLI)*, pages 589–594. IEEE.

Welch, R. L. and Venayagamoorthy, G. K. (2006). Comparison of two optimal control strategies for a grid independent photovoltaic system. In *Proceedings of the 41st Annual 41st IEEE Industry Applications Conference, 2006*, volume 3, pages 1120–1127. IEEE.

Wolpert, D. M. and Ghahramani, Z. (2000). Computational principles of movement neuroscience. *Nature Neuroscience*, **3**, 1212–1217.

Xu, H. Z. and Wang, L. S. (2014). Nonlinear model predictive control for automatic train operation with actuator saturation and speed limit. In *Advances in*

Mechatronics and Control Engineering III, volume 678 of *Applied Mechanics and Materials*, pages 377–381. Trans Tech Publications.

Zhang, L. and Zhuan, X. (2015). Development of an optimal operation approach in the MPC framework for heavy-haul trains. *IEEE Transactions on Intelligent Transportation Systems*, **16**(3), 1391–1400.